Universiteit Leiden

Master Computer Science

Anonymity in Temporal Networks

Name: Shaoxuan Zhang
Student ID: s3426505

Date: 30/08/2024

Specialisation: Data Science

Supervisors: Frank Takes

Mark van der Loo

Rachel de Jong

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Contents

Abstract

Social networks often contain sensitive information that needs to be protected against disclosure risks. Evaluating and preserving the anonymity of nodes in a network has become one of the top interests in network science. While previous studies focused mainly on preserving anonymity in static networks, there is a growing need to address such problems in temporal networks, in which the nodes and edges evolve over time. In this thesis, we first investigate the measurement of anonymity in temporal networks and the relationship between anonymity and network properties in a temporal context. Second, we propose a greedy perturbation algorithm that deletes edges that have the highest impact on the overall anonymity over time. Experiments using a node anonymity measure based on the number of nodes and edges in a node's ego network are performed. Results show how in real-world temporal social networks there is a high correlation between anonymity and the density of the network over time. Moreover, we find that nodes with higher centrality values are more likely to be less anonymous in temporal networks. Finally, we demonstrate to what extent the perturbation experiments are able to increase anonymity over time. The proposed approach can be applied to monitor and enhance overall node anonymity with the aim to better protect sensitive data about individuals in temporal social networks.

Chapter 1

Introduction

The rapid development and application of Web 2.0 technologies make it possible for people to connect with each other and contribute user-generated content through social platforms such as Twitter, Facebook and Instagram [25]. The number of users on Facebook has almost reached three billion and is continuously growing. The people and their social connections invisibly form networks, a type of data with nodes representing entities and edges representing connections. The connections between different nodes compose structural properties and patterns over the entire networks [38].

Real-world complex networks have shown that they are evolving and growing over time [24]. Nodes and edges emerging and disappearing constantly. Some networks record the temporal changes over time, such as person-to-person communication [10, 28, 26], disease spreading [17, 37], etc. By analyzing these temporal networks, we can find how different parts of the network connect with each other at different points in time. A common analysis approach of temporal networks is to integrate nodes and edges into a sequence of static observations [28].

Real networks often contain sensitive data such as personal information or economic transactions. Such data faces the threat of being breached or leaked. With the possession of certain individuals in the networks, a network attacker (adversary) can re-identify an entity and its connections. Therefore, keeping the nodes anonymous in the network is imperative for social network publishing and analysis. Temporal networks pose additional challenges as with the connections changing over time, nodes can become less or more anonymous at different points in time. Therefore, it is important to find a way to evaluate the anonymity of nodes in order to deploy anonymization techniques. Researchers have proposed different methods such as k-anonymity[33], l-diversity[42] over the years. These methods often utilize the local structure of the nodes and compare the similarity with other nodes in the network in order to assess how anonymous the nodes are in the network.

The naive method of anonymization, simply deleting the node labels, is inadequate for networks, because the network structure still contains vast amounts of information. Over the years, researchers have proposed different anonymization

methods on networks. Node aggregation [2] aggregates nodes to supernodes, and node clustering [41] clusters the nodes into different groups. Network modification methods are often implemented to modify the network so that the anonymity of the nodes increases while keeping the overall network properties. However, most of the anonymity studies are conducted on static networks and have not yet proved their effectiveness on temporal networks.

In this thesis, we focus on measuring and improving anonymity in temporal networks using the local structural information of nodes. Specifically, based on snapshots of the temporal network, We measure the node anonymity using the nodes' ego networks at different points in time and identify the non-anonymous nodes. Based on this, we conduct experiments between network anonymity and network properties. Additionally, we anonymize the temporal network using a perturbation technique based on the proposed node anonymity.

The main research question of the thesis is: How does node anonymity evolve in temporal networks?

To answer this question we look into the following sub-questions:

1. How should anonymity in a temporal network be measured?
2. How does anonymity change through time in temporal networks?
3. How does anonymity relate to the network properties and centrality in temporal networks?
4. How can we perturb the network over time to make it more anonymous?

This work is structured as follows: Relevant works on network anonymity measurements and anonymization are summarized in Chapter 2. In Chapter 3, we give important definitions, and the problem of anonymizing temporal networks is formalized. Chapter 4 gives the details of the approaches including the measurement of anonymity in temporal networks, global network properties and perturbation techniques. The used temporal network datasets are introduced in Section 5. In Chapter 6, we demonstrate experimental results to show how anonymity evolves over time, and how anonymity relates to network properties and node centrality. We also discuss the effectiveness of anonymization algorithms. Chapter 7 gives the conclusion of the research and possible future work.

Chapter 2

Related Work

Network anonymity and anonymization have become important research focuses in network analysis in recent years, especially with the growth of social networks. In this section, we first summarize how different anonymity measurements have been proposed to evaluate the anonymity of the network, including k-anonymity and its variants. Afterward, we will give a brief summary of node-based and edge-based anonymization techniques.

2.1 Anonymity Measures

Related work about one of the most commonly used anonymity measures, k-anonymity and its variants are introduced in 2.1.1, and measurements using other methods are summarized in 2.1.2.

2.1.1 k-Anonymity and its variants

The methods of k-anonymity [33] and its variants [9, 16] focus on the neighborhood structure of the node from direct neighborhood to larger peripheral structures. In networks, k-anonymity evaluates the overall anonymity with the size of the equivalent class. The key idea behind it is to partition nodes with identical neighborhood structures in the network into the same equivalence class. When each node in the network has at least $k-1$ other nodes in its equivalent class, the network is k-anonymous.

Depending on the selection of graph structure to be evaluated, different variants of k-anonymity are proposed and applied. The anonymity based on nodes' ego networks [31] uses the node's direct neighborhood as the structural knowledge. The work by de Jong et al. [9] proposed d-k-anonymity that expands the structural knowledge of a node up to its distance d. Mohapatra et al. [27] firstly defined k-degree anonymity using minimal frequency (k) of all unique degrees to generate k-anonymous degree sequences. Then the authors combined it with closeness centrality to determine the k-anonymity of the graph by finding nodes

with the same degree as the k-anonymous nodes in the one-step neighborhood distance.

2.1.2 Other Anonymity Measures

Centrality measures can also serve as an index for the importance of the nodes in the network and the priority of anonymization. For example, in a large network, the degree distribution tends to follow a power law distribution [4]. As a result, there are fewer nodes with high degrees and more nodes with few edges (lower degrees). Degree centrality can be viewed as an empirical index for anonymity as the nodes with high degrees tend to be more unique in the network and thus more vulnerable to attacks. Xiao and Tao [39] proposed a so-called *personalized anonymity*, different from k-anonymity techniques as a general method for the entire network, the personalized anonymity allows each node to define its own importance and need for anonymization. This is built on the generalization scheme in which the probability of each node being re-identified is calculated. The main drawback of k-anonymity is its lack of diversity in data attributes which can be vulnerable to certain adversary attacks. To address the problem, Machanavajjhala et al. [23] proposed l-diversity for representing the l values of sensitive attributes of the data such as edge labels and edge weights. On top of it, Li et al. [22] build the equivalent classes with t-closeness by comparing the distance between the attribute distribution over the entire data and sensitive attributes.

To sum up, node anonymity is usually measured by exploiting the nodes' local structural information, with the combination of network properties such as centrality measures. The anonymity of each node is thus evaluated to determine if it is similar to other nodes in the network.

2.2 Network Anonymization

There are generally three categories of network anonymization methods[1]: The first aims to modify the network's nodes or edges so that the network reaches k-anonymity. The second method uses probabilistic models to randomize the edges. The third method uses network generalization. We focus on the first category in this work. In the network, both the nodes and edges are possible to be identified by the adversary, with entity disclosure attack and link disclosure attack [3], respectively. Based on this, various anonymization countermeasures have been proposed to modify the network to reach k-anonymity.

The clustering-based methods cluster similar nodes or edges into different groups which form super-nodes and super-edges. The individual nodes or edges can become anonymous [43]. The application of k-anonymity in clustering methods ensures that in each cluster, there are k records. Byun et al.[5] proposed a k-member clustering algorithm which minimizes the intra-cluster distance while keeping at least k nodes in each cluster to reach k-anonymity. The method helps reduce information loss during anonymization. Campan and Truta [6]

proposed a greedy anonymization method in which they uniformly partition the nodes and generalize the edges at the same time to ensure there are at least k elements in each cluster. The structural information loss during clustering is measured by the proximity of nodes' neighborhoods.

On the other hand, modifications of networks including addition and deletion of nodes and edges also show their strength in anonymizing a network. Chester et al.[7] partition the degree sequences into subsequences with k elements and add dummy nodes to pair with the non-anonymous nodes to reach k-anonymity. Kiabod et al. [19] proposed a method which classifies each node as positive or negative based on whether the node's degree is increased or decreased after anonymization. The edge deletion algorithm is applied to reduce the degree of negative nodes.

Wang et al. [36] first proposed the Class Safety Condition (CSC) to partition nodes with similar attributes into different classes. Then, CSC is modified for the temporal network so that the nodes in the same class are added at the same timestamp. Using the time-series-based CSC, dummy nodes are added to the classes to enhance privacy.

Anonymizing temporal networks has not yet been widely researched because of the complexity of temporal dynamics in the networks. Time series analysis and network sequences at certain timestamps are often used for the problem. In this thesis, we choose the second method and build temporal networks over time using a sequence of networks.

Chapter 3

Preliminaries

In this chapter, we give formal definitions of concepts that are important throughout the research including network and temporal network.

3.1 Network

A network is defined as a graph $G = (V, E)$, in which V is the node set and E contains the edges that link the nodes in the network. $\{u, v\}, (u, v \in V)$ is an undirected edge between node u and v. We use $|V|$ to denote the number of nodes in G and $|E|$ to denote the number of edges.

The local structure of a node in a network is defined as the neighborhood of the node. We define the *neighbors* of node v as the set of nodes directly linked with v. $N(v) = \{u \in V : \{u, v\} \in E\}$. The degree of node v is $deg(v) = |N(v)|$. The ego network of node v, denoted as G_v^o in this research, is defined as a sub-graph of G: $G_v^o = (V_v^o, E_v^o)$, where the node set is node v and its direct neighbors: $V_v^o = N(v) \cup \{v\}$, and the edge set contains the edges between these nodes. Figure 3.1 shows the ego network of node u. It consists of the node u, the nodes directly connect with u, and the edges between them.

3.2 Temporal Network

Static networks remain the same regardless of time. However, there are also temporal networks that change through time. In a temporal network denoted as $\mathcal{G}$, each edge contains an additional timestamp element t. For example, edge $(\{u, v\}, t)$ means the edge between node u and v appears at timestamp t. Different studies have given different formal definitions of temporal networks, especially on the aspect of time [11, 15, 30]. In this work, we use the concept of *snapshot* to define the temporal network. Let $\mathcal{T} = \{t_0, t_1, ..., t_T\}$ represent a series of discrete time domains. A temporal network consists of several snapshots $\mathcal{G} = \{G_{t_0}, G_{t_1}, ..., G_{t_T}\}$. Each snapshot $G_{t_i} = (V_{t_i}, E_{t_i})$ contains the node set

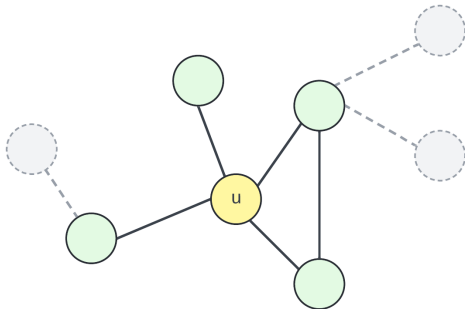

Figure 3.1: Example of an ego network in a network, node u, its direct neighbors (green nodes), and all the edges among these nodes together form the ego network of u.

V_i, the edge set E_i at timestamp t_i, namely:

$$V_{t_i} = \{v \in V : \exists(\{u, v\}, t) \in E, t \leq t_i\}$$

$$E_{t_i} = \{(\{u, v\}, t) \in E : t \leq t_i\}$$

This means that all edges with timestamps no later than a designated timestamp t_i are stored in the snapshot. Therefore, a temporal network can be viewed as a set of static networks as it stores the static network at certain timestamps. One assumption made for all the temporal networks studied in the thesis is that there are no nodes and edges disappearing through time; we only consider the networks with a growing number of nodes and edges.

Chapter 4

Methodology

In this chapter, we explain the approach used for measuring temporal network properties and for measuring anonymity in the temporal network. Section 4.1 introduces the anonymity measurement we use for the thesis. Based on it, in Section 4.2 we show and how it is calculated in the temporal networks. In Section 4.3, we explain the important network properties and centrality measurements in the temporal networks. Section 4.4 introduces the random perturbation algorithm and the uniqueness-based perturbation algorithm, which aim to increase the overall anonymity of temporal networks.

4.1 Anonymity Measurement

We assume a scenario in which the adversary knows that a certain individual exists in an anonymized network and the number of connections this individual has. The adversary aims to identify the individual in the network. The connections are represented as the degree of the target node in the network.

In order to achieve this, the adversary needs to utilize the local structural information of the nodes to classify nodes with a similar structure leading to the method of k-anonymity. A k-anonymized network ensures that each node is similar to at least $k-1$ other nodes. For example, when $k=4$, each node has at least three other nodes in the network that have the same local structure. If the adversary aims to identify an individual, they would find at least four nodes as the possible candidates of the target individual. Each node therefore has a possibility equal to or less than $1/k$ of being identified. Therefore, a network reaching higher k-anonymity tends to be safer and more anonymous.

Often, nodes with the unique structure in the network are more vulnerable to be identified, because there are no other nodes in the network similar to them. Therefore, in this work, we focus on the scenario of $k=2$ so that each node that has at least one other similar node will be considered anonymous. We use the measure of (n,m)-anonymity, a variant of k-anonymity to measure the anonymity of each node in the network. We will refer to it as anonymity in the

rest of the thesis.

As defined in the previous chapter, the ego network of a target node u consists of the target node, its direct neighbors and the edges between them. As Figure 4.1 shows, we use n and m to represent the number of nodes and edges of the ego network. The combination of (n, m) is defined as the *ego state* of the target node u.

Definition 4.1.1 (Ego State)**.** Given a network $G = (V, E)$ and a node $u \in V$, the ego state of u is defined as:

$$s(u) = (n, m) = (|V_u^o|, |E_u^o|)$$

.

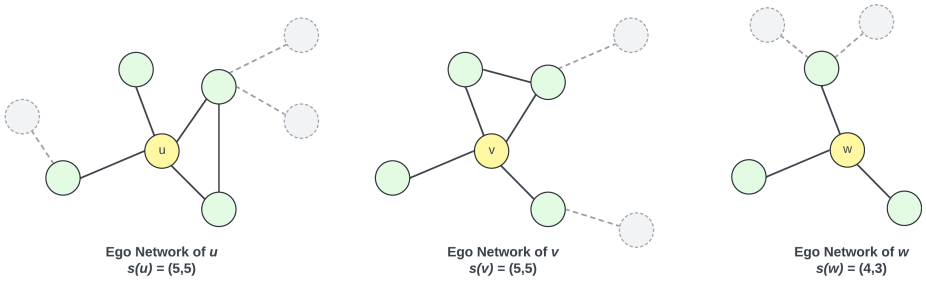

Figure 4.1: Examples of ego state of three ego networks

The state of a node can be used to measure the uniqueness of the node's neighborhood structure. We give the definition of (n, m) equivalent nodes as follows:

Definition 4.1.2 ((n, m)-equivalence)**.** Node u and v are (n, m) equivalent if $s(u) = s(v)$.

The nodes with the same ego state reside in the same equivalent class. The anonymity is determined based on the size of the equivalent class.

Definition 4.1.3 ((n, m)-uniqueness)**.** A node u is (n, m) unique when the size of its equivalence class is 1, i.e., $\forall v \in V - \{u\}, s(u) \neq s(v)$.

For simplicity, we refer to (n, m)-uniqueness and (n, m)-equivalence as simply uniqueness and equivalence in the rest of the thesis. A unique state in the network means a unique local structure of a node. A node with a unique state can be re-identified by the adversary. On the other hand, if two or more nodes share the same state, the adversary cannot de-anonymize the nodes solely based on the information of ego network structure, the nodes are therefore anonymous. For example, node u and v in Figure 4.1 have the same 1-neighborhood structure, therefore they cannot be uniquely identified under limited adversary knowledge.

4.2 Measuring Anonymity in Temporal Networks

Using the defined anonymity measurement and snapshots defined in Chapter 3, we can now measure anonymity in temporal networks. The algorithm for measuring the anonymity of a temporal network is shown in Algorithm 1. The algorithm works as follows: With the timestamps given in the input network, we separate the temporal network $\mathcal{G}$ to several snapshots $\mathcal{G} = \{G_{t_0}, G_{t_1}, ..., G_{t_T}\}$. For each snapshot, we calculate the ego state of each node u in the node set of the snapshot (Line 5). Afterward, for each node u in the snapshot, the frequency of ego state $freq(s(u))$ is calculated over the entire snapshot and stored as how many nodes in the current snapshot share the same state as node u, i.e., the size of the equivalence class the node is in (Line 10). The same process is carried out for all the snapshots and the overall anonymity is updated accordingly.

Algorithm 1 Anonymity for Temporal Networks

Input: Temporal Network $\mathcal{G}$, timestamps set $\mathcal{T}$
Output: Anonymity A

1: $A = \{\}$ `// Anonymity of the temporal network`
2: **for** $t = t_0, t_1, ..., t_T$ **do**
3: $S_t = \{\}$ `// Ego state of current snapshot`
4: **for** $u \in V_t$ **do**
5: $S_t[u] = (|V_u^o|, |E_u^o|)$ `// Ego state of node` u
6: **end for**
7: $A_t = \{\}$ `// Anonymity of snapshot`
8: **for** u in S_t **do**
9: $A_t[u] = freq(S_t[u])$ `// Calculate frequency of` $s(u)$
10: **end for**
11: $A[t] = A_t$
12: **end for**
13: **return** A

The algorithm outputs the anonymity of the temporal network which records the anonymity of every node in each snapshot. For each snapshot, we can calculate the *uniqueness percentage* by:

$$U(G_t) = \frac{|\{v \in V_t : freq(s(v)) = 1\}|}{|V|}$$

The percentage of unique nodes is crucial for understanding how anonymous the snapshot is. The higher the value of $U(G_t)$ is, the more unique nodes the snapshot has, and the less anonymous it is. By measuring $U(G_t)$ for different snapshots of a temporal network, we can observe the change in uniqueness of the temporal network over time.

4.3 Network Properties

The main difference between temporal networks and static ones is that temporal networks encode temporal changes in the network over time. Such changes represent how networks evolve. In order to understand the network anonymity dynamics, we will explain some important network properties.

4.3.1 Global Network Properties

Global network properties help explain the network structure from a macro perspective and demonstrate how the network is constructed. The density of the network demonstrates how densely nodes are connected with each other. Given an undirected network $G = (V, E)$, the density is denoted as:

$$density(G) = \frac{2 \cdot |E|}{|V| \cdot |V - 1|}$$

In temporal networks, the analysis of density can be done per snapshot or as the average value over a certain period of time. A growing network is usually denser in the beginning as fewer nodes are joining and edges forming among these nodes, which indicates a more interconnected and cohesive network structure. In later phases, especially for large networks, as more and more nodes join the network, the possible number of edges that can exist in the network grows quadratically, whereas actual edges do not. The density will start to fluctuate or drop, making the network sparse.

Density depicts how dense or sparse the network is. To evaluate the connections between two nodes that are not directly connected, we use distance. The distance of node u and v, $dist(u, v)$ is the shortest path from node u to v. For example, if $\{u, w\}, \{w, v\} \in E, \{u, v\} \notin E$, $dist\{u, v\} = 2$. In temporal networks, the average distance l_G shows how far two nodes are away from each other on average.

$$l_G = \frac{1}{|V| \cdot |V - 1|} \sum_{u \neq v} dist(u, v)$$

In temporal networks with a growing number of edges, the distance between two existing nodes will decrease over time as new paths are constantly forming. However, because of the addition of new nodes in the network, the average distance of the temporal network may grow.

The clustering coefficient of a node v is defined as:

$$CC(v) = \frac{2 \cdot |\{\{u, w\} \in E : \{u, v\}, \{u, w\}, \{w, v\} \in E\}|}{deg(v) \cdot (deg(v) - 1)}$$

The average clustering coefficient (ACC) of the network is:

$$ACC(G) = \frac{1}{|V|} \sum_{v \in V} CC(v)$$

ACC explains how many common neighbors nodes share. A higher ACC means the nodes form many triangles with other nodes. The forming of triangles in temporal networks affects the structure of the nodes' ego networks and therefore changes the uniqueness of nodes over time.

4.3.2 Betweenness Centrality

Centrality measurements evaluate the importance of each node in the network. For example, degree centrality ranks the nodes by their degree, while closeness centrality evaluates how close a node is to all the other nodes. In this work, we focus on betweenness centrality. Betweenness centrality assesses nodes based on how often they form the shortest paths between other nodes in the network.

Betweenness centrality is defined as [12]:

$$C_B(v) = \sum_{u \neq v \neq w \in V} \frac{\sigma_{uw}(v)}{\sigma_{uw}}$$

Here, σ_{uw} denotes the number of shortest paths from node u to w, $\sigma_{uw}(v)$ denotes the number of shortest paths from node u to w which pass node v.

An extended definition of betweenness centrality in temporal networks uses temporal paths which represent how the connection between two nodes changes over certain time intervals [34]. Different temporal shortest paths algorithms have been proposed [14, 20, 29] but they expand the types of paths to incorporate. Therefore, since we use snapshots to represent temporal networks, we still apply the definition of betweenness centrality in static networks as given to each snapshot.

4.4 Perturbation-based Anonymization

The unique local structures of the nodes in the network make them unique and possible to be identified by an adversary. In order to protect the unique nodes, perturbation strategies may use edge addition and deletion around certain nodes so that they can have the same local structure as other nodes in the network. In this section, we introduce random deletion first and propose a uniqueness-based deletion algorithm on temporal networks.

4.4.1 Random Deletion

The idea of random deletion is to randomly remove a certain number of edges in the network. In temporal networks, we choose to delete the newly added edges between two snapshots. This ensures we take the temporal aspects into consideration because the structure of the previous snapshot is preserved.

The algorithm of random deletion is shown in Algorithm 2. The algorithm requires a preset deletion percentage p, this represents the possibility of an edge being deleted. For each snapshot, we compare it with the previous one to get the

set of new edges *new_edges* (Line 3). Afterwards, each new edge has a possibility of p to be deleted from *new_edges* (Line 5). Finally, the processed *new_edges* is added to the previous snapshot to form the current perturbed snapshot, and the previous snapshot is updated accordingly (Line 7) until all snapshots are perturbed. The algorithm outputs the perturbed temporal network.

Algorithm 2 Random Deletion on Temporal Networks

Input: Temporal Network $\mathcal{G}$, timestamps set $\mathcal{T}$, deletion percentage p

1: $new\_edges = \{\}$
2: **for** $t = t_1, ..., t_T$ **do**
3: $new\_edges = E_t - E_{t-1}$ `// New edges in current snapshot`
4: **for** *edge* in *new_edges* **do**
5: Remove *edge* with probability p
6: **end for**
7: $E_t = E_{t-1} \cup new\_edges$ `// Update edge set`
8: **end for**
9: **return** $\mathcal{G}'$ `// Perturbed temporal network`

4.4.2 Uniqueness-based Deletion

Random deletion helps reshape the network structure and thus influences the network properties and uniqueness. The drawback of random deletion is that the edges deleted cannot be guaranteed to be the ones that directly connect with the unique nodes. Since the goal of perturbation is to decrease the percentage of unique nodes in the network, edges that connect with the unique nodes should have priority for deletion.

Based on this idea, we propose a perturbation algorithm based on the node uniqueness introduced in Chapter 4.1. We refer to the edges connected with at least one unique node as *unique edge*. This method ensures that for the current snapshot, the unique edges are deleted first. An example of the method is given in Fig 4.2. In the figure, all the unique nodes are colored green and the edges linked with them are shown in red (unique edges). In the previous snapshot, node 1 is unique with the state of (5,5). For the current snapshot, nodes 10, 11 and 12 are added to the network, which makes node 3 unique. The current network has 2 unique nodes and 7 unique edges.

With a preset deletion percentage p, for each snapshot, there are a certain number of edges to be deleted. For example, a 15% deletion percentage in the current snapshot means there are 2 edges to be deleted. With the proposed method, the unique edges are always randomly selected and deleted first. Therefore, in the perturbed snapshot, edges (1,5) and (3,11) are deleted. Both nodes 1 and 3 are not unique nodes anymore. For larger networks, if the unique edges in the current snapshot are less than the number of edges to be deleted, the unique edges are first deleted, followed by random deletion in the new edges.

