

(c) 2020 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Using Metrics Suites to Improve the Measurement of Privacy in Graphs

Yuchen Zhao and Isabel Wagner, *Senior Member, IEEE*

Abstract—Social graphs are widely used in research (e.g., epidemiology) and business (e.g., recommender systems). However, sharing these graphs poses privacy risks because they contain sensitive information about individuals. Graph anonymization techniques aim to protect individual users in a graph, while graph de-anonymization aims to re-identify users. The effectiveness of anonymization and de-anonymization algorithms is usually evaluated with privacy metrics. However, it is unclear how strong existing privacy metrics are when they are used in graph privacy. In this paper, we study 26 privacy metrics for graph anonymization and de-anonymization and evaluate their strength in terms of three criteria: *monotonicity* indicates whether the metric indicates lower privacy for stronger adversaries; for within-scenario comparisons, *evenness* indicates whether metric values are spread evenly; and for between-scenario comparisons, *shared value range* indicates whether metrics use a consistent value range across scenarios. Our extensive experiments indicate that no single metric fulfills all three criteria perfectly. We therefore use methods from multi-criteria decision analysis to aggregate multiple metrics in a metrics suite, and we show that these metrics suites improve monotonicity compared to the best individual metric. This important result enables more monotonic, and thus more accurate, evaluations of new graph anonymization and de-anonymization algorithms.

Index Terms—graph anonymization, graph de-anonymization, privacy, privacy metrics, monotonicity, metrics suites

1 INTRODUCTION

The usage of Internet-based communications systems such as email or social networks leaves traces that can be collected and stored in graph form. In these graphs, nodes represent users and edges represent relationships between users. Many graph data sets have already been published for scientific or commercial use [16], [17]. Graph data can help us understand social networks [15] and improve recommendations [19], but can also harm user privacy because of sensitive information revealed by relationships.

To protect privacy, graphs can be anonymized by removing node identifiers and by changing the graph structure. As a result, the nodes in the anonymized graph cannot easily be mapped to their original identifiers if the adversary does not possess additional knowledge. However, a common assumption is that adversaries know about an auxiliary graph with similar structure as well as the correct mapping for a small number of nodes. Finding new methods for anonymization and de-anonymization is an active research area [9], and researchers usually use privacy metrics to evaluate the effectiveness of their new methods.

The most commonly used metric in graph privacy is the *adversary's success rate*, which gives the percentage of correctly re-identified nodes [10], [23], [27]. However, even though we show in this paper that the adversary's success rate is indeed a good metric in many scenarios, it has two important shortcomings: First, it does not reveal much detail about the privacy of individual nodes because it measures

privacy on a per-graph level. Second, its common definition favors de-anonymization algorithms that primarily rely on local, instead of global, graph properties.

To find a better metric for graph privacy, we analyze privacy metrics proposed in other fields [36] in terms of three criteria: monotonicity, evenness, and shared value range [42]. Monotonicity requires that privacy metrics indicate lower privacy as the adversary's strength increases. Evenness requires that the metric values are spread evenly over their value range, which improves within-scenario comparisons. Shared value range requires that metrics use a common value range even when they are applied to different datasets, anonymization, or de-anonymization algorithms. This improves between-scenario comparisons.

In this paper, we make the following contributions in the area of privacy measurement:

- We propose a framework for the evaluation of privacy metrics for graph privacy based on our previous methodology [34], [42]. In contrast to the previous work, we tailor the method to graph privacy by using de-anonymization algorithms to model the adversary, defining adversary strength levels, and adding an anonymization step to the definition of user data.
- We conduct extensive experiments to analyze the strength of 26 privacy metrics using 11 graph datasets, 11 anonymization algorithms, and 6 de-anonymization algorithms and find that no single metric excels in all three criteria.
- We find that several popular metrics are not monotonic in graph privacy, including entropy and the anonymity set size, and give a detailed analysis why this is the case. This finding is in contrast to results for other domains, such as vehicular network

• Y. Zhao is with the School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, United Kingdom.
• I. Wagner (corresponding author) is with the Cyber Security Centre, De Montfort University, Leicester, LE1 9BH, United Kingdom.
E-mail: yuchen.zhao@soton.ac.uk, isabel.wagner@dmu.ac.uk

privacy [35].

- To the best of our knowledge, we are the first to use methods from multi-criteria decision analysis to integrate several metrics into a metrics suite. We evaluate four concrete metrics suites, synthesized using the Weighted Product Model, and show that our metrics suites have higher monotonicity than the best individual metric.

Our findings are important because our synthesis of metrics suites shows a convenient way how the strengths of multiple privacy metrics can be combined to improve the overall measurement of privacy.

2 BACKGROUND

2.1 Graph Anonymization and De-anonymization

Structural graph de-anonymization algorithms and corresponding anonymization algorithms have been an active research area since the mid-2000s [2], when it was discovered that simply removing identifiers from nodes is not sufficient to prevent node re-identification. Since then, many anonymization and de-anonymization algorithms have been proposed [9].

In this paper, we use existing algorithms and implementations [4], [5], [10], [39] to evaluate the strength of privacy metrics for graph anonymization and de-anonymization. We give more detail about the algorithms we used in Sections 3.2 and 3.3.

2.2 Privacy Metrics for Graph Privacy

The most commonly used privacy metrics in graph anonymization and de-anonymization are the number of re-identified nodes [22] and the adversary’s success rate [10], [23], [27]. These metrics quantify privacy as the actual privacy breach that an adversary causes.

In the wider privacy research area, other classes of privacy metrics have been proposed [36], for example measuring the adversary’s uncertainty, information gain, or error. For example, information-theoretic metrics such as entropy quantify privacy as the uncertainty that the adversary faces when re-identifying nodes. These metrics have shown good strength in other fields such as vehicular networks [42], but it is unknown whether they are suitable for graph privacy. In this paper, we examine the suitability and strength of a wide range of privacy metrics in graph privacy.

De-anonymizability quantification focuses on quantifying structural properties of graph pairs, such as the edge difference, to study the maximum number of nodes that could be de-anonymized, given only structural graph information for a graph and a partially overlapping auxiliary graph [8], [24]. However, de-anonymizability quantification does not consider the interplay between anonymization and de-anonymization algorithms and has limited applicability in concrete practical scenarios due to its focus on theoretical limits for abstract graph models [9].

2.3 Utility Metrics for Graph Privacy

There is a trade-off between the privacy achieved by graph anonymization algorithms and the utility of the anonymized

graph. Utility is often measured according to how well the anonymized graph preserves the characteristics of the original graph, such as node degrees and path lengths [10]. In this paper, we focus on the quality of privacy metrics, and thus utility and utility measures are complementary to our study: How much utility a graph anonymization algorithm provides, or how this utility is measured, is independent of how its privacy properties are measured. We note that utility measures may be evaluated with similar methods as ours, but consider this to be out of scope for the current paper.

2.4 Criteria for Privacy Metrics

Several authors have proposed criteria that good privacy metrics should fulfill. For example, metrics should show the adversary’s chances of success [1], show the level of privacy and the potential for privacy violations [3], measure the amount of resources needed by the adversary [30], and integrate measurements for different aspects of privacy [28].

These criteria focus on properties that are desirable specifically in *privacy* measurement, such as consideration of the adversary. However, they do not focus on essential properties for measurement in general, such as consistency. In addition, these criteria do not allow to judge whether a privacy metric can discern differences in privacy levels between different users in a scenario, or between different scenarios.

To overcome these shortcomings, we use three criteria that we have proposed in prior work: monotonicity, evenness, and shared value range. Taken together, these three criteria allow to select privacy metrics that result in consistent measurements, and from among those, to select privacy metrics that support the intended type of evaluation.

2.4.1 Monotonicity

Monotonicity indicates the consistency of measurement by requiring that higher amounts of privacy in a scenario are reflected in higher privacy levels. We have previously proposed to evaluate the monotonicity of privacy metrics with respect to the adversary, i.e. that they should show lower privacy levels for stronger adversaries [34], [42]. For example, we expect that, with increasing adversary strength, the values of the adversary’s success rate (as a lower-better metric) decrease, and that the values of entropy (as a higher-better metric) increase.

Monotonicity is a key property of privacy metrics because a non-monotonic metric can make privacy-enhancing technologies, such as anonymization algorithms, appear stronger than they actually are. For example, non-monotonic metrics may indicate the same privacy level for strong and weak adversaries, or for strong and weak PETs. If these weak PETs are then deployed in practice, the use of non-monotonic metrics can lead to real-world privacy violations.

We use the algorithm originally proposed in [34] to compute monotonicity scores for a privacy metric M . The algorithm uses two statistical tests (t-test and rank-sum test) to compare the mean values of the privacy metric for each pair of adjacent adversary strengths in a scenario c , where a scenario is given by the combination of anonymizer, de-anonymizer and graph. For each statistical test, if the test indicates significance ($p < 0.05$) as well as a change in

the expected direction, the algorithm increases the metrics' monotonicity score by 1. If the test indicates a change in the wrong direction, the algorithm subtracts 1 from the monotonicity score. If the changes in metric values change direction, e.g. increasing for one pair and decreasing for the next, the monotonicity score is reduced by 2 because such a peak may indicate the same privacy levels for both strong and weak adversaries and is thus undesirable. A metric's final monotonicity score is the average of the scores for the two statistical tests, normalized to $[0, 1]$.

We formalize the notion of monotonicity specifically for graph privacy in Section 3.5.

2.4.2 Evenness and Shared Value Range

In addition to monotonicity, evenness and shared value range are desirable criteria for good privacy metrics, especially if the metrics will be used for within-scenario comparisons and between-scenario comparisons [42].

Within-scenario comparisons compare the privacy levels of different nodes within the same graph. Ideally, the values of metrics for within-scenario comparisons should be spread evenly over the entire value range (*evenness* E). We compute the evenness of a metric's value range based on the Cramér-von Mises criterion, which measures the goodness of fit between the uniform distribution $F(x_{ic})$ and the normalized metric values x_{ic} for all adversary strengths in a scenario c . We normalize the Cramér-von Mises criterion by the number of metric values n to offset the influence of the number of samples.

$$E = \frac{1}{n} \left(\frac{1}{12n} + \sum_{i=1}^n \left[\frac{2i-1}{2n} - F(x_{ic}) \right]^2 \right) \quad (1)$$

Between-scenario comparisons compare privacy levels between different graphs, and ideal metrics use the same value range regardless of the graph's characteristics (*shared value range* R). We formalize this notion by computing the portion of the global value range for each metric that is used when the metric is applied to a specific graph.

$$R = \frac{\max(x_c) - \min(x_c)}{\max(x) - \min(x)} \quad (2)$$

3 METHODOLOGY

We evaluate the strength of privacy metrics for social graphs in terms of monotonicity, evenness, and shared value range. In this section, we explain our methodology, which we have adapted from our prior work [34], [42], [43]. Our method follows six steps, as illustrated in Figure 1:

- 1) Import an existing graph data set (Section 3.1)
- 2) Anonymize the input graph with different anonymization algorithms (Section 3.2)
- 3) Subset the input graph to create an auxiliary graph and define seed mappings as prior information given to the adversary (Section 3.3)
- 4) Apply different de-anonymization algorithms to the anonymized graphs, controlling the adversary's strength using the amount of prior information given to the adversary (Section 3.3)
- 5) Compute the values of privacy metrics based on the output of the de-anonymizer (Section 3.4)

- 6) Analyze the strength of each privacy metric according to the criteria of monotonicity (Sections 2.4.1 and 3.5), evenness, and shared value range (Section 2.4.2)

3.1 Graph Datasets

We used eleven graph datasets, all available from Konect [16], to evaluate the strength of privacy metrics. Table 1 summarizes the datasets and highlights some of the graph statistics commonly used to characterize graphs. Our selection represents a diverse sample of graphs with respect to 19 graph statistics available in Konect.

