Future Generation Computer Systems 83 (2018) 413–424

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

QRE: Quick Robustness Estimation for large complex networks

Sebastian Wandelt [a,b], Xiaoqian Sun [a,b,*], Massimiliano Zanin [c,d], Shlomo Havlin [e]

[a] School of Electronic and Information Engineering, Beihang University, 100191 Beijing, China
[b] Beijing Laboratory for General Aviation Technology, 100191 Beijing, China
[c] The Innaxis Foundation and Research Institute, 28006 Madrid, Spain
[d] Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
[e] Department of Physics, Bar-Ilan University, Ramat-Gan 52900, Israel

HIGHLIGHTS

- We develop a robustness estimation technique for large complex networks.
- A set of subquadratic-time network metrics is exploited for node importance.
- Sampling of robustness is performed based on equi-depth intervals.
- Experiments show that our technique estimates R values better than betweenness centrality.

ARTICLE INFO

Article history:
Received 31 October 2016
Received in revised form
23 January 2017
Accepted 8 February 2017
Available online 14 February 2017

Keywords:
Complex networks
Robustness estimation
Scalability

ABSTRACT

Robustness estimation is critical for the design and maintenance of resilient networks. Existing studies on network robustness usually exploit a single network metric to generate attack strategies, which simulate intentional attacks on a network, and compute a metric-induced robustness estimation, called R. While some metrics are easy to compute, e.g. degree, others require considerable computation efforts, e.g. betweenness centrality. We propose Quick Robustness Estimation (QRE), a new framework and implementation for estimating the robustness of a network in sub-quadratic time, i.e., significantly faster than betweenness centrality, based on the combination of cheap-to-compute network metrics. Experiments on twelve real-world networks show that QRE estimates the robustness better than betweenness centrality-based computation, while being at least one order of magnitude faster for larger networks. Our work contributes towards scalable, yet accurate robustness estimation for large complex networks.

1. Introduction

During the last decades, empirical studies have characterized a plethora of real-world systems as complex networks [1,2], including air transportation networks [3–5], electrical power grids [6,7], Internet backbone [8], inter-bank networks [9], etc. Most networks present a well-recognized resistance against random failures [10] but disintegrate rapidly under intentional attacks targeting relatively important nodes in the network first [11,12]. Moreover, initial shocks can sometimes lead to cascading failures [13]. Examples of recent extensive, wide-ranging network failures include European air traffic disruption caused by Icelandic volcano Eyjafjallajökull [14], large-scale power outages in the United States [15], computer virus spreading [16], cross-continental supply-chain shortages in the Japanese tsunami aftermath [17]. Such disruptions cause high economic costs [18], making the analysis and improvement of resilience become one of the most critical challenges in applied network theory [19,20].

A first proposal for a metric assessing the network robustness was presented in [21]. Given a network with N nodes, the robustness is defined as $R = \frac{1}{N}\sum_{Q=1}^{N} s(Q)$, where $s(Q)$ is the size of the giant component (GC size) after removing Q nodes. The value of R depends significantly on the underlying attacking strategy, i.e. the order in which node removals occur. Studies on network robustness usually select a single network metric, e.g. degree, betweenness, or collective influence [22], to rank nodes or links; such

[*] Corresponding author at: School of Electronic and Information Engineering, Beihang University, 100191 Beijing, China.
E-mail addresses: wandelt@informatik.hu-berlin.de (S. Wandelt), sunxq@buaa.edu.cn (X. Sun), mzanin@innaxis.org (M. Zanin), havlins@gmail.com (S. Havlin).

http://dx.doi.org/10.1016/j.future.2017.02.018
0167-739X/© 2017 Elsevier B.V. All rights reserved.

ranking is then used to sequentially remove the elements of networks, from the most important to the least important one.

This standard approach has three limitations. First, the obtained R value reflects the robustness against a particular attacking strategy, for instance, by attacking high-degree nodes first. Consequently, attacking the same network according to different strategies yields different values of R, each of them only capturing a specific dimension of the network resilience. In general, one is interested in the worst-case robustness, represented by the minimum R value. Second, different network metrics have significantly different computational requirements. Notably, methods obtaining the lower R scores often have the highest computational requirements. For instance, computing the betweenness of all nodes in the network, a strategy which has been shown rather effective for attacking a network, needs time at best quadratic with the number of nodes. This makes it very difficult to obtain good attacks (leading to smaller, realistic R values) for very large networks with millions of nodes. The computation of betweenness values in a network with 100,000 nodes takes almost one day on today's consumer computers; and doubling the network size further quadruples the execution time. Third, existing studies often employ a fixed-length interval sampling strategy in order to avoid computing the GC size N times. As we show below, this static sampling yields an over-estimated network robustness, particularly for large vulnerable networks.

In this paper, we propose a new technique (QRE = Quick Robustness Estimation) for estimating the network robustness in subquadratic time. We assess the importance of nodes by exploiting a set of network metrics which can be computed in linear time regarding the number of nodes and edges in the network. Furthermore, we iteratively adapt sampling intervals fitting the shape of the robustness curves, yielding an increasingly improved solution after each iteration. Experiments on 12 real-world networks show that QRE estimates R-values better than interactive betweenness centrality, while having attractive computational properties. Our work contributes towards scalable, yet accurate robustness estimation for large complex networks. In our study, we use the size of the giant component under an attack as the robustness measure. In the literature, several other views/terminologies on criticality of networks have been proposed, addressing a multitude of measures and observations; see [7] for an overview on measures for power grids. Examples for these different terminologies include reliability [23], disturbance [24], contingency [24], vulnerability [25], and stability [26]. See also [27] for a review on modeling and simulation of interdependent critical infrastructure systems. Moreover, in our study we focus on the case of single networks, while critical infrastructures are recently often modeled as networks of networks [28,29].

The remainder of this paper is structured as follows. In Section 2, we describe the design and implementation of QRE. Section 3 evaluates QRE and competitors against a set of random and real-world networks. The paper is concluded with a discussion and some ideas for future work in Section 4.

2. Methods

The description of the QRE robustness estimation methodology is here organized around five subsections. Section 2.1 introduces and discusses the measurement of network robustness. In Section 2.2, we describe how attack traces induced by network metric rankings can be exploited for robustness estimation and the limitation of this method. We develop the notion of partial attack traces in Section 2.3 and generalize it towards multiple partial attack traces in Section 2.4. The overall framework and implementation of QRE is presented in Section 2.5.

2.1. Robustness estimation

Inspired on the well-known concept of percolation in statistical physics [30–34], the robustness of a network is usually defined as the critical fraction of nodes that causes a sudden disintegration [11], the latter being measured as the relative reduction in the size of the giant (largest connected) component. Note that the disintegration is higher when the GC size is smaller [35]. The intuition here is that the functionality of a network strongly depends on the number of connected nodes. In this paper we use the robustness measure R, as defined in [21] and described in Section 1. Given a network with N nodes, the robustness is defined as $R = \frac{1}{N} \sum_{Q=1}^{N} s(Q)$, where $s(Q)$ is the GC size after removing Q nodes. The computation of R for a network requires a strategy to derive a node ranking, based on which the nodes are removed from the network. In Fig. 1, we show the robustness curves and their corresponding R values for three example networks, when nodes are removed according to their degree, i.e., highly connected nodes first (please refer to Section 3 for a description of the corresponding datasets). The football network (Fig. 1, left panel) can be considered robust: Only after attacking 50% of the nodes, the GC size is significantly reduced. The network netscience (Fig. 1, right panel), on the other hand, is very fragile: Attacking less than 5% of the nodes leads to an almost completely disintegrated network. As previously discussed, these results only provide a limited view on the network robustness, as they correspond to a specific attack strategy; one may then ask if a better node ranking criterion could be devised, and what would be the impact on the robustness estimation.

2.2. Single attack traces

In previous studies on network robustness, node rankings are usually defined by a single network metric. While the use of network metrics have been proposed in the past, we review five most significant ones:

1. **Degree**: The number of neighbors of a node, i.e., how many nodes can be reached within one hop. The degree of all nodes can be obtained in linear time regarding the number of nodes.
2. **Betweenness centrality**: $B_i = \sum_{s \neq t} \frac{\sigma_{st}(i)}{\sigma_{st}}$, where σ_{st} is the number of shortest paths going from node s to node t; $\sigma_{st}(i)$ is the number of shortest paths going from node s to node t and passing through node i [36]. This metric essentially indicates the number of shortest paths going through a node. The betweenness of all nodes can be best obtained in quadratic time regarding the number of nodes.
3. **Eigenvector centrality**: The eigenvector for the largest eigenvalue of the adjacency matrix. Nodes with high eigenvector centrality also connect to other nodes which have high eigenvector centrality.
4. **Katz centrality**: Computes the relative influence of a node within a network by measuring the number of the immediate neighbors (first degree nodes) and also all other nodes in the network that connect to the node under consideration through these immediate neighbors [31].
5. **PageRank**: A ranking of the nodes based on the structure of the incoming links, developed for assessing the importance of web pages [37].