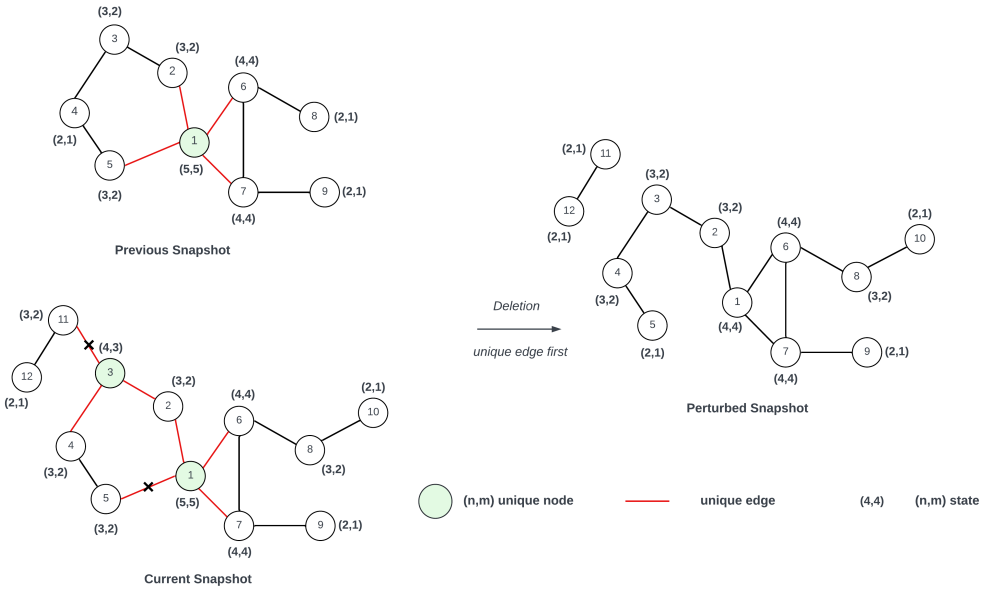

Figure 4.2: Example of uniqueness-based deletion, the green nodes represent (n, m) unique nodes, the red edges are the unique edges that connect with at least one unique node. In the perturbed snapshot, the edges between node 3 and 11, 1 and 5 are deleted, making nodes 1 and 3 anonymous.

Algorithm 3 details the uniqueness-based deletion. First, *edge_uniq* is created to store the unique edges in the snapshot (Line 2). Before the actual deletion, the anonymity algorithm (Algorithm 1) is conducted on the current snapshot G_t so that we obtain the anonymity A_t, which contains the size of the equivalent class, i.e., the frequency of the ego state for each node.

Afterward, the nodes with the size of 1 of the equivalent class are selected as unique nodes (Line 6). Each edge in the snapshot is then examined if it is connected with at least one unique node, we treat it as a unique edge and store it in *edge_uniq* (Line 7). With the preset deletion percentage p, the number of edges to be deleted is calculated per snapshot (Line 10). The unique edges are first deleted, followed by randomly deleting new edges until we delete enough edges (Line 16). Finally, the snapshot after deletion is stored as the new snapshot (Line 19). After all snapshots are perturbed, the algorithm outputs the perturbed temporal network.

Algorithm 3 Uniqueness-based Deletion on Temporal Networks

Input: Temporal network $\mathcal{G}$, timestamps set $\mathcal{T}$, deletion percentage p

$new\_edges = \{\}$
$edge\_uniq = \{\}$ //Unique edge set
for $t = t_1, ..., t_T$ **do**
  $A_t = anonymity(\mathcal{G}, \{t\})$ //Snapshot anonymity (Algorithm 1)
  for $(\{u, v\}, t)$ in E_t **do**
    if $A_t[u] = 1$ **or** $A_t[v] = 1$ **then** //Check if the node is unique
      $edge\_uniq = edge\_uniq \cup \{(\{u, v\}, t)\}$ //Store unique edges
    end if
  end for
  $new\_edge = E_t - E_{t-1}$
  $del = p * |E_t|$ //Number of edges to delete
  for $n = del, del - 1, ..., 1$ **do**
    if $edge\_uniq \neq \emptyset$ **then** //Delete unique edges first
      Randomly delete one edge from $edge\_uniq$ and E_{t-1}
    else //Delete new edges then
      Randomly delete one edge from new_edge
    end if
  end for
  $E_t = E_{t-1} \cup new\_edges$ // Update snapshot edge set
end for
return $\mathcal{G}'$ // Perturbed temporal network

Chapter 5

Data

In this chapter, we introduce the real-world temporal network datasets used for the thesis and explain how the snapshots are created.

5.1 Datasets

In this section, the datasets used are introduced. The criteria for selecting the data for this research is that the network represents real-world social connections over time. In this thesis, all networks are processed and viewed as undirected, unweighted networks.

- **Contacts Hypertext** [18] This dataset represents face-to-face contacts during the ACM Hypertext Conference in 2009. In total, there were 113 contacts and 2,498 unique connections within the network over a time period of three days.
- **Reality Mining** [32] Reality Mining is a dataset which records the call relationships among a group of people from MIT. The time period of the data collected is over 9 months with 6,809 nodes and 9,484 edges.
- **Sp Infectious** [18] As an art exhibition contact network in Ireland, Sp infectious captures 10,972 users in total who visited the exhibition during the same time period of three days in 2009. The number of connections is 52,761.
- **Mooc Action** [21] Mooc Action contains course interactions of users at a MOOC platform. The nodes have two groups representing the users and MOOC courses. Every edge represents that a certain user takes a certain course on the platform. This bipartite network consists of 7,047 nodes and 178,406 edges in total.
- **Internet AS** [40] Created in 2004, Internet AS integrates Internet topology at the Autonomous System level. Edges represent an existing connec-

tion at the corresponding timestamp. There are 34,761 nodes and 114,496 edges.

- **Facebook Wall** [35] This dataset records the Facebook friendship relations and wall post interactions during a two-year time period from 2009. The network has 45,813 users, and 264,004 edges representing that a user posts a message on another user's Facebook Wall.
- **Slashdot** [13] This is a network representing Slashdot user replies over different threads. Each edge represents a reply from the source user to the target user. The network has 51,083 nodes and 130,370 edges.

Table 5.1 shows the size of the network at 20%, 50%, 80% and 100% of the total timestamps. It can be seen from the table that, except for the smaller network (Contacts Hypertext), all the other networks grow steadily with time, both in the number of nodes and the edges. The evolving patterns of temporal network sizes are important for the property and anonymity experiments in the following chapter.

Table 5.1: Network size at different points of time

Dataset		**Timestamp of total time**			
		20%	50%	80%	100%
Contacts Hypertext	$\|V\|$	82	104	111	113
	$\|E\|$	499	1,249	1,998	2,498
Reality Mining	$\|V\|$	1,421	3,382	5,400	6,809
	$\|E\|$	1,896	4,742	7,587	9,484
Sp Infectious	$\|V\|$	2,265	5,366	9,208	10,972
	$\|E\|$	10,552	26,380	42,208	52,761
Mooc Action	$\|V\|$	3,745	5,875	6,675	7,047
	$\|E\|$	35,681	89,203	142,724	178,406
Internet AS	$\|V\|$	12,182	19,174	28,295	34,761
	$\|E\|$	22,899	57,248	91,596	114,496
Facebook Wall	$\|V\|$	15,490	23,902	35,003	45,813
	$\|E\|$	52,800	132,002	211,203	264,004
Internet AS	$\|V\|$	12,182	19,174	28,295	34,761
	$\|E\|$	22,899	57,248	91,596	114,496
Slashdot	$\|V\|$	10,908	21,631	36,433	51,083
	$\|E\|$	26,074	65,185	104,296	130,370

5.2 Snapshot Creation

The way this work analyzes the temporal network is by creating a series of *snapshots*, i.e., we capture a static network at a certain timestamp, the snapshot network contains all the nodes and edges that appeared before this timestamp. By selecting a series of timestamps, we create a series of snapshots of the

temporal network. In this work, the first snapshot is created at the 5% of total timestamp, and the step size of the timestamp is 2% of the total timestamps. Therefore, each network is separated into 48 snapshots.

Figure 5.1 shows the number of nodes and edges grows over time. Based on how we build the snapshot, the same number of edges are added to the network for each new snapshot. Therefore, we can see a linear growth in the number of edges. As for nodes, most networks also witness a growth. Nodes in Mooc Action and Contacts Hypertext datasets, on the other hand, are increasing faster in the early snapshots, compared to later ones. This shows that networks have different ways of growing: constantly expanding new entities/users or maturing the user groups more densely by forming more edges within the groups.

