Detection of Infections using Graph Signal Processing in Heterogeneous Networks

Seyyedali Hosseinalipour, Jie Wang, Huaiyu Dai, Wenye Wang
Department of Electrical and Computer Engineering, North Carolina State University
Email: {shossei3,jwang50,hdai,wwang}@ncsu.edu

***Abstract*—Determining the causality of abnormalities in a network is the prerequisite for developing countermeasures. In this paper, we focus on infection detection in heterogeneous networks. Given a snapshot of the network which demonstrates the condition of the nodes, the goal is to distinguish between random failures and epidemic scenarios. We model the network situation as a graph signal based on the nodes' status. Detection metrics motivated by graph signal processing are introduced for the infection detection problem in hand, and an effective algorithm is proposed to solve it. Simulation results indicate a dramatic improvement in terms of detection probability compared to the current state-of-the-art.**

***Index Terms*—Graph signal processing, Infection detection, Heterogeneous networks.**

I. Introduction

A. Infection detection problem

Abnormality detection has been an important and well-studied topic in various systems [1]–[3]. While many existing detection measures focus on examining the existence of a fault-/abnormality in one single entity, detection of abnormality in a networked system is more important. This importance is due to the fact that some faults can propagate through connections in the network. An example of such an abnormality could be a worm that takes advantage of vulnerabilities of computers and disguise itself in emails/messages to infect other victims. Given a network which is infected by an abnormality, it is desirable to quickly determine whether the symptom can spread to other nodes in the network (epidemic), or it is merely due to independent random failures, before any countermeasure shall be deployed. The most straightforward detection measure is to look into the source nodes, which may not always be available. Moreover, the above determination method is dependent on the implementation of the virus. Hence, it might be necessary to redesign the detection methods even for the same network for different viruses. In this sense, a *generic* detection method to distinguish between epidemics and random failures is essential for further treatment of abnormality.

The solution to such problems can also be applied to detection of rumors in online social networks [4], or disease control in a population [5]. In these cases, the state evolution of the network is driven by the spreading process which is usually described by an *epidemic* process. Stemmed from epidemiology, an epidemic process is useful for capturing the spreading behavior when individuals change their states upon having a *contact* with others. However, in addition to rumors or epidemics, individuals in a network can receive ideas or information independently or get sick randomly, which may also lead to irregular behaviors of the nodes. In these cases, the node behavior is independent of others. Clearly, effective countermeasures to one kind of abnormality are not necessarily effective to the other. Therefore, we are motivated to address the following question: *How to quickly detect the root cause of an abnormality based on the observation of the network condition*?

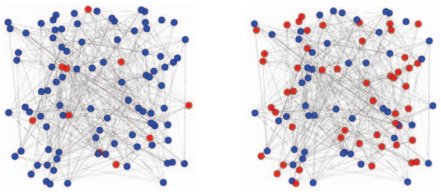
Even if the structure of the network is known *a priori*, it is still challenging to distinguish between random failures and an epidemic by solely observing a snapshot of the network status. The difficulty lies in the incorporation of the network structure into the determination while taking into account the stochastic evolution of the epidemics and random failures. For example, Fig. 1 and Fig. 2 show the snapshots of an Erdős–Rényi (ER) network of 100 nodes subject to an epidemic and random failures, respectively. The similarity of the two indicates the difficulty of detection even in this small network. The detection is more challenging when the epidemic stemmed from diverse seeds since it resembles even more random infections. In addition, noisy observations due to false positives (a healthy node reported as infected) and false negatives (an infected node reported as healthy) add to the challenges of the determination.

B. Related work

Differentiating epidemics from random illness has been studied before through different means. In the most recent work, Milling *et al.* designed a Median Ball algorithm for the same purpose [6]. The rationale of the Median Ball algorithm is that when an epidemic starts from one single source, the infection should spread evenly in all directions, creating a ball consisting of infected nodes. Also, when an epidemic occurs, nodes closer to the seed of the epidemic have a higher probability to be infected, since they are more exposed to infected nodes. This fact leads to more infected nodes in the core area and fewer infected ones in the peripheral areas. The Median Ball algorithm is an algorithm suitable for large homogeneous networks. Nevertheless, it can not adequately address the detection problem when following conditions exist: 1) Existence of multiple (possibly overlapping) balls: When the epidemics stem from various initial infected nodes, especially when the two or more sources are located far apart in the

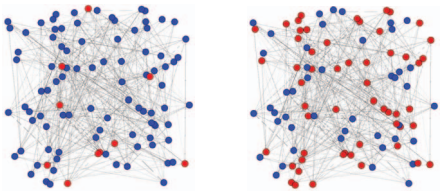
This work was supported in part by the U.S. National Science Foundation under Grants ECCS-1307949 and EARS-1444009, and in part by the U.S. Army Research Office under Grant W911NF-17-1-0087.

