Interpreting Graph-Based Sybil Detection Methods as Low-Pass Filtering

Satoshi Furutani, Toshiki Shibahara, Mitsuaki Akiyama, *Member, IEEE*, and Masaki Aida, *Senior Member, IEEE*

***Abstract*— Online social networks (OSNs) are threatened by Sybil attacks, which create fake accounts (also called Sybils) on OSNs and use them for various malicious activities. Therefore, Sybil detection is a fundamental task for OSN security. Most existing Sybil detection methods are based on the graph structure of OSNs, and various methods have been proposed recently. However, although almost all methods have been compared experimentally in terms of detection performance and noise robustness, theoretical understanding of them is still lacking. In this study, we show that existing graph-based Sybil detection methods can be interpreted in a unified framework of low-pass filtering. This framework enables us to theoretically compare and analyze each method from two perspectives: filter kernel properties and the spectrum of shift matrices. Our analysis reveals that the detection performance of each method depends on the effectiveness of the low-pass filtering. Furthermore, on the basis of the analysis, we propose a novel Sybil detection method called SybilHeat. Numerical experiments on synthetic graphs and real social networks demonstrate that SybilHeat performs consistently well on graphs with various structural properties. This study lays a theoretical foundation for graph-based Sybil detection and leads to a better understanding of Sybil detection methods.**

***Index Terms*— Online social networks, Sybil detection, graph signal processing.**

I. Introduction

ONLINE Social Networks (OSNs) are essential platforms for people to interact with each other, communicate information, and spread social influence. According to the Pew Research Center's report [1], about 70% of Americans were on Facebook in 2021, and seven in ten of them visited the site daily. However, OSNs are under threat from Sybil attacks, which create fake accounts (also called Sybils) on OSNs and use them for various malicious activities, such as distributing spam, phishing URLs, and malware, and manipulating public opinion and the stock market by spreading fake news. For example, Sybils have been exploited to propagate anti-vaccine messages [2], [3] and manipulate online political discussions [4], [5]. Therefore, Sybil detection is a fundamental task for OSN security.

The common Sybil detection approach is the graph-based approach that detects Sybils on the basis of the graph structure of OSNs (i.e., the friendship relation between users on OSNs). This approach is motivated by the following observation: Sybils tend to be densely connected to other Sybils and sparsely connected to benign users because malicious attackers can easily control the connection between Sybils while they cannot control the connection between Sybils and benign users [6]. Therefore, it is expected that one can distinguish between a Sybil region and a benign region by exploiting the graph structure of OSNs.

Most graph-based methods predict unknown node labels (Sybil or benign) by assigning a prior reputation score to each node using known node labels and then updating and propagating the reputation score locally on a graph. Two kinds of propagation algorithms are often used: random walk-based and loopy belief propagation-based. Random walk-based methods [7], [8], [9], [10] propagate the trust or badness score by random walks from known benign or Sybil nodes and rank the Sybil-likeness of unknown nodes. Loopy belief propagation methods [11], [12], [13], [14], [15] model the OSN structure as a pairwise Markov random field and compute the marginal distribution for each node (i.e., the probability that a node is Sybil) by a loopy belief propagation algorithm or its approximation.

However, although various Sybil detection methods have been proposed over the past decade, almost all methods have been compared just experimentally in terms of detection performance and noise robustness. Since experimental results often depend on experimental conditions, such as dataset properties and experimental settings, a good result for an experimental condition does not guarantee the same for other ones. To understand why and under what conditions each method works well, we need to compare them theoretically. To this end, in our previous work [16], we formulated the random walk with restart and the loopy belief propagation algorithm as low-pass filtering and attempted a theoretical comparison of the performance of random walk-based and loopy belief propagation-based Sybil detection methods. However, this work does not provide a comprehensive comparison of existing detection methods (only a comparison between CIA [7] and SybilBelief [11]), nor can it explain the differences in detection performance for differences in structural properties of graphs (such as degree heterogeneity and modularity).

Manuscript received 19 June 2022; revised 9 December 2022; accepted 9 January 2023. Date of publication 18 January 2023; date of current version 23 January 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Issa Traore. *(Corresponding author: Satoshi Furutani.)*

Satoshi Furutani is with NTT Social Informatics Laboratories, Tokyo 180-8585, Japan, and also with the Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan (e-mail: satoshi.furutani.ek@hco.ntt.co.jp).

Toshiki Shibahara and Mitsuaki Akiyama are with NTT Social Informatics Laboratories, Tokyo 180-8585, Japan (e-mail: toshiki.shibahara.de@hco.ntt.co.jp; akiyama@ieee.org).

Masaki Aida is with the Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan (e-mail: aida@tmu.ac.jp).

Digital Object Identifier 10.1109/TIFS.2023.3237364

In this study, extending our previous work [16], we show that existing representative graph-based Sybil detection methods (CIA [7], SybilRank [8], SybilWalk [10], SybilBelief [11], and SybilSCAR [15]) can be interpreted in a unified framework of low-pass filtering. This framework enables us to theoretically compare and analyze each method from two perspectives: filter kernel properties and the spectrum of shift matrices. Our analysis reveals that the detection performance of each method depends on how good the low-pass filtering is. In other words, for a Sybil detection method to perform well, 1) the filter kernel must properly emphasize (remove) low (high) frequency components, and 2) the low frequency eigenvectors of the shift matrix must have high community detectability. Furthermore, on the basis of the analysis, we propose a novel detection method, called SybilHeat, with the filter kernel and the shift matrix that satisfies the above two requirements. Our main contribution are summarized as follows:

- We present the low-pass filtering framework for theoretically comparing and analyzing Sybil detection methods and identify the requirements for high performance of a Sybil detection method.
- We propose a Sybil detection method called SybilHeat that performs consistently better than other methods on graphs with various structural properties.
- We demonstrate the validity of our analysis and the performance of the proposed detection method through numerical experiments on synthetic graphs generated by the stochastic block model (SBM) and real social networks.

The rest of this paper is organized as follows: In Section II and Section III, we present related work and preliminaries, respectively. We interpret the existing methods as low-pass filtering in Section IV. In Section V, we provide a theoretical comparison of the existing methods based on the interpretation and discuss our proposed method, SybilHeat. In Section VI, we evaluate the validity of our analysis and the performance of SybilHeat through numerical experiments. Finally, Section VII concludes our paper.

II. Related Work

In this section, we give a brief overview of Sybil detection methods. Random walk-based methods [7], [8], [9], [10], [17], [18], [19] detect Sybils by random walks from known labeled nodes on a graph. SybilGuard [17] and SybilLimit [18] detect Sybils using special random walks called random routes. In a normal random walk, the destination of a walker is randomly chosen for each step, whereas in random routes, the destination is predetermined by the permutation π_v for each node v. That is, random routes that enter from an edge e always exit from edge $\pi_v(e)$. An unlabeled node is approved as a benign node when the random routes originating from it intersect with the random routes from a known benign node. SybilInfer [19] builds a probabilistic model of benign regions and uses it to detect potential Sybil regions. SybilGuard, SybilLimit, and SybilInfer are not scalable to large OSNs and are not robust to label noise because they only use information from known benign nodes. CIA [7] propagates the badness score of each node by random walks with restart from known Sybil nodes. SybilRank [8] evaluates the trust score of each node by computing the landing probability of early-terminated random walks from known benign nodes. Íntegro [9] improves SybilRank by learning the edge weights and then considering random walks on the weighted graph. The random walk-based method described above has the limitation that only labeled benign nodes or labeled Sybil nodes can be used (not both). To overcome this problem, SybilWalk [10] computes the badness score of each node by random walks on the augmented graph with two additional nodes (Sybil label node and benign label node).

Loopy belief propagation methods [11], [12], [13], [14], [15] model the OSN structure as a pairwise Markov random field and compute the marginal distribution for each node (i.e., the probability that a node is Sybil) by a loopy belief propagation algorithm or its approximation. SybilBelief [11] first assigns the prior probability to each node using known node labels and then uses loopy belief propagation to calculate the posterior probability of them. Later studies [12], [13], [14] have demonstrated that learning and exploiting node and edge features improve the performance of Sybilbelief. SybilBelief and its variants rely on loopy belief propagation for inference, which is not scalable and has no convergence guarantees. Wang et al. [15] provided a general framework that integrates random walk-based and loopy belief propagation-based methods and proposed SybilSCAR, a random walk-like score propagation algorithm, by approximating the loopy belief propagation algorithm. SybilSCAR is more scalable than SybilBelief, and convergence is guaranteed. However, this framework does not provide theoretical insight into the performance of existing Sybil detection methods.

Other Sybil detection methods include behavior-based detection methods [20], [21], [22], [23], [24], [25], [26]. They often use machine learning to classify users into benign or Sybil on the basis of their social behavior. Most of them consist of two steps: 1) extracting behavior-based (and sometimes graph-based) features that contribute to Sybil detection (e.g., tweet content and timing, follower/followee information, node centrality, etc.), and then 2) constructing a detection model using the extracted features. A major limitation of behavior-based methods is that attackers can easily imitate the behavior of benign users, thereby compromising the effectiveness of the method.

III. Preliminaries

A. Graph Signal Processing

In this subsection, we briefly introduce the basic concepts of graph signal processing [27], [28]. Let $G = (V, E)$ be an unweighted undirected graph without self-loops and multiple edges, where $V = \{1, 2, \ldots, N\}$ is the node set and $E \subset V \times V$ is the edge set. A graph signal $x : V \to \mathbb{R}$ is the real-valued function defined on the node set V and is represented as N-dimensional vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$. A shift matrix $\boldsymbol{S} = [S_{ij}] \in \mathbb{R}^{N \times N}$ is a matrix such that the off-diagonal element $S_{ij} \neq 0$ iff $(i, j) \in E$. When the graph signal $\boldsymbol{x}$ is multiplied by the shift matrix $\boldsymbol{S}$, each element of the shifted signal $\tilde{\boldsymbol{x}} = \boldsymbol{S}\boldsymbol{x}$ is a linear combination of the signal value

of its adjacent nodes, that is, the original graph signal is shifted over the graph. In general, the adjacency matrix and Laplacian matrix are often used as the shift matrix [27], [28]. The adjacency matrix $\boldsymbol{A} = [A_{ij}] \in \mathbb{R}^{N\times N}$ is a real symmetric matrix defined as $A_{ij} = 1$ if $(i, j) \in E$ and $A_{ij} = 0$ otherwise. The Laplacian matrix is defined as $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A}$ where $\boldsymbol{D} := \mathrm{diag}(d_1, d_2, \ldots, d_N)$ is the degree matrix and $d_i := \sum_{i=1}^{N} A_{ij}$ is node i's degree.