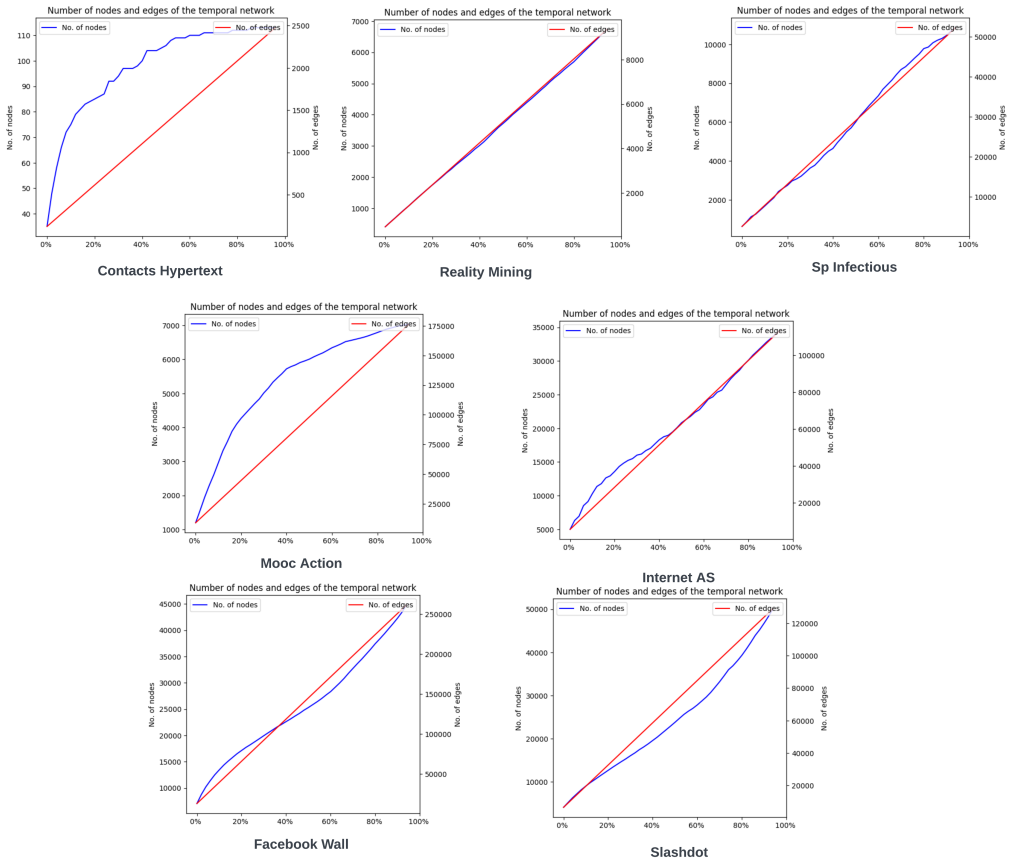

Figure 5.1: Number of nodes and edges in the snapshots of each dataset over time

Chapter 6

Results

In this chapter, we demonstrate experimental results of anonymity and uniqueness in temporal networks for answering the research questions proposed in Chapter 1. In Section 6.1, we briefly introduce the experimental setup. Section6.2 gives the results of how node anonymity changes over time to answer research question 2. In Section 6.3, the uniqueness and network properties are together analyzed, and node centrality is compared with anonymity in Section 6.4. Together they provide insights for research question 3. Moreover, to answer research question 4, we provide the perturbation results with random deletion and uniqueness-based deletion in Section 6.5.

6.1 Experimental Setup

Pre-processing is performed on each dataset due to different data formats. We extract only the (source node ID, target node ID, timestamp) from the original data. For duplicated edges, only the edge with the earliest timestamp is preserved. Additionally, all networks are treated as undirected networks.

Since every network starts from an empty network, if the beginning phase of the network is included, the fast growth of the network in the beginning is not suitable for analyzing the overall change of the network, as there is often a sharp change for every item. Therefore, in the following experiments, all the results start from 5% of the total timestamp to 100%. With the step size of 2% of the total timestamp, 48 snapshots are created for each network. All experiments are performed on Intel Core i7 CPU 16GB RAM. All of the codes and results can be accessed through the Github repository [1].

[1] https://github.com/JaviAnton/tempo-network-anonymity

6.2 Network Uniqueness

First, the anonymity is computed for each snapshot of each network. In Figure 6.1, we show the number of unique nodes and the percentage of unique nodes for each snapshot of the network.

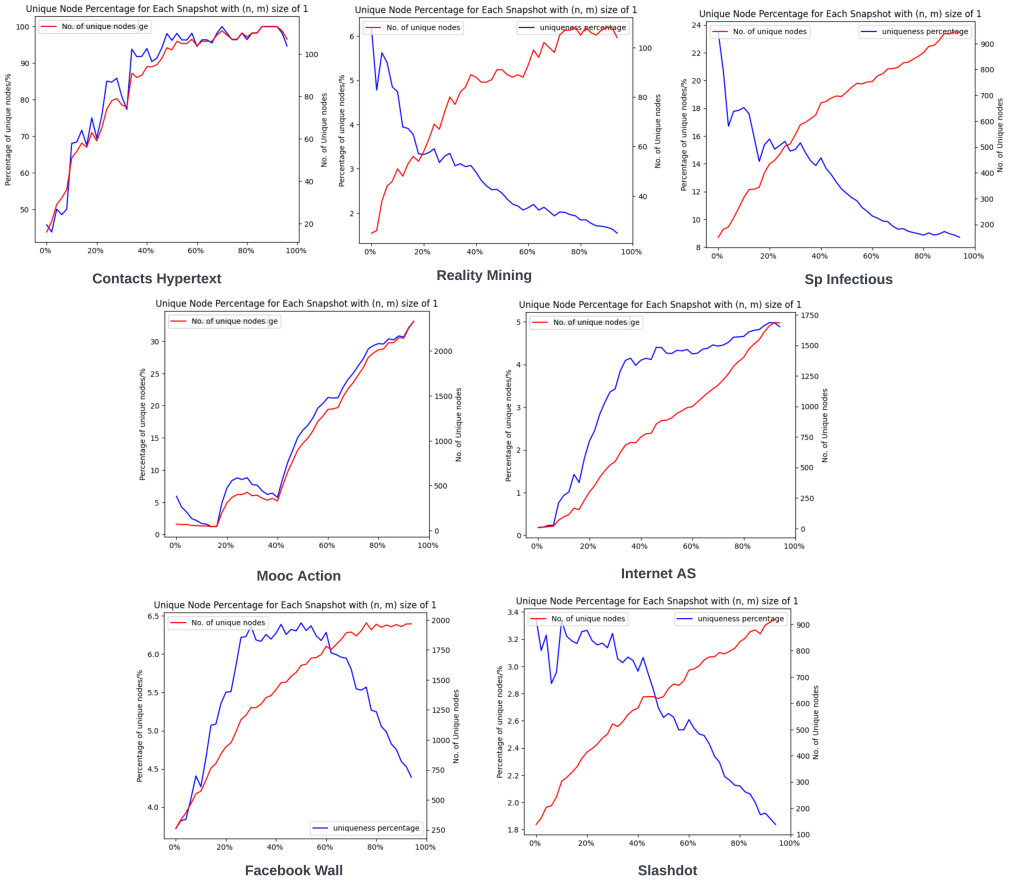

Figure 6.1: Number of unique nodes and uniqueness percentage of each network over time

For all the networks, the number of unique nodes keeps increasing. As for the uniqueness percentage, the performance of each network varies. Overall there are three different patterns. 1) The first pattern occurs in dataset Mooc Action, Contact Hypertext and Internet AS. In these three networks, the uniqueness percentage increases with the number of unique nodes. This indicates that the new nodes and edges joining the network help form new unique nodes. 2) In dataset Reality Mining, Sp Infectious and Slashdot, we witness a decrease in

the uniqueness percentage over time. This represents that the network becomes more anonymous with the new connections forming. 3) In the Facebook Wall dataset we can observe the uniqueness percentage reaches a plateau of 6.3% around 30% of total timestamps, then it slightly drops to 4.5%. This may indicate that the network gradually matures in the beginning and the nodes joining in the late phase increase the network's anonymity.

Additionally, by comparing the overall uniqueness of each network, we can see that for large networks (Slashdot, Facebook Wall), the uniqueness percentage is lower than for small networks. This finding suggests that large networks are more anonymous than small networks.

6.3 Uniqueness and Network Properties

To understand how temporal networks evolve through time, in this section we focus on network properties of the temporal networks over time. Then, for each snapshot of the network, the following properties are reported: average degree, clustering coefficient, density and average distance.

Since the values of network properties vary in scale, we normalized the value of each item to the range of 0 to 1. Figure 6.2 shows the normalized values of the aforementioned properties with the uniqueness percentage of the snapshot over time. Table 6.1 lists the uniqueness percentage and average degree, density and average clustering coefficient at the timestamp of 20%, 50%, 80% and 100%.

From Figure 6.2 we can see that for all networks except for Internet AS, the uniqueness percentage and the density follow similar patterns, as density drops, the network becomes sparser and the uniqueness gradually drops, and the network becomes more anonymous. This scenario applies to Reality Call, Sp Infectious and Slahdot. When density increases in Contact Hypertext and Mooc Action, the uniqueness also increases.

For all networks except for Sp Infectious, the uniqueness percentage and the Average Clustering Coefficient plots per network are also similar. The reason for the difference in Sp Infectious may be because the exhibition visits were represented in the data, the triangle in the network means the three persons were at the exhibition at the same time point. Therefore we witness the high values at the beginning and the end. The other properties perform differently with the uniqueness percentage per network.

In order to quantitatively evaluate the correlation between uniqueness percentage and other network properties, we choose Pearson correlation coefficient and p-value [8] as they indicate the significance of linear association between two sets of data, in our case, the set of uniqueness percentage per snapshot and the set of network properties per snapshot. For the Pearson correlation coefficient, the closer the value is to 1, the more positively correlated the two variables are. For p-vale, when it is smaller than a significance level ($p < 0.05$), the correlation can be considered statistically significant. Table 6.2 gives the results of the networks.