978-1-5090-5019-2/17/$31.00 ©2017 IEEE



(a) 10 infected nodes. (b) 50 infected nodes.

Figure 1: Snapshots of epidemics in an ER graph. Red nodes are infected while blue nodes are in healthy state.



(a) 10 infected nodes. (b) 50 infected nodes.

Figure 2: Snapshots of random failures in an ER graph. Red nodes are infected while blue nodes are in healthy state.

graph, the infected nodes will not form a ball-shaped region anymore.

2) Failure of obtaining a unique radius: When the graph is highly irregular, or the connections in the network are heterogeneous, it is hard to obtain a functioning radius to implement the median ball algorithm.

To address the infection detection problem in heterogeneous networks, Milling *et. al.* proposed the Ball Density algorithm and its variation, the Relative Ball Density algorithm [7]. However, the threshold radius parameter in the algorithm should be set empirically and has to be manually obtained for every scenario. In addition, these algorithms are highly complex, since they need the $k-$hop neighbor information[1] for each node to calculate the distance between any two nodes.

In this work, we make a connection between the network epidemiology and graph signal processing (GSP). Introduction of GSP and some analysis can be found in [8], [9], [10]. GSP methods are mostly applied to the problems in the image processing area due to their inherent consistency in describing an image structure.

C. Contributions

In this paper, we provide a new approach for the infection detection problem by modeling the network condition as a graph signal, where the signal is constructed based on the nodes' situation. We adapt metrics in GSP to the problem of infection detection. Moreover, some new metrics are defined based on the signal graphs, including high energy concentration, low energy concentration and generalized smoothness. An algorithm is proposed to detect the infection spreading on a heterogeneous network using the considered model and defined metrics. The proposed algorithm is tested on the scale-free, Erdős–Rényi and grid graphs and its superiority is shown as compared to existing literature. To the best of our knowledge, this is a first work which applies the concepts of GSP to the problem of infection detection in heterogeneous networks and achieves significant performance improvement over the current art.

Structure of paper: Section II is devoted to the infection system model and the infection detection problem. Section III introduces the basic concepts of GSP and the metrics we propose for infection detection. In Section IV, the algorithm for infection detection is presented together with the simulation results. Finally, Section V concludes the paper.

II. System Model

A. Representation of network condition

Consider a slotted time representation of a network. The network is described as an undirected graph $G = (V, E, A)$, where V denotes the set of nodes, E denotes the set of edges and A denotes the weighted adjacency matrix of the graph which contains the distances between all pairs of adjacent nodes. For an arbitrary labeling order of the nodes in the set V, $\{v_1, \cdots, v_N\}$, $s_i(t)$ denotes the state of the node $v_i \in V$ at time t: more specifically it demonstrates the time that has passed since the node has become infected. At any time t the nodes can be classified into two categories: infected (v_i is exhibiting abnormal behavior, $s_i(t) > 0$), or healthy (v_i is behaving normally, $s_i(t) = 0$). Upon occurrence of an epidemic, each node in the set of infected nodes $v_i \in \mathcal{I}_t = \{v_i \in V | s_i(t) > 0\}$ attempts to infect any of its susceptible neighbors $N(v_i) = \{v_j \in V \setminus \mathcal{I}_t | (v_j, v_i) \in E\}$ with probability $0 \leq \frac{\beta}{d(v_j, v_i)} \leq 1$. Here, $d(v_j, v_i)$ is the one-hop distance between v_j and v_i which is assumed to be greater or equal to one, and $0 \leq \beta \leq 1$ is the basic infection rate. The distance between two nodes can be interpreted as the tendency of spreading between the nodes which might be due to the accessibility of the nodes. In the random failure scenario, each node $v \in V$ is assumed to fail with the probability p_{fail} independently at each slot. For a random failed node, $s_v(t)$ will be uniformly distributed in $[1, t]$ irrelevant to the states of other nodes.

