Detecting Cascades from Weak Signatures

Eli A. Meirom, *Member, IEEE*, Constantine Caramanis, *Senior Member, IEEE*, Shie Mannor, *Senior Member, IEEE*, Ariel Orda, *Fellow, IEEE*, and Sanjay Shakkottai, *Fellow, IEEE*

Abstract—Inspired by cyber-security applications, we consider the problem of detecting an infection process in a network when the indication that any particular node is infected is extremely noisy. Instead of waiting for a single node to provide sufficient evidence that it is indeed infected, we take advantage of the graph structure to detect cascades of weak indications of failures. We view the detection problem as a hypothesis testing problem, devise a new inference algorithm, and analyze its false positive and false negative errors in the high noise regime. Extensive simulations show that our algorithm is able to obtain low errors in the high noise regime by taking advantage of cascading topology analysis.

Index Terms—Epidemic detection, hypothesis testing

1 Introduction

Interconnection is at the core of the functionality of our modern infrastructure, spreading ideas, technology and information. But network and virus attacks, from denial of service, to theft of personal information, and even to state-driven cyberwarfare, are problems of increasing relevance and potential threat. An early diagnosis of a spreading epidemic is critical. Anti-virus software relies on and searches for the signature of a *known* worm, malware or virus—a reliable signal of infection. But how do we intercept malware spread in actionable time-scales, and critically, *before we know what is spreading*—and thus before we have malware-specific signatures? In this paper we ask what can be done when such strong signatures do not exist, or in any case are not known in time for early detection. In the absence of signatures that pinpoint the presence of malware, or expert medical opinion and tests that diagnose human illness, we necessarily resort to not signatures, but *indications*. Malware, similarly to many human illnesses with long dormancy periods, produces slight deviations in system behavior, for example, hard disk spin up, or a spike in network activity, or a function call accessing photographs and immediately afterwards SMS. And as with human illness, these indicators may be extremely weak, possibly symptomatic of nothing abnormal, and even if caused by malware, they might quickly disappear if the malware goes dormant.

We ask: can we use indications of abnormality—we call these *flags*– that are so weak that on their own they are *statistically indistinguishable from noise*, to make an accurate global diagnosis about the presence of a spreading epidemic? The central conceptual contribution of this paper is that by harnessing the dynamics of the spread, even these weak signals can be used to diagnose an epidemic spread accurately and quickly.

Our model, defined in Section 3, is an extreme model of the scenario discussed above. We adopt such an extreme model precisely to bring the focus on the power of network and the signature of the spread itself to reveal the presence of an epidemic. We assume that under "normal" (uninfected) behavior, a given node produces flags according to a Poisson process. Infections proceed according to an SI epidemic. We assume that upon infection, a single flag is raised, after which point the henceforth infected node resumes the same (random) behavior pattern prior to its infection. The problem is to decide, based on the flags, whether there is an epidemic spreading in the network, or if the flags' appearance are consistent with normal behavior. It is easy to see that viewed in isolation, a single node's flag pattern offers little more statistical strength than random guessing. The network spread, however, correlates these flags; we characterize when this correlation is detectable, and when it allows us to separate the two hypotheses with high probability. Our results depend on the network topology, and indicate that the more local expansion a network possesses, the easier epidemics are to detect. The analysis of our main algorithm hinges on analyzing the spread of SI epidemics on different graph topologies; for this, we leverage results from first passage percolation on grids [1], [2], [3], as well as limit theorems for random graphs [4], [5].

2 Prior and Related Work

Recently, numerous works have explored the epidemic spread characteristics, such as the dependence of the infection rate on the topology (e.g., [6]), or the time it will take an epidemic to be detected by a fixed sensor network [7]. These are *forward problems*: given the initial conditions, predict the epidemic evolution.

- *E. A. Meirom, S. Mannor, and A. Orda are with the Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 3200003, Israel. E-mail: bloodymeli@gmail.com, {shie, ariel}@ee.technion.ac.il.*
- *C. Caramanis and S. Shakkottai are with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712. E-mail: constantine@utexas.edu, shakkott@austin.utexas.edu.*

Manuscript received 18 Aug. 2016; revised 16 Apr. 2017; accepted 22 June 2017. Date of publication 17 Oct. 2017; date of current version 11 Dec. 2018. (Corresponding author: Eli A. Meirom.)
Recommended for acceptance by M. Lelarge.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TNSE.2017.2764444

Our work focuses on the inverse question: given an online, dynamic, noisy map of the network activity, infer, at each point in time, if there was an epidemic outbreak. In addition to correct detection, *early detection* is important. Related inference questions have gained considerable attention in other contexts, such as estimating the epidemic parameters [8], [9], [10], or the identification of the epidemic source (e.g., [11], [12], [13], [14], [15]).

A similar inference problem is presented in [16], [17], [18], [19]. However, these works address a static problem: equipped with a static network map of reporting nodes, containing a large number of false positives (and false negatives, should an epidemic occur), decide whether an epidemic occurred or not. These works do not address the evolution of the epidemic and accessible information in time. Furthermore, The problem at hand is more difficult, as the epidemic is hidden both in the *time dimension*—every single epidemic flag is indistinguishable from the multitude of normal activity flags surrounding it in time, and in the *network dimension*—we detect an epidemic even when the fraction of infected nodes is infinitesimally small with respect to network size.

3 The Basic Model and Problem

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote our graph, with $\mathcal{V}$ the set of n nodes, and $\mathcal{E}$ the set of edges between nodes in $\mathcal{V}$. The next two definitions specify *normal* and *epidemic* behavior.

Definition 1. *Under* normal behavior, *node $v \in \mathcal{V}$ outputs flags independently of all other nodes, and according to a Poisson process of rate μ. In a time period $[0,t]$, the expected number of flags raised by any given node is μt. We assume that the Poisson process is time-homogeneous, though this does not seem to be critical for our results.*

Definition 2. *In the case of* epidemic, *we assume that infections* spread *over edges according to a standard SI model: when a node becomes infected, it starts an independent exponential clock of rate λ on each incident edge. When a clock expires, it infects its other incident node (if not already infected). The instant a node becomes infected, it outputs a flag deterministically. Prior to this event, and after this event, the node's behavior is precisely "normal" behavior as described above (see Fig. 1).*

The motivation for this setup is that flags are some (very weak) indicator of anomaly. For instance, consider malware spreading over a mobile network. A flag could correspond to "increased network activity over a one-second duration." While this flag is symptomatic of a malware, various benign applications could also generate such a flag, and hence we model normal nodes raising flags as a Poisson process at rate μ. An infected node will also output such a Poisson sequence of flags (due to benign applications); however in addition, may outputs flags due to malicious behavior. We take the extreme view here, and model malware as provoking *exactly one additional flag when the node becomes infected*. Crucially, this extra flag is indistinguishable from a "normal" flag to any observer.

Thus, from a single node's perspective, we are trying to distinguish between a Poisson process at rate μ, and another process which is a Poisson process at identical rate however with *one additional flag* embedded within the infinite Poisson

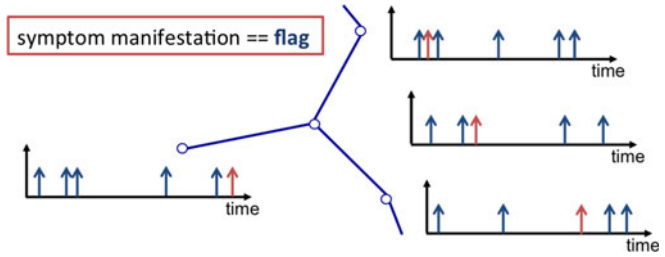

Fig. 1. Malware Spread: A flag is an indicator of abnormality, such as increased network activity over a one second time window. While a node will exhibit such a flag when it gets infected, it is clear that the node will exhibit such flags even when not infected (usual network usage). In our model, each node has a baseline process that generates noise flags (usual usage) as a Poisson process at rate μ (shown by blue impulses). The hypothesis testing question is the following: Is there a "causal tree" of epidemic flags (one flag per node shown by red impulses) overlaid on top of this baseline noise? In other words, an infected node manifests a epidemic flag *exactly once* when it gets infected, each of its neighbors manifest a flag exactly once when they get infected and so on. However, the one extra flag, even if present, is indistinguishable from a noise flag (i.e., impulse's color in the figure is not revealed).

flag train at some arbitrary instant of time! As is apparent, this task is statistically impossible with any degree of confidence. Surprisingly, we see that by appropriately correlating across a network, we can "amplify" the epidemic signal, and indeed distinguish between normal and infected networks with high probability.

We let $F^{(i)}_{\text{normal}}(t_1,t_2)$ denote the number of flags output by node i in time $[t_1,t_2]$ under normal behavior, and $F^{(i)}_{\text{normal}}(t) \triangleq F^{(i)}_{\text{normal}}(0,t)$; thus $F^{(i)}_{\text{normal}}(t)$ is a Poisson random variable of rate μt. Let $I^{(i)}(t_1,t_2)$ denote the indicator function that node i became infected in the interval $[t_1,t_2]$, and let $I^{(i)}(t) \triangleq I^{(i)}(0,t)$. Thus, the total number of flags by node i in interval $[0,t]$ is given by $F^{(i)}_{\text{total}}(t) = F^{(i)}_{\text{normal}}(t) + I^{(i)}(t)$. Indeed, note that as long as $\mu t = O(1)$, the probability of correctly diagnosing an infection from the flags in $[0,t]$ is bounded away from 1, and for μ fixed, this correctness probability quickly decays to 50 percent (that of random guessing) as t grows. While only the process $F^{(i)}_{\text{total}}(t)$ is observable, we denote by $F^{(i)}_{\text{normal}}(t)$ the number of flags due to normal behavior in both the epidemic and non-epidemic setting.

We show that with knowledge of the spreading dynamics and the network structure, and knowledge of the flags raised, we can correctly (with probability tending to 1) diagnose the presence of an epidemic.

