On the Robustness of Diffusion in a Network under Node Attacks

Alvis Logins, Yuchen Li, and Panagiotis Karras

Abstract—How can we assess a network's ability to maintain its functionality under attacks? *Network robustness* has been studied extensively in the case of deterministic networks. However, applications such as online information diffusion and the behavior of networked public raise a question of robustness in *probabilistic* networks. We propose three novel robustness measures for networks hosting a diffusion under the Independent Cascade or Linear Threshold model, susceptible to attacks by an adversarial attacker who disables nodes. The outcome of such a process depends on the selection of its initiators, or seeds, by the *seeder*, as well as on two factors outside the seeder's discretion: the attacker's strategy and the probabilistic diffusion outcome. We consider three levels of seeder awareness regarding these two *uncontrolled* factors, and evaluate the network's viability aggregated over all possible extents of an attack. We introduce novel algorithms from building blocks found in previous works to evaluate the proposed measures. A thorough experimental study with synthetic and real, scale-free and homogeneous networks establishes that these algorithms are effective and efficient, while the proposed measures highlight differences among networks in terms of robustness and the surprise they furnish when attacked. Last, we devise a new measure of diffusion entropy, and devise ways to enhance the robustness of probabilistic networks.

Index Terms—Graphs and networks, Stochastic processes, Reliability and robustness

1 Introduction

Networks are ubiquitous in the modelling of infrastructures [1], [2], social interactions [3], [4], [5], physical and life-science phenomena [6], [7]. Yet such networks are subjects to failures or *attacks* [8], whereby some of their elements may be disabled or removed. The impact of such structural alterations on network performance depends on the desirable features of a network's operation in a particular application.

Network robustness is the ability of a network to retain critical features of its topology and functionality in the face of uncertainty regarding its components. Quantitative measures of robustness express the degree to which a network retains such features despite attacks [9].

Deterministic robustness. Some measures of network robustness gauge the change of a *deterministic* graph property after *random* failures. The measured property may be the network's diameter, average path length [10], or inverse shortest path length [11]. A critical measure is the size of the largest connected component (LCC) [10], [9], uses in domains from power grids [6] to biological systems [7]. Such analysis is grounded on *percolation theory* [12], which studies problems such as the dependence of a network's largest cluster on node failure probability and predicts phase transitions, i.e., rapid and cardinal changes of network affordance when a parameter reaches a critical value.

Stochastic robustness. In applications such as information diffusion and epidemiology, there is uncertainty regarding the *connections* in the network, i.e., the network is *stochastic*. In this paper, we study the operation of such a stochastic network under *targeted* node failures (or, equivalently, attacks on nodes), expressed as the expected number of activated (or infected) nodes under some parameters of a diffusion process. We refer to this type of robustness as *probabilistic* network robustness. Despite the extensive study of deterministic network robustness [13], its probabilistic counterpart has been only scantily studied. There are studies on how to engineer some kind of robust diffusion in an uncertain or adversarial environment [14], [15], but an investigation on how robustness is to be measured in such environments is missing.

In this paper, we study network robustness expressed as the capacity to carry out a successful diffusion under node attacks. We introduce three robustness measures built around two sources of uncertainty: attacks on nodes and probabilistic diffusion outcomes on edges.

In more detail, our main contributions are the following:

1) We define the concept of *network robustness under adversarial node attacks* (Section 2.2), which represents the capability of a network to host a diffusion process starting from some *seeds* under the independent cascade and linear threshold models;
2) We introduce the notion of *seeder awareness* and propose algorithms to measure network robustness under different awareness levels (Section 3);
3) We utilize, and enhance the runtime of, recent solutions to the problem of *Robust Influence Maximiza-*

tion [14] so as to compute the effects of node attacks (Section 3.2.3);

4) We enhance the DAGGER [16] *reachability index* and use it in the case where the seeder is aware of network outcomes (Section 4);
5) We measure the robustness of scale-free and homogeneous, synthetic and real-world networks, investigate interrelationships among the robustness measures we propose, and suggest ways to enhance probabilistic network robustness (Section 5).

This paper comprises an extended version of our conference paper on the same subject [17].

2 Background

Processes in large networks, such as electricity flow, package routing, and protein delivery, are vulnerable to attacks. Such events may cause electricity blackouts [18] or epidemics [19]. There is a need to gauge the extent of such effects and design protection mechanisms. *Network robustness* assesses the impact of a network alteration on such processes.

2.1 Deterministic Robustness of Integrity

Network robustness reflects a network's ability to maintain its connectivity under attacks [20]. The connectivity of an *undirected* network is measured by the expected size of its largest connected component (LCC) after an attack [20], also defined on probabilistic undirected networks [21].

The study of robustness under *random failures* is inspired by results of *percolation theory*, which describes the physics of *phase transitions* [12], [22] in systems such as magnets, fluids [12], and proteins [23]. A phase transition occurs when certain network parameters pass a *critical threshold* [22]. Percolation theory examines the expected maximum size [22] of a cluster made of particles in the same *active* state, as with the spontaneous magnetization of the *Ising magnet* [12], [24]. An example of the critical threshold is the Molloy-Reed criterion [22], by which a giant component appears in a general graph if $\frac{\langle k^2 \rangle}{\langle k \rangle} > 2$, where $\langle \cdot \rangle$ denotes expectation and k denotes a node's degree. The Molloy-Reed criterion shows that scale-free networks are extremely robust to *random node* failures, while being vulnerable to *targeted* node attacks [25]; increasing their robustness against targeted attacks conflicts with maintaining their robustness against random failures [22]. Some robustness measures take into consideration both random and target failures, yet do not provide a method to achieve high robustness in those terms [26]. Schneider et al. [6] propose a local-search heuristic that rewires edges so as to increase an *inclusive* measure of robustness against targeted attacks, while maintaining node degrees. Such an inclusive measure considers all cases of a malicious attack or failure, including those in which the network does not collapse but suffers a big damage [6]. The heuristic in [6] leads to an onion-like graph structure, with nodes of similar degree tending to be connected to each other. A LCC-based measure of robustness under *random edge* failures is the *reliability polynomial* [27]:

$$\mathrm{Rel}(G) = \sum_{i=1}^{m} F_i (1-p)^i p^{m-i}$$

where m is the number of edges, F_i is the number of sets of i edges whose removal leaves G connected, and p is an independent probability that an each edge is present. Denoting the probability that an edge is absent as $x = 1 - p$, we derive an equivalent definition [27], $\mathrm{Rel}(G) = \sum_{i=1}^{m} R_{m-i} x^i (1-x)^{m-i}$, where R_{m-i} is the number of connected subgraphs made of $m-i$ existing edges. The closely related problem of securing connectivity between two predefined node sets under edge failures is known as *network reliability* problem [28].

To the best of our knowledge, the only previous work studying networks where *both* nodes and edges are subjects to failures is the study of percolation in infinite networks by Chayes and Schonmann [29], which derives inequalities that bind the critical values of edge failures, node failures, and the network's maximal degree. However, there is no notion of an attack in [29]. We study the robustness of stochastic diffusion processes under node attacks; this notion of robustness generalizes the robustness of deterministic networks under targeted node attacks and random edge failures.

2.2 Stochastic Robustness of Diffusion

Network robustness also refers to a network's capacity to host a diffusion process despite the exclusion of some network elements [30], [31], [14], [22]. The mathematical modelling of diffusion is independent of semantics: it may be a diffusion of *information*, a *cascading failure*, or a viral *infection epidemic* [18]. Similarly, a node attack is mathematically equivalent to a node immunization or failure. As the effect of node attacks is evaluated by a stochastic process, we formulate the corresponding concept of *stochastic robustness*.

A diffusion may be *epidemic*, *threshold*, or *cascading* [32]. There are two popular epidemic models [33]: By the **SIS** model, nodes are either ***s**usceptible* or ***i**nfected*; By the **SIR** model, it may ***r**ecover* and becomes immune. The *expected* size of an SIR epidemic starting at u is equal to the expected size of the connected component that contains u [34]. Epidemic models typically consider a homogeneous *infection rate*, yet two models study *information diffusion* with heterogeneous rates [35]: the *Independent Cascade* (IC) model (a special case of SIR [36]) and the *Linear Threshold* (LT) model [37], [3]. By these models, the *Influence Maximization* (IM) problem [37] seeks a set of initially active nodes (*seeds*) that maximizes the expected number of activated nodes.

2.3 Robustness under the IC and LT models

The IC and LT models are used to study word-of-mouth effects in social networks [3]. A diffusion proceeds in discrete time steps. At time $t = 0$, a set of *seed* nodes $S \in V$ are activated. By the IC model, any node v activated at time t tries to activate its out-neighbours at time $t+1$, and succeeds with an independent probability $p_e = p_{uv}$ for each neighbor u. In case of success, the edge e is active. By the LT model, each node v picks a random threshold $\nu \in [0, 1]$, and is activated only if $\sum_{u \in \text{in-neigh}(v)} p_{uv} > \nu$; in such a case, only *one* of the incoming edges is considered active [37], selected randomly according to the assigned edge probabilities. This cascading process ends when there are no more trials for activation. The set of active nodes and edges at the end of this process forms a deterministic

live-edge graph g [37], conditioned on the outcomes of the random choices. The *spread*, or expected number of activated nodes, is the expected number of nodes reachable from S.

Problems related to our work are those of *sensitivity to edge perturbations* [38] and *robust influence maximization* (RIM) under edge perturbation [15] or any adversarial source of uncertainty [14]. Given a finite set of adversarial strategies Θ, the objective in [14] is:

$$\max_{S,|S|\leq k} \min_{\theta\in\Theta} \frac{\sigma_\theta(S)}{\sigma_\theta(S^*_\theta)} \tag{1}$$

where $\sigma_\theta(S)$ is the spread achieved by seed set S under strategy θ, S^*_θ is the optimal seed set for θ, and k is a budget constraint; the normalization by $\sigma_\theta(S^*_\theta)$ measures the fraction over optimal influence; an absolute measure is used with continuous θ in [39].