We define the diagonal matrix $\boldsymbol{\Lambda} := \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$ with eigenvalues $\lambda_1 \le \lambda_2 \le \cdots \le \lambda_N$ of $\boldsymbol{S}$ and the matrix $\boldsymbol{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_N)$ with the eigenvector $\boldsymbol{v}_\mu$ corresponding to λ_μ. The graph Fourier transform (GFT) of $\boldsymbol{x}$ is defined as $\hat{\boldsymbol{x}} := \boldsymbol{V}^{-1}\boldsymbol{x}$ and inverse GFT is $\boldsymbol{x} := \boldsymbol{V}\hat{\boldsymbol{x}}$. The graph filtering (also called graph convolution) of input signal $\boldsymbol{x}_{\mathrm{in}}$ is defined as

$$\boldsymbol{x}_{\mathrm{out}} = \boldsymbol{V}h(\boldsymbol{\Lambda})\boldsymbol{V}^{-1}\boldsymbol{x}_{\mathrm{in}}, \tag{1}$$

where $h(\boldsymbol{\Lambda}) := \mathrm{diag}(h(\lambda_1), h(\lambda_2), \ldots, h(\lambda_N))$ and $h(\lambda)$ is a filter kernel function defined on the region $[\lambda_1, \lambda_N]$. As with filtering in the classical signal processing, the graph filtering operation is interpreted as transforming the graph signal into the frequency domain signal by GFT, multiplying filter $h(\lambda)$, and then transforming back into the graph signal by the inverse GFT. This outputs a signal in which specific frequency components of the input signal are amplified or attenuated.

B. Stochastic Block Model

A typical structural feature of real-world social networks is the existence of community structure. Roughly speaking, the community is a group of nodes that are densely connected within a group and sparsely connected between groups. One of the most basic models for generating random graphs with communities is the SBM [29].

Denoting k communities by $\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_k$, SBM assumes node i and node j are connected with the probability

$$\Pr(A_{ij} = 1) = \frac{C_{l(i),l(j)}}{N}, \tag{2}$$

where the symmetric matrix $\boldsymbol{C} = [C_{ab}] \in \mathbb{R}^{k\times k}$ is the connectivity matrix and C_{ab}/N is the probability of the edge being connected between nodes belonging to $\mathscr{C}_a$ and $\mathscr{C}_b$. The map $l : V \to \{1, 2, \ldots, k\}$ assigns a community to each node. As a special case, let us consider the SBM with two symmetric communities ($|\mathscr{C}_1| = |\mathscr{C}_2| = N/2$) and denote $C_{ab} = c_{\mathrm{in}}$ if $a = b$ and $C_{ab} = c_{\mathrm{out}}$ if $a \ne b$. In this case, it was conjectured in [30] that two communities are detectable if and only if the inequality

$$\frac{c_{\mathrm{in}} - c_{\mathrm{out}}}{2} > \sqrt{\frac{c_{\mathrm{in}} + c_{\mathrm{out}}}{2}} \tag{3}$$

holds, and it was proved in [31] and [32].

The main limitation of the SBM is that all nodes within each community have the same average degree. Degree-Corrected SBM (DCSBM) [33] is a more realistic model, which takes into account the degree heterogeneity of nodes within a community. DCSBM connects node i and node j with the probability

$$\Pr(A_{ij} = 1) = \theta_i\theta_j \frac{C_{l(i),l(j)}}{N}, \tag{4}$$

where θ_i is the intrinsic connectivity of node i. For each node i, θ_i is randomly sampled from the distribution $p(\theta)$ with $\mathbb{E}[\theta] = 1$ and $\mathbb{E}[\theta^2] = \Phi$. The intrinsic connectivity is proportional to the expected node degree (i.e., $\mathbb{E}[d_i] \propto \theta_i$) and produces an arbitrary degree distribution. DCSBM includes SBM as a special case where $\forall i.\ \theta_i = 1$. In [34], the detectable condition (3) for SBM is generalized to DCSBM as

$$\frac{c_{\mathrm{in}} - c_{\mathrm{out}}}{2} > \sqrt{\frac{c_{\mathrm{in}} + c_{\mathrm{out}}}{2\Phi}}. \tag{5}$$

IV. Interpretation of Sybil Detection as Low-Pass Filtering

In this section, we explain how to interpret existing graph-based Sybil detection methods as low-pass filtering. For an undirected graph $G = (V, E)$, let $V_s \subset V$ be the set of labeled Sybil nodes and $V_b \subset V$ be the set of labeled benign nodes. Given a prior reputation score $\boldsymbol{q} = (q_1, q_2, \ldots, q_N)^\top$ and a graph G, existing graph-based Sybil detection methods can be understood as methods that iteratively update the reputation score $\boldsymbol{p}^{(t)} = (p_1^{(t)}, p_2^{(t)}, \ldots, p_N^{(t)})^\top$ at step t following a certain update rule

$$\boldsymbol{p}^{(t)} = f\left(\boldsymbol{p}^{(t-1)}; \boldsymbol{q}, G\right) \tag{6}$$

until convergence, and then predict a label of each node using the final score $\boldsymbol{p} = \lim_{t\to\infty} \boldsymbol{p}^{(t)}$ [35]. The prior reputation score $\boldsymbol{q}$ and the update rule $f(\cdot)$ differ from method to method.

We here consider reformulating (6) as the low-pass filtering

$$\boldsymbol{p} = \boldsymbol{V}h(\boldsymbol{\Lambda})\boldsymbol{V}^{-1}\boldsymbol{q}, \tag{7}$$

where $\boldsymbol{\Lambda}$ and $\boldsymbol{V}$ are the diagonal matrix of eigenvalues of a shift matrix $\boldsymbol{S}$ and the invertible matrix consisting of its eigenvectors, respectively. $h(\cdot)$ is the low-pass filter kernel. This formulation gives a low-pass filtering interpretation to the existing Sybil detection methods and enables a theoretical comparison between them. Hereafter, we describe the low-pass filtering interpretation of the following representative Sybil detection methods: CIA [7], SybilRank [8], SybilWalk [10], SybilBelief [11], and SybilSCAR [15]. Note that, for simplicity, we here consider unweighted undirected graphs, but our approach is easy to extend to weighted ones.

A. CIA

CIA [7] propagates the badness score of each node by random walks with restart from labeled Sybil nodes V_s. For a restart parameter $0 < \alpha < 1$, the update rule is given by

$$\boldsymbol{p}^{(t)} = \alpha\boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{p}^{(t-1)} + (1-\alpha)\boldsymbol{p}^{(0)}. \tag{8}$$

The initial score of node i is set to $p_i^{(0)} = 1$ if $i \in V_s$ and $p_i^{(0)} = 0$ if $i \notin V_s$. Denoting $\boldsymbol{q} = \boldsymbol{p}^{(0)}$, we have the fixed

point $\boldsymbol{p}$ of (8) as

$$\begin{aligned}\boldsymbol{p} &= (1-\alpha)(\boldsymbol{I}-\alpha\boldsymbol{A}\boldsymbol{D}^{-1})^{-1}\boldsymbol{q}\\ &= (1-\alpha)(\boldsymbol{I}-\alpha(\boldsymbol{I}-\boldsymbol{\mathcal{L}}_{\mathrm{rw}}))^{-1}\boldsymbol{q}\\ &= \boldsymbol{V}_{\mathrm{r}}(1-\alpha)(\boldsymbol{I}-\alpha(\boldsymbol{I}-\boldsymbol{\Lambda}_{\mathrm{r}}))^{-1}\boldsymbol{V}_{\mathrm{r}}^{-1}\boldsymbol{q}, \end{aligned} \tag{9}$$

where $\boldsymbol{\Lambda}_{\mathrm{r}}$ and $\boldsymbol{V}_{\mathrm{r}}$ are matrices consisting of eigenvalues and eigenvectors of the random walk Laplacian $\boldsymbol{\mathcal{L}}_{\mathrm{rw}} := \boldsymbol{I}-\boldsymbol{A}\boldsymbol{D}^{-1}$, respectively.

B. SybilRank

SybilRank [8] evaluates the trust score of each node by computing the landing probability of early-terminated random walks from labeled benign nodes V_b. This is motivated by the hypothesis that since the connection between Sybil and benign nodes is sparse, a random walk starting from a benign node and terminating in a finite step is less likely to reach a Sybil node, and thus the landing probability is higher for benign nodes and lower for Sybil nodes. Setting the initial score to $p_i^{(0)} = 1/|V_b|$ if $i \in V_b$ and $p_i^{(0)} = 0$ if $i \notin V_b$, the trust score $\boldsymbol{p}^{(t)}$ is updated by

$$\boldsymbol{p}^{(t)} = \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{p}^{(t-1)}. \tag{10}$$

SybilRank calculates the final trust score by terminating the above update equation at a finite step $\Gamma = O(\log N)$ and then normalizing the trust score by the degree to eliminate the degree bias (i.e., $p_i = p_i^{(\Gamma)}/d_i$). Since $\boldsymbol{p}^{(\Gamma)} = (\boldsymbol{A}\boldsymbol{D}^{-1})^{\Gamma}\boldsymbol{p}^{(0)}$, we have the final trust score

$$\boldsymbol{p} = \boldsymbol{D}^{-1}(\boldsymbol{I}-\boldsymbol{\mathcal{L}}_{\mathrm{rw}})^{\Gamma}\boldsymbol{q} = \boldsymbol{D}^{-1}\boldsymbol{V}_{\mathrm{r}}(\boldsymbol{I}-\boldsymbol{\Lambda}_{\mathrm{r}})^{\Gamma}\boldsymbol{V}_{\mathrm{r}}^{-1}\boldsymbol{q}, \tag{11}$$

with $\boldsymbol{q} = \boldsymbol{p}^{(0)}$. Therefore, SybilRank can be interpreted as the operation combining the low-pass filtering by $\boldsymbol{\mathcal{L}}_{\mathrm{rw}}$ and degree-normalization.

C. SybilWalk

SybilWalk [10] computes the badness score of each node by random walks on the augmented graph $\widehat{G} = (V \cup \{l_s, l_b\}, \widehat{E})$, obtained by adding two label nodes (Sybil label node l_s and benign label node l_b) to an original graph G. For the augmented graph $\widehat{G}$, label nodes l_s and l_b are respectively connected to known Sybil nodes and known benign nodes (i.e., $\widehat{E} = E \cup \{(i, l_b) \mid i \in V_b\} \cup \{(i, l_s) \mid i \in V_s\}$). The badness score for each node $i \in V$ is calculated as the probability that a random walk starting from node i will reach l_s before reaching l_b as follows:

$$p_i^{(t)} = \sum_{j \in \widehat{\partial} i} \frac{a_{ij}}{\widehat{d}_i} p_j^{(t-1)}, \tag{12}$$

where $\widehat{d}_i$ is the degree of node i in the augmented graph $\widehat{G}$ (i.e., $\widehat{d}_i = d_i + 1$ if $i \in V_b \cup V_s$ and $\widehat{d}_i = d_i$ otherwise) and $\widehat{\partial} i$ is the set of neighbors of node i in $\widehat{G}$. The badness scores of label nodes are given by $p_{l_b} = 0$ and $p_{l_s} = 1$.