3.2 Anonymizing Algorithms

An anonymizer A computes an anonymized graph based on the original graph and some parameters p_A , i.e. $A : G_{\text{orig}} \times p_A \rightarrow G_{\text{anon}}$. Graph anonymization algorithms modify the nodes or edges of a graph to prevent the individual nodes from being identified. The simplest method is to remove the original identifiers of all nodes while leaving the graph structure intact. This method has been shown to be susceptible to de-anonymization attacks [2], [22] and we use it to represent the baseline for non-anonymized graphs.

In addition, we use ten anonymization algorithms that change the graph structure: Switch [41], Random Walk (RW) [20], k-Degree Anonymity (k-DA) [18], Univariate Micro-Aggregation for Graph Anonymization (UMGA) [4], Graph k-Anonymization (GraphAnon) [5], k-Symmetry Anonymity (k-SA) [38], dk-series based Differential Privacy (dk-DP) [26], Hierarchical Random Graph based Differential Privacy (HRG-DP) [39], Weighted Privacy Integrated Queries based Differential Privacy (WPINQ-DP) [25], and Bounded t-Means (t-Means) [31]. All algorithms operate on a graph $G_{\text{orig}} = (V_{\text{orig}}, E_{\text{orig}})$, where V_{orig} is the set of nodes (vertices) and E_{orig} is the set of edges. We use $G_{\text{anon}} = (V_{\text{anon}}, E_{\text{anon}})$ to denote the anonymized graph. We note that for algorithms that synthesize graphs, such as the DP-based algorithms, we restrict our study to those that generate an isomorphic vertex set; algorithms that generate other synthetic graphs are out of scope.

Switch [41] is a randomization algorithm that does not guarantee a notion of privacy. *Switch* randomly chooses two edges in E_{orig} and switches them to change G_{orig} 's structure. This process is repeated $r|E_{\text{orig}}|$ times, where r is the portion of edges to be switched. The *random walk* algorithm [20] is designed to protect link privacy and does not hide the identities of vertices. It selects a sequence of vertices starting from v , for each vertex $v \in V_{\text{orig}}$, as a "walking path" of length t . At the end of each sequence, one edge is added between the starting vertex and the ending vertex to change the graph's structure. We have included both algorithms as additional baseline comparisons to evaluate if privacy metrics are monotonic even if the anonymization algorithm does not enforce identity privacy.

The *k-DA* algorithm [18] modifies the graph to achieve k-degree anonymity, so that for each node there are at least $k-1$ other nodes with the same degree. To achieve this, the algorithm adds edges to E_{orig} until the node degree for all nodes meets the requirement. Similarly, the *UMGA* algorithm [4] achieves k-degree anonymity through first

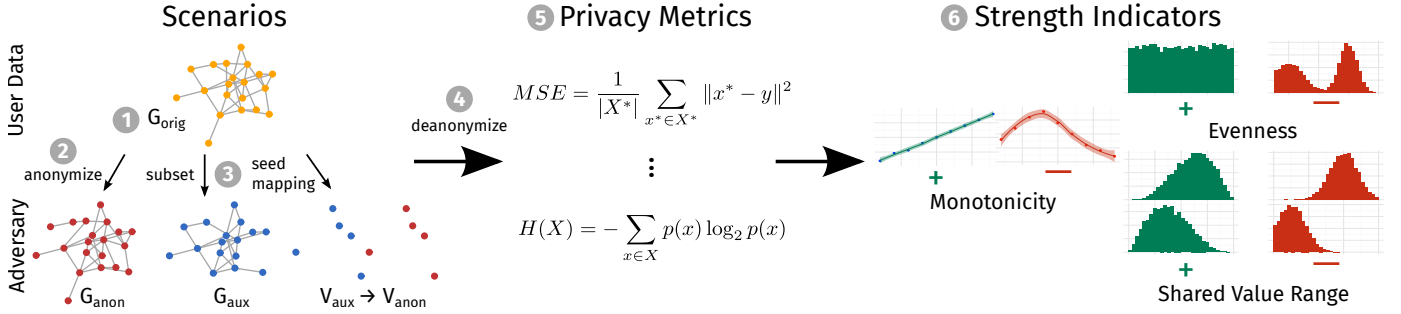


Fig. 1. Our method to evaluate the strength of privacy metrics for graph privacy, adapted from prior work [42], [43].

TABLE 1
Characteristics of the 11 graph datasets used in our experiments, showing their diversity in several key graph statistics.

Dataset	Type	Nodes	Edges	Diameter	Avg. degree	Avg. short-est path	Clustering coefficient	Gini coefficient
dblp	citation	12591	49743	10	8	4.4	0.062	0.66
cora	citation	23166	91500	20	8	5.9	0.117	0.52
arxiv	coauthors	18771	198050	14	21	4.2	0.318	0.61
dnc	communication	2029	5598	8	39	3.4	0.089	0.71
irvine	communication	1899	20296	8	63	3.1	0.057	0.65
manufacturing	communication	167	5784	5	993	1.9	0.541	0.44
caida	computer	26475	53381	17	4	3.9	0.007	0.63
elections	online contact	7118	103675	7	29	3.2	0.125	0.75
pgp	online contact	10680	24316	24	5	7.5	0.378	0.59
google	social	23628	39242	8	3	4.0	0.004	0.66
facebook ego	social	4039	88234	8	2	3.7	0.519	0.54

transforming the graph into a degree sequence and then using it to calculate a new degree sequence that satisfies k -degree anonymity. The graph will be modified based on the new sequence through deleting, adding, and switching edges. Compared with k -DA, it conducts the minimum number of edge modifications and considers edge relevance to reduce information loss in anonymization. The *GraphAnon* algorithm [5] achieves k -degree anonymity by adding dummy nodes, which is different from k -DA and *UMGA* that only modify edges. This not only changes structural relationships among nodes, but also increases the size of the space of an adversary’s inference. The k -SA algorithm [38] achieves k -anonymity through orbit copying operation to guarantee that each orbit (i.e., a partition of the graph containing automorphically equivalent nodes) and its duplicates together have at least k nodes.

The dk -DP algorithm [26] transforms the graph to a dK -series and injects noise in it. Then the perturbed dK -series is used to re-construct a differentially private graph. The anonymization level of dk -DP is controlled by the parameter ϵ and the resulting graph satisfies ϵ -differential privacy. The *HRG-DP* algorithm [39] transforms the graph to a sampled dendrogram and uses it to calculate the amount to noise to be added, based on which a differentially private graph will be constructed. Both dk -DP and *HRG-DP* achieve differential privacy through graph transformation and reconstruction, but use different transformation strategies. The *WPINQ-DP* algorithm [25] generates differentially private measurements from a graph and uses these measurements to construct the anonymized graph. The *t-Means* algorithm [31] clusters all the vertices in V_{orig} into t clusters based on their degree distances and then matches the degrees of vertices

in a cluster to the degree of their center vertex by adding or removing edges.

3.3 De-anonymizing Algorithms

Graph de-anonymization attempts to re-identify nodes in an anonymized graph. De-anonymization algorithms commonly assume that the adversary has additional knowledge in the form of an auxiliary graph $G_{aux} = (V_{aux}, E_{aux})$ and a small set of seed mappings between the nodes in V_{aux} and the nodes in V_{anon} . A de-anonymizer D computes a mapping between nodes in G_{anon} and G_{orig} , with probabilities $p_{v_{anon}, v_{orig}}$ for each candidate mapping, based on the anonymized graph G_{anon} , an auxiliary graph G_{aux} , a set of seed mappings $S = \{V_{aux}, V_{anon}\}$, and some parameters p_D .

The auxiliary graph is a sub-graph of the original graph, i.e. $V_{aux} \subset V_{orig}$ and $E_{aux} \subset E_{orig}$, and the adversary knows the identifiers of nodes in the auxiliary graph. In our experiments, we use the size of the overlap of the auxiliary graph with the original graph $|G_{orig} \cap G_{aux}|$ as the first way to control the adversary’s strength: a larger overlap results in a stronger adversary.

The set of seed mappings contains some relationships between nodes in the anonymized graph G_{anon} and the auxiliary graph G_{aux} that the adversary is assumed to know. This set of seed mappings is typically small compared to the size of the graph, and most nodes are unmapped. We use the number of seed mappings $|S|$ as the second way to control the adversary’s strength: a larger number of mappings results in a stronger adversary.

We use ordered sequences of values for $|S|$ and $\frac{|G_{orig} \cap G_{aux}|}{|G_{orig}|}$, respectively, to define an ordered sequence of

adversaries with increasing strength, and will use subscripts s and o , for seed mappings and overlap, respectively, to indicate a particular index in these sequences.

The adversary’s goal is to use the auxiliary graph G_{aux} and the seed mappings to generate more mappings between the unmapped nodes in G_{anon} and known nodes in G_{aux} , thus breaching graph privacy. Formally, the de-anonymizer is $D_{s,o} : G_{anon} \times G_{aux,o} \times S_s \times p_D \rightarrow \{V_{anon}, V_{orig}, p_{v_{anon}, v_{orig}}\}_{s,o}$. In our experiments, we use six de-anonymization algorithms: Narayanan/Shmatikov (NS) [23], Ji/Li/Srivatsa/Beyah (JLSB) [11], Korula/Lattanzi (KL) [14], Yartseva/Grossglauser (YG) [40], Adaptive De-Anonymization (ADA) [12], and Distance Vector based de-anonymization (DV) [29]. Three algorithms (KL, NS, and YG) rely on seed mappings as a local property of the nodes to be matched, i.e. only inferring among the nodes that are one hop away from seed nodes, while the other three algorithms (ADA, DV, and JLSB) use properties of the global graph structure.

NS is an iterative algorithm. In each step, the adversary randomly chooses an unmapped node v_{anon} . If any of v_{anon} ’s neighbors have a known seed mapping to a node v_{aux} , then the set of v_{aux} ’s neighbor nodes forms the set of candidate original nodes for v_{anon} . Each candidate is scored according to its node degree, and if the node with the highest score satisfies an eccentricity criterion it is selected as the most likely original node. If the reverse mapping from G_{aux} to G_{anon} of the chosen original node is v_{anon} , the resulting mapping is added to the set of seed mappings and the process is repeated. The inference stops when the set of seed mappings stops growing.

The KL algorithm also bootstraps its inference from seed mappings using a heuristic called similarity witness. The number of similarity witnesses for a pair of known and anonymized nodes is the number of seed mapping relationships between their neighbors. In each step, the pairs with the highest number of similarity witnesses are chosen and added to the set of seed mappings.

Similarly, the YG algorithm randomly selects one unused seed mapping in each iteration and increments the scores of all its neighbor mappings (i.e. mappings between the neighbors of the known nodes and the neighbors of the anonymized node in the selected seed mapping). Once a mapping’s score reaches a threshold, it is added to the set of seed mappings.

DV uses a node’s distance vector, i.e. the distances between the node and the seed nodes, to describe the structural similarity between anonymized nodes and known nodes. The algorithm maps pairs of nodes that have the highest structural similarity scores.

