Universiteit Leiden

The Cascading Effect of De-Anonymization Attacks in Social Networks

Angela van Delft (s3801322)

Supervisors:
Frank Takes & Rachel de Jong

Bachelor Thesis– Informatica

Leiden Institute of Advanced Computer Science (LIACS)
liacs.leidenuniv.nl

September 3, 2026

Abstract

In a social network pseudonymization alone is insufficient to protect privacy, as the structure of the network itself can be used to identify individuals, known as de-anonymization. This thesis studies the cascading effect of de-anonymization attacks in real-world networks. We investigate three important questions. First, we characterize the cascading effect by measuring how much of a network a single identified node, the seed node, can expose. Second, we quantify the minimum number of seed node identities needed to fully de-anonymize a network, which we call the seed set identification problem. Third, we compare two attacker scenarios, being DEGREE and VRQ, with and without so-called twin nodes that are structurally identical. To answer these questions, we propose a cascading algorithm that computes for each node the set of nodes that become identifiable once that node is identified, and an approximation algorithm that finds an upperbound for the minimum number of node identities needed in order to de-anonymize the entire network. For all seventeen considered networks, at least one of the investigated configuration allows a single node to de-anonymize more than 78.4% of the nodes in the network, and in nine datasets a single seed node suffices to de-anonymize the entire network. Under the VRQ combined with twin nodes configuration, no dataset requires more than 1.9% of nodes as seeds to de-anonymize the entire network. Twin node uniqueness significantly reduces the number of seeds needed across all datasets. These results show that pseudonymized networks, particularly when VRQ is used, are highly vulnerable to cascading de-anonymization.

Contents

1 Introduction

Networks are a way to represent relationships between entities. They appear in many forms, for example: traffic networks, computer networks, biological networks and social networks. In many of these settings, the data underlying the network is sensitive, for example because it is about humans. A social network may encode who communicates with whom, a medical network may encode relationships between patients and healthcare providers, and a financial network may encode transactions between individuals or organizations.

Network science research is typically about getting a better understanding of the connected structure of relevant interacting entities. In order to perform this research, real-world network data is often needed. Real-world networks differ from random networks, since they are mostly sparse, small world, have a skewed degree and are highly clustered. However, as mentioned above, network data can contain sensitive information. To enable data sharing and research while protecting user privacy, network data is commonly pseudonymized. Pseudonymization removes directly identifying information such as names, email addresses, phone numbers or bank accounts, and replaces it with artificial identifiers. The idea is that in this way the network can be shared and studied safely. However, it has been shown that in a social network pseudonymization alone is insufficient to protect privacy, as the structure of the network itself can be used to identify individuals [10]. As a result, even pseudonymized network data can pose privacy risks.

The reason that nodes can be identified, is that nodes in a network can be structurally unique. The signature of a node is the value of some measure that describes the node. If a node's signature is unique among all nodes, it can be identified by anyone who knows that signature. Therefore, these measures represent so-called attacker scenarios, where the attacker is assumed to have access to that structural information and uses it to identify nodes.

Even simple measures such as the degree of a node, which is the number of connections it has, can be sufficient to uniquely identify a node. More expressive measures such as the Vertex Refinement Query (VRQ) [3], which captures the degree sequence of a node's neighbours, further increase the number of uniquely identifiable nodes. In this thesis, the DEGREE and VRQ measure are used as signatures that are used as attacker scenarios. These measures are specific to networks.

Once a node is identified, meaning its real-world identity becomes known, this information can propagate: neighbors that are unique based on some measure become identifiable as well, which potentially enables further identifications. This iterative propagation of identification is what we call the *cascading effect*. The node from which this propagation starts is called the seed node. For example, suppose an attacker uses DEGREE as a measure in a social network. If Bob is the only person with exactly 3 friends, his corresponding node is uniquely identifiable and is used as the seed node. Once Bob is identified, the attacker examines his 3 neighbours. If one of those neighbours has a unique degree among Bob's neighbours, for example, one has degree 5 while the other two both have degree 2, then the neighbour with degree 5 is uniquely identifiable as well. This process then repeats for the newly identified node, potentially cascading through a large portion of the network.

Despite the well-established work on network de-anonymization [1, 2, 5, 9, 10, 14], the cascading effect has not been studied as a phenomenon in its own right. Understanding the cascading effect is important because it shows the re-identification risk a single identified node can have. Prior work has focused on single-step identification, but has not characterized how far and how broadly a single seed node can cascade through a network, nor addressed how many seed nodes are sufficient

to de-anonymize an entire network through cascading alone. The latter is what we call the seed set identification problem for network de-anonymization. This is important as it shows the vulnerability of networks against cascading de-anonymization attacks, since it tells us the worst-case effect of knowing the identity of just one node, and the minimum number of node-identities an attacker needs in order to fully de-anonymize a network.

This thesis addresses these gaps with the goal of understanding the vulnerability of pseudonymized networks to cascading de-anonymization. We do so through the following research question and subquestions:

- How strong is the cascading effect in real-world social networks?
  - How much of the network can be de-anonymized by cascading from one seed node?
  - How can we quantify the number of seed nodes needed in order to de-anonynimze the entire network?

To answer these questions, we develop a cascading algorithm that computes for each node in a network the cascade set, i.e., the set of all nodes that become identifiable when that node is used as a seed. To solve the seed set identification problem and find how many nodes we need to identify all nodes, we propose a new approximation algorithm that finds an upperbound on the minimum number of seed nodes needed to de-anonymize the entire network. The algorithm constructs a directed *cascade graph* to which we then apply Kosaraju's algorithm to find strongly connected components. We prove that nodes in the same strongly connected component have identical cascading effects, and that the number of source components gives an upper bound on the number of seeds needed. Both the cascade sets for each node, their size distributions, and the seed set identification problem are analyzed across seventeen real-world networks of different types and sizes, using DEGREE and VRQ as attacker scenarios, with and without twin node handling. Twin nodes are sets of nodes connected to exactly the same neighbours, making them structurally identical. Since twin nodes reveal identical structural information, it does not matter which twin is which for the purpose of cascading. By treating them as a group, cascading can continue through them, potentially exposing a larger fraction of the network than would otherwise be possible. Not accounting for twin nodes therefore underestimates the true vulnerability of a pseudonymized network.

The rest of this thesis is structured as follows; Section 2 discusses related work; Section 3 provides necessary background; Section 4 describes the methods that are used in the experiments; Section 5 describes the experiments; Section 6 describes the results of the experiments; Section 7 discusses the results; Section 8 concludes, identifies limitations and explores ideas for future work.

2 Related Work

Work on structural de-anonymization of networks, showed that removing identifying information from a social network is insufficient to guarantee privacy [10]. They developed a re-identification algorithm based on network topology and showed that it could identify a substantial fraction of users in an pseudonymized Twitter graph by using a network from Flickr as background knowledge. Their algorithm works by first identifying a small set of seed nodes whose neighbourhoods match across the two graphs, and then propagating identification outward. This work shows that network structure

alone can be enough to de-anonymzize a network. Other work studied attacks on pseudonymized social networks [1]. The attacks rely on the propagation of identity information through local neighbourhood structure, which is closely related to the cascading mechanism we study. However, their work focuses on specific targeted attacks requiring either colluding users or active injection of nodes, whereas this thesis studies cascading re-identification assuming that a node's identity is known.

The concept of k-anonymity [14] not per se related to graphs, provides a formal framework for measuring privacy in data where person-specific information is removed, but the remaining data can still be used to identify the individual. An individual is k-anonymous if at least $k-1$ other individuals are equivalent based on the remaining data. When $k=1$, a person is unique and can be identified. This framework is extended to graphs, for example by defining degree-based k-anonymity and studying how graph structure can be modified to achieve it [9]. The cascading effect studied in this thesis is based on k-anonymity. A node is k-anonymous when there are at least $k-1$ other nodes that are equivalent to it based on some anonymity measure. As a result the value of k could be 1, which means that the node is uniquely identifiable. This can possibly trigger further cascading.

The work most closely related to this thesis is that of de Jong et al. [5], who introduced the anonymity-cascade algorithm that makes use of k-anonymity, and studied the infectiousness of uniqueness in networks. Their research focused primarily on whether cascading occurs and on the total fraction of nodes that become identifiable. No attention has been given to how cascade sizes are distributed across nodes, what the influence of a single node can be, and how many independently identified nodes are sufficient to cover the entire network. These are important in real-world scenarios where an attacker may have access to only a single node identity, or where the goal is to fully de-anonymize a network with minimal effort.

The seed set identification problem studied in this thesis is related to the influence maximization problem [2], where the goal is to find a minimum set of nodes whose effect covers the entire network. However, there are important differences. Influence maximization is a probabilistic spreading process where each edge activates with some probability. Whereas cascading re-identification in this thesis is deterministic. A node is identifiable if and only if it is structurally unique among its neighbours at the time of identification. This determinism allows us to compute cascade effects exactly and to find an upper bound on the minimum seed set.

To our knowledge, no prior work has studied the cascading effect in network de-anonymization as a phenomenon in its own right, leaving open the questions of how cascade sizes are distributed, how much a single node can cascade, how attacker scenarios compare in terms of privacy risk, and how many seed nodes are needed to de-anonymize an entire network, which are explored in the remainder of this thesis.

3 Background

This chapter introduces the concepts underlying the cascading re-identification framework developed in this thesis. First, the relevant graph theory terminology is introduced. Then, the notion of anonymity in networks is discussed, followed by the structural measures used for node identification. Finally, foundations for the algorithms used in this thesis are presented.

3.1 Networks

Networks are a connected structure of relevant interacting entities, such as traffic networks that model roads and intersections, computer networks that model devices and connections, biological networks that model proteins and their interactions, and social networks that model people and their relationships.

3.1.1 Graphs

A way to represent a network is as a graph. A graph $G = (V, E)$ consists of a set of vertices V and a set of edges E, where each edge connects two vertices. Entities are represented as vertices and the connections between entities are represented as edges.
A graph can be either undirected or directed. An undirected graph has edges representing two-way, mutual relationships. An edge between node u and v with $u, v \in V$ is unordered and represented as $\{u, v\}$. The edges can be traversed both directions. A directed graph has edges with specific orientations, indicating one-way relationships. An edge between node u to node v is represented as an ordered pair (u, v). Where u is the source and v is the target. The edge can only be traversed in the direction of the arrow.

Traversing an edge means moving from one node to an adjacent node along that edge. Graph traversal is the process of visiting nodes in a graph by repeatedly traversing edges, starting from an initial node, until all reachable nodes have been visited. A path in a graph is a sequence of nodes $v_1, v_2, \ldots, v_k$ such that there exists an edge between each consecutive pair of nodes $(v_i, v_{i+1}) \in E$. The length of a path is the number of edges traversed. In a directed graph, a path must follow the direction of the edges. A shortest path is a path between two nodes with the shortest length possible.

A characteristic of real-world networks is the emergence of a giant component (GC). This is the largest connected component that occupies a large portion of the nodes. The remainder of the network consists out of smaller connected components. A connected component is a set of nodes where each node can be reached by every other node in the component. When edge directionality is introduced in a network, there are two types of connected components. A weakly connected component (WCC) is a maximal set of nodes such that there exists a path from every node to every other node in the set, if the direction of the edges is entirely ignored. Ignoring the direction of the edges means that we treat the network as undirected. A strongly connected component (SCC) is a maximal set of nodes such that there exists a directed path from every node to every other node in the set. Nodes in a SCC can be contracted into one node. Contracting each SCC into a single node results in a directed acyclic graph (DAG), called the condensation. A source SCC is a SCC with no incoming edges in the condensation DAG, meaning it cannot be reached from any other SCC.

3.1.2 Network Metrics

Several metrics can be derived from the structure of a graph. From now on, we assume undirected graphs, since our data only contains undirected graphs. We use a directed graph in Section 4 for our *cascade graph*.
At the node level, the **direct neighborhood** of a node v is the set of all nodes directly connected to it:

$$N(v) = \{u \in V \mid \{u, v\} \in E\} \tag{1}$$

The degree of a node v is the number of its direct neighbours, and is defined as:

$$\deg(v) = |N(v)| \tag{2}$$

The **clustering coefficient** of a node v measures the fraction of pairs of neighbours of v that are also connected to each other:

$$cc(v) = \frac{|\{(u, w) \in E \mid u, w \in N(v)\}|}{|N(v)|(|N(v)| - 1)/2} \tag{3}$$

A high clustering coefficient indicates that the neighbourhood of v is highly connected. The **average clustering coefficient** is the mean of $cc(v)$ over all nodes in the network and serves as a global measure of local connectivity.