Indeed, SybilWalk is equivalent to an absorbing Markov chain with l_b and l_s as absorbing nodes. For a random walk on $\widehat{G}$, the transition matrix between user nodes is $\widehat{\boldsymbol{D}}^{-1}\boldsymbol{A} \in \mathbb{R}^{N \times N}$ where $\widehat{\boldsymbol{D}} := \mathrm{diag}(\widehat{d}_1, \widehat{d}_2, \ldots, \widehat{d}_N)$, and the transition matrix from user nodes to label nodes is given by $\boldsymbol{Q} = (\boldsymbol{q}_b, \boldsymbol{q}_s) \in \mathbb{R}^{N \times 2}$. Here, each component of $\boldsymbol{q}_s$ is defined as $q_{si} = 1/\widehat{d}_i$ if $i \in V_s$ and $q_{si} = 0$ if $i \notin V_s$, and $\boldsymbol{q}_b$ is defined in the same way. Hence, (12) is rewritten as $\boldsymbol{p}^{(t)} = \boldsymbol{\Pi}\boldsymbol{p}^{(t-1)}$ by using the transition matrix

$$\boldsymbol{\Pi} := \left(\begin{array}{c|c} \widehat{\boldsymbol{D}}^{-1}\boldsymbol{A} & \boldsymbol{Q} \\ \hline \boldsymbol{O} & \boldsymbol{I}_2 \end{array}\right),$$

where $\boldsymbol{I}_2$ is the 2×2 identity matrix. Therefore, we have

$$\begin{aligned}\boldsymbol{p} &= \lim_{t\to\infty} \boldsymbol{\Pi}^t \boldsymbol{p}^{(0)} = \left(\begin{array}{c|c} \boldsymbol{O} & (\boldsymbol{I}-\widehat{\boldsymbol{D}}^{-1}\boldsymbol{A})^{-1}\boldsymbol{Q} \\ \hline \boldsymbol{O} & \boldsymbol{I}_2 \end{array}\right)\begin{pmatrix}\vdots\\ 0\\ 1\end{pmatrix}\\ &= (\boldsymbol{I}-\widehat{\boldsymbol{D}}^{-1}\boldsymbol{A})^{-1}\boldsymbol{q}_s = \boldsymbol{V}_{\mathrm{a}}\boldsymbol{\Lambda}_{\mathrm{a}}^{-1}\boldsymbol{V}_{\mathrm{a}}^{-1}\boldsymbol{q}_s, \end{aligned} \tag{13}$$

where $\boldsymbol{\Lambda}_{\mathrm{a}}$ and $\boldsymbol{V}_{\mathrm{a}}$ are matrices consisting of eigenvalues and eigenvectors of the augmented normalized Laplacian $\boldsymbol{\mathcal{L}}_{\mathrm{aug}} := \boldsymbol{I} - \widehat{\boldsymbol{D}}^{-1}\boldsymbol{A}$, respectively.

D. SybilBelief

SybilBelief [11] models the OSN structure as a pairwise Markov random field and computes the marginal distribution for each node (i.e., the probability that a node is Sybil) by the standard loopy belief propagation [36]. Let us associate a random variable $s_i \in \{-1, +1\}$ to each node $i \in V$. $s_i = +1$ means that node i is Sybil, and $s_i = -1$ means that it is benign. The pairwise Markov random field is defined as

$$p(s_1, s_2, \ldots, s_N) = \frac{1}{Z}\prod_{(i,j)\in E}\psi_{ij}(s_i, s_j)\prod_{i\in V}\phi_i(s_i), \tag{14}$$

where Z is a normalization constant (called partition function), and $\phi_i(s_i)$ and $\psi_{ij}(s_i, s_j)$ are node and edge potential functions defined as follows, respectively:

$$\phi_i(s_i) = \begin{cases} q_i & \text{if } s_i = +1 \\ 1-q_i & \text{if } s_i = -1 \end{cases}, \tag{15}$$

$$\psi_{ij}(s_i, s_j) = \begin{cases} w_{ij} & \text{if } s_i s_j = +1 \\ 1-w_{ij} & \text{if } s_i s_j = -1, \end{cases} \tag{16}$$

where $\vec{E}$ is the set of oriented edges of E and satisfies $|\vec{E}| = 2|E|$. We can determine whether a node i is Sybil or not by evaluating the marginal distribution $p_i(s_i)$. However, it is exponentially hard to compute the marginal distribution directly from the joint distribution in (14). The loopy belief propagation algorithm is a common method to calculate an approximate marginal distribution $b_i(s_i) \approx p_i(s_i)$. This algorithm iteratively updates the probability distribution (called message) $\mu_{ij}(s_j)$ for each directed edge $(i, j) \in \vec{E}$. The message from node i to node j at step $t+1$ is given by

$$\mu_{ij}^{(t+1)}(s_j) = \frac{1}{Z_{ij}}\sum_{s_i=\pm 1}\phi_i(s_i)\,\psi_{ij}(s_i, s_j)\prod_{k\in\partial_{i\setminus j}}\mu_{ki}^{(t)}(s_i), \tag{17}$$

where $\partial_{i\setminus j} := \partial i \setminus \{j\}$ is the set of neighbors of node i excluding recipient node j. By using a converged message

μ_{ij}^{∞}, the approximate marginal distribution $b_i(s_i)$ is computed as

$$b_i(s_i) = \frac{1}{Z_i}\phi_i(s_i)\prod_{k\in\partial i}\mu_{ki}^{\infty}(s_i). \tag{18}$$

If G is a tree (G has no loops), $b_i(s_i)$ is exactly equal to $p_i(s_i)$. If G has loops, $b_i(s_i)$ is not equal to $p_i(s_i)$ but often provides a good approximation [37].

Since the loopy belief propagation algorithm is nonlinear, we have to linearize (17) to represent it as (7). In our previous study [16], we linearize (17) around a fixed point and reformulate SybilBelief as low-pass filtering by using a Bethe-Hessian matrix. The Bethe-Hessian matrix is defined as

$$\boldsymbol{H}(r) := (r^2-1)\boldsymbol{I} + \boldsymbol{D} - r\boldsymbol{A}, \tag{19}$$

with the parameter $r \in \mathbb{R}$. When $r = 1$, the Bethe-Hessian matrix becomes the Laplacian matrix $\boldsymbol{L}$. Hereafter, unless otherwise noted, we set the parameter r of the Bethe-Hessian $\boldsymbol{H}(r)$ is set to $r = [(\sum_i d_i^2)/(\sum_i d_i) - 1]^{1/2}$ as in [38]. This setting has the advantage that informative eigenvalues are negative while the bulk of uninformative eigenvalues is positive, making them easy to distinguish between them.

The low-pass filtering interpretation of SybilBelief is given by

$$\boldsymbol{p} = \boldsymbol{V}_{\mathrm{H}}\, g(\boldsymbol{\Lambda}_{\mathrm{H}})\boldsymbol{V}_{\mathrm{H}}^{-1}\boldsymbol{q}, \tag{20}$$

where $\boldsymbol{\Lambda}_{\mathrm{H}}$ and $\boldsymbol{V}_{\mathrm{H}}$ are matrices consisting of eigenvalues and eigenvectors of $\boldsymbol{H}(r)$, respectively. The function $g(\lambda)$ is the ideal low-pass filter kernel; i.e., $g(\lambda) = 1$ if $\lambda \le \lambda'$ and $g(\lambda) = 0$ otherwise. For details on the derivation of (20), see Appendix.

E. SybilSCAR

SybilBelief has the limitation of low scalability and no convergence guarantee of (17). To overcome these limitations, SybilSCAR [15] computes the probability p_i of each node i being Sybil by approximating (17) by replacing $\partial_{i\setminus j}$ with ∂i. SybilSCAR assigns the prior probability q_i to each node as

$$q_i = \begin{cases} 0.5+\theta & \text{if } i \in V_b \\ 0.5-\theta & \text{if } i \in V_s \\ 0.5 & \text{otherwise} \end{cases},$$

where $\theta \in (0, 0.5]$ indicates assigning high prior probabilities to labeled Sybil nodes. For a variable y, let us define a residual variable $\check{y} := y - 1/2$. The update function of SybilSCAR is given by

$$\check{\boldsymbol{p}}^{(t)} = 2\check{\boldsymbol{W}}\check{\boldsymbol{p}}^{(t-1)} + \check{\boldsymbol{q}}, \tag{21}$$

where $\check{\boldsymbol{W}} = (\check{w}_{ij})$ is a residual weight matrix with the element $\check{w}_{ij} = 0$ if $(i, j) \notin E$, and we set to $\check{\boldsymbol{p}}^{(0)} = \check{\boldsymbol{q}}$.

In [15], two SybilSCAR algorithms are proposed: SybilSCAR-C and SybilSCAR-D. For an edge $(i, j) \in E$, SybilSCAR-C has a constant residual weight (i.e., $\check{w}_{ij} = 1/(2d_{\max})$), while SybilSCAR-D has a degree-normalized residual weight (i.e., $\check{w}_{ij} = 1/(2d_j)$). For a fixed point $\check{\boldsymbol{p}}$ of (21), we have $\check{\boldsymbol{p}} = 2\check{\boldsymbol{W}}\check{\boldsymbol{p}} + \check{\boldsymbol{q}} = (\boldsymbol{I} - 2\check{\boldsymbol{W}})^{-1}\check{\boldsymbol{q}}$, and thus SybilSCAR-C is rewritten as

$$\check{\boldsymbol{p}} = \left(\boldsymbol{I} - \frac{1}{d_{\max}}\boldsymbol{A}\right)^{-1}\check{\boldsymbol{q}} = \boldsymbol{V}_{\mathrm{m}}\boldsymbol{\Lambda}_{\mathrm{m}}^{-1}\boldsymbol{V}_{\mathrm{m}}^{-1}\check{\boldsymbol{q}}, \tag{22}$$

where $\boldsymbol{\Lambda}_{\mathrm{m}}$ and $\boldsymbol{V}_{\mathrm{m}}$ are matrices consisting of eigenvalues and eigenvectors of the maximum degree-normalized Laplacian $\boldsymbol{\mathcal{L}}_{\max} := \boldsymbol{I} - \frac{1}{d_{\max}}\boldsymbol{A}$, respectively. Also, SybilSCAR-D is rewritten as

$$\check{\boldsymbol{p}} = \left(\boldsymbol{I} - \boldsymbol{A}\boldsymbol{D}^{-1}\right)^{-1}\check{\boldsymbol{q}} = \boldsymbol{V}_{\mathrm{r}}\boldsymbol{\Lambda}_{\mathrm{r}}^{-1}\boldsymbol{V}_{\mathrm{r}}^{-1}\check{\boldsymbol{q}}. \tag{23}$$

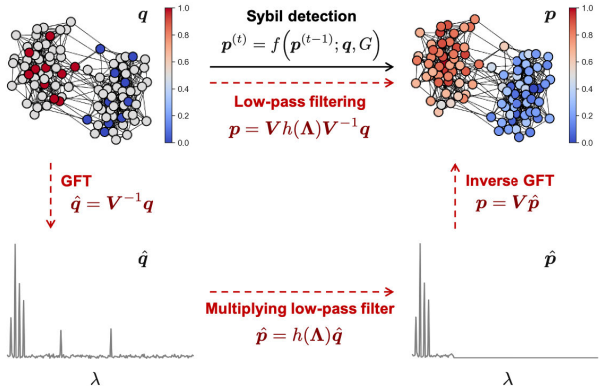

Fig. 1. Schematic diagram of low-pass filtering interpretation of Sybil detection.