JLSB uses five heuristics to calculate the structural similarity between anonymized nodes and known nodes: the node degree, neighborhood, top-K reference distance, landmark reference distance, and sampling closeness centrality. These heuristics measure both the local and global properties of each node. The algorithm summarizes these heuristics into a structural similarity score that indicates how much an anonymized node is similar to a known node, thereby finding the most likely mappings.

Similarly, the ADA algorithm calculates structural similarity scores from three centrality measurements: closeness centrality, betweenness centrality, and degree. In addition, ADA takes into account the relative distance similarity (similar to the distance vector in DV) and inheritance similarity (controlling the similarity loss over iterations).

3.4 Metrics for Graph Privacy

A privacy metric M computes the level of privacy based on the output of the de-anonymizer. Some privacy metrics also depend on the ground truth G_{orig} and parameters p_M , i.e., $M_{s,o} : D_{s,o}(\cdot) \times G_{orig} \times p_M \rightarrow \mathbb{R}$.

Based on our survey of privacy metrics [36], we selected 26 privacy metrics to evaluate in our experiments. These metrics include not only metrics that have already been used in graph privacy, but also metrics from other domains. Table 2 summarizes the metrics we used (please see [36] for details on domain, codomain, and equations for each metric). According to the taxonomy in [36], the metrics in our study fall into five categories: uncertainty, information gain/loss, error, similarity, and success.

TABLE 2

Graph privacy metrics used in our experiments. H/L: high (H) or low (L) values indicate high privacy. Per-graph metrics give one privacy value for the entire graph, the other metrics give one privacy value for each node. The *chunk* column indicates whether a metric is negatively affected by SecGraph’s chunking (see Section 5.3.2).

Cat.	Metric	H/L	per-graph	gnd. truth	chunk
Uncertainty	Anonymity set size	H	-	-	✓
	Collision entropy	H	-	-	✓
	Conditional entropy	H	-	✓	-
	Conditional privacy	H	-	✓	-
	Entropy	H	-	-	✓
	Inherent privacy	H	-	-	✓
	Max-entropy	H	-	-	✓
	Min-entropy	H	-	-	✓
	Normalized entropy	H	-	-	-
Information gain	Amount leaked information	L	✓	✓	-
	Conditional privacy loss	L	✓	✓	✓
	Information surprisal	L	-	✓	-
	Loss of anonymity	L	✓	✓	✓
	Mutual information	L	-	✓	✓
	Pearson correlation	L	-	✓	-
	Relative entropy	H	-	✓	-
Error	Absolute error	H	-	✓	-
	Incorrectness	H	-	✓	-
	Mean squared error	H	-	✓	-
	% incorrectly classified	H	✓	✓	-
Sim.	Normalized variance	H	-	✓	-
Success	Adversary’s success rate	L	✓	✓	-
	Adversary’s overall success	L	✓	✓	-
	Hiding property	H	✓	-	-
	User-specified innocence	H	✓	-	-

3.4.1 Uncertainty metrics

Uncertainty metrics measure how uncertain the adversary is about her estimate, assuming that higher uncertainty corresponds to better privacy.

For example, the *anonymity set size* for each node indicates how many other nodes the adversary cannot distinguish from this node. We approximate this notion of

indistinguishability by counting how many candidate nodes have been assigned a non-zero probability.

All other uncertainty metrics in our study are based on the information theoretic concept of entropy. *Entropy* measures the adversary’s uncertainty based on the probabilities assigned to each candidate node and indicates the number of additional bits of information an adversary needs to successfully de-anonymize a node.

Rényi entropy is a generalization of entropy that introduces the parameter α , with the entropy above using $\alpha = 1$. With increasingly larger values of α , the influence of high-probability nodes on the metric increases. For example, *min-entropy* with $\alpha = \infty$ is based only on the node for which the adversary has the highest probability, representing the worst-case privacy. In contrast, *max-entropy* with $\alpha = 0$ is based only on the number of nodes. Because max-entropy does not take into account the adversary’s probabilities, it represents the best-case privacy a user can hope for. *Collision entropy* with $\alpha = 2$ is another variant.

Normalized entropy uses max-entropy to normalize entropy to a common value range of $[0,1]$ that does not depend on the number of nodes. *Quantiles on entropy* aims to mitigate the influence of low-probability outliers on entropy, and thus computes entropy only based on a percentile of the adversary’s estimated probabilities. *Conditional entropy* describes the entropy of the true mapping, conditioned on the adversary’s estimate.

Inherent privacy, or scaled anonymity set size, is based on entropy and interpreted as the number of additional yes/no questions the adversary has to answer to de-anonymize a node correctly. *Conditional privacy* is similar to inherent privacy, measuring the privacy inherent in the true mapping, given the adversary’s estimate.

3.4.2 Information gain/loss metrics

Information gain/loss metrics focus on the amount of information that the adversary gains.

The *amount of information leaked* counts how many nodes the adversary re-identified correctly. *Pearson correlation* computes the correlation between the adversary’s estimate and true node mapping. High values indicate a positive correlation and thus mean low privacy.

The remaining metrics in this category are based on information theory. *Information surprisal* focuses on the probability the adversary assigns to the true node and can be interpreted as the amount of surprise felt by the adversary on learning the true mapping.

Mutual information indicates how much information is shared between the adversary’s estimate and the true mapping, with more shared information indicating lower privacy. *Loss of anonymity* is the maximum mutual information for any node, and thus indicates the worst-case privacy. *Conditional privacy loss* is a way of normalizing mutual information and can be interpreted as the fraction of privacy lost through the adversary’s estimate.

Relative entropy, or Kullback-Leibler divergence, measures the distance between the adversary’s estimate and the true mapping, thus indicating how far the adversary’s estimate is from the truth.

3.4.3 Error metrics

Error metrics quantify the difference between the adversary’s estimate and the true mapping of candidate nodes.

Incorrectness is the expectation of the adversary’s estimation error, where successful resp. unsuccessful identification of a node is encoded as 0 resp. 1, and the expectation is computed using the adversary’s estimated probabilities.

The *mean squared error* measures the error between the adversary’s estimated probability distribution and the true outcome. The *absolute error* measures the difference between the adversary’s probability for the true node and the adversary’s highest probability, i.e. for node the adversary believes to be the true node.

The *percentage of incorrectly classified* nodes describes the percentage of nodes that the adversary has de-anonymized incorrectly, in relation to the total number of nodes in the graph.

3.4.4 Similarity metrics

Similarity metrics focus on statistical properties that measure similarity between the adversary’s estimate and the ground truth.

Normalized variance computes the variance of the difference between the true mapping and the adversary’s estimate, normalized by the variance of the true mapping. A higher variance is thought to correlate with higher privacy.

3.4.5 Success metrics

Success metrics evaluate how likely it is that the adversary succeeds. The *adversary’s success rate*, or accuracy, indicates the percentage of nodes that the adversary identified correctly. In the variant implemented by SecGraph, this metric indicates the percentage in relation to the number of nodes that the adversary has attempted to de-anonymize, *not* to the total number of nodes in the graph.

The *adversary’s overall success rate* indicates the success rate based on the entire graph, not just on the number of de-anonymization attempts.

The *hiding property* counts the number of nodes for which the adversary’s largest probability is below a specified threshold. This metric indicates the number of nodes that the anonymizing algorithm has successfully protected from the adversary. *User-specified innocence* counts the number of nodes for which the adversary’s estimated probability for the true outcome is below a specified threshold, indicating the number of nodes that can reasonably claim that the adversary’s de-anonymization attempt was not reliable.

3.5 Monotonicity

Monotonicity is defined on a sequence of privacy metrics computed for increasing adversary strengths, for example $(M_{s_1,o}, M_{s_2,o}, \dots, M_{s_n,o})$, where $s_1 < s_2 < \dots < s_n$. Let $L = (l_1, l_2, \dots, l_n)$ be a sequence of s, o combinations where either s or o is increasing, and the other kept constant.

Then, we define monotonicity as $\text{Mono} : (M_{l_1}, M_{l_2}, \dots, M_{l_n}) \rightarrow [0, 1]$.

We use the algorithm described in Section 2.4.1 to compute monotonicity values in our experiments.

4 EXPERIMENTS

We conducted extensive experiments to evaluate monotonicity, evenness, and shared value range in a wide range of scenarios. In this section, we give details on the implementation and availability of software, the parameter settings we used, and how we controlled the statistical significance of our results.

4.1 Implementation

We have implemented our framework for the evaluation of privacy metrics in Python. Our implementation of privacy metrics follows the description in [36] and relies on Python libraries `numpy`, `scipy`, and `scikit-learn`.

For anonymization algorithms, we used open-source software including `SecGraph` [10], `k-Anonymity` simulator [13], `GraphAnon` [5], and `UMGA` [4]. For de-anonymization algorithms, we used `SecGraph` [10]. However, `SecGraph`'s implementation of de-anonymization algorithms only outputs the adversary's node mapping and success rate. To be able to compute other privacy metrics, we modified the de-anonymization methods to additionally output the ground truth and the adversary's estimated probability distribution. The probability distribution is based on the adversary's scores for candidate nodes. When the adversary tries to map a node v_{anon} to a candidate in a set of candidate nodes $\{v_{\text{aux}}^1, v_{\text{aux}}^2, \dots\}$, the random variable X describes the probabilities for each candidate in the set. The probability for each candidate node is defined as its score divided by the sum of all scores in the set. In this way, the adversary creates one probability distribution for each node she attempts to re-identify.

To automate our experiments and make use of computing resources available to us, we packaged our framework in a Docker container and used `Boinc` and `boinc2docker` to distribute the computation to around 60 PCs in student labs. Our source code is available at `CodeOcean` [37].

4.2 Parameter Settings

We reproduce parameter settings for the anonymization and de-anonymization algorithms as much as possible from [10]. However, the parameters given in the `SecGraph` paper do not always match the implementation in their software, and the software parameter choices are not documented. In these cases, we chose parameter settings that seemed reasonable to us. Table 3 summarizes our parameter settings. The last two rows show the sequence of parameters we used to increase the adversary's strength. We varied seed numbers and auxiliary ratios in independent sets of experiments and used the value given as default for the variable that was kept constant.

Our experiments are based on random inputs, for example the selection of seed mappings and the overlap of the auxiliary graph. To obtain statistically significant results, we replicated the experiments until the relative error for the metric values for each combination of dataset, anonymizer, and de-anonymizer was below 5%. 100 replications were sufficient in most cases, but in some cases we computed up to 50,000 replications.

These replications also help with two other issues. First, because the de-anonymization algorithms are based on

TABLE 3
Parameter settings used in our experiments.

Parameters for anonymizers p_A	
DP-based	$\epsilon = 1$
k-anon.-based	$k = 5$
RW	distance = 2
Switch	fraction $r = 0.05$
t-means	max size = 30
Parameters for de-anonymizers p_D	
ADA	$\theta = 0$, chunk size = 100, $\epsilon = 0.5$, weights: $w_{\text{distance}} = 0.6$, $w_{\text{structural}} = 0.2$, $w_{\text{inheritance}} = 0.2$
DV	$\theta = 0$, chunk size = 100
JLSB	$\theta = 0$, chunk size = 100, weights: $w_{\text{degree}} = 0.3$, $w_{\text{neighbor}} = 0.3$, $w_{\text{ref distance}} = 0.4$
KL	$\theta = 1$, chunk size = 100
NS	$\theta = 0.5$
YG	$\theta = 2$
Seed numbers S	5, 10, 20, 35, 50, 100 (default 50)
Auxiliary ratios	0.6, 0.7, 0.8, 0.85, 0.9, 0.95 (default 0.85)
$\frac{ G_{\text{orig}} \cap G_{\text{aux}} }{ G_{\text{orig}} }$	

heuristics, it is not guaranteed that increasing the number of available seed nodes or the overlap of the auxiliary graph will always lead to better de-anonymization performance. However, the average performance over many independent replications should be increasing with adversary strength.