At the global level, the **density** of a graph is the fraction of possible edges that are present:

$$\text{density} = \frac{2|E|}{|V|(|V| - 1)} \tag{4}$$

A density close to 0 indicates a sparse graph, while a density close to 1 indicates a dense graph. The **degree variance** measures how much node degrees vary across the network:

$$\sigma^2_{deg} = \frac{1}{|V|} \sum_{v \in V} (\deg(v) - \mu_{deg})^2 \tag{5}$$

where μ_{deg} is the average degree. A low degree variance indicates that most nodes have similar degrees, while a high degree variance indicates that some nodes have many more connections than others. Finally, the **diameter** of a graph is the length of the longest shortest path between any two nodes:

$$\text{diam}(G) = \max_{u,v \in V} d(u, v) \tag{6}$$

where $d(u, v)$ is the shortest path length between u and v. A large diameter indicates that some nodes are far apart in the network, meaning cascades must traverse many intermediate nodes to reach them.

3.1.3 Kosaraju's Algorithm

Kosaraju's algorithm finds all strongly connected components of a directed graph in linear time $O(|V| + |E|)$.

3.2 Anonymity Measures

Based on the structure of a network, individuals can be identified. Each anonymity measure corresponds to a specific attacker scenario, as it represents the structural information an adversary is assumed to have access to. Nodes are partitioned into equivalence classes based on the chosen measure, where nodes with the same signature belong to the same class. The signature of a node v is the value of that measure, denoted $M(v)$. We can determine node anonymity based on its equivalence class. A node is k-anonymous if it is equivalent to at least $k - 1$ other nodes. When $k = 1$, the node is the only member of its equivalence class and is therefore uniquely identifiable. For $k > 1$, the node is considered to be anonymous.

3.2.1 Degree

Degree is the number of direct neighbours of a node. When DEGREE is used as a measure by the attacker, a nodes signature it denoted by:

$$M_{deg}(v) = deg(v) = |N(v)| \tag{7}$$

This measure requires the least amount of information and is the least computationally demanding [3].

3.2.2 VRQ

Vertex Refinement Query (VRQ) is the degree sequence of the neighbors of a node. This measure is related to DEGREE, since the length of the sequence equals the degree of the node. When VRQ is used as a measure by the attacker, a node's signature it denoted by:

$$M_{VRQ}(v) = \{deg(v) \mid u \in N(v)\} \tag{8}$$

This attacker scenario is stronger than DEGREE. This attacker model assumes that an adversary can observe or infer local degree information about the direct neighborhoud of a node and uses it to identify unique nodes. This identification strategy has been shown to be computationally efficient and effective in de-anonymization [3].

3.2.3 Twin nodes

Twin nodes are sets of nodes that are connected to the same set of neighbors, and have the same signature. Therefore, they are structurally the same, which means they reveal the same information. It does not matter for the further identification of nodes who is who, since they reveal the same information. A node is twin-unique if all members of its equivalence class are twin nodes of each other. For more details, see [5].

3.3 Cascading Effect

If one node is identified, it has influence on the anonymity of its direct neighbors. The cascading is done level by level, where each level builds on the nodes newly identified in the preceding one. Each cascade level corresponds to the distance from the initial node to the nodes in that level. Let $U \subseteq V$ denote the set of nodes newly found to be uniquely identifiable at the previous cascade level, and let $I \subseteq V$ be the set of already identified nodes. A direct neighbor $u \in N(v)$ of the identified node v will be identified as well if this node has a unique signature $M(u)$ among the other unidentified neighbors of v. To formalize this, $C(v, I)$ denotes the set of unidentified neighbors of v that have a unique signature within $N(v)$:

$$C(v, I) = \{u \in N(v) \setminus I \mid \nexists w \in N(v) \setminus I : M(u) = M(w)\} \tag{9}$$

Then we let $C(U, I)$ denote the set of neighbours of all nodes in U that have a unique signature among their unidentified neighbours.

$$C(U, I) = \bigcup_{u \in U} C(u, I) \tag{10}$$

These unique neighbors potentially cause their neighbors to be unique as well, which potentially enable further identifications. This iterative propagation of identification is what we call the cascading effect. The cascading effect can be defined as:

$$C_{l_{max}}(U, I) = (C(C_{l_{max}-1}(U, I) \setminus C_{l_{max}-2}(U, I), C_{l_{max}-2}(U, I)) \cup C_{l_{max}-1}(U, I)) \setminus \{v\} \quad (11)$$

Here, U denotes the set of nodes newly identified at the previous cascade level:

$$U = C_{l-1}(U, I) \setminus C_{l-2}(U, I) \quad (12)$$

Here, I denotes all the identified nodes until level l:

$$I = C_{l-1}(U, I) \quad (13)$$

Here, $(C(C_{l_{max}-1}(U, I) \setminus C_{l_{max}-2}(U, I), C_{l_{max}-2}(U, I))$ are the nodes identified in the new cascade layer, and $C_{l_{max}-1}(U, I)$ are all the nodes that are identified by cascading in the previous layers. The base cases are:

$$C_0(v, I) = \{v\} \quad (14)$$

$$C_{-1}(v, I) = \emptyset \quad (15)$$

Cascading stops when no unique neighbors can be found. The depth of the cascade is finite and can be denoted by l_{max}.

$$l_{max} = \min\{l \in \mathbb{N} \mid C_l(U) = C_{l+1}(U)\} \quad (16)$$

The node v from which the cascading effect starts, is what we call the *seed node*. It is not included in the cascading effect, since it is not identified by another node.

3.3.1 Cascade Set

The cascade set of a node v is defined as the set of all nodes that become identified when v is used as the initial seed. For a node v, its cascade set is defined as:

$$S(v) = C_{l_{max}}(U, I) \cup \{v\} \quad (17)$$

Each node has its own, possibly not unique, cascade set.

3.4 Equivalence relation

An equivalence relation is a binary relation, denoted by $\sim$. This groups elements that share a specific property in the same set. Elements that are part of the same set are considered equal. A relation between two elements is an equivalence relation if it satisfy three properties: reflexivity ($a \sim a$), symmetry ($a \sim b \Rightarrow b \sim a$), and transitivity ($a \sim b$ and $b \sim c \Rightarrow a \sim c$). Elements belonging to the same equivalence class are considered equal with respect to the relation. In 4.2.3 we use the equivalence relation in a proof.

4 Methods

This section describes the algorithms developed to analyze the cascading effect in networks. First, the cascading algorithm is described, which computes the cascade size and set for each node. Then, the cascade graph construction and seed node identification algorithms are described, which computes an upper bound on the minimum number of seed nodes needed to de-anonymize the entire reachable network.

4.1 Cascading Algorithm

Before cascading begins, a preprocessing step is performed when twin node handling is taken into account. Twin nodes, nodes connected to exactly the same set of neighbours, are identified and grouped together. Within each group, twin nodes are merged into the same equivalence class. If all nodes in an equivalence class are twin nodes of each other, they can be treated as one identifiable unit during cascading.

Algorithm 1 computes, for each node in the graph the complete set of nodes that become identifiable when v is used as the initial seed node. It saves the cascading set S for each seed node v, which include all the nodes that are identified by knowing the identity of the seed node, and the seed node itself. The seed node is included in order to make it easier to see if nodes have the same effect or not. The algorithm is implemented using a Breadth First Search (BFS) approach, expanding the cascade level by level.

The algorithm iterates over every node in L as an initial seed node. For now, assume $L = V$. For each seed v, a BFS is performed, where I ensures that each node is visited at most once per seed. Then $C(U, I)$ computes the unique neighbors of node all nodes in U, by checking if the unvisited neighboring nodes are not in the same equivalence partition. The unique neighbors are added to I, and are being processed further. The cascade terminates when no new uniquely identifiable neighbours can be found, for the considered seed node.

Algorithm 1: CascadeAll(G, L)

Input: Graph G
Output: S for all nodes

foreach $v \in L$ **do**
  $S(v) \leftarrow \emptyset$;
  $C_{-1}(U, I) \leftarrow \emptyset$;
  $C_0(U, I) \leftarrow \{v\}$;
  $S(v) = C_{l_{max}}(U, I) \cup \{v\}$
return S;

Complexity: For each of the number of nodes in the giant component $|V_{GC}|$, used as seed nodes, the BFS visits each node and each edge at most once, giving a worst-case complexity of $O(|V| \cdot (|V| + |E|))$.

4.2 Seed Set Identification

CascadeAll (Algorithm 1) computes the cascade set $S(v)$ for every node v if $L = V$. However, the cascade sets by themselves do not directly reveal how many independent seed nodes are needed to

de-anonymize the entire network. The goal of seed set identification for network de-anonymization is to find the smallest set of initially identified nodes $L \subseteq V$. In other words, it asks to find a set L^* of minimum size:

$$L^* \in \arg\min_{L \subseteq V} \{|L| : \text{CascadeAll(G, L)} = V_{GC}\} \tag{18}$$

This problem is related to the set cover problem, which is a combinatorial optimization problem that involves finding the minimum size subset of sets that covers a given finite set. In this case the set of nodes that are part of the giant component V_{GC} is the finite set we want to cover. The cascading sets are the subsets of this set $\{S(v)|v \in V_{GC}\}$. We want to select the minimal number of subsets in order to cover the entire network. The minimal number of subsets are equal to the number of seed nodes needed in order to de-anonymize or cover the entire network. Since set cover is NP-hard, calculating an exact solution is unfeasible for large networks. The BuildCascadeGraph 2 and FindSeedSets 3 algorithms together compute an upper bound on this minimum in polynomial time.

4.2.1 Conditional and Unconditional Cascading

Before describing the algorithms, it is important to distinguish between two types of cascading, since the seed identification approach is built on this distinction. A node v is unconditionally identifiable from u if v is the only member of its equivalence class among the neighbours of u, regardless of which other nodes have already been identified. This means the cascade edge from u to v always holds, independent of cascade order. A node v is conditionally uniquely identifiable from u if v becomes the only member of its equivalence class among the unvisited neighbours of u, after some neighbouring nodes have already been identified. In other words, if v gets identified, depends on the prior identification of other nodes in the cascade. Unconditional cascading is deterministic, whereas conditional cascading is order-dependent. Conditional cascading identifies at least the nodes identified by unconditional cascading. The CascadeAll 1 algorithm is based on conditional and unconditional cascading, since it performs a BFS and updates the set of unvisited neighbours at each step. The BuildCascadeGraph 2 and FindSeedSets 3 algorithms are based on unconditional cascading only. This is because unconditional edges represent identification relationships that hold regardless of cascade order. Conditional cascading is the reason why the seed node fraction computed by FindSeedSets 3 is an upper bound rather than an exact minimum.

4.2.2 Building the Cascade Graph

To identify which nodes require independent seeds, the initial undirected graph is transformed in a directed cascade graph $G_{cascade} = (V, E_{cascade})$. A directed edge $(u, v) \in E_{cascade}$ exists if and only if v is unconditionally uniquely identifiable from u. This means that v can always be identified by cascading from node u. This results in a directed graph that represents the unconditional cascading relationships between nodes, independently of any particular cascade order.

Algorithm 2: BUILDCASCADEGRAPH(G)

Input: Graph G

Output: Directed cascade graph $G_{cascade}$

```
E_cascade ← ∅;
// For each node create an edge with its unique neighbors
foreach v ∈ V do
    foreach u ∈ C(v, I) do
        E_cascade ← E_cascade ∪ {(v, u)};
return G_cascade = (V, E_cascade);
```

For each node v, $C(v, I)$ is computed with an empty identified set $I = \emptyset$, since unconditional cascading is performed. Each uniquely identifiable neighbour v of u results in a directed edge (u, v) in $E_{cascade}$.

Complexity: $O(|E|)$, since each edge of the undirected graph can be traversed twice in the worst case scenario.

4.2.3 Strongly Connected Components and Cascading Equivalence

Kosaraju's algorithm is applied to $G_{cascade}$ to find all strongly connected components (SCCs), based on unconditional cascading. We define the relation $\sim$ on nodes in $G_{cascade}$ as: $v \sim w$ if and only if v and w belong to the same SCC. We show that $\sim$ is an equivalence relation by verifying the three required properties:

- **Reflexivity:** $v \sim v$, since every node can reach itself, and thus belongs to its own SCC.
- **Symmetry:** if $v \sim w$, then v and w are in the same SCC, meaning there is a directed path from v to w and from w to v. By the same reasoning, $w \sim v$.
- **Transitivity:** if $v \sim w$ and $w \sim x$, then v can reach w and w can reach x, so v can reach x. By the same reasoning, x can reach v. Therefore v, w and x are all reachable from each other.

Since $\sim$ is an equivalence relation, nodes in the same equivalence class are interchangeable. The following theorem shows that this interchangeability extends to the unconditional cascading effect.