TABLE I
SUMMARIZATION OF LOW-PASS FILTERING INTERPRETATION OF REPRESENTATIVE SYBIL DETECTION METHODS

Method	Shift matrix $\boldsymbol{S}$	Filter kernel $h(\lambda)$
CIA	$\boldsymbol{\mathcal{L}}_{\mathrm{rw}}$	$(1-\alpha)/(1-\alpha(1-\lambda))$
SybilRank	$\boldsymbol{\mathcal{L}}_{\mathrm{rw}}$	$(1-\lambda)^{\Gamma}$
SybilWalk	$\boldsymbol{\mathcal{L}}_{\mathrm{aug}}$	$1/\lambda$
SybilSCAR-C	$\boldsymbol{\mathcal{L}}_{\max}$	$1/\lambda$
SybilSCAR-D	$\boldsymbol{\mathcal{L}}_{\mathrm{rw}}$	$1/\lambda$
SybilBelief	$\boldsymbol{H}(r)$	ideal low-pass filter

V. THEORETICAL COMPARISONS

In Section IV, we described the low-pass filtering interpretation of some representative Sybil detection methods. As shown in Table I, the differences between these methods can be attributed to the differences in their corresponding shift matrix and filter kernel. Low-pass filtering is an operation consisting of the following three steps (see Fig. 1): (i) transforming a graph signal into a frequency signal by the GFT defined by the spectrum of a certain shift matrix, (ii) extracting (removing) low (high) frequency components by multiplying a low-pass filter to it, and then (iii) transforming it back to a (modified) graph signal by the inverse GFT. Thus, the output of low-pass filtering depends on the property of the low-pass filter kernel and the choice of shift matrix (i.e., what Fourier basis is used for the frequency transform). As is well known in the context of spectral clustering and graph signal processing, the eigenvectors corresponding to the small (low frequency) eigenvalues of the Laplacian contain rich information about the

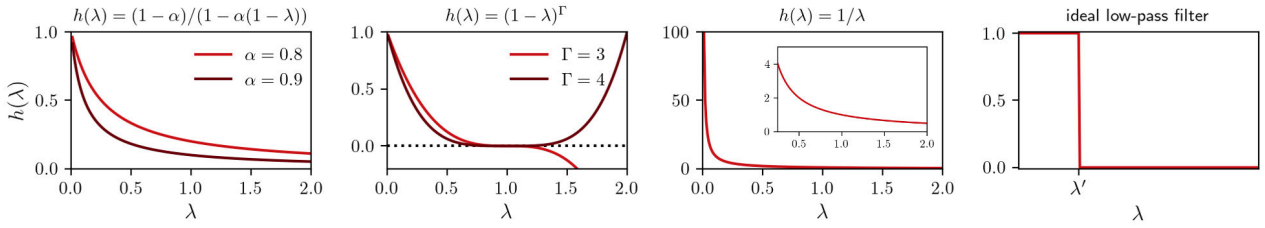

Fig. 2. Low-pass filter kernels of existing Sybil detection methods.

global community structure of a graph, while the eigenvectors corresponding to large (high frequency) eigenvalues contain noisy information [27], [39]. Thus, the performance of the Sybil detection method is expected to depend on how well the low pass filtering can extract the low frequency components and remove the high frequency components of the input signal. In this section, we compare and analyze each method from two perspectives: filter kernel properties and the spectrum of the shift matrix. Furthermore, on the basis of the theoretical insights, we propose a novel Sybil detection method called SybilHeat.

A. Filter Kernel Properties

Figure 2 plots the four different filter kernels in Table I. First, the CIA filter kernel $h(\lambda) = (1-\alpha)/(1-\alpha(1-\lambda))$ does not remove high frequency components sufficiently, so the output signal may be affected by noisy high frequency components. For this reason, CIA is expected to have poor detection performance and noise robustness.

Next, the SybilRank filter kernel $h(\lambda) = (1-\lambda)^{\Gamma}$ removes frequencies in the middle range ($0.5 \leq \lambda \leq 1.5$) but passes high frequency components ($\lambda > 1.5$). Therefore, if the largest eigenvalue λ_N takes a large value, SybilRank may be strongly affected by high frequency components. As described below, since the largest eigenvalue λ_N^{r} of the random walk Laplacian tends to become larger for a sparse graph, SybilRank is expected to perform poorly on sparse graphs.

The filter kernel $h(\lambda) = 1/\lambda$ strongly emphasizes low frequency components, and thus the contribution of high frequency components is relatively small. Since $h(\lambda) \to \infty$ for $\lambda \to 0$, the contribution of the eigenvector $\boldsymbol{v}_1$ corresponding to the smallest eigenvalue, which is uninformative in general, is dominant. Particularly, since the smallest eigenvalue of $\boldsymbol{\mathcal{L}}_{\mathrm{rw}}$ is 0, SybilSCAR-D may fail detection. However, if low frequency eigenvalues and high frequency eigenvalues are sufficiently separated, that is, the eigengap between informative and uninformative eigenvalues, $|\lambda_k - \lambda_{k+1}|$, is sufficiently large, high detection performance and noise robustness are expected.

The filter kernel corresponding to SybilBelief equally extracts the low frequency components and completely removes the high frequency components, by definition. Therefore, SybilBelief is expected to exhibit high detection performance and noise robustness as long as the low frequency eigenvectors are informative.

B. Spectrum of Shift Matrices

The output of low-pass filtering for a graph signal depends on how the frequencies (eigenvalues) are distributed and what Fourier basis is used for frequency transformation (i.e., the choice of shift matrix). We here discuss each method focusing on the spectrum (eigenvalues and eigenvectors) of the shift matrix.

1) Eigenvalues of Shift Matrices: First, we discuss detection methods in terms of eigenvalues of the shift matrix. Although frequencies are sampled evenly in classical signal processing, in graph signal processing, however, the frequency (eigenvalue) distribution is uneven and differs depending on the shift matrix. Since the quality of the low-pass filtering is determined by how well it can extract low frequency components, it is anticipated that the more clearly low frequency eigenvalues are isolated from the bulk of high frequency eigenvalues on the eigenvalue distribution, the better the low-pass filtering is. Figure 3 shows the spectral distribution of shift matrices for a dense and sparse modular graph generated by SBM. For dense modular graphs, k small (low frequency) eigenvalues are clearly isolated from the bulk of high frequency eigenvalues for each shift matrix. On the other hand, for sparse modular graphs, the bulk of uninformative eigenvalues is spread out, making them difficult to distinguish informative and uninformative eigenvalues. This suggests that the high frequency components are difficult to sufficiently remove by low-pass filtering for sparse graphs, and thus the detection performance of any detection method will be worse than that of the dense graph case.

2) Eigenvectors of Shift Matrices: Next, we discuss detection methods in terms of the eigenvectors of the shift matrix. Since Sybil detection is essentially a problem of identifying Sybil and benign regions of a graph, the more informative the low frequency eigenvectors of each shift matrix are about the community structure of the graph, the better the detection performance is likely to be. Therefore, to measure the informativeness of low frequency eigenvectors, we evaluate the community detectability of each shift matrix. Specifically, we estimate communities by spectral clustering algorithm (Algorithm 1) with each shift matrix of graphs with $k = 2$ communities generated by SBM and DCSBM and compare the Normalized Mutual Information (NMI) score of true and estimated communities.

Figure 4 shows the community detectability of low frequency eigenvectors of each shift matrix for sparse modular graphs generated by SBM and DCSBM. The vertical dashed lines represent the detectability threshold on the right-hand side of equations (3) and (5), respectively. First, for SBM graphs, all shift matrices can detect communities in the detectable region, and in particular, the Bethe-Hessian shows the highest detection performance. For DCSBM graphs, the

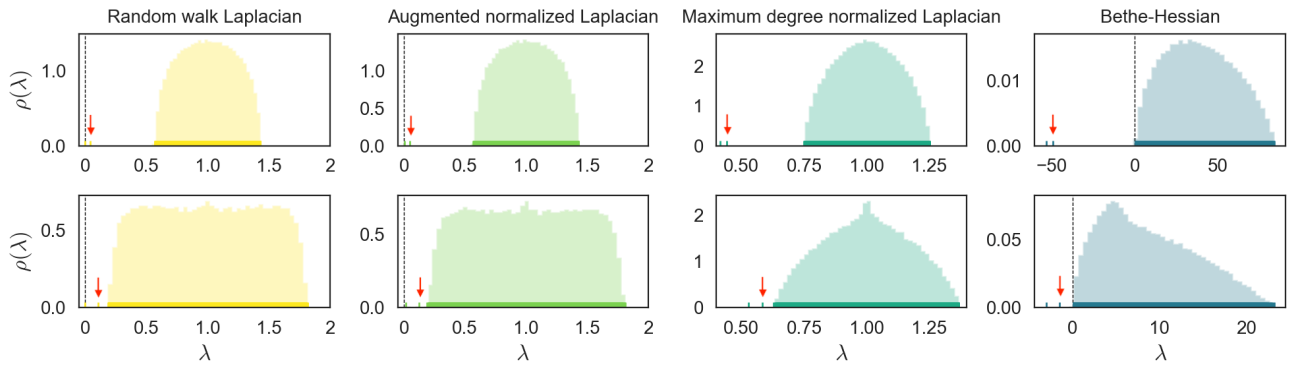

Fig. 3. Spectral distributions of shift matrices for a dense modular graph generated by SBM with $N = 3000$, $k = 2$, $d_{\text{ave}} = 20$, $c_{\text{out}} = 1$ (upper panels) and a sparse modular graph generated by SBM with $N = 3000$, $k = 2$, $d_{\text{ave}} = 5$, $c_{\text{out}} = 1$ (lower panels). In each panel, the mark ("|") stemming from the baseline is the position of eigenvalues, and the red arrow points to the second smallest eigenvalue λ_2.