The Saturate Greedy (SatGreedy) algorithm [14] solves the RIM problem by targeting the cumulative effect of all strategies, which is a submodular objective. This algorithm provides a bi-criteria approximation guarantee: violating the budget constraint k by an $O(k \ln |\Theta|)$ factor leads to an $(1 - \frac{1}{e})$ approximation of the optimal solution. We adopt the RIM objective as a component in one of our measures.

3 Measures of Diffusion Robustness

We propose three robustness measures, anchored on the awareness of a *seeder*, who selects seed nodes, on node attacks and diffusion outcomes. Table 1 lists our notations.

Stochastic graph with nodes V and edges E	$G = (V, E)$
Number of nodes and edges of G	n, m
Deterministic graph sampled from G	$g \sim G$
Edge probability parameter	W
Set of attack strategies removing ℓ nodes	$\Theta(\ell) = \{\theta^i(\ell)\}$
Degree of a node v	$d(v \in V)$
Number of blocked (removed) nodes	ℓ
Reachability indicator function	$I(v, S)$
ℓ-sampling parameter	α
EMR-RNI	D
Expected number of activated nodes	σ
Seed set S and size of seed set k	$S, k = \|S\|$
Number of active nodes	$\mathcal{I}$
Fraction of active nodes	$\nu = \mathcal{I}/n$
Assortativity coefficient [40]	r

TABLE 1: Notations

3.1 Attack Strategies

We measure robustness against an attacker who disables nodes. A consideration of all possible attack strategies amounts to the NP-hard problem of *node immunization* [41], [31], [36], [42]. We rather demarcate a strategic set of attack strategies on a directed stochastic network G, $\Theta_G = \{\theta^i_G\}$ [14]; $\theta^i_G(\ell)$ is a set of ℓ nodes in G chosen by strategy θ^i_G; g_θ denotes the graph obtained by removing nodes from a deterministic instance g of G according to $\theta^i_G(\ell)$. We opt for strategies that are also node ranking functions, i.e., each strategy defines a node order. We select six strategies that represent each type and cluster in [43], plus a spectral-based baseline, NetShield [36], [31], [44], [45]:

1) *Degree* picks nodes with the largest degree;
2) *Random* picks seed nodes uniformly at random;
3) *Acquaintance* [46] picks a random node's neighbor;
4) *PageRank* ranks nodes by PageRank values [47];
5) *Katz* centrality [48] equals $x_i = \alpha \sum_j \mathbf{A}_{ij} x_j + \beta$, where $\alpha = 0.1$, $\beta = 1$, and $\mathbf{A}$ the network's adjacency matrix.
6) *Betweenness* centrality is the sum of the fraction of all-pairs shortest paths that pass through a node.
7) *NetShield* greedily selects a set of nodes S, aiming to maximize its *Shield value*:

$$Sv(S) = \sum_{i\in S} 2\lambda \mathbf{u}(i)^2 - \sum_{i,j\in S} \mathbf{A}_{ij}\mathbf{u}(i)\mathbf{u}(j)$$

where λ and $\mathbf{u}$ are the largest eigenvalue and the corresponding eigenvector of the adjacency matrix $\mathbf{A}$ containing edge probabilities; λ indicates the effectiveness of a stochastic spread in the network. The algorithm works on undirected networks; we transform any network to undirected by ignoring directions and removing duplicates.

3.2 Awareness-based Robustness Measures

We distinguish three levels of *seeder awareness* regarding attacks and diffusion events and define one robustness notion for each seeder awareness level.

3.2.1 EMR

Assume an *omniscient* seeder with access to an oracle that predicts the outcome g of a diffusion on G and of an attack on g that produces g_θ. As discussed in Section 2.1, the robustness of a deterministic undirected network G can be expressed in terms of the largest connected component (LCC) [20]. When G is a *directed* network, the LCC-equivalent substructure is either of the largest strongly or weakly connected components [49]. Still, none of these LCC generalization expresses the maximum number of nodes a seeder can reach. We denote the number of nodes that a seeder can reach in an immunized live-edge instance of a directed network, g_θ, with a diffusion from a seed set S of size k, as $\sum_{v\in g_\theta} I(v, S)$; $I(v, S)$ is a binary function indicating whether there exists a path from S to node v. Maximizing the sum amounts to finding a maximum forest with at most k roots. Let *Expected Maximum Reach* (EMR) be the expected number of nodes an omniscient seeder reaches in G under the worst $\theta \in \Theta(\ell)$:

$$EMR_G(\ell) = \min_{\theta\in\Theta(\ell)} \mathbb{E}_{g_\theta\sim G}\left[\max_{S:|S|\leq k} \sum_{v\in g_\theta} I(v, S)\right] \tag{2}$$

Our first robustness measure aggregates $EMR_G(\ell)$ for all sizes ℓ of an attack, normalized by network size; we call it *sum of expected maximum reach* or *SEMR*:

$$SEMR_G = \frac{1}{n}\sum_{\ell=1}^{n} EMR_G(\ell) \tag{3}$$

We introduce a novel algorithm for SEMR computation in Section 4 and study its efficiency in Section 5.6.

3.2.2 RNI

We now consider an *informed* seeder lacking knowledge of diffusion outcomes, but having access to an oracle that predicts node attacks. The *maximum* number of nodes such a seeder can expect to reach in G under the worst-case attack strategy θ is the maximum, over all cases of S, of

the *expected* size, over all instances $g_\theta \sim G$, of the number of nodes $v \in g_\theta$ to which a path exists from S. A strategy $\theta \in \Theta(\ell)$ that *minimizes* this quantity yields the *Robust Network Immunization* (RNI) measure:

$$RNI_G(\ell) = \min_{\theta \in \Theta(\ell)} \max_{S:|S|\leq k} \mathbb{E}_{g_\theta \sim G} \left[\sum_{v \in g_\theta} I(v, S) \right] \quad (4)$$

Our second robustness measure aggregates $RNI_G(\ell)$ over all ℓ, normalized by network size. We call it $SRNI$:

$$SRNI_G = \frac{1}{n} \sum_{\ell=1}^{n} RNI_G(\ell) \quad (5)$$

The computation of SRNI requires solving an influence maximization (IM) problem on a graph with $\theta(\ell)$ nodes removed for each attack strategy $\theta \in \Theta$ and each value of ℓ. We do so while building sampled networks g_θ incrementally, using the Dynamic IM algorithm (DIM) [50].

3.2.3 RIM

Last, we consider an *agnostic* seeder who has information neither on diffusion outcomes, nor on node attacks. The best such a seeder can do is to try to solve a problem of robust influence maximization [14]. The *worst-case* number of nodes such a seeder can expect to reach in a stochastic network G with seed set S is the minimum, over all attack strategies $\theta \in \Theta(\ell)$, of the *expected* size, over all instances $g_\theta \sim G$, of the number of nodes $v \in g_\theta$ to which a path exists from S. The seeder should opt for a seed set S that *maximizes* this worst-case quantity, yielding the *Robust Influence Maximization* (RIM) measure:

$$RIM_G(\ell) = \max_{S:|S|\leq k} \min_{\theta \in \Theta(\ell)} E_{g \sim G} \left[\sum_{v \in g_\theta} I(v, S) \right] \quad (6)$$

While inspired from Equation 1, RIM is based on node removals rather than edge perturbation, and is *not* normalized by the optimal spread for a given g_θ, gauging robustness in the absolute sense. Our third measure, SRIM, aggregates $RIM_G(\ell)$ for all ℓ, normalized by network size:

$$SRIM_G = \frac{1}{n} \sum_{\ell=1}^{n} RIM_G(\ell) \quad (7)$$

To calculate SRIM, we apply *SatGreedy* [14] with the objective in Equation 1 modified to account for node removals and normalizing spread by network size $|V|$ rather than by the optimal spread under strategy θ:

$$\max_S \rho'(S) = \max_S \min_\theta \frac{\sigma_\theta(S)}{|V|}$$

We enhance the runtime of SatGreedy using the same *dynamic* approach as for SRNI [50]. We also consider the baselines proposed in [14]: *SingleGreedy* selects k seeds sequentially, maximizing the objective in each step; *AllGreedy* finds the best seed set for each adversary, and selects the one of these that maximizes the objective.

3.2.4 Summary

Our three measures form a sequence, tuning the seeder's awareness regarding the sampling of g and the application of an immunization strategy θ to g by means of two choices: the order of $\max$ and $\min$ determines whether the seeder is aware of the immunization strategy; the positioning of expectation $\mathbb{E}$ indicates whether the seeder is aware of the sampling of g. Table 2 depicts the relationships among the three measures with respect to these key properties. ne case in the table is not covered by the hitherto described measures, namely the case corresponding to an unaware seeder, yet with a strategy chosen *posterior* to the sampling of g. We call this measure EMinR (Expected *Minimum* Reach):

$$EMinR_G(\ell) = \max_{S:|S|\leq k} \mathbb{E}_{g \sim G} \left[\min_{\theta \in \Theta(\ell)} \sum_{v \in g_\theta} I(v, S) \right] \quad (8)$$

	strategy-aware seeder	agnostic seeder
sampling first	RNI (Eq. 4)	RIM (Eq. 6)
seeds first	EMR (Eq. 2)	*EMinR* (Eq. 8)

TABLE 2: Relationships between robustness measures.

4 Computation of SEMR

The algorithms for SRNI and SRIM computation discussed in Sections 3.2.2 and 3.2.3, respectively, were based on existing solutions. In this section, we introduce a novel algorithm for SEMR computation.