Let $r_i(t)$ denote the reported status at node v_i at time t. A *snapshot* of the network G is defined as the collection of the reports from all nodes at time t, that is, $\{r_i(t)\}_{v_i \in V}$[2]. In the presence of noise, false positive ($\{r_i(t) > 0 | s_i(t) = 0\}$) and false negative ($\{r_i(t) = 0 | s_i(t) > 0\}$) events might happen; otherwise, $r_i(t) = s_i(t)$.

Without loss of generality, one can safely assume that the spread of an epidemic begins at $t = 0$. The initial infection set includes the nodes that are infected at time 0, denoted by $\mathcal{I}_0$. In real scenarios, the evolution time of an epidemic is often unknown. Hence, for simplicity we omit the time index t and

[1]Here, k is an arbitrary number smaller than the size of the network.

[2]In some occasions $r_i(t)$ may not be available. We show by simulations that our algorithm also allows for a coarse 'binary' input $\{r_i(t)\}_{v \in V}$, where $r_i(t) = Const. \geq 1$ for all infected nodes $v_i \in \mathcal{I}_t$. In other words, our algorithm can work with both grey-scale and black-and-white snapshots.

replace $r_i(t)$ with r_i. In this case, the *snapshot* of the network can be represented as $R = \{r_i\}_{v_i \in V}$.

B. Infection detection problem

Distinguishing between an epidemic and random failures, given the network snapshot, is the goal of the infection detection problem. In this case, the detection problem consists of two hypotheses H_0 and H_1, which correspond to the occurrence of random failures and an epidemic in the network, respectively.

As in classic detection theory, we use the following probabilities to measure the performance of an infection detection algorithm. 1) Probability of success: the probability that the algorithm makes the right decision based on the network snapshot ($P(H_1|H_1), P(H_0|H_0)$). 2) Probability of failure: the probability of making the wrong decisions ($P(H_0|H_1), P(H_1|H_0)$), where $P(H_0|H_1)$ is the probability of miss and $P(H_1|H_0)$ is the probability of false alarm. A good algorithm should exhibit low probabilities of miss and false alarm simultaneously.

III. Infection Detection using GSP

A. Graph signal

1) Background and definitions: As it is stated before, any arbitrary network structure can be represented by a graph $G = (V, E, A)$. For an arbitrary labeling order of the nodes in the set $V = \{v_1, \cdots, v_N\}$ and an arbitrary set of real numbers $S = \{s_1, \cdots, s_N\}$, where $|S| = N$, a graph signal can be considered as the one to one mapping between the elements of V and S [8]: $v_i \rightarrow s_i,\ 1 \leq i \leq N$.

The Laplacian matrix for a graph is defined as:

$$L := D - A, \tag{1}$$

where the degree matrix D is a diagonal matrix with its i^{th} diagonal element d_i equals to the sum of the weights of all the edges incident to node v_i, and A is the weighted adjacency matrix. Because the graph Laplacian L is a real symmetric matrix, it has a complete set of real and nonnegative eigenvalues which can be ordered as $0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{N-1} = \lambda_{max}$. Moreover, we denote the corresponding set of eigenvectors as $\{u_l\},\ l = 0, 1, \cdots, N-1$.

The graph Fourier Transform $\hat{S}$ of a graph signal $S \in \mathbb{R}^{\mathbb{N}}$ can be considered as the expansion of S onto the space of graph Laplacian eigenvectors. Mathematically,

$$\hat{S}(\lambda_l) =< S, u_l >= \sum_{i=1}^{N} S(i) u_l(i). \tag{2}$$

2) Constructing the graph signal based on the data: In the problem of interest, we are given a snapshot of the network situation (condition), where some nodes are reported to be infected and the rest of them reported to be healthy. We create the graph signal as follows: we obtain the weighted adjacency matrix for the graph, where the weights correspond to distances between nodes. Then, for an arbitrary node i we assign the corresponding signal value to be $S(i) = g(r_i)$. Here, g is an increasing and convex function of r_i, the reported status of node i.