Algorithm 1 Spectral clustering algorithm

Input: Shift matrix $\boldsymbol{S}$, the number k of communities

1: Compute k smallest eigenvectors of $\boldsymbol{S}$ and construct the $N \times k$ eigenvector matrix $\boldsymbol{V}_k = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_k)$
2: Normalize the rows of $\boldsymbol{V}_k$
3: For $i = 1, 2, \ldots, N$, let $\boldsymbol{y}_i$ be the vector corresponding to the i-th row of $\boldsymbol{V}_k$
4: Cluster the points $\{\boldsymbol{y}_i\}_{i=1}^N$ with k-means into k communities

Output: Estimated communities $\hat{\mathscr{C}}_1, \hat{\mathscr{C}}_2, \ldots, \hat{\mathscr{C}}_k$

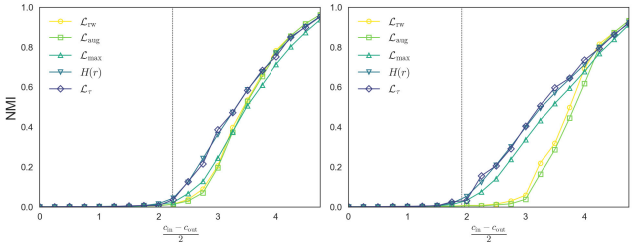

Fig. 4. Community detectability of low frequency eigenvectors of the random walk Laplacian, augmented normalized Laplacian, maximum degree-normalized Laplacian, Bethe-Hessian, and regularized Laplacian for sparse modular graphs generated by SBM (left) and DCSBM (right). The vertical dotted line shows the detectability threshold. Parameters are $N = 1000$, $k = 2$, $d_{\text{ave}} = 5$, and $\theta_i = 1$ for SBM and $\theta_i \sim [\mathcal{U}(3,7)]^3$ for DCSBM. Simulations are averaged over 100 runs.

maximum degree-normalized Laplacian $\boldsymbol{\mathcal{L}}_{\max}$ and the Bethe-Hessian $\boldsymbol{H}(r)$ can detect communities in the detectable region, while the random walk Laplacian $\boldsymbol{\mathcal{L}}_{\text{rw}}$ and the augmented normalized Laplacian $\boldsymbol{\mathcal{L}}_{\text{aug}}$ cannot detect the community when $(c_{\text{in}} - c_{\text{out}})/2$ is small (i.e., weakly modular) even in the detectable region. The same is true for $k > 2$ communities. This suggests that SybilBelief and SybilSCAR-C perform well on sparse, degree heterogeneous, strong modular graphs.

To explain the reason for the high community detectability of $\boldsymbol{\mathcal{L}}_{\max}$ and $\boldsymbol{H}(r)$, we have to review the regularization of a matrix. In the previous studies [40], [41], a method called regularized spectral clustering (RSC) was proposed to improve spectral clustering by using the regularized Laplacian $\boldsymbol{\mathcal{L}}_\tau = \boldsymbol{I} - \boldsymbol{D}_\tau^{-1/2}\boldsymbol{A}\boldsymbol{D}_\tau^{-1/2}$ or $\boldsymbol{\mathcal{L}}_\tau = \boldsymbol{I} - \boldsymbol{D}_\tau^{-1}\boldsymbol{A}$ instead of the random walk Laplacian $\boldsymbol{\mathcal{L}}_{\text{rw}}$. Here, $\boldsymbol{D}_\tau := \boldsymbol{D} + \tau\boldsymbol{I}$ is a regularized degree matrix and $\tau \in \mathbb{R}_{\geq 0}$ is a regularization parameter. For a graph generated by DCSBM, Qin et al. [41] proved that the upper bound of the clustering error of RSC is proportional to $1/(d_{\min} + \tau)$ and $1/\bar{\lambda}_k^2$, where $\bar{\lambda}_k$ is the k smallest eigenvalue of the expectation of $\boldsymbol{\mathcal{L}}_\tau$, $\mathscr{L}_\tau = \mathbb{E}[\boldsymbol{\mathcal{L}}_\tau]$. For small τ (i.e., insufficient regularization), $1/(d_{\min} + \tau)$ becomes large, and conversely, for large τ (i.e., excessive regularization), $1/\bar{\lambda}_k$ becomes large. Therefore, this result suggests that the clustering error of RSC can be minimized by setting appropriate τ. Qin et al. [41] proposed $\tau = d_{\text{ave}}$ as a suitable choice. Indeed, as shown in Fig. 4, the community detectability of $\boldsymbol{\mathcal{L}}_\tau$ with $\tau = d_{\text{ave}}$ is comparable to that of the Bethe-Hessian, which has the best performance.

Given the above, the results in Fig. 4 can be explained as follows. Defining the diagonal matrix $\widehat{\boldsymbol{I}}$ as $[\widehat{I}]_{ii} = 1$ if $i \in V_b \cup V_s$ and $[\widehat{I}]_{ii} = 0$ otherwise, the augmented normalized Laplacian is represented as $\boldsymbol{\mathcal{L}}_{\text{aug}} = \boldsymbol{I} - (\boldsymbol{D} + \widehat{\boldsymbol{I}})^{-1}\boldsymbol{A}$ and can be regarded as being insufficiently and unevenly regularized. The maximum degree-normalized Laplacian can be regarded as being excessively and unevenly regularized since it can be rewritten as $\boldsymbol{\mathcal{L}}_{\max} = \boldsymbol{I} - (\boldsymbol{D} + \boldsymbol{D}_{\text{diff}})^{-1}\boldsymbol{A}$ with $[D_{\text{diff}}]_{ii} := d_{\max} - d_i$. From (19), Bethe-Hessian has the following relationship with the regularized Laplacian:

$$\begin{aligned} \boldsymbol{H}(r)\boldsymbol{v} &= \lambda\boldsymbol{v} \\ &\Leftrightarrow (\boldsymbol{I} - r(\boldsymbol{D} + (r^2 - \lambda - 1)\boldsymbol{I})^{-1}\boldsymbol{A})\boldsymbol{v} = 0 \\ &\Leftrightarrow \boldsymbol{\mathcal{L}}_{r^2-\lambda-1}\boldsymbol{v} = \frac{r-1}{r}\boldsymbol{v}. \end{aligned}$$

Hence, the spectrum of the Bethe-Hessian and the regularized Laplacian are closely related.

C. SybilHeat

The above analysis reveals that the detection performance of each method depends on the effectiveness of the low-pass filtering. More specifically, for a Sybil detection method to perform well, 1) the low-pass filter kernel $h(\lambda)$ must properly emphasize (remove) low (high) frequency components, and 2) low frequency eigenvectors of the shift matrix $\boldsymbol{S}$ must have high community detectability. On the basis of this result, we propose a novel Sybil detection method (SybilHeat) with the filter kernel $h(\lambda) = e^{-s\lambda}$ $(s \geq 0)$ and the shift matrix $\boldsymbol{S} = \boldsymbol{\mathcal{L}}_\tau = \boldsymbol{I} - \boldsymbol{D}_\tau^{-1/2}\boldsymbol{A}\boldsymbol{D}_\tau^{-1/2}$ $(\tau = d_{\text{ave}})$ that satisfy the

above two requirements. The filter kernel $h(\lambda) = e^{-s\lambda}$ is called the heat kernel, and the larger the scaling parameter $s \geq 0$ is, the more strongly high frequency components are reduced. The eigenvectors of $\mathcal{L}_\tau$ have high community detectability comparable to the Bethe-Hessian, as described above. For given prior reputation score $\boldsymbol{q}$, SybilHeat calculates the posterior reputation score $\boldsymbol{p}$ as

$$\boldsymbol{p} = \boldsymbol{V}_\tau e^{-s\boldsymbol{\Lambda}_\tau} \boldsymbol{V}_\tau^{-1} = e^{-s\mathcal{L}_\tau}\boldsymbol{q} = h(\mathcal{L}_\tau)\,\boldsymbol{q}, \tag{24}$$

where $\boldsymbol{\Lambda}_\tau$ and $\boldsymbol{V}_\tau$ are matrices consisting of eigenvalues and eigenvectors of $\mathcal{L}_\tau$, respectively.

Although all eigenvalues and eigenvectors are needed to naively compute $h(\mathcal{L}_\tau)$, the time complexity of the eigenvalue decomposition is $O(N^3)$, which is not scalable for large N. However, fortunately, the Chebyshev polynomial approximation [42] enables us to avoid computing the eigenvalue decomposition and approximately compute (24). Specifically, by using the (shifted) Chebyshev polynomials $\{\tilde{T}_k(\lambda)\}_{k=0}^{\infty}$ defined on the range $\lambda \in [0, 2]$, we approximate $h(\mathcal{L}_\tau)\boldsymbol{q}$ as

$$h(\mathcal{L}_\tau)\boldsymbol{q} \approx \frac{\tilde{c}_0}{2}\boldsymbol{q} + \sum_{k=1}^{K} \tilde{c}_k \tilde{T}_k(\mathcal{L}_\tau)\boldsymbol{q}, \tag{25}$$

where K is the approximation order. The Chebyshev coefficient $\tilde{c}_k$ $(k = 0, 1, \dots)$ is calculated by

$$\tilde{c}_k = \frac{2}{\pi}\int_0^\pi h(\cos\theta + 1)\cos(k\theta)d\theta. \tag{26}$$

By definition of the Chebyshev polynomials, $\tilde{T}_0(\mathcal{L}_\tau) = \boldsymbol{I}$, $\tilde{T}_1(\mathcal{L}_\tau) = \mathcal{L}_\tau - \boldsymbol{I}$, and, for $k \geq 2$, $\tilde{T}_k(\mathcal{L}_\tau)$ satisfies

$$\tilde{T}_k(\mathcal{L}_\tau) = 2(\mathcal{L}_\tau - \boldsymbol{I})\tilde{T}_{k-1}(\mathcal{L}_\tau) - \tilde{T}_{k-2}(\mathcal{L}_\tau).$$

This indicates that a vector $\tilde{T}_k(\mathcal{L}_\tau)\boldsymbol{q}$ in (25) can be recursively computed from $\tilde{T}_{k-1}(\mathcal{L}_\tau)\boldsymbol{q}$ and $\tilde{T}_{k-2}(\mathcal{L}_\tau)\boldsymbol{q}$. Dominant in this computational cost is the matrix-vector multiplication of $\mathcal{L}_\tau$, and its time complexity is $O(|E|)$ [42]. Thus, the overall time complexity of (25) is $O(K|E|)$. Since social networks are often sparse (i.e., $|E| \propto N$), SybilHeat is applicable to large social networks. For comparison, we present the time complexity of each method in Table II, based on the complexity analysis of prior works [8], [10], [11], [15]. Note that t is the number of iterations. Considering that $\log N$ is very small and that $|\widehat{E}|$ is almost equal to $|E|$, we see that all methods have almost the same scalability (linear time complexity with respect to $|E|$) theoretically. However, as shown in Section VI-D, the empirical runtimes of each method are different.