Theorem 1. *$v \sim w$ if and only if $S_u(v) = S_u(w)$.*

Here, $S_u(v)$ is the unconditional cascade set of a node v.

Proof. To prove that the unconditional cascade sets are equal, we must demonstrate mutual containment of the sets, namely that $S_u(v) \subseteq S_u(w)$ and $S_u(w) \subseteq S_u(v)$.

First, assume $v \sim w$. By definition, v and w belong to the same strongly connected component in $G_{cascade}$, which implies there exists a directed path from v to w and a directed path from w to v.

To prove $S_u(v) \subseteq S_u(w)$, let x be an arbitrary node in $S(v)$. This means there exists a path from v to x. Since $v \sim w$ guarantees a directed path from w to v, we can concatenate the path from w to v with the path from v to x to construct a path from w to x. Consequently, any unconditional cascade initiated by w will identify v, and thereby trigger identification of x. Thus, $x \in S_u(w)$, which shows that $S_u(v) \subseteq S_u(w)$.

To prove $S_u(w) \subseteq S_u(v)$, let y be an arbitrary node in $S_u(w)$, meaning there is a path from w to y. Because $v \sim w$ guarantees a directed path from v to w, we can concatenate the path from v to w with the path from w to y to construct a path from v to y. Consequently, any unconditional cascade initiated by v will identify w, and thereby trigger the de-anonymization of y. Thus, $y \in S_u(v)$, which shows that $S_u(w) \subseteq S_u(v)$.

Combining both subset relations, we conclude that If $v \sim w$, then $S_u(v) = S_u(w)$.

Conversely, assume $S_u(v) = S_u(w)$. By definition, $v \in S_u(v)$, since v trivially identifies itself at the start of its own cascade. Because $S_u(v) = S_u(w)$, it follows that $v \in S_u(w)$, which means there exists a directed path from w to v.

By the same reasoning, $w \in S_u(w) = S_u(v)$, so $w \in S_u(v)$, which means there exists a directed path from v to w.

Since there is a directed path from v to w and a directed path from w to v, v and w are mutually reachable, and therefore belong to the same strongly connected component. By definition of $\sim$, this means $v \sim w$.

Combining both directions, we conclude that $v \sim w$ if and only if $S_u(v) = S_u(w)$. □

The converse holds as well. Two nodes with the same unconditional cascade set $S_u(v) = S_u(w)$ satisfy $v \sim w$, since the cascade graph $G_{cascade}$ only captures unconditional cascading edges. The cascading effect is performed while taking conditional cascading into account as well. Conditional cascading happens when a node becomes uniquely identifiable after some of its neighbours have already been identified. Which can cause nodes in different SCCs to produce identical cascade sets.

This result implies that the number of distinct unconditional cascading sets are at most the number of SCCs. For unconditional cascading, all nodes within the same SCC are interchangeable as entry points, since identifying any one of them produces the same unconditional cascade.

4.2.4 Source SCCs as an Upper Bound on Seed Nodes

A source SCC is an SCC with no incoming edges in the condensation DAG, meaning it cannot be triggered by unconditional cascading from any other component.

Theorem 2. *The number of source SCCs is an upper bound on the minimum number of seed nodes needed to de-anonymize the entire reachable network.*

Proof. Each source SCC requires at most one independently identified seed node, since no cascade from outside can unconditionally reach it. Conversely, identifying one representative node from each source SCC triggers unconditional cascades that reach all SCCs reachable from that source in the condensation DAG. Therefore at most one seed per source SCC is needed, giving the upper bound. □

Theorem 3. *If twin nodes are not taken into account, twin nodes form their own source SCC*

When twin nodes are not taken into account, each twin node does not have a unqiue signature, since a group of twin nodes share the same signature. Therefore twin nodes can not be unconditionally reached through cascading by their neighboring nodes. Therefore twin nodes will not have any incoming arrows, which means they will form a SCC by itself, which is a source SCC.

4.2.5 FindSeedSets Algorithm

The FindSeedSets algorithm 3 builds the cascade graph, identifies source SCCs with Kosaraju's algorithm, collects the cascade effect of one representative per source SCC, and removes redundant effects. It returns the seed node fraction, which is the percentage of the network that needs to be identified in order to de-anonymize the entire network.

This seed node fraction is an upper bound of the minimum percentage of seeds needed. The true minimum may be lower if some seed sets overlap in coverage, because of conditional cascading. This means that fewer seed nodes could suffice in practice. Computing the exact minimum is an NP-hard set cover problem.

Algorithm 3: FindSeedSets(G)

Input: Graph G
Output: Set of seed nodes corresponding to seed sets L

$G_{cascade} \leftarrow$ BuildCascadeGraph(G);
$\mathcal{S} \leftarrow$ Kosaraju($G_{cascade}$);
`// Assign each node to its SCC and initialize source flags`
foreach $\mathcal{S}_i \in \mathcal{S}$ **do**
  foreach $v \in \mathcal{S}_i$ **do**
    $scc\_id[v] \leftarrow i$;
  $is\_source[i] \leftarrow$ `true`;
`// Mark SCCs with incoming edges from other SCCs as non-source`
foreach $(u, v) \in E_{cascade}$ **do**
  if $scc\_id[u] \neq scc\_id[v]$ **then**
    $is\_source[scc\_id[v]] \leftarrow$ `false`;
`// Collect cascade sets of source SCCs`
$L \leftarrow \emptyset$;
foreach $\mathcal{S}_i \in \mathcal{S}$ **do**
  if $is\_source[i]$ **then**
    $x \leftarrow$ representative node of $\mathcal{S}_i$;
    $L = L \cup \{x\}$
`// Remove non-maximal and duplicate sets`
$S(L) \leftarrow \{\mathcal{X} \in S(L) \mid \forall \mathcal{X}' \in S(L) : \mathcal{X} \subseteq \mathcal{X}' \Rightarrow \mathcal{X} = \mathcal{X}'\}$;
return L;

After Kosaraju's algorithm assigns each node to an SCC, source SCCs are identified by marking any SCC that receives an incoming edge from another SCC as non-source. One representative node per source SCC is selected and its cascade set is collected into $\mathcal{S}$. The subset elimination step retains only the sets which are not fully contained in any other set: a seedset $\mathcal{X}$ is kept if it is not contained in any other seedset $\mathcal{X}'$, formally $\forall \mathcal{X}' \in S(L) : \mathcal{X} \subseteq \mathcal{X}' \Rightarrow \mathcal{X} = \mathcal{X}'$. These sets are removed because any coverage provided by a set is already provided by another set containing it. $|L|$ represents the upper bound of the number of seed sets needed, where each element corresponds to a seed node from a source SCC. It is an upper bound since there can still be seed sets that are partially contained in other seed sets, because of conditional cascading. Therefore the number of seed sets found is an approximation of the minimal number of seeds needed.

Complexity: $O(|E|)$ for building the cascade graph, $O(|V| + |E|)$ for Kosaraju's algorithm, $O(|V| + |E|)$ for source SCC identification, and $O(k^2 \cdot n)$ for subset elimination, where k is the number of source SCCs and n is the average cascade set size. The total complexity is dominated by the subset elimination step: $O(k^2 \cdot n)$.

5 Experimental Setup

This section describes the experimental setup used to evaluate the cascading effect across seventeen real-world networks. First, the datasets and their properties are presented. Then, the four configurations under which each dataset is evaluated are described. Finally, the metrics used to quantify the cascading effect are defined, followed by a description of the implementation.

5.1 Datasets

In total, 17 real-world anonymous network datasets are included. These datasets are selected since they have varying characteristics. All networks are undirected. Since the cascading effect requires connectivity between nodes, all analysis are restricted to the giant component of each network. From now on the term network refers to the giant component in the next chapters.

Table 1: Properties of the giant component per dataset after preprocessing. Type is the type of network. $|V_{GC}|$ and $|E_{GC}|$ are the number of nodes and edges in the giant component. Density is computed as $2|E_{GC}|/(|V_{GC}|(|V_{GC}| - 1))$. Deg. var. is the degree variance of the giant component. Avg. CC is the average clustering coefficient. Diam. is the diameter.

Dataset	Type	$\lvert V_{GC}\rvert$	$\lvert E_{GC}\rvert$	Density	Deg. var.	Avg. CC	Diam.	
CA-GrQc	Collaboration	4,158	13,422	0.0016	74.41	0.557	17	[8]
calls	Communication	347	477	0.0079	4.33	0.181	22	[12]
fb_friends	Social	800	6,418	0.0201	179.12	0.315	7	[12]
arenas-email	Communication	1,133	5,451	0.0085	87.23	0.220	8	[7]
netscience	Collaboration	379	914	0.0128	15.42	0.741	17	[7]
dnc-temporal	Communication	1,833	4,366	0.0026	328.93	0.216	8	[7]
gene_fusion	Biological	110	124	0.0207	13.21	0.002	9	[7]
maayan-faa	Infrastructure	1,226	2,408	0.0032	13.47	0.068	17	[7]
moreno_health	Social	2,539	10,455	0.0032	18.59	0.147	10	[7]
moreno_innov.	Social	117	465	0.0685	15.98	0.219	5	[7]
powergrid	Infrastructure	4,941	6,594	0.0005	3.21	0.080	46	[7]
radoslaw_email	Communication	167	3,250	0.2345	993.84	0.592	5	[7]
euroroad	Infrastructure	1,039	1,305	0.0024	1.44	0.019	62	[7]
primaryschool	Contact	242	8,317	0.2852	706.01	0.526	3	[13]
sms	Communication	457	628	0.0060	3.33	0.159	20	[12]
socfb-Reed98	Social	962	18,812	0.0407	1254.53	0.318	6	[11]
socfb-Simmons81	Social	1,510	32,984	0.0290	1288.53	0.317	7	[11]

5.2 Configurations

Each dataset is evaluated under four configurations, combining the choice of anonymity measure with the twin node setting:

Table 2: The four experimental configurations applied to each dataset.

Configuration	**Measure**	Twin nodes
DEGREE	DEGREE	No
DEGREE + TWINS	DEGREE	Yes
VRQ	VRQ	No
VRQ + TWINS	VRQ	Yes

The DEGREE configuration represents the weakest attacker scenario, requiring the least structural knowledge. VRQ represents a stronger attacker with access to direct neighborhood information. Enabling twin nodes extends each configuration by treating structurally identical nodes as the same, allowing cascading to propagate through them. For each dataset, each configuration is visualized per metric to allow direct comparison across settings.

5.3 Implementation

The cascading algorithm and the seed node identification algorithm are implemented building on an existing C ++ codebase developed by de Jong [4], which provides functions related to anonymity in networks. This codebase is extended with the implementation of the cascading algorithm described in Chapter 4, as well as the BUILDCASCADEGRAPH and FINDSEEDSETS algorithms for seed node identification. The seed node fraction computation is split across two components: the cascade graph construction and SCC identification are performed in C ++, while the subset elimination and deduplication steps are performed in Python. Besides this, Python NetworkX [6] is used for graph loading and giant component extraction.

6 Results

In this section, the results of the experiments are described. The results are reported per dataset and per configuration, allowing comparison between DEGREE and VRQ and between the twin and non-twin settings.

6.1 Distribution of Cascade Sizes

For each node in the giant component, the cascade size $|S(v)|$ is computed and normalized by $|V_{GC}|$. We use $|S(v)|$ since the cascade set of node v $S(v)$ also includes the seed node, which is not identified by the cascading effect. However, we want to see how much of the network can be identified by cascading from one node. Therefore we also include the seed node, since otherwise the entire network could never be fully de-anonymized. The cascade distributions reveal patterns that

are grouped into 7 groups. For each group, a representative figure is shown.

Group 1: fb_friends, moreno_innovation, radoslaw_email, primaryschool, socfb-Reed98, socfb-Simmons81
All four configurations de-anonymize close to 100% of the giant component, with a single seed node. A single unique set combined with 100% identification means that cascading from any node in the network will de-anonymize the entire giant component. In other words, the starting point for cascading is irrelevant. These networks have relatively to the other networks a high density, ranging from 0.0201 (fb_friends) to 0.2852 (primaryschool). Twin nodes consistently ensure that at least the VRQ + TWIN configuration has a single unique set where 100% of the network is de-anonymized. DEGREE + TWIN shows the same behavior, except for fb_friends, moreno_innovation, socfb-Simmons81 where it is close to collapsing into one set that identifies 100%. In general, cascading is effective regardless of measure or twin configuration.

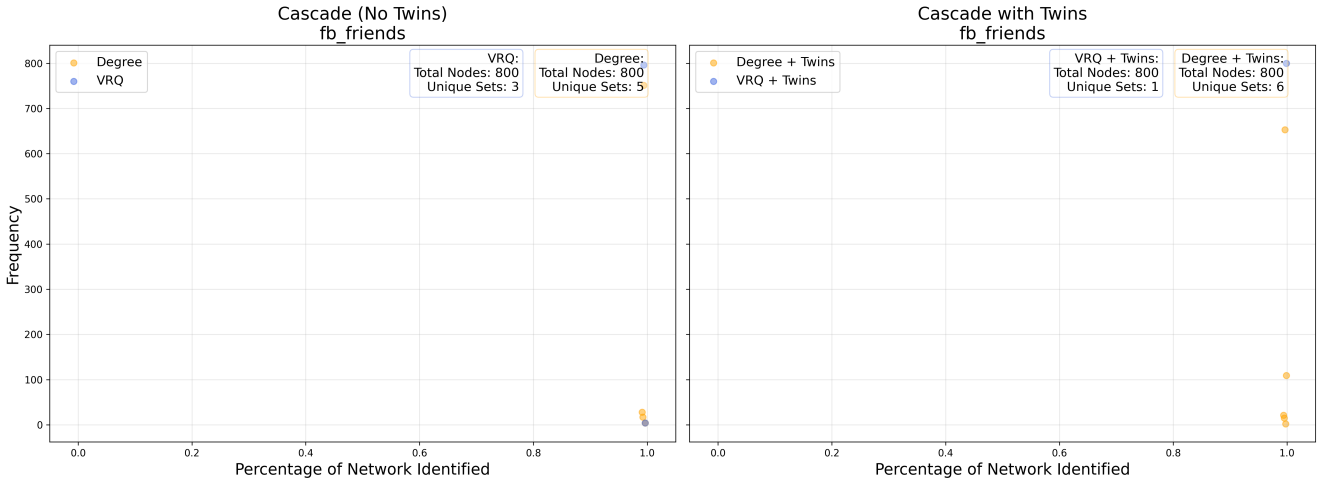

Figure 1: Cascade size distribution for fb_friends, representative of Group 1. All configurations de-anonymize close to 100% of the giant component with one to few unique cascade sets.

Group 2: moreno_health, arenas-email
Like the previously discussed datasets, all four configurations de-anonymize close to 100% of the giant component with cascading. One difference is that twin nodes ensure that VRQ identifies the entire network with any seed node, while DEGREE produces 71 (moreno_health) and 883 (arenas-email) unique sets that de-anonymize 95% to close to 100% of the network.

The cascading effect is strong regardless of measure, but VRQ reveals that most nodes lead to identical cascades sets. This means that while an adversary using DEGREE would observe many different re-identification sets depending on the seed node, an adversary using VRQ would more likely reach the same set of nodes regardless of where they start. Besides this, when twin nodes are taken into account, VRQ can de-anonymize the entire network, no matter from which node the cascading starts.

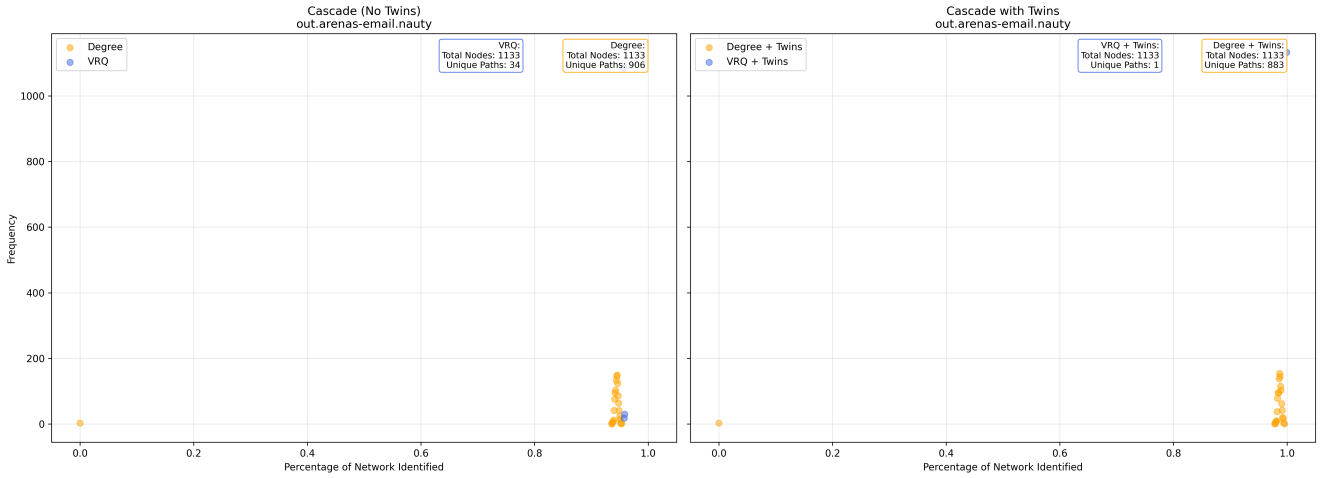

Figure 2: Cascade size distribution for arenas-email, representative of Group 2. All four configurations de-anonymize close to 100% of the giant component. With twin nodes taken into account, VRQ collapses to a single unique cascade set covering the entire network, while DEGREE produces multiple unique sets ranging from 95% to 100% coverage.

Group 3: CA-GrQc, sms, maayan-faa

Cascade sizes are concentrated near zero and near a larger cascade size, with almost nothing in between, showing a bimodal distribution. VRQ outperforms DEGREE visibly in all four datasets. With twins, the distribution shifts rightward for both measures, except for some nodes that have little to no cascading effect.

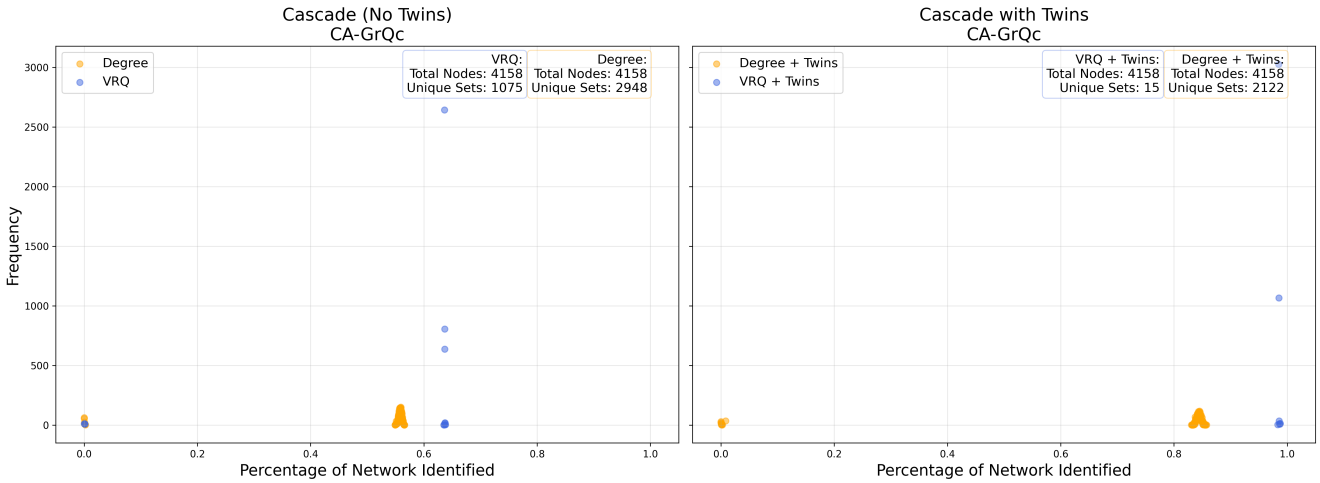

Figure 3: Cascade size distribution for CA-GrQc, representative of Group 3. A clear bimodal pattern is visible without twins, where the relatively higher values shift rightward when twins are taken into account.

Group 4: dnc-temporal

Without twins, DEGREE and VRQ cascades are concentrated around 0.2 and 0.3 respectively. With twins taken into account, the entire distribution shifts dramatically rightward to 0.8 and 1.0 This distinguishes dnc-temporal from the previous networks, where points are split between relatively small and large values for both measures.

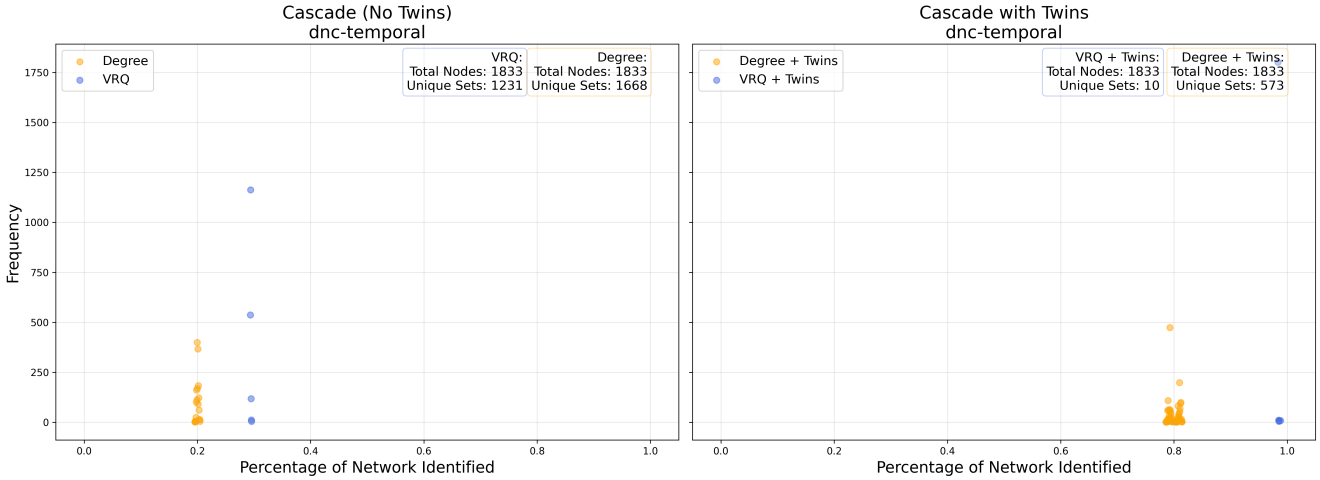

Figure 4: Cascade size distribution for dnc-temporal, the only member of Group 4. Sizes are clustered around one value for both measures without twins, this distribution shifts entirely rightward when twins are taken into account.

Group 5: calls, netscience, gene_fusion

Cascade sizes are spread across many values, with a weak bimodal distribution. When twins are taken into account, cascades grow noticeably larger except for some nodes that have little to no cascading effect, just as in group 3. Enabling twin nodes does not influence the spread. The cascading effect for DEGREE grows, but is not close to de-anonymize the entire network from a single node. For VRQ the cascading effect is close to de-anonymize the entire network. However, only gene_fusion VRQ + TWINS collapses to a single unique set, where cascading from any node achieves the identification of the entire GC.

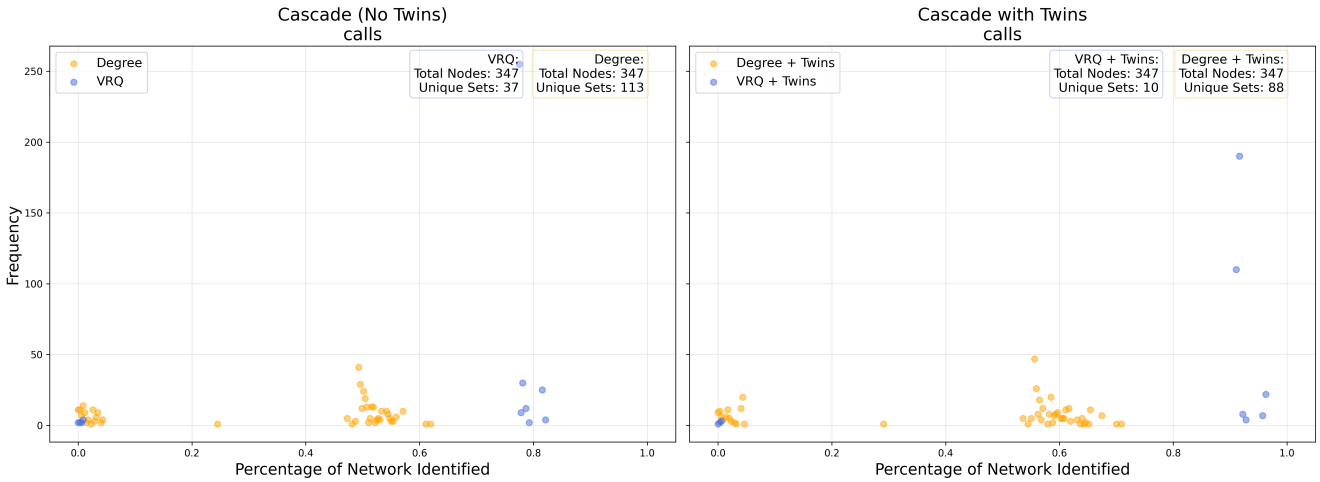

Figure 5: Cascade size distribution for calls, representative of Group 5. Cascade sizes are spread across many values without twins, but show a weak bimodal distribution. Accounting twin nodes shifts cascades rightward for both measures, except for some nodes with little to no cascading effect, and the spread remains. VRQ with twins approaches full network identification

Group 6: powergrid

This is the most distinctive result across all datasets. DEGREE cascades are spread across many small sizes. Without twins DEGREE achieves cascades concentrated around 35%, with twins pushing this to approximately 40%. Without twins VRQ achieves cascades concentrated around 78%, with twins pushing this to approximately 90%. For DEGREE and VRQ all points either stay at the same place, or they shift rightward when twin nodes are taken into account. The number of unique sets decreases for both measures. This suggests that cascade sets get extended with twin nodes and sometimes morphed together with other sets. The fact that DEGREE does not de-anonymize as much of the network with one node, can be because of the relatively low degree variance of 3.21, which suggests that nodes have similar degrees. Therefore it is harder for nodes to be unique based on DEGREE. VRQ does not suffer from this problem, since it takes the structure of the direct neighborhood of a node into account.