B. Detection metrics

In order to distinguish between an epidemic and random failures, nodes' signal values along with the background graph structure should be jointly considered. As it is stated before, the graph Fourier Transform can be viewed as the projection of the signal onto the space of the graph Laplacian eigenvectors. Considering the fact that projection and correlation have similar meanings in the signal processing area, the following insight is immediate: *each element of the spectrum of a graph Fourier Transform reflects the similarity between the signal and the corresponding Laplacian eigenvector.*

The notion of frequency in graph Fourier Transform is different from the classic discrete time Fourier Transform. For a graph signal, the frequency is related to the variation of signal values both among the neighboring nodes and the whole graph. In this sense, a high frequency signal has rapid signal value variations among close nodes[3]. In a graph Fourier Transform, the eigenvector corresponding to $\lambda_0 = 0$ has the same value for all the nodes. Due to this fact, this eigenvector captures the "DC" value in the signal. Furthermore, as the eigenvalues become larger, their corresponding eigenvector has its components change more rapidly among close nodes in the graph. In conclusion, upon projection of the signal onto the space of the Laplacian matrix eigenvectors, the eigenvectors corresponding to small eigenvalues capture low variations of the signal while the eigenvectors corresponding to large eigenvalues capture high variations of the signal. Using this fact, we can distinguish between graph signals with low and high signal variations among close nodes by observing the spectrum of the graph Fourier Transform. In this regard, the following metrics are introduced to analyze the spectrum of the graph Fourier Transform.

Definition 1. *For any signal $S \in \mathbb{R}^N$ with the graph Fourier Transform $\hat{S}$, we define the energy of the spectrum $Eng(\hat{S})$ as the sum of the magnitude of the frequency components, i.e.*

$$Eng(\hat{S}) = ||\ \hat{S}\ ||_1. \tag{3}$$

Definition 2. *The $\alpha-$ **high energy concentration ratio** $ECRH_\alpha(\hat{S})$ is the ratio between the energy of the highest α portion of the spectrum and the total energy, where $\alpha \in [0, 1]$. Mathematically, this can be written as:*

$$ECRH_\alpha(\hat{S}) = \frac{\sum_{i=\lfloor N(1-\alpha) \rfloor}^{N} \hat{S}(\lambda_i)}{Eng(\hat{S})}. \tag{4}$$

*Similarly, the $\gamma-$ **low energy concentration ratio** $ECRL_\gamma(\hat{S})$ is the ratio between the energy of the lowest γ portion of the spectrum and the total energy, where $\gamma \in [0, 1]$.*

[3]We call nodes which have short distance from each other as close nodes throughout the paper.

Mathematically:

$$ECRL_{\gamma}(\hat{S}) = \frac{\sum_{i=1}^{\lceil \gamma N \rceil} \hat{S}(\lambda_i)}{Eng(\hat{S})}. \quad (5)$$

Intuitively, the $ECRH$ and $ECRL$ reflect the concentration of the spectrum at high and low frequencies, respectively. As mentioned before, eigenvectors corresponding to larger eigenvalues capture high variations of the signal (high frequency). As $ECRH_{\alpha}(\hat{S})$ increases, it indicates a higher similarity of the signal to these eigenvectors, and thus higher variation of the graph signal S among neighboring vertices.

Let us consider creating the graph signal by assigning some non-zero values to the infected nodes and zero values to the healthy nodes. If there is an epidemic in the network, the neighboring nodes around the source will form a group with similar signal values in the graph. This is consistent with a graph signal with low variations among close nodes, and thus a low frequency graph signal. However, when the network is affected by random failures, the snapshot contains randomly positioned failed nodes and the nodes with similar signal values would be spread throughout the whole network randomly. This is congruous with a high frequency graph signal. Therefore, the energy of the spectrum is focused more on the higher components in the case of random failures. However, in the case of an epidemic the energy of the spectrum is concentrated more on the lower part of the spectrum.

As it can be seen, $ECRL$ and $ECRH$ metrics need the knowledge of eigenvectors of the graph Laplacian matrix. In large scale networks, calculating the eigenvalues and eigenvectors might be cumbersome. Here, we introduce another metric which does not need such knowledge and can be incorporated in the detection problem in hand.

Definition 3. *Smoothness of a graph signal S with respect to the underlying graph $G = (V, E, A)$ is defined as:*

$$Smoothness(G, S) = \sum_{i \in V} \left[\sum_{j \in N_i} (A_{i,j})[S(j) - S(i)]^2 \right]^{\frac{1}{2}}, \quad (6)$$

where N_i is the set of vertices which are one hop away from vertex v_i.