Second, because some nodes in auxiliary graphs may be more valuable than others (e.g., hub nodes in a star topology), metrics may be non-monotonic in a single replication if some of the auxiliary graphs contain hub nodes but others do not [29]. Computing monotonicity over multiple independent replications minimizes this effect.

5 RESULTS ON THE MONOTONICITY OF GRAPH PRIVACY METRICS

Our experiments have yielded results for 1440 individual scenarios, i.e. all combinations of dataset, anonymizer, de-anonymizer, and adversary strength type, with one exception where `UMGA` failed to k-anonymize the manufacturing graph. Each scenario is composed of results for six adversary strength levels. In this section, we discuss our results for monotonicity and analyze the factors that influence monotonicity. Results for the additional criteria evenness and shared value range are in the next section.

5.1 Overview of Results

Figure 2 shows detailed results for three metrics in six of our 1440 scenarios. Each subfigure shows of a sequence of box plots, one for each adversary strength level, with rotated histograms indicating the distribution of metric values collected from all replications. Black horizontal lines indicate confidence intervals for the mean, italic values on top of each box indicate the mean value, and the green line at the top (resp. bottom) indicates whether higher (resp. lower) values indicate higher privacy.

According to our monotonicity criterion, we expect that boxes on the right-hand side of the plots are closer to the green line than boxes on the left, i.e. that metrics indicate higher privacy for lower adversary strengths. The first row shows the *adversary's success rate* in two scenarios (Figures 2a–2b). In both cases, the metric indicates a clear

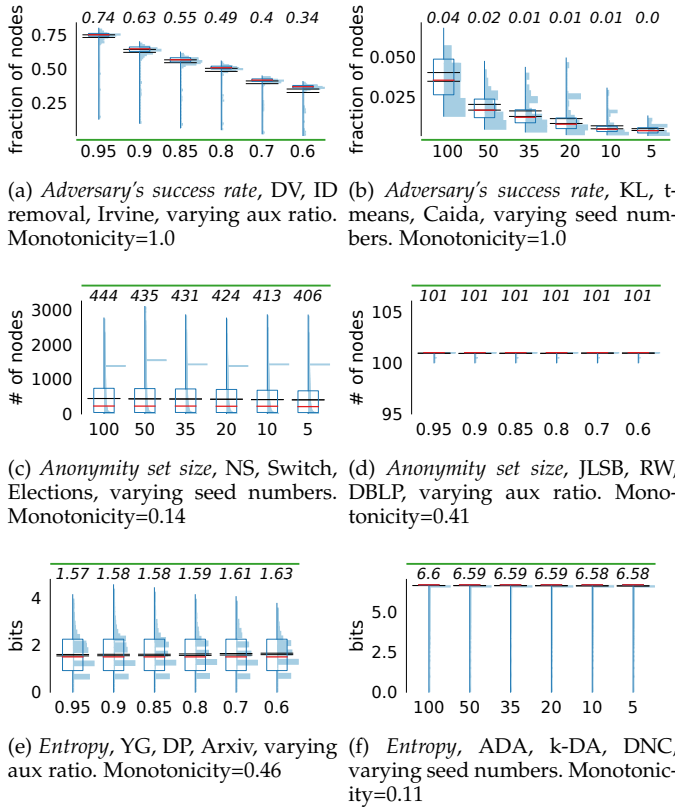


Fig. 2. Detailed results for a selection of metrics.

decrease in the success rate from left to right, with a monotonicity score of 1.0.

The plots in the second row show the *anonymity set size* in two scenarios (Figures 2c–2d). In Figure 2c, the low monotonicity score results from a change in the wrong direction: instead of the *anonymity set size* getting larger with decreasing adversary strength, it is actually getting smaller.

The lack of variation in Figure 2d is due to an undocumented implementation detail in SecGraph: for all de-anonymizers that use the global graph structure (ADA, DV, JLSB), the SecGraph implementations process the graph in chunks to keep the runtime within reasonable limits (as the plot shows, we chose a chunk size of 100). This chunking does not have a large effect on the *adversary's success rate* because the nodes in the anonymized and auxiliary graphs are ordered by their degrees before chunking. However, metrics that use the adversary's estimated probability distribution are skewed by this artificial limit.

The third row (Figures 2e–2f) shows entropy in two scenarios. *Entropy* measures the adversary's uncertainty, and we expect that *entropy* increases with higher anonymization levels. In Figure 2e, the general trend of the metric is in the right direction. However, the monotonicity score is only a medium 0.46 because some of the adversary strength levels have no statistically significant difference. Figure 2f shows how *entropy* is affected by SecGraph's chunking strategy: the artificial limit of 100 candidates for each node results in almost constant values for *entropy* throughout.

We have applied the algorithm described in Section 2.4.1 to summarize the individual results from Figure 2 into

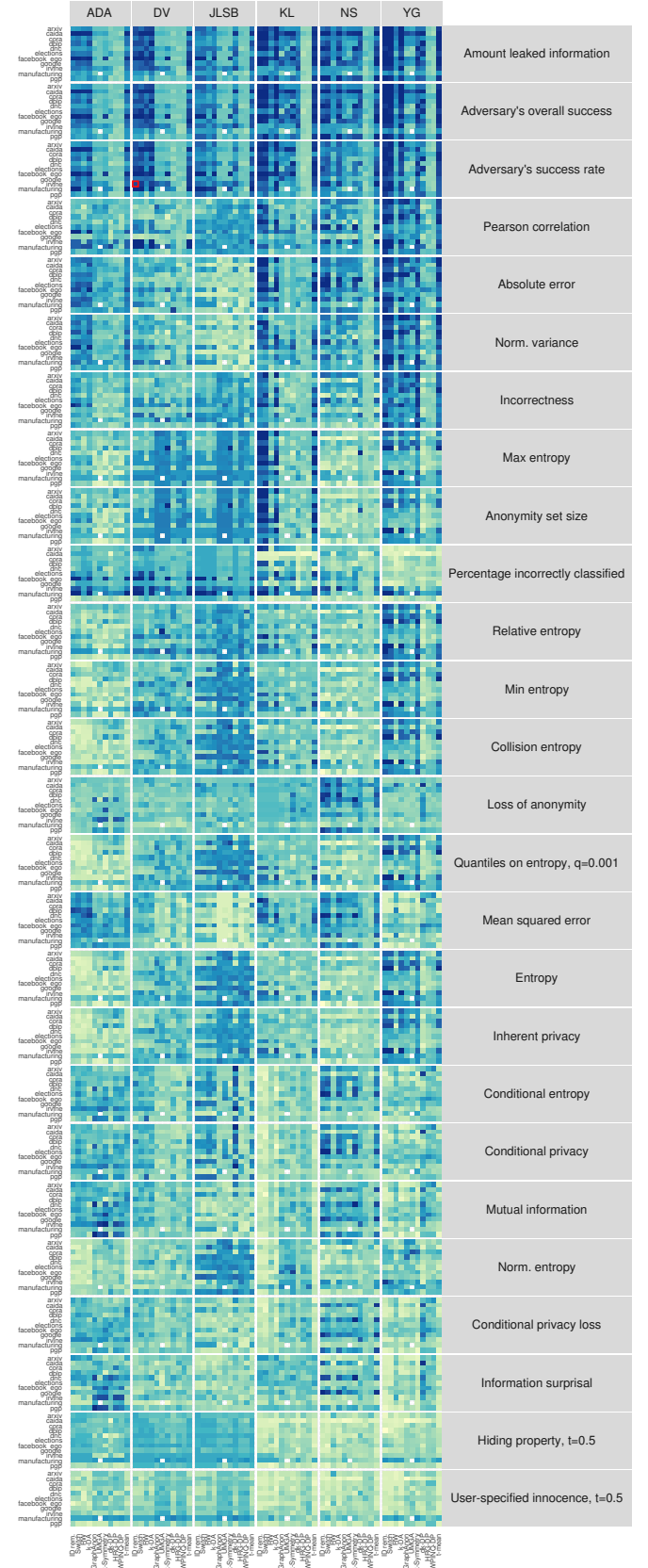


Fig. 3. Heat map visualizing monotonicity scores for all privacy metrics in our study, depending on the de-anonymization algorithm (top), anonymization algorithm (bottom), and graph dataset (left). Each cell averages results for varying seed numbers, varying aux ratios, and all replications. Light yellow colors indicate low monotonicity (weak metric), and dark blue colors indicate high monotonicity (strong metric).

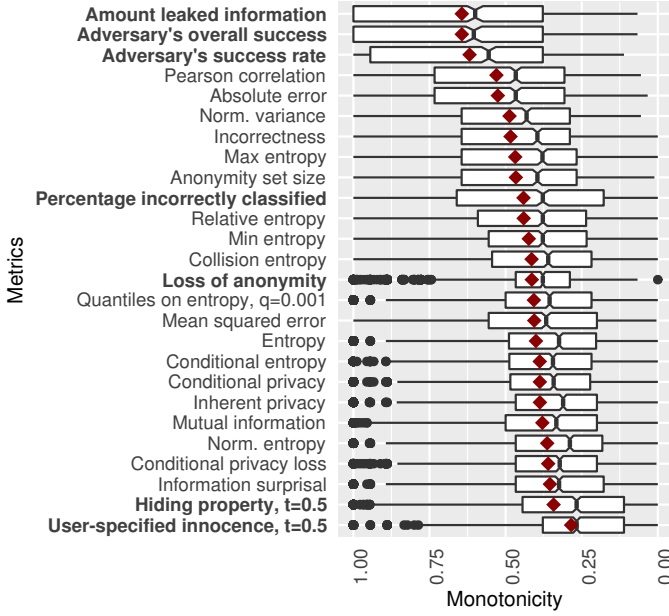


Fig. 4. Box plot showing the ranking of privacy metrics according to their monotonicity scores. Each box summarizes monotonicity for all anonymization and de-anonymization algorithms and all graph datasets. Per-graph metrics are marked in bold.

monotonicity scores¹. Figure 3 shows these monotonicity scores on a heat map. Each field in the heat map represents one set of box plots from Figure 2. As an example, we have highlighted the field corresponding to Figure 2a with a red outline. The heat map shows that only few metrics have high monotonicity, most notably the *adversary's success rate* and the *amount of leaked information*. These metrics are based on the count of nodes that the adversary has successfully re-identified.

Most other metrics have medium or low monotonicity. This includes almost all metrics that are based on the adversary's probability distribution, for example *entropy* and the metrics derived from entropy.

Figure 4 shows the monotonicity scores for all metrics in a box plot where each box combines the monotonicity scores for all anonymization algorithms, all de-anonymization algorithms, and all graph datasets. The plot is ordered according to the average score. We can see that most metrics have monotonicity scores below 0.5, which indicates that they are not monotonic in most cases and thus should be used with caution, if at all. In addition, many metrics that perform well in other domains do not perform well when applied to graph privacy, for example *normalized entropy* and the *anonymity set size*. We investigate the reasons for this in the following two sections.

5.2 Analysis of Factors Influencing Monotonicity

We have defined monotonicity as a property of privacy metrics, so we expect that the metrics should be the main determining factor for the value of monotonicity. However, we have evaluated monotonicity in complex scenarios that include different graph datasets, anonymization algorithms,

1. The full set of scores for monotonicity, evenness, and shared value range is included in the supplementary material, available online.

TABLE 4
MARS model showing which factors influence monotonicity, in order of importance

monotonicity = 0.34		
-	0.048	* dataset pgp
+	0.063	* metric Relative entropy
+	0.063	* metric Percentage incorrectly classified
+	0.15	* metric Pearson correlation
+	0.11	* metric Norm. variance
+	0.046	* metric Min-entropy
+	0.091	* metric Max-entropy
+	0.11	* metric Incorrectness
+	0.088	* metric Anonymity set size
+	0.27	* metric Amount leaked information
+	0.24	* metric Adversary's success rate
+	0.27	* metric Adversary's overall success
+	0.15	* metric Absolute error
-	0.038	* anonymizer HRG-DP
+	0.087	* anonymizer IDrem.
+	0.08	* anonymizer k-DA
+	0.042	* anonymizer k-Symmetry
+	0.088	* anonymizer Switch
+	0.081	* anonymizer t-mean
-	0.047	* anonymizer WPINQ-DP
+	0.025	* de-anonymizer DV
+	0.059	* de-anonymizer JLSB
+	0.043	* de-anonymizer YG
+	2.4e-06	* max(0, 7118 - nodes)
-	7.7e-07	* max(0, nodes - 7118)
-	0.0022	* max(0, 15 - avg. degree)

and de-anonymization algorithms. Each of these factors may influence the value of monotonicity as well. To find out to what extent the variation in monotonicity can be explained by each of these factors, we used multivariate adaptive regression splines (MARS) [7] to analyze how much each of them contributes to the final monotonicity score. MARS is more suitable than linear regression in this case because it accounts for non-linearity and interactions between the variables and automatically selects the most important variables to include in the model.

Table 4 shows the contribution of each factor according to the MARS analysis. The results show that metrics are indeed the most important contributors and have by far the largest influence on the monotonicity score. In contrast, the influence from anonymizing/de-anonymizing algorithms and graph statistics is much smaller. In many cases, the influence was so small that the variables were not important enough to be included in the model at all. In particular, the factor that indicates whether de-anonymization algorithms use global or local information was not important enough to be included in the final MARS model.

We fit a linear regression to the data to confirm this finding and found that all graph statistics combined account for less than 1% of the variation in monotonicity ($R^2 = 0.006$), and the anonymizer, de-anonymizer, dataset and graph type combined account for less than 9% ($R^2 = 0.086$).

We conclude that despite our complex evaluation setup, monotonicity is primarily influenced by the choice of metric, and not by the dataset or anonymization/de-anonymization algorithms. Because we have evaluated a wide range of datasets, anonymizers, and de-anonymizers, we are confident that our results on the monotonicity of graph privacy metrics hold also for other datasets and algorithms.

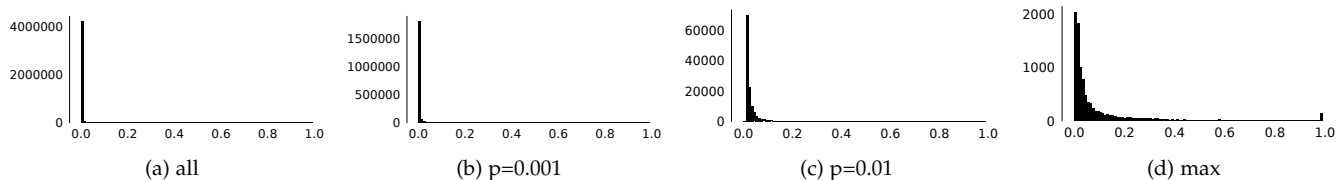


Fig. 5. Distribution of the adversary's estimate for elections, NS, k-DA. Figure (a) shows all values, (b) and (c) cut off all probability values smaller than p , and (d) shows the probability distribution for the most likely candidate node.

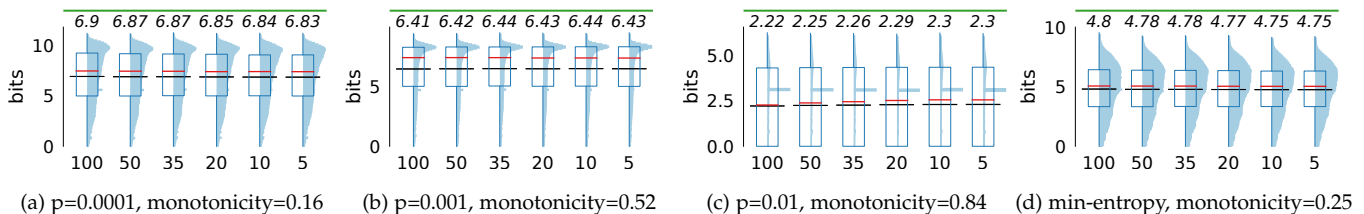


Fig. 6. *Entropy* based on the adversary's estimated probabilities without probabilities below p , for elections, NS, k-DA.

5.3 Analysis of Non-monotonic Metrics

Our results show that some metrics that are popular and monotonic in other application domains are not monotonic when measuring graph privacy. In this section we analyze why this is the case for information theoretic metrics based on *entropy* (Section 5.3.1), give examples of metrics that are affected by implementation details of the de-anonymization algorithm, e.g. the *anonymity set size* (Section 5.3.2), and analyze the behavior of the *adversary's success rate* in detail (Section 5.3.3).

5.3.1 Entropy affected by low-probability candidates

One reason why *entropy* and metrics derived from entropy have low monotonicity scores may be that low-probability candidates have a large influence on *entropy* [6], [21]. To find out whether this is the case here, we plot the adversary's probability distribution using a histogram with 100 bins (Figure 5a). The histogram clearly shows that most values are in the left-most bin, indicating that most probabilities are below 1%, which confirms the presence of a large number of low-probability candidates.

As suggested in [6], a possible way of dealing with these low-probability candidates is to calculate *entropy* based on a quantile of the adversary's probability distribution, i.e. to remove a certain portion of low-probability candidates before calculating the entropy value. To evaluate the effect of this, we define versions of *entropy* that cut off probability values below p before the calculation. Figure 6 shows the detailed results and monotonicity scores for three of these modified entropies, with $p = \{0.0001, 0.001, 0.01\}$, and Figures 5b–5c show the corresponding histograms for the adversary's probability distribution. We can see that removing probability values below 0.0001 does not improve the monotonicity score for *entropy*, 0.001 shows a moderate improvement, and 0.01 shows a clear improvement.

Depending on the anonymization algorithm, removing small probability values corresponds to removing a certain percentile of the adversary's probability distribution: removing probability values below 0.0001 corresponds to

removing 0.16% of the probability distribution, removing $p < 0.001$ corresponds to removing 55%, and removing $p < 0.01$ corresponds to 97% of the probability distribution. Even though *entropy* in the last case is monotonic, it is questionable whether it still makes sense as a privacy metric because it only evaluates the adversary's uncertainty based on the top 3% of her probability distribution.

Another way of dealing with low-probability candidates is to use *min-entropy*, a variant of *entropy* that is calculated based only on the most likely candidate. However, as Figure 6d shows, *min-entropy* shows the same non-monotonic behavior as *entropy*. To explain why, we plot the distribution of the adversary's probability for the most likely candidate for each node in Figure 5d. The figure shows that for the majority of nodes, the adversary makes a mapping decision based on a probability smaller than 1% (the left-most bin). Just like *entropy*, *min-entropy* is therefore influenced by low-probability candidates.

As a result, entropy-based metrics do not seem to be suitable to evaluate graph privacy, even if only a certain percentile of the adversary's distribution is used in their calculation.

5.3.2 Metrics affected by SecGraph's chunks

We have shown in Figure 2d that the anonymity set size can be influenced by SecGraph's strategy of processing graphs in chunks. This chunking does not necessarily result in low monotonicity scores because the metric values are often indistinguishable from one adversary strength to the next, which results in a medium monotonicity. However, the semantics and interpretation of the metric are affected: in the case of the *anonymity set size*, the interpretation is no longer "all nodes that the adversary cannot distinguish," but instead "all nodes that the adversary cannot distinguish and that are in a group of 100 nodes with similar node degrees." This change in interpretation can make the metric unsuitable for evaluating graph privacy.

To analyze whether the *anonymity set size* is the only metric affected by SecGraph's chunking, we plot the maximum

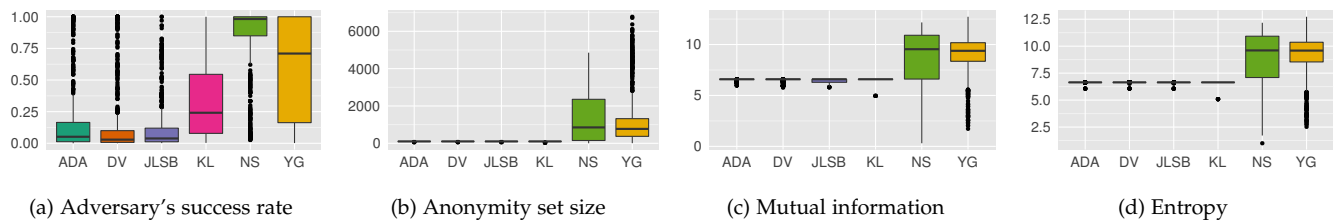


Fig. 7. Maximum values for four metrics across all datasets and anonymizers, by de-anonymizing algorithm. The *adversary's success rate* is not affected by SecGraph's chunking, whereas the other three metrics are: their maximum values have an artificial upper limit.

values for each metric separately for each de-anonymizing algorithm. If a metric is affected by chunking, the four de-anonymizers that use chunking (ADA, DV, JLSB, KL) should show a consistent upper limit, whereas NS and YG should not. Figure 7 shows these plots for four example metrics. It is clear that the *anonymity set size*, *mutual information*, and *entropy* are affected by chunking, but the *adversary's success rate* is not. Table 2 (column *chunk*) indicates which metrics are negatively affected by SecGraph's chunking.

5.3.3 Adversary's Success Rate vs Overall Success Rate

We have already mentioned the difference between the *adversary's success rate* and *adversary's overall success* in Section 3.4: success rate is based on the number of attempted de-anonymizations, whereas overall success rate is based on the total number of nodes in the graph. We will show in this section why this difference can lead to misjudgment of privacy levels when comparing different de-anonymization algorithms.

We have studied two groups of de-anonymization algorithms: global algorithms (ADA, DV, JLSB) re-identify nodes based on the global graph structure, whereas local algorithms (KL, NS, YG) use only local information such as the number of neighbors. Local algorithms generally bootstrap from the given seed mappings and iteratively add nodes to the set of seeds, stopping when no more nodes can be added. As a result, local algorithms attempt to re-identify fewer nodes than global algorithms, and this increases their success rate (which is based on attempted nodes) compared to global algorithms.

Figure 8 compares the two metrics for each of the six de-anonymization algorithms. We can see that the *adversary's overall success* (light color) is nearly equal for all six algorithms, especially when comparing the median value indicated by the line in each box. For the global algorithms, the *adversary's success rate* is very close to the *adversary's overall success*. In contrast, the local algorithms report a much higher success rate than overall success rate.

There is no doubt that the *adversary's success rate* can be a useful metric, for example to highlight the potential of a de-anonymizer if the right kind of auxiliary information is available to the adversary. However, it is unsuitable for comparing different de-anonymizing algorithms because it reports inflated success rates for local algorithms.

6 RESULTS ON ADDITIONAL CRITERIA

Our analysis so far has focused on the monotonicity of privacy metrics. However, as the comparison between the

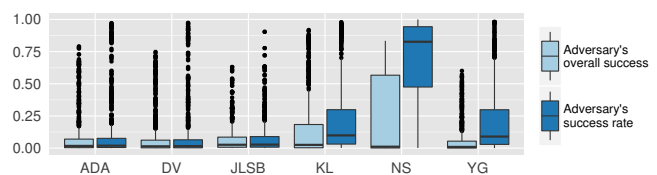


Fig. 8. Comparison between the *adversary's success rate* and the *adversary's overall success*. De-anonymizers that use local information (KL, NS, YG) have an inflated success rate because it is normalized with the number of attempted re-identifications.

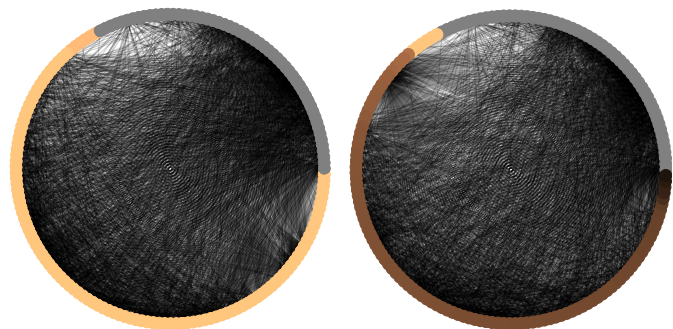


Fig. 9. Metric values visualized on the edges of a circular graph layout. Light colors: high privacy, gray: de-anonymization not attempted.

(a) *Incorrectness*, Manufacturing, DV, DP. Evenness=0.14
 (b) *Pearson correlation*, Manufacturing, DV, DP. Evenness=0.89

adversary's success rate and *overall success rate* has shown, monotonicity is not always a sufficient criterion to select privacy metrics. Prior work has identified evenness and shared value range as additional criteria for metric selection especially for within-scenario comparisons (evenness) and between-scenario comparisons (shared value range) [42].

6.1 Evenness

To illustrate the requirement for evenness, Figure 9 compares two privacy metrics applied to the same scenario. Each subfigure shows the graph dataset in a circular layout, with the nodes colored according to their privacy. Light colors indicate high privacy, and nodes for which de-anonymization has not been attempted are colored gray. *Incorrectness* on the left (Figure 9a) has a low evenness score (0.14), which can be seen by the absence of medium and dark colors. In contrast, *pearson correlation* on the right (Figure 9b) has a high evenness score (0.89), which is visible in the clear representation of light, medium, and dark colors. Metrics with a high evenness score thus allow to analyze

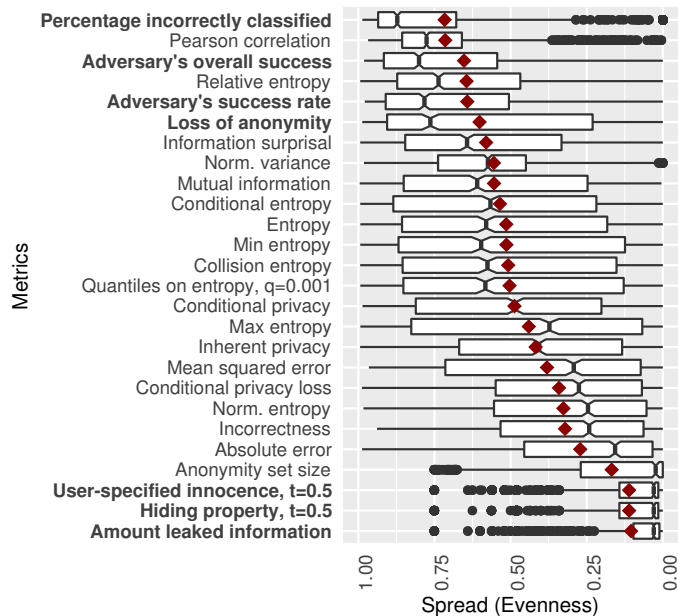


Fig. 10. Box plot showing the ranking of privacy metrics according to their evenness scores.

which nodes in a graph have better privacy protection than others.

We have computed evenness scores according to the description in Section 2.4.2 for all 1440 scenarios. Figure 10 shows a ranking of privacy metrics according to their average evenness score, with the boxes indicating the distribution of scores across all scenarios. Highly even metrics are particularly desirable when comparing nodes within a graph, which only the per-node metrics (non-bold in Figure 10) can do. Among the metrics with an average evenness score above 0.5, only two metrics are per-node and also have a monotonicity score above 0.5: *pearson correlation* and *normalized variance*. If comparisons between nodes are needed, these two metrics are recommended.

6.2 Shared Value Range

To illustrate the requirement for a shared value range, we show two metrics, *pearson correlation* and *anonymity set size*, in two scenarios each in Figure 11. The top row shows *pearson correlation* with a consistent value range of [0,1] in every scenario. In contrast, the bottom row shows the *anonymity set size*, where each scenario uses a different portion of the global value range, depending on the size of the graph, the anonymization/de-anonymization algorithms and how they are implemented. This indicates that the *anonymity set size* is less suitable for comparisons between scenarios than *pearson correlation*.

Figure 12 shows the distribution and average values for the shared value range score across all scenarios. We note that none of the per-graph metrics have a high shared value range. As a result, even for comparisons between graphs, per-node metrics are recommended. Only three metrics have a shared value range score above 0.5 as well as a monotonicity score above 0.5: *pearson correlation*, *absolute error*, and *normalized variance*.

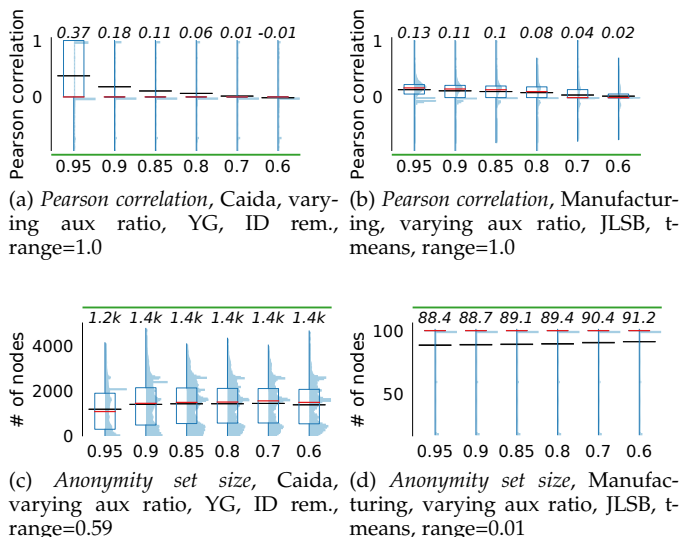


Fig. 11. Shared value range: *pearson correlation* in the top row has a consistent value range across scenarios, while the *anonymity set size* in the bottom row uses a different value range (y axis) for each scenario.

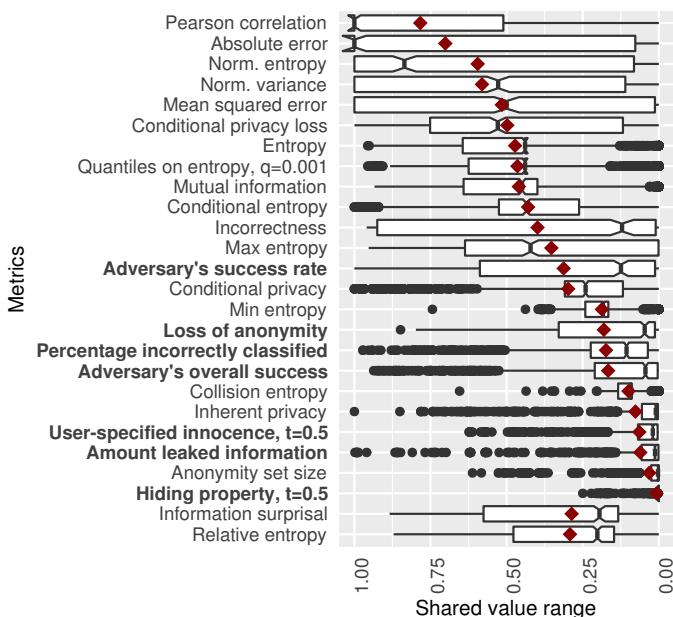


Fig. 12. Box plot showing the ranking of privacy metrics according to their shared value range scores.

6.3 Analysis of Factors Influencing Evenness and Shared Value Range

We again used MARS and linear regression to analyze how much of the variation in evenness and shared value range can be explained by characteristics of the dataset, anonymization, and de-anonymization algorithms. Tables 5 and 6 show that, similar to monotonicity, evenness and shared value range are mostly influenced by the metrics and not by the other components of our experiments.

The linear regression confirmed that the anonymizer, de-anonymizer, dataset and graph type together account for under 20% of the variation in evenness ($R^2 = 0.194$) and under 17% of the variation in shared value range ($R^2 = 0.166$). The graph statistics account for under 2% ($R^2 =$

TABLE 5
MARS model showing which factors influence evenness

evenness =			
-0.018			
+ 0.54	*	metric	Relative entropy
+ 0.4	*	metric	Quantiles on entropy, q=0.001
+ 0.61	*	metric	Percentage incorrectly classified
+ 0.61	*	metric	Pearson correlation
+ 0.45	*	metric	Norm. variance
+ 0.22	*	metric	Norm. entropy
+ 0.45	*	metric	Mutual information
+ 0.41	*	metric	Min-entropy
+ 0.27	*	metric	Mean squared error
+ 0.33	*	metric	Max-entropy
+ 0.49	*	metric	Loss of anonymity
+ 0.31	*	metric	Inherent privacy
+ 0.48	*	metric	Information surprisal
+ 0.21	*	metric	Incorrectness
+ 0.41	*	metric	Entropy
+ 0.23	*	metric	Conditional privacy loss
+ 0.38	*	metric	Conditional privacy
+ 0.43	*	metric	Conditional entropy
+ 0.4	*	metric	Collision entropy
+ 0.06	*	metric	Anonymity set size
+ 0.53	*	metric	Adversary's success rate
+ 0.54	*	metric	Adversary's overall success
+ 0.16	*	metric	Absolute error
- 0.039	*	anonymizer	k-Symmetry
+ 0.15	*	de-anonymizer	DV
+ 0.2	*	de-anonymizer	JLSB
+ 0.2	*	de-anonymizer	KL
+ 0.33	*	de-anonymizer	NS
+ 0.23	*	de-anonymizer	YG
+ 3.1e-08	*	max(0, 2e+06 - wedges)	
- 6.6e-09	*	max(0, wedges - 2e+06)	
- 0.0011	*	max(0, 94 - spectral norm)	

0.015) of the variation in evenness and under 1% ($R^2 = 0.0006$) of the variation in shared value range.

This replicates our finding for monotonicity and confirms that evenness and shared value range are primarily influenced by the choice of metric, and not by the dataset or anonymization/de-anonymization algorithms.

7 RECOMMENDATIONS: METRICS SUITES

Based on our experimental results and the analysis in the previous sections, we can see that no single metric is the ideal metric for all scenarios. The metrics with the highest monotonicity are per-graph metrics, which do not provide information about the privacy levels of individual nodes in a graph. Most of the other metrics have very low monotonicity, and only seven metrics have an average monotonicity score of 0.5 or above. When considering evenness and shared value range in addition to monotonicity, only two metrics have high scores in all three criteria (*pearson correlation* and *normalized variance*). Finally, another important consideration is the semantics of metrics and their interpretation. In this regard, it is desirable to measure different aspects of privacy, for example in terms of the categories described in Section 3.4 [28].

Therefore, we recommend to combine several privacy metrics in a metrics suite. To allow for easy comparisons between anonymizing or de-anonymizing algorithms, and to be able to decide which provides the best or worst privacy levels, it is desirable to combine the metrics in a suite into

TABLE 6
MARS model showing which factors influence shared value range

shared value range =			
0.22			
+ 0.14	*	metric	Relative entropy
+ 0.31	*	metric	Quantiles on entropy, q=0.001
+ 0.63	*	metric	Pearson correlation
+ 0.43	*	metric	Norm. variance
+ 0.44	*	metric	Norm. entropy
+ 0.31	*	metric	Mutual information
+ 0.36	*	metric	Mean squared error
+ 0.2	*	metric	Max-entropy
- 0.077	*	metric	Inherent privacy
+ 0.13	*	metric	Information surprisal
+ 0.24	*	metric	Incorrectness
- 0.15	*	metric	Hiding property, t=0.5
+ 0.32	*	metric	Entropy
+ 0.34	*	metric	Conditional privacy loss
+ 0.14	*	metric	Conditional privacy
+ 0.28	*	metric	Conditional entropy
- 0.055	*	metric	Collision entropy
- 0.12	*	metric	Anonymity set size
- 0.093	*	metric	Amount leaked information
+ 0.16	*	metric	Adversary's success rate
+ 0.55	*	metric	Absolute error
- 0.063	*	anonymizer	HRG-DP
- 0.07	*	anonymizer	WPINQ-DP
- 0.15	*	de-anonymizer	DV
- 0.15	*	de-anonymizer	JLSB
- 0.18	*	de-anonymizer	KL
+ 0.12	*	de-anonymizer	NS
+ 0.049	*	de-anonymizer	YG

a single number. This is not a straightforward task because metrics have different scales of measurement and different directions (for some, higher values indicate higher privacy, for others, lower values). A simple average is therefore unlikely to yield good results, and any normalization has to carefully consider direction and value range.

7.1 Combining Metrics in a Metrics Suite

The problem of combining metrics is similar to a problem in operations research: given a set of alternatives and a number of criteria for each alternative, which alternative is best? Many methods have been proposed for multi-criteria decision analysis [33]. The simplest method is the weighted sum model (WSM), which ranks alternatives according to a weighted sum of the criteria values. However, this method is only applicable if the criteria are measured on the same scale and in the same units. The weighted product model (WPM) instead computes the relative importance Q_i of each alternative i using a weighted product. Applied to the case of privacy metrics, each criterion j corresponds to one privacy metric, and the set of alternatives is the set of scenarios (e.g., anonymization algorithms or parameter settings) that need to be compared in terms of their privacy. According to [33], the relative importance of each alternative i is computed as

$$Q_i = \prod_{j=1}^n (\bar{x}_{ij})^{w_j},$$

where w_j is the weight assigned to each metric, and \bar{x}_{ij} is a normalization of the metric value x_{ij} such that $\bar{x}_{ij} = \frac{x_{ij}}{\max_i x_{ij}}$ if higher values indicate higher privacy, and

$\bar{x}_{ij} = \frac{\min_i x_{ij}}{x_{ij}}$ if lower values indicate higher privacy. The advantages of WPM are that different units of measurement are effectively eliminated due to the use of multiplication instead of addition, that the built-in normalization can integrate higher-better and lower-better metrics, and that it does not suffer from rank reversals that can occur with additive methods [32].

7.2 Choosing Metrics for a Metrics Suite

As we have already discussed above, every metric in a metrics suite should have a monotonicity score of at least 0.5. The choice of metrics can then depend on the specific demands of the application and might include metrics with high evenness, metrics with high shared value range, metrics from different categories, and metrics that are easy to interpret in the application context. Importantly, metrics suites should also include metrics that satisfy the tick-box criteria described in Section 2.4.

In our experiments, the seven metrics that have high monotonicity scores are the *adversary's overall success*, *amount leaked information*, *adversary's success rate*, *absolute error*, *pearson correlation*, *normalized variance*, and *incorrectness*. This list includes metrics with high evenness and shared value range, and covers four different categories (information gain/loss, error, similarity, success probability). The list also includes the *adversary's success rate* which is the most common metric to evaluate graph privacy, and it fulfills some of the criteria from Section 2.4.

7.3 Monotonicity of Metrics Suites

It is not immediately clear how metrics and their weights should be selected so that the metrics suite achieves the best monotonicity. In general, the problem can be formulated as an optimization problem, but due to the size of the search space it may only be solvable with heuristic optimization algorithms. We consider this out of scope for the current paper. Instead, we have evaluated all metrics suites with equal weights that result from combinations of the metrics with a monotonicity score of 0.5 or higher, that is, 120 suites constructed from the top-7 monotonic metrics. We then manually optimized their weights to improve monotonicity.

To evaluate the monotonicity of metrics suites, we note that a good metrics suite should create a monotonic ranking of alternatives. Our experimental data consists of 1440 scenarios, each with six adversary strength levels with increasing strength defined either by an increase in the number of seed mappings or by an increase in the overlap of the auxiliary graph. We can therefore use each scenario as one set of six alternatives, using the increasing strength levels as the ground truth for how the alternatives should be ranked. For each set of alternatives, we used the mean metric values across all replications as x_{ij} and then count how many of the six strength levels have been ranked monotonically by the metrics suite score Q_i^2 .

We find that most metrics suites create monotonic rankings for more than 80% of scenarios, and only metrics suites that include none of the top-3 metrics create monotonic

2. The full set of mean metric values is included in the supplementary material, available online.

TABLE 7
Composition of four metrics suites (S2 to S5) and the percentage of monotonic rankings resulting from their aggregation with WPM, compared with the best individual metric (S1).

	Metrics	Weights	% mono
S1	Amount leaked information	equal	82.3
S2	Adversary's overall success, Norm. variance	equal	82.4
S3	Pearson correlation, Adversary's overall success, Norm. variance, Amount leaked information	equal	82.5
S4	Pearson correlation, Adversary's overall success, Norm. variance, Amount leaked information, Incorrectness	0.1, 0.35, 0.1, 0.35, 0.1	82.4
S5	Pearson correlation, Norm. variance, Incorrectness, Amount leaked information, Adversary's success rate, Adversary's overall success, Absolute error	0.1, 0.1, 0.1, 0.25, 0.1, 0.25, 0.1	82.9

rankings for less than 70% of scenarios. In comparison, the best individual metrics (*adversary's overall success* and *amount information leaked*) create monotonic rankings for 82.3% of scenarios.

Table 7 summarizes the composition and weights for four of the top-scoring metrics suites and the best individual metric. When choosing all weights w_j to be equal, the best metrics suite, consisting of *pearson correlation*, *adversary's overall success*, *normalized variance* and *amount leaked information*, ranks 82.5% of scenarios monotonically. Importantly, this is better than the result for the best individual metric.

When choosing unequal weights, the best metrics suite we have been able to construct consists of *pearson correlation*, *normalized variance*, *incorrectness*, *amount leaked information*, *adversary's success rate*, *adversary's overall success*, and *absolute error* with weights $w_i = 0.1, 0.1, 0.1, 0.25, 0.1, 0.25, 0.1$. This metrics suite creates monotonic rankings for 82.9% of scenarios.

In summary, we find that combining privacy metrics in a metrics suite can improve monotonicity above the monotonicity of individual metrics. This is an important result that shows a concrete method towards more monotonic and thus more consistent evaluations of privacy.

7.4 Evenness and Shared Value Range of Metrics Suites

The metrics suite score Q_i represents the weighted, normalized product of metric values. We can therefore use Q_i as the aggregated privacy value for each scenario and compute the evenness and shared value range scores for each metrics suite. Figure 13a shows that the evenness scores of the five metrics suites are better than the scores for individual metrics (Figure 10). However, the shared value range scores shown in Figure 13b are not as good as the scores for individual metrics. Together, these findings indicate that the aggregated privacy values Q_i are particularly suitable to evaluate the overall privacy performance, ensuring a monotonic and evenly spread ranking of alternatives. To analyze per-node privacy and to interpret privacy values, the metric values of individual metrics seem more suitable.

We also find that computing the metrics suite score Q_i for suites with a single metric can greatly improve the metric's evenness and shared value range scores. For example,

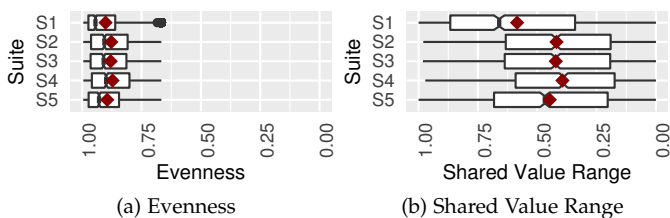


Fig. 13. Box plots showing the evenness and shared value range scores of five metrics suites.

TABLE 8

Monotonicity, evenness, and shared value range scores for elections, HRG-DP, JLSB, for metrics that make up metrics suite S5 and for S5.

Metric	Monotonicity	Evenness	Range
Absolute error	0.467	0.051	0.0026
Adversary's overall success	0.467	6.667e-16	0.004
Adversary's success rate	0.467	6.667e-16	0.004
Amount leaked information	0.467	6.667e-16	0.001
Incorrectness	0.307	0.593	0.0002
Norm. variance	0.467	0.037	0.0007
Pearson correlation	0.271	0.142	0.031
Suite S5	0.833	0.929	0.431

the first row of Figures 13a and 13b corresponds to metrics suite S1, which consists of one single metric, *amount of leaked information*. Compared with the evenness and shared value range scores for the *amount of leaked information* individually (Figures 10 and 12), this row shows a significant improvement in the evenness and shared value range scores.

7.5 Application Example

To show how metrics suites can be applied in practice, we select an example combination of graph, anonymizer, and de-anonymizer: elections, HRG-DP, and JLSB. Table 8 shows the monotonicity, evenness, and shared value range scores for metrics suite S5 and for the seven metrics that S5 is composed of. We can see that S5 has significantly higher scores than any of the individual metrics. Therefore, the metrics suite is more suitable than the individual metrics to compare and rank the overall privacy values for different graphs or algorithms. However, the detail needed for per-node comparisons is lost in the aggregation, which means that one of the individual metrics needs to be used for this. In this example, incorrectness as the metric with the highest individual evenness score would be most suitable.

8 CONCLUSION

In this paper, we analyzed the strength of 26 privacy metrics for graph privacy in terms of their monotonicity, evenness, and shared value range. We conducted extensive experiments on 11 public graph datasets, using 11 anonymization algorithms and 6 de-anonymization algorithms. We found that most metrics are not monotonic when applied in graph privacy, including several metrics that are popular and monotonic in other fields. Our detailed analysis of the strengths and weaknesses of these metrics led us to propose metrics suites, that is, combinations of privacy metrics that can combine the strengths and mitigate the weaknesses of

individual metrics. To the best of our knowledge, we were the first to apply techniques from multi-criteria decision analysis to privacy measurement and found that the resulting metrics suites can indeed increase monotonicity above the monotonicity of the best individual metric. This result opens up a new line of research that may lead to significant improvements in privacy measurement.

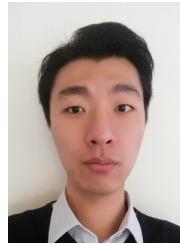
ACKNOWLEDGMENTS

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/P006752/1 and used the ARCHER UK National Supercomputing Service (<http://www.archer.ac.uk>). The authors thank the anonymous reviewers for their insightful comments.

REFERENCES

- [1] J. Alexander and J. Smith, "Engineering Privacy in Public: Confounding Face Recognition," in *3rd International Workshop on Privacy Enhancing Technologies (PET)*. Dresden, Germany: Springer LNCS, volume 2760, Mar. 2003, pp. 88–106.
- [2] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore Art Thou R3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography," in *Proceedings of the 16th International Conference on World Wide Web*, ser. WWW '07. Banff, Alberta, Canada: ACM, May 2007, pp. 181–190.
- [3] E. Bertino, D. Lin, and W. Jiang, "A Survey of Quantification of Privacy Preserving Data Mining Algorithms," in *Privacy-Preserving Data Mining: Models and Algorithms*. Springer Advances in Database Systems, Jul. 2008, no. 34, ch. 8, pp. 183–205.
- [4] J. Casas-Roma, J. Herrera-Joancomarti, and V. Torra, "k-Degree Anonymity and Edge Selection: Improving Data Utility in Large Networks," *Knowledge and Information Systems*, vol. 50, no. 2, pp. 447–474, feb 2017.
- [5] S. Chester, B. M. Kapron, G. Ramesh, G. Srivastava, A. Thomo, and S. Venkatesh, "Why Waldo Befriended the Dummy? k-Anonymization of Social Networks with Pseudo-Nodes," *Social Network Analysis and Mining*, vol. 3, no. 3, pp. 381–399, sep 2013.
- [6] S. Clauß and S. Schiffner, "Structuring Anonymity Metrics," in *Proceedings of the 2nd ACM Workshop on Digital Identity Management (DIM)*, Alexandria, VA, USA, Nov. 2006, pp. 55–62.
- [7] J. H. Friedman, "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991.
- [8] S. Ji, W. Li, S. Yang, P. Mittal, and R. Beyah, "On the Relative De-Anonymizability of Graph Data: Quantification and Evaluation," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.
- [9] S. Ji, P. Mittal, and R. Beyah, "Graph Data Anonymization, De-Anonymization Attacks, and De-Anonymizability Quantification: A Survey," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1305–1326, Secondquarter 2017.
- [10] S. Ji, W. Li, P. Mittal, X. Hu, and R. Beyah, "SecGraph: A Uniform and Open-source Evaluation System for Graph Data Anonymization and De-anonymization," in *Proceedings of the 24th USENIX Security Symposium (USENIX Security)*, Washington, D.C., Aug. 2015, pp. 303–318.
- [11] S. Ji, W. Li, M. Srivatsa, and R. Beyah, "Structural Data De-anonymization: Quantification, Practice, and Implications," in *Proceedings of the 2014 ACM Conference on Computer and Communications Security*, Scottsdale, AZ, USA, Nov. 2014, pp. 1040–1053.
- [12] S. Ji, W. Li, M. Srivatsa, J. S. He, and R. Beyah, "Structure Based Data De-Anonymization of Social Networks and Mobility Traces," in *Proceedings of the 17th International Conference on Information Security*, Hong Kong, China, Oct. 2014, pp. 237–254.
- [13] G. Keinan, "k-Anonymity Simulator for Social Networks," <https://github.com/Keinang/K-Anonymity>, 2017.
- [14] N. Korula and S. Lattanzi, "An Efficient Reconciliation Algorithm for Social Networks," *Proceedings of the VLDB Endowment*, vol. 7, no. 5, pp. 377–388, Jan. 2014.
- [15] R. Kumar, J. Novak, and A. Tomkins, "Structure and Evolution of Online Social Networks," in *Link Mining: Models, Algorithms, and Applications*, 2010, pp. 337–357.

- [16] J. Kunegis, "KONECT: The Koblenz Network Collection," in *Proceedings of the 22nd International Conference on World Wide Web*. Rio de Janeiro, Brazil: ACM, May 2013, pp. 1343–1350.
- [17] J. Leskovec and R. Sosič, "SNAP: A General-Purpose Network Analysis and Graph-Mining Library," *ACM Transactions on Intelligent Systems and Technology*, vol. 8, no. 1, pp. 1:1–1:20, Jul. 2016.
- [18] K. Liu and E. Terzi, "Towards Identity Anonymization on Graphs," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, Vancouver, Canada, Jun. 2008, pp. 93–106.
- [19] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender Systems with Social Regularization," in *Proceedings of the 4th ACM International Conference on Web Search and Data Mining (WSDM)*, Hong Kong, China, Feb. 2011, pp. 287–296.
- [20] P. Mittal, C. Papamanthou, and D. Song, "Preserving Link Privacy in Social Network Based Systems," in *Proceedings of the 20th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, Feb. 2013, pp. 1–16.
- [21] S. J. Murdoch, "Quantifying and Measuring Anonymity," in *Proceedings of the 8th International Workshop on Data Privacy Management and Autonomous Spontaneous Security (DPM)*. Egham, UK: Springer LNCS, volume 8247, Sep. 2013, pp. 3–13.
- [22] A. Narayanan and V. Shmatikov, "Robust De-Anonymization of Large Sparse Datasets," in *IEEE Symposium on Security and Privacy*. Oakland, CA, USA: IEEE, May 2008, pp. 111–125.
- [23] —, "De-anonymizing Social Networks," in *IEEE Symposium on Security and Privacy*. Oakland, CA, USA: IEEE, May 2009, pp. 173–187.
- [24] P. Pedarsani and M. Grossglauser, "On the Privacy of Anonymized Networks," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. San Diego, California, USA: ACM, Aug. 2011, pp. 1235–1243.
- [25] D. Proserpio, S. Goldberg, and F. McSherry, "A Workflow for Differentially-Private Graph Synthesis," in *Proceedings of the 2012 ACM workshop on Workshop on online social networks - WOSN '12*. New York, New York, USA: ACM Press, 2012, p. 13.
- [26] A. Sala, X. Zhao, C. Wilson, H. Zheng, and B. Y. Zhao, "Sharing Graphs Using Differentially Private Graph Models," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement conference (IMC)*, Berlin, Germany, Nov. 2011, pp. 81–98.
- [27] K. Sharad and G. Danezis, "An Automated Social Graph De-anonymization Technique," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES)*. Scottsdale, Arizona, USA: ACM, 2014, pp. 47–58.
- [28] R. Shokri, G. Theodorakopoulos, J. Y. Le Boudec, and J. P. Hubaux, "Quantifying Location Privacy," in *IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, USA, May 2011, pp. 247–262.
- [29] M. Srivatsa and M. Hicks, "De-anonymizing Mobility Traces: Using Social Network as a Side-Channel," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, Raleigh, NC, USA, Oct. 2012, pp. 628–637.
- [30] P. Syverson, "Why I'm Not an Entropist," in *Proceedings of the 17th International Workshop on Security Protocols*. Cambridge, UK: Springer LNCS, volume 7028, Apr. 2009, pp. 213–230.
- [31] B. Thompson and D. Yao, "The Union-Split Algorithm and Cluster-Based Anonymization of Social Networks," in *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security (ASIACCS)*, Sydney, Australia, Mar. 2009, pp. 218–227.
- [32] C. Tofallis, "Add or Multiply? A Tutorial on Ranking and Choosing with Multiple Criteria," *INFORMS Transactions on Education*, vol. 14, no. 3, pp. 109–119, May 2014.
- [33] E. Triantaphyllou, *Multi-Criteria Decision Making Methods: A Comparative Study*. Boston, MA: Springer US, 2000.
- [34] I. Wagner, "Evaluating the Strength of Genomic Privacy Metrics," *ACM Transactions on Privacy and Security (TOPS)*, vol. 20, no. 1, pp. 2:1–2:34, Jan. 2017.
- [35] —, "Measuring Privacy in Vehicular Networks," in *Proceedings of the 42nd IEEE Conference on Local Computer Networks (LCN)*, Singapore, Oct. 2017, pp. 183–186.
- [36] I. Wagner and D. Eckhoff, "Technical Privacy Metrics: A Systematic Survey," *ACM Computing Surveys*, vol. 51, no. 3, pp. 57:1–57:46, Apr. 2018.
- [37] I. Wagner and Y. Zhao, "Privacy metrics for graph anonymization and de-anonymization," CodeOcean, <http://doi.org/cxv4>, December 2018.
- [38] W. Wu, Y. Xiao, W. Wang, Z. He, and Z. Wang, "k-symmetry model for identity anonymization in social networks," in *Proceedings of the 13th International Conference on Extending Database Technology - EDBT '10*. New York, New York, USA: ACM Press, 2010, p. 111.
- [39] Q. Xiao, R. Chen, and K.-L. Tan, "Differentially Private Network Data Release via Structural Inference," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*. New York, New York, USA: ACM Press, 2014, pp. 911–920.
- [40] L. Yartseva and M. Grossglauser, "On the Performance of Percolation Graph Matching," in *Proceedings of the 1st ACM Conference on Online Social Networks*, Boston, MA, USA, Oct. 2013, pp. 119–130.
- [41] X. Ying and X. Wu, "Randomizing Social Networks: a Spectrum Preserving Approach," in *Proceedings of the 2008 SIAM International Conference on Data Mining*, Atlanta, GA, USA, Apr. 2008, pp. 739–750.
- [42] Y. Zhao and I. Wagner, "On the Strength of Privacy Metrics for Vehicular Communication," *IEEE Transactions on Mobile Computing*, 2018.
- [43] —, "POSTER: Evaluating Privacy Metrics for Graph Anonymization and De-anonymization," in *ASIA CCS '18: 2018 ACM Asia Conference on Computer and Communications Security*. Incheon, Republic of Korea: ACM, June 2018, pp. 817–819.



Yuchen Zhao received the BEng and MSc degrees, both in information security from the Huazhong University of Science and Technology, in 2011 and Wuhan University, in 2013, respectively, and the PhD degree in computer science from the University of St Andrews, in 2017. Between 2017 and 2018 he was a research fellow in the Cyber Security Centre with De Montfort University, Leicester. He is a research fellow in computer science with the University of Southampton, United Kingdom. His research interests include privacy protection and human-data interaction.



Isabel Wagner received the MSc degree in computer science (Dipl.-Inf. Univ.) and the PhD degree in engineering (Dr.-Ing.) from the University of Erlangen, in 2005 and 2010, respectively. She is an associate professor in computer science with De Montfort University, Leicester, United Kingdom. In 2011 she was a JSPS postdoctoral fellow in the research group of Masayuki Murata with Osaka University, Japan. Her research interests include privacy and privacy-enhancing technologies, particularly on metrics to quantify the effectiveness of privacy protection mechanisms, as well as on privacy-enhancing technologies in genomics, smart cities, vehicular networks, and smart grids. She is a senior member of the IEEE.