VI. Empirical Evaluations

In the previous section, on the basis of the low-pass filtering interpretation of existing Sybil detection methods, we explained the reasons for the superiority or inferiority of the performance and the requirements for high performance of each method. In addition, we proposed SybilHeat on the basis of our analysis. In this section, we validate our analysis and evaluate the detection performance and noise robustness of SybilHeat through experiments on synthetic graphs and real-world social networks. Furthermore, we evaluate scalability in terms of the empirical runtime on synthetic graphs.

TABLE II
Comparison of the Time Complexity of Each Method

Method	Complexity	Source
CIA	$O(t\|E\|)$	[10]
SybilRank	$O(N \log N)$	[8]
SybilWalk	$O(t\|\widehat{E}\|)$	[10]
SybilSCAR-C	$O(t\|E\|)$	[15]
SybilBelief	$O(t\|E\|)$	[11]
SybilHeat	$O(K\|E\|)$	–

A. Experimental Setup

We use synthetic graphs and some real-world social networks for the experiments. Synthetic graphs are generated by the SBM and DCSBM with average degree $d_{\text{ave}} = 5$ and $k = 2$ even-sized communities. We assume that one community is the benign region and the other is the Sybil region. We also assume that 10% of nodes randomly selected from the benign region are labeled benign nodes and 10% of nodes randomly selected from the Sybil region are labeled Sybil nodes.

We also use real social networks with community structure that are benchmark datasets for community detection tasks for evaluating the detection performance and noise robustness: Zachary karate club [43] (34 nodes and 78 edges), dolphin social network [44] (62 nodes and 159 edges), American colledge football [45] (115 nodes and 613 edges), and political blogs [46] (1224 nodes and 33430 edges). For evaluation, we used the largest connected component of each graph, with half of the communities as benign regions and the rest as Sybil regions. The $\max(3, \lfloor 0.1N \rfloor)$ nodes randomly selected from the benign region were labeled benign nodes, and the $\max(3, \lfloor 0.1N \rfloor)$ nodes randomly selected from the Sybil region were labeled Sybil nodes.

We compare SybilHeat with CIA, SybilRank, SybilWalk, SybilSCAR-C, and SybilBelief. SybilSCAR-D is excluded because its convergence is not stable. The experimental parameters for each method were set as: the restart parameter $\alpha = 0.85$ for CIA, the number of iteration $\Gamma = \lfloor \log N \rfloor$ for SybilRank, the residual prior probability $\theta = 0.5$ for SybilSCAR, and the scaling parameter $s = 8$ for SybilHeat.

B. Detection Performance

Since the Sybil detection method provides a ranking for each node such that Sybil nodes are ranked higher than benign nodes [47], we adopt the Area Under the Receiver Operating Characteristic Curve (AUC) to evaluate detection performance. The AUC of a method is the probability that a (randomly selected) Sybil node ranks higher than a benign node, and the AUC is 1.0 if all Sybil nodes rank higher than benign nodes. If all nodes are ranked uniformly at random, the AUC is 0.5.

Figure 5 shows the detection performance of each method with respect to the community strength $|c_{\text{in}} - c_{\text{out}}|/2$ on synthetic graphs generated by the SBM (left) and DCSBM (right). We observe the following results from this figure. First, the detection performance of all methods increases as

TABLE III
DETECTION PERFORMANCE ON REAL SOCIAL NETWORKS. NUMBERS IN BOLD INDICATE THE BEST PERFORMANCE AND UNDERLINED ONES ARE THE SECOND BEST

	karate			dolphins			football			polblogs		
Method	$\varepsilon = 0.0$	0.1	0.2	$\varepsilon = 0.0$	0.1	0.2	$\varepsilon = 0.0$	0.1	0.2	$\varepsilon = 0.0$	0.1	0.2
CIA	0.8	0.58	0.54	0.97	0.78	0.71	0.78	0.69	0.55	0.75	0.7	0.63
SybilRank	0.95	0.66	0.56	0.96	0.76	0.61	0.82	0.73	0.59	0.97	0.96	0.92
SybilWalk	0.99	0.78	**0.61**	**1.0**	**0.97**	0.77	0.88	0.77	0.59	0.75	0.77	0.75
SybilSCAR-C	0.97	0.72	0.55	0.99	0.82	0.7	0.89	0.79	0.6	0.97	0.93	0.84
SybilBelief	**1.0**	**0.93**	0.53	**1.0**	**0.97**	**0.78**	**0.91**	**0.82**	**0.61**	0.97	0.96	0.9
Ours	0.99	0.79	0.52	**1.0**	0.93	0.74	0.89	0.8	0.6	**0.98**	**0.98**	**0.96**

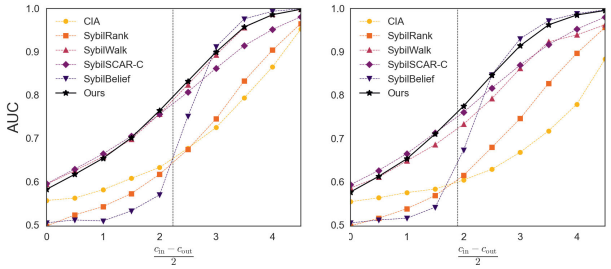

Fig. 5. Detection performance of CIA, SybilRank, SybilWalk, SybilSCAR, SybilBelief, and SybilHeat for sparse modular graphs generated by SBM (left) and DCSBM (right). The vertical dotted line shows the detectability threshold. Parameters are $N = 1000$, $k = 2$, $d_{\text{ave}} = 5$, and $\theta_i = 1$ for SBM and $\theta_i \sim [\mathcal{U}(3,7)]^3$ for DCSBM. Simulations are averaged over 100 runs.

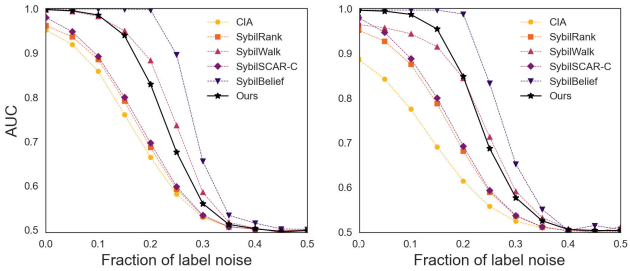

Fig. 6. Noise robustness of CIA, SybilRank, SybilWalk, SybilSCAR, SybilBelief, and SybilHeat for sparse modular graphs generated by SBM (left) and DCSBM (right). Parameters are $N = 1000$, $k = 2$, $d_{\text{ave}} = 5$, $c_{\text{out}} = 0.5$, and $\theta_i = 1$ for SBM and $\theta_i \sim [\mathcal{U}(3,7)]^3$ for DCSBM. Simulations are averaged over 100 runs.

the modularity (i.e., the difference between intra- and inter-community connectivity) increases. Second, random walk-based methods (CIA, SybilRank, and SybilWalk) perform worse for DCSBM graphs than for SBM graphs. This is due to the low community detection performance of the shift matrices corresponding to these methods for DCSBM graphs, as shown in Fig. 4. Moreover, SybilBelief shows the best performance in the detectable region, while in the undetectable region, its detection performance drops sharply. In other words, SybilBelief performs well only on strongly modular graphs. This is due to the fact that SybilBelief relies on only the k small eigenvectors of the Bethe-Hessian (which have no information about the community structure in the undetectable region) to perform detection. On the other hand, SybilHeat performs consistently better over the two regions than the other methods. That is, SybilHeat performs better next to SybilBelief in the detectable region and performs best in the undetectable region, comparable to SybilWalk and SybilSCAR-C.

C. Robustness to Label Noise

In practice, training datasets may contain noise due to human error [48]. That is, some labeled benign (Sybil) nodes are actually Sybil (benign). To evaluate the noise robustness of each method, we compare their detection performance when the node labels of training data with a fraction ε (≤ 0.5) are flipped.

Figure 6 shows the detection performance of each method with respect to the fraction of label noise on synthetic graphs. First, as discussed in Section V-A, SybilBelief is quite robust against label noise because the corresponding filter can completely remove high frequency components, while CIA and SybilRank are not robust because of insufficient low-pass filtering. SybilHeat has higher noise robustness than other methods except SybilBelief. This is because the filter kernel $h(\lambda) = e^{-s\lambda}$ corresponding to SybilHeat greatly removes high frequency components. In particular, in the low noise range ($\varepsilon \leq 0.1$), SybilHeat performs almost no worse than in the noiseless case ($\varepsilon = 0.0$). Although SybilWalk and SybilSCAR-C have the same filter kernel $h(\lambda) = 1/\lambda$, SybilSCAR-C has lower noise robustness than SybilWalk. This is because the contribution of high frequency components cannot be neglected since eigenvalues of the shift matrix corresponding to SybilSCAR-C on sparse graphs are aggregated around $\lambda = 1$ (i.e., low and high frequency eigenvalues of $\mathcal{L}_{\max}$ are close to each other), as shown in Figure 3.

Table III shows the detection performance of each method for $\varepsilon = 0.0$, 0.1, and 0.2 on real-world social networks. As in the results on synthetic graphs, SybilBelief is robust to label noise on all datasets. SybilWalk and SybilHeat are next to SybilBelief in robustness. However, SybilHeat performs more consistently than SybilWalk because the performance of SybilWalk varies with the data, as shown in the results for polblogs.

D. Scalability

We evaluate scalability in terms of the runtime of each method for varying the number of edges in synthetic graphs. Note that all methods are iterative algorithms, so their runtimes are highly dependent on the number of iterations t. To avoid

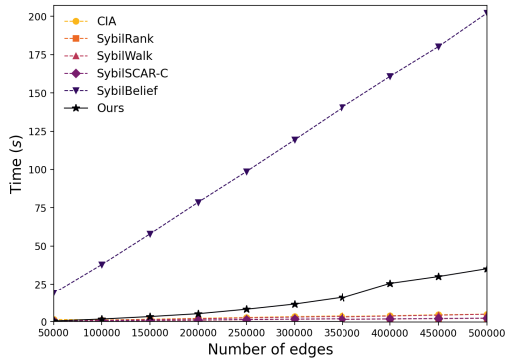

Fig. 7. Runtimes of CIA, SybilRank, SybilWalk, SybilSCAR, SybilBelief, and SybilHeat with respect to the number of edges.

bias due to the number of iterations, we run these methods with the same number of iterations ($t = 20$). For SybilHeat, we set the approximate order $K = 20$.

Figure 7 shows runtimes of SybilHeat and the compared methods with respect to the number of edges. All methods have linear time complexity with respect to the number of edges, which is consistent with the analysis in Section V-C. Furthermore, CIA, SybilRank, SybilWalk, and SybilSCAR are almost equally scalable, whereas SybilBelief requires more runtime than the other methods. Similar results are reported in the previous work [10], [15]. SybilHeat requires slightly more runtime than CIA, SybilRank, SybilWalk, and SybilSCAR, but is more scalable than SybilBelief.

VII. Conclusion

We have shown that existing graph-based Sybil detection methods can be interpreted in a unified framework of low-pass filtering. According to this interpretation, the performance of a Sybil detection method depends on how good the low-pass filtering is. In other words, for a Sybil detection method to perform well, 1) the filter kernel $h(\lambda)$ must properly emphasize (remove) low (high) frequency components, and 2) the low frequency eigenvectors of the shift matrix $\boldsymbol{S}$ must have high community detectability. Therefore, we have compared and analyzed existing detection methods from two perspectives (filter kernel properties and spectrum of the shift matrices) and have provided theoretical explanations of the superiority or inferiority of the performance and the conditions for high performance of each method. Furthermore, we proposed the Sybil detection method (called SybilHeat) with the heat kernel as the filter kernel and the regularized Laplacian as the shift matrix, which satisfies the above two requirements. SybilHeat is applicable to large social networks because it can be approximated by the Chebyshev polynomial approximation in the linear order with respect to the number of edges. Numerical experiments show that SybilHeat performs consistently better than other methods on graphs with various structural properties.

Although we proposed a novel Sybil detection method using heat kernel and regularized Laplacian as the filter kernel and shift matrix, respectively, the performance might be improved by using other better filter kernels or shift matrices. Also, as stated in existing studies [9], [12], [13], [14], learning node features and edge weights will improve detection performance. We hope that this study leads to a deeper theoretical understanding and further improvement of graph-based Sybil detection methods.

Appendix
Linearization and Filtering Interpretation of Loopy Belief Propagation

We first explain the linearization of loopy belief propagation by an approach presented in [49]. We respectively define the node potential function and edge potential function as $\phi_i(s_i) = \exp(\beta\theta_i s_i)$ and $\psi_{ij}(s_i, s_j) = \exp(\beta J_{ij} s_i s_j)$ where β is the inverse temperature, θ_i is the local magnetic field on node i, and J_{ij} is the interaction strength between node i and node j. Note that, the definitions of potential functions in (15) and (16) can be recovered by normalizing the above definitions such that $\sum_{s_i} \phi_i(s_i) = 1$ and $\sum_{s_i, s_j} \psi_{ij}(s_i, s_j) = 1$.

Let us introduce the one-parametrized message $\nu_{ij} := \tanh^{-1}(\mu_{ij}(+1) - \mu_{ij}(-1))$ instead of the message $\mu_{ij}(s_j)$. For simplicity, denoting $\mu_{ij}^{+} := \mu_{ij}(+1)$ and $\mu_{ij}^{-} := \mu_{ij}(-1)$, we obtain

$$\begin{aligned}
&\tanh(\nu_{ij}) \\
&= \frac{1}{Z_{ij}} \left(e^{\beta J_{ij}} - e^{-\beta J_{ij}}\right) \left(e^{\beta\theta_i} \prod_{k\in\partial_{i\setminus j}} \mu_{ki}^{+} - e^{-\beta\theta_i} \prod_{k\in\partial_{i\setminus j}} \mu_{ki}^{-}\right) \\
&= \tanh\left(\beta J_{ij}\right) \frac{e^{\beta\theta_i} \prod_{k\in\partial_{i\setminus j}} \mu_{ki}^{+} - e^{-\beta\theta_i} \prod_{k\in\partial_{i\setminus j}} \mu_{ki}^{-}}{e^{\beta\theta_i} \prod_{k\in\partial_{i\setminus j}} \mu_{ki}^{+} + e^{-\beta\theta_i} \prod_{k\in\partial_{i\setminus j}} \mu_{ki}^{-}}.
\end{aligned}$$

Since

$$\begin{aligned}
\log \prod_{k\in\partial_{i\setminus j}} \left(\frac{\mu_{ki}^{+}}{\mu_{ki}^{-}}\right)^{\frac{1}{2}} &= \sum_{k\in\partial_{i\setminus j}} \frac{1}{2} \log \frac{\mu_{ki}^{+} + \mu_{ki}^{-} + \mu_{ki}^{+} - \mu_{ki}^{-}}{\mu_{ki}^{+} + \mu_{ki}^{-} - \mu_{ki}^{+} + \mu_{ki}^{-}} \\
&= \sum_{k\in\partial_{i\setminus j}} \frac{1}{2} \log \frac{1 + (\mu_{ki}^{+} - \mu_{ki}^{-})}{1 - (\mu_{ki}^{+} + \mu_{ki}^{-})} \\
&= \sum_{k\in\partial_{i\setminus j}} \tanh^{-1}(\mu_{ki}^{+} - \mu_{ki}^{-}) = \sum_{k\in\partial_{i\setminus j}} \nu_{ki},
\end{aligned}$$

by using ν_{ij}, we can rewrite (17) as

$$\tanh\left(\nu_{ij}^{\text{new}}\right) = \tanh\left(\beta J_{ij}\right) \tanh\left(\beta\theta_i + \sum_{k\in\partial_{i\setminus j}} \nu_{ki}\right). \quad (27)$$

The approximate marginal distribution $b_i(s_i)$ can also be one-parametrized by its expectation (called magnetization) $m_i = \langle s_i \rangle = b_i(1) - b_i(-1)$ as follows:

$$\begin{aligned}
m_i &= \frac{e^{\beta\theta_i} \prod_{k\in\partial i} \mu_{ki}^{\infty}(1) - e^{-\beta\theta_i} \prod_{k\in\partial i} \mu_{ki}^{\infty}(-1)}{e^{\beta\theta_i} \prod_{k\in\partial i} \mu_{ki}^{\infty}(1) + e^{-\beta\theta_i} \prod_{k\in\partial i} \mu_{ki}^{\infty}(-1)} \\
&= \tanh\left(\beta\theta_i + \sum_{k\in\partial i} \nu_{ki}^{\infty}\right). \quad (28)
\end{aligned}$$

Denoting $\boldsymbol{\nu} = (\nu_{ij}) \in \mathbb{R}^{|\vec{E}|}$, let $\mathcal{BP} : \boldsymbol{\nu} \mapsto \boldsymbol{\nu}^{\text{new}}$ be the nonlinear operator that maps $\boldsymbol{\nu}$ to $\boldsymbol{\nu}^{\text{new}}$ by following (27). The element of the Jacobian matrix $\boldsymbol{B} = \mathcal{BP}'(\boldsymbol{\nu}) \in \mathbb{R}^{|\vec{E}|\times|\vec{E}|}$ of $\mathcal{BP}$ is given by

$$B_{ij,kl} = \frac{\partial \nu_{ij}^{\text{new}}}{\partial \nu_{kl}} = \frac{\tanh(\beta J_{ij})\left(1 - \tanh^2(h_{i\setminus j})\right)}{1 - \tanh^2(\beta J_{ij})\tanh^2(h_{i\setminus j})}\delta_{il}(1-\delta_{jk}), \tag{29}$$

where $h_{i\setminus j} := \beta\theta_i + \sum_{k\in\partial_{i\setminus j}} \nu_{ki}$. The matrix $\boldsymbol{B}$ is called a non-backtracking matrix and its (ij, kl)-element takes a non-zero value if two directed edges are consecutive (i.e., $i = l$) but do not back track (i.e., $j \neq k$). To simplify the analysis, we assume the vanishing local field condition (i.e., $\theta_i = 0$ for all $i \in V$). This means that there is no prior information for each node. In this case, we have $B_{ij,kl} = \tanh(\beta J_{ij})\delta_{il}(1-\delta_{jk})$ and thus (27) can be written as $\boldsymbol{\nu}^{\text{new}} \approx \boldsymbol{B}\boldsymbol{\nu}$ by the linearization around the trivial fixed point $\boldsymbol{\nu}^* = 0$. Hence, when the spectral radius $\rho(\boldsymbol{B}) < 1$, the message $\boldsymbol{\nu}$ converges to the trivial fixed point $\boldsymbol{\nu}^*$. On the other hand, when $\rho(\boldsymbol{B}) > 1$, $\boldsymbol{\nu}$ leaves from $\boldsymbol{\nu}^*$. The eigenvector associated with an eigenvalue larger than 1 is expected to correspond approximately to non-trivial (hopefully informative) fixed points of the loopy belief propagation [50].

We consider unweighted non-backtracking matrix below (i.e., $B_{ij,kl} = \delta_{il}(1-\delta_{jk})$). For small $|\nu_{ki}|$, the magnetization of each node is given by

$$m_i = \tanh\left(\sum_{k\in\partial i} \nu_{ki}^{\infty}\right) \approx \sum_{k\in\partial i} \nu_{ki}^{\infty}. \tag{30}$$

The eigenvector $\boldsymbol{\nu}$ associated with an eigenvalue $\eta > 1$ satisfies

$$\eta\nu_{ij} = \sum_{(k,l)\in\vec{E}} B_{ij,kl}\nu_{kl} = \sum_{k\in\partial_{i\setminus j}} \nu_{ki} = m_i - \nu_{ji}. \tag{31}$$

Similarly, $\eta\nu_{ji} = m_j - \nu_{ij}$ holds. Thus, we have $\nu_{ij} = (\eta m_i - m_j)/(\eta^2 - 1)$. By substituting this into (30), we obtain

$$(\eta^2 - 1)m_i + |\partial i|m_i - \eta\sum_{k\in\partial i} m_k = 0. \tag{32}$$

Alternatively, we rewrite the above equation as $\boldsymbol{H}(\eta)\boldsymbol{m} = 0$ by using the Bethe-Hessian matrix.

The small eigenvalues of $\boldsymbol{H}(r)$ are closely related to the informative eigenvalues of $\boldsymbol{B}$, and the corresponding eigenvectors approximately give the magnetization $\boldsymbol{m}$ calculated by the loopy belief propagation [38], [50]. On the basis of this observation, a community detection algorithm using eigenvectors corresponding to small (low frequency) eigenvalues of the Bethe-Hessian matrix has been proposed and its performance has been demonstrated to be comparable to loopy belief propagation [38], [51], [52]. In the same spirit, we can interpret SybilBelief as low-pass filtering as in (20) using the ideal low-pass filter kernel $g(\omega)$ that extracts only the low frequency spectrum of $\boldsymbol{H}(r)$.

Note that we have assumed the vanishing local field condition through our analysis. This assumption is quite strong since it implies ignoring all known node labels. However, since the local magnetic field helps accelerate the convergence of the message and biases it toward the desired local minimum [53], a similar effect is expected by low-pass filtering of the known label vector $\boldsymbol{q}$.

REFERENCES

[1] B. Auxier and M. Anderson, "Social media use in 2021," *Pew Res. Center*, vol. 1, pp. 1–4, Apr. 2021.

[2] D. A. Broniatowski et al., "Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate," *Amer. J. Public Health*, vol. 108, no. 10, pp. 1378–1384, 2018.