It can be seen that a signal which has similar values among close nodes in the graph possess a small value of smoothness. The above metric is defined based on the one-hop neighbors of a node. However, in highly irregular graph structures, we may need the information from the nodes within a certain distance of a node to maximize the probability of success in the detection. For this purpose, we define the generalization of this metric, *generalized smoothness* as:

$$G\_Smoothness(G, S, \eta) = \sum_{i \in V} \left[\sum_{j \in N_i^{\eta}} (B_{i,j})[S(j) - S(i)]^2 \right]^{\frac{1}{2}}, \quad (7)$$

where

$$N_i^{\eta} = \{j | dist(v_i, v_j) < \eta\} = \{j | B_{i,j} < \eta\}. \quad (8)$$

Here, $B_{i,j}$ is an element of the distance matrix and represents the shortest distance between the nodes v_i and v_j.

G_smoothness as an extension for the smoothness metric considers the difference in the signal amplitude between a node and the rest of the nodes within a certain range. In particular, if the range is set to be the diameter of the graph, a node's signal value will be compared to those of the rest. In the case of an epidemic occurrence, this metric value should be much smaller as compared to the case of random failures

IV. Detection of Infections in Various Networks

A. Algorithm design

We design our algorithm based on the statistical characteristics of the introduced metrics. For a given network, we first extract the statistical characteristics for all the metrics in case of random failures by checking sufficiently many scenarios. Note that this process may be considered as the learning part of the algorithm, which can be done offline. When a snapshot of the network condition is given, the algorithm performs two steps. Firstly, it constructs the graph signal based on the given snapshot as described in Section III. Secondly, it calculates the values of the metrics for the constructed graph signal. If these values lie outside the statistical interval of random failures, the algorithm considers the snapshot as an epidemic. Otherwise, the algorithm reports random failures.

B. Simulation setting

In the problem in hand, a given snapshot of the network condition can correspond to two possible scenarios: the presence of random failures and an epidemic. The goal is to distinguish between these two possible scenarios, so the detection of each scenario is necessary and important. Hence, the utilized metric for performance comparison should be able to reflect the success of the performed algorithms in both of these cases. For this purpose, the probability of detection is used, defined as:

$$P_D = P(H_0|H_0)P_0 + P(H_1|H_1)P_1, \quad (9)$$

where P_0 and P_1 indicate the probability in which the network is infected by random failures and epidemic, respectively. For simulations we set $P_0 = P_1 = 0.5$ and examine 2000 different scenarios. Simulations are performed on two irregular graph structures, Erdős–Rényi (ER) and scale-free (SF) graphs, and a regular grid graph. Here, the considered regular grid has a size of 4900, where all vertical edges have the same weight 1, and all the horizontal edges have the same weight 10. For ER and SF graphs, size of 100 is considered, and weights of the edges are chosen uniformly at random between one and 100. Given a generated snapshot for each of the two possible cases, 10% of infected nodes are changed to healthy nodes to reflect false negatives, and an extra 10% of infected nodes are added to the snapshot to reflect false positives. This change in the network condition eliminates the possibility of having one connected component as an epidemic, and reflects the noise in the observation. In simulations we examine the performance of the proposed algorithm for both the case where there is a single initial infected node and the case where there are multiple initial infected nodes. The former is examined on the grid graph, while the latter is examined on the ER and SF graphs, for which the epidemic starts from

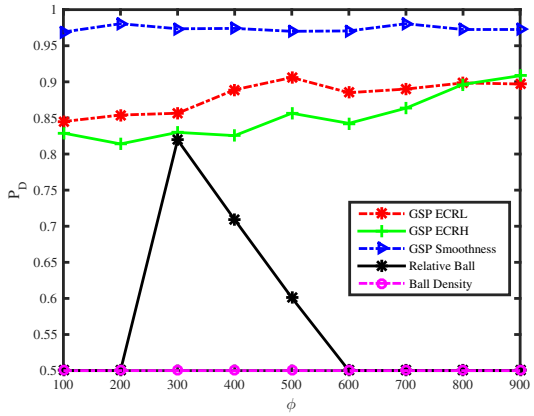


Figure 3: Detection probability of the proposed GSP-based algorithms and ball density-based algorithms in an Erdős–Rényi graph with $p = 0.0075$ and $N = 1000$ with respect to the number of infections.

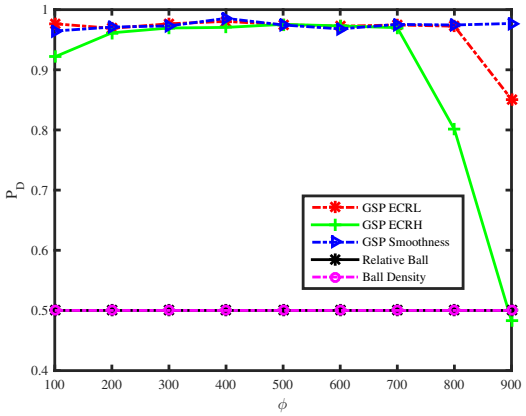


Figure 4: Detection probability of the proposed GSP-based algorithms and ball density-based algorithms in a scale-free graph with $l = 1$ and $N = 1000$ with respect to the number of infections.

four different initial nodes that are far away from each other. Also, the performances of the ball density and relative ball density algorithms introduced in [7] are compared with the proposed algorithm. There are two parameters needed in ball density-based algorithms, radius of the ball and a threshold of comparison. Authors in [7] obtained the radius of the ball empirically in the simulations without explicitly reporting it. In our simulations, we check a wide range of radii of the ball and use the one which results in the best performance, which is $300 * ratio\ of\ the\ infected\ nodes$ in ER and SF graphs, and $190 * ratio\ of\ the\ infected\ nodes$ in the grid structure. Also, we set the value of the threshold to be 3 for all the cases. Note that we set $\gamma = \alpha = 0.3$ for $ECRH$ and $ECRL$ metrics.

C. Performance comparison

Fig. 3 compares the performance between different metrics and the ball density-based algorithms for an Erdős–Rényi graph with the wiring probability $p = 0.0075$[4]. Here, ϕ denotes the number of nodes reported to be infected. As can be seen from the figure, GSP-based algorithms with different metrics perform much better than the ball density-based algorithms [7]. This superiority is observed in the complete range of ϕ, indicating the GSP-based algorithms are robust to the number of reported infections, and not sensitive to false positive or false negatives. It is worth mentioning that the ball density algorithm exhibits a 50% P_D, which is equivalent to making a random guess. This is due to the incapability of the ball density algorithm which leads to failure in the detection of epidemics with multiple initial seeds.

Fig. 4 demonstrates the performance comparison where the underlying graph is a scale-free graph with parameter $l = 1$[5]. Scale-free graph is an important model for social networks. This graph has a small diameter which implies any two nodes can be reached within a few hops. Due to this special characteristic, the ball density-based algorithms will naturally be more likely to fail, as evidenced by their poor detection performance shown in the figure. This is due to the fact that any ball will be dense (containing many nodes), resulting in a false detection of the infection boundaries. On the contrary, the GSP-based algorithms will not be affected by the dense connections. In addition, we can see that the GSP-based algorithms (especially with the smoothness metric) are also capable of differentiating random failures from epidemics stemming from multiple initial infections.

Fig. 5 illustrates the detection probabilities of different algorithms on a regular grid. This simulation is done as a baseline for comparison between GSP-based and ball density-based algorithms. This is due to the fact that ball-density algorithms are more suitable for regular graph structures where epidemic stemmed from one initial node. Despite the weight differences in vertical and horizontal edges, a grid is a rather regular structure compared to ER and SF graphs. As can be seen from the figure, our GSP-based detection algorithms perform well with all three metrics ($P_D \geq 0.95$) at all infection ratios. When the infection ratio is higher than 50%, neither the ball density nor the relative ball density algorithm is an eligible option. In fact, the ball density algorithm always reports an epidemic in every snapshot as which results in the worst detection probability $P_D = 0.5$.

For better insights, the probability of detecting random failures $P(H_0|H_0)$ and the probability of detecting epidemics $P(H_1|H_1)$ are depicted in Fig. 6 and Fig. 7, respectively. Fig. 6 shows the excellent performance in terms of probability

[4]Wiring probability is defined as the probability of having a connection between any two nodes.

[5]Here, l indicates the number of edges between a newly added node and the previously developed graph at each iteration in constructing a scale-free graph.

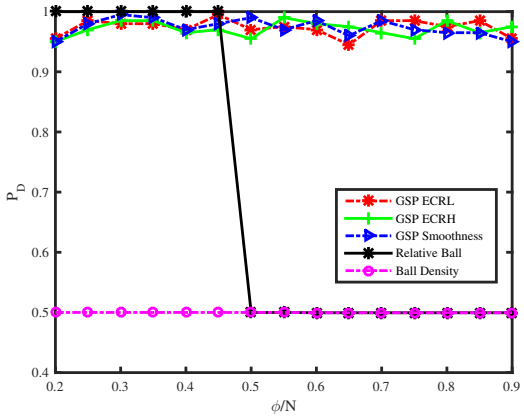


Figure 5: Detection probability of the proposed GSP-based algorithms and ball density-based algorithms in a grid of 4900 nodes, where all vertical edges have the same weight 1, and all the horizontal edges have the same weight 10.

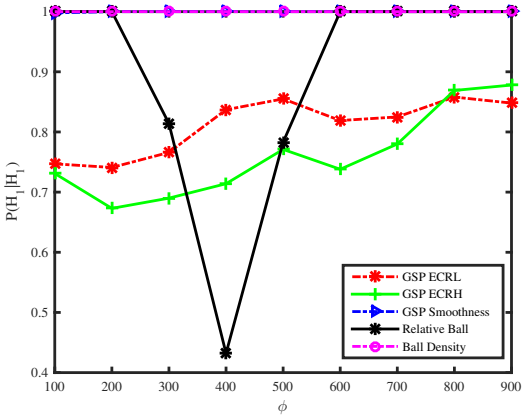


Figure 7: Comparison between the GSP-based algorithms, the ball density algorithm, and the relative ball density algorithm in an Erdős–Rényi graph with $p = 0.0075$ in terms of probability of detecting epidemics $P(H_1|H_1)$.

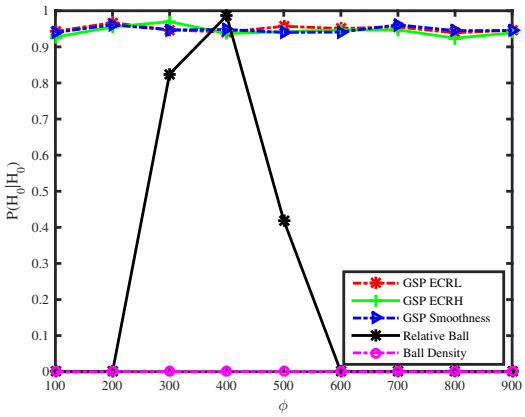


Figure 6: Comparison between the GSP-based algorithms, the ball density algorithm, and the relative ball density algorithm in an Erdős–Rényi graph with $p = 0.0075$ in terms of probability of detecting random failures $P(H_0|H_0)$.

of detecting random failures for all the introduced metrics. Moreover, the ball density algorithm has the worst performance and it never works in the presented range of infection size. Fig. 7 illustrates that the ball density algorithm along our GSP-based algorithm with the smoothness metric have the best performance compared to the other metrics. It can be concluded from these two figures that the ball density algorithm always reports an epidemic no matter the snapshot contains an epidemic or not. This fact is also demonstrated in Fig. 3.

V. Conclusion

In this paper, we utilized tools from the GSP area as an efficient way of overcoming the problem of infection detection in heterogeneous networks. Based on the classical metrics in the GSP area, we defined metrics to explore this problem. We designed an effective detection algorithm based on our defined metrics and compared its performance to the existing algorithms in this area. Results presented in the paper indicate the superiority of the proposed GSP-based algorithm compared to the state-of-the-art algorithms. More precisely, the algorithm has a good performance in regular and irregular graph structures, and it is able to detect an epidemic when it stemmed from multiple initial nodes in a network.

References

[1] B. Ayhan, M. Y. Chow, and M. H. Song, "Multiple discriminant analysis and neural-network-based monolith and partition fault-detection schemes for broken rotor bar in induction motors," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 4, pp. 1298–1308, June 2006.

[2] J. Yin, Q. Yang, and J. J. Pan, "Sensor-based abnormal human-activity detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 8, pp. 1082–1090, Aug 2008.

[3] H. Li and Z. Han, "Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3554–3565, November 2010.

[4] D. Centola, "The spread of behavior in an online social network experiment," *Science*, vol. 329, no. 5996, pp. 1194–1197, 2010.

[5] A. L. Lloyd and R. M. May, "How viruses spread among computers and people," *Science*, vol. 292, no. 5520, pp. 1316–1317, 2001.

[6] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Detecting epidemics using highly noisy data," in *Proceedings of the Fourteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '13. New York, NY, USA: ACM, 2013, pp. 177–186.

[7] ——, "Local detection of infections in heterogeneous networks," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 1517–1525.

[8] 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 Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[9] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, June 2014.

[10] ——, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, April 2013.