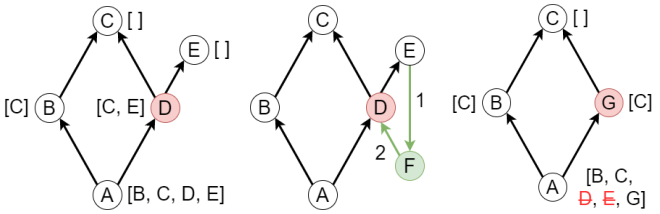
To compute SEMR for a single seed, we need to find the expected maximum tree sizes over a sequence of network samples $g \sim G$ under each attack strategy $\theta \in \Theta$. We consider attack strategies θ such that the set of blocked nodes under strategy θ for $\ell + 1$ is a superset of that for ℓ, i.e., $\theta(\ell) \subset \theta(\ell + 1)$. To obtain a sequence of attack sets $\theta_g(\ell)$ for different values of ℓ on g, it suffices to sequentially remove nodes from g. Equivalently, since we are interested in all values of ℓ, we sequentially add nodes, in reverse. We compute maximum tree sizes over several random samples g from G, with edges pre-sampled and nodes incrementally added according to each strategy $\theta \in \Theta$:

$$\mathbb{E}_{g \sim G}[\{T(g \setminus \theta_g(\ell))\}_{\ell=1}^{n}] \approx \frac{1}{M} \sum_{j=1}^{M} \sum_{\ell=n-1}^{0} T(g_j \setminus \theta_{g_j}(\ell)),\ \forall \theta$$

where M is total number of graph samples and the decreasing ℓ indicates that nodes are incrementally *added* to g_j.

To compute the maximum tree size $T(\cdot)$ efficiently, we build upon the DAGGER algorithm [16], employing a dynamic reachability index that returns nodes reachable from any node and also supports node insertions. Given g, the index maintains a directed acyclic graph (DAG), where each node represents a strongly connected component (SCC) in g, called *graph condensation*. A node's insertion implies the insertion of its pre-sampled incident edges. Assume a new edge $e = (u, v)$ is inserted. Let s and t be the SCCs u and v belong to, respectively. DAGGER checks whether there is a path from t to s, using its reachability index. If there is a path, then the insertion of e merges at least two existing SCCs. To find all SCCs to be merged, DAGGER recursively traverses the path from t to s, while pruning descendants of t that do not have a path to s.

We extend DAGGER with a query that returns a network's maximum tree size. Let $g' = (V', E')$ be the DAG that corresponds to g. For each node $v' \in V'$, we maintain a label $v'.r$ as the set of nodes $u' \in V'$ reachable from v': $v'.r = \{u' \in V' | \exists \, \text{path} \, v' \rightarrow u'\} \cup \{v'\}$. After inserting a new node w to g, we obtain the corresponding $w' \in g'$, such that w' represents the SCC w belongs to, calculate $w'.r$ based on the out-neighbours of w' in g', and propagate $w'.r$ to all ascendant nodes of w'. Since a new node w may result in the removal of a SCC, we propagate a set of ids of the removed SCCs to ascendant nodes of w' as well. Finally, we maintain a heap of root nodes, valued by the size of their labels. Once an update from w' reaches a node u' with zero in-degree, we add u' to the heap, or update the heap's value if it already contains that w'.



(a) A condensation (b) Adding node (c) Updating labels

Fig. 1: Maintaining the DAG reachability index.

Figure 1a shows an example graph condensation. Let node D represent an SCC. Node A can reach nodes B, C, D, E, node B can reach C, and D can reach C, E. Besides the reachability labels, we assume a DAGGER index built on the graph. Assume a new node E is added to the graph (Figure 1b). The algorithm first adds an edge (E, F), then (F, D). After (F, D) is added, the DAGGER index determines that there exists a path from D to F through E, so F, D and E have to be merged into a new SCC, represented by node G (Figure 1c). Thereafter, G constructs a new reachability label based on the former label of D, and propagates the information about the added and the removed nodes to its ancestor node A.

Algorithm 1 illustrates how we compute the SEMR measure incrementally, by calling SEMR(), which calls INSERT(w, H). H is a heap organizing the nodes of DAG, ordered by the sum of reachable SCC sizes. When performing a node insertion, we first perform the insertion, as explained in the above, by the DAGGER.INSERT() query, then collect the ids of the new node's SCC (Line 3) and all invalidated SCCs (Line 4). Lines 5-6 calculate the reachability of node w', which corresponds to the new node w in the DAG. Lines 8-11 traverse all nodes reachable from w' in the reverse DAG $(g')^T$ by breadth-first search. We update the reachability label of each reverse reachable DAG node u' according to the set of removed SCC's R, and the set of DAG nodes reachable from w'. Last, if during a traversal we reach a root node (Lines 10-11), we insert or update the corresponding reachability value in the heap H.

The SEMR() function in Algorithm 1 returns SEMR for a single seed. For k seeds, in Line 19 we greedily pick k nodes from the heap H, prioritized by marginal gain in terms of reachable nodes in g. We apply the *lazygreedy* optimization [51] while collecting top root nodes. Assume i nodes are already picked, and Ω is the set of picked nodes. Once the $(i+1)$-st node v is picked from H, the algorithm checks whether there exists a node u reachable from v and $w \in \Omega$, such that u is reachable from w. If so, the algorithm updates the label of v, enheaps v again, and picks the next node from H. The lazygreedy optimization is applicable, since (i) adding a new root to a set of selected roots does not change the set of candidate roots, and (ii) maximizing reachable nodes is a submodular objective.

The performance of SEMR computation depends on set union and subtraction operations (Lines 6 and 9). We implement a variant where all operations on reachability labels are performed by a bitset data structure; this measure reduces the time of set operations, but incurs an overhead in calculating the heap value, as it queries single bits for each root node. More advanced set data structures, such as Binary Decision diagrams [52], may improve efficiency further.

Algorithm 1 SEMR Computation

```
1: function INSERT(w, H)
2:     DAGGER.INSERT(w)
3:     w' ← SCC(w)    ▷ w' is a node in g', that corresponds to SCC in g and has a label r
4:     R ← a set of removed nodes from g'
5:     for all v'|(w', v') ∈ g' do
6:         w'.r ← w'.r ∪ v'.r
7:     Q ← {u'|∃ path w' ⇝ u' in (g')^T}
8:     for all u' ∈ Q do
9:         u'.r ← u'.r ∪ w'.r \ R
10:        if ∄v'|(v', u') ∈ E' then
11:            H.insert(< u', |{v ∈ g|SCC(v) ∈ u'.r}| >)
12: function SEMR
13:    for all θ ∈ Θ do
14:        s_θ ← empty list
15:        Initialize DAGGER with empty graph
16:        H ← a descending heap of < key, value >
17:        for all v ∈ θ.reverse() do
18:            INSERT(w, H)
19:            v', s ← H.top()                ▷ Apply lazygreedy for k > 1
20:            s_θ[ℓ] = s
21:    s_min ← empty list
22:    s_min[ℓ] ← min_θ s_θ[ℓ] ∀ℓ
23:    return ∑ s_min
```

4.1 Complexity Analysis

An iteration of SEMR computation involves DAG maintenance, reachability label propagation, and greedy root selection. The complexity of an edge insertion that does not create a new SCC is constant; in case a new SCC is created, the worst-case complexity is $O(m')$, where m' is the running number edges in the DAG [16]. Reachability label propagation takes $O(m'^2)$, as it updates labels for all ancestors of a new node in the DAG, and each update requires a set union operation on sets of size at most m'. For greedy root selection, it traverses all roots of the DAG and calculates the total size of all SCCs reachable from each root. As we maintain SCC sizes and a list of roots while building the DAG, single root selection takes $O(m')$. We select k roots, resulting in $O(k \cdot m')$, while the lazygreedy optimization makes it significantly faster. With the bitset data structure, there is an additional step to calculate the number of nodes reachable from roots. Each DAG node maintains the number of corresponding SCCs, and a set of reachable DAG nodes. To get the number of reachable nodes, we traverse all bits, incurring an additional m' factor.

To calculate the minimum over all considered immunization strategies, we evaluate $SEMR$ per each strategy independently, hence a $|\Theta|$ factor. Summing up, the calculation of $SEMR$ takes $O(|\Theta| \cdot nm'(m'+k))$. Using a sampling factor

$\alpha > 1$, we perform greedy selection only for sampled nodes. With $q = \lfloor \log_\alpha(1 + n(\alpha - 1)) \rfloor$ samples, the complexity becomes $O(|\Theta| \cdot (nm'^2 + qk))$.

4.2 Approximation Guarantees

Algorithm 1 estimates the maximum forest size in a single MC iteration by selecting root nodes greedily. This method is a special case of Influence Maximization [37] on a network with all edge probabilities equal to 1, hence achieves a $(1 - \frac{1}{e})$ approximation guarantee. Since simulation instances are i.i.d., we use the Chernoff bound to prove the approximation ratio of our randomized greedy algorithm. The failure probability of θ samples is $Pr[M_g - (1 - \frac{1}{e} - \epsilon)\, M^* < 0] = O(\exp(-\epsilon\theta\,(1 - \frac{1}{e})\, M^*))$, where M_g is the average of maximum forest sizes returned by our algorithm, M^* is the true average maximum forest size, and ϵ is a parameter that trades off between error and number of samples generated. Thus, M_g converges to a $(1 - \frac{1}{e} - \epsilon)$-approximate solution with probability of failure exponentially decreasing with respect to sample size θ.

RNI inherits the guarantees of DIM [50]; given a sufficient number of samples, it returns a seed set S such that $\sigma(S) \geq (1 - \frac{1}{e} - \epsilon)\, \sigma(S^*)$ with probability at least $1 - \frac{1}{n}$, where σ is expected spread, and S^* is an optimal seed set.