[3] J.-P. Allem and E. Ferrara, "Could social bots pose a threat to public health?" *Amer. J. Public Health*, vol. 108, no. 8, p. 1005, 2018.

[4] A. Bessi and E. Ferrara, "Social bots distort the 2016 U.S. Presidential election online discussion," *1st Monday*, vol. 21, nos. 7–11, Nov. 2016.

[5] M. T. Bastos and D. Mercea, "The Brexit botnet and user-generated hyperpartisan news," *Social Sci. Comput. Rev.*, vol. 37, no. 1, pp. 38–54, Feb. 2019.

[6] L. Alvisi, A. Clement, A. Epasto, S. Lattanzi, and A. Panconesi, "SoK: The evolution of Sybil defense via social networks," in *Proc. IEEE Symp. Secur. Privacy*, May 2013, pp. 382–396.

[7] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu, "Analyzing spammers' social networks for fun and profit: A case study of cyber criminal ecosystem on Twitter," in *Proc. 21st Int. Conf. WWW*, 2012, pp. 71–80.

[8] Q. Cao, M. Sirivianos, X. Yang, and T. Pregueiro, "Aiding the detection of fake accounts in large scale social online services," in *Proc. 9th USENIX Symp. Networked Syst. Design Implement.*, 2012, pp. 197–210.

[9] Y. Boshmaf et al., "Íntegro: Leveraging victim prediction for robust fake account detection in large scale OSNs," *Comput. Secur.*, vol. 61, pp. 142–168, Jan. 2016.

[10] J. Jia, B. Wang, and N. Z. Gong, "Random walk based fake account detection in online social networks," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2017, pp. 273–284.

[11] N. Z. Gong, M. Frank, and P. Mittal, "Sybilbelief: A semi-supervised learning approach for structure-based Sybil detection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 6, pp. 976–987, Jun. 2014.

[12] H. Fu, X. Xie, Y. Rui, N. Z. Gong, G. Sun, and E. Chen, "Robust spammer detection in microblogs: Leveraging user carefulness," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 6, pp. 1–31, 2017.

[13] P. Gao, B. Wang, N. Z. Gong, S. R. Kulkarni, K. Thomas, and P. Mittal, "SYBILFUSE: Combining local attributes with global structure to perform robust Sybil detection," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, May 2018, pp. 1–9.

[14] A. Dorri, M. Abadi, and M. Dadfarnia, "SocialBotHunter: Botnet detection in Twitter-like social networking services using semi-supervised collective classification," in *Proc. IEEE 16th Int. Conf. Dependable, Autonomic Secure Comput., 16th Int. Conf. Pervasive Intell. Comput., 4th Int. Conf. Big Data Intell. Comput. Cyber Sci. Technol. Congr. (DASC/PiCom/DataCom/CyberSciTech)*, Aug. 2018, pp. 496–503.

[15] B. Wang, J. Jia, L. Zhang, and N. Z. Gong, "Structure-based Sybil detection in social networks via local rule-based propagation," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 3, pp. 523–537, Jul. 2019.

[16] S. Furutani, T. Shibahara, K. Hato, M. Akiyama, and M. Aida, "Sybil detection as graph filtering," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[17] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "SybilGuard: Defending against Sybil attacks via social networks," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, Aug. 2006, pp. 267–278.

[18] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao, "SybilLimit: A near-optimal social network defense against Sybil attacks," in *Proc. IEEE Symp. Secur. Privacy*, May 2008, pp. 3–17.

[19] G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil nodes using social networks," in *Proc. 16th Annu. Netw. Dist. Syst. Secur. Symp.* San Diego, CA, USA, 2009, pp. 1–15.

[20] A. H. Wang, "Don't follow me: Spam detection in Twitter," in *Proc. Int. Conf. Secur. Cryptogr.*, Jul. 2010, pp. 1–10.

[21] K. Lee, J. Caverlee, and S. Webb, "Uncovering social spammers: Social honeypots + machine learning," in *Proc. 33rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2010, pp. 435–442.

[22] C. Yang, R. C. Harkreader, and G. Gu, "Die free or live hard? Empirical evaluation and new design for fighting evolving Twitter spammers," in *Proc. 14th Int. Workshop Recent Adv. Intrusion Detect.* Cham, Switzerland: Springer, 2011, pp. 318–337.

[23] A. A. Amleshwaram, N. Reddy, S. Yadav, G. Gu, and C. Yang, "CATS: Characterizing automation of Twitter spammers," in *Proc. 5th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2013, pp. 1–10.

[24] B. Wang, A. Zubiaga, M. Liakata, and R. Procter, "Making the most of tweet-inherent features for social spam detection on Twitter," 2015, *arXiv:1503.07405*.

[25] X. Zheng, Z. Zeng, Z. Chen, Y. Yu, and C. Rong, "Detecting spammers on social networks," *Neurocomputing*, vol. 159, pp. 27–34, Jul. 2015.

[26] G. Jethava and U. P. Rao, "User behavior-based and graph-based hybrid approach for detection of Sybil attack in online social networks," *Comput. Electr. Eng.*, vol. 99, Apr. 2022, Art. no. 107753.

[27] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2012.

[28] A. Ortega, P. Frossard, J. Kovačevič, J. M. F. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.

[29] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Netw.*, vol. 5, no. 2, pp. 109–137, 1983.

[30] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, "Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, Dec. 2011, Art. no. 066106.

[31] E. Mossel, J. Neeman, and A. Sly, "Stochastic block models and reconstruction," 2012, *arXiv:1202.1499*.

[32] L. Massoulié, "Community detection thresholds and the weak Ramanujan property," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, May 2014, pp. 694–703.

[33] B. Karrer and M. E. J. Newman, "Stochastic blockmodels and community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 83, Jan. 2011, Art. no. 016107.

[34] L. Gulikers, M. Lelarge, and L. Massoulié, "An impossibility result for reconstruction in the degree-corrected stochastic block model," *Ann. Appl. Probab.*, vol. 28, no. 5, pp. 3002–3027, Oct. 2018.

[35] B. Wang, J. Jia, and N. Zhenqiang Gong, "Graph-based security and privacy analytics via collective classification with joint weight learning and propagation," 2018, *arXiv:1812.01661*.

[36] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Burlington, MA, USA: Morgan Kaufmann, 1988.

[37] K. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," 2013, *arXiv:1301.6725*.

[38] A. Saade, F. Krzakala, and L. Zdeborová, "Spectral clustering of graphs with the Bethe Hessian," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–15.

[39] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.

[40] K. Chaudhuri, F. Chung, and A. Tsiatas, "Spectral clustering of graphs with general degrees in the extended planted partition model," in *Proc. 25th Annu. Conf. Learn. Theory*, 2012, pp. 1–35.

[41] T. Qin and K. Rohe, "Regularized spectral clustering under the degree-corrected stochastic blockmodel," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 26, 2013, pp. 1–13.

[42] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.

[43] W. W. Zachary, "An information flow model for conflict and fission in small groups," *J. Anthropol. Res.*, vol. 33, no. 4, pp. 452–473, Dec. 1977.

[44] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behav. Ecol. Sociobiol.*, vol. 54, no. 4, pp. 396–405 2003.

[45] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Apr. 2002.

[46] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. 3rd Int. Workshop Link Discovery*, 2005, pp. 36–43.

[47] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based Sybil defenses," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 363–374, 2010.

[48] G. A. Wang et al., "Social turing tests: Crowdsourcing Sybil detection," in *Proc. 20th Annu. Netw. Dist. Syst. Secur. Symp.*, 2013, pp. 1–16.

[49] J. M. Mooij and H. J. Kappen, "On the properties of the Bethe approximation and loopy belief propagation on binary networks," *J. Stat. Mech., Theory Exp.*, vol. 2005, no. 11, Nov. 2005, Art. no. P11012.

[50] A. Saade, "Spectral inference methods on sparse graphs: Theory and applications," 2016, *arXiv:1610.04337*.

[51] L. Dall'Amico, R. Couillet, and N. Tremblay, "Revisiting the Bethe-Hessian: Improved community detection in sparse heterogeneous graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[52] L. Dall'Amico, R. Couillet, and N. Tremblay, "Optimal Laplacian regularization for sparse spectral community detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3237–3241.

[53] C. Knoll and F. Pernkopf, "On loopy belief propagation—Local stability analysis for non-vanishing fields," in *Proc. 33rd Conf. Uncertainty Artif. Intell.*, 2017, pp. 1–10.

Satoshi Furutani received the B.E. and M.E. degrees in system design engineering from Tokyo Metropolitan University, Japan, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 2018, he has been engaged in research and development on network science. He is currently a Researcher with NTT Social Informatics Laboratories. His research interests include analysis of social network dynamics and graph signal processing. He is a member of IPSJ and IEICE.

Toshiki Shibahara received the B.E. degree in engineering and the M.E degree in information science and technology from The University of Tokyo, Tokyo, Japan, in 2012 and 2014, respectively, and the Ph.D. degree in information science and technology from Osaka University, Osaka, Japan, in 2020. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 2014, he has been engaged in research on cyber security and machine learning. He is currently a Researcher with NTT Social Informatics Laboratories, Tokyo.

Mitsuaki Akiyama (Member, IEEE) received the M.E. and Ph.D. degrees in information science from the Nara Institute of Science and Technology, Japan, in 2007 and 2013, respectively. Since joining Nippon Telegraph and Telephone Corporation (NTT) in 2007, he has been engaged in research and development on cybersecurity. He is currently a Senior Distinguished Researcher at NTT Social Informatics Laboratories. His research interests include cybersecurity measurement, offensive security, and usable security and privacy. He is a Senior Member of IPSJ and a member of IEICE. He received the Cybersecurity Encouragement Award of the Minister for Internal Affairs and Communications in 2020, the ISOC NDSS 2020 Distinguished Paper Award in 2020, and the IPSJ/IEEE Computer Society Young Computer Researcher Award in 2022.

Masaki Aida (Senior Member, IEEE) received the B.S. degree in physics and the M.S. degree in atomic physics from Saint Paul's University, Tokyo, Japan, in 1987 and 1989, respectively, and the Ph.D. degree in telecommunications engineering from The University of Tokyo, Japan, in 1999. In April 1989, he joined NTT Laboratories. From April 2005 to March 2007, he was an Associate Professor at the Faculty of Systems Design, Tokyo Metropolitan University. He has been a Professor of the Graduate School of Systems Design, Tokyo Metropolitan University, since April 2007. His current research interests include analysis of social network dynamics and distributed control of computer communication networks. He is a fellow of IEICE and a member of ACM and ORSJ. He received the Best Tutorial Paper Award and the Best Paper Award from the IEICE Communications Society in 2013 and 2016, respectively, and IEICE 100-Year Memorial Paper Award in 2017.