Differentially Private Release of Heterogeneous Network for Managing Healthcare Data

RASHID HUSSAIN KHOKHAR, Concordia Institute for Information Systems Engineering, Concordia University, Canada
BENJAMIN C. M. FUNG, School of Information Studies, McGill University, Canada
FARKHUND IQBAL, College of Technological Innovation, Zayed University, United Arab Emirates
KHALIL AL-HUSSAENI, Rochester Institute of Technology, United Arab Emirates
MOHAMMED HUSSAIN, College of Technological Innovation, Zayed University, United Arab Emirates

With the increasing adoption of digital health platforms through mobile apps and online services, people have greater flexibility connecting with medical practitioners, pharmacists, and laboratories and accessing resources to manage their own health-related concerns. Many healthcare institutions are connecting with each other to facilitate the exchange of healthcare data, with the goal of effective healthcare data management. The contents generated over these platforms are often shared with third parties for a variety of purposes. However, sharing healthcare data comes with the potential risk of exposing patients' sensitive information to privacy threats. In this article we address the challenge of sharing healthcare data while protecting patients' privacy. We first model a complex healthcare dataset using a heterogeneous information network that consists of multi-type entities and their relationships. We then propose *DiffHetNet*, an edge-based differentially private algorithm, to protect the sensitive links of patients from inbound and outbound attacks in the heterogeneous health network. We evaluate the performance of our proposed method in terms of information utility and efficiency on different types of real-life datasets that can be modeled as networks. Experimental results suggest that *DiffHetNet* generally yields less information loss and is significantly more efficient in terms of runtime in comparison with existing network anonymization methods. Furthermore, *DiffHetNet* is scalable to large network datasets.

CCS Concepts: • **Security and privacy**; • **Information systems → Data management systems**; *Data mining*;

Additional Key Words and Phrases: heterogeneous information network, differential privacy, healthcare data management, information utility

ACM Reference Format:
Rashid Hussain Khokhar, Benjamin C. M. Fung, Farkhund Iqbal, Khalil Al-Hussaeni, and Mohammed Hussain. 2023. Differentially Private Release of Heterogeneous Network for Managing Healthcare Data. 1, 1 (January 2023), 29 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Authors' addresses: Rashid Hussain Khokhar, rashidhussain.khokhar@mail.concordia.ca, Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Quebec, Canada; Benjamin C. M. Fung, ben.fung@mcgill.ca, School of Information Studies, McGill University, Montreal, Quebec, Canada; Farkhund Iqbal, farkhund.iqbal@zu.ac.ae, College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates; Khalil Al-Hussaeni, kxacad@rit.edu, Rochester Institute of Technology, Dubai, United Arab Emirates; Mohammed Hussain, mohammed.hussain@zu.ac.ae, College of Technological Innovation, Zayed University, Dubai, United Arab Emirates.

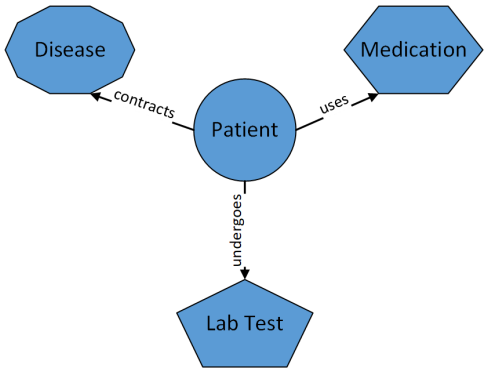


Fig. 1. Network schema

1 INTRODUCTION

In the past decade, heterogeneous information networks (HINs) have gained increasing attention in various application domains, such as social media, communications, energy, and health informatics, mainly due to its ubiquitousness and capability of representing rich semantics [58]. Many complex networks are modeled as graphs, where entities are described by nodes and their relationships are represented by edges. Recent database technologies have evolved to accommodate the need of storing large networks of connected data. The healthcare industry follows certain standards and requirements for managing healthcare data [24], such as providing better and timely services and mitigating privacy risks by protecting patients' sensitive information against privacy threats. To address these challenging requirements, we model a complex de-identified healthcare dataset that contains patients' medical histories, medications, laboratory tests, and demographics, using a heterogeneous information network that consists of multi-type entities and their multi-type relationships. A network schema of a heterogeneous health information network (HHIN) is illustrated in Fig. 1, which is a graphical representation of real-life health-related data. In the illustrated network schema, *Patient*, *Disease*, *Medication*, and *Lab Test* are entities, whereas *contracts*, *uses*, and *undergoes* represent relationships between entities. We use the terms *network* and *graph* interchangeably.

Fig. 2 provides an overview of privacy-preserving data publishing of HHIN. In the presented scenario a health information custodian (HIC) collects health-related data from multiple data sources (where a data source is denoted by DS in the figure). The collected data from all sources pertains to the same set of patients and is maintained in a single repository. The fusion of all the collected data results in a typical heterogeneous network. The goal of HIC is to publish the collected data to a data recipient for data analysis without compromising the patients' privacy. To address this real-life problem for health-network data and to bring additive advantages to HIC by properly balancing privacy and utility requirements, we propose a method that converts de-identified health network data into a differentially-private version.

It has been a common practice by the HICs to maintain health-related data in central storage to facilitate administrative operations, improve healthcare services, and support medical research [27]. Health data contains sensitive information about patients, and HICs must ensure the protection of patients' private information during the collection, use, and release of health data as mandated by regional and global data privacy laws [3, 24, 70]. Many health-service providers follow the practice of obtaining patients' consent when sharing their health data [31], and some use de-identification

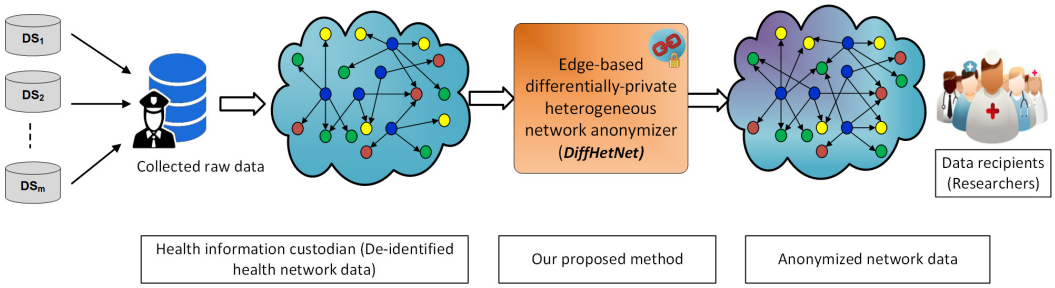


Fig. 2. Privacy-preserving health network data publishing

methods [57] to strip out explicitly identifying information. However, HICs have faced increasing privacy breaches of different natures [2, 4, 31] due to negligence of administrative employees, compliance failures, and the deployment of weak de-identification methods [6]. We argue that de-identification alone is not sufficient for privacy protection when data is required to be released openly without restricting it to authorized partners and covered entities.

Health social networking sites such as MedXCentral, Sermo and PatientsLikeMe have been increasingly adopted by healthcare professionals and patients for exchanging health-related information. The contents generated over these platforms are often shared with third parties for a variety of purposes, which poses risks of privacy breaches [41, 69]. Data sharing carries mutual benefits to both the HIC and the data recipient, but it comes with conflicting requirements on data privacy and data utility. To bridge the gap between these two conflicting requirements, several privacy models were proposed in the literature for network or graph anonymization. These models can be apprehended into two types: *syntactic* and *semantic* models.

There is a line of research [5, 44, 77, 79] based on syntactic privacy models that focuses on preserving structural information in networks. The works in [44, 77] prevent node re-identification, whereas some other works [5, 12, 79] focus on protecting against both node re-identification and edge disclosure in the presence of structural background knowledge of an adversary. Most of these works focus on undirected networks. It is not a good practice to utilize the same methods for anonymizing directed graphs. Generally, if a directed network is anonymized under syntactic-based models without considering the direction of edges it may be prone to re-identification attacks [9], and it also causes a loss of information utility because of the structural properties of the network. Among all privacy models, the works of [5, 12] are relatively better for privacy protection. They both are rooted in k-isomorphism. Chen et al. [10] show that an adversary with moderate background knowledge can identify certain links among nodes on a k-isomorphic graph [12] due to its deterministic nature. The work in [5] provides (k, δ)-privacy to resist against k-core attacks. It is also scalable to massive network data, but its application is limited to *homogeneous networks*, where nodes and edges are to be of a single type.

Another line of research [8, 10, 14, 25, 30, 71] applies *differential privacy* (*DP*) for anonymizing network data. It is a semantic model that provides strong privacy guarantees to an individual independently of an adversary's background knowledge [15]. Two frameworks, namely *interactive* and *non-interactive*, have been introduced regarding the utilization of the privacy budget ϵ [10, 15, 75]. The primary difference is that in the interactive framework the data custodian holds the raw network data, and a data analyst submits a set of queries in real time, for which the data custodian provides differentially-private answers. Each query would utilize a fraction of ϵ to produce a noisy answer. When the entire ϵ has been consumed, a data analyst would not be able to get the answer by querying the database. On the other hand, in

the non-interactive framework, the data custodian first anonymizes its raw network data by utilizing the entire privacy budget. Later, the anonymous data is released to the data analyst, who can perform analysis without any limitations on the data usage. This approach, widely known as *privacy-preserving data publishing* (*PPDP*) [20], is more appropriate in many real-life network data-sharing scenarios because of the flexibility for a data analyst to perform an analysis without specifying a target analysis. Therefore, in this article we focus on the non-interactive framework for network data publishing.