The SatGreedy algorithm returns a seed set S' that approximates the original objective $\rho(S)$ with guarantee

$$\rho(S') \geq \left(1 - \frac{1}{e}\right) \cdot \rho(S^*) - \gamma$$

where k is constraint on the number of seeds and $\gamma \in (0, 1)$ is an approximation parameter. For the guarantee to hold, γ has to be related to β as $\beta = 1 + \ln|\Theta| + \ln\frac{1}{\gamma}$, where Θ is the employed set of strategies. It is worth noting that, even with large $\gamma = 0.9$ and only two strategies, $\beta = 1 + \ln 3|\Sigma|/\gamma \approx 2.89$, i.e. SatGreedy requires to increase the seed set size more than 2 times for its approximation guarantee to hold. The authors set $\gamma = 2 \cdot 10^{-3} \cdot |\Sigma|$ empirically [53], while keeping $\beta \leq 2$, therefore the approximation guarantee does not hold for the experiments presented in the paper; the solution operates as a heuristic.

5 Experimental Study

Experiments ran on a 378G RAM Intel Xeon CPU @ 3.10GHz running Ubuntu 18.04. All algorithms are implemented[1] in C++ and compiled with gcc 7.4 with -O3 optimization. We set timeout 10h per one measure computation. Runtime and timeout do not include time for the strategy set Θ computation, which is the same for all measures. We assign edge probabilities by either *Random* or *Uniform* assignment. By *Random* assignment, we pick a value for each edge uniformly from 0 to W, where W is a parameter. By *Uniform* assignment, we assign a certain W value to each edge. When working with the LT model, we additionally divide the value on each edge incident to a node u by the in-degree of u. We use the IC model unless stated otherwise.

In the following, we first describe our data (Section 5.1). Then we perform a comparative study of SatGreedy to simpler baselines proposed in [14] on the SRIM objective (Section 5.2), concluding that one of those baselines (SingleGreedy) is more suitable for calculating SRIM; and use that in subsequent experiments. Section 5.3 studies the three presented measures on various synthetic and real-world data, illustrating patterns of behaviour and the expressiveness of the new SEMR measure in comparison to other two. In Section 5.4, we propose a notion of diffusion entropy as the difference between SEMR and SRNI, and in Section 5.5 we show how the expressiveness of SEMR can be exploited to obtain stochastic networks with more robust structure. The last three sections study the efficiency and scalability of the proposed algorithms.

[1] The code is available at https://github.com/allogn/robustness.

5.1 Datasets

Synthetic Networks. We study *power-law networks*, represented by the **Barabási-Albert** (BA) model, and *homogeneous networks*, represented by the **Gaussian Random Partition** (GRP) [54]. For **BA**, we use the algorithm of Holme and Kim [55], which extends the original Barabási-Albert model, yet use the BA label as its basis. The algorithm randomly creates μ edges for each node in a graph, and for created edge with a probability p adds an edge to one of its neighbors, thus creating a triangle. **GRP** groups nodes so that group sizes follow a Gaussian distribution with expected size s and variance of size equal to s/v, where v is a shape parameter. It uses a probability value p_{in} for edges across nodes in the same group, and p_{out} otherwise.

Network	$\lvert V\rvert$ [$\cdot 10^3$]	$\lvert E\rvert$ [$\cdot 10^3$]	$d_{max}, \overline{d}$	cl	r
Blogs	1.2	19.0	467, 31	0.336	-0.2309
Minnesota	2.6	3.3	5, 2	0.024	-0.1848
VK	2.8	40.8	288, 29	0.247	-0.1711
Advogato	6.6	47.3	947, 14	0.211	-0.0951
DBLP	12.6	49.7	710, 8	0.117	-0.0540
Brightkite	56.7	212.9	1134, 8	0.117	0.0108
Gnutella	62.6	147.9	95, 5	0.007	-0.0063
Stanford	281.9	2312.5	38626, 16	0.597	-0.1220

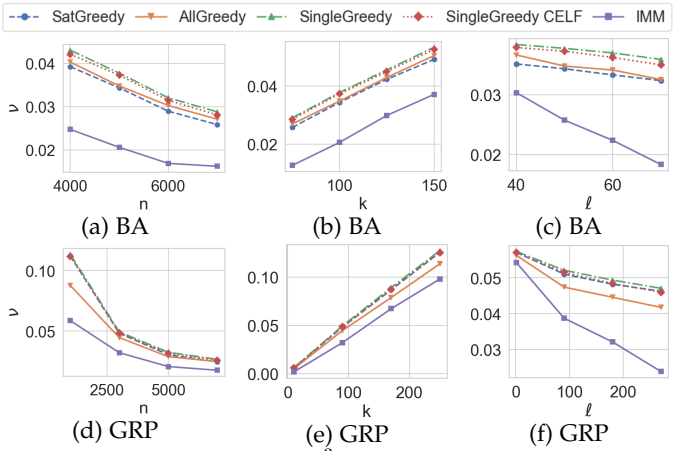
TABLE 3: Real-world datasets; $d_{max}, \overline{d}$: maximum and average degree; cl: average clustering coefficient [56]; r: assortativity coefficient [40].

Real-world networks. We use real-world datasets of various sizes and degree distributions: Blogs contains frontpage hyperlinks between blogs during the 2004 US election [57], [58]. DBLP is a citation network of scientific papers [59], [58]. Advogato is a network of trust relationships in an online community platform for free-software developers [60], [58]. Minnesota is a road network [61]. VK is a social network with influence probabilities derived from the content of posts published by users [42]. Brightkite is a location-based social network [62]. Gnutella is a snapshot of the Gnutella peer-to-peer file sharing network [63]. Stanford represents pages and hyperlinks of the Stanford University web site [64]. Table 3 lists our real-world datasets. Our experiments employ directed networks; we transform undirected networks to directed ones, replacing each undirected edge with two directed edges in opposite directions.

5.2 Choice of Algorithm for RIM Computation

As a preliminary experimental choice, we study the performance of methods for RIM calculation, including algorithms and baselines in [14]. We use the IMM algorithm for influence maximization [65] as a non-robust baseline. We include

the SingleGreedy method *with* the CELF (i.e., lazygreedy) optimization, proposed in [14], and also its variant *without* this optimization, given that, on this non-submodular problem objective, the CELF optimization affects quality.

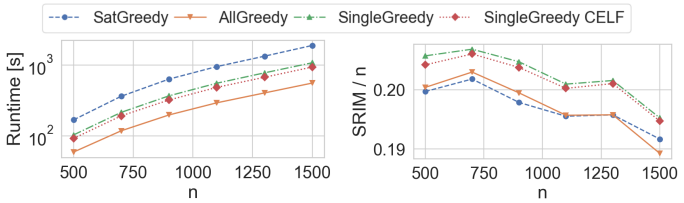


(a) BA (b) BA (c) BA

(d) GRP (e) GRP (f) GRP

Fig. 2: RIM. BA ($n=5\cdot10^3$, $\ell=50$, $k=100$, $\mu=2$), GRP ($n=3\cdot10^3$, $\ell=180$, $k=90$). $W=0.1$, SatGreedy: $\gamma=10^{-4}$.

We first compare the performance of algorithms under node attacks, i.e., in the computation of the unaggregated RIM objective, with BA and GRP data. Figure 2 depicts the fraction of active nodes ν vs. graph size n, seed set size k, and number of attacked nodes ℓ. SatGreedy outperforms other methods. However, SingleGreedy CELF achieves almost the same quality as SatGreedy. IMM has a significant disadvantage over other solutions which grows with ℓ, imprinting the significance of using robust algorithms.



(a) Runtime of SRIM baselines (b) Objective of SRIM baselines

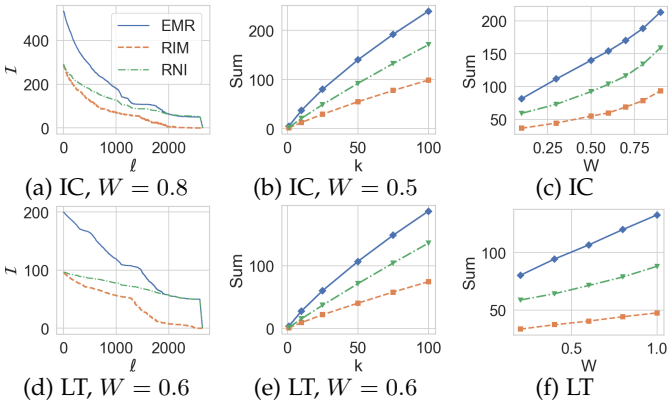
Fig. 3: SRIM on BA, $p=0.4$, $\mu=10$, $W=0.3$, $k=50$.

Now we drop the non-robust IMM algorithm out of the comparison, and study the performance of robust algorithms, with the DIM algorithm embedded, on the runtime for computing, and value of, the *aggregate* SRIM measure on the BA network. Figure 3 shows our results for $k=50$ seeds. As in Figure 2, SingleGreedy stands out in terms of objective, at the cost of higher runtime. The difference in objective is more prominent now, as we aggregate the measure over all values from 1 to ℓ. The runtime for computing Θ is negligible, reaching 4s for the largest network.

These results indicate that SingleGreedy (w/o CELF) offers the best effectiveness, but low efficiency. SingleGreedy with CELF at least matches the performance of SingleGreedy and SatGreedy, is more efficient, and does not require any accuracy parameter γ, as SatGreedy does. Ergo, we opt for SingleGreedy *with* CELF in the following.

5.3 Measure Relationships

We now study the relation between measures on small networks, and their sensitivity to the set of attack strategies, using two homogeneous networks (Minnesota and GRP) and two power-law networks (Blogs and VK).