From the table, we can observe that the Pearson correlation coefficients

Table 6.1: Network Properties and uniqueness percentage (%) at different timestamps

Dataset	Properties	Timestamp			
		20%	**50%**	**80%**	**100%**
Contacts Hypertext	Unique %	71.605	91.346	96.396	94.690
	Avg Degree	11.531	23.115	34.901	43.823
	Avg Dist	2.016	1.874	1.729	1.659
	Density	0.144	0.224	0.317	0.391
	ACC	0.254	0.367	0.443	0.494
Reality Mining	Unique %	3.911	2.613	2.014	1.545
	Avg Degree	2.652	2.815	2.810	2.781
	Avg Dist	5.337	4.601	4.359	4.222
	Density	1.959E-03	8.556E-04	5.290E-04	4.132E-04
	ACC	4.983E-03	3.931E-03	2.963E-03	2.434E-03
Sp Infectious	Unique %	15.902	13.225	9.131	8.698
	Avg Degree	9.487	9.922	9.191	9.624
	Avg Dist	5.128	4.570	4.380	4.275
	Density	4.492E-03	1.904E-03	1.014E-03	8.868E-04
	ACC	0.367	0.427	0.451	0.436
Mooc Action	Unique %	1.199	11.113	27.270	33.092
	Avg Degree	18.899	29.937	42.374	50.168
	Avg Dist	2.005	2.003	2.001	2.000
	Density	5.270E-03	5.127E-03	6.172E-03	7.126E-03
	ACC	0.013	0.015	0.023	0.028
Internet AS	Unique %	1.427	4.114	4.517	4.883
	Avg Degree	3.694	5.916	6.452	6.571
	Avg Dist	3.044	3.194	3.504	3.752
	Density	3.137E-04	3.121E-04	2.302E-04	1.905E-04
	ACC	0.005	0.029	0.047	0.049
Facebook Wall	Unique %	5.068	6.257	5.530	4.392
	Avg Degree	6.646	10.931	12.083	11.671
	Avg Dist	5.112	4.962	5.257	5.564
	Density	4.404E-04	4.618E-04	3.501E-04	4.404E-04
	ACC	0.066	0.083	0.086	0.085
Slashdot	Unique %	3.187	2.946	2.190	1.837
	Avg Degree	4.699	6.021	5.725	5.148
	Avg Dist	4.103	4.279	4.453	4.532
	Density	4.458E-08	2.838E-04	1.591E-04	1.027E-04
	ACC	0.012	0.012	0.009	0.006

between uniqueness percentage and density and average clustering coefficient for the networks are higher than the average degree and average distance. The uniqueness percentage of Sp Infectious correlates more with the network density. And that of Internet AS correlates more with the average clustering coefficient.

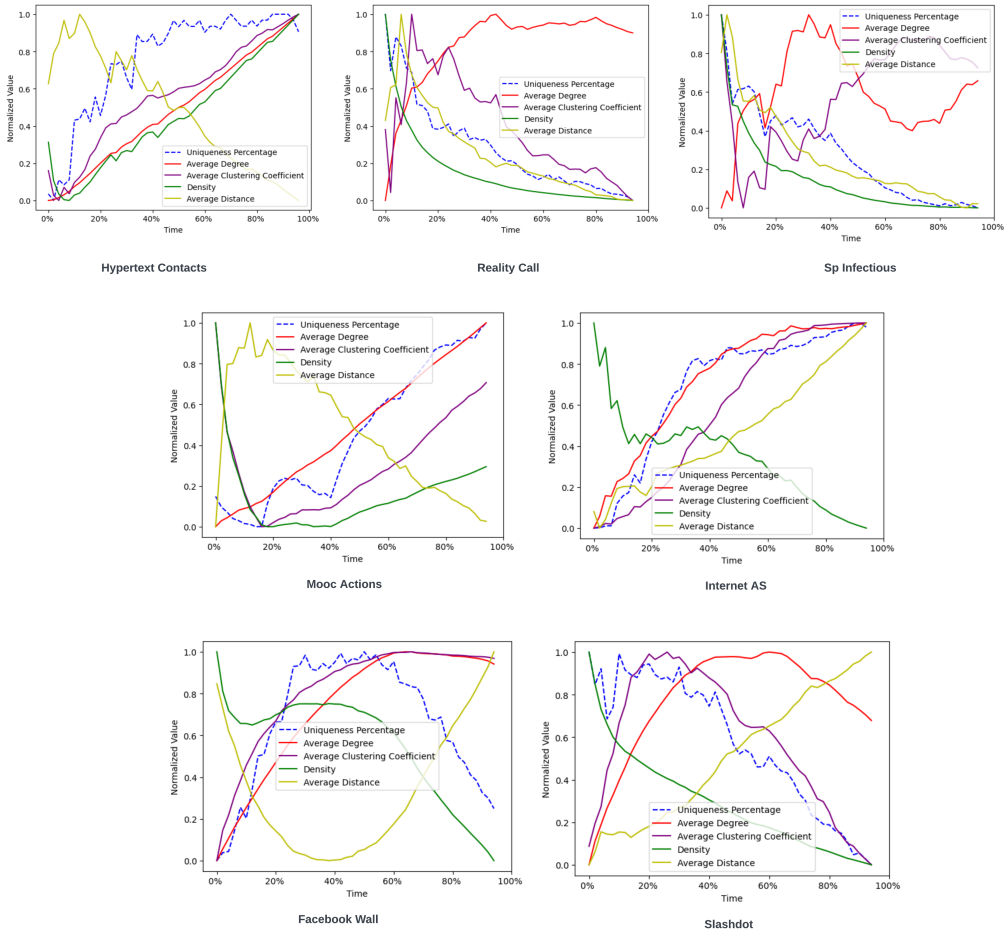

Figure 6.2: Normalized value of uniqueness percentage(blue dotted line) with normalized network properties (average degree, average clustering coefficient, density and average distance)

For other networks density and average clustering coefficient both connect with the uniqueness. Moreover, the results of p-values show that the correlation between uniqueness and density is significant. Therefore we suggest that if the temporal network becomes denser or has more triangles over time, more nodes tend to become unique and the network tends to become less anonymous.

Table 6.2: Pearson correlation coefficient between uniqueness percentage and density, average degree, average distance and average clustering coefficient (ACC)

Dataset		**Density**	**Avg Degree**	**Avg Dist**	**ACC**
Hypertext	Pearson	0.7683	0.8445	-0.7564	0.8908
	p-value	1.1586E-10	2.4331E-14	3.2523E-10	1.0183E-17
Reality Mining	Pearson	0.9338	-0.8885	0.9073	0.6144
	p-value	3.6126E-22	3.6162E-17	6.3835E-19	3.3907E-6
Sp Infectious	Pearson	0.9057	-0.0627	0.9129	-0.6422
	p-value	9.2077E-19	0.6720	1.6010E-19	8.6617E-7
Mooc Action	Pearson	0.1404	0.9763	-0.8872	0.6277
	p-value	0.3413	3.1520E-32	4.5947E-17	1.7914E-6
Internet AS	Pearson	-0.7912	0.9825	0.8291	0.9043
	p-value	2.1879E-11	3.1661E-35	3.3824E-13	1.2657E-18
Facebook Wall	Pearson	0.2949	0.5706	-0.8176	0.6624
	p-value	0.0418	2.2958E-5	1.3296E-12	2.9199E-7
Slashdot	Pearson	0.8468	-0.3640	-0.9657	0.7476
	p-value	3.3725E-14	0.0110	1.4052E-28	1.0273E-9

6.4 Uniqueness and Betweenness centrality

In this section, we look into the question of how anonymity is related to node centrality.

For each snapshot of the network, we calculate the betweenness centrality for every node and rank them based on these centrality scores. Afterwards, with the number of N nodes in the snapshots, we take the top 10%, 10%-20%, 40% -50% and the last 10% of $|V|$ nodes from the betweenness centrality ranking. For each subset of nodes, we count how many of them are unique.

Figure 6.3 shows the number of unique nodes in the full snapshot and in each betweenness centrality ranking subgroup. We can observe that the red line, i.e., the top 10% group of nodes, has more unique nodes than other subgroups. This shows that the unique nodes are more likely to also be the nodes with higher centrality values. Moreover, we can observe that for every network, the top 10% and top 20% of the nodes contain most unique nodes in the network.

Additionally, we calculate the precision and recall of the unique nodes in the top 10% and 20% of the centrality ranking nodes. The results are shown in Table 6.3. For precision, all seven networks reach over 0.25 for the top 10% of the centrality nodes. This means for nodes in the top 10% of centrality ranking, at least 25% of them are unique.

As for the recall, i.e., how many unique nodes are indeed included, the top 10% of centrality nodes in Dataset Reality Mining, Internet AS, and Slashdot contain over 85% of the unique nodes of the entire snapshot, on average. The recall is close to 1 for the top 20% ranking. For other datasets, 37%, 68% and 24% of the unique nodes are also the top 10% of the centrality nodes

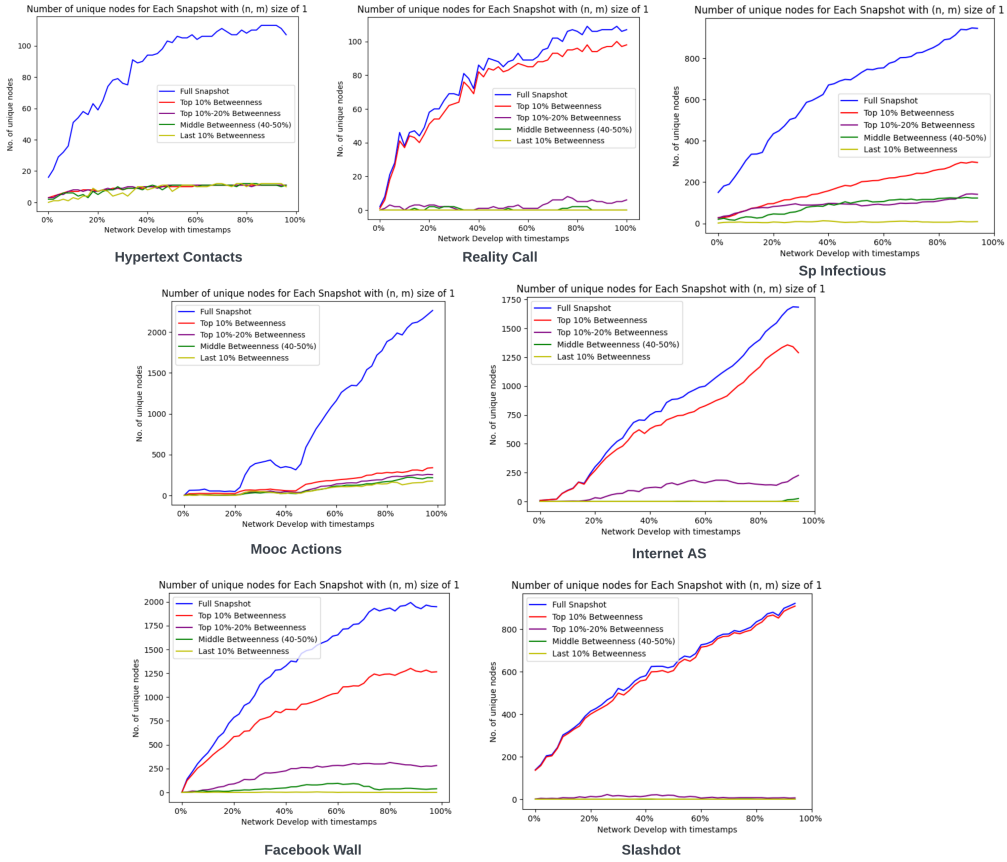

Figure 6.3: Number of unique nodes of full snapshot and different node sets based on the betweenness centrality ranking (top 10%, top 10-20%, top 40-50%, last 10 %) over time

for datasets Moon Action, Facebook Wall and Sp Infectious, respectively. An exception is the Contact Hypertext dataset, the unique nodes are spread out in the subgroups. Because it is relatively small and nearly all the nodes become unique in the end. However, the top 10% or 20% of centrality ranking nodes win over all the other sections for the proportion of unique nodes. These results suggest that for most temporal networks, the nodes in the top 20% of the betweenness centrality ranking contain most of the unique nodes. The higher the centrality value of a node is, the more likely the node is unique, and the more possible it can be re-identified by the adversary.

Table 6.3: Average uniqueness percentage with average precision & recall for top 10% and 20% of the nodes in the betweenness ranking

Dataset	**Average Unique Percentage (%)**	Top 10% Betweenness		Top 20% Betweenness	
		Precision	**Recall**	**Precision**	**Recall**
Contacts Hypertext	86.4053	0.9943	0.1145	0.9693	0.2281
Reality Mining	2.8627	0.2796	0.9775	0.1423	0.9923
Sp Infectious	12.8449	0.3014	0.2406	0.2550	0.4040
Mooc Action	15.4899	0.5131	0.3688	0.4650	0.6194
Internet AS	3.4852	0.2922	0.8648	0.1693	0.9775
Facebook Wall	5.5033	0.3678	0.6751	0.3005	0.7095
Slashdot	2.7087	0.2639	0.9754	0.1346	0.9936

6.5 Perturbation

In order to decrease the uniqueness of the temporal network datasets, we deployed two perturbation methods: random deletion and uniqueness-based deletion as discussed in Chapter 4.3. The deletion percentage p is set to 20%. Figure 6.4 shows the uniqueness percentage of the original network over time, the network after random deletion and the network after uniqueness-based deletion. To ensure randomness, all experiments are run three times and the average results are recorded.

From the result, we can observe that for Mooc Actions and Sp Infectious datasets, the uniqueness-based deletion performs better in decreasing the overall uniqueness percentage than random deletion, especially in the later phases of the network. Since the uniqueness percentage is relatively large compared with other datasets, the edge set consists of more unique edges. Therefore, the improvement is more visible. Contact Hypertext dataset is very small with 200 edges and a very high uniqueness in the end. The deletion causes fluctuation in both the random method and the uniqueness-based. Whereas for other datasets, because of the low uniqueness percentages (around 5%) and large network sizes, the uniqueness-based deletion does not substantially improve network anonymity. This is because by deleting unique edges in the network, the ego state of the unique nodes changes. However, in the meantime, the neighbor nodes' states also change. This could result in making the previous anonymous nodes unique. Therefore we witness a change in anonymity but no significant increase.

Table 6.4 gives the average uniqueness percentage over the original network, the network after random deletion and the network after uniqueness-based deletion. The results show that uniqueness-based deletion can improve the average uniqueness percentage of all networks, where effectiveness differs per network. Random deletion fails to accomplish it.

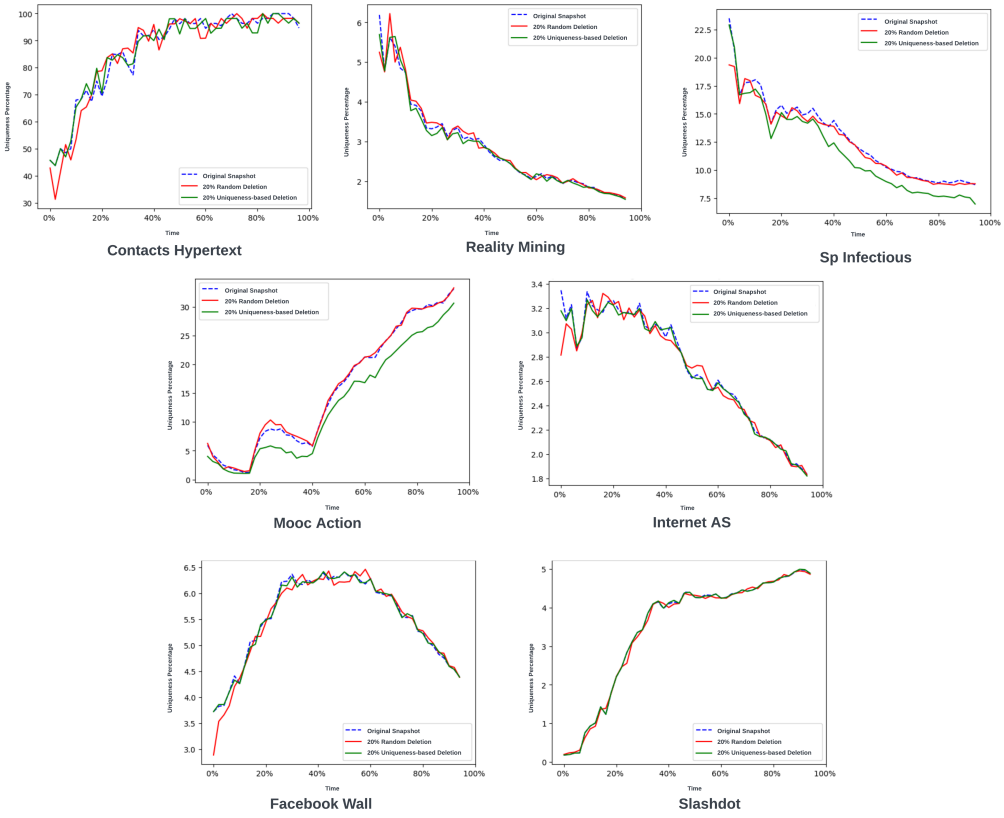

Figure 6.4: Uniqueness percentage (%) of original snapshot, snapshot after random deletion and snapshot after uniqueness-based deletion over time

Table 6.4: Average uniqueness percentage (%) and standard deviation for original network, network after 20% random deletion and network after 20% uniqueness-based deletion

	Original	Random Deletion	Uniqueness Deletion	
			Mean	Std dev
Contacts Hypertext	86.405	86.693	**86.077**	15.790
Reality Mining	2.863	2.886	**2.849**	1.126
Sp Infectious	12.845	12.490	**11.781**	3.678
Mooc Action	15.490	15.723	**10.626**	7.836
Internet AS	3.487	3.492	**3.339**	1.543
Facebook Wall	5.503	5.496	**5.482**	1.733
Slashdot	2.709	2.695	**2.679**	0.365

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis, we studied the measurement of temporal network anonymity using a measure which utilizes the node's ego network structure. We selected seven real-world social temporal network datasets and created temporal networks by means of snapshots in order to process and analyze anonymity over time. Experiments show that the number of non-anonymous nodes grows over time in each dataset. However, the networks vary in the overall uniqueness percentage, depending on whether new edges form more unique nodes.

Moreover, we explored the connection of anonymity to network properties and betweenness centrality. The experiments show that density positively correlates with the uniqueness percentage of the network. The more sparse the network is, the more anonymous the network tends to be. As for node centrality, the majority of nodes with higher betweenness centrality values (top 10% and top 20%) are unique nodes, the proportion of unique nodes in the top-ranked centrality nodes exceeds those with lower centrality values. Therefore, central nodes are more likely to be non-anonymous in the networks.

Finally, we proposed a uniqueness-based deletion algorithm which aims to target the edges connected directly with the unique nodes in the network in the current snapshot of the temporal network. Experiments show that the network after applying the proposed method of uniqueness-based deletion can decrease the average uniqueness percentage compared to the original temporal network and may perform better than random deletion. Although, additional experiments are needed to determine the statistical significance of these results.

7.2 Future Work

There are still some aspects for improvement for this research. As for the anonymity measurement, the scope of measuring anonymity can be extended to a larger radius of the target node so that more structural information can be used to evaluate anonymity. As for anonymization techniques, further improvements can be made to the selection of target edges to be deleted while keeping the anonymous nodes still safe. This can be approached by checking nearby nodes' state change after deletion and making sure fewer or no new unique nodes are created after deletion. Finding an optimized deletion percentage for each network is also crucial for better balancing anonymization and data utility.

Bibliography

[1] Charu C Aggarwal, Haixun Wang, et al. *Managing and mining graph data*, volume 40. Springer, 2010.

[2] Baruch Awerbuch and Yuval Shavitt. Topology aggregation for directed graphs. *IEEE/ACM Transactions On Networking*, 9(1):82–90, 2001.

[3] Korra Sathya Babu, Sanjay Kumar Jena, Jhalaka Hota, and Bijayinee Moharana. Anonymizing social networks: a generalization approach. *Computers & Electrical Engineering*, 39(7):1947–1961, 2013.

[4] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[5] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. Efficient k-anonymization using clustering techniques. In *International Conference on Database Systems for Advanced Applications*, pages 188–200. Springer, 2007.

[6] TMT Alina Campan and TM Turta. A clustering approach for data and structural anonymity. In *In Proceedings of the 2nd ACM SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD'08), in Conjunction with KDD*, volume 8, 2008.

[7] Sean Chester, Bruce M Kapron, Ganesh Ramesh, Gautam Srivastava, Alex Thomo, and S Venkatesh. k-anonymization of social networks by vertex addition. *ADBIS (2)*, 789:107–116, 2011.

[8] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. *Noise Reduction in Speech Processing*, pages 1–4, 2009.

[9] Rachel G de Jong, Mark PJ van der Loo, and Frank W Takes. Algorithms for efficiently computing structural anonymity in complex networks. *ACM Journal of Experimental Algorithmics*, 28:1–22, 2023.

[10] Jean-Pierre Eckmann, Elisha Moses, and Danilo Sergi. Entropy of dialogues creates coherent structures in e-mail traffic. *Proceedings of the National Academy of Sciences*, 101(40):14333–14337, 2004.

[11] Maximilian Franzke, Tobias Emrich, Andreas Züfle, and Matthias Renz. Pattern search in temporal social networks. In *Proceedings of the 21st International Conference on Extending Database Technology*, 2018.

[12] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.

[13] Vicenç Gómez, Andreas Kaltenbrunner, and Vicente López. Statistical analysis of the social network and discussion threads in slashdot. In *Proceedings of the 17th International Conference on World Wide Web*, pages 645–654, 2008.

[14] H Habiba, C Tantipathananandh, and T Berger-Wolf. Betweenness centrality measure in dynamic networks. *Department of Computer Science, University of Illinois at Chicago*, 2007.

[15] Steve Hanneke and Eric P Xing. Discrete temporal models of social networks. In *ICML Workshop on Statistical Network Analysis*, pages 115–125. Springer, 2006.

[16] Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. Anonymizing social networks. *Computer Science Department Faculty Publication Series*, page 180, 2007.

[17] Petter Holme. Temporal network structures controlling disease spreading. *Physical Review E*, 94(2):022305, 2016.

[18] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271(1):166–180, 2011.

[19] Maryam Kiabod, Mohammad Naderi Dehkordi, and Behrang Barekatain. A fast graph modification method for social network anonymization. *Expert Systems with Applications*, 180:115148, 2021.

[20] Hyoungshick Kim and Ross Anderson. Temporal node centrality in complex networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 85(2):026107, 2012.

[21] Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1269–1278, 2019.

[22] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2006.

[23] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3–es, 2007.

[24] Naoki Masuda and Renaud Lambiotte. *A guide to temporal networks*. World Scientific, 2016.

[25] R Mchaney and David Sachs. Web 2.0 and social media. *London: Bookboon*, 2016.

[26] Giovanna Miritello, Esteban Moro, and Rubén Lara. Dynamical strength of social ties in information spreading. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 83(4):045102, 2011.

[27] Debasis Mohapatra and Manas Ranjan Patra. k-degree closeness anonymity: A centrality measure based approach for network anonymization. In *Distributed Computing and Internet Technology: 11th International Conference, ICDCIT 2015, Bhubaneswar, India, February 5-8, 2015. Proceedings 11*, pages 299–310. Springer, 2015.

[28] Raj Kumar Pan and Jari Saramäki. Path lengths, correlations, and centrality in temporal networks. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 84(1):016105, 2011.

[29] Fabiola SF Pereira, Sandra de Amo, and Joao Gama. Evolving centralities in temporal graphs: a twitter network analysis. In *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, volume 2, pages 43–48. IEEE, 2016.

[30] Anton V Proskurnikov and Roberto Tempo. A tutorial on modeling and analysis of dynamic social networks. part I. *Annual Reviews in Control*, 43:65–79, 2017.

[31] Daniele Romanini, Sune Lehmann, and Mikko Kivelä. Privacy and uniqueness of neighborhoods in social networks. *Scientific Reports*, 11(1):20104, 2021.

[32] Ryan A. Rossi and Nesreen K. Ahmed. The network data repository with interactive graph analytics and visualization. In *AAAI*, 2015.

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

[34] Ioanna Tsalouchidou, Ricardo Baeza-Yates, Francesco Bonchi, Kewen Liao, and Timos Sellis. Temporal betweenness centrality in dynamic graphs. *International Journal of Data Science and Analytics*, 9:257–272, 2020.

[35] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 37–42, 2009.

[36] Chih-Jui Lin Wang, En Tzu Wang, and Arbee LP Chen. Anonymization for multiple released social network graphs. In *Advances in Knowledge Discovery and Data Mining: 17th Pacific-Asia Conference, PAKDD 2013, Gold Coast, Australia, April 14-17, 2013, Proceedings, Part II 17*, pages 99–110. Springer, 2013.

[37] Wei Wang, Yanyi Nie, Wenyao Li, Tao Lin, Ming-Sheng Shang, Song Su, Yong Tang, Yi-Cheng Zhang, and Gui-Quan Sun. Epidemic spreading on higher-order networks. *Physics Reports*, 1056:1–70, 2024.

[38] Stanley Wasserman and Katherine Faust. Social network analysis: Methods and applications. 1994.

[39] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, pages 229–240, 2006.

[40] Beichuan Zhang, Raymond Liu, Daniel Massey, and Lixia Zhang. Collecting the internet as-level topology. *ACM SIGCOMM Computer Communication Review*, 35(1):53–61, 2005.

[41] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *2008 IEEE 24th International Conference on Data Engineering*, pages 506–515. IEEE, 2008.

[42] Bin Zhou and Jian Pei. The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowledge and Information Systems*, 28(1):47–77, 2011.

[43] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM Sigkdd Explorations Newsletter*, 10(2):12–22, 2008.