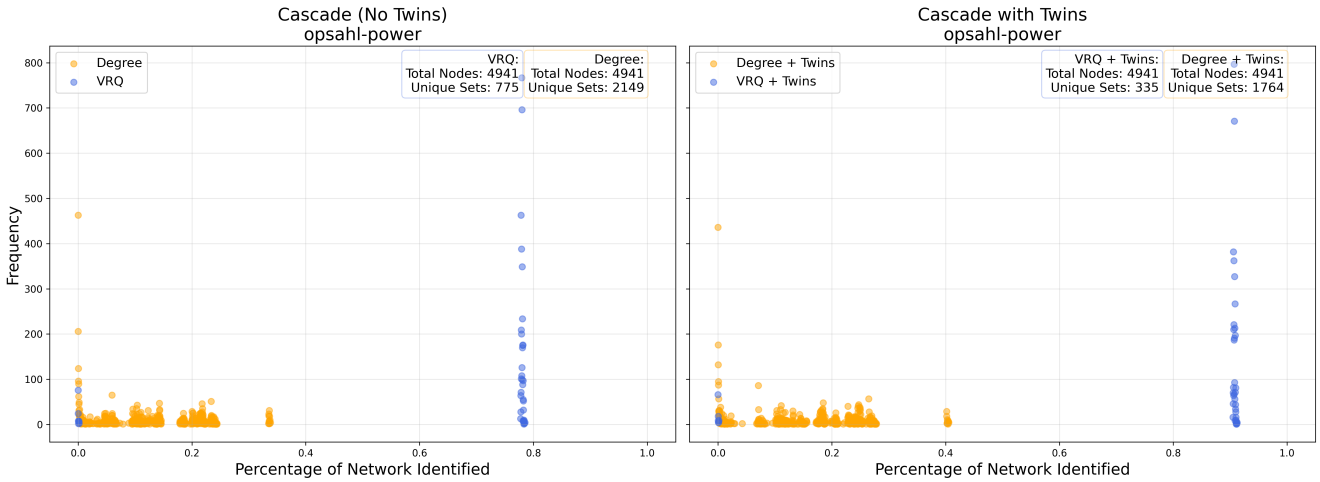

Figure 6: Cascade size distribution for powergrid, the only member of Group 6. Without twins, DEGREE cascades are maximally around 35% and VRQ maximally around 78% of the giant component. Enabling twin nodes pushes these to approximately 40% and 90% respectively.

Group 7: euroroad

Cascade sizes are concentrated near zero and near a larger cascade size, with almost nothing in between, resembling group 3. One difference with group 3 is that the sizes are more spread around certain values, just as in group 5. The difference with group 3 and group 5 is that both with and without twins the distributions look nearly identical for both measures. Enabling twin nodes barely gives a benefit. The relatively low degree variance of 1.44 explains the large gap between DEGREE and VRQ in the percentage of the GC de-anonymized by one node, for the same reasons as for group 6.

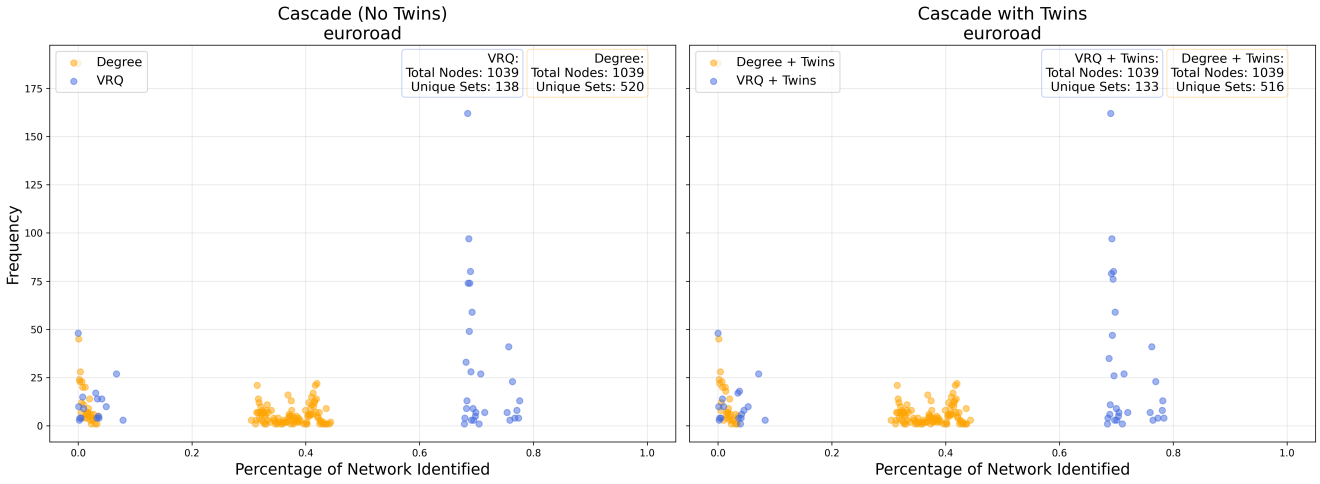

Figure 7: Cascade size distribution for euroroad, the only member of Group 7. Cascade sizes concentrate near zero and near a larger value, with little in between. Enabling twin nodes provides minimal benefit, and distributions remain nearly identical with and without twins for both measures.

6.1.1 Overall results

In nearly all datasets, the VRQ measure outperforms the DEGREE measure in terms of de-anonymization of the giant component with cascading from a single seed node. However, the difference is in most cases not that significant.

When twins are being taken into account, it consistently improves the identification percentages and reduces the number of unique sets. This shows that twin nodes enable cascade sets to merge together. For example, if you have x twin nodes that are structurally the same, they would usually produce x different sets. When twin nodes are taken into account, this just becomes one set. However, the difference between DEGREE and VRQ does not seem to become smaller, when twin nodes are taken into account.

6.2 Largest Cascade Set

The largest cascade set is defined as:

$$S^* = \max_{v \in V_{GC}} |S(v)| \tag{19}$$

and is reported as a fraction of $|V_{GC}|$. This fraction represents the maximum fraction of the giant component that a single node can identify through cascading. Figure 8 shows this fraction per dataset across all four configurations. It captures the maximum re-identification risk a single node can pose, representing the worst-case cascading scenario. The largest set is additionally visualized directly on the network. Nodes that are re-identified through the anonymity measure (DEGREE or VRQ) are coloured red, and nodes that are re-identified through twin node uniqueness are coloured green. Nodes that are not identified remain grey. This allows direct visual comparison between configurations and gives insight into which parts of the network are most vulnerable to cascading re-identification.

For all seventeen datasets, with cascading from a single node DEGREE can de-anonymize at least 18.2% without twin nodes and 40.6% with twin nodes, of the giant component. VRQ can de-anonymize at least around 29.6% of the giant component without twin nodes, and 78.8% with twin nodes. In 9 datasets cascading with a single node suffices to de-anonymize the entire network.

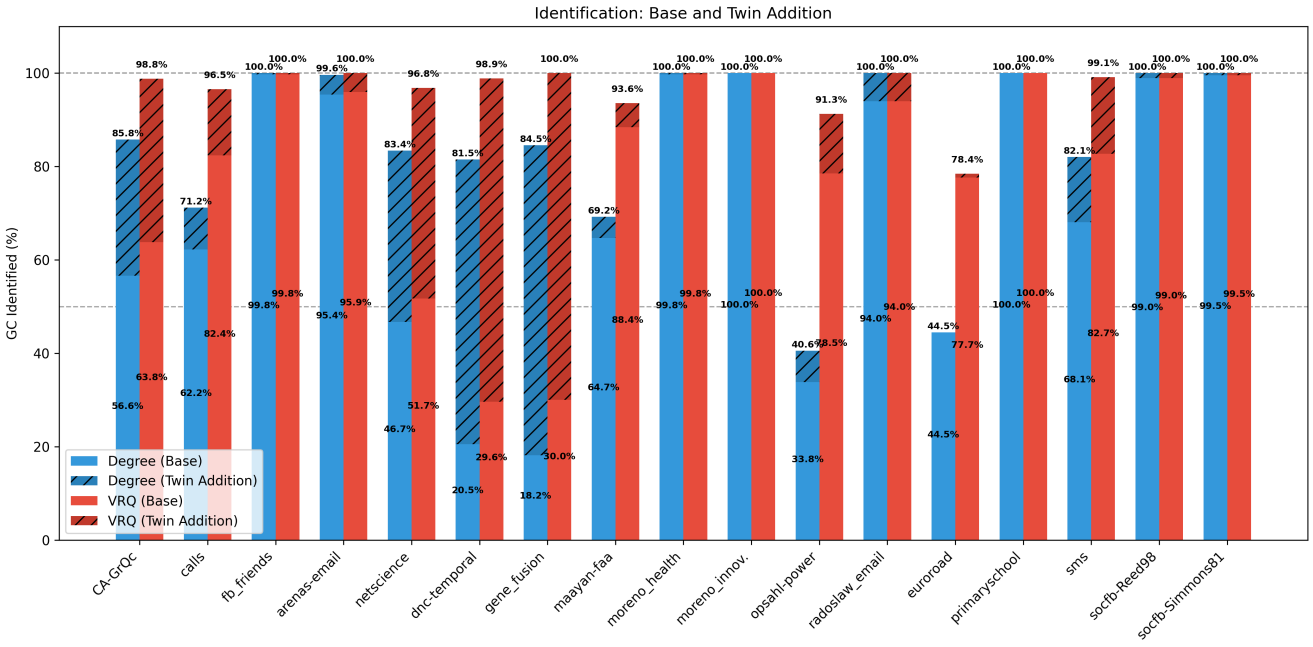

Figure 8: The percentage of nodes from the giant component de-anonymized by a single seed node, shown for all four configurations across all seventeen datasets.

For almost every dataset, the difference in the fraction of the network de-anonymized by cascading between VRQ and DEGREE is relatively small. The fraction of the network that VRQ de-anonymizes with cascading is equal to or higher than that de-anonymized with cascading by DEGREE. With the exception of powergrid and euroroad, which show the largest gap between the two measures. As mentioned before, these two networks both have a relatively small variety of degrees, which can be seen by the degree variance values in table 1.

Twins substantially increase the largest cascading set. The largest increases are in CA-GrQc, netscience, dnc-temporal and gene_fusion. The gene_fusion network shows the power of twin node identification most clearly. For both measures twin nodes increase the cascading effect for DEGREE from 20% to 73% and for VRQ from 33% to 77%. However, twin nodes do not close the gap between DEGREE and VRQ identification.

The results are in line with the results from the cascading distribution.