Each of these metrics can be used to define an attack: Nodes are removed according to their ranking, from highly important (i.e. High metric values) to secondary (low metric values) ones. In general, two variants of a metric can be used: Static attack (which computes the metric only one time for the complete network) and interactive attack (which recomputes the metric after each removal of the top-ranked node) [38–40]. The choice of the metric is far from trivial. While the betweenness may be a good option,

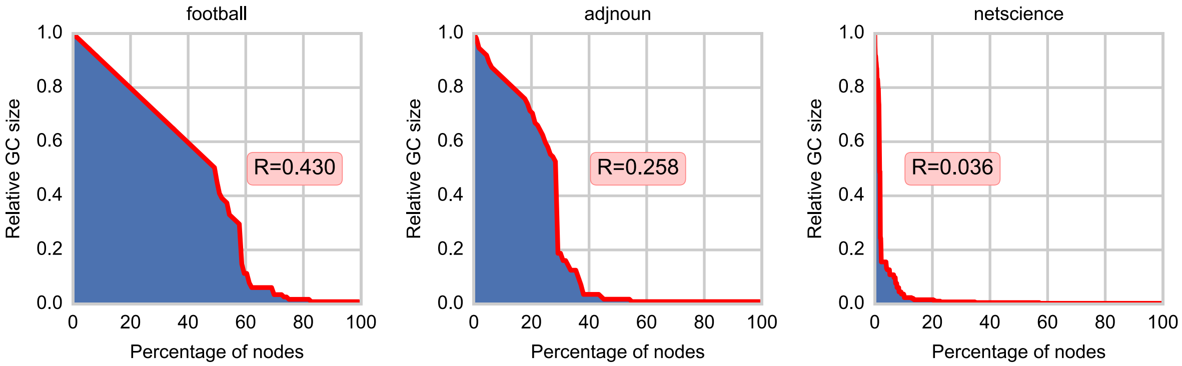
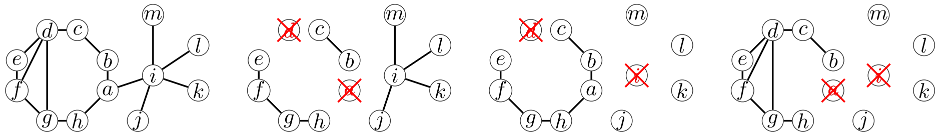


Fig. 1. Robustness curves and computed R values for three networks: Football (rather robust with $R = 0.430$), adjnoun (slightly fragile with $R = 0.258$), netscience (very fragile with $R = 0.036$). We have used the interactive degree of nodes to attack the network.



(a) Original network.

(b) Optimal attack (GC size = 5).

(c) Degree-based attack (GC size = 7).

(d) Betweenness-based attack (GC size = 7).

Fig. 2. An example of a network attack. The process starts with a target network (a), where we want to attack two nodes. In (b) we show an optimal attack, which reduces the GC size from 13 to 5. A degree/betweenness-based attack, as shown in (c) and (d), reduces the GC size to 7; while attacking different node pairs. The combined knowledge from both metrics could be helpful to find the optimal attack, since node d is identified as important by the degree-based ranking and node a is identified as important by the betweenness-based ranking.

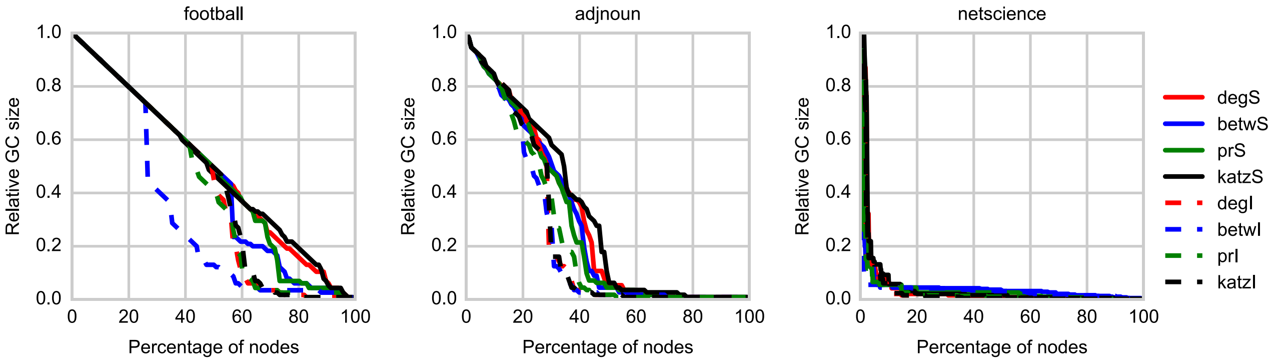


Fig. 3. Comparison of attacking strategies induced by different network metrics (S = static, I = interactive). Different network metrics induced significantly different robustness curves, and consequently, notably different R values. For instance, with interactive betweenness centrality as an attack strategy, football (left) becomes similar fragile as adjnoun (center). The difference in R values for football makes up 40% of the range of R (from minimum 0 to maximum 0.5).

due to its ability in describing between-communities connectivity, it underestimates the importance of local connectivity patterns, which are best captured by metrics like the node degree. Therefore, the choice of the network metric is instrumental for obtaining a realistic R value. This issue is illustrated in Fig. 2 through a simple toy example; Fig. 3 further develops the idea, by showing how the GC size evolves as a function of the considered attack. Note that no exact bound on the time complexity is known for the metrics Eigenvector, Katz, and PageRank, yet experiments usually show a linear running time regarding the number of nodes and links.

An additional aspect often neglected by the physical community is the complexity cost associated with metric computation. While many metrics can be computed in linear time (regarding the number of nodes/links in the network), some need at least quadratic time complexity. If the difference is not relevant for small networks, it becomes important for large graphs, up to the point that the computation of a single betweenness-based attack is often intractable. To illustrate, the computation of a node-degree based attack requires only 2.7 s for a network with 16,706 nodes, while the betweenness computation for the same network takes more than one hour. The divergence increases tremendously with larger networks. Computing an R value is thus a trade-off between accuracy and time complexity, where spending more time often corresponds to higher accuracy. As we will show in our evaluation, the combination of few fast-to-compute network metrics can often outperform computationally expensive metrics.

2.3. *Single partial attack traces*

In the previous section we have discussed how to compute the R value of a network based on a node ranking. For large networks, the process of computing the R value itself, given an existing ranking, requires a significant amount of time: For each node in the ranking list, the node is removed and the GC size is computed. Since measuring the GC sizes takes at least $O(N)$ steps, the overall computation takes quadratic time. Therefore, computing the exact value of R for a large network – even if a good ranking is provided – is still intractable. Below, we discuss how to obtain an approximate R value based on sampling.

The intuition for sampling-based approximation is that, given the formulation of $R = \frac{1}{N}\sum_{Q=1}^{N} s(Q)$, the contribution of a single

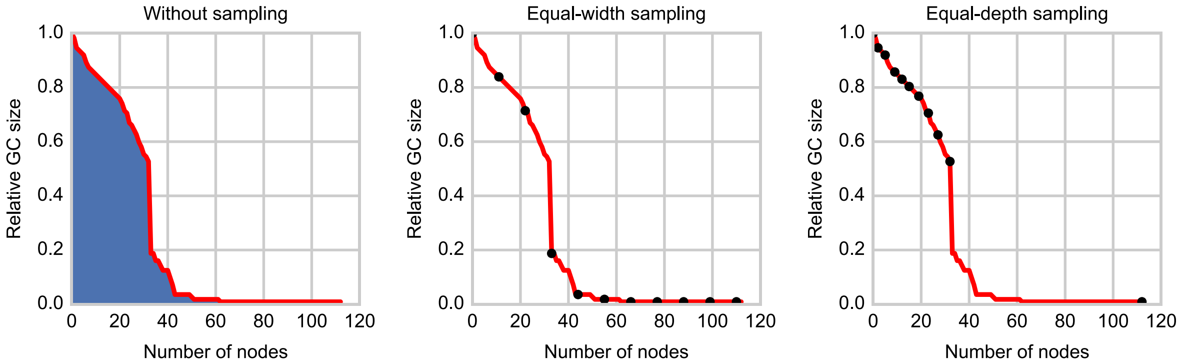


Fig. 4. Sampling for the network adjnoun: The complete robustness curve for adjnoun (left); Equal-width sampling creates many measurements at network snapshots where the network is already disintegrated (center); Equi-depth sampling creates sampling points for network snapshots where the network is still robust (right).

$s(Q)$ term is rather small for large networks: The maximum value of $s(Q)$ is 1 and therefore the contribution of $s(Q)$ cannot exceed $\frac{1}{N}$. For large N, this value converges towards zero. Therefore, we propose to sample the measurements of $s(Q)$ for a limited number of values in $Q \in \{1, \dots, N\}$. Assuming that we sample the measurements at the interval points $\{I_1, \dots, I_m\}$, how do we approximate the values of $s(Q)$ for $I_i < Q < I_{i+1}$? In general, we do not just want any approximation, but we want to avoid underestimation of R, as this may be troublesome in real applications. Thus, we discard linear interpolation. We propose to set $s(Q) = s(I_i)$ for $I_i < Q < I_{i+1}$, i.e., all points inherit the GC size from the closest sample point to the left. Using this strategy, the robustness might be overestimated between two samples, but never underestimated, yielding an upper bound.

Given the upper bound approximation between sample points, the next question is how to distribute sample points. One solution is to perform an equal-width sampling, where all intervals have exactly the same size. This strategy is good for an initial robustness estimation without additional knowledge about the robustness trace. However, once we obtained an initial approximation of the robustness trace, it is often beneficial to adapt the sampling according to the currently best-known trace. Particularly, if the network is very fragile, results become more accurate, if samples points are preferably placed during the initial phase of the attack trace generation.

Instead of equal-width sampling, we propose a method inspired by equal-depth binning for the approximation of an attack trace. The idea is to split the area under the currently best-known curve into regions of approximately the same size. This strategy focuses on parts of the network with higher robustness. Once the network is almost destroyed, it does not make sense to collect further samples in this region anymore, since improvements will only slightly change the R value. For a comparison of equi-width and equi-depth sampling, please see Fig. 4.

2.4. Multiple attack traces, multiple iterations, and approximate betweenness

So far, we have discussed the case of using a single network metric for generating an attack trace by sampling points based on equi-depth sampling. This section deals with the use of several metrics and how to select the one mostly reducing the GC size. Our hypothesis, supported by a comprehensive evaluation in Section 3, is that combining several metrics which have a low time complexity, can outperform harder-to-compute metrics such as betweenness. Our evaluation on real-world networks below confirms this hypothesis.

Finally, in addition to the previously discussed network metrics, we propose to exploit approximate betweenness [41] for ranking the importance of nodes. Instead of computing the exact betweenness of all nodes, an approximation is computed based on a given number of sample node pairs. Thanks to this metric, the time complexity is significantly reduced, while results are still comparable to the exact betweenness analysis.

Algorithm 1 Function: $QRE(G, metrics, iter, maxSampleCount)$

```
Input: Graph $G$, set of metrics $metrics$, number of iterations $iter$, maximum number of samples $maxSampleCount$
Output: Estimated $R$
Let $M = [1, \dots, 1]$ be a list of length $|G.nodes()|$
Let $initgcs = \frac{|LC|}{|G.nodes()|}$, where $LC$ is the largest component of $G$
for $i \in \{1, \dots, iter\}$ do
    Let $samples_x = getSampleIntervals(M, maxSampleCount)$
    Let $pat = getPartialAttackTraceDynamic(G, samples_x, metrics, initgcs)$
    Let $M = updateM(M, samples_x, pat)$
end for
Let $R = sum(M)/|M|$
```

Algorithm 2 Function: $getSampleIntervals(M, maxIntervalCount)$

```
Input: Current minimum trace $M$, sample points $samples_x$, sample observations $samples_{gcs}$
Output: Updated $M$
Let $R = sum(M)/|M|$
Let $result = []$
Let $cursum = 0$
for $i \in \{0, \dots, |M| - 2\}$ do
    $cursum = cursum + \frac{M[i]}{|M|}$
    if $cursum \geq \frac{|result| * R}{maxIntervalCount}$ then
        Append $i$ to result
    end if
end for
```

2.5. Overall framework

In this section, we present our framework for scalable robustness estimation, merging all the insights discussed in Sections 2.1–2.4, we design an algorithm for estimating the robustness of a network. Our overall algorithm is presented in Algorithm 1. First, we initialize the list M, representing the aggregation of currently best known attack traces, with ones.

Firstly we define M as a vector of the size $|N|$, which will encode the aggregation of the best attack traces known at each iteration. Intuitively, M represents the best known attack trace, i.e., the relative GC size as a function of the number of nodes disabled. We initialized all elements in M with 1, indicating that we have no knowledge about any trace initially. Next, we execute the loop

Algorithm 3 Function: *getPartialAttackTraceDynamic*$(G, samples_x, metrics, initgcs)$

```
Input: Network $G$, sample points $samples_x$, set of metrics $metrics$, initial size of giant component $initgcs$
Output: Sampled values $samples_{gcs}$
Let $samples_{gcs} = []$
Let $G_1$ be a copy of G
Let $gcs = \frac{GCSizeRelative(G1,G)}{initgcs}$
Append $gcs$ to $samples_{gcs}$
for $i \in \{0, \dots, |samples_x| - 1\}$ do
    Let $end = |G_1.nodes()|$
    if $i + 1 < |samples_x|$ then
        $end = samples_x[i + 1]$
    end if
    Let $mingcs = 1$
    Let $bestranking = []$
    for $metric \in metrics$ do
        Let $G_2$ be a copy of $G_1$
        Let $ranking$ be the ranking obtained from $metric$ over $G_2$
        for $i \in \{samples_x[i], \dots, end\}$ do
            Remove node $n$ from $G2$
        end for
        Let $gcs = \frac{|LC|*initgcs}{|G.nodes()|}$, where $LC$ is the largest component of $G2$
        if $gcs < mingcs$ then
            $mingcs = gcs$
            $bestranking = ranking$
        end if
    end for
    for $n \in bestranking$ do
        Remove node $n$ from $G1$
    end for
    Append $mingcs$ to $samples_{gcs}$
end for
```

iterations times, and perform three operations in each pass: (1) We get the sample points $samples_x$ based on the current M. In the first iteration, this method will return an equi-width sampling, because M is initialized with equal values. In the following iterations, the sample intervals are based on the equi-depth binning of M. (2) We compute the partial attack traces for the given set of metrics at the sample points $samples_x$. (3) We update the values of M according to the partial attack trace sampled from $samples_x$. After iterating for *iterations* times, we estimate the R value as $R = sum(M)/|M|$. The implementation of these three methods inside the loop are explained in detail below.

In Algorithm 2, we describe our sampling procedure, given the currently best known attack trace encoded in M. Intuitively, we want to calculate the points that uniformly divide the area under the curve. First, we compute the R value from M. Afterwards, we iterate over all sample points and sum up the area under the curve obtained so far in variable *cursum*. Once the aggregated area is larger than the required area for the jth sample point, determined by $\frac{j*R}{maxIntervalCount}$, we add a new sample point to the result. Finally, the result is returned. The worst-case time complexity of this sub-algorithm is $\mathcal{O}(|N|)$ for a network with nodes N: We iterate over all values in M, where M has $|N|$ elements. All other operations in the algorithm are atomic and independent of the network size.

Algorithm 3 is at the heart of our robustness estimation technique: Given pre-computed sample points and a list of network metrics, the algorithm computes the partial attack trace combined from all metrics. Intuitively, for each sample interval, the algorithm probes all metrics, and continues based on the best local attack for the next sample interval. The algorithm is described in detail as follows. First, we create a copy of the network, which is being attacked throughout the remainder. Second, we iterate over all sample points in $samples_x$. We determine the number of to-be-removed nodes first. Then we iterate over the provided network metrics in *metrics*. For each metric we create a copy of the currently attacked network, and then remove $|\{samples_x[i], \dots, end\}|$ top-ranked nodes from the network. We calculate the relative GC size and compare it to the previously best known attack. Throughout the loop, we keep track of the best attack and append the lowest recorded relative GC size to $samples_{gcs}$. Moreover, we remove the nodes with the best ranking from $G1$.

Algorithm 4 Function: *updateM*$(M, samples_x, samples_{gcs})$

```
Input: Current minimum trace $M$, sample points $samples_x$, sample observations $samples_{gcs}$
Output: Updated $M$
for $i \in \{0, \dots, |samples_x|\}$ do
    Let $end = |M|$
    if $i + 1 < |samples_x|$ then
        $end = samples_x[i + 1]$
    end if
    for $i2 \in \{samples_x[i], \dots, end - 1\}$ do
        if $samples_{gcs}[i] < M[i2]$ then
            Let $M[i2] = samples_{gcs}[i]$
        end if
    end for
end for
```

In Algorithm 4, we update the currently best known attack trace minimum by using the sample indices $samples_x$ and their relative GC size measurements $samples_{gcs}$. Essentially, the algorithm iterates over all points in M and checks whether the current point is below the sampling-induced lowest relative GC size for that point. If yes, we update M accordingly. The time complexity of Algorithm 4 is $O(|N|)$, since we access each element in M exactly one time.

Fig. 5 presents an example of the whole process, for five iterations of the algorithm when applied to the adjnoun network; it depicts the change of M in each iteration for the network together with the selected sample points.

3. Results

3.1. Evaluation setup and data sources

We report the results of our evaluation on 12 real-world networks. All experiments were executed on a server with 32 cores and 386 GB RAM, running Fedora 24 (Linux 4.7.5–200.fc24.x86_64). The QRE framework was implemented in a single-threaded fashion, using the network analysis library NetworKit.[1] All datasets are available to download from the UCI Network Data Repository,[2] and have been studied extensively in the past [42–45]. These networks cover a variety of network structures and network scales, see Table 1. We use six network metrics in our evaluation: Degree (deg), betweenness (betw), approximate betweenness (abetw), eigenvector (ev), pagerank (pr), and katz centrality (katz).

3.2. Fine-tuning of parameters

First, we perform a sensitivity analysis of the approximate betweenness against the number of sample node pairs. For four networks of different sizes (dolphins, celegansneural, netscience, and power), we have computed the exact betweenness of nodes

[1] Available at https://networkit.iti.kit.edu/.

[2] Available at https://networkdata.ics.uci.edu/index.php.

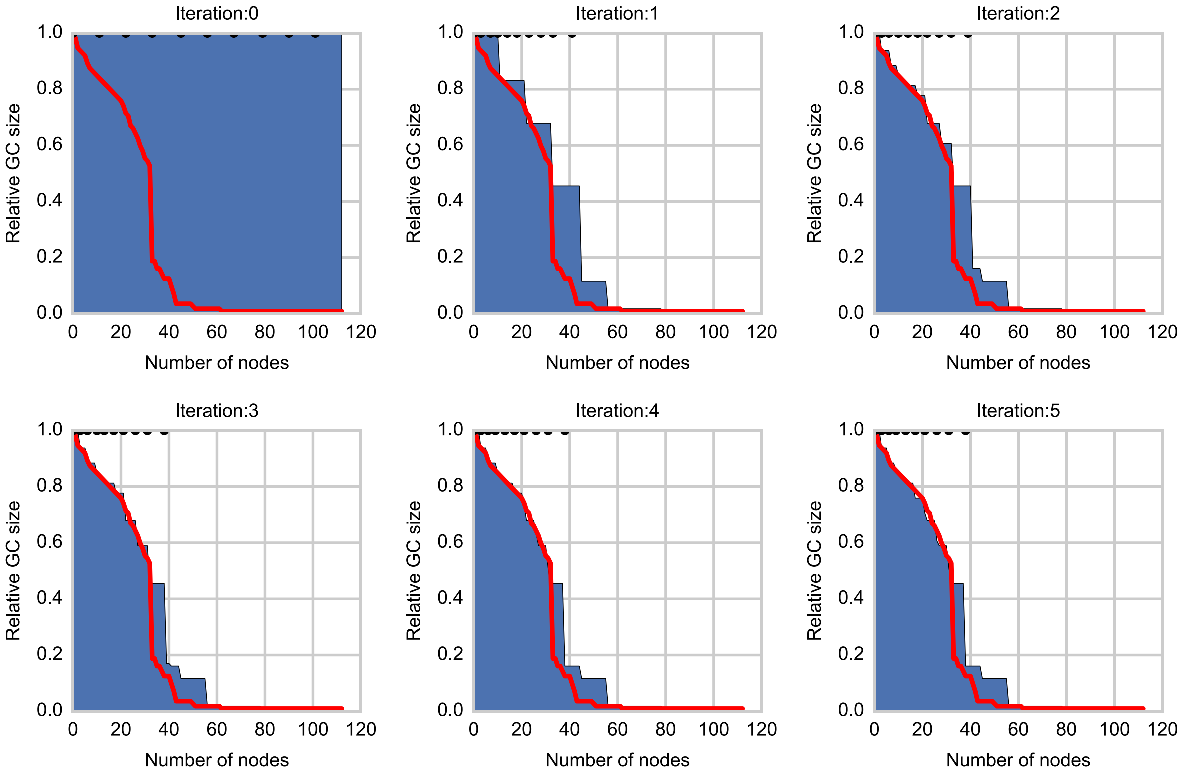


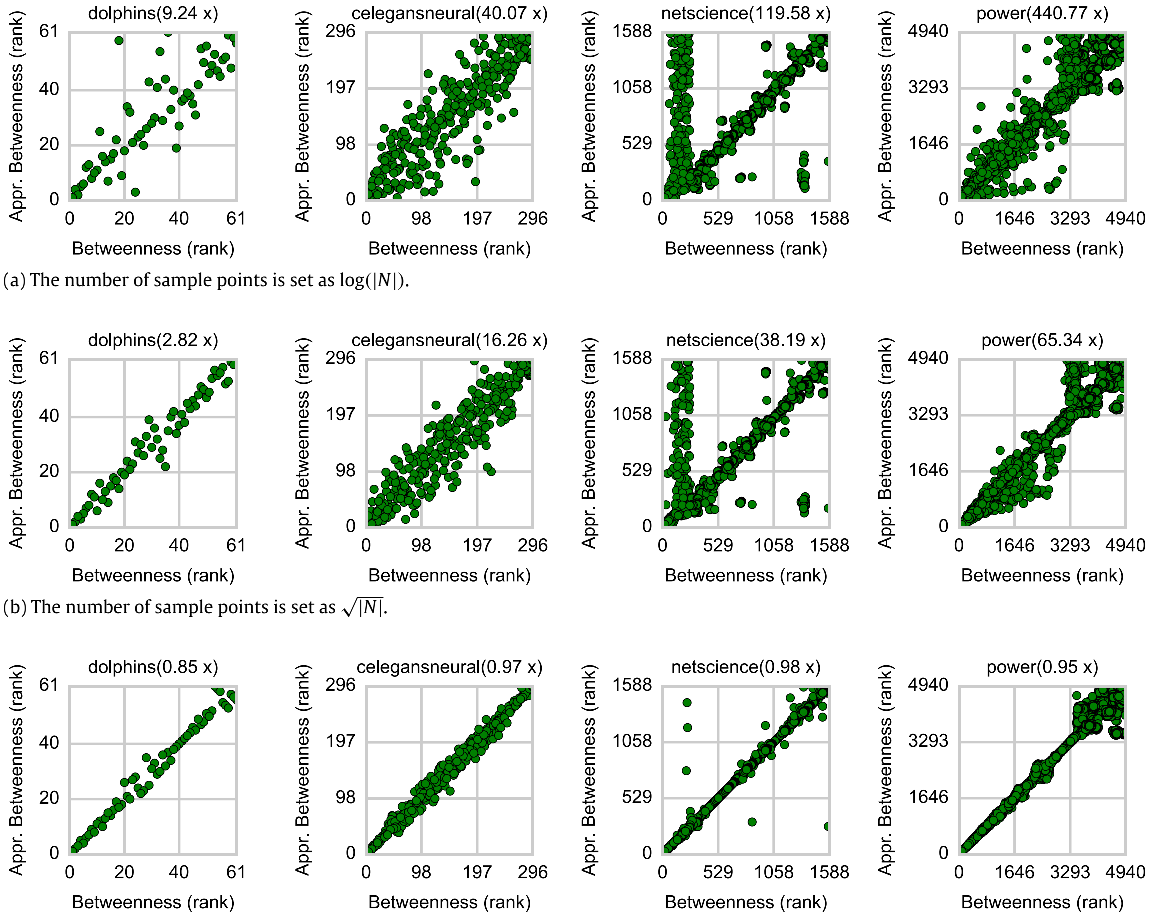
Fig. 5. Visualization of M and ten sampling points with Algorithm 1 for the example network adjnoun and the strategy interactive degree. The red line in all charts corresponds to the robustness curve obtained without sampling, therefore representing the baseline. Initially, M is a vector filled with ones, as represented by the blue area in Iteration 0. After the first equi-depth sampling, which is equivalent to equi-width here, we obtain the M for Iteration 1. Subsequent iterations improve the approximation, with M converging to the actual robustness curve without sampling (red lines). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
List of twelve datasets used in our experiments, ranked by increasing number of nodes. We report standard network properties: Number of nodes (N) and links (L), density, number of connected components (Comp.), relative size of the giant component (Rel. GC), diameter, radius, and assortativity.

Network	N	L	Density	Comp.	Rel. GC	Diameter	Radius	Assortativity
Karate	34	78	13.9%	1	100.0%	5	3	−0.48
Dolphins	62	159	8.41%	1	100.0%	8	5	−0.04
Lesmis	77	254	8.68%	1	100.0%	5	3	−0.17
Polbooks	105	441	8.08%	1	100.0%	7	4	−0.13
Adjnoun	112	425	6.84%	1	100.0%	5	3	−0.13
Football	115	613	9.35%	1	100.0%	4	3	0.16
Celegansneural	297	2 148	4.89%	1	100.0%	5	3	−0.16
Polblogs	1 490	16 715	1.51%	268	82.01%	–	–	−0.22
Netscience	1 589	2 742	0.22%	396	23.85%	–	–	0.46
Power	4 941	6 594	0.05%	1	100.0%	46	23	0.00
hep-th	8 361	15 751	0.05%	1332	69.79%	–	–	0.29
astro-ph	16 706	121 251	0.09%	1029	88.86%	–	–	0.24

and compared it to the approximate betweenness scores as a function of the number of sample points [41]. The number of samples is a function of the number of nodes N in the network. We have evaluated three cases: $\log(|N|)$, $\sqrt{|N|}$, and $|N|$ pairs of nodes. Note that for the exact betweenness computation one needs to consider a number of node pairs quadratic in the number of nodes. Fig. 6 reports the results of our experiments. We compare the ranks of nodes according to the exact betweenness computation (x-axis) with the rank of nodes for approximate betweenness (y-axis). The ranking differences between $\log(|N|)$ and $\sqrt{|N|}$ are insignificant, with the former being 4–8 times faster than the latter. The case with N samples obtains very good rankings, but is even slightly slower than the exact betweenness computation. We conclude that $\log(|N|)$ is a good trade-off to perform a sampling-based approximation of betweenness, with a good initial ranking at low computation costs. However, if the approximate betweenness of a node is wrongly estimated, the importance of the node can be identified by one of the other metrics or be revealed during a later iteration with different sample points. Our results indicate that $\log(|N|)$ sample pairs are often sufficient to find interesting sub-attacks.

In the following, we analyze the sensitivity of the computed R value given a number of sample points for partial attack traces. We evaluate four datasets and six static network metrics. The results are visualized in Fig. 7. After an initial phase of steep descent, a plateau is reached, after which increasing the number of sample points does not yield improved results. Our experimental results suggest, that 100 sample points are sufficient to accurately estimate the robustness. The non-determinism of approximate betweenness can be observed for the network netscience. In very few cases, the R value based on approximate betweenness can be rather high, if the number of samples is too small. For the remainder of our experiments we set the number of sample points to 100. We will show below that 100 samples points are sufficient,



(a) The number of sample points is set as log(|N|).

(b) The number of sample points is set as $\sqrt{|N|}$.

(c) The number of sample points is set as N.

Fig. 6. Sensitivity analysis of the number of samples for the approximate betweenness computation, with different number of samples: log(|N|), $\sqrt{|N|}$, and N, where |N| is the number of nodes in the network. The title of each subchart lists the dataset, as well as, the speed up factor, i.e., how many times faster the approximate betweenness calculation is, compared to the exact computation. For the network netscience, we observe that the approximate betweenness has problems to distinguish the betweenness centrality of several nodes, which is probably due to the larger number of components in netscience: With a few sample pairs, the approximate betweenness algorithm leaves several nodes undiscovered.

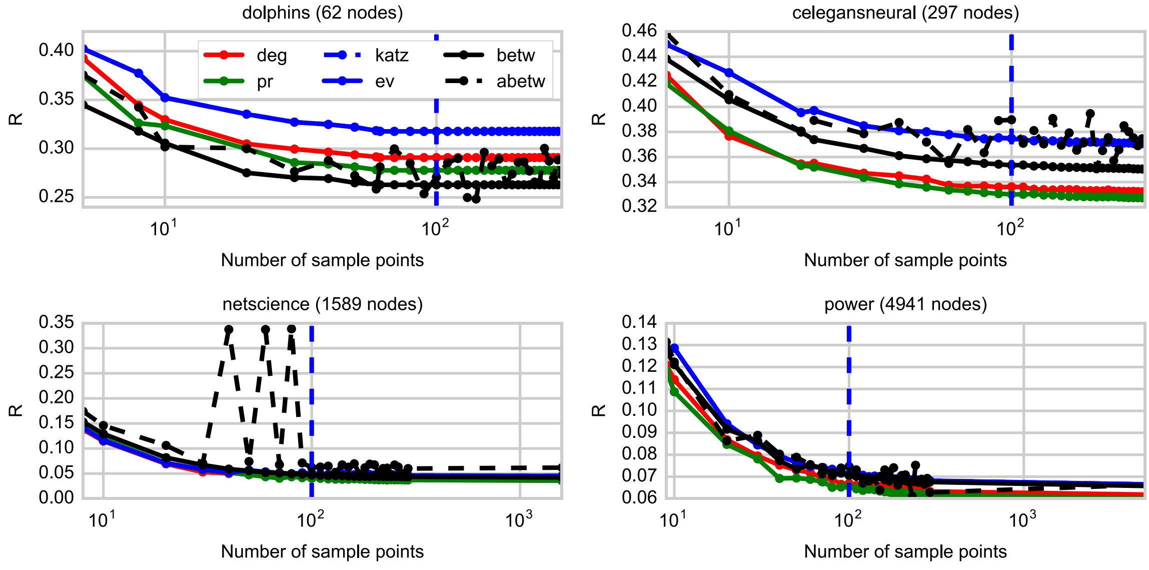


Fig. 7. Sensitivity analysis for the number of sample points for computing a partial attack trace. In the charts, we highlight the case of 100 sample points with a dashed vertical line.

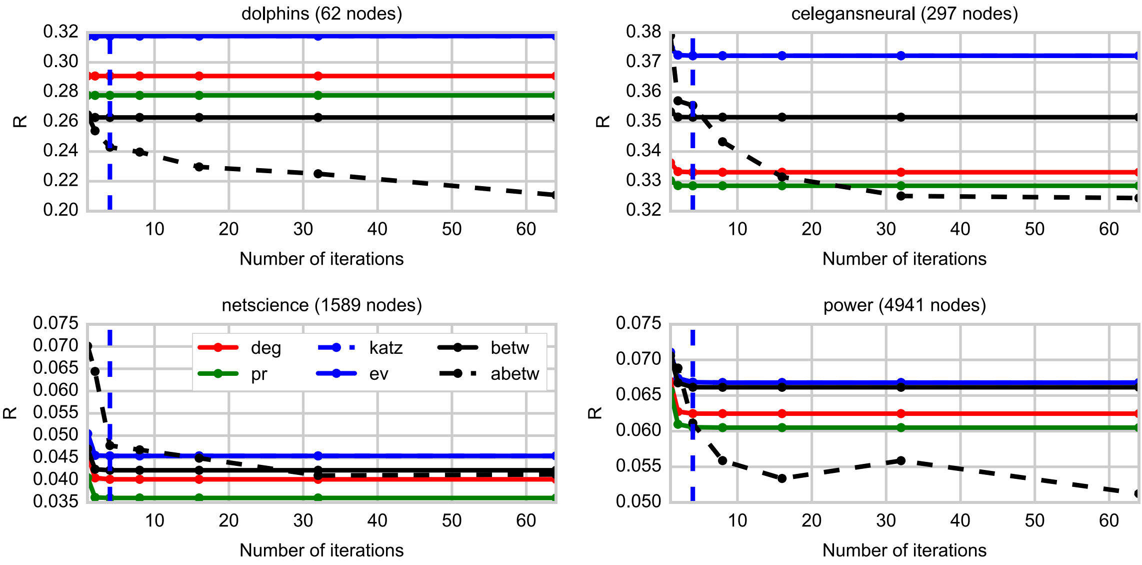


Fig. 8. Sensitivity analysis for the number of iterations. In the charts, we highlight the case of 4 iterations with a dashed vertical line.

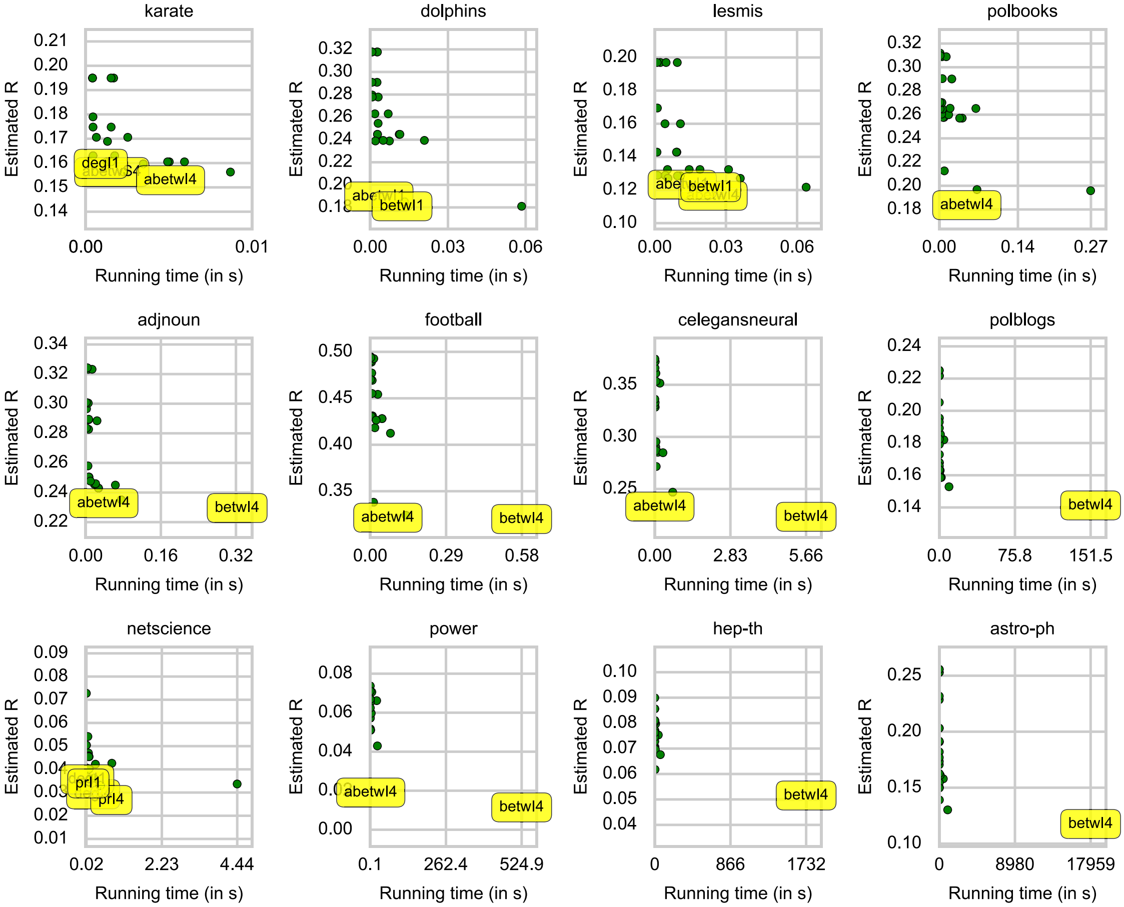


Fig. 9. Estimated *R* versus running time for QRE with single network metrics. Each green circle corresponds to one competitor. Methods belonging to the Pareto front and within 0.01 to the best obtained *R* value are labeled with their name. The label of a competitor consists of the network metric, *S*/*I* (for static/interactive) and 1 or 4 (for the number of iterations). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

particularly, as long as we do not rely on a single network metric for ranking the nodes.

The influence of the number of iterations on the estimated robustness of a network is evaluated next. Again, we evaluate four datasets and six static network metrics. Fig. 8 presents the results, with 100 sample points. For all network metrics, except from approximate betweenness, we can see that the robustness estimation is rather stable after a few number of iterations (horizontal line). Approximate betweenness, on the other hand, can still find lower *R* values, with an increasing number of iterations. Overall, we propose that for all network metrics, a small number of iterations is sufficient. For the remainder of our experiments we fix the number of iterations to either 1 (fast) or 4 (stable) for all five static network metrics. The variation of approximate betweenness will be

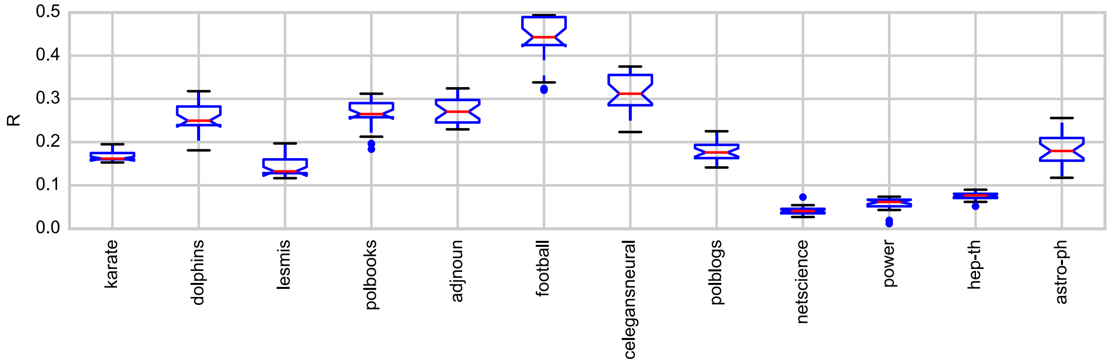


Fig. 10. Variation of robustness estimation with 24 different methods. For some datasets, e.g., football and celegansneural, the R values vary by 0.2, which is 40% of the total robustness range.

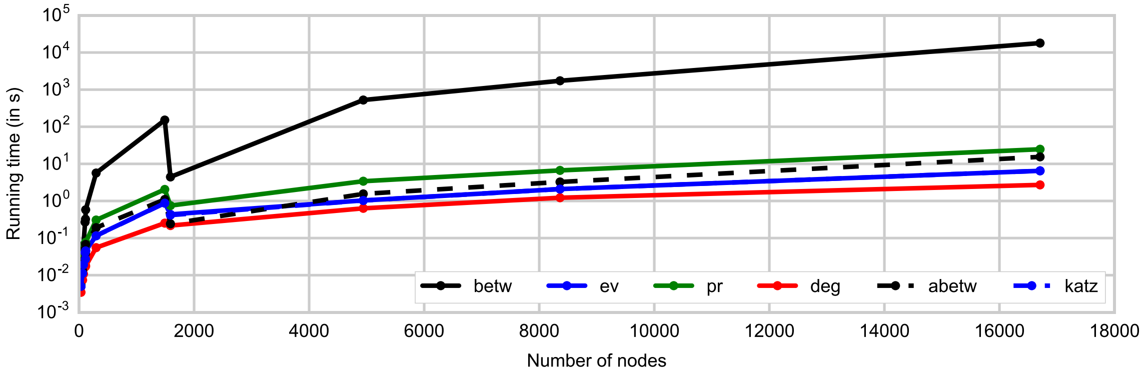


Fig. 11. Scalability of metric-based robustness estimation. The running time is presented as a function of the number of nodes.

Table 2
Comparison of robustness estimation, as computed with R, by standard network metrics (betwI, degI, evI, katzI, prI), and two selected QRE instances: ADI (approximate betweenness, degree; all interactive) and ADEKPI (approximate betweenness, degree, eigenvector, katz, pagerank; all interactive). Minimum values (=better) are highlighted in bold.

Network	betwI	degI	evI	katzI	prI	ADI	ADEKPI
Karate	0.156	0.160	0.161	0.161	0.161	**0.145**	**0.145**
Dolphins	0.181	0.239	0.245	0.245	0.239	0.176	**0.173**
Lesmis	0.122	0.129	0.132	0.132	0.127	0.102	**0.100**
Polbooks	0.197	0.260	0.258	0.258	0.265	**0.177**	0.182
Adjnoun	0.235	0.258	0.250	0.250	0.246	0.227	**0.226**
Football	0.324	0.430	0.431	0.431	0.427	**0.323**	0.335
Celegansneural	0.247	0.295	0.290	0.290	0.295	**0.226**	**0.226**
Polblogs	0.153	0.166	0.168	0.168	0.163	0.151	**0.150**
Netscience	0.043	0.036	0.040	0.040	0.034	0.017	**0.016**
Power	0.043	0.057	0.057	0.057	0.065	**0.013**	0.014
hep-th	0.068	0.077	0.077	0.077	0.075	**0.062**	**0.062**
astro-ph	**0.130**	0.192	0.203	0.203	0.155	0.138	0.138

addressed by using multiple partial attack traces, instead of single partial traces (see evaluation below).

3.3. Robustness estimations by QRE

Our sensitivity analysis in the previous section has led to the following parameters for Algorithm 1: Approximate betweenness is sampled by $\log(|N|)$ random node pairs, we compute partial attack traces with 100 sample points, and iterate the main loop in Algorithm 1 for either one or four times.

In Fig. 9, we report the results of considering the network metric separately. In total we have 24 competitors, implemented in QRE: six metrics, static/interactive, and 1/4 iterations. We can see that the best metrics are usually approximate betweenness (abetw) and exact betweenness (betw). In a few cases, degree (deg) is in the Pareto front as well, with very short running times. Overall, we can conclude that for the majority of networks, only betweenness-based metrics can accurately estimate the robustness; where for larger networks, exact betweenness takes much longer to compute than the approximate variant. In Fig. 10, we visualize the distribution of R values of the 24 competitors for each dataset as a box-plot. It can be seen that the variation of R values is rather large; even the median is rather far away from the minimum values. This highlights that one should not randomly select a network metric to compute the R values, because this often leads to a significant overestimation of network robustness.

In Fig. 11, we analyze the running times of the interactive QRE instances with 4 iterations; according to our evaluation above, these instances provide the best results. The running time of exact betweenness computation increases tremendously with the

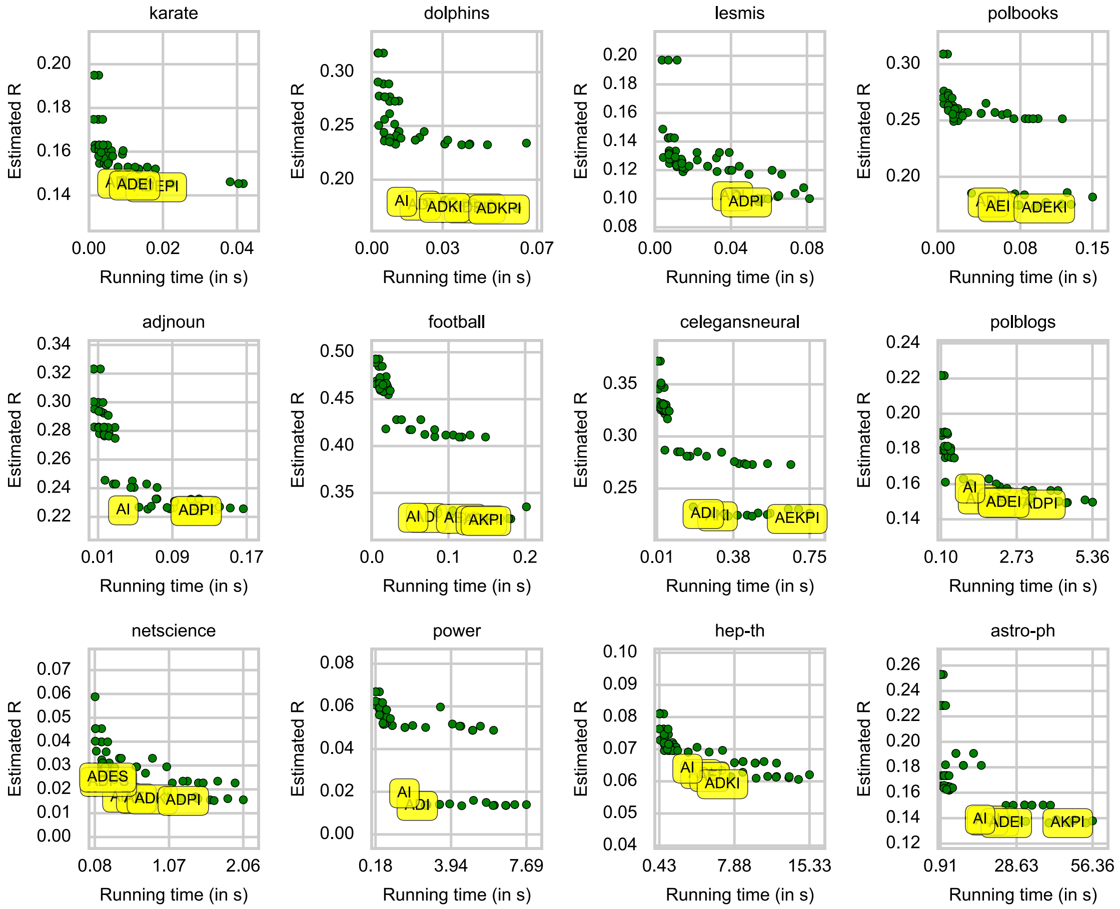


Fig. 12. Estimated *R* versus running time for QRE with multiple network metrics. Each green circle corresponds to one competitor. Methods belonging to the Pareto front and within 0.01 to the best obtained *R* value are labeled with their name. The label of a competitor consists of the network metric abbreviations. The configuration ADI is among the best Pareto method in 11 out of 12 cases. All other configuration appear significantly less often (at most 7 times). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

number of nodes in the network. The execution times of the other five metrics (including approximate betweenness) grow significantly slower. The kink around 1500 nodes is caused by the dataset polblogs, which has a significantly higher number of links than the other datasets with similar number of nodes, e.g., netscience. Therefore, the computations for polblogs always take longer.

In Fig. 12, we further evaluate QRE with multiple network metrics. For these experiments, we have enumerated all possible subsets of $\{\text{abetw} = A, \text{betw} = B, \text{deg} = D, \text{ev} = E, \text{katz} = K, \text{pagerank} = P\}$ and four iterations, such that ADEI denotes the sequential application of metrics. Again, we analyze the Pareto front. We can conclude that for each network, only a few instances of QRE belong to the best competitors. Particularly often, we find that ADI (approximate betweenness, degree; interactive) is among the best results, and the difference between ADI and the best result per network is usually small, i.e., less than 0.005.

4. Discussion

In Table 2, we summarize the robustness estimations for five standard (single) interactive network metrics using our methodology, together with the cases of ADI (approximate betweenness, degree; all interactive) and ADEKPI (approximate betweenness, degree, eigenvector, katz, pagerank; all interactive). In all but one case, the two ADI/ADEKPI identify the minimum *R* values for the networks. It should be noted that the difference between ADI and ADEKPI is usually less than 0.005 (except for polblogs). Among single network metrics, interactive betweenness performs best as an attacking strategy. The difference to other network metrics is significant (up to 0.1 smaller *R* values). However, the major limitation of interactive betweenness is its high computation time. In Table 3, we compare the computation times for all approaches. We can clearly see the computational requirements for interactive betweenness. Moreover, the interactive degree is usually the fastest method to attack the network, but the obtained *R* might not be accurate. ADI and ADEKPI provide very good robustness estimations at acceptable running times.

In synthesis, we have proposed a computationally efficient framework, Quick Robustness Estimation (QRE), for estimating the robustness of networks. We show that instances of QRE can be used to obtain *R* values smaller than the state-of-the-art with subquadratic computation times. The results of our study show that efficient, yet accurate robustness estimation is possible even for very large networks. We believe that this work contributes to a better understanding of real-world network robustness in face of big data. We envision that our technique can be extended to analyze robustness of networks of networks [46–48]. Moreover, recent progress on exploitation of artificial intelligence search techniques for network resilience analysis promotes to go beyond the use of network metrics only [49–51].

Acknowledgment

This study is supported by the National Natural Science Foundation of China (Grant Nos. 61650110516 and 61601013).

Table 3
Comparison of robustness estimation running time (in s) by standard network metrics (betwI, degI, evI, katzI, prI), and two selected QRE instances: ADI (approximate betweenness, degree; all interactive) and ADEKPI (approximate betweenness, degree, eigenvector, katz, pagerank; all interactive). Minimum values (=better) are highlighted in bold.

Network	betwI	degI	evI	katzI	prI	ADI	ADEKPI
Karate	0.002	**0.001**	0.001	0.001	0.002	0.008	0.022
Dolphins	0.012	**0.002**	0.003	0.003	0.005	0.021	0.062
Lesmis	0.024	0.01	**0.005**	0.015	0.006	0.041	0.081
Polbooks	0.067	**0.004**	0.007	0.008	0.019	0.049	0.15
Adjnoun	0.076	**0.005**	0.006	0.007	0.021	0.054	0.166
Football	0.136	**0.004**	0.008	0.008	0.023	0.071	0.202
Celegansneural	0.672	**0.012**	0.032	0.032	0.046	0.234	0.749
Polblogs	9.741	**0.065**	0.102	0.102	0.222	1.402	5.364
Netscience	0.764	**0.043**	0.064	0.066	0.076	0.454	2.061
Power	24.704	**0.145**	0.178	0.18	0.462	2.258	7.69
hep-th	64.099	**0.233**	0.324	0.327	0.703	4.561	15.331
astro-ph	997.342	**0.559**	0.988	0.988	2.777	18.065	56.356

References

[1] Luis A.N. Amaral, Julio M. Ottino, Complex networks, Eur. Phys. J. B 38 (2) (2004) 147–162.

[2] Luciano da Fontoura Costa, Osvaldo N. Oliveira Jr., Gonzalo Travieso, Francisco Aparecido Rodrigues, Paulino Ribeiro Villas Boas, Lucas Antiqueira, Matheus Palhares Viana, Luis Enrique Correa Rocha, Analyzing and modeling real-world phenomena with complex networks: a survey of applications, Adv. Phys. 60 (3) (2011) 329–412.

[3] Massimiliano Zanin, Fabrizio Lillo, Modelling the air transport with complex networks: A short review, Eur. Phys. J. Spec. Top. 215 (1) (2013) 5–21.

[4] Xiaoqian Sun, Sebastian Wandelt, Florian Linke, Temporal evolution analysis of the European air transportation system: air navigation route network and airport network, Transportm. B: Transport Dyn. 3 (2) (2015) 153–168.

[5] Xiaoqian Sun, Sebastian Wandelt, Network similarity analysis of air navigation route systems, Transp. Res. Part E: Logist. Transp. Rev. 70 (0) (2014) 416–434.

[6] Réka Albert, István Albert, Gary L. Nakarado, Structural vulnerability of the north American power grid, Phys. Rev. E 69 (2) (2004) 025103.

[7] Lucas Cuadra, Sancho Salcedo-Sanz, Javier Del Ser, Silvia Jiménez-Fernández, Zong Woo Geem, A critical review of robustness in power grids using complex networks concepts, Energies 8 (9) (2015) 9211–9265.

[8] Soon-Hyung Yook, Hawoong Jeong, Albert-László Barabási, Modeling the Internet's large-scale topology, Proc. Natl. Acad. Sci. 99 (21) (2002) 13382–13386.

[9] Michael Boss, Helmut Elsinger, Martin Summer, Stefan Thurner, Network topology of the interbank market, Quant. Finance 4 (6) (2004) 677–684.

[10] Luıs A.N. Amaral, Antonio Scala, Marc Barthélemy, H. Eugene Stanley, Classes of small-world networks, Proc. Natl. Acad. Sci. 97 (21) (2000) 11149–11152.

[11] Réka Albert, Hawoong Jeong, Albert-László Barabási, Error and attack tolerance of complex networks, Nature 406 (2000) 378–382.

[12] Min Ouyang, Critical location identification and vulnerability analysis of interdependent infrastructure systems under spatially localized attacks, Reliab. Eng. Syst. Saf. 154 (2016) 106–116.

[13] Paolo Crucitti, Vito Latora, Massimo Marchiori, Model for cascading failures in complex networks, Phys. Rev. E 69 (4) (2004) 045104.

[14] Peter Brooker, Fear in a handful of dust: aviation and the icelandic volcano, Significance 7 (3) (2010) 112–115.

[15] J. Ash, D. Newth, Optimizing complex networks for resilience against cascading failure, Physica A 380 (2007) 673–683.

[16] Steven H. Strogatz, Exploring complex networks, Nature 410 (6825) (2001) 268–276.

[17] Yusoon Kim, Yi-Su Chen, Kevin Linderman, Supply network disruption and resilience: A network structural perspective, J. Oper. Manage. 33 (2015) 43–59.

[18] Jianxi Gao, Xueming Liu, Daqing Li, Shlomo Havlin, Recent progress on the resilience of complex networks, Energies 8 (10) (2015) 12187.

[19] Min Ouyang, Lijing Zhao, Liu Hong, Zhezhe Pan, Comparisons of complex network based models and real train flow model to analyze Chinese railway vulnerability, Reliab. Eng. Syst. Saf. 123 (2014) 38–46.

[20] Dawei Zhao, Lianhai Wang, Yong-feng Zhi, Jun Zhang, Zhen Wang, The robustness of multiplex networks under layer node-based attack, Sci. Rep. 6 (24304) (2016).

[21] Christian M. Schneider, AndréA. Moreira, José S. Andrade, Shlomo Havlin, Hans J. Herrmann, Mitigation of malicious attacks on networks, Proc. Natl. Acad. Sci. 108 (10) (2011) 3838–3841.

[22] Flaviano Morone, Hernán A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (2015) 65–68.

[23] Prabha Kundur, John Paserba, Venkat Ajjarapu, Göran Andersson, Anjan Bose, Claudio Canizares, Nikos Hatziargyriou, David Hill, Alex Stankovic, Carson Taylor, et al., Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions, IEEE Trans. Power Syst. 19 (3) (2004) 1387–1401.

[24] Mathaios Panteli, Pierluigi Mancarella, Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies, Electr. Power Syst. Res. 127 (2015) 259–270.

[25] Min Ouyang, Zhezhe Pan, Liu Hong, Lijing Zhao, Correlation analysis of different vulnerability metrics on power grids, Physica A 396 (2014) 204–211.

[26] Ettore Bompard, Tao Huang, Yingjun Wu, Mihai Cremenescu, Classification and trend analysis of threats origins to the security of power systems, Int. J. Electr. Power Energy Syst. 50 (2013) 50–64.

[27] Min Ouyang, Review on modeling and simulation of interdependent critical infrastructure systems, Reliab. Eng. Syst. Saf. 121 (2014) 43–60.

[28] Stefano Boccaletti, Ginestra Bianconi, Regino Criado, Charo I. Del Genio, Jesús Gómez-Gardenes, Miguel Romance, Irene Sendina-Nadal, Zhen Wang, Massimiliano Zanin, The structure and dynamics of multilayer networks, Phys. Rep. 544 (1) (2014) 1–122.

[29] Irene Eusgeld, Cen Nan, Sven Dietz, System-of-systems approach for interdependent critical infrastructures, Reliab. Eng. Syst. Saf. 96 (6) (2011) 679–686.

[30] Reuven Cohen, Keren Erez, Daniel Ben-Avraham, Shlomo Havlin, Resilience of the Internet to random breakdowns, Phys. Rev. Lett. 85 (21) (2000) 4626.

[31] Mark E.J. Newman, Networks-An Introduction, Oxford University Press, 2010.

[32] Reuven Cohen, Shlomo Havlin, Complex Networks: Structure, Robustness and Function, Cambridge University Press, 2010.

[33] Duncan S. Callaway, Mark E.J. Newman, Steven H. Strogatz, Duncan J. Watts, Network robustness and fragility: Percolation on random graphs, Phys. Rev. Lett. 85 (25) (2000) 5468.

[34] Xuqing Huang, Jianxi Gao, Sergey V. Buldyrev, Shlomo Havlin, H. Eugene Stanley, Robustness of interdependent networks under targeted attack, Phys. Rev. E 83 (6) (2011) 065101.

[35] Mark E.J. Newman, The structure and function of complex networks, SIAM Rev. 45 (2) (2003) 167–256.

[36] Linton C. Freeman, Centrality in social networks conceptual clarification, Social Networks 1 (3) (1978) 215–239.

[37] Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd, The pagerank citation ranking: bringing order to the web. Technical Report, 1999.

[38] Petter Holme, Beom Jun Kim, Chang No Yoon, Seung Kee Han, Attack vulnerability of complex networks, Phys. Rev. E 65 (5) (2002) 056109.

[39] Xiaoqian Sun, Sebastian Wandelt, Florian Linke, On the topology of air navigation route systems, in: Proceedings of the Institution of Civil Engineers-Transport, Vol. 170, Thomas Telford Ltd., 2016, pp. 46–59.

[40] Sebastian Wandelt, Xiaoqian Sun, Evolution of the international air transportation country network from 2002 to 2013, Transp. Res. Part E: Logist. Transp. Rev. 82 (2015) 55–78.

[41] Robert Geisberger, Peter Sanders, Dominik Schultes, Better approximation of betweenness centrality, in: Proceedings of the Meeting on Algorithm Engineering & Expermiments, Society for Industrial and Applied Mathematics, 2008, pp. 90–100.

[42] Mark E.J. Newman, Scientific collaboration networks. I. Network construction and fundamental results, Phys. Rev. E 64 (1) (2001) 016131.

[43] Duncan J. Watts, Steven H. Strogatz, Collective dynamics of ?small-world?networks, Nature 393 (6684) (1998) 440–442.

[44] David Eppstein, Emma S. Spiro, The h-index of a graph and its application to dynamic subgraph statistics, in: Workshop on Algorithms and Data Structures, Springer, 2009, pp. 278–289.

[45] Christophe Schülke, Federico Ricci-Tersenghi, Multiple phases in modularity-based community detection, Phys. Rev. E 92 (4) (2015) 042804.

[46] Jianxi Gao, Daqing Li, Shlomo Havlin, From a single network to a network of networks, Natl. Sci. Rev. 1 (3) (2014) 346–356.

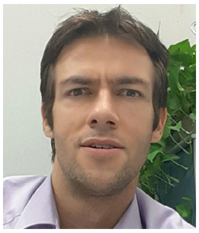
[47] Massimiliano Zanin, Can we neglect the multi-layer structure of functional networks? Physica A 430 (2015) 184–192.

[48] Dawei Zhao, Zhen Wang, Gaoxi Xiao, Bo Gao, Lianhai Wang, The robustness of interdependent networks under the interplay between cascading failures and virus propagation, Europhys. Lett. EPL 115 (5) (2016) 58004.

[49] Sebastian Wandelt, Xiaoqian Sun, Xianbin Cao, Computationally efficient attack design for robustness analysis of air transportation networks, Transportmetrica A: Transp. Sci. 11 (10) (2015) 939–966.

[50] Ye Deng, Jun Wu, Yue-Jin Tan, Optimal attack strategy of complex networks based on tabu search, Physica A 442 (2016) 74–81.

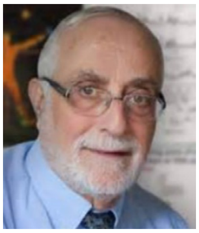
[51] Manuel Lozano, Carlos Garcia-Martinez, Francisco J. Rodriguez, Humberto M. Trujillo, Optimizing network attacks by artificial bee colony, Inform. Sci. 377 (2017) 30–50.



Sebastian Wandelt is a professor at Beihang University. Before joining Beihang, he was a postdoc in the group Knowledge Management in Bioinformatics at Humboldt-University of Berlin. He obtained a Ph.D. degree on Semantic Web reasoning techniques at Hamburg University of Technology in 2011. His work is broadly in the intersection between intelligent transportation systems and computer science. In addition, he is interested in scalable techniques for storing and indexing data. His research is published in venues such as IEEE Transactions on Intelligent Transportation Systems, Transportation Research Part E, Transportmetrica A/B, PVLDB, IEEE TCBB, and SIGMOD Record.

Xiaoqian Sun is an associate professor with the School of Electronic and Information Engineering at Beihang University. She obtained her Ph.D. in Aerospace Engineering from German Aerospace Center/Hamburg University of Technology in 2012. From 2012 to 2015, she worked as a postdoc in German Aerospace Center, Hamburg. She received her M.S. and B.S. degrees in Aerospace Engineering from Northwestern Polytechnical University in Xi'an, China. Dr. Sun has several publications in Transportation Research Part E, IEEE Transactions on Intelligent Transportation Systems, Transportmetrica A/B, and Journal of Aircraft. Her research interests mainly include air transportation networks and multi-criteria assessment.

Massimiliano Zanin is a Principal Researcher at Innaxis, graduated in Aeronautical Management at the Universidad Autónoma de Madrid and got his Ph.D. from Universidade Nova de Lisboa. With more than 90 published peer-reviewed contributions in international conferences and journals, he has vast experience in complex systems and data mining research. His main topics of interest are Complex Networks, Data Science and their application to several real-world problems, including air transport and biomedicine. He is a member of the editorial team of Nature Scientific Reports, the European Journal of Social Behaviour, PeerJ and PeerJ Computer Science.



Shlomo Havlin graduated from Bar-Ilan and Tel-Aviv Universities with Highest Distinction. He obtained an academic position at Bar-Ilan University in 1972 where he became a full Professor in 1984. During 1978–1979 he was a Royal Society Visiting Fellow at the University of Edinburgh. In 1984 he became the Chair of the Physics Department at Bar-Ilan University until 1988. During 1983–1984 and 1989–1991, Professor Havlin was a Visiting Scientist at NIH where he collaborated much with Drs. George Weiss, Ralph Nossal and other members of NIH. During 1984–1985 and 1991–1992 he was a Visiting Professor at Boston University, where he collaborated with Professor H. Eugene Stanley. He served as President of the Israel Physical Society (1996–1999), Dean of Faculty of Exact Sciences (1999–2001), Chairman, Department of Physics (1984–1988). Professor Havlin had more than 100 graduate students and postdocs, and collaborated with more than 300 scientists around the globe. He published more than 600 articles and 11 books.