The Testing Problem. We wish to distinguish between no epidemic in our graph at any point during the time window being considered, versus the case of some infection starting at some node, at some point during our time window. We can reduce this problem to a simple testing problem as follows. Consider *the universe of only two possibilities*. Assume there are two possible events in the universe: H_0: No epidemic originated at any node in time interval $[0,\tau]$ (so all nodes normal); and $H^{(i)}_{\text{epidemic}}$: An epidemic started at node i at time 0. The fundamental hypothesis testing problem we solve uses the network structure and the flags $F^{(i)}_{\text{total}}(0,\tau)$ raised in $[0,\tau]$ to separate these hypotheses. Now, consider the *more general universe* with more than two hypotheses, where we observe the system over a time interval $[0, T+\tau]$, where either no epidemic began at any node in $[0,T]$, or an epidemic began at some node and at time $t \in [0,T]$.

While all processes are continuous, there is an inherently discrete event-driven clock defined by each raised flag, and by definition, epidemics can only begin at the raise of a flag. Consider a series of flags $\{t_i, v_i\}$, ordered lexicographically by increasing time t_i and source node v_i. Given an algorithm that detects a time-zero epidemic given flags in $[0, \tau]$, any monitoring process would apply the (time-shifted) algorithm over $[t_1, t_1 + \tau]$, $[t_2, t_2 + \tau]$, ... The false positive rate requires a union bound over all flags in $[0, T]$. This quantity is proportional to $|V|$ and T. We show that the concentrations involved are exponential, which immediately implies that for essentially all results presented here, we can successfully perform early detection.

Therefore, we focus on the *testing sub-problem*: no epidemic occurs during time $[0, \tau]$, or, alternately, an epidemic begins at some specific node $i \in \mathcal{V}$ at time 0. The type I error, or false positive probability – the probability of false detection—is denoted by $e_I(i, \tau)$. Type II error, or false negative probability—the probability of missed detection—is denoted by $e_{II}(i, \tau)$. If the node i is fixed, we drop the index and refer to $e_I(\tau)$ and $e_{II}(\tau)$.

The key parameters in this definition are τ and $|\mathcal{V}|$. The larger the value of τ, the smaller the probability of a false negative, but the number of potentially infected nodes is greater, failing *early detection*. The larger the graph, and the larger the overall window of time over which we wish to control false positives, the more difficult it is to do early detection.

4 Algorithm and Meta-Theorem

Intuition: Testing with Oracle Windows: The statistical fluctuations in the number of normal flags raised in $[0, \tau]$ under normal behavior is $O(\sqrt{\tau})$:ĂŞ much greater than the *single* "extra" flag raised due to epidemic. The central idea of the algorithm is to correlate events using the network structure. Suppose an oracle reveals there is an epidemic that began from a specific node $i \in \mathcal{V}$ at some time t_0 (which implies that there must have been a flag at node i at time t_0). We shift time so $t_0 = 0$. Suppose further, that the oracle reveals the order and the *time slices* of width Δ within which each node was infected. Looking at those small time slices, we should see at least one epidemic flag in each. Now, in any fixed Δ-width time slice, the probability of a flag under normal (no epidemic) behavior is Poisson of rate $\mu\Delta$. If Δ is small (such that $\mu\Delta \ll 1$), independence of the noise processes across nodes implies that the number of flags under the nomal hypothesis is roughly $\mu\Delta|V|$. On the other hand, with the epidemic hypothesis, the number of observed flags observed is roughly $(\mu\Delta|V| + |V|)$, which is much greater than the expected number of flags under the normal hypothesis. Further the excess flags are very unlikely to occur by a random fluctuation. Unlike the single node setting, the signal now is "amplified" by the oracle windows, and is much stronger than the noise.

Identifying without an Oracle: As we have no such oracle, the key is to use the dynamics of the spread and the topology of the graph, to determine which nodes to look at, in which order, and for which time slices. Note that we cannot take the window size to be very small or we would miss most of the epidemic-related flags. Specifically, even with network knowledge, we will be in a setting where $\Delta = \omega(1)$, and typically increases with $|V|$. Thus, the oracle intuition above breaks down because the noise within the window is $\mu\Delta|V|$, which is order-wise larger than the signal with strength $|V|$.

However, as we will see below, we can still detect an epidemic with high probability in this setting. *The key idea is that the signal needs to rise only above the standard deviation of the noise, and need not be stronger than the mean noise.* Continuing with the oracle notation, a more nuanced view indicates that under the normal hypothesis, the number of flags is roughly $(\mu\Delta|V| \pm \sqrt{\mu\Delta|V|})$. Under the epidemic hypothesis, the number of flags is $(|V| + \mu\Delta|V| \pm \sqrt{\mu\Delta|V|})$. Thus, it is clear that if $|V| = \omega(\sqrt{\mu\Delta|V|})$, detection is possible even in the presence of noise that seemingly drowns the signal. Additionally, even if we do not observe all nodes, and the windows are wrongly chosen for many of the observed nodes, the epidemic can be detected as long as the *observed signal* (number of epidemic flags within the estimated windows) *rises above the standard deviation of the noise within the estimated windows.* The FlagCounter algorithm below, and the following theorems make this discussion rigorous.

Algorithm 1. FlagCounter

Input: Patient Zero, $i \in \mathcal{V}$; Nodes $\mathcal{S} \subseteq \mathcal{V}$, Time Windows $\mathcal{W} = \{[w_j, w_j + \Delta_j]\}_{j \in \mathcal{S}}$, Flags $\{F^{(j)}_{\text{total}}([w_j, w_j + \Delta_j])\}_{j \in \mathcal{S}}$, threshold χ.
Output: EPIDEMIC or NORMAL
  for all $j \in S$ **do**
    $F(\mathcal{S}, \mathcal{W}) \leftarrow F(\mathcal{S}, \mathcal{W}) + F^{(j)}_{\text{total}}([w_j, w_j + \Delta_j])$.
  end for
  if $F(\mathcal{S}, \mathcal{W}) \leq \chi$ **then**
    return NORMAL
  else
    return EPIDEMIC
  end if

Algorithm. The FlagCounter algorithm takes as input the set of nodes $\mathcal{S}$ to be considered, time windows $\mathcal{W}$ about each node, and the full flag process F_{total}. It counts the flags $F(\mathcal{S}, \mathcal{W})$ raised by the nodes in $\mathcal{S}$ in windows $\mathcal{W}$. if this is above the threshold χ, it declares an epidemic at node i at time 0, otherwise it reports normal behavior.

We now give a meta-theorem that provides structure for the main results of this paper. It expresses the following simple idea. Given any set of nodes $\mathcal{S}$ and corresponding window sizes $\mathcal{W}$, let $N(\mathcal{S}, \mathcal{W}) = \sum_{j \in \mathcal{S}} I^{(j)}(w_j, w_j + \Delta_j)$ be the number of infections that occurred in the windows. While we cannot *observe* this number, in the sequel we bound it via measure concentration. If $N(\mathcal{S}, \mathcal{W})$ is much bigger than *the variations in the number of flags*, $F(\mathcal{S}, \mathcal{W})$ seen at the nodes in $\mathcal{S}$ during windows $\mathcal{W}$, then the probability of a false positive is very small. By the same token, in the event of an epidemic, we see the flags due to epidemic, and to normal behavior. If the latter cannot have big downward deviations, compared to $N(\mathcal{S}, \mathcal{W})$, then again the probability of false negatives will be very small.

We define $r_t = r_t(\mathcal{S}, \mathcal{W}) = \mathbb{E} \sum_{j \in \mathcal{S}} F^{(j)}_{\text{normal}}(w_j, w_j + \Delta_j)$, i.e., r_t is the expected number of flags raised by nodes in $\mathcal{S}$ during windows $\mathcal{W}$ under normal behavior. The set $\mathcal{S}$ is

often, as we shall later see, a subset of the neighbors of the candidate patient zero. The index t is the close of the last window, i.e., the latest observation time. The total window lengths and the average window lengths are denoted $W_{\text{total}} = \sum_{j\in\mathcal{S}} \Delta_j = W_{\text{avg}}|\mathcal{S}|$. Note that $r_t/\mu = |\mathcal{S}| \cdot W_{\text{avg}}$.

The following conditions essentially say that the windows are not too big, and at the same time, they do capture most of the epidemic-flags:

Key Conditions. (A) W_{total}, the total sum of the window lengths, scales more slowly than the square of the number of nodes in $\mathcal{S}$, i.e., : $W_{\text{total}}/\mu = o(|\mathcal{S}|^2)$. This implies that $W_{\text{avg}} = o(|\mathcal{S}|)$. Let $\alpha > 0$ be such that $W_{\text{avg}} = O(|\mathcal{S}|^{1-\alpha})$. (B) With probability $1-\xi$, where $\xi \to 0$[1] in the number $|\mathcal{S}|$ of nodes we consider, the windows specified in $\mathcal{W}$ capture a significant number of the times when the nodes in $\mathcal{S}$ become infected, i.e., $\sum_{j\in\mathcal{S}} I^{(j)}(w_j, w_j+\Delta_j) \geq |\mathcal{S}|^{1-\eta}$ for $\eta \geq 0$. Trivially, reducing α reduces η as well; we need a fine balance between these two quantities, $\alpha > 2\eta$.

Theorem 3. *Let $\mathcal{S}$, $\mathcal{W}$ be given fixed sets. We run the algorithm on the flag sequence generated by nodes in $\mathcal{S}$ in time windows in $\mathcal{W}$, using threshold value $\chi = r_t + r_t^{1/2+\epsilon}/2$, for $\epsilon = (\alpha-2\eta)/2(2-\alpha)$. Assume conditions (A) and (B) hold.*

- *False Positives.* The probability our algorithm declares we have an epidemic, when in fact the flags are generated by normal behavior is exponentially small: $e_I(t) \leq \exp(-|S|^{2\epsilon}/3)$.
- *False Negatives.* The probability our algorithm declares we have no epidemic when in fact an epidemic began at node i at time zero, is exponentially small: $e_{II}(t) \leq \exp(-|S|^{2\epsilon}/3)$.

We can extend this result to the general case, in which the algorithm continuously monitors the flag appearance for a time duration T. Namely, we apply the FlagCounter algorithm on every flag observed in every node in the network during $[0, T+\tau]$. The following result immediately follows using a union bound.

Theorem 4. *Consider a network of size $|V|$, monitored over time $[0, T+\tau]$. The probability that an epidemic is falsely declared (false positive) is bounded by $2|V| \cdot T \cdot \exp(-|S|^{(\alpha-2\eta)/(2-\alpha)})$. In particular, the* FlagCounter *algorithm succeeds when $\mathcal{S} \in \omega(\log^x n)$ for exponentially long operating time, $T = |V|^y$ for any $y > 0$.*

We call this a *meta-theorem* because we have pushed the work into checking Conditions (A) and (B). That is, its statements become interesting only when we can characterize when we can find node sets $\mathcal{S}$ and windows $\mathcal{W}$ that satisfy the two assumptions with high probability, and when we are able to show that their size is controlled. Thus, in the sequel, we consider various graph topologies, and show that Conditions (A) and (B) are satisfied, and hence Theorem 3 holds, while giving bounds on how many nodes are infected by the time we detect the infection. We note that this is primarily a function of the topology of the network, which we therefore try to understand. The parameter λ controls how quickly an infection spreads across a particular edge between an infected and non-infected node. Indeed, it controls the spreading rate, but it is less important in terms of controlling the number of infected nodes we must allow before we can detect an infection with high probability.

Accordingly, the remainder of this paper considers specific graph topologies. In Section 5 we consider d-dimensional grids, which model geometric graphs. In Section 6 we consider Erdös-Renyi graphs in the sparse regime where a giant component emerges. In Section 7, we consider graphs that have very high-degree hubs, such as stars and power law graphs, and there we see the hub-property suggests a more efficient algorithm.

5 Grids

Grids model geographic connections, where geographic proximity and graph distance are correlated. In this section, we consider the d-dimensional grid, and specialize the meta-theorem above. We show that we can accurately detect an epidemic before it spreads beyond a logarithmic portion of the network.

Our main tool in tracking the epidemic frontier, is the so-called Shape Theorem (Theorem 1.7 from [1]). As t increases, it is clear that the footprint of the infection also expands. Kesten's results provides concentrations on the set of infected nodes. Specifically [1] shows that for a given t, there exist an inner shell and an outer shell, such that the all nodes within the inner shell are infected and all nodes outside the outer shell are not infected, with high probability. The difference between the radii of the shells is of order $t^{\gamma(d)}$, where $\gamma(d) \leq 1$ for every dimension d. We use this result of Kesten [1] to pick the right windows about each node in the set, in order to catch almost all the epidemic-flags. For every node i, we set $f_i = \inf\{t | i \in t\mathcal{L}_0\}$, where $\mathcal{L}_0$ is some primitive shape that depends on the lattice dimension (see also Proposition 15 in Section 9 for a more formal statement).

Proposition 5. *Pick $l = (\log n)^{1/d}$. Let $\mathcal{S}$ be the shell of outer radius l and inner radius $l/2$: $\mathcal{S} = \{j | l/2 \leq d(\vec{0}, j) \leq l\}$. Define the windows $\mathcal{W}$ about each node $j \in \mathcal{S}$ as $[w_j, w_j+\Delta_j] = [f_j - f_j^{0.6}, f_j + f_j^z]$, for $z = 1 - \frac{1}{(2d+4)}$. Let t denote the last time instant considered in $\mathcal{W}$. Then in the event of an infection initiating at node 0 at time 0, with high probability at least half of the epidemic flags occur in the set $\mathcal{S}$, in the windows $\mathcal{W}$.*

Putting this in the context of our Meta-Theorem, we have the following:

Corollary 6. *The conditions of the Meta-Theorem are satisfied with: (A) $W_{\text{avg}} = O(|\mathcal{S}|^{1-\alpha})$, where $1-\alpha = (1-1/(2d+4))/d$; and (B) with high probability, the windows $\mathcal{W}$ capture at least half the epidemic flags in $\mathcal{S}$, hence we can take $\eta = 0$. Therefore the epidemic is detected with high probability before the infection travels farther than $(\log n)^{1/d}$ nodes from its source, i.e., before more than $\text{poly}(\log n)$ nodes are infected.*

In Section 9, we show further that by the final time t considered in $\mathcal{W}$, with high probability, at most 1/2 of the infected nodes lie outside of set $\mathcal{S}$.

6 Erdös Renyi Graphs

We specialize our results to the sparse Erdös Renyi graph $G(n,p)$, and show how to choose the algorithm parameters

1. We note that we do not need ξ to decay with any particular rate, though for some of our results a fast rate is immediate.

(the set $\mathcal{S}$ and windows $\mathcal{W}$) so that the meta theorem can be applied. We assume that in the event of an epidemic, the source node is part of a component of size $\Theta(n)$. That is, we assume that the network is past the percolation threshold, $np > 1$, and that "patient-zero" is in the giant component. We define the following constants: the mean degree is $d = np$, $\beta = \frac{d}{d-1} > 1$, and $\gamma = (d-1)^{-1}$. Our results show our algorithm converges correctly for $d > 2$.

We choose the set $\mathcal{S}$ as the set of nodes within a ball of radius $k \leq \alpha \log n$ about the source i, where the distance metric is the hop-count, and for every node in this ball set the window to be $[0, c \cdot k]$, irrespective of its distance from the source. We take c large enough so that $I(c) = c - 1 - \log c > (1+\epsilon)d$. It is clear from this definition that $W_{\text{avg}}|\mathcal{S}| = o(|S|^{1-\alpha})$ for $\alpha = 1 - \epsilon$ for any $\epsilon > 0$, and therefore Condition (A) is satisfied. We next show that Condition (B) is satisfied, namely, that the windows in $\mathcal{W}$ capture many of the epidemic flags raised by nodes in $\mathcal{S}$. Like the grid setting, we also have $\eta = 0$ for Condition (B), i.e., all the nodes inside of $\mathcal{S}$ infected with high probability.

The proof outline is as follows, (for the full details see Section 9). While we do not have a shape theorem as we do in the grid setting, we do know that locally, Erdös-Renyi random graphs are trees; this allows us to leverage so-called *speed* conditions on trees. We show that the set of nodes in the ball $\mathcal{S}$ is bounded very close to what one would expect for a d-regular depth k tree, namely, d^k. We then show that with high probability, by time ck (recall that we have re-scaled time so that $\lambda = 1$) all nodes in $\mathcal{S}$ would be infected in the event of an epidemic starting at node i. Finally, we use the speed condition to show that not too many infected nodes lie outside of $\mathcal{S}$.

Proposition 7. *Assume $d > 2$. Let $\mathcal{S}$ and $\mathcal{W}$ be as above. With high probability, all nodes in $\mathcal{S}$ are infected in $\mathcal{W}$. In particular, the key conditions are satisfied with (A) $\alpha = 1 - \epsilon$, for any $\epsilon > 0$, and (B) $\eta = 0$.*

This then gives us the following:

Corollary 8. *In the event of an epidemic, we can detect it before it travels more than $\alpha \log n$ hops from its point of origin, and therefore when a sublinear (vanishing) fraction of the total nodes are infected.*

7 Hub-FlagCounter Algorithm

Real world networks have been observed to have heavy tails in their degree sequences, and much study has gone into understanding the prevalence and properties of so-called power-law graphs [20]. A key characteristic of such graphs is a *hub property*: these graphs have a special collection of nodes called hubs, such that nearly every node is close to one of these hubs, and these hub-nodes have significantly higher degree than other nodes. The star network, cliques and wheel graphs are all examples of easy-to-visualize hub networks, while social networks, hyperlinks on the world-wide-web, and the Internet AS graph are real-world examples where empirically the power-law phenomenon (and hence the presence of hubs) has been observed. For such networks, one can dramatically improve on our FlagCounter algorithm, by directly leveraging the presence of the hubs.

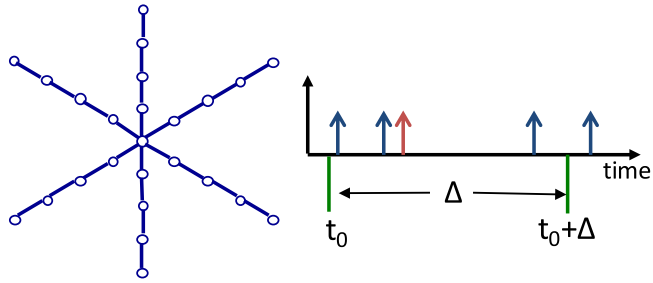

Fig. 2. An extended star network with $\sqrt{n}$ spokes, each spoke having a line of $\sqrt{n}$ nodes. The hub node has $\sqrt{n}$ neighbors. The Hub-FlagCounter counts the number of flags within a window of interval $\Delta \sim \Theta(1)$ at each of the neighbors of the hub node. The window begins at a random time t_0 that is the purported time of the infection hitting the hub.

The star graph is stylized, yet captures the essential two properties of hub networks: wherever the epidemic begins, it quickly spreads to the hub, and then a statistically detectable number of nodes become infected very soon after that. This intuition suggests the following variation of the FlagCounter algorithm. We call it the Hub-FlagCounter algorithm:

Define a set $\mathcal{H}$ of hubs. For every flag in a node $i \in \mathcal{H}$, shift time to 0, and count the number of flags in node i' s nearest neighbors within $[0, \Delta_i]$. If this number exceeds χ_i, declare an epidemic; else issue no warning. We can choose the threshold χ_i to be a function of the degree of node i.

The Hub-FlagCounter algorithm needs to monitor only the hubs for flags, unlike the FlagCounter algorithm which tests *every flag* in *every node*. This improves the false positive rate and reduces the computational resources required for monitoring the network activity. Next, since we explore extremely short windows, we have the potential to very rapidly detect an epidemic, compared to if we were to run the FlagCounter algorithm. Finally, only information on the nearest neighbors is required, rather than knowledge on the flag appearance in the extended local environments.

The performance of the Hub-FlagCounter algorithm depends on two key properties of the graph: how quickly a node reaches a hub, and how many of the hub's neighbors are infected by the time the hub becomes infected. We illustrate the power of the Hub-FlagCounter algorithm compared to the FlagCounter algorithm by considering an extended-star-network on n nodes.

Extended Star Network. Let $G = (V, E)$ be a network on $(n+1)$ nodes, with 1 center node, and $\sqrt{n}$ spokes each of length $\sqrt{n}$, as depicted in Fig. 2. In this setting, the spreading properties of an epidemic are highly dependent on the location of patient zero, and the nodes to which the epidemic has spread. Initially, the epidemic spreads along the spoke patient zero is on until it hits the hub. Once the center (hub) node becomes infected, the rate of infection-flags significantly increases, and many more flags appear in a very short period of time. Thus, a good algorithm would define a set $\mathcal{S}$ and windows $\mathcal{W}$ to be topology-dependent, and moreover, dependent on the (potential) evolution of the epidemic. One can think of the Hub-FlagCounter algorithm in this context: it considers a set $\mathcal{S}$ only around the potentially-infected hub, and considers a very short period of time.

To this end, in the above extended-star network, let $\mathcal{S}$ be the hub and its 1-hop neighbors, so that $|\mathcal{S}| = (1 + \sqrt{n})$.

Recall that we have normalized time so that μ, the rate of normal flags, is equal to 1. The rate of infection is λ. For each of $\sqrt{n}$ one-hop neighbors, monitor it in the time window $[t_0, t_0 + \lambda^{-1}\delta]$, where t_0 is the (purported) time of infection of the hub[2]. Since the infection spreads along each edge as an exponential random variable with parameter λ, it immediately follows that a fixed neighbor of the hub manifests an epidemic flag is $(1 - e^{-\delta})$. Thus, a constant fraction of the neighbors get infected within the chosen window; the probability that this does not happen asymptotically goes to 0 (from Hoeffding's inequality). With this choice of nodes and windows, it is now clear that probability of detection with HUB-FLAGCOUNTER goes to one, and that the false positive probabilities goes to zero. To see this, the expected total number of normal flags scales as $\lambda^{-1}\delta\sqrt{n}$, and the total number of epidemic flags, denoted by $E[\sum_{i\in\mathcal{S}} I^{(i)}(t_0, t_0 + \lambda^{-1}\delta)]$, scales as $\Theta(\sqrt{n})$. Thus, the standard deviation of the number of normal flags is $o(E[\sum_{i\in\mathcal{S}} I^{(i)}(t_0, t_0 + \lambda^{-1}\delta)])$, with appropriate concentrations following from Hoeffding's inequality.

Finally, we note that by choosing δ to be small (but constant), we do not expect the epidemic to spread too far away form the hub node during the interval $[t_0, t_0 + \lambda^{-1}\delta]$; specifically, the expected number of infected nodes in the network is still $\Theta(\sqrt{n})$.

It is worth comparing this to the FLAGCOUNTER algorithm. If it is to choose $\mathcal{S}$ and $\mathcal{W}$ independently of topology and location of initial detection, it must choose the time window to be $[0, \lambda^{-1}\sqrt{n}]$: terminating earlier would miss the scenario where the epidemic begins far out on one of the spokes, while adjusting the window to not begin at time 0 might miss most of the infection signal, in the scenario where the epidemic begins at or very close to the hub. While one can show that the FLAGCOUNTER algorithm will successfully detect the epidemic, it cannot guarantee to do so before a constant fraction of nodes are infected. The HUB-FLAGCOUNTER algorithm, on the other hand, succeeds in detecting the epidemic in the worst-case when $O(\sqrt{n})$ nodes have become infected – this is the case when the epidemic starts far out on one of the spokes, and the HUB-FLAGCOUNTER algorithm must wait until the infection reaches the hub.

We demonstrate this further on so-called Forest Fire graphs in Section 8. Forest Fire graphs are a class of random graphs unlike Erdös-Renyi graphs, that exhibit the presence of Hubs. We show that in those less stylized examples, as with the star example here, the HUB-FLAGCOUNTER algorithm greatly outperforms the FLAGCOUNTER algorithm.

8 EXPERIMENTS

We simulate the performance of FLAGCOUNTER and HUB-FLAGCOUNTER in various network settings. In the plots, n corresponds to network size, λ is the SI epidemic spread rate, and μ is the rate of flags for the normal Poisson process at each node. The inspected set size is $m = |\mathcal{S}|$.

Training. The theoretical analysis in Section 3 provides an asymptotic prescription for all the parameters required for a correct application of the algorithm. Nevertheless, for small networks and predefined inspected set size, we have found that the algorithms run well if we learn the optimal parameters on a training set. For our experiments, we fixed the inspected set size and time windows according to the asymptotic prescription, and trained the various classifiers on a training set composed of n_s standard activity sessions and n_s epidemic scenarios. Unless specified otherwise, we used $n_s = 1000$ for each scenario. The reported results were obtained on a disjoint testing set, composed of the same number of standard activity sessions and epidemic scenarios. In any real scenario, the training set could be obtained using either historical data, or simulated epidemic cascades on the real topology.

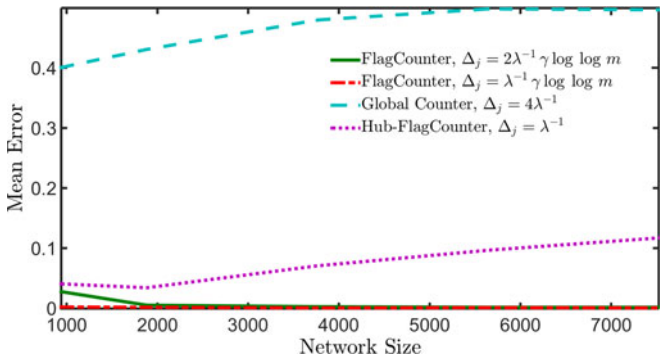

Fig. 3. The mean error rate of the FLAGCOUNTER algorithm on an Erdös-Renyi graph. The window function specified in Theorem 7 provides superior results, regardless if the window width is $2\log\log m$ or $\log\log m$. The HUB-FLAGCOUNTER performs fairly well, while the GLOBALCOUNTER algorithm has a success rate close to a random guess.

Baseline Algorithm. As a baseline algorithm, we considered a threshold test applied on the flags over the entire network in a given time window. Should this number exceed the optimal threshold, as found on the training set, an epidemic state is declared. The optimal threshold in each scenario was evaluated as above. We refer to this naive algorithm as the GLOBALCOUNTER algorithm. We tested this algorithm and found that the global counter algorithm failed to achieve correct early diagnosis in challenging settings where the FLAGCOUNTER and HUB-FLAGCOUNTER algorithms performed very well (for example, Fig. 3).

Additional Implementation Details. For the HUB-FLAGCOUNTER algorithm, a hub is a node with degree $\geq \lceil \log n \rceil$. The mean error is defined as the average of the false positive and the false negative probabilities. When a classifier was unable to detect an epidemic before it infected a giant network component of size αn (here, $\alpha = 0.75$), the instance was declared a false negative.

8.1 Random Networks

We tested our algorithms on different types of random networks exhibiting different characteristics: Erdös-Renyi networks, and Forest Fire graphs.

Our theoretical results in Sections 6 and 7 provide an understanding of when we would expect the FLAGCOUNTER and HUB-FLAGCOUNTER algorithms to do well. We test them both here, and present the results)error rates) in Fig. 3, on the $G(n, p = 3/n)$. As our theoretical results from Section 6 predict, the FLAGCOUNTER does very well, with error rates quickly going to zero in the size of the network. The HUB-FLAGCOUNTER algorithm, however, does not share this

2. Note that since we do not know t_0, we run a parallel instance of HUB-FLAGCOUNTER each time we see a flag at the hub.

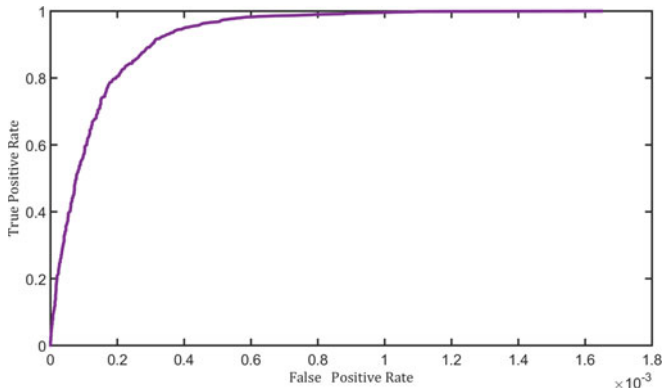

Fig. 4. The ROC curve for the Erdös Renyi network $G(n = 6000, p = 3/n)$, when using the FlagCounter Algorithm. The inspected set size is $\log^2(n)$.

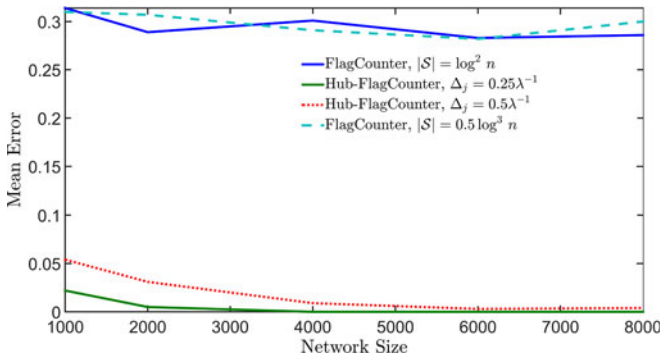

Fig. 5. The mean error rate on an Forest Fire network for various windowing schemes. The Hub-FlagCounter algorithm provided to be best-in-class, even as the inspected set size of the flag counter was increased to $\log^3 n/2$.

success. As detailed in Section 7, the Hub-FlagCounter algorithm is able to detect an epidemic only once it hits a hub. By altering the definition of a hub based on node degree, we can tradeoff between accuracy and the expected infected fraction before detection. However, as an Erdös-Renyi network does not have a heavy tail; even by carefully choosing the definition of a hub, the Hub-FlagCounter seems unable to drive error rates to zero.

We display the tradeoff between false positive and false negative error rates for the FlagCounter algorithm on the 6,000-node Erdös-Renyi graph in Fig. 4, where we give the ROC curve. Our results show that the algorithm achieves very low false positive and false negative error rates.

The Forest Fire Model. The degree distribution of many real-world networks follows a power law, and therefore such networks are not well-modeled by an Erdös-Renyi graph.

The Forest Fire random network model [7], [21], was devised in order to incorporate numerous statistical attributes of real world networks, including degree distributions one does not observe in an Erdös-Renyi graph. We chose the forward-burning and the back-burning probabilities of the Forest Fire model as (0.37,0.32), as this configuration corresponds to slowly densifying networks. As this network typically has a large number of well-connected hubs, one of the first infected nodes after a contagion outbreak is likely to be a hub.

We compare our FlagCounter and Hub-FlagCounter algorithms on networks generated in this way, and provide the results in Fig. 5. We see that the plots perform as our theory from Section 7 would suggest. The overhead in infected nodes and detection time of the Hub-FlagCounter algorithm is low, and accordingly, this algorithm is able to quickly drive the error rates to zero, without requiring many nodes to become infected. This is in contrast to the FlagCounter algorithm's performance, even with our best efforts to tune window sizes (we plot two different parameters).

8.2 Real World Networks

In this section, we consider two real-world networks. The first, is the Internet graph, which is made up of 27,894 nodes. The second considers the communication graph made up of 33,969 unique email addresses from the Enron network. Using these networks, we explore the advantages in performance, but also the susceptibility to noise, of the Hub-FlagCounter algorithm, as described in Section 7.

Misconfiguration of Internet router, also known as BGP attacks, are often characterized by an increase in packet drop out. However, drop outs occur naturally in the Internet due to various reasons, e.g., shutdown of servers and links for maintenance. In many settings, it is essential to have an extremely low false negative probability, while allowing for a minimal rate of false positives. Our algorithms may be applied in order to differentiate such attacks from normal activity. We test our two algorithms in such a scenario on the Internet graph. We performed a training on a short interval of duration 0.2 time units (the random flag rate was scaled to one), and optimized the threshold for very low values of false negative probability (less than 0.03). Then, we measured the mean number of false positive counts for operating duration of up to 10 times of the training duration. We present the false positive counts per Autonomous System (AS) in Fig. 6.

Recall the intuition Hub-FlagCounter algorithm—once a hub gets infected, a strong "signal" results soon-after due to a sudden increase in the number of infected nodes. The Internet is known for having a large number of high-degree nodes (hubs) such as Tier 1 or Tier 2 ASs; this thus provides a good setting to empirically evaluate the Hub-FlagCounter algorithm (as discussed in Section 7).

Indeed in our empirical study in Fig. 6, the Hub-FlagCounter provided perfect classification and zero false

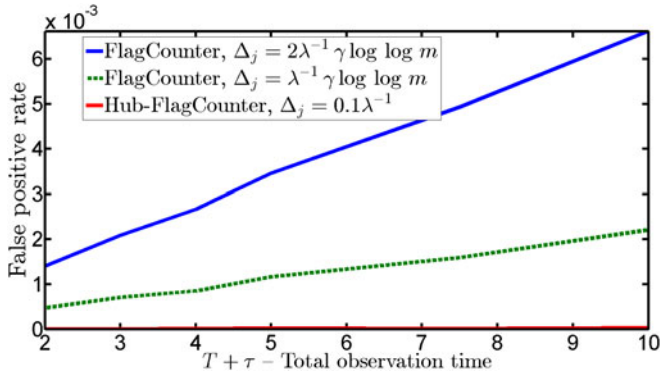

Fig. 6. Internet Graph: The false positive per node error rate as a function of time after on the Internet graph, composed of 27,894 nodes. The training session consisted of 250 trials, and each testing session had 100 trials.

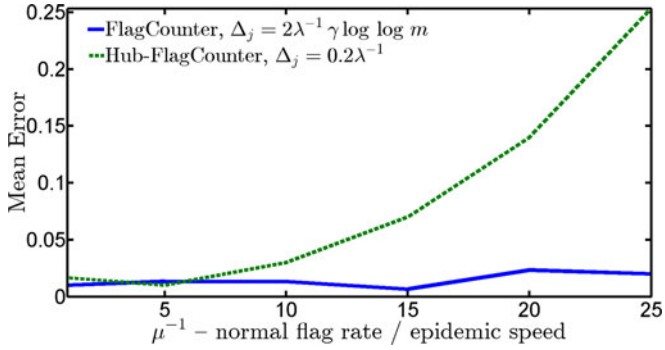

Fig. 7. Enron Graph: The mean classification error as a function of the ratio of normal activity flagging rate to the epidemic speed. The normal activity rate was scaled to one. The plot was generated using 150 epidemic cases and 150 normal activity sessions as a training session, repeated by an identical validation session.

positive counts in all trials. This plot suggests that on a moderate size network, with a few tens of thousands of nodes, the application time of our algorithm can be a *few orders of magnitude longer* than the training time, with just a few false positive alarms. Thus, this empirical study bears out the intuition motivating the Hub-FlagCounter developed in Section 7.

We next consider the performance of our algorithms on the Enron graph. This graph models communication between nodes, and hence is appropriate to study the potential for malware to spread from node to node.

A stealthy malware may be difficult to spot. One possible approach to cloak malware activity is to reduce the frequency of resources it consumes, or even to stay dormant for long periods of time. This behavior could be simulated by changing the ratio of the normal flag rate to the epidemic speed. As this increases, the signal (flags from the epidemic spreading) gets weaker compared to the noise (the normal flag process). We investigate the performance of our algorithms in this setting, i.e., as the activity rate of the malware decreases. Fig. 7 shows the performance of our algorithms on a network of 33,969 emails addresses. In this network [22] (Enron dataset), two addresses are linked if an email was sent from one to the other. This figure shows that the FlagCounter algorithm has a very weak linear dependence on the malware spreading speed, and operates well even as the appearance rate of the random flags is 25 times greater than the infection rate μ. Namely, the algorithm can successfully identify the malware, even when its activity is highly concealed by normal activity.

On the other hand, we see that the Hub-FlagCounter algorithm is more sensitive to increased levels of noise. The fluctuations in the number of flags during normal activity scales with the square root of the inspected set size. Since the Hub-FlagCounter inspects a smaller environment, it is prone to such fluctuations, and as the flagging density increases, its performance deteriorates faster than the FlagCounter algorithm. Nevertheless, in absolute terms, its error remains low even when the epidemic spreading speed is extremely slow compared to the normal activity flagging speed.

9 Proof Details

9.1 Meta-Theorem

In this section we prove our meta-theorem (Theorem 3). For convenience, we restate the relevant definitions here.

Definition 9. *Given a set of nodes $\mathcal{S}$ and corresponding window sizes $\mathcal{W}$, let $N(\mathcal{S},\mathcal{W}) = \sum_{j\in\mathcal{S}} I^{(j)}(w_j, w_j+\Delta_j)$ be the number of infections that occur in the windows. We define $r_t = r_t(\mathcal{S},\mathcal{W}) = \mathbb{E}\sum_{j\in\mathcal{S}} F^{(j)}_{\text{normal}}(w_j, w_j+\Delta_j)$, i.e., r_t is the expected number of flags raised by nodes in $\mathcal{S}$ during windows $\mathcal{W}$ under normal behavior. The index t is the close of the last window, i.e., the latest observation time. We also define the total window lengths and the average window lengths by $W_{\text{total}} = \sum_{j\in\mathcal{S}}\Delta_j = W_{\text{avg}}|\mathcal{S}|$. Note that $r_t/\mu = |\mathcal{S}|\cdot W_{\text{avg}}$.*

For convenience, we term a flag that was generated by the epidemic process upon infection an *epidemic-flag*, while all other flags are called *normal-flags*. Obviously, the epidemic-flags are indistinguishable from the normal-flags.

The following lemma bounds the false positive probability $e_I(i,\tau)$ of the hypothesis testing sub-problem. It is an immediate application of the Chernoff bound.

Lemma 10. *Consider an inspected set $\mathcal{S}$ and windows $\mathcal{W}$. The probability that the total number of normal-flags in the corresponding time windows exceeds $r_t + r_t^{1-y}$ or is less than $r_t - r_t^{1-y}$ is bounded by $\exp(-r_t^{1-2y}/3)$. Namely,*

$$\mathbb{P}\left(\left|\sum_{j\in\mathcal{S}} F^{(j)}_{\text{normal}}(w_j, w_j+\Delta_j) - r_t\right| \geq r_t^{1-y}\right) \leq \exp(-r_t^{1-2y}/3).$$

Proof. A) First, we rescale all the normal-flag Poisson processes. For every node $j\in\mathcal{S}$, we set $\Delta'_j = \mu_j\Delta_j$. Then, the number of normal-flags in $[w_j, w_j+\Delta_j]$ is distributed according to a Poisson process with rate 1 and duration Δ'_j. The total number of normal-flags is distributed as a Poisson process with rate 1, applied for a duration of

$$\sum_{j\in\mathcal{S}} {\Delta'}_j = \sum_{j\in\mathcal{S}} \mu_j\Delta_j = r_t.$$

The expected number of flags is r_t. Set $x = r_t^{1-y}/r_t = r_t^{-y}$. Using a standard concentration inequality for Poisson processes, we have

$$\mathbb{P}\left(\left|\sum_{j\in\mathcal{S}} F^{(j)}_{\text{normal}}(w_j, w_j+\Delta_j) - r_t\right| > r_t^{1-y}\right) \leq \exp(-r_t H(x)) \leq \exp\left(-r_t^{1-2y}/3\right),$$

where $x = r_t^{-y}$, and $H(x) = (1+x)\log(1+x) - x$. The last inequality follows since as $r_t\to\infty$, $x\to 0$ and $H(x) = x^2/2 - O(x^3)$. □

The FlagCounter algorithm dictates a specific choice of time slice windows which may depend on the purported source node. Hence, in general, for every node j and source node i, we have $w_j = w_j(i)$, and $\Delta_j = \Delta_j(t)$, though for simplicity, we omit the explicit dependence on the source node i in the following.

Definition 11. *Recall that $\sum_{j\in S} I_j(w_j; w_j+\Delta_j)$ denotes the number of epidemic flags that fall in the inspected time window. Let ξ denote the probability that this number is less than $|\mathcal{S}|^{1-\eta}$, i.e.,*

$$\xi \triangleq \mathbb{P}\left(\sum_{j\in S} I_j(w_j; w_j+\Delta_j) \leq |\mathcal{S}|^{1-\eta}\right).$$

We now turn to analyze the false negative probability. The following lemma states that if our choice of windows and inspected set $\mathcal{S}$ is such that we catch at least $|\mathcal{S}|^{1-\eta}/2$ epidemic-flags out of the maximal value of $|\mathcal{S}|$ epidemic-flags, then the false negative probability tends to zero.

Lemma 12. *Assume an epidemic has started from node i. Consider the FlagCounter algorithm with threshold $\chi = r_t + r_t^{1/2+\epsilon}/2$, for $\epsilon = (\alpha - 2\eta)/2(2-\alpha)$. Assume that the probability that less than $|\mathcal{S}|^{1-\eta}$ are detected is ξ and $\xi \to 0$, and $1 - \alpha > \eta$. Then, the false negative probability satisfies*

$$e_{II}(t,i) \leq \exp(-|\mathcal{S}|^{(\alpha-2\eta)/(2-\alpha)}) + \xi.$$

Proof. The algorithm successfully classifies an epidemic if there occurred at least $\chi = r_t + r_t^{1/2+\epsilon}/2$ flags in the specified windows. The probability that less than $|\mathcal{S}|^{1-\eta}$ epidemic-flags were detected is bounded by ξ which goes to zero in $|\mathcal{S}|$, and thus $e_{II}(t,i)$ is bounded by

$$\mathbb{P}\left(\sum_{j\in\mathcal{S}} F^{(j)}_{\text{normal}}(w_j, w_j+\Delta_j) \leq r_i + \frac{r_t^{1/2+\epsilon} - |\mathcal{S}|^{1-\eta}}{2} - \frac{|\mathcal{S}|^{1-\eta}}{2}\right). + \xi$$

Since $|\mathcal{S}|^{1-\eta} \in \omega\left(r_t^{1/2+\epsilon}\right)$, we have

$$e_{II}(t,i) \leq \exp\left(-|S|^{(\alpha-2\eta)/(2-\alpha)}\right),$$

by Lemma 10. □

The Meta-Theorem (Theorem 3) is an immediate result of the last two lemmas. The Meta-Theorem Corollary then follows by noting that the total number of normal flags raised across the network in an interval $[0,T]$ concentrates exponentially in $T|V|$. the theorem and union bound then gives the proof of the corollary.

9.2 Grids

In this section we discuss the case of a d dimensional grid. We show that we can specify an inspected set $\mathcal{S}$ and corresponding windows such that the probability that we observe at least $|\mathcal{S}|/2$ epidemic-flags tends to 1 exponentially.

Our main tool is the result from Theorem 1.7 in Kesten [1] which shows that for a given t, there exists an inner shell and an outer shell, such that all nodes within the inner shell are infected and all nodes outside the outer shell are uninfected. The difference between the radii of the shells is of order $t^{\gamma(d)}$, where $\gamma(d) \leq 1$ for every dimension d. We restate here an immediate corollary of Kesten's theorem. For simplicity of notation, we assume the initial infection time is at $t_0 = 0$ and the source of the epidemic is at the origin.

For a given set $\mathcal{A} \in \mathbb{R}^d$, and $x \in \mathbb{R}$, the set $x\mathcal{A}$ is the set resulting from bloating $\mathcal{A}$ by a factor x, $x\mathcal{A} \triangleq \left\{\vec{r} \in \mathbb{R}^d \middle| \vec{r}/x \in \mathcal{A}\right\}$.

Theorem 13. *[Kesten] [1] There exists a set $\mathcal{L}_0$ and constants c_1 to c_5 such that for $x = t^\beta$ for any constant $0.5 \geq \beta > 0$,*

$$\mathbb{P}\left\{\mathcal{S}(t) \subset \left(1 + x/\sqrt{t}\right)t\mathcal{L}_0\right\} \geq 1 - \exp(-c_1 x),$$

and

$$\mathbb{P}\left\{\mathcal{S}(t) \supseteq \left(1 - c_3 t^{-1/(2d+4)}\right)t\mathcal{L}_0\right\} \geq 1 - \exp\left(-c_2 t^{\frac{d+0.9}{2d+4}}\right),$$

for large t ($t \to \infty$).

Definition 14. *For every node i, set $f_i = \inf\{t | i \in \mathcal{L}_0 t\}$.*

We use Kesten's theorem to pick the right window sizes.

Proposition 15. Pick $m \in \omega(1)$, and set $l = m^{1/d}$. Let $\mathcal{S}$ be the shell of outer radius l and inner radius $l/2$: $\mathcal{S} = \{j | l/2 \leq d(\vec{0}, j) \leq l\}$. Define the windows $\mathcal{W}$ about each node $j \in \mathcal{S}$ as $[w_j, w_j + \Delta_j] = [f_j - f_j^{0.6}, f_j + f_j^z]$, for $z = 1 - \frac{1}{(2d+4)}$. Let t denote the last time instant considered in $\mathcal{W}$. Then in the event of an infection initiating at node 0 at time 0, with high probability at least half of the epidemic flags occur in the set $\mathcal{S}$, in the windows $\mathcal{W}$.

Proof. We denote the set of nodes infected by time t as $\mathcal{A}_0(t)$. The probability that an epidemic-flag did not occur in j before w_j is greater than the probability that at time $t = w_j$ the infection is contained within the shape $f_j\mathcal{L}_0$.

$$\mathbb{P}\big(I_j(0; w_j) = 0\big) \geq \mathbb{P}\big(\mathcal{A}(w_j) \subset f_j\mathcal{L}_0\big).$$

Using our choice of $w_j = f_j - f_j^{0.6}$,

$$f_j \geq w_j + w_j^{0.6}.$$

Substituting $f_j = \big(1 + x/\sqrt{t}\big)t$ and $t = w_j$ in Kesten's theorem, we have

$$\big(1 + x/\sqrt{w_j}\big)w_j \geq f_j.$$

Using our choice of $w_j = f_j - f_j^{0.6}$, we have

$$f_j \geq w_j + w_j^{0.6}.$$

Namely,

$$\big(1 + x/\sqrt{w_j}\big)w_j \geq w_j + w_j^{0.6}.$$

Solving for x,

$$x \geq w_j^{0.1}$$

and using Kesten's theorem (Theorem 13), we have

$$\mathbb{P}\big(\mathcal{L}(w_j) \subset f_j\mathcal{L}_0\big) \geq 1 - \exp\left(-c_1 w_j^{0.1}\right),$$

where the c_i are constants. Therefore, the probability that an epidemic flag precedes the specified window, is

$$\mathbb{P}\big(I_j(0; w_j) = 1\big) \leq \exp\left(-c_1 w_j^{0.1}\right).$$

Similarly, the probability that an epidemic-flag in node j is observed by time $w_j + \Delta$ is

$$\mathbb{P}\big(I_j(0; w_j + \Delta) = 1\big) \geq \mathbb{P}\big(\mathcal{A}(w_j + \Delta) \supseteq f_j\mathcal{L}_0\big).$$

Since $w_j + \Delta_j = f_j + f_j^z$, where

$$z = 1 - \frac{1}{(2d+4)}. \tag{1}$$

we have, using the second inequality in Kesten's theorem,

$$\mathbb{P}\big(\mathcal{A}(w_j+\Delta)\supseteq f_j\mathcal{L}_0\big)\geq 1-\exp\Big(-c_2\big(w_j+\Delta_j\big)^{\frac{d+0.9}{2d+4}}\Big).$$

Therefore, the probability an epidemic-flag in node i is not observed by time $w_i+\Delta$ is bounded by

$$\mathbb{P}\big(I_j(w_j;0)=0\big)\leq\exp\Big(-c_2w_j^{0.3}\Big),$$

for all $d>1$. By using a union bound, the probability that a flag is not detected in node i in the associated window is bounded by $2\exp(-c_2w_j^{0.1})$, with $c_3=\min\{c_1,c_2\}$.

Hence, we can define a random indicator variable X_i for each node i, indicating that an epidemic flag was not detected in node i. The success probability $p_i\triangleq\Pr(X_i=1)$ satisfies

$$p_i\leq 2\exp\Big(-c_3w_j^{0.1}\Big).$$

Recall that the volume of a d dimensional sphere is

$$v=\frac{\pi^{d/2}}{\Gamma(1+d/2)}.$$

Note that as w_j monotically increases with $d(0,j)$, we have $w_j\leq d(0,j)/2$, and in particular

$$p_j\leq 2\exp\big(-c_3l^{0.1}\big),$$

for some c_3. For convenience, $p_0\triangleq 2\exp(-c_3l^{0.1})$.

Next, we show that the probability that less than $|\mathcal{S}|/2$ epidemic-flags are detected tends to zero for $n\to\infty$. The latter (see definition) is bounded by using Markov's inequality,

$$\begin{aligned}\xi(\alpha,\mathcal{S},\mathcal{W})&\leq\mathbb{P}(G\leq|\mathcal{S}|/2)\\&\leq\mathbb{P}\left(\sum_{i\in\mathcal{S}}X_i\;>\;|\mathcal{S}|/2\right)\\&\leq\frac{2|\mathcal{S}|\max_{i\in\mathcal{S}}p_i}{|\mathcal{S}|}\leq 2p_0\\&\to 0\end{aligned}$$

since and $p_0\to 0$.

In conclusion, we have proved that for our specific choice of $\mathcal{S}$ and windows, the probability that less than $|\mathcal{S}|/2$ epidemic flags are not detected tends to zero. □

Note that in order for Theorem 3 to hold, we may pick $\mathcal{S}\in\omega(1)$. However, in order for Theorem 4 to hold we must pick $\mathcal{S}\in\omega(\log^x n)$ for some x, or equivalently, $m\in\omega(\log^x n)$.

9.3 Erdös-Renyi Graphs

We now show how to choose the algorithm parameters for the Erdös-Renyi graph $G(n,p)$. We shall assume, should an epidemic arise, that the source node is part of a connected component of size $\Theta(n)$. That is, we assume that the network is past the percolation threshold, $np>1$, and that "patient-zero" is in the giant component. To simplify notation, we assume time is rescaled so that we can take $\lambda=1$, i.e., the infection virulence is normalized. We use $d=np$ to denote the average degree of the graph.

constants γ and c so that $\gamma\ll 1$ – in particular so that $2c\gamma\log d\ll 1$, and c large enough so that $I(c)=c-1-\log c>\log d$. For a fixed $i\in\mathcal{V}$, let the set $\mathcal{S}=\mathcal{S}_k$ be the nodes of depth (distance) up to k from i. We assume that $k<\gamma\log n$, where n is the total number of nodes in our graph. Choose the window sizes uniformly across nodes as $[w_j,w_j+\Delta_j]=[0,ck]$.

We need to show that under the epidemic scenario (beginning at node i), most of the nodes in $\mathcal{S}$ become infected in the specified window period. This guarantees that the "signal" will be strong enough, and hence our detection algorithm will succeed. Then, in order to certify that we have succeeded in performing early detection, we need to guarantee that not too many nodes outside $\mathcal{S}$ have become infected.

The key technical result we require is an upper and lower bound on the number of elements in the set $\mathcal{S}$.

Lemma 16. *For any $\epsilon>0$, the set $\mathcal{S}$ of nodes a distance at most $\gamma\log n$ from the root node i satisfies: $(d(1-\epsilon))^k\leq|\mathcal{S}|\leq(d(1+\epsilon))^k$, w.h.p. in n.*

This follows from concentrations in the neighborhood growth rate of random graphs, e.g., Lemma 5.16 in [20]. In particular, choosing γ small enough (recall $k<\gamma\log n$) guarantees that subset $\mathcal{S}$ is in fact a tree of average degree d (see also [23]).

Using this, a simple union bound reveals that with high probability, all nodes in $\mathcal{S}$ are infected by time ck. Indeed, for any node $v\in\mathcal{S}$, it is at most k hops away from the root node, and hence the probability it is not infected in $[0,ck]$ is the probability that k rate-one exponentials sum to more than ck, i.e., for $X_j\sim\exp(1)$, iid,

$$\mathbb{P}(\sum_{j=1}^{k}X_i\;>\;ck)\leq\exp(-kI(c)),$$

where $I(c)=1-c-\log c$ is the rate-function of a rate-one exponential. Now a union bound combined with the upper bound of the above lemma reveals that all nodes in $\mathcal{S}$ are infected with probability at least $1-\exp(-k(I(c)-\log d(1+\epsilon)))$. Choosing c so that $I(c)>\log d(1+\epsilon)$, the probability of all nodes in $\mathcal{S}$ being infected goes to 1 exponentially in k. Thus Condition (B) is satisfied with $\eta=0$; moreover, that $k<\log|\mathcal{S}|$, and therefore Condition (A) is satisfied, as claimed in Proposition 7 with any $\alpha<1$.

Finally, consider the set of nodes that are infected by the last time considered, i.e., by time ck, and denote this set by $\hat{S}$. The speed condition for Erdös-Renyi random graphs [17] shows that $\hat{S}$ must be contained in a ball of radius at most $1.1ck$ centered at the root (node i). By the lemma above, however, this ball contains at most $(1+\epsilon)d^{1.1ck}$ nodes. Since $k<\gamma\log n$, the depth is at most $2c\gamma\log n$, and since (by our choice) γ satisfies $2c\gamma\log d<1$, this implies, in particular, that at most a vanishing fraction of the network has been infected by time ck.

10 Additional Details and Experiments

In this section we revisit the Erdös-Renyi graph and the FlagCounter algorithm, and demonstrate better results using more refined window selection techniques.

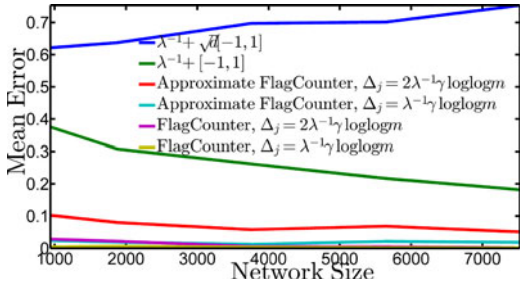

Fig. 8. The mean error rate of the FlagCounter algorithm on an Erdös-Renyi graph. These plots represent the error in an *exterior shell* windowing scheme.

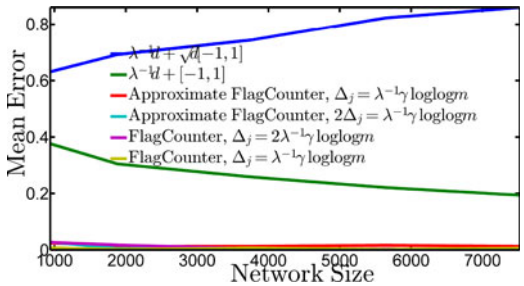

Fig. 9. The mean error rate of the FlagCounter algorithm on an Erdös-Renyi graph. These plots represent the error in the *fixed window* scheme.

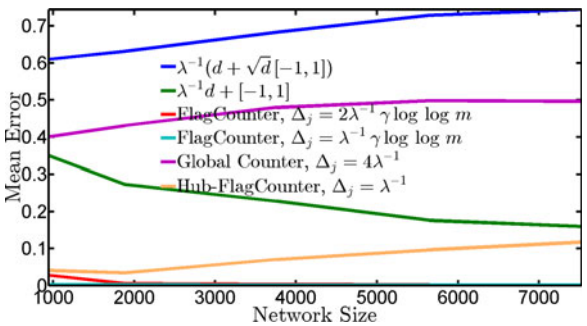

Fig. 10. The mean error rate of the various algorithms on an Erdös-Renyi graph. These plots represent the error in the *peeled windows* scheme.

Window Selection Variants. In Section 5 we have considered a particular network, a $d-$ dimensional grid, in which both the start time w_j and the window width Δ_j are an increasing function of node j's distance from the source. We call this choice of windows, in which the window parameters are a function of the distance, *peeled windows*. On the other hand, in Section 6 an alternative scheme has been presented. In this scheme, both the window start time and the window width do not depend on the distance from the source, namely, it is

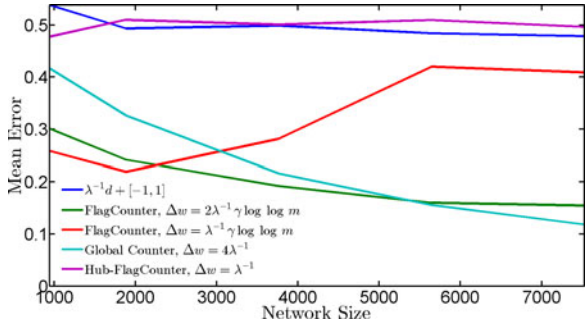

Fig. 11. The mean error rate of the various algorithm on an Erdös-Renyi graph with an inspected set size of $3\log n$. These plots represent the error in the *peeled windows* scheme. The error is fairly high, and does not seem to tend to zero.

a *fixed window* scheme. This choice proved successful, since for small environments, an Erdös-Renyi network is approximately a tree. For a constant degree tree (with $d > 1$), a constant fraction of the nodes are at the leaves. It is therefore sufficient to detect only the epidemic generated flags at the outer shell of the tree. In fact, we might hope to get a better signal to noise ratio by considering a test based on the flag activity at this outer shell only, and disregard all events in the lower shells. This suggest a third scheme, an *exterior shell* windowing scheme, in which the window width is zero for all nodes except nodes on the exterior shell of the inspected set. In the *peeled windows* and *exterior shell* windowing schemes, for every node j at distance d from the source, we set $m = m(j)$ to be the number of nodes in the $d-1$ ball about the source node. This applies to the window functions $[w_j, w_j + \Delta_j] = (\lambda(1-c))^{-1}(\log m/4 + a \log\log n[-1,1])$ where $a = 0.5$ and $a = 1$.

In all schemes the windows are set according to the number of neighbors in the d-shells about the source (a d-shell is the set of nodes at distance d from the source). This requires the enumeration of the d shells of a source for every raised flag. Alternatively, an approximation can be obtained by replacing the exact, node-specific values, by their mean network values. As can be seen in Figs. 8, 9, and 10, this approximation works well for short time windows, but performance deteriorates for large windows.

We applied different windowing schemes using different window function. Figs. 8, 9, and 10 describe the results, while Table 1 present the values with the corresponding standard deviations. The FLAGCOUNTER algorithm works well in all schemes, while there is no scheme which saves the other algorithms from an unacceptable classification rate. Note that it is reasonable that the error rate may be

TABLE 1
The Mean Error and Corresponding Standard Deviations in the *Peeled Windows* Detection Scheme

n	1000	2000	6000	8000
$\lambda^{-1}(d+\sqrt{d}[-1,1])$	0.610 ± 0.163	0.631 ± 0.196	0.729 ± 0.276	0.744 ± 0.300
$\lambda^{-1}d+[-1,1]$	0.350 ± 0.237	0.273 ± 0.250	0.176 ± 0.238	0.160 ± 0.232
Approximate FlagCounter, $\Delta = 2\lambda^{-1}\gamma\log\log m$	0.099 ± 0.190	0.068 ± 0.154	0.070 ± 0.160	0.049 ± 0.123
Approximate FlagCounter, $\Delta = \lambda^{-1}\gamma\log\log m$	0.014 ± 0.065	0.014 ± 0.072	0.013 ± 0.063	0.008 ± 0.042
FlagCounter, $\Delta = 2\lambda^{-1}\gamma\log\log m$	0.028 ± 0.113	0.005 ± 0.050	0.002 ± 0.027	0.002 ± 0.027
FlagCounter, $\Delta = \lambda^{-1}\gamma\log\log m$	0.002 ± 0.022	0.002 ± 0.016	0.001 ± 0.001	0.001 ± 0.001
Global Counter, $\Delta = 4\lambda^{-1}$	0.401 ± 0.200	0.430 ± 0.173	0.498 ± 0.032	0.497 ± 0.042
Hub-FlagCounter, $\Delta = \lambda^{-1}$	0.040 ± 0.061	0.034 ± 0.087	0.096 ± 0.114	0.117 ± 0.125

greater than 0.5, due to the early detection requirement and that the performance was evaluated on a twice longer operating time than the training session.

We also check the performance with small inspected sets (in Fig. 11), of size scaling as $\log n$, and we see again that for such small size, the error rates are high and do not seem to tend to zero.

11 Conclusion

The central message of this work is that if an epidemic (malware, or otherwise) spreads, even if it does so with extreme stealth so that locally it is statistically undetectable, it leaves a signature of its presence encoded in the very network it uses to spread. Put differently, the network that spreads the epidemic also correlates extremely weak signals across the network, and if this pattern can be identified, the weak signal can be revealed. Our results cover various different settings and topologies, including grids and random graphs, and provide efficient, light-weight algorithms that can detect epidemics with error rates quickly tending to zero. Questions of robustness, network homogeneity, and adversarially adapted malware or other network agents are natural directions to consider in this extreme low-information regime.

Acknowledgments

This research was supported in part by the European Union through the CONGAS project in the 7th Framework Programme, US National Science Foundation Grant CNS-1320175, EECS-1056028, DTRA grant HDTRA 1-08-0029, ARO Grants W911NF-16-1-0377 and W911NF-14-1-0387, and the US DOT supported D-STOP Tier 1 UTC. An early version of this work was presented (as an invited talk) at the 52nd Annual Allerton Conference on Communication, Control, and Computing, October 2014 and the 2016 International Conference on the Science of Electrical Engineering, Eilat, Israel.

References

[1] H. Kesten, "On the speed of convergence in first-passage percolation," *Ann. Appl. Probability*, vol. 3, no. 2, pp. 296–338, Nov. 1993.

[2] A. Ganesh, L. Massoulie, and D. Towsley, "The effect of network topology on the spread of epidemics," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies.*, 2005, pp. 1455–1466.

[3] N. D. Blair-Stahn, "First passage percolation and competition models," p. 24, May 2010. [Online]. Available: http://arxiv.org/abs/1005.0649

[4] S. Bhamidi, R. van der Hofstad, and G. Hooghiemstra, "First passage percolation on the Erdos-Renyi random graph," May 2010. [Online]. Available: http://arxiv.org/abs/1005.4104

[5] S. Bhamidi, R. van der Hofstad, and G. Hooghiemstra, "Universality for first passage percolation on sparse random graphs," Oct. 2012. [Online]. Available: http://arxiv.org/abs/1210.6839

[6] A. Gopalan, S. Banerjee, A. K. Das, and S. Shakkottai, "Random mobility and the spread of infection," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 999–1007.

[7] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2007, Art. no. 420.

[8] G. Streftaris and G. Gibson, "Statistical inference for stochastic epidemic models," *Scandinavian J. Statistics*, vol. 32, no. 2, pp. 265–280, Jun. 2005.

[9] N. Demiris and P. O'neill, "Bayesian inference for epidemics with two levels of mixing," *Scandinavian J. Statist.*, vol. 32, pp. 265–280, 2005.

[10] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," May 2011. [Online]. Available: http://arxiv.org/abs/1105.0697

[11] N. Karamchandani and M. Franceschetti, "Rumor source detection under probabilistic sampling," in *Proc. IEEE Int. Symp. Inf. Theory*, Jul. 2013, pp. 2184–2188.

[12] D. Shah and T. Zaman, "Detecting sources of computer viruses in networks: Theory and experiment," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 38, pp. 203–214, 2010.

[13] Z. Wang, W. Dong, W. Zhang, and C. W. Tan, "Rumor source detection with multiple observations," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 1–13, Jun. 2014.

[14] W. Luo, W. P. Tay, and M. Leng, "How to identify an infection source with limited observations," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 586–597, Aug. 2014.

[15] B. A. Prakash, J. Vreeken, and C. Faloutsos, "Spotting culprits in epidemics: How many and which ones?" in *Proc. IEEE 12th Int. Conf. Data Mining*, Dec. 2012, pp. 11–20.

[16] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Network forensics," in *Proc. 12th ACM SIGMETRICS/PERFORMANCE Joint Int. Conf. Meas. Model. Comput. Syst.*, Jun. 2012, p. 223.

[17] C. Milling, C. Caramanis, S. Mannor, and S. Shakkottai, "Detecting epidemics using highly noisy data," in *Proc. 14th ACM Int. Symp. Mobile ad hoc Netw. Comput.*, Jul. 2013, Art. no. 177.

[18] E. Arias-Castro, E. J. Candès, and A. Durand, "Detection of an anomalous cluster in a network," *Ann. Statist.*, vol. 39, no. 1, pp. 278–304, Feb. 2011.

[19] E. A. Meirom, C. Milling, C. Caramanis, S. Mannor, A. Orda, and S. Shakkottai, "Localized epidemic detection in networks with overwhelming noise," Feb. 2014. [Online]. Available: http://arxiv.org/abs/1402.1263

[20] F. Chung and L. Lu, *Complex Graphs and Networks (Cbms Regional Conference Series in Mathematics)*. Boston, MA, USA: Amer. Math. Soc., 2006.

[21] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution," *ACM Trans. Knowl. Discovery Data*, vol. 1, no. 1, Mar. 2007, Art. no. 2.

[22] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *Proc. Eur. Conf. Mach. Learn.*, 2004, pp. 217–226.

[23] R. Durrett, *Random Graph Dynamics (Cambridge Series in Statistical and Probabilistic Mathematics)*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

Eli A. Meirom received the BSc (summa cum laude) degree in mathematics and physics in 2002, and the MSc degree in Physics (cum laude) in 2007, both from the Technion–Israel Institute of Technology, Haifa, Israel. He is currently pursuing working toward the PhD degree in electrical engineering in the Technion–Israel Institute of Technology. Previously, he was a research scientist in St. Jude Medical and a research intern in IBM Research. His research interests include complex networks, multi agent decision making, and machine learning. He is a recipient of the applied materials and Mel Berlin fellowships. He is a member of the IEEE.

Constantine Caramanis (SM) received the PhD degree in electrical engineering and computer science from the Massachusetts Institute of Technology, in 2006. Since then, he has been on the faculty in the Department of Electrical and Computer Engineering, The University of Texas at Austin. He received the NSF CAREER award in 2011. His current research interests include robust and large scale optimization and control, machine learning and high-dimensional statistics, with applications to large scale networks, and computer aided design. He is a senior member of the IEEE.

Shie Mannor (S'00-M'03-SM-09') received the BSc degree in electrical engineering, the BA degree in mathematics, and the PhD degree in electrical engineering from the Technion-Israel Institute of Technology, Haifa, Israel, in 1996, 1996, and 2002, respectively. From 2002 to 2004, he was a Fulbright scholar and a postdoctoral associate with M.I.T. He was in the Department of Electrical and Computer Engineering, McGill University from 2004 to 2010 where he was the Canada Research chair in machine learning. He has been with the faculty of electrical engineering at the Technion since 2008 where he is currently a professor. His research interests include machine learning and pattern recognition, planning and control, multi-agent systems, and communications. He is a senior member of the IEEE.

Ariel Orda (S'84–M'92–SM'97–F'06) received the BSc (summa cum laude), MSc, and DSc degrees from the Technion–Israel Institute of Technology, Haifa, Israel, in 1983, 1985, and 1991, respectively, all in electrical engineering. Since 1994, he has been in the Department of Electrical Engineering, Technion-Israel Institute of Technology, where he is currently the Herman and Gertrude Gross Professor of Communications. Since 2014, he has been the dean of the Department of Electrical Engineering with the Technion–Israel Institute of Technology. His research interests include network routing, the application of game theory to computer networking, survivability, QoS provisioning, wireless networks, and network pricing. He received several awards for research, teaching, and service, and most recently, the 2009 Henry Taub Prize for Academic Excellence and the 2011 TCCC Outstanding Service Award. He served as a program co-chair of the IEEE INFOCOM 2002, the program chair of WiOpt 2010, and the general chair of NETGCOOP 2012. He was an editor of the *IEEE/ACM Transactions on Networking*. He is fellow of the IEEE.

Sanjay Shakkottai (M'02–SM'11–F'14) received the PhD degree from the ECE Department, University of Illinois at Urbana-Champaign, in 2002. He is with The University of Texas at Austin, where he is currently a professor in the Department of Electrical and Computer Engineering. He received the NSF CAREER award in 2004, and elected as an IEEE fellow in 2014. His current research interests include network architectures, algorithms and performance analysis for wireless networks, and learning and inference over social networks. He is a fellow of the IEEE.