6.3 Seed Node Fraction

The seed node fraction is defined as $|L^*|/|V_{GC}|$, where $|L^*|$ is the minimal number of seed nodes found by the FINDSEEDSETS algorithm 3, and $|V_{GC}|$ is the number of nodes in the giant component. It represents an upper bound of the minimal fraction of nodes in the giant component needed as seed nodes to de-anonymize the entire network with cascading. It is computed using the FINDSEEDSETS algorithm 3, which is an approximation algorithm for the underlying NP-hard set cover problem. Because the algorithm only accounts for unconditional cascading when identifying source SCCs, the true minimum number of seeds may be lower. A lower fraction indicates a more vulnerable network, where fewer node identities are needed to trigger cascades that cover the entire giant

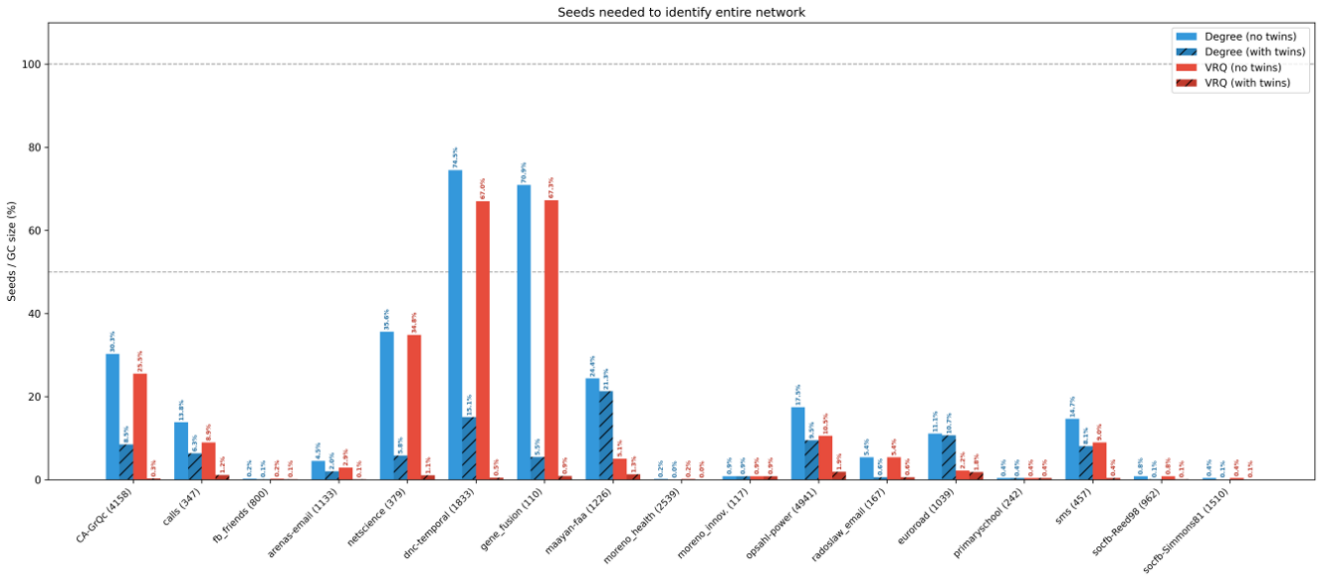

Figure 9: The percentage of nodes from the giant component needed in order to de-anonymize the entire giant component, shown for all four configurations across all seventeen datasets.

component. Figure 9 shows this fraction per dataset across all four configurations.

Across all seventeen datasets, VRQ consistently requires fewer or equal seed nodes compared to DEGREE. This difference is most visible for the mayaan-faa network where without twin nodes DEGREE needs 24.4% of the GC and VRQ needs 5.1% of the GC in order to de-anonymize the entire GC with cascading. When accounting for twin nodes DEGREE needs 21.3% of the GC and VRQ needs 1.3% of the GC in order to de-anonymize the entire GC with cascading. This confirms that VRQ is a more powerful identification measure, the fact that is takes more structural information into account, enables cascades to propagate further from each seed, reducing the number of seeds needed to cover the network.

Accounting for twin nodes consistently reduces the fraction of seeds required. The most dramatic reductions occur in, for example, dnc-temporal where DEGREE drops from 74.5% to 15.1% and VRQ from 67.0% to 0.5%, and gene_fusion where DEGREE drops from 70.9% to 5.5% and VRQ from 67.3% to 0.9%. The networks that benefit most from twins in the largest cascade set metric (Figure 8) are the same networks that show the largest reductions in seed fraction here.

fb_friends, moreno_health, moreno_innovation, primaryschool, socfb-Reed98 and socfb-Simmons81 require fewer than 1% in order to de-anonymize the entire GC with cascading, regardless of measure and if twin nodes are taken into account or not. These results are consistent with Group 1 and Group 2 from the cascade size distribution analysis, where one node is able to de-anonymize the entire network with cascading. All other networks CA-GrQc, calls, arenas-email, netscience, dnc-temporal, gene_fusion, mayaan-faa, powergrid, euroroad and sms require fewer than 1.9% in order to de-anonymize the GC but only when VRQ is used in combination with twin nodes. When DEGREE is used in combination with twin nodes the percentage of seeds needed ranges from 2% to 21.3%.

6.3.1 Overall results

The above results are strongly consistent with the cascade size distribution and the largest set findings. Networks where a single node has a large cascade that is almost as big or as big as the entire network, (Figure 8) are the same networks that require few seeds here. Networks that benefit the most from twin nodes in the largest cascade set metric (Figure 8) are the same networks that show the largest reductions in seed fraction here. When DEGREE and VRQ are used as measures without twin node handling the networks differ in vulnerability. As soon as twin nodes are taken into account the networks become more vulnerable. This can be explained by Theorem 3. Especially when VRQ and twin nodes are used as a measure. Then an attacker needs at most 1.9% of the giant component in order to de-anonymize the entire giant component. This shows that using VRQ + TWIN makes a network very vulnerable for attackers, regardless of the type of network.

7 Discussion

7.1 How strong is the cascading effect in networks?

The results demonstrate that the cascading effect is strong in most real-world networks. For all seventeen datasets, a single node can de-anonymize more than 78% of the giant component under at least one configuration, and in 9 datasets a single node suffices to de-anonymize the entire network with cascading. This shows the extend of the re-identification risk caused by the cascading effect. The strength of the cascading effect is determined by which measure is used, and the amount of twin nodes. Besides that, it seems that the strength is related to the variety of degrees in the network. Where more variety creates a larger effect.

7.2 How much of the network can be de-anonymized by cascading from one node?

7.2.1 Cascading Effect of Degree and VRQ

Both DEGREE and VRQ are effective cascade measures in most networks, but their effectiveness varies considerably across network types. In networks such as primaryschool, socfb-Reed98, and fb_friends, both measures de-anonymize close to 100% of the giant component by cascading from a single seed, with or without twins. This suggests that these networks probably have a more diverse degree sequence, since DEGREE performs similar to VRQ.

Networks that have a cascade size distribution which are bimodal or spread, suggests that the starting position of the seed node matters. It could be that nodes in structurally divers parts of the network trigger large cascades while nodes in structurally homogeneous parts do not. Notably, for the networks we analyzed, the cascade size distribution is independent of the initial node's own properties such as degree or centrality. It seems that the cascading effect is a property of the network structure surrounding the seed rather than the seed itself.

7.2.2 Difference between Degree and VRQ

VRQ consistently outperforms DEGREE across all seventeen datasets and all configurations.

If a network does not have a variant degree sequence it it harder to identify nodes based on their DEGREE. While VRQ's neighborhood information provides enough structural discrimination to cascade effectively. In general, this means that an adversary with access to only degree information is less of a threat than an adversary with access to VRQ.

The difference between DEGREE and VRQ is also visible in the number of unique cascade sets. VRQ consistently produces fewer unique sets than DEGREE, for example from 906 to 34 in arenas-email and from 2948 to 1075 in CA-GrQc. This means that under VRQ, most nodes lead the same cascade sets, making the re-identification risk more predictable across seed nodes. Under DEGREE, different seed nodes lead to more diverse cascade sets, meaning the risk depends more heavily on which node is identified.

Twin nodes consistently increase the largest cascade set for both measures, but the relative difference between DEGREE and VRQ remains approximately the same when twins are taken into account. Twins amplify the cascading effect without closing the gap between DEGREE and VRQ. The combination of VRQ with twins is therefore the most powerful configuration in every dataset, not because twins benefit VRQ more than DEGREE, but because VRQ already outperforms DEGREE and twins extend both measures by a comparable amount. Twin nodes also reduce the number of unique sets for both measures, showing that it merges sets together.

The gene_fusion network illustrates this most clearly, where enabling twins more than doubles the cascade reach for both measures, from 20% to 73% for DEGREE and from 33% to 77% for VRQ. The absolute gain is similar for both measures, and the gap between DEGREE and VRQ remains small with or without twins.

Twin nodes are mostly leaf nodes that extend an existing cascade by a small number of additional identifications. Without twin identification cascades that would otherwise merge, remain separate. With twin identification these cascade sets are merged, which results in less unique cascade sets.

7.3 How can we quantify the number of seed nodes needed to de-anonymize the entire network?

To quantify the number of seed nodes needed, we apply the FindSeedSets algorithm 3 which finds an upper bound on the minimum seed set by identifying source strongly connected components in the cascade graph.

Under VRQ + TWINS, all seventeen datasets require fewer than 1.9% of nodes as seeds to de-anonymize the entire giant component. This means that an adversary who can identify a small number of nodes can potentially de-anonymize the entire network through cascading alone.

The number of seeds needed is directly related to the number of source strongly connected components in the cascade graph. A network with few source components needs to identify less nodes, making it more vulnerable. The number of seed nodes needed in order to de-anonymize the entire network decreases significantly when twin nodes are taken into account. The reason for this is because of the fact that when twin nodes are not taken into account, each twin node has its own source SCC. After twin nodes are taken into account source SCC that share twin nodes merge together, which significantly decreases the number of source SCC and thereby the number of seed nodes needed. VRQ with twins consistently produces the fewest source components across all datasets, both by its detailed signature and by merging components through twin identification.

Since FINDSEEDSETS is an approximation algorithm that computes an upper bound, the true minimum fraction of seeds needed may be lower than reported, meaning real-world networks could

be more vulnerable than these results suggest.

8 Conclusions

This thesis investigated the cascading effect in networks across seventeen real-world datasets under four configurations: DEGREE, VRQ, DEGREE with and without twins. Three experiments were conducted to analyze the cascading effect: the distribution of cascade sizes, the largest cascade set, and the seed node fraction needed to de-anonymize the entire giant component.

The results show that the cascading effect is strong in most real-world networks. Even with DEGREE alone, a single node can de-anonymize almost 100% of the giant component for 8 out of the 17 datasets. This shows that cascading identification poses a serious privacy risk even under the weakest attacker scenario. VRQ and the use of twin nodes represent stronger attacker scenarios that consistently increase the cascading effect. VRQ + TWIN de-anonymizes the entire or almost the entire giant component from a single seed in several datasets. The individual properties of the seed node do not appear to influence the size of the cascade, the cascading effect seems to be determined by the structural properties of the network as a whole rather than by the structural properties of the node from which cascading starts.

The number of seeds needed to de-anonymize the entire network is small in most datasets. Under VRQ + TWINS, seven out of seventeen datasets require fewer than 1% of nodes as seeds. All datasets require fewer than 1.9% of nodes as seeds. Whereas for DEGREE + TWINS the datasets can require up to 21.3% of nodes as seeds.

Overall, our findings show that cascading re-identification is a serious threat to privacy in anonymized network data, even under weak attacker scenarios, if one selects the right seeds. A problem which can in practice be solved using our proposed algorithm.

8.1 Limitations

This thesis has several potential limitations. First, the seed node fraction computed by the FINDSEEDSET algorithm is an upper bound on the minimum number of seeds needed, not the exact minimum. The algorithm only considers unconditional cascading edges when building the cascade graph. Conditional cascading, where a node becomes uniquely identifiable only after some of its neighbours have already been identified, is not captured. Therefore, nodes that are not reachable through unconditional cascading from a seed node, could be reachable through conditional cascading, reducing the true minimum seed set below the reported upper bound.

Second, the analysis assumes that the attacker has access to structural information about the network, specifically DEGREE or VRQ. Of these, DEGREE is the most accessible attacker scenario, as it only requires knowledge of the number of connections of each node. VRQ requires knowledge of the degree sequence of a node's entire direct neighbourhood, which may be harder to obtain in practice. When interpreting the results it should be kept in mind that while VRQ consistently outperforms DEGREE, the DEGREE results represent the more realistic lower bound on re-identification risk, since substantially less knowledge is required.

Third, all datasets used are undirected networks. Directed and weighted networks may exhibit different cascading behaviour due to asymmetric neighbourhood relationships and edge weights providing additional structural information.

8.2 Future Work

The work in this thesis revealed areas where further research could be beneficial. First, structural network properties and its relationship with the cascading effect could be researched. In order to understand how network properties influence cascading vulnerability. This could be done by using theoretical graph models. Second, the exact minimum seed set could be approximated more precisely by taking conditional cascading into account. This would provide a better estimate of network vulnerability under cascading de-anonymization attacks than the upper bound reported in this thesis. Third, the cascading effect could be studied on directed, weighted, and temporal networks, where additional structural information may enable more cascading than in the undirected networks studied here. The interaction between cascading re-identification and anonymization defenses such as differential privacy could be studied to assess whether existing protections are sufficient to prevent cascading.

References

[1] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x? In *Proceedings of the 16th International Conference on World Wide Web*, pages 181–190, 2007.

[2] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 199–208, New York, NY, USA, 2009. Association for Computing Machinery.

[3] R. G. de Jong, M. P. J. van der Loo, and F. W. Takes. A systematic comparison of measures for k-anonymity in networks. *arXiv preprint arXiv:2407.02290*, 2024.

[4] Rachel G. de Jong. Anonymitycascade. https://github.com/RacheldeJong/Anonymitycascade, 2024. GitHub repository.

[5] Rachel G. de Jong, Mark P. J. van der Loo, and Frank W. Takes. The effect of distant connections on node anonymity in complex networks. *Scientific Reports*, 14, 2024.

[6] A. A. Hagberg, D. A. Schult, and P. J. Swart. Exploring network structure, dynamics, and function using networkx. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA, 2008.

[7] Jérôme Kunegis. KONECT: The koblenz network collection. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1343–1350, 2013. Accessed: 2026-02-20.

[8] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. https://snap.stanford.edu/data/, June 2014. Accessed: 2026-02-20.

[9] K. Liu and E. Terzi. Towards identity anonymization on graphs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 93–106, 2008.

[10] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[11] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015. Accessed: 2026-02-20.

[12] Piotr Sapiezynski, Arkadiusz Stopczynski, David Dreyer Lassen, and Søren Lassen Jørgensen. The Copenhagen networks study interaction data. Figshare. https://doi.org/10.6084/m9.figshare.7267433.v1, 2019. Accessed: 2022-05-01.

[13] SocioPatterns. Sociopatterns: Datasets. http://www.sociopatterns.org/datasets/, 2021. Accessed: 2026-02-20.

[14] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

Appendices

A Distribution of Cascade Sizes

A.1 Group 1

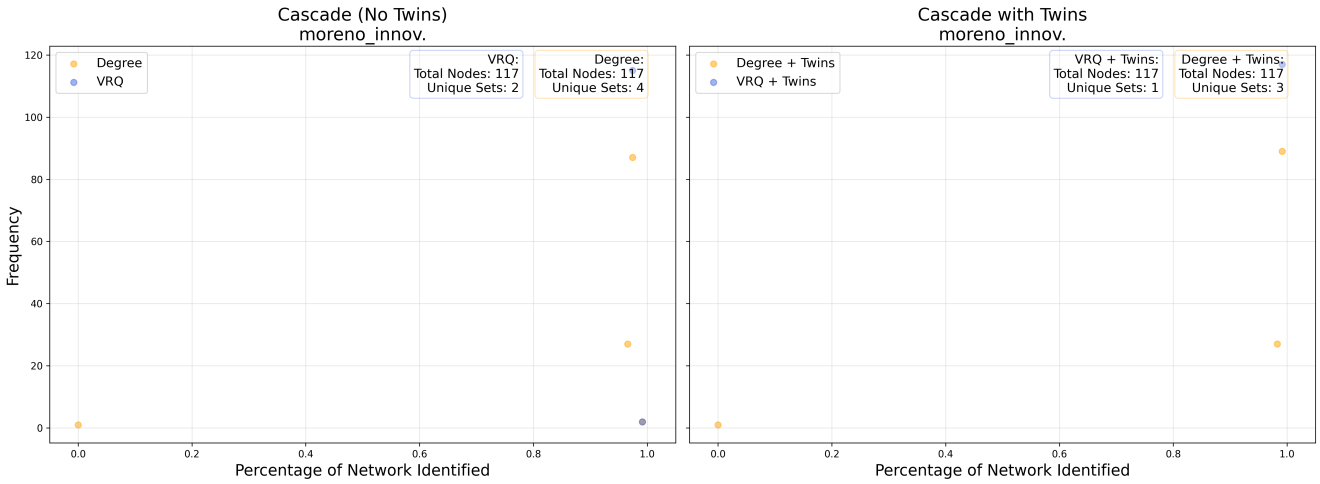

Figure 10: Cascade size distribution for moreno_innov., part of Group 1. All configurations de-anonymize close to 100% of the giant component with one to few unique cascade sets.

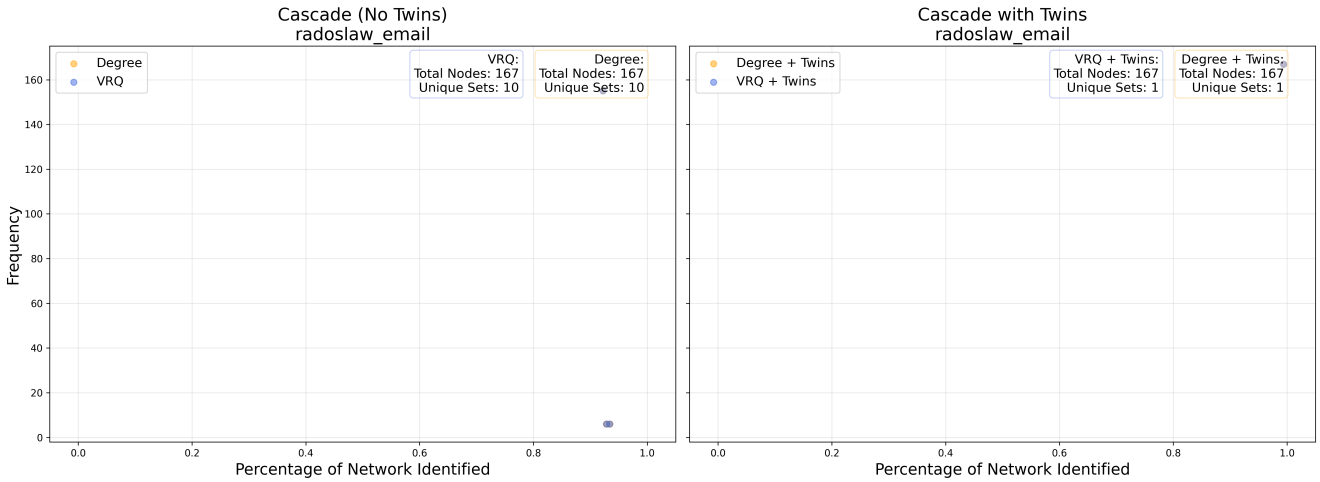

Figure 11: Cascade size distribution for radoslaw_email, part of Group 1. All configurations de-anonymize close to 100% of the giant component with one to few unique cascade sets.

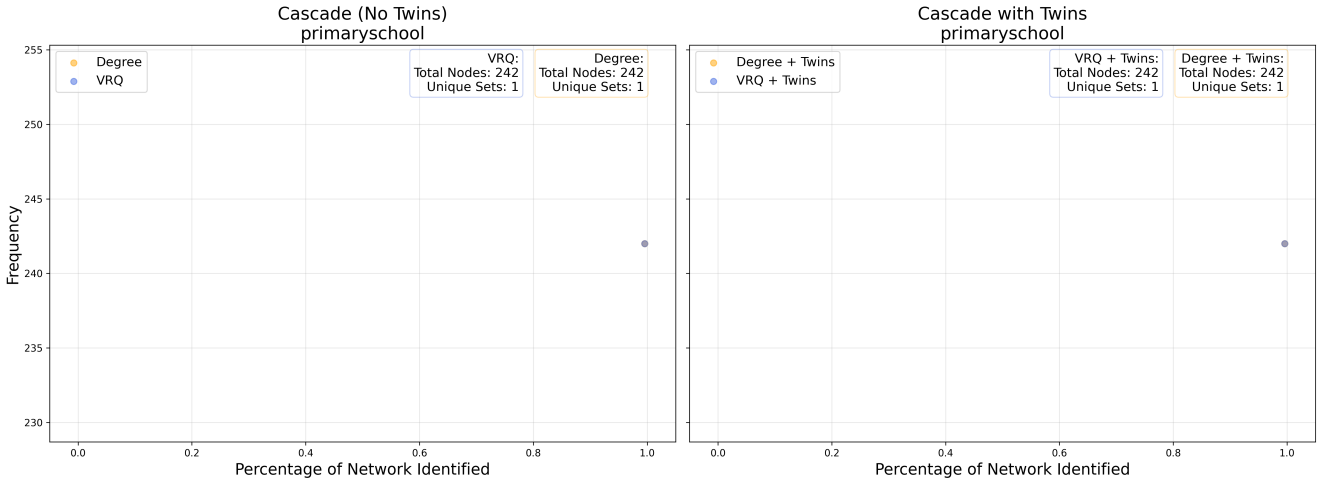

Figure 12: Cascade size distribution for primaryschool, part of Group 1. All configurations de-anonymize close to 100% of the giant component with one to few unique cascade sets.

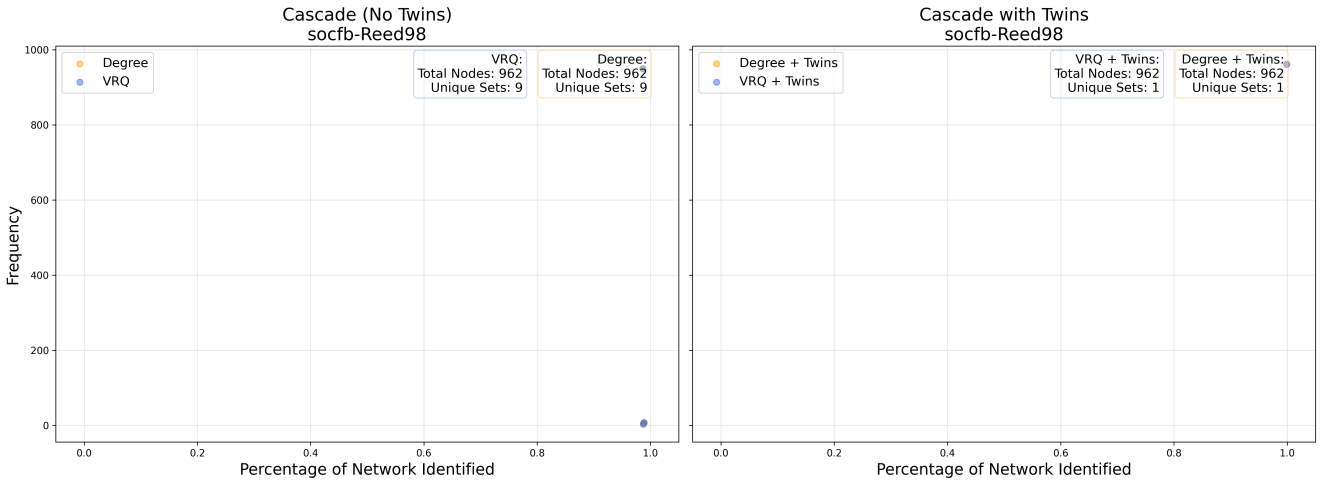

Figure 13: Cascade size distribution for socfb-Reed98, part of Group 1. All configurations de-anonymize close to 100% of the giant component with one to few unique cascade sets.

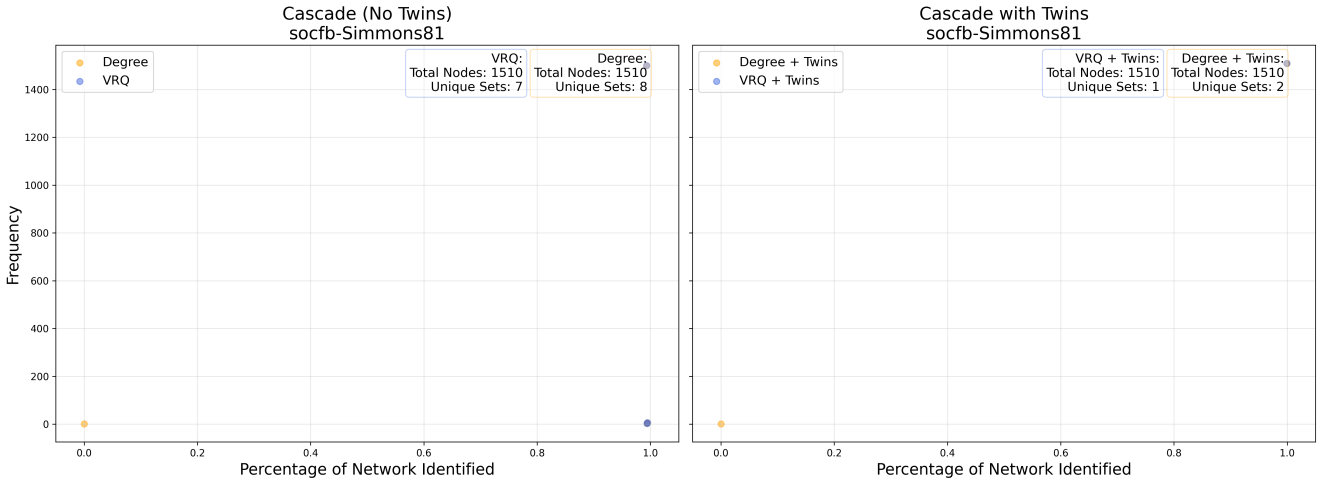

Figure 14: Cascade size distribution for socfb-Simmons81, part of Group 1. All configurations de-anonymize close to 100% of the giant component with one to few unique cascade sets.

A.2 Group 2

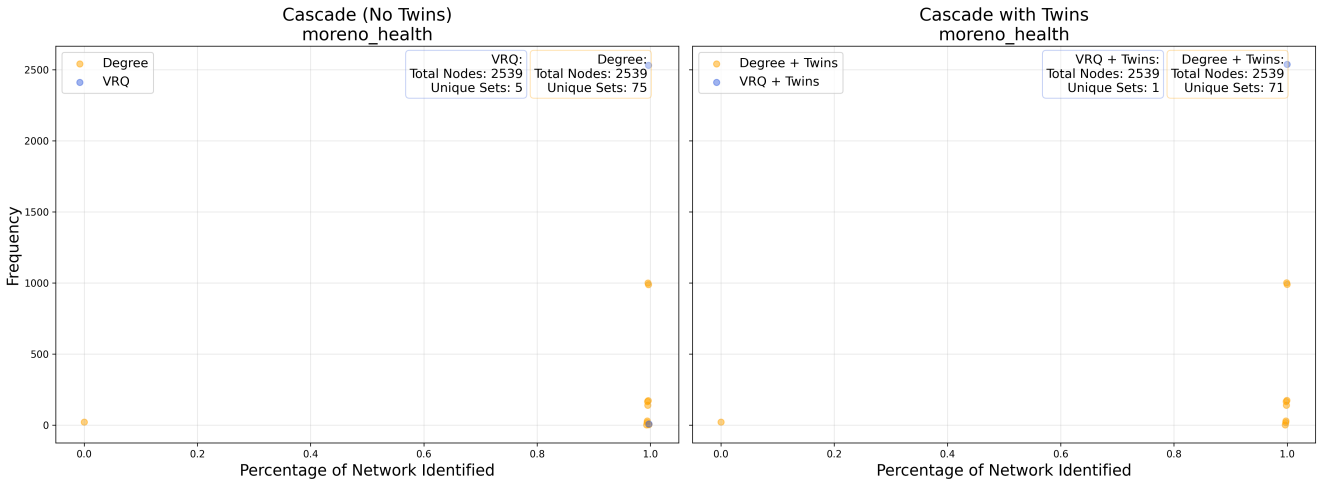

Figure 15: Cascade size distribution for moreno_health, part of Group 2. All four configurations de-anonymize close to 100% of the giant component. With twin nodes taken into account, VRQ collapses to a single unique cascade set covering the entire network, while DEGREE produces multiple unique sets ranging from 95% to 100% coverage.

A.3 Group 3

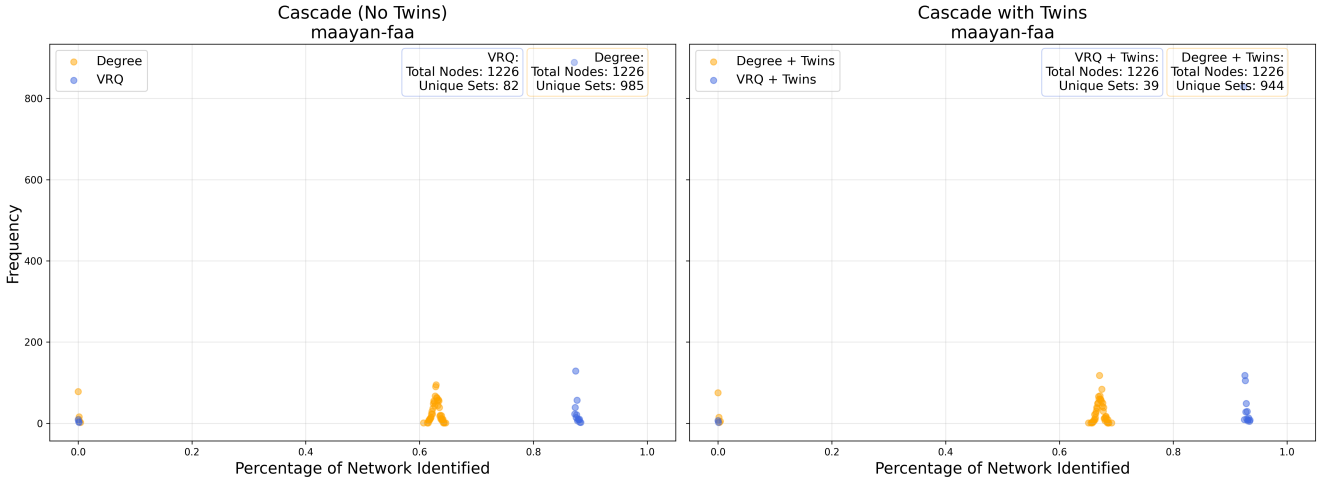

Figure 16: Cascade size distribution for mayaan-faa, part of Group 3. A clear bimodal pattern is visible without twins, where the relatively higher values shift rightward when twins are taken into account.

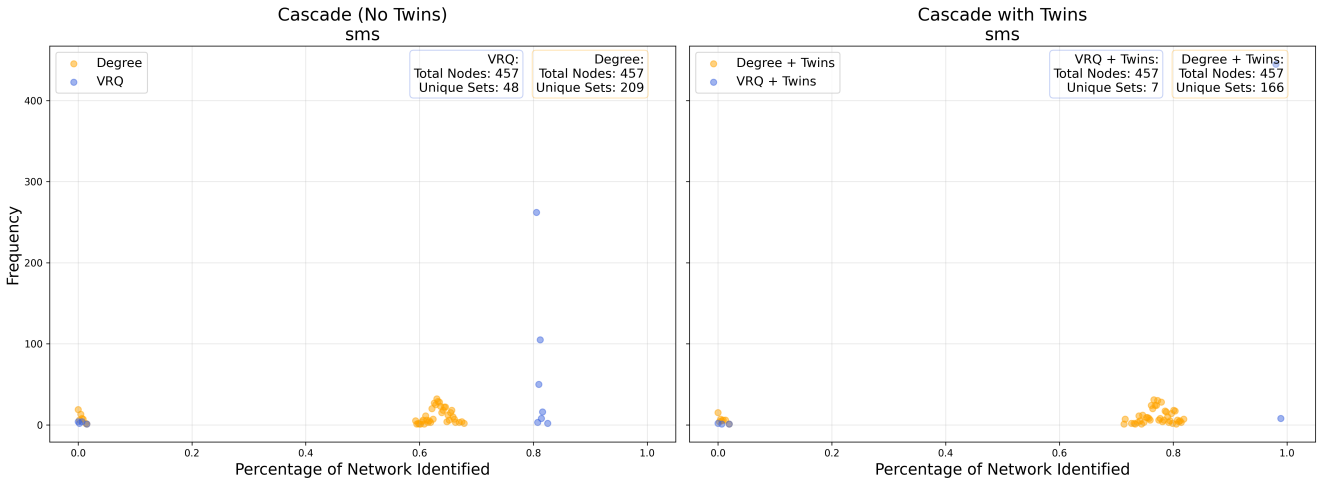

Figure 17: Cascade size distribution for sms, part of Group 3. A clear bimodal pattern is visible without twins, where the relatively higher values shift rightward when twins are taken into account.

A.4 Group 5

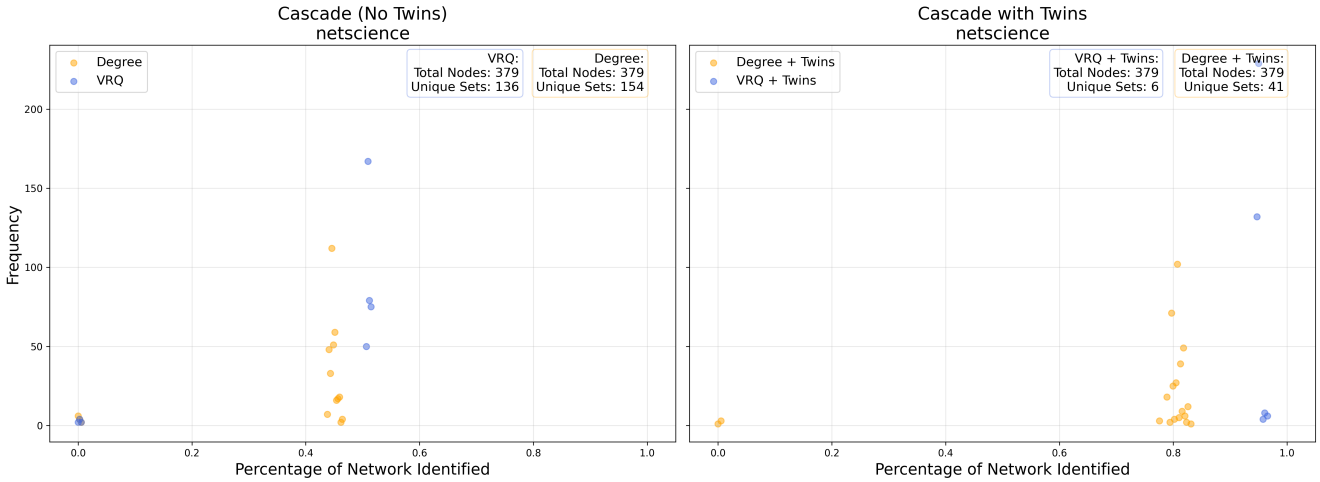

Figure 18: Cascade size distribution for netscience, part of Group 5. Cascade sizes are spread across many values without twins, but show a weak bimodal distribution. Accounting twin nodes shifts cascades rightward for both measures, except for some nodes with little to no cascading effect, and the spread remains. VRQ with twins approaches full network identification

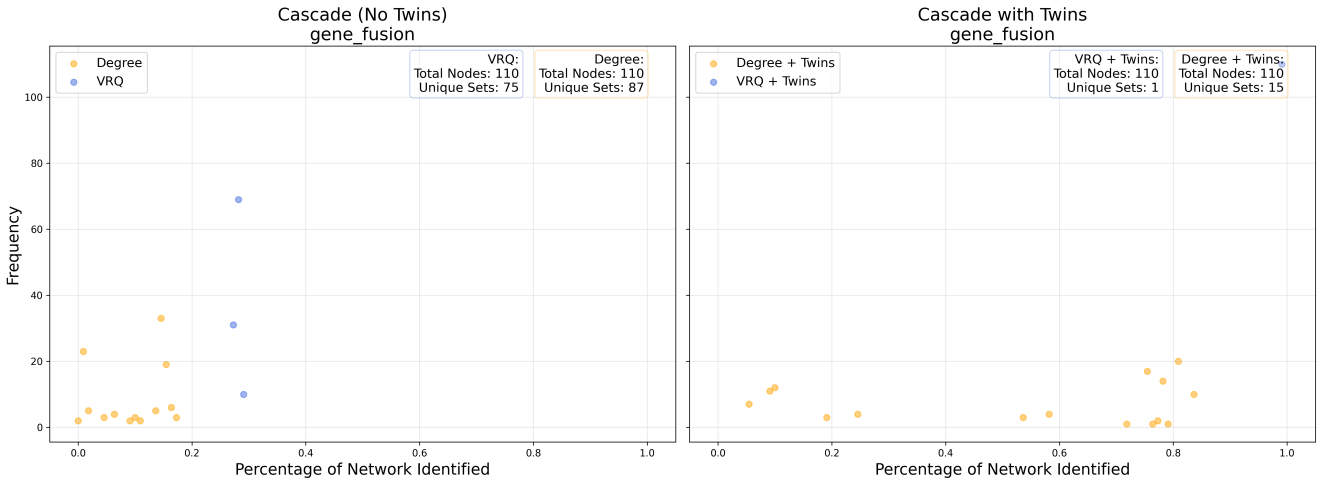

Figure 19: Cascade size distribution for gene_fusion, part of Group 5. Cascade sizes are spread across many values without twins, but show a weak bimodal distribution. Accounting twin nodes shifts cascades rightward for both measures, except for some nodes with little to no cascading effect, and the spread remains. VRQ with twins approaches full network identification