(a) IC, $W=0.8$ (b) IC, $W=0.5$ (c) IC

(d) LT, $W=0.6$ (e) LT, $W=0.6$ (f) LT

Fig. 4: Measures on Minnesota road network. $k=50$

Figure 4a plots plain EMR, RNI, and RIM values, without aggregation, vs. ℓ on Minnesota. Values decrease gradually, revealing some irregularities of graph structure in the middle range of ℓ. EMR and RNI follow a similar pattern, while RIM differs. For instance, from $\ell=1000$ to 2000 EMR and RNI present two abrupt drops at the same value of ℓ. RIM presents several smaller irregularities. Figures 4b and 4c present the summed measures (SEMR, SRNI, and SRIM) vs. seed set size k and influence probabilities W, respectively. The difference between measures grows, especially with the size of seed set. We obtained similar results with the LT model, shown in Figures 4d, 4e, and 4f. We will see that a similar trend vs. seed set size appears in power-law networks, in Figures 7c and 8b.

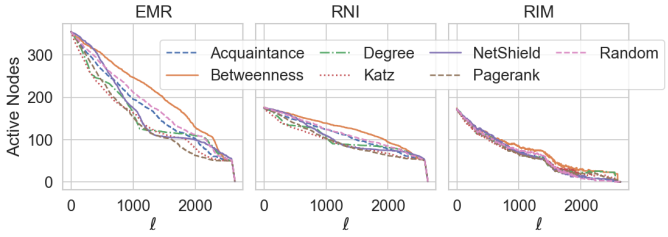
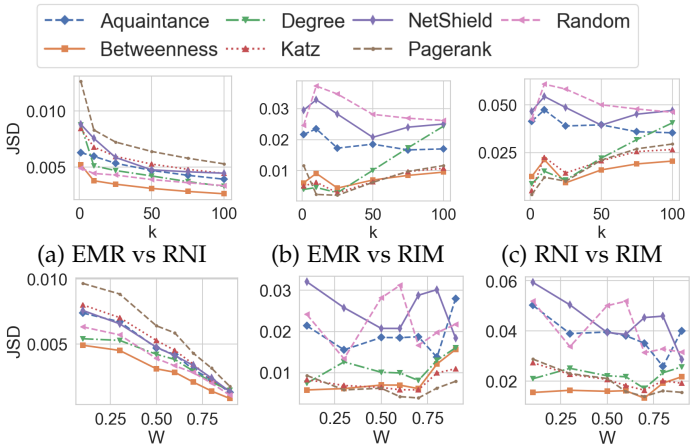


Fig. 5: Minnesota, 7 attack strategies, $W=0.5$, $k=50$, *Random*.

Figure 5 presents a decomposition of measures: instead of taking a minimum over all strategies, we plot the expected spread per strategy, with the seed set selected by each algorithm. EMR and RNI follow the same trend *also* for each strategy separately. This is conspicuous with NetShield, which shows poor performance in its immunization objective for small ℓ, but swiftly improves in the middle range; it then becomes the most effective strategy for a short ℓ range, but looses that position to PageRank. Remarkably, results for RNI presents the same outline, but scaled to a smaller values of active nodes. On the other hand, RIM exhibits a different behaviour, as all strategies mostly produce the same response to the selected seeds. This result illustrates the difference of RIM from the other two measures: RIM is based on the worst case among the complete set of strategies by nature, hence can afford to let the selected seeds perform almost equally well on any attack.

In Figure 6, we take another view on decomposition of measures: We plot the Jensen-Shannon divergence for each pair of measure distributions over ℓ, and for each strategy,

with varying k and W. For example, one point on Figure 6a shows $\mathrm{JSD}(\mathrm{EMR}(\ell)\,||\,\mathrm{RNI}(\ell))$ for a specific k. JSD values for EMR vs RNI are much smaller than for other two pairs, and smoothly converge to zero; values are larger for more effective attack strategies. On the other hand, the divergence of RIM from both EMR and RNI is unstable and non-monotonic, with diverse trends for different strategies. For example, for Degree and NetShield, JSD grows significantly with number of seeds k, while for Random it drops.

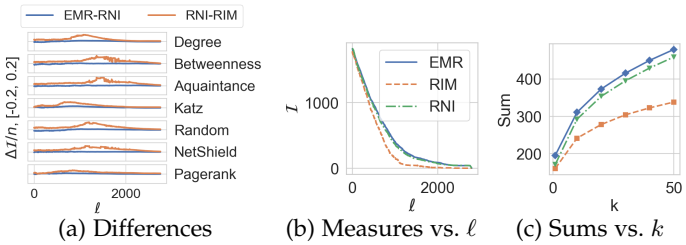


(a) EMR vs RNI (b) EMR vs RIM (c) RNI vs RIM

(d) EMR vs RNI (e) EMR vs RIM (f) RNI vs RIM

Fig. 6: JS divergence of measures per strategy.

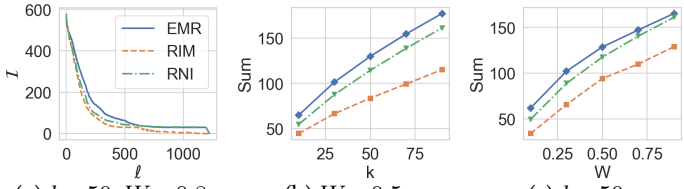
Figure 7a plots the differences EMR-RNI and RNI-RIM vs. ℓ on the VK network. RNI-RIM has a convex shape with a maximum in the middle-range ℓ, while EMR-RNI is almost zero in the whole range. This behavior differs from the one we observed with the BA and DBLP networks, where there is a peak on EMR-RNI. Figure 7b plots non-aggregate measure values for $k = 40$. RNI is very close to EMR along the whole range of ℓ; on the other hand, RNI-RIM also peaks close to the maximum curvature of lines. Figure 7c shows that the effect becomes stronger with larger k, *aggregating* over all ℓ values: SRNI remains close to SEMR, while SRIM diverges from the others; this divergence implies that, on power-law networks, knowledge about the attack, gained when moving from RIM to RNI, is more valuable than knowledge about the stochastic edge outcome, gained when moving from RNI to EMR.



(a) Differences (b) Measures vs. ℓ (c) Sums vs. k

Fig. 7: Dependency of measures on VK social network.

Figures 8 and 9 show the proximity among the three measures on the Blogs and GRP networks. On the *power-law* Blogs network, trends are similar to VK, with RNI close to EMR. However, on the *homogeneous* GRP network, RNI is close to RIM for the whole spectrum of network shape parameters. We conclude that network topology determines what gain of knowledge matters most; on a homogeneous network, knowledge about a probabilistic outcome is more valuable than knowledge about the attack.



(a) $k=50$, $W=0.8$ (b) $W=0.5$ (c) $k=50$

Fig. 8: Measures vs. ℓ, k, W, Blogs network. *Random*.

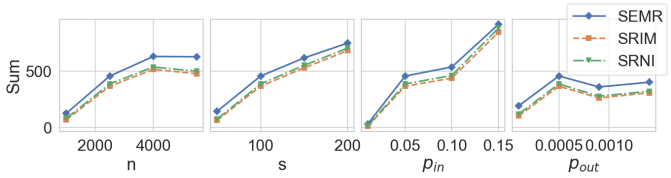


Fig. 9: Dependency on network parameters. GRP. $n=2500$, $s=100$, $p_{in}=0.05$, $p_{out}=5\cdot10^{-4}$, $k=10$, $W=0.3$, *Random*.

Another interesting feature is the shape of the tail of distribution (Figures 5, 7b and 8a). There exists a value of $\ell = \ell'$, such that all three measures converge to the value of k or ℓ grows towards ℓ', but for $\ell > \ell'$ RIM drops to 0, while others remain at the value of k. The drop of RIM is concave, with a gap of first derivative. The region $\ell > \ell'$ corresponds to the case where the attacker blocks all nodes by at least one strategy for any seed set. That strategy determines RIM. However, for EMR and RNI, seeds are selected after the attack, therefore there are at least k non-blocked nodes.

5.4 EMR vs RNI: the diffusion entropy

The EMR and RNI measures both pertain to a seeder aware of the attacker's actions, i.e., to *robust immunization*. Their difference lies in the fact that, by EMR, the seeder is also aware of the probabilistic network outcome. This difference expresses the surprise effect or, so to speak, *negative entropy* that a probabilistic diffusion outcome presents to the attacker; it shows how much worse the spread can be in the case of a seeder aware of probabilistic outcomes in comparison to the best guess of a seeder unaware of such outcomes. We study this difference in more detail, using uniform probability assignment so as to focus on structural effects. We consider the absolute difference $D = \mathrm{EMR} - \mathrm{RNI}$, and the relative difference $D_r = (\mathrm{EMR} - \mathrm{RNI})/\mathrm{RNI}$.

Figure 10 shows the surface of D_r vs. ℓ and W on the Minnesota network. D_r is larger for smaller ℓ, and drops with larger edge probabilities. Still, it is not monotonic vs. W; it obtains a maximum value around $W = 1.5$, and the peak is more explicit with smaller ℓ.

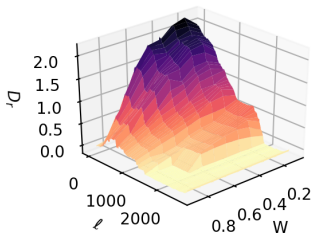
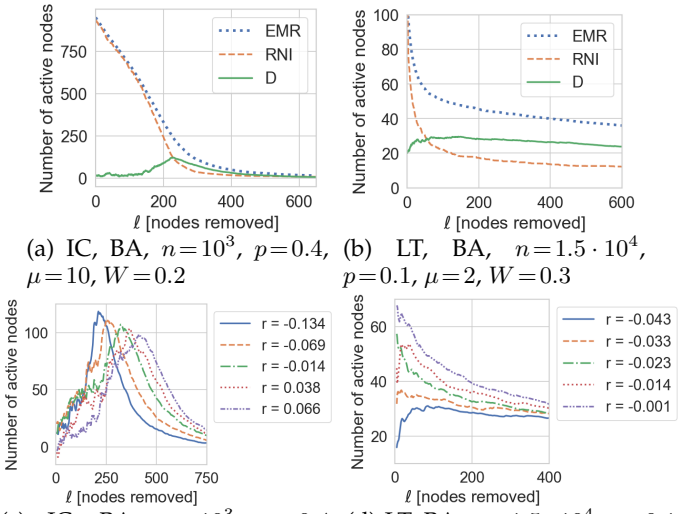


Fig. 10: IC, D_r, Minnesota, $k = 10$

Figure 11a shows that the non-monotonic behavior of D also appears with respect to ℓ on a BA network. We observe

similar behavior with the LT model in Figure 11b; most of the immunization effect is credited to low ℓ values, with the peak shifting leftwards. We observe an opposite concavity of the tails, in comparison with Figure 11a.



(a) IC, BA, $n=10^3$, $p=0.4$, $\mu=10$, $W=0.2$ (b) LT, BA, $n=1.5\cdot10^4$, $p=0.1$, $\mu=2$, $W=0.3$

(c) IC, BA, $n=10^3$, $p=0.4$, $\mu=10$, $W=0.2$ (d) LT, BA, $n=1.5\cdot10^4$, $p=0.1$, $\mu=2$, $W=0.3$

Fig. 11: Local maximum of D; BA: $k=5$

Figures 11c and 11d show D values for BA networks of different assortativity coefficients r, with the IC and LT models. The assortativity coefficient [40], defined as $r = \frac{\sum_{xy} xy(e_{xy}-a_x b_y)}{\sigma_a \sigma_b}$ with $a_x = \sum_y e_{xy}$ and $b_y = \sum_x e_{xy}$, where e_{xy} is the joint probability distribution of the degree values x and y and σ_a, σ_b, the standard deviations of the distributions of a_x and b_y, respectively. r expresses the tendency for nodes of similar degree to be connected. We generate a BA network and increase the coefficient using the edge rewiring technique of [6]; that is a local search heuristic that randomly samples pairs of edges, e.g., pair $\{(v_1,u_1),(v_2,u_2)\}$, and rewires them to $\{(v_1,u_2),(v_2,u_1)\}$ if that improves the objective. The figure legends show the r values resulting after epochs of 2000 iterations, starting with the original BA network. Notably, the observed peaks become flattened as r grows.

D also relates to the relative marginal gain seeds addition by the seeder. We define $\delta_{\theta_i}(\ell)$ as the relative marginal gain of the second seed for any strategy $\theta_i \in \Theta$ under ℓ attacked nodes:

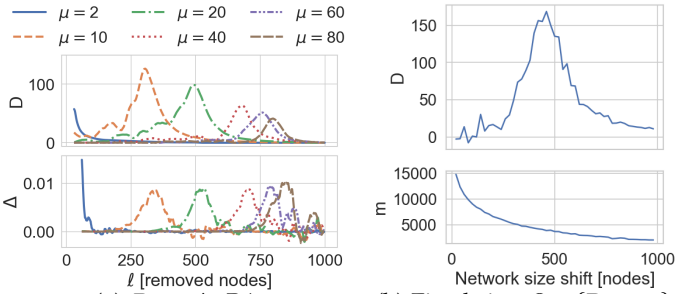
$$\delta_{\theta_i}(\ell) = \frac{\max_{S:|S|=1} \sigma_{\theta_i(\ell)}(S) - \max_{S:|S|=2} \sigma_{\theta_i(\ell)}(S)}{\max_{S:|S|=1} \sigma_{\theta_i(\ell)}(S)}$$

We then calculate a new quantity $\Delta(\ell)$ as the maximum differential quotient of δ over all strategies for each ℓ:

$$\Delta(\ell) = \max_{\theta_i \in \Theta} \{\delta_{\theta_i}(\ell) - \delta_{\theta_i}(\ell-1)\}$$

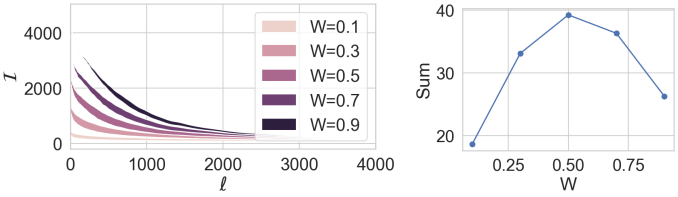
Figure 12a juxtaposes D and Δ for different BA model parameters, plotted with moving average smoothing. Remarkably, their two peaks align, with a slight shift to the right for Δ. This finding implies that, on BA networks, the values of ℓ for which the network ceases to be strongly centralized, hence Δ flattens out, would also cause the highest surprise to an attacker.

Figure 13a plots D as a colored interval vs. ℓ, on the DBLP real-world network, while varying edge probabilities.



(a) D vs. Δ, BA (b) Fixed size, $\Theta=\{\text{Degree}\}$

Fig. 12: Local maximum of D; BA; $n=1000$, $p=0.4$, $\mu=10$; $k=1$



(a) Colored interval shows difference (b) D (SEMR-SRNI)

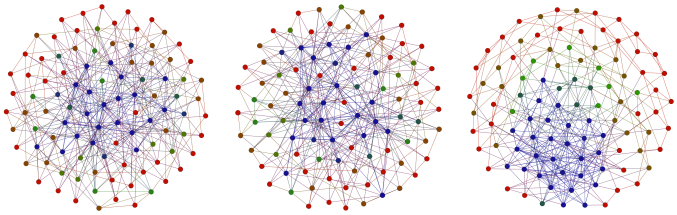
Fig. 13: D on DBLP. $\alpha=1.075, k=50, \beta=40$, *Random*.

D is largest (i.e., widest) for middle-range W values. Figure 13b illustrates this fact, summing over all ℓ values. This non-monotonic dependence of D on W suggests that we may control a network's robustness by tuning edge weights.

We exploit the behaviour of D to generate networks of *enhanced robustness*: we fix size to 1000 nodes, yet first generate a network of larger size and then remove superfluous nodes by the Degree strategy. We call the amount of nodes first added and then removed *shift*. Figure 12b plots D vs. shift. Shifting improves network robustness in terms of D; we create networks in which a seeder has the potential to perform surprisingly well against an attacker. The lower subfigure plots the number of edges in the obtained network; as there is no correlation between the peak of D and number of edges, the peak must be attributed to the network's structure.

5.5 Case Studies

We provide examples of robust networks using the edge rewiring technique introduced in Section 5.4 [6] with a robustness measure as an objective. We experiment with SRIM and SEMR, since SRNI exhibits similar behavior to SEMR (see Section 5.3). The sampling proceeds until $|E|$ iterations bring no change.

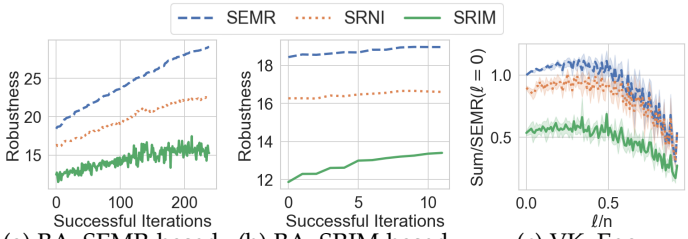


(a) Original (b) SRIM-based (c) SEMR-based

Fig. 14: Robust BA networks

We experiment with a random BA network of 100 nodes, uniform edge probability of 0.5, and 2 seeds. Figure 14 shows the original network (non-robust), and two networks obtained by the aforementioned procedure for SRIM and SEMR, respectively. Colors indicate similar node degrees,

blue for larger, green for medium, and red for smaller. We plot the networks using the Fruchterman Reingold algorithm [66]. We note that the network targeting SEMR has an onion-like structure, as robust static networks do [67], while the others show no evident patterns.



(a) BA, SEMR-based (b) BA, SRIM-based (c) VK, Ego

Fig. 15: Effect of rewiring (a,b), node removals (c)

Figure 15 shows robustness values through the algorithm's iterations. When targeting SEMR (Figure 15a), SRIM is oscillating with a rising trend. When targeting SRIM itself, beneficial changes are hard to find, with only 11 successful iterations and minor robustness improvement (Figure 15b). This result highlights SEMR as a more expressive measure.

In another approach, we incrementally remove nodes that lead to the highest SEMR gain on sampled ego networks of VK. The ego network of node v contains v (*ego*), adjacent nodes of v (*alters*), and the edges between them. Ego networks are useful in understanding the micro-level structure of social networks [68]. We sample ego networks of size between 10 and 50 nodes, having from 10 to 300 edges. We normalize SEMR by the value at $\ell = 0$. Figure 15c presents our results. Solid lines are averaging over 100 samples, and shaded regions represent standard deviations; the x-axis is normalized by n. Greedy node removal leads to 10% of SEMR increase after removing 30% of nodes, and a positive improvement for any number of nodes until $\sim 50\%$ of all nodes removed; other measures are less affected.

5.6 SEMR Computation Efficiency

Here, we compare the runtime of S-Dagger and BIT-Dagger on BA and Minnesota to the following baselines, which progressively introduce SEMR algorithm features: DFS finds the maximum tree by depth-first search for each node in each MC iteration; TD-SCC performs a deterministic graph condensation (i.e., finds SCCs that form a DAG) and runs a *top-down* breadth-first search from each root node in the DAG to find the maximum tree; BU-SCC performs a deterministic graph condensation with *bottom-up* reachability labelling, similarly to S-Dagger, but without a dynamic reachability index; DynSCC performs graph condensation with *dynamic* bottom-up labeling, but in lieu of using DAGGER, it maintains DAGs naively, decreasing ℓ and rerunning Tarjan's algorithm [69] for each affected DAG node.

DAGGER-based algorithms achieve a significant runtime improvement in comparison to baselines. BA (Figure 16a) is a denser, power-law network, while Minessota (Figure 16b) is a sparse homogeneous network. In both cases, DFS is the worst approach; graph condensation significantly improves runtime. On Minessota, the runtime of TD-SCC and BU-SCC even improves as weight W grows, as more SCCs appear. On this sparse network, the efficient maintenance of SCCs is crucial. DynSCC maintains SCCs less efficiently than DAGGER, hence its runtime deteriorates as W grows. BIT-Dagger is less efficient than S-Dagger on the sparse graph, though more efficient on the dense power-law network, as traversing the labels of each root node and retrieving SCC sizes corresponding to reachable nodes is less costly for sets than bitstrings. On a sparse graph, the bitset data structure incurs a large overhead traversing sparse bit strings. Still, on a dense graph, the bitset structure compensates by significantly more efficient set operations. Henceforward, we use S-Dagger as the default option.

(a) BA, $W = 0.3, p = 0.4, \mu = 10$. (b) Minnesota road network

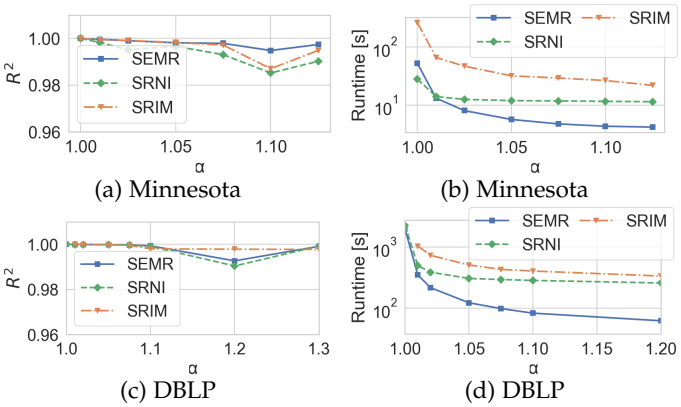
Fig. 16: EMR computation. $\Theta = \{\text{Degree}\}$, $k = 1$; the immunization runtime is trivial.

5.7 Sampling Accuracy

To scale to larger networks, we evaluate our measures using sampled values of ℓ only. As the measures present a rapid decrease in the beginning of the ℓ range, we set a large initial sampling rate, and increase the sample interval geometrically with ℓ. A parameter α defines that geometric growth. We thereby sample ℓ values of

$$\ell \in \{0\} \cup \left\{ \sum_{i=1}^{j} \alpha^{i-1} \right\}_{j=2}^{\lfloor \log_\alpha (1+n(\alpha-1)) \rfloor}$$

If $\alpha = 1$, we sample the complete set of ℓ values. For $\alpha > 1$, we use cubic splines [70] to fit the sampled values and thus obtain robustness measures for the complete range of ℓ.



(a) Minnesota (b) Minnesota

(c) DBLP (d) DBLP

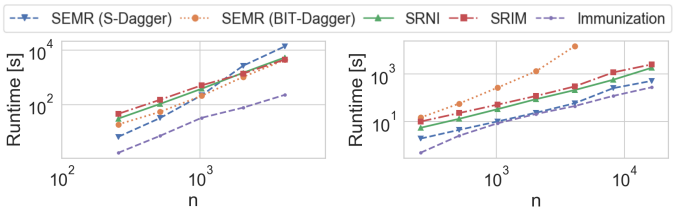
Fig. 17: Performance vs. α; R^2 averaged over adversaries, *Random* prob; Minnesota $W = 0.7$, $k = 10$; DBLP $W = 0.5$, $k = 50$; immunization time 2s on Minnesota, 82s on DBLP.

Figure 17 shows the effect of α on Minnesota and DBLP networks (power-law and homogeneous, respectively). We measure the coefficient of determination R^2 between $\sum_\ell \nu$ calculated for each ℓ (observation), and using cubic splines over sampled values of ℓ (model). On DBLP, SRIM does not terminate with $\alpha = 1$, so we use a fitted model for $\alpha = 1.01$ as ground truth; α increasing from 1 to 1.075 does not affect the accuracy of fit significantly, while runtime drops. Yet R^2 drops for $\alpha = 1.1$ on Minnesota and $\alpha = 1.2$ on DBLP. Ergo, we set $\alpha = 1.075$ as a default value in the following.

5.8 Scalability

Figure 18 presents the scalability of the proposed algorithms, as well as with runtime required for immunization. BIT-Dagger looses in runtime in all cases except on dense

networks of larger size, where S-Dagger becomes the slowest due to its costly set operations. SRNI and SRIM have nearly similar runtimes in all cases.



(a) Dense BA, $p=0.9$, $W=0.1$, $\mu=20$, $k=30$, *Uniform*

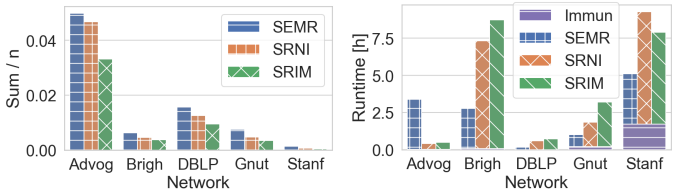
(b) Sparse BA, $p=0.2$, $W=0.1$, $\mu=3$, $k=30$, *Random*

(c) Dense GRP, $W=0.6$, $k=50$, $s=60$, $v=0.5$, $p_{in}=0.015$, $p_{out}=5\cdot10^{-4}$, *Random*.

(d) Sparse GRP, $W=0.1$, $k=50$, $s=40$, $v=0.5$, $p_{in}=0.01$, $p_{out}=10^{-4}$, *Random*.

Fig. 18: Scalability on synthetic networks. $\alpha = 1.075$.

Figure 19 shows results for larger real-world networks. Advogato yields a larger fraction of active nodes $\sum_\ell \nu$, so SEMR performs poorly. In reverse, on Brightkite the sum is small, so SEMR is the fastest. DBLP allows a larger sum than Brightkite and Gnutella, yet yields lower runtime, due to its more *flat* structure, i.e., small edge density and maximum degree. The SRNI-SRIM and SEMR-RNI differences appear similar, except for Advogato, where awareness of the attack leads to a large increase. On Stanford data, SRNI and SRIM did not terminate within 10h for $W = 0.5$ and $\alpha = 1.075$; we decreased the problem complexity by setting $W = 0.1$ and $\alpha = 1.2$, hence the observed spread is low.



(a) Robustness (b) Runtime

Fig. 19: Scalability on larger networks. $W=0.5$, *Random*, $k=100$, $\alpha=1.075$; on Stanford: $W=0.1$, $\alpha=1.2$

6 Conclusions

We introduced three aggregate measures that evaluate the diffusion robustness of probabilistic networks. We anchor these measures on a seeder who orchestrates an Independent Cascade diffusion under node attacks. Each measure is based on a notion of worst-case maximum expected spread. We introduced efficient algorithms to calculate these measures and sample-based versions thereof that enable their computation on realistic networks of up to 10^5 nodes. Our experimental study revealed that measures sharing the same notion of seeder awareness regarding the adversarial attack are closer on scale-free networks, while those sharing the same notion of awareness regarding the network instance are closer on homogeneous networks. Our results provide tools for assessing and enhancing the robustness of real-world probabilistic networks. In the future, we plan to study measuring robustness on anonymized networks [71], [72] while safeguarding privacy, in the spirit of [73].

References

[1] A. Pagani, G. Mosquera, A. Alturki, S. Johnson, S. Jarvis, A. Wilson, W. Guo, and L. Varga, "Resilience or robustness: identifying topological vulnerabilities in rail networks," *Royal Society Open Science*, vol. 6, no. 2, p. 181301, 2019.

[2] A. Logins, P. Karras, and C. S. Jensen, "Multicapacity facility selection in networks," in *ICDE*, 2019, pp. 794–805.

[3] Y. Li, J. Fan, Y. Wang, and K. Tan, "Influence maximization on social graphs: A survey," *IEEE TKDE*, vol. 30, no. 10, pp. 1852–1872, 2018.

[4] S. Ivanov, K. Theocharidis, M. Terrovitis, and P. Karras, "Content recommendation for viral social influence," in *SIGIR*, 2017.

[5] Y. Li, J. Fan, G. V. Ovchinnikov, and P. Karras, "Maximizing multifaceted network influence," in *ICDE*, 2019, pp. 446–457.

[6] C. M. Schneider, A. A. Moreira, J. S. Andrade, S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proc. National Academy of Sciences*, vol. 108, no. 10, pp. 3838–3841, 2011.

[7] N. E. Brunk, L. S. Lee, J. A. Glazier, W. Butske, and A. Zlotnick, "Molecular jenga: the percolation phase transition (collapse) in virus capsids," *Physical Biology*, vol. 15, no. 5, p. 056005, 2018.

[8] W. Ellens and R. E. Kooij, "Graph measures and network robustness," *CoRR*, vol. abs/1311.5064, 2013.

[9] T. A. Schieber, M. G. Ravetti, and P. M. Pardalos, "A review on network robustness from an information theory perspective," in *Proc. 9th Intl Conf. on Discr. Optim. and Oper. Res.*, 2016, pp. 50–60.

[10] J. Liu, M. Zhou, S. Wang, and P. Liu, "A comparative study of network robustness measures," *Frontiers of Computer Science*, vol. 11, no. 4, pp. 568–584, 2017.

[11] V. H. L. Patricio, F. Daolio, H. J. Herrmann, and M. Tomassini, *Propagation Phenomena in Real World Networks. Generating Robust and Efficient Networks Under Targeted Attacks*, ser. Intelligent Systems Reference Library. Springer, 2015, pp. 215–224.

[12] D. Stauffer and A. Aharony, *Introduction to percolation theory*. Taylor & Francis, 2003.

[13] G. W. Klau and R. Weiskircher, "Robustness and resilience," in *Network Analysis*. Springer Berlin Heidelberg, 2005, pp. 417–437.

[14] X. He and D. Kempe, "Robust influence maximization," in *KDD*, 2016, pp. 885–894.

[15] W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou, "Robust influence maximization," in *KDD*, 2016, pp. 795–804.

[16] H. Yildirim, V. Chaoji, and M. J. Zaki, "DAGGER: A scalable index for reachability queries in large dynamic graphs," *CoRR*, vol. abs/1301.0977, 2013.

[17] A. Logins, Y. Li, and P. Karras, "On the robustness of cascade diffusion under node attacks," in *TheWebConf*, 2020, pp. 2711–2717.

[18] P. Crucitti, V. Latora, and M. Marchiori, "Model for cascading failures in complex networks," *Physical Review E*, vol. 69, no. 4, p. 045104, 2004.

[19] E. Vynnycky and R. White, *An introduction to infectious disease modelling*. Oxford University Press, 2010.

[20] O. Lordan and M. Albareda-Sambola, "Exact calculation of network robustness," *Reliability Engineering & System Safety*, vol. 183, pp. 276–280, 2019.

[21] S. Ivanov and P. Karras, "Harvester: Influence optimization in symmetric interaction networks," in *IEEE DSAA*, 2016, pp. 61–70.

[22] A.-L. Barabási, *Network science*. Cambridge university press, 2016.

[23] J. K. Weber and V. S. Pande, "Percolation-like phase transitions in network models of protein dynamics," *The Journal of Chemical Physics*, vol. 142, no. 21, p. 215105, 2015.

[24] B. Wang, G. Chen, L. Fu, L. Song, and X. Wang, "DRIMUX: Dynamic rumor influence minimization with user experience in social networks," *IEEE TKDE*, vol. 29, no. 10, pp. 2168–2181, 2017.

[25] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, p. 378, 2000.

[26] G. Paul, T. Tanizawa, S. Havlin, and H. E. Stanley, "Optimization of robustness of complex networks," *The European Physical Journal B*, vol. 48, no. 1, pp. 149–149, 2005.

[27] Y. Khorramzadeh, "Network reliability: Theory, estimation, and applications," Ph.D. dissertation, Virginia Tech, 2015.

[28] C. Frey, A. Züfle, T. Emrich, and M. Renz, "Efficient information flow maximization in probabilistic graphs," *IEEE TKDE*, vol. 30, no. 5, pp. 880–894, 2018.

[29] L. Chayes, R. H. Schonmann *et al.*, "Mixed percolation as a bridge between site and bond percolation," *The Annals of Applied Probability*, vol. 10, no. 4, pp. 1182–1196, 2000.
[30] I. Bogunovic, "Robust protection of networks against cascading phenomena," Master's thesis, ETH Zürich, 2012.
[31] C. Chen, H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau, "Node immunization on large graphs: Theory and algorithms," *IEEE TKDE*, vol. 28, no. 1, pp. 113–126, 2016.
[32] H. Zhang, S. Mishra, M. T. Thai, J. Wu, and Y. Wang, "Recent advances in information diffusion and influence maximization in complex social networks," *Opportunistic Mobile Social Networks*, vol. 37, no. 1.1, p. 37, 2014.
[33] K. J. Rothman, S. Greenland, and T. L. Lash, *Modern Epidemiology*. Lippincott Williams & Wilki, 2013.
[34] R. Durrett, "Some features of the spread of epidemics and information on a random graph," *Proc. National Academy of Sciences*, vol. 107, no. 10, pp. 4491–4498, 2010.
[35] S. Lim, J. Shin, N. Kwak, and K. Jung, "Phase transitions for information diffusion in random clustered networks," *The European Physical Journal B*, vol. 89, no. 9, p. 188, 2016.
[36] Y. Zhang and B. A. Prakash, "Data-aware vaccine allocation over large networks," *ACM TKDD*, vol. 10, no. 2, pp. 20:1–20:32, 2015.
[37] D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD*, 2003, pp. 137–146.
[38] S. Tsugawa and H. Ohsaki, "On the robustness of influence maximization algorithms against non-adversarial perturbations," in *ASONAM*, 2017, pp. 91–94.
[39] D. Kalimeris, G. Kaplun, and Y. Singer, "Robust influence maximization for hyperparametric models," in *ICML*, 2019.
[40] M. E. J. Newman, "Mixing patterns in networks," *Phys. Rev. E*, vol. 67, p. 026126, Feb 2003.
[41] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model," in *SDM*, 2012, pp. 463–474.
[42] A. Logins and P. Karras, "An experimental study on network immunization," in *EDBT*, 2019, pp. 726–729.
[43] M. B. Baig and L. Akoglu, "Correlation of node importance measures: An empirical study through graph robustness," in *WWW Conference Companion*, 2015, pp. 275–281.
[44] K. Scaman, A. Kalogeratos, L. Corinzia, and N. Vayatis, "A spectral method for activity shaping in continuous-time information cascades," *CoRR*, vol. abs/1709.05231, 2017.
[45] A. Logins and P. Karras, "Content-based network influence probabilities: Extraction and application," in *ICDM Workshops*, 2019.
[46] R. Cohen, S. Havlin, and D. Ben-Avraham, "Efficient immunization strategies for computer networks and populations," *Physical review letters*, vol. 91, no. 24, p. 247901, 2003.
[47] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.
[48] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.
[49] N. Schwartz, R. Cohen, D. ben Avraham, A.-L. Barabási, and S. Havlin, "Percolation in directed scale-free networks," *Physical Review E*, vol. 66, no. 1, p. 015104, 2002.
[50] N. Ohsaka, T. Akiba, Y. Yoshida, and K. Kawarabayashi, "Dynamic influence analysis in evolving networks," *PVLDB*, vol. 9, no. 12, pp. 1077–1088, 2016.
[51] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization techniques*. Springer, 1978.
[52] D. E. Knuth, *The Art of Computer Programming, Volume 4, Fascicle 1: Bitwise Tricks & Techniques; Binary Decision Diagrams*. Addison-Wesley Professional, 2009.
[53] X. He, personal communication, 2019.
[54] U. Brandes, M. Gaertler, and D. Wagner, "Experiments on graph clustering algorithms," in *ESA*, 2003, pp. 568–579.
[55] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Physical review E*, vol. 65, no. 2, p. 26107, 2002.
[56] T. Schank and D. Wagner, "Approximating clustering coefficient and transitivity," *Journal of Graph Algorithms and Applications*, vol. 9, no. 2, pp. 265–275, 2005.
[57] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 U.S. election: Divided they blog," in *Proc. 3rd Intl Workshop on Link Discovery*, 2005, pp. 36–43.
[58] J. Kunegis, "KONECT: the koblenz network collection," in *WWW Conference*, 2013, pp. 1343–1350.
[59] M. Ley, "The DBLP computer science bibliography: Evolution, research issues, perspectives," in *SPIRE*, 2002, pp. 1–10.
[60] P. Massa, M. Salvetti, and D. Tomasoni, "Bowling alone and trust decline in social network sites," in *The 8th IEEE Intl Conf. Dependable, Autonomic and Secure Computing*, 2009, pp. 658–663.
[61] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015, pp. 4292–4293. [Online]. Available: http://networkrepository.com
[62] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *KDD*, 2011.
[63] J. Leskovec, J. M. Kleinberg, and C. Faloutsos, "Graph evolution: Densification and shrinking diameters," *ACM TKDD*, vol. 1, no. 1, p. 2, 2007.
[64] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Math.*, vol. 6, no. 1, pp. 29–123, 2009.
[65] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *SIGMOD*, 2015, pp. 1539–1554.
[66] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *ICWSM*, 2009.
[67] Y. Hayashi and N. Uchiyama, "Onion-like networks are both robust and resilient," *Scientific Reports*, vol. 8, no. 1, p. 11241, 2018.
[68] V. Arnaboldi, M. Conti, M. L. Gala, A. Passarella, and F. Pezzoni, "Ego network structure in online social networks and its impact on information diffusion," *Comp. Comm.*, vol. 76, pp. 26–41, 2016.
[69] R. E. Tarjan, "Depth-first search and linear graph algorithms," *SIAM Journal on Computing*, vol. 1, no. 2, pp. 146–160, 1972.
[70] C. de Boor, *A Practical Guide to Splines*. Springer-Verlag, 2001.
[71] M. Xue, P. Karras, C. Raïssi, P. Kalnis, and H. K. Pung, "Delineating social network data anonymization via random edge perturbation," in *CIKM*, 2012, p. 475–484.
[72] S. Nobari, P. Karras, H. Pang, and S. Bressan, "L-opacity: Linkage-aware graph anonymization," in *EDBT*, 2014, pp. 583–594.
[73] P. Karras, A. Nikitin, M. Saad, R. Bhatt, D. Antyukhov, and S. Idreos, "Adaptive indexing over encrypted numeric data," in *SIGMOD*, 2016, pp. 171–183.

Alvis Logins received the PhD degree in Computer Science from Aarhus University, Denmark, in 2020. His research interests include Data Mining, Network Science, Databases, and Machine Learning.

Yuchen Li received double BSc degrees in applied math and computer science (both with first class honors) and a PhD degree in computer science from the National University of Singapore (NUS), in 2013 and 2016, respectively. He is an assistant professor with the School of Information Systems, Singapore Management University (SMU). His research interests include graph analytics and heterogeneous computing.

Panagiotis Karras is an Associate Professor of Computer Science at Aarhus University. In his research he designs robust and versatile methods for data access, mining, analysis, and representation. He received an MSc in Electrical and Computer Engineering from the National Technical University of Athens and a PhD in Computer Science from the University of Hong Kong. He has been awarded a Hong Kong Young Scientist Award, a Singapore Lee Kuan Yew Postdoctoral Fellowship, a Rutgers Business School Teaching Excellence Fellowship, and a Skoltech Best Faculty Performance Award.