The intuition of differential privacy is that an individual's information should not be revealed from the output of the analysis in the anonymized data whether or not the individual opted in to be part of the database. *Node-differential privacy* [8, 14, 30] and *edge-differential privacy* [10, 25, 71] are the most common formulations for network data anonymization in the literature. In node-DP, two graphs are neighboring graphs if they differ by *at most* one node and, by extension, all its edges. In edge-DP, two graphs are neighboring graphs if they differ by *at most* one edge. In this article we follow the formulation of edge-differential privacy to tackle the problem of protecting sensitive links of a patient in the heterogeneous health network. We focus on preventing the disclosure of sensitive relationships between patient nodes and non-patient nodes from adversarial inbound and outbound privacy attacks.

Compared with existing work on edge-DP [10], our solution to the problem is different in several aspects. First, in contrast to homogeneous network solutions, our solution aims to protect sensitive links of an individual in a *heterogeneous network* that is characterized by having multiple types of nodes and edges. Second, our proposed solution takes the direction of edges into account to maintain the structural properties of the network. Third, our solution extracts the network structural properties without performing vertex labeling [10] (which is required in order to form dense regions for effective anonymization) on an input network; thus, our solution is not sensitive to the density of the input network. Finally, the underlying procedure for anonymization is also different. Our solution comprises two phases, where each phase provides both *indegree* and *outdegree* protection for the input network. The two phases integrate the exponential mechanism that uses the degree-centrality function, which yields a real-valued score. For an input network, the first phase protects vulnerable nodes by picking nodes that are prone to adversarial attacks due to having fewer incoming or outgoing connections. In the second phase, we preserve information utility by choosing nodes having higher scores and connecting them to the nodes that were picked in the first phase to protect their inbound and outbound connections.

Contributions. This is the first edge-differentially-private, non-interactive framework providing a practical solution to health information custodians (HICs) who wish to release real-life heterogeneous health-network data. Our contributions are summarized as follows:

- We model complex, de-identified healthcare data as a heterogeneous information network that consists of multi-type entities along with their directional relationships. Existing solutions [10, 25, 71] consider nodes and edges to each be of a single type and edges to be bidirectional (or undirected). Thus, these solutions cannot maintain important semantics and structural information of the heterogeneous network.
- We propose *DiffHetNet*, a differentially-private method to protect patients' sensitive links in a health network. Compared with the anonymization method for undirected networks in [10], our method offers better protection against an adversary's inbound and outbound attacks for learning the existence of a patient's sensitive information. Experimental results suggest that our method generally yields less information loss and is significantly more efficient in terms of runtime when compared with related anonymization methods from the literature. Furthermore, our method effectively extracts the structural properties of an input network, and it is not sensitive

to the density of edges in the network. Our experiments demonstrate the density-insensitivity feature of our method.

- We evaluate the performance of our proposed method with respect to information utility and efficiency using different real-life network datasets. In addition, we demonstrate that our approach is scalable to large network datasets.

The rest of this article is organized as follows. In Section 2, we review the related work. In Section 3, we define the problem. In Section 4, we present our proposed differentially-private algorithm. In Section 5, we compare our proposed method with the existing methods and evaluate the performance in terms of information utility, efficiency, and scalability. In Section 6, we provide a discussion on our approach, limitations, and future work. Finally, Section 7 concludes this article and present the background knowledge related to information networks, network measures, differential privacy for network data, and information-loss measures in appendices.

2 RELATED WORK

We group the related work into two categories: network data anonymization under non-differential privacy models, and network data anonymization under differential privacy models.

2.1 Network data anonymization under non-differential privacy models

A family of works [12, 44, 77, 79] has proposed to preserve the structural information in graph networks. Liu and Terzi [44] proposed an approach to construct an anonymous graph of k-degree anonymity, which requires generation of at least $k-1$ other nodes, for every node v. This notion of anonymity prevents identity disclosure from structural attacks based on adversary knowledge on a certain degree of nodes. Zhou and Pei [77] proposed k-neighborhood anonymization to prevent an adversary's attack with 1-neighborhood background knowledge about the victim. The goal of this approach is to ensure that the identity of an individual may not be revealed with a confidence greater than $1/k$ in the sanitized version of the original graph. Cheng et al. [12] proposed k-isomorphism, a solution that generates k disjoint subgraphs for an input graph G. k-isomorphism prevents an adversary inference on re-identification of nodes and disclosure of edges in the published k-secure graph, denoted by G_k.

Zou et al. [79] developed *K-Match* algorithm, which has the following techniques: graph partitioning, graph alignment, and edge copy to achieve k-automorphism. According to this algorithm, for each node v in the published graph, denoted by G^*, there exist $k-1$ symmetric nodes to resist any structural attacks. They argue that an adversary cannot distinguish v from its other $k-1$ symmetric nodes based on any structural information, and also it cannot identify the target node with a probability higher than $1/k$. Fung et al. [19] presented a method to k-anonymize a social network while preserving frequent-sharing patterns and maximal frequent-sharing patterns. The purpose of the aforementioned graph anonymization algorithms is to defend against graph structural attacks. Zhang et al. [76] argued that these algorithms are not effective in preserving the privacy of an anonymized heterogeneous information network. Kumar et al. [39] proposed an algorithm based on the fuzzy sets to preserve the privacy of users in online social networks. Their method applies to homogenous networks, in which nodes and edges are of a single type. Generally, all the above works provide prevention against node re-identification, edge disclosure, or both, based on the assumption that the adversary has access to limited background knowledge about a victim. We propose a solution that does not make any assumptions about the adversary's knowledge of victims by adopting the differential privacy model [15], which provides strong privacy guarantees independently of an adversary's background knowledge.

2.2 Network data anonymization under differential privacy models

In the literature, node-differential privacy [8, 14, 30] and edge-differential privacy [10, 25, 71] are the most prevalent formulations for network data anonymization. Node-DP is too strong to get the desired utility in a sparse network. To overcome this problem Kasiviswanathan et al. [30] developed a customized notion of low-sensitivity based projection operators to preserve certain graph statistics. They employed Laplace and Cauchy distributions for output perturbation. In addition, they devised a generic method to apply any differentially-private algorithm for bounded-degree graphs to an arbitrary graph. They assumed that the tail of the degree distribution decreases rapidly, which resembles the characteristics of scale-free networks [29]. A similar problem was also studied by Borgs et al. [8]. They proposed a node-DP algorithm for fitting a high-dimensional statistical model to a sparse network by the use of non-parametric block model approximation. They employed Lipshitz extensions inside the exponential mechanism [46] to control the sensitivity of the score functions. Raskhodnikova et al. [53] proposed some Lipschitz extensions for designing a node-private algorithm to release the degree distribution of a graph. The extensions use convex programming and can be computed in polynomial time. It provides more accurate graph statistics than [30].

Day et al. [14] proposed a graph projection technique to transform an input graph to be θ-degree-bounded for releasing node-private degree distributions. They showed that the sensitivity from the projection is 2θ+1 when releasing a degree histogram, whereas for a cumulative degree histogram the sensitivity is θ+1. Their results indicate a significant improvement over the flow-based approach [53] in releasing node degree distributions. Song et al. [60] proposed a node-private algorithm for online graphs based on the assumption of a bounded maximum degree in the entire graph sequence. They showed that the sequence of differences in the computed graph statistics has low sensitivity, which can yield better privacy-accuracy trade-off.

The group of works [10, 25, 71], based on edge-DP, prevents disclosure of sensitive relationships among nodes. Sala et al. [55] proposed a partition-based approach to divide the dK-2-series into subseries and then inject the noise proportional to its local maximum degree to generate synthetic graphs. They used large privacy parameters $\epsilon \in [5, 100]$ to evaluate degree-based metrics and node-separation metrics on the resulting DP-synthetic graphs. Under stringent privacy parameters (e.g., $\epsilon \leq 1.0$), the error is large because of the high noise injected by the dK-Perturbation Algorithm (dK-PA) into dK-2, resulting in a significant deviation from the original graph. Wang et al. [71] determined degree correlation parameters from the input graph and then enforced edge-DP on graph-model parameters to generate a perturbed graph. They adopted the concept of smooth sensitivity [49] for calibrating noise magnitude to guarantee privacy.

Chen et al. [10] proposed *DER*, in which a notion of correlation parameter k is introduced to provide a similar differential privacy guarantee when releasing network data with the consideration of data correlation. They formed dense regions from an adjacency matrix of input graph by first identifying a good vertex labeling, then adopting a standard quadtree [17] to explore the dense regions, and finally, making use of the exponential mechanism to reconstruct the leaf nodes of a quadtree. They assumed any record in database $\mathcal{D}$ can be correlated to *at most* $k-1$ other records. It is different from k-edge differential privacy [23], where the goal is to protect k edges' collective information but not to conceal the presence of any single edge in the correlated setting. Hu et al. [25] proposed a differentially-private method to protect sensitive edges by converting a deterministic graph into an uncertainty form. In this method, they computed the probability for each edge independently of an original structure of the network to inject uncertainty. Lin et al. [43] proposed a DP-graph structural-clustering algorithm, called *DP-SCAN*, in which they define edge-DP of adjacent graphs, and then add the Laplace noise proportional to the global sensitivity of the function. This algorithm partitions an

input graph into several clusters, bridge connections, and outliers while preserving sensitive information. The above edge-DP methods focus on preserving privacy in *homogeneous networks*, whereas our proposed edge-DP algorithm protects individuals' sensitive links in *heterogeneous networks*. Existing solutions assume that edges are bidirectional and that nodes and edges are of a single type, each. In contrast, heterogeneous networks are characterized by having multiple types of nodes and edges. Thus, solutions that are intended for homogeneous networks will not be able to maintain important semantics and structural information if applied to heterogeneous networks. We propose a solution for anonymizing network data that not only takes into account the types of nodes and edges in a given network, but also considers the direction of edges in the network. It applies to a scenario where data is required to be anonymized before sharing with a third party for research or commercial purposes.

3 PROBLEM DEFINITION

Suppose a HIC wants to publish collected healthcare-network data in a privacy-preserving manner to a data recipient or a data miner for gaining valuable insights, predicting outbreaks of epidemics, preventing chronic diseases, reducing the cost of healthcare delivery, and improving outcomes for patients, etc. The raw data are fused across multiple data sources, resulting in a typical heterogeneous network, $G = (V, E)$, with a node type-mapping function $\varphi : V \rightarrow \mathcal{E}$ and an edge type-mapping function $\psi : E \rightarrow \mathcal{R}$. Each node $v \in V$ belongs to one particular node type in the node type set $\mathcal{E} : \varphi(v) \in \mathcal{E}$, and each edge $e \in E$ belongs to a particular relation type in the relation type set $\mathcal{R} : \psi(e) \in \mathcal{R}$. If two edges belong to the same relation type, the two edges share the same starting node type as well as the ending node type. Fig. 1 illustrates the network schema of a heterogeneous health information network (HHIN), where multiple types of nodes $|\mathcal{E}| > 1$ and multiple types of relations $|\mathcal{R}| > 1$ exist in the network. We illustrate the problem in the following example.

Example 3.1. Consider a heterogeneous directed health network illustrated in Fig. 3. In this example, Patient (P), Disease (D), Medication (M), *and* Lab Test (LT) *are nodes of different types in the node type set $\mathcal{E}$, whereas* contracts $(L_{(1)})$, uses $(L_{(2)})$, *and* undergoes $(L_{(3)})$ *are the types of relationships between nodes in the relation type set $\mathcal{R}$. The number of nodes types $|\mathcal{E}| = 4$, and types of relationships $|\mathcal{R}| = 3$. The total number of nodes $|V| = 14$, and edges $|E| = 26$. Below we discuss potential linkage attacks on a patient's privacy.*

In an indegree linkage attack, an adversary attempts to link structural background knowledge in the context of incoming connections to a node. For a given two types of nodes $\mathcal{U}, \mathcal{V}$ and their relation $L_{(i)}$ in the relation type set $\mathcal{R}$, where $u \in \mathcal{U}$ and $v \in \mathcal{V}$, the set of incoming connections to v_i from u_i with relation-type $L_{(i)}$ are the possible candidates for an indegree linkage. In this example, P2 undergoes LT1, LT2, and LT3. It is safe for P2 because the patient has had multiple lab tests. However, among the lab tests, LT3 is taken only by P2, and none are taken by the other patients. Thus, there is a change of indegree linkage attack.

In an outdegree linkage attack, an adversary attempts to link structural background knowledge in the context of outgoing connections from a node. For a given two types of nodes $\mathcal{U}, \mathcal{V}$ and their relation $L_{(i)}$ in the relation type set $\mathcal{R}$, where $u \in \mathcal{U}$ and $v \in \mathcal{V}$, the set of outgoing connections from u_i to v_i with relation-type $L_{(i)}$ are the possible candidates for an outdegree linkage. In this example, P1, P2, and P3 contract D2. In the context of indegree linkage, D2 is safe because multiple patients have contracted it, so an adversary may not be confident in relation to which patient contracted disease D2. However, among the patients, P3 is only contracted with D2 and none of the other diseases. So, there is a chance of outdegree linkage attack. ■

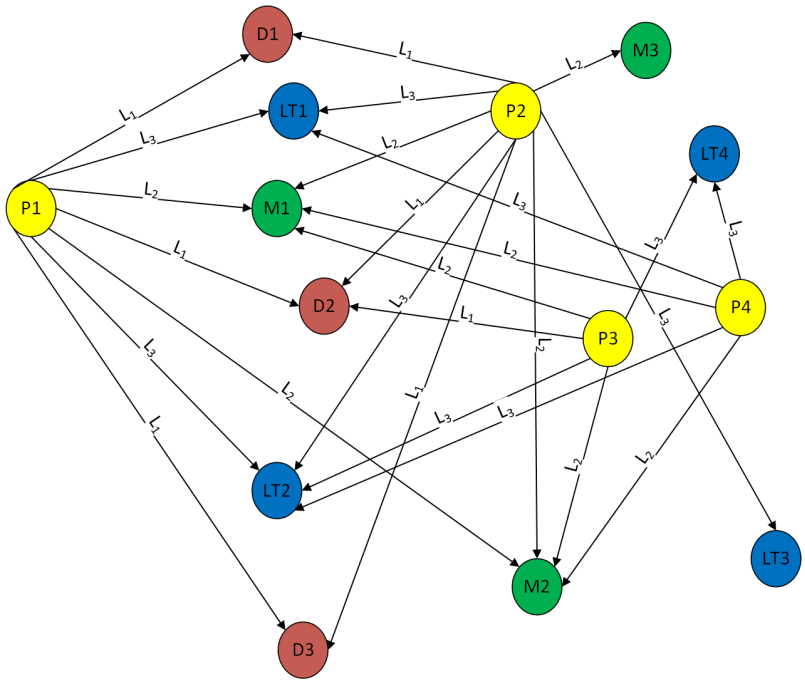


Fig. 3. An example of original health network

In this article, we propose a method to achieve edge-differential privacy with the goal of preventing the aforementioned linkage attacks in a heterogeneous network while releasing the data to a third party for research purposes. It is different from the work based on edge-differential privacy under correlation [10] for a homogeneous undirected network as detailed in previous sections. We first present the definition of edge-differential privacy for heterogeneous networks, followed by our problem statement.

Definition 3.2. (*Edge-differential privacy of heterogeneous networks*). *Given a heterogeneous graph* $G_1 = (V_1, E_1)$, *where* V_1 *or* E_1 *are of multiple types (as per Definition A.2), a heterogeneous graph* $G_2 = (V_2, E_2)$ *is a neighboring graph to* G_1 *if the difference between* G_1 *and* G_2 *is* at most *one edge (i.e.,* $|V_1 \oplus V_2| + |E_1 \oplus E_2| = 1$*). A sanitization mechanism* $\mathcal{M}$ *provides* edge-differential privacy *if for any two neighboring heterogeneous graphs, and for any possible sanitized graph* $\hat{G}$, *we have*

$$Pr[\mathcal{M}(G_1) = \hat{G}] \leq e^{\epsilon} \times Pr[\mathcal{M}(G_2) = \hat{G}]. \blacksquare$$

Problem (Edge-differential privacy in HHIN). Given a heterogeneous health information network $G = (V, E)$, where each node $v \in V$ belongs to one particular node type in the node type set $\mathcal{E}$, and each edge $e \in E$ belongs to a particular relation type in the relation type set $\mathcal{R}$, nodes are of multiple types $|\mathcal{E}| > 1$ and relationships are of multiple types $|\mathcal{R}| > 1$, and privacy budget ϵ, the goal is two-fold:

- To publish an anonymized version of network G, denoted by G', that protects patients' privacy by preventing adversarial inference on each incoming and outgoing edge $e \in E$ in accordance with edge-differential privacy.

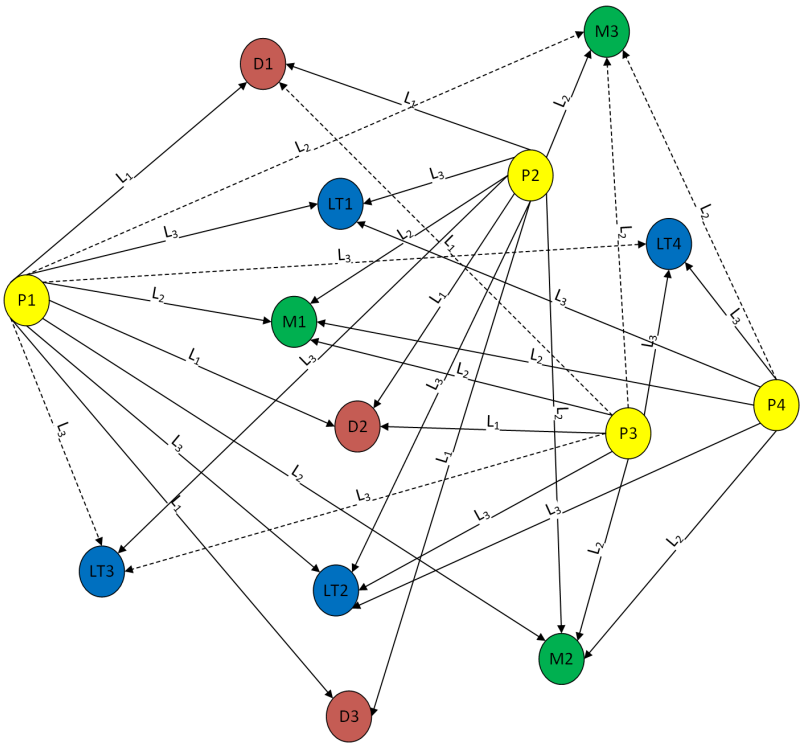


Fig. 4. Anonymized version of the example health network

- To minimize the impact of anonymization on all edges E in G by reducing the errors generated by the mean absolute error, the average relative error, and the Kullback-Leibler divergence, as defined in Eqs. 10, 11, and 12, respectively.

4 PROPOSED SOLUTION

In this section, we present an edge-based differentially-private solution to protect the sensitive links of a patient from adversarial inbound and outbound attacks in a heterogeneous health network, while minimizing information loss inflicted on edges. Our solution addresses the concern of a health information custodian (HIC) on preserving privacy and the concern of a data recipient on information utility. Section 4.1 presents an overview of our proposed *DiffHetNet*, an algorithm based on edge-differential privacy for anonymizing heterogeneous network data. Section 4.2 presents the operations for exploring subgraphs favoring lower scores when selecting candidate nodes. Section 4.3 presents the operations for generating noisy counts. Section 4.4 presents the operations for exploring subgraphs favoring higher scores when selecting candidate nodes. Section 4.5 presents the process of edge perturbation in the network. Section 4.6 presents the privacy and utility analysis.

4.1 Overview

We first provide a high-level description of our proposed method in Algorithm 1, followed by detailed discussions of each step.

4.1.1 High-level description. We study the problem of protecting patients' privacy when sharing healthcare data. We propose a privacy-preserving solution to this problem. However, our solution is also applicable to other network-data publishing scenarios sharing the same privacy and utility concerns.

The intuition is that many people would not suffer from a critical disease resulting in fewer connections from patient-nodes to the disease (i.e., non-patient) node. The probability of finding a potential sensitive connection of a patient to the disease is very high. We discover those vulnerable connections and intend to protect privacy of patient sensitive information. Conversely, many people would have a common disease that is not treated as sensitive. It does not reflect threat to a patient privacy because probability of identifying a patient is very low. Our proposed solution is based on edge-differential privacy to anonymize a heterogeneous network. We impose edge-differential privacy on the relationships between patient nodes and non-patient nodes to prevent adversarial indegree and outdegree linkage attacks, i.e., identifying sensitive relationships. We provide an illustration of privacy attacks in Example 3.1. Compared with existing works that preserve privacy in homogeneous networks [10, 25, 71], our proposed solution not only considers different types of nodes and edges in a given network, but it also takes into account the direction of edges in the network. Our solution takes a heterogeneous graph G and a privacy budget ϵ as inputs and outputs a differentially-private graph G'.

We observe that nodes with a low number of directed edges are more vulnerable to adversary linkage attacks than nodes with a high number of edges. Our aim is to identify such nodes, i.e., nodes with a low number of directed edges. To do so, we consider the following two cases based on the direction of edges: *indegree* linkage to identify patients, and *outdegree* linkage with respect to the adversary's confidence about a target patient's relationship. In the first case, the identified node is a non-patient node, e.g., a disease or lab test. Consequently, the privacy of patients connected to such a node is at risk because very few patients have a relationship with this node, e.g., contracted a disease or received a lab test. In the second case, the identified node is a patient node. The privacy of the patient node is at risk because the patient's node is connected to only a few other nodes. A patient node with a low number of (outdegree) edges results in a high level of an attacker's confidence with respect to a particular relationship associated with this node, e.g., a patient contracting a disease.

Based on the above observation, we want to protect vulnerable nodes from identification attacks by connecting these nodes to less vulnerable ones. Our proposed solution accomplishes this goal across two phases. The first phase searches the network and identifies nodes with a low number of directed edges. The second phase preserves information utility by choosing nodes with a high number of directed edges, since these nodes are less vulnerable to identification attacks. After that, the second phase connects the nodes picked in this phase to the nodes that were picked in the first phase to protect their inbound and outbound connections.

4.1.2 Algorithm. Algorithm 1 presents the anonymization operations, which we split into two phases. Before describing the lines of Algorithm 1, we explain how the input privacy budget ϵ is distributed throughout the algorithm. The input privacy budget ϵ is divided into three portions in Line 1. The first portion, denoted by ϵ_{slo}, is consumed when exploring lower-scoring candidate nodes. The second portion, denoted by ϵ_{nc}, is utilized when generating a noisy count for each candidate node. The third portion, denoted by ϵ_{shi}, is consumed when determining higher-scoring candidate nodes. ϵ_{slo}

Algorithm 1 DiffHetNet Algorithm

Input: Original network $G = (V, E)$, privacy budget ϵ
Output: Anonymous differentially-private network G'

1: Allocation of privacy budget $\epsilon \leftarrow \epsilon_{slo} + \epsilon_{nc} + \epsilon_{shi}$; /* for indegree and outdegree of directed network*/
2: Set $\alpha_b = dir$; // input direction
3: Lower-scoring candidates $C_{lo}^{\alpha_b} \leftarrow exploreSGsInOutDegFavLowScores(\alpha_b, \epsilon_{slo}^{\alpha_b}, G)$;
4: **for** $c_i \in C_{lo}^{\alpha_b}$ **do**
5: $\quad \epsilon_{nc}^{\alpha_b}{}' \leftarrow \frac{\epsilon_{nc}^{\alpha_b}}{|C_{lo}^{\alpha_b}|}$;
6: $\quad$ Noisy count $Nc^{\alpha_b} \leftarrow genNoisyCount(c_i, \alpha_b, \epsilon_{nc}^{\alpha_b}{}', G)$;
7: $\quad$ Higher-scoring candidates $C_{hi}^{\alpha_b} \leftarrow findCandsFavHighScoresProtectInOutDeg(Nc^{\alpha_b}, c_i, \alpha_b, \epsilon_{shi}^{\alpha_b}, G)$;
8: $\quad$ Anonymized sub-network $\tilde{G}^{\alpha_b} \leftarrow edgePerturbation(\forall C_{hi}^{\alpha_b}, Nc^{\alpha_b}, c_i, \alpha_b, G)$;
9: **end for**
10: Generate G' from $\tilde{G}^{\alpha_b}$;
11: **return** G';

is allocated to the first phase, and ϵ_{nc} and ϵ_{shi} are allocated to the second phase. We divide ϵ such that the summation of ϵ_{slo} and ϵ_{shi} constitutes the majority of ϵ (i.e., 80%), and ϵ_{nc} (i.e., 20%) is less than ϵ_{slo} and ϵ_{shi}, respectively. The reason for allocating a larger portion of ϵ to ϵ_{slo} (phase 1) is because Algorithm 1 in Line 3 will attempt to discover vulnerable candidate nodes (due to having fewer incoming or outgoing connections). In order to accurately discover nodes that are more prone to adversarial attacks, differential privacy necessitates allocating a larger portion of privacy budget. Similarly, the reason for allocating a larger portion of the budget to ϵ_{shi} (phase 2) in Line 7 is because we intend to preserve more information utility by choosing candidates that are less vulnerable to identification attacks.

Algorithm 1 in Line 3 explores subgraphs in the input network G and picks candidate nodes having lower scores, denoted by $C_{lo}^{\alpha_b}$. The score for each candidate is computed using the degree-centrality function that yields a real-valued score. We design a procedure that uses the exponential mechanism to favor candidates with lower scores. Next, we generate a noisy count, denoted by Nc^{α_b}, that represents the number of newly-generated edges to be added to each node $c_i \in C_{lo}^{\alpha_b}$ by using the Laplace mechanism in Line 6. Based on the generated noisy count, Line 7 scans the input network G and uses the exponential mechanism to pick nodes favoring higher scores, denoted by $C_{hi}^{\alpha_b}$. Subsequently, we protect the corresponding inbound and outbound connections of each node c_i by adding edges from $C_{hi}^{\alpha_b}$, or removing corresponding edges, to have an anonymized version of sub-network $\tilde{G}^{\alpha_b}$ in Line 8. Finally, the differentially-private sub-networks of both indegree and outdegree are combined to form an anonymized network G'.

4.2 Selecting candidates favoring lower scores

The rationality of exploring subgraphs in the heterogeneous network G is that nodes having fewer incoming or outgoing connections are more prone to adversarial attacks. Procedure 1 attempts to discover vulnerable candidate nodes in the network. It takes a heterogeneous network G, a privacy budget $\epsilon_{slo}^{\alpha_b}$, and the type of degree direction $\alpha_b = \{in|out\}$ as inputs, and it outputs a list of candidate nodes having lower scores, denoted by $C_{lo}^{\alpha_b}$.

Line 1 allocates a portion of the given privacy budget to each candidate by dividing the given budget from the total number of nodes under a specified direction. Line 4 computes the score for each node v using the normalized degree-centrality metric for a directed graph that yields a real-valued score for each node v under the node type V_{τ^i} in the node type set $\mathcal{E}$ and the corresponding relation-type $L_{(i)}$ in the relation type set $\mathcal{R}$. It is defined as follows:

$$CD(G, v, L_{(i)}^{\alpha_b}) = \frac{d^{\alpha_b}(v)_{v \in V_{\tau^i}, L_{(i)}^{\alpha_b} \in \mathcal{R}}}{|V| - 1} \tag{1}$$

Procedure 1 *exploreSGsInOutDegFavLowScores* Procedure

Input: Original network $G = (V, E)$
Input: Privacy budget $\epsilon_{slo}^{\alpha_b}$, direction α_b
Output: Lower-scoring candidates $C_{lo}^{\alpha_b}$

1: $\epsilon_{slo}^{\alpha_b\prime} \leftarrow \frac{\epsilon_{slo}^{\alpha_b}}{|V_{\tau^i}^{\alpha_b}|}$;
2: $C_{lo}^{\alpha_b} \leftarrow \emptyset$;
3: **for** each pair of neighboring vertices $v_i, v_j \in V$ **do**
4: Compute the score for every $v \in V_{\tau^i}$ according to Eq. (1);
5: Select $v \in V_{\tau^i}$ with probability $\propto \exp(\frac{\epsilon_{slo}^{\alpha_b\prime}}{2\Delta \mathsf{u}} \cdot \mathsf{u}(G, v, L_{(i)}^{\alpha_b}))$ favoring lower score;
6: Add v to the list $C_{lo}^{\alpha_b}$;
7: **end for**
8: **return** $C_{lo}^{\alpha_b}$;

Example 4.1. We continue from Example 3.1. Consider the type of degree direction $\alpha_b = \{out\}$, i.e., representing the outgoing connections, the type of node $V_{\tau^i} = \{P\}$, i.e., representing a patient's node label, and the relation-type $L_{(1)} = \{contracts\}$, i.e., representing the relationship to the adjacent node(s) of type $V_{\tau^j} = \{D\}$, i.e., representing a disease's node label, in Fig. 3. The normalized degree-centrality scores of nodes $\{P1, P2, P3\} = \{0.23, 0.23, 0.08\}$ by Eq. (1), whereas the number of outgoing connections d^{out} of nodes $\{P1, P2, P3\} = \{3, 3, 1\}$. ■

DiffHetNet makes novel use of the exponential mechanism in Line 5. In this step, the exponential mechanism favors lower scores to choose a candidate node v from a set of candidate nodes under the node type V_{τ^i}. It is presented in Theorem 4.2. The sensitivity of $\Delta \mathsf{u}$ is 1, because the addition or removal of a single edge in G would change $CD(G, v, L_{(i)}^{\alpha_b})$ by at most 1.

Theorem 4.2. *Choosing a candidate score from a set of candidate scores satisfies ϵ'-differential privacy.*

Proof. Let $Cand_i$ be the set of candidate scores from which a single score is to be chosen for lower scores. Our algorithm selects a candidate score $v_i \in Cand_i$ with the following probability:

$$\frac{\exp(\frac{\epsilon_{slo}^{\alpha_b\prime}}{2\Delta \mathsf{u}} \cdot \mathsf{u}(G, v_i, L_{(i)}^{\alpha_b}))}{\sum_{v \in Cand_i} \exp(\frac{\epsilon_{slo}^{\alpha_b\prime}}{2\Delta \mathsf{u}} \cdot \mathsf{u}(G, v, L_{(i)}^{\alpha_b}))} \tag{2}$$

where $\mathsf{u}(G, v_i, L_{(i)}^{\alpha_b})$ is a score computed from a utility function according to Eq. (1), and $\Delta \mathsf{u}$ is the sensitivity of the utility function u. According to Theorem C.6, selecting a score with probability proportional to $\exp(\frac{\epsilon' \mathsf{u}(G,t)}{2\Delta \mathsf{u}})$ satisfies ϵ'-differential privacy. □

The scores are inverted for the exponential mechanism to favor lower-scoring candidates. At each iteration, the lower-scoring candidate selected by the exponential mechanism is added to the list $C_{lo}^{\alpha_b}$ in Line 6. This process runs until equilibrium is reached or there are no more lower-scoring candidates in the network. Finally, the list of selected lower-scoring candidate nodes is returned by this procedure.

4.3 Generating noisy counts

After obtaining the list of lower-scoring candidate nodes $C_{lo}^{\alpha_b}$, Procedure 2 generates a noisy count for each candidate c_i in the list. A portion of the given budget, denoted by $\epsilon_{nc}^{\alpha_b\prime}$, is allocated to each candidate by dividing it from the total number of lower-scoring candidate nodes. Line 1 generates a noisy count Nc^{α_b} from the Laplace distribution

Procedure 2 *genNoisyCount* Procedure

Input: Original network $G = (V, E)$
Input: Privacy budget $\epsilon_{nc}^{\alpha_b\prime}$
Input: Selected candidate c_i, direction α_b
Output: Noisy count Nc^{α_b}

```
1: Nc^{α_b} ← Lap(1/ε_nc^{α_b}′);
2: if Nc^{α_b} < 0 then
3:     Nc^{α_b} = 0;
4: end if
5: if Nc^{α_b} ≥ 1 then
6:     if c_i ∈ V_{τ^i} then
7:         Nc^{α_b} = Nc^{α_b}  mod (ln |𝕌_{V_{τ^j}}|);
8:     end if
9: end if
10: return Nc^{α_b};
```

$\mathtt{Lap}(1/\epsilon_{nc}^{\alpha_b\prime})$. It can be a positive or negative value. The noise count Nc^{α_b} of a selected candidate c_i is calibrated according to the potential connecting candidate c_j' of node type V_{τ^j} by considering the set of all possible candidates that can exist in any network dataset. Formally, it is defined as follows:

$$Nc^{\alpha_b} = Nc^{\alpha_b} \mod (\ln |\mathbb{U}_{V_{\tau^j}}|) \tag{3}$$

where $|\mathbb{U}_{V_{\tau^j}}|$ represents the size of the universal set of all possible nodes under the given node type that can exist in any network data.

4.4 Selecting candidates favoring higher scores

The rationality of selecting nodes with a high number of directed edges in the heterogeneous network G is to preserve information utility. These nodes are less vulnerable to identification attacks, and drawing edges from them have a low impact on the overall structure of the network. The composition of a heterogeneous network entails nodes and edges to be of multiple types, so the centrality scores for the influential nodes pose different semantics according to their respective types and the incoming and outgoing directions of their edges.

Procedure 3 takes the network G, a privacy budget $\epsilon_{shi}^{\alpha_b}$, a noisy count Nc^{α_b}, a candidate node c_i, and the type of degree direction α_b as inputs and outputs a list of candidate nodes having higher scores, denoted by $C_{hi}^{\alpha_b}$. Line 1 allocates a portion of the given privacy budget to each candidate by dividing the given budget from the product of the total number of lower-scoring candidate nodes and the noisy count. Line 5 computes the score for each node v using the normalized degree-centrality metric for a directed graph by Eq. (4) that yields a real-valued score for each node v under the node type V_{τ^j} in the node type set $\mathcal{E}$. $\widetilde{\alpha_b}$ represents the opposite degree direction. The score is computed as follows:

$$CD(G, v, \widetilde{\alpha_b}) = \frac{d^{\widetilde{\alpha_b}}(v)_{v \in V_{\tau^j}}}{|V| - 1} \tag{4}$$

Example 4.3. We continue from Example 3.1. Let us assume that Procedure 1 returns $\{P3 = 0.08\}$ *as one of the lower-scoring outdegree candidate nodes having the relation-type* $L_{(1)} = \{contracts\}$ *with D2. To protect its outbound connection we need to find indegree candidate nodes having higher scores based on the exponential mechanism. The type of a potential candidate's degree direction is* $\widetilde{\alpha_b} = \{in\}$*, i.e., representing the incoming connections, the type of node* $V_{\tau^j} = \{D\}$*, i.e., representing a disease's node label in Fig. 3. The potential candidate* $D1$*'s centrality score is computed by Eq. (4) is* 0.15. ■

DiffHetNet makes novel use of the exponential mechanism in Line 6. In contrast to the presented Theorem 4.2, this step utilizes the exponential mechanism to choose a candidate node v favoring a higher score from a set of candidate

Procedure 3 *findCandsFavHighScoresProtectInOutDeg* Procedure

Input: Original network $G = (V, E)$
Input: Privacy budget $\epsilon_{shi}^{\alpha_b}$, Noisy count Nc^{α_b}
Input: Selected candidate c_i, direction α_b
Output: Higher-scoring candidates $C_{hi}^{\alpha_b}$

1: $\epsilon_{shi}^{\alpha_b\prime} \leftarrow \frac{\epsilon_{shi}^{\alpha_b}}{|C_{lo}^{\alpha_b}|\cdot|Nc^{\alpha_b}|}$;
2: $C_{hi}^{\alpha_b} \leftarrow \emptyset$;
3: **while** $|c_i| < |Nc^{\alpha_b}|$ **do**
4: **for** each pair of neighboring vertices $v_i, v_j \in V$ **do**
5: Compute the score for every $v \in V_{\tau^j}$ according to Eq. (4);
6: Select $v \in V_{\tau^j}$ with probability $\propto \exp(\frac{\epsilon_{shi}^{\alpha_b\prime}}{2\Delta \mathrm{u}} \cdot \mathrm{u}(G, v, \widetilde{\alpha_b}))$ favoring higher score;
7: Add v to the list $C_{hi}^{\alpha_b}$;
8: **end for**
9: **end while**
10: **return** $C_{hi}^{\alpha_b}$;

Procedure 4 *edgePerturbation* Procedure

Input: Original network $G = (V, E)$
Input: Candidates $C_{hi}^{\alpha_b}$, Noisy count Nc^{α_b}
Input: Selected candidate c_i, direction α_b
Output: Anonymized network $\tilde{G}^{\alpha_b}$

1: **if** $Nc^{\alpha_b} == 0$ **then**
2: Remove corresponding edges of c_i from network $\tilde{G}^{\alpha_b}$;
3: **end if**
4: **while** $c_j' \in C_{hi}^{\alpha_b}$ **do**
5: **if** $c_i \in V_{\tau^i}^{\alpha_b}$ **then**
6: Add edge (c_j', c_i) or vice versa to network $\tilde{G}^{\alpha_b}$;
7: Set the corresponding relation type $L_{(i)}$;
8: **end if**
9: **end while**
10: **return** $\tilde{G}^{\alpha_b}$;

nodes under the node type V_{τ^j}. At each iteration, the higher-scoring candidate selected by the exponential mechanism is added to the list $C_{hi}^{\alpha_b}$ in Line 7. This process repeats until the number of candidate nodes is less than the size of the noisy count. Finally, the list of selected higher-scoring candidate nodes is returned by this procedure.

4.5 Edge perturbation

This procedure takes the network G, lower-scoring candidate node c_i selected by Procedure 1, a noisy count Nc^{α_b} by Procedure 2, list of higher-scoring candidate nodes $C_{hi}^{\alpha_b}$ by Procedure 3, and the type of degree direction α_b as inputs, and it outputs an anonymized version of sub-network $\tilde{G}^{\alpha_b}$. It protects the corresponding inbound or outbound connections of each candidate node c_i in the list of lower-scoring candidates $C_{lo}^{\alpha_b}$ by either removing the corresponding edges from $\tilde{G}^{\alpha_b}$ or by adding edges from higher-scoring candidate nodes $C_{hi}^{\alpha_b}$.

Line 2 removes the corresponding edge pairs (c_i, c_j) or vice versa of candidate node c_i from $\tilde{G}^{\alpha_b}$ when the noisy count is 0. Line 5 matches the selected candidate's node type $V_{\tau^i}^{\alpha_b}$ along with the degree direction α_b, and then it adds an edge (c_j', c_i) or vice versa (Line 6) if it does not exist already in the given network G or was added previously in the $\tilde{G}^{\alpha_b}$. Next, the corresponding relationship $L_{(i)}$ is assigned based on the types of source and destination nodes in Line 7. This process repeats for each potential candidate c_j' in the list of higher-scoring candidate nodes $C_{hi}^{\alpha_b}$. Finally, the anonymized version of sub-network $\tilde{G}^{\alpha_b}$ is returned by this procedure.

Example 4.4. Fig. 4 illustrates a possible anonymized version of the example health network. We continue from Example 3.1. Let us assume that Procedure 3 returns $\{D1 = 0.15\}$ *as one of the higher-scoring indegree candidate nodes for the node* $P3$ *by Procedure 1. The corresponding edge is added between* $P3$ *and* $D1$*, and the relationship* $L_{(1)}$ *is assigned based on the types of source and destination nodes. Now consider that Procedure 1 returns* $\{LT4 = 0.15\}$ *as one of the lower-scoring indegree candidate nodes having the relation-type* $L_{(3)} = \{undergoes\}$ *with* $P3$ *and* $P4$*. To protect its inbound connection, we need to find higher-scoring outdegree candidate nodes based on the exponential mechanism. Suppose Procedure 3 returns* $\{P1 = 0.15\}$ *as one of the higher-scoring outdegree candidate nodes for the node* $LT4$*. The corresponding edge is added between* $P1$ *and* $LT4$*, and the relationship* $L_{(3)}$ *is assigned based on the types of source and destination nodes.* ■

4.6 Analysis

In this section, we present the privacy and utility analysis of Algorithm 1.

4.6.1 Privacy analysis. We prove that Algorithm 1 satisfies ϵ-differential privacy over heterogeneous network data under the given network schema of Fig. 1.

Theorem 4.5. *For a given privacy budget* ϵ*, Algorithm 1 is* ϵ*-differentially private over heterogeneous network data.*

Proof. Algorithm 1 picks lower-scoring candidates from a set of candidate nodes by employing the exponential mechanism according to Theorem 4.2 in Line 3. Each candidate is dedicated with a privacy budget portion $\epsilon_{slo}^{\alpha_b\prime} = \frac{\epsilon_{slo}^{\alpha_b}}{|V_{\tau^i}^{\alpha_b}|}$ by leveraging *sequential composition* property (Theorem C.7). A noisy count is generated for each candidate $c_i \in C_{lo}^{\alpha_b}$ by drawing noise from Laplace distribution $\mathsf{Lap}(\frac{\Delta f}{\epsilon})$ (according to Theorem C.5) using a privacy budget portion $\epsilon_{nc}^{\alpha_b\prime} = \frac{\epsilon_{nc}^{\alpha_b}}{|C_{lo}^{\alpha_b}|}$ in Line 6. Next, for each candidate c_i, the algorithm picks potential higher-scoring candidate(s) from a set of candidate nodes using the exponential mechanism in Line 7. Each candidate is dedicated with a privacy budget portion $\epsilon_{shi}^{\alpha_b\prime} = \frac{\epsilon_{shi}^{\alpha_b}}{|C_{lo}^{\alpha_b}|\cdot|Nc^{\alpha_b}|}$ by leveraging sequential composition property. Finally, the algorithm post-processes [35] the differentially private inputs $c_i \in C_{lo}^{\alpha_b}$, Nc^{α_b}, and $C_{hi}^{\alpha_b}$ to perturb the network. Hence, Algorithm 1 is ϵ-differentially private because $\epsilon = \epsilon_{slo} + \epsilon_{nc} + \epsilon_{shi}$ by the property of sequential composition (Theorem C.7). □

4.6.2 Utility analysis. We measure the utility loss on the anonymized network with respect to the original network by mean absolute error, average relative error, and Kullback–Leibler divergence presented in Section D.

Considering the network schema of Fig. 1, the goal is to generate a sanitized graph G' so as close to G as possible to minimize the error $\sum_{i=1}^{|V|}|CD(G', v_i) - CD(G, v_i)|$. When G' is identical to G, $\sum_{i=1}^{|V|}|CD(G', v_i) - CD(G, v_i)| = 0$; when G' is totally different from G, $\sum_{i=1}^{|V|}|CD(G', v_i) - CD(G, v_i)| = |V_{\tau^i}^{\alpha_b}| \cdot \sum_{j=1}^{k}|V_{\tau^j}^{\widetilde{\alpha_b}}|$, where $i \neq j$.

4.6.3 Conditions. In Section 4.1.2 we discuss the distribution of privacy budget ϵ and its consumption across all the phases of Algorithm 1. The utility guarantee of our proposed algorithm is dependent on the privacy parameter ϵ. Consider s and $\widetilde{s}$ are the scores of a node $v \in V_{\tau^i}^{\alpha_b}$ in G and G', respectively. We specify the conditions for comparing scores. When $\widetilde{s} < s$: (1) no new edge is added to a node v, and an existing edge has removed from the node v, and (2) a worst case would be when all existing edges are removed from the node v; when $\widetilde{s} = s$: (1) no new edge is added to or removed from a node v, and (2) an equal number of edges are added and removed from the node v; when $\widetilde{s} > s$: (1) a new edge is added to a node v while maintaining all existing edges of the node v, and (2) a worst case would be when newly added edges to the node v are reached to the maximum.

Table 1. Statistics of the datasets

Dataset	$\|V\|$	$\|E\|$	Edge Density
ca-GrQc	5, 242	28, 980	0.001055
wiki-Vote	7, 115	103, 689	0.002049
MIMIC-T1	13, 947	103, 023	0.000530
MIMIC-MultiType	5, 786	183, 795	0.005491

5 EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of our *DiffHetNet* algorithm in terms of both information utility and efficiency. We compare our method *DiffHetNet* with the *DER* [10] method and its variant *DE* and a random graph [12] (referred to as *Random*). In *DE*, the step of *ArrangeEdge* is simply replaced by randomly inserting edges in each leaf region based on the noisy count. We use three real-life datasets, namely *ca-GrQc*[1], *wiki-Vote*[1], and *MIMIC*[2] from three different types of networks. *ca-GrQc* is an undirected network, extracted from the scientific collaboration network of arXiv GR-QC (General Relativity and Quantum Cosmology) category, where two authors are connected if they co-authored at least one paper. *wiki-Vote* is a directed network extracted from the Wikipedia adminship voting network, where a Wikipedia user is considered for promotion to adminship based on the community votes in favor of or against the promotion. *MIMIC* contains health-related data from a large number of Intensive Care Unit (ICU) patients. It integrates de-identified, comprehensive health data of patients admitted to the Beth Israel Deaconess Medical Center in Boston, Massachusetts. It is accessible to researchers internationally under a data use agreement. *MIMIC-T1* represents a network of a single relation type having nodes that are of different types, i.e., the number of node types $|\mathcal{E}| = 2$, and types of relationships $|\mathcal{R}| = 1$, whereas *MIMIC-MultiType* represents a network of multiple nodes and relations types, i.e., the number of node types $|\mathcal{E}| = 4$, and types of relationships $|\mathcal{R}| = 3$. The statistics of the datasets are shown in Table 1. All experiments were performed on a PC with Intel Core i7 2.80GHz and 16GB RAM.

5.1 Measuring information loss

We measure the information loss on the anonymized network with respect to the original network by mean absolute error, average relative error, and Kullback–Leibler divergence introduced in Section D.

Fig. 5 presents the mean absolute error (MAE) by the *DiffHetNet* method. Fig. 5(a) depicts the MAE under privacy budget ϵ varying from 0.6 to 1.0 on the *MIMIC-MultiType* dataset. It exhibits no change with the increase in ϵ. Fig. 5(b) depicts the MAE under a privacy budget varying from 0.6 to 1.0 while fixing the data size to be $0.4 \times |V|$ on the *ca-GrQc*, *wiki-Vote*, and *MIMIC-T1* datasets. The absolute errors on the *ca-GrQc* dataset are slightly greater than the other datasets. However, they remain unchanged with the increase in ϵ and are consistently small on all datasets. Fig. 5(c) depicts the MAE under varying data size while fixing the privacy budget to be $\epsilon = 1.0$ on the *ca-GrQc*, *wiki-Vote*, and *MIMIC-T1* datasets. It generally decreases with the increase in size on all datasets. The results suggest that *DiffHetNet* well preserves the global structure of the anonymized network.

Fig. 6 presents the average relative error (ARE) by the *DiffHetNet* method. Fig. 6(a) depicts the ARE under privacy budget ϵ varying from 0.6 to 1.0 on the *MIMIC-MultiType* dataset. It generally increases monotonically with the increase in ϵ. Fig. 6(b) depicts the ARE under varying privacy budget from 0.6 to 1.0 while fixing the data size to be $0.4 \times |V|$ on the *ca-GrQc*, *wiki-Vote*, and *MIMIC-T1* datasets. It exhibits non-decreasing monotonicity with the increase in ϵ on

[1]It is publicly available in the Stanford large network dataset collection at: http://snap.stanford.edu/data/index.html
[2]Available at: https://mimic.physionet.org

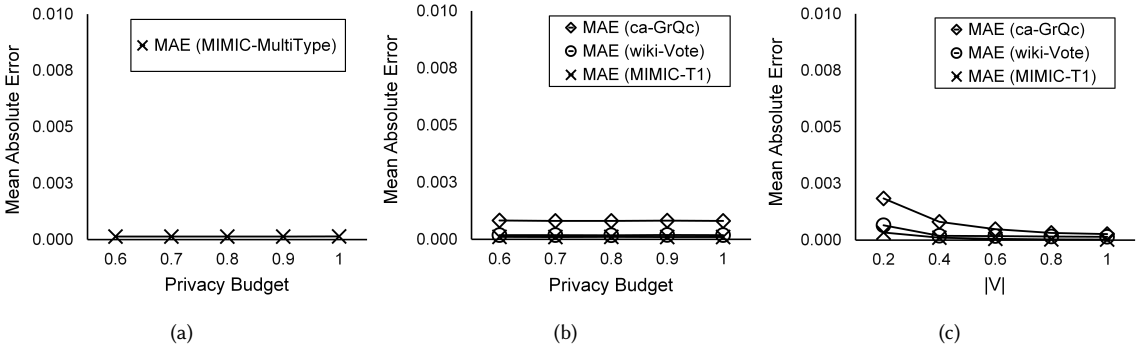


Fig. 5. Mean absolute error by *DiffHetNet* method under varying ϵ in (a) and (b), and fixed $\epsilon = 1.0$ and varying data size in (c)

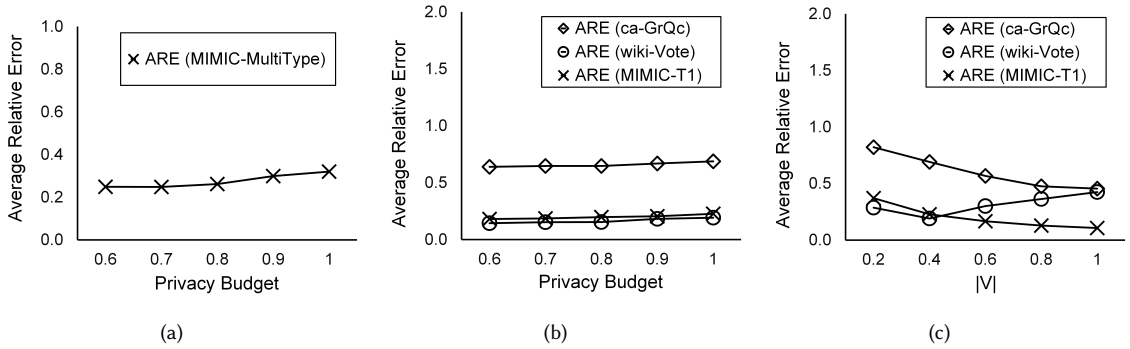
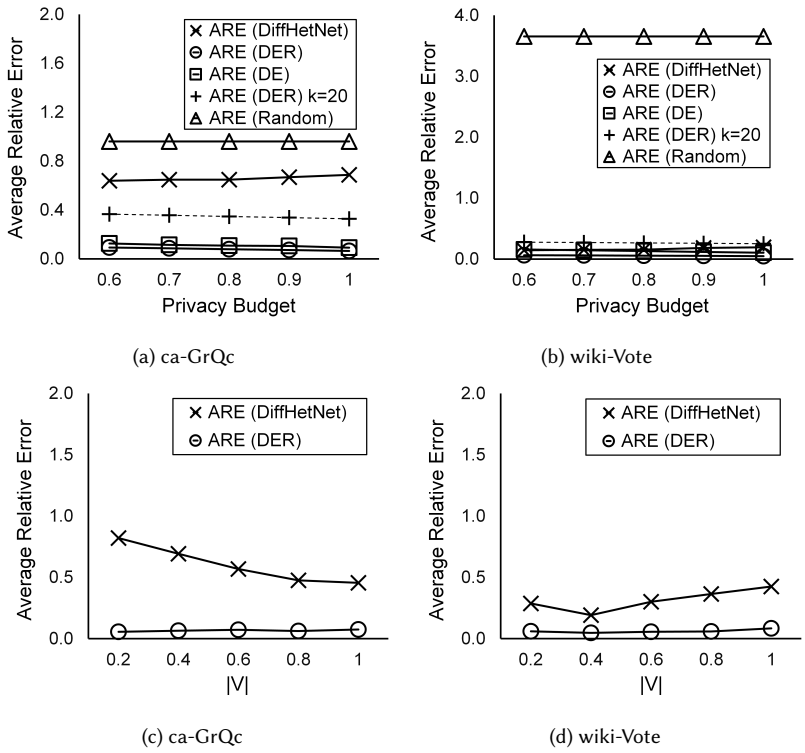


Fig. 6. Average relative error by *DiffHetNet* method under varying ϵ in (a) and (b), and fixed $\epsilon = 1.0$ and varying data size in (c)

all datasets. The relative errors on the *ca-GrQc* dataset are higher than the other datasets because more vulnerable candidate nodes are protected in the network. Fig. 6(c) depicts the ARE under varying data size while fixing the privacy budget to be $\epsilon = 1.0$ on the *ca-GrQc*, *wiki-Vote*, and *MIMIC-T1* datasets. The relative errors generally decrease on *ca-GrQc* and *MIMIC-T1* datasets with the increase in data size, while on *wiki-Vote* they first decrease when data size increases from $0.2 \times |V|$ to $0.4 \times |V|$ and later increase with the increase in data size. The reason for this non-monotonicity is that the addition of noise considerably changes the degree-centrality scores for the potentially vulnerable nodes in the anonymized network.

Fig. 7 presents the comparison of different methods on average relative error (ARE). Figs. 7(a) and 7(b) depict the ARE of *DiffHetNet*, *DER*, *DE*, and *Random* under varying privacy budget ϵ from 0.6 to 1.0 while fixing the data size to be $0.4 \times |V|$ on *ca-GrQc* and *wiki-Vote* datasets. The relative errors of *DER* and its variant *DE*, when $k = 1$ (static correlation parameter) are smaller on both *ca-GrQc* and *wiki-Vote* datasets. However, their relative errors increase with an increase of k. It is observed that *DiffHetNet* performs better than *DER* when the correlation parameter $k = 20$, and it is closer to *DE* when $k = 1$ on the *wiki-Vote* dataset. The relative errors of *Random* are greater than the other methods in all settings. Our method *DiffHetNet* does not specify static correlation parameter k because of the dynamicity nature of the network. Figs. 7(c) and 7(d) depict the ARE of *DiffHetNet* and *DER* under varying data size, while fixing the privacy budget to be $\epsilon = 1.0$ on *ca-GrQc* and *wiki-Vote* datasets. The relative errors of *DiffHetNet* decrease on *ca-GrQc* when data size



(a) ca-GrQc (b) wiki-Vote

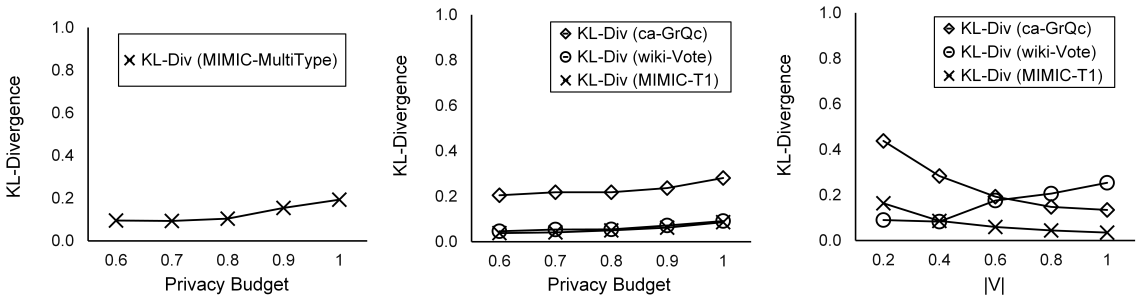
(c) ca-GrQc (d) wiki-Vote

Fig. 7. Comparison of *DiffHetNet*, *DER*, *DE*, and *Random* methods on average relative error under varying ϵ in (a) and (b), and *DiffHetNet* and *DER* on average relative error under varying data size in (c) and (d)

increases, while on the *wiki-Vote* dataset they first decrease when data size increases from $0.2 \times |V|$ to $0.4 \times |V|$, and later increase with the increase in data size. The relative errors of the *DER* method on both datasets are small because the correlation parameter is set as low $k = 1$.

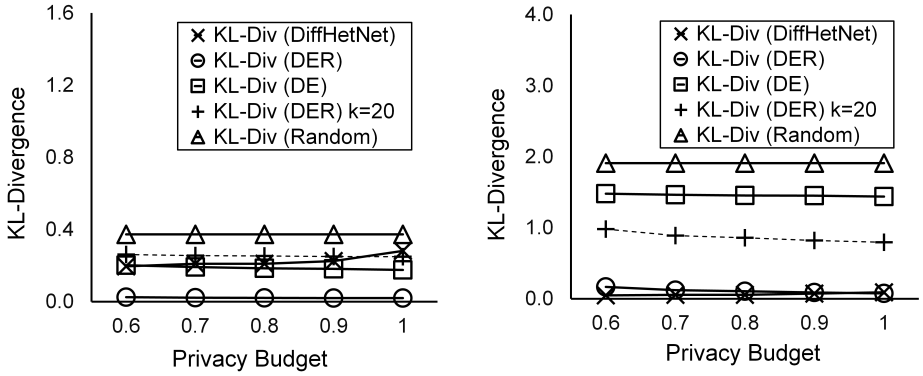
Fig. 8 presents the KL-divergence by the *DiffHetNet* method. Fig. 8(a) depicts the KL-divergence under varying privacy budget ϵ from 0.6 to 1.0 on the *MIMIC-MultiType* dataset. It generally increases monotonically with the increase in ϵ. When $\epsilon = 1.0$, it reaches 0.19. Fig. 8(b) depicts the KL-divergence under varying privacy budget from 0.6 to 1.0 while fixing the data size to be $0.4 \times |V|$ on the *ca-GrQc*, *wiki-Vote*, and *MIMIC-T1* datasets. It exhibits non-decreasing monotonicity with the increase in ϵ on all datasets. The KL divergences on the *ca-GrQc* dataset are higher than the other datasets. The maximum difference on them is 0.19 when $\epsilon = 1.0$. Fig. 8(c) depicts the KL-divergence under varying data size while fixing the privacy budget to be $\epsilon = 1.0$ on the *ca-GrQc*, *wiki-Vote*, and *MIMIC-T1* datasets. The KL divergences exhibit decreasing monotonicity on *ca-GrQc* and *MIMIC-T1* datasets with the increase in data size, while they are not monotonic on *wiki-Vote* with the increase in data size.

Fig. 9 presents the comparison of different methods on KL-divergence. Figs. 9(a) and 9(b) depict the KL divergences of *DiffHetNet*, *DER*, *DE*, and *Random* under varying privacy budget ϵ from 0.6 to 1.0 while fixing the data size to be $0.4 \times |V|$ on *ca-GrQc*, and *wiki-Vote* datasets. In Fig. 9(a), the KL divergences of *DER* when $k = 1$ (static correlation parameter) are small on the *ca-GrQc* dataset. However, they increase with an increase of k. It is observed that *DiffHetNet*



(a) (b) (c)

Fig. 8. KL-Divergence by *DiffHetNet* method under varying ϵ in (a) and (b), and fixed $\epsilon = 1.0$ and varying data size in (c)



(a) ca-GrQc (b) wiki-Vote

Fig. 9. Comparison of *DiffHetNet*, *DER*, *DE*, and *Random* methods on KL-Divergence under varying ϵ in (a) and (b)

performs better than *DER* when the correlation parameter $k = 20$, and closer to *DE* when $k = 1$ on the *ca-GrQc* dataset. Fig. 9(b) depicts that *DiffHetNet* outperforms all the other methods on the *wiki-Vote* dataset. A significant difference of $\gtrapprox 0.7$ in KL divergences can be observed between *DiffHetNet* and *DER* ($k = 20$) under varying ϵ. The KL divergences of *Random* are greater than the other methods in all settings on both datasets.

5.2 Link prediction

We compute the accuracy of link prediction methods on anonymized data (i.e., by *DiffHetNet* method) compared to non-anonymized data. Figs. 10(a-d) depict the average accuracies of Adamic-Adar, common neighbors, and preferential attachment link prediction methods [42, 78], denoted by AALP, CNLP, and PALP, respectively, under varying privacy budget ϵ from 0.6 to 1.0 while fixing the data size to be $0.4 \times |V|$ on different datasets. Fig. 10(a) exhibits nondecreasing monotonicity in the accuracies of AALP and CNLP with the increase in ϵ on the *ca-GrQc* dataset. However, PALP accuracy decreases when $\epsilon = 0.8$. Fig. 10(b) exhibits non-decreasing monotonicity in the accuracies of AALP and CNLP with the increase in ϵ on the *wiki-Vote* dataset except $\epsilon = 0.8$ and $\epsilon = 0.7$, respectively. However, PALP accuracy decreases slightly when $\epsilon = 0.7$ and 0.8. Fig. 10(c) exhibits non-decreasing monotonicity in the accuracies of AALP,

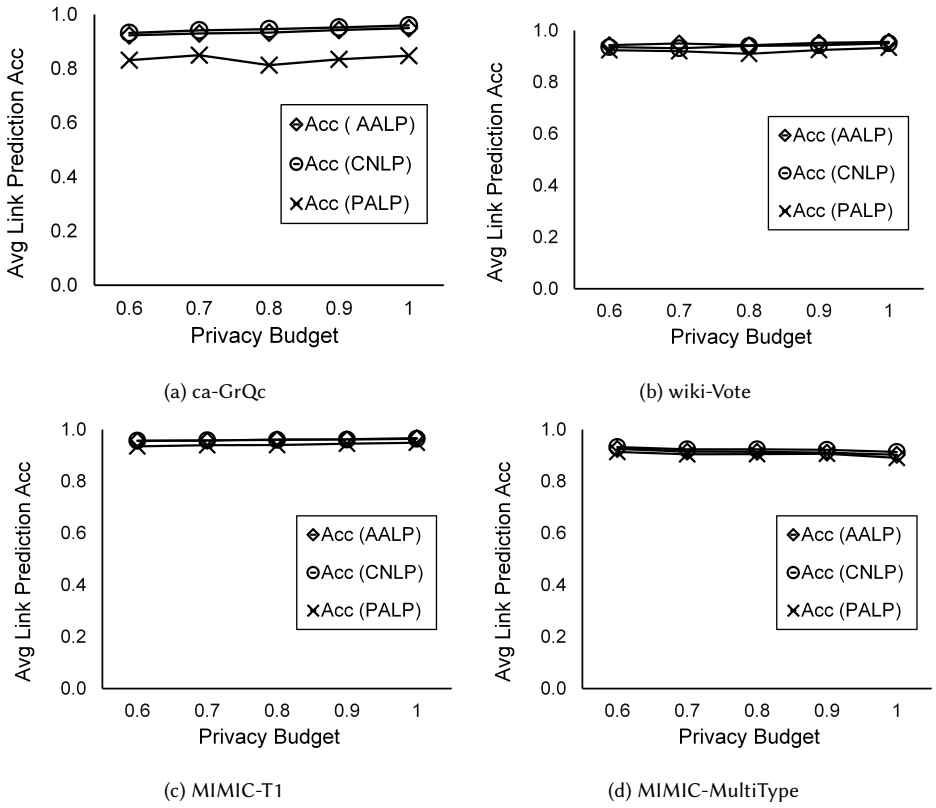
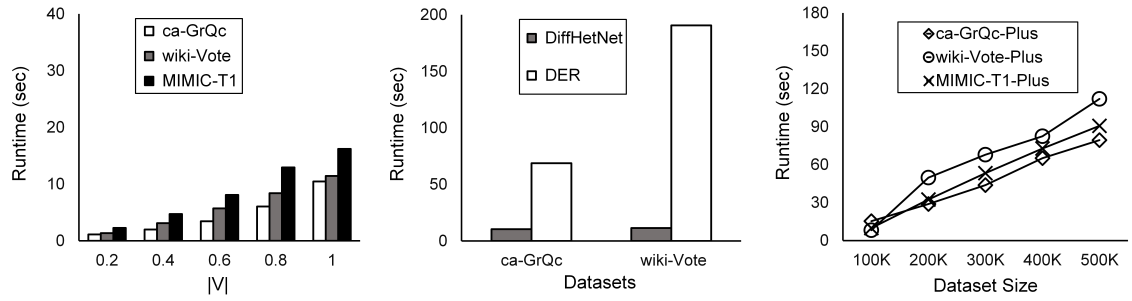


Fig. 10. Accuracy of Adamic-Adar, common neighbors, and preferential attachment link prediction methods on datasets (a) ca-GrQc, (b) wiki-Vote, (c) MIMIC-T1, and (d) MIMIC-MultiType under varying privacy budget ϵ.

CNLP, and PALP with the increase in ϵ on the *MIMIC-T1* dataset. Fig. 10(d) depicts average link prediction accuracy on the *MIMIC-MultiType* dataset. The accuracies of AALP, CNLP, and PALP generally decrease with the increase in ϵ. Overall, our method preserves high accuracy in predicting links.

5.3 Efficiency

Fig. 11(a) depicts the runtime of the *DiffHetNet* method under varying data size $|V|$ while fixing the privacy budget to be $\epsilon = 1.0$ on the *ca-GrQc*, *wiki-Vote*, and *MIMIC-T1* datasets. We observe that on all three datasets runtime grows with the increase in data size from 0.2 to 1.0. The runtime to produce anonymization results by *DiffHetNet* on *MIMIC-T1* with $1.0 \times |V|$ data size is approximately 16 s. Fig. 11(b) depicts the comparison of *DiffHetNet* and *DER* methods on runtime when $\epsilon = 1.0$ and data size is $1.0 \times |V|$ on both *ca-GrQc* and *wiki-Vote* datasets. *DiffHetNet* takes approximately 10 s and 11 s on the *ca-GrQc* and *wiki-Vote* datasets, respectively. The results show that our method is more efficient in running time over the *DER* method. In Fig. 11(c), we fix ϵ to 1.0 and evaluate the scalability of *DiffHetNet* using three datasets: *ca-GrQc-Plus*, *wiki-Vote-Plus*, and *MIMIC-T1-Plus*. The X-axis represents the number of records in thousands, ranging from 100, 000 to 500, 000 records. An edge going from one node to another node represents a single record. We consider no multiple edges (no duplicate records). For each 100K records, we add randomly-generated nodes and



(a) Varying data size with fixed $\epsilon = 1.0$ (b) Comparison of *DiffHetNet* and *DER* (c) Scalability

Fig. 11. Runtime comparison of *DiffHetNet*

edges for *ca-GrQc* and *wiki-Vote* to extend their original size. We name these two extended datasets *ca-GrQc-Plus* and *wiki-Vote-Plus*, respectively. As for *MIMIC-T1-Plus*, this dataset is the result of extracting 500K records from the *MIMIC* data table (MIMIC-III v1.4), which contains 651, 047 records representing ICD (International Classification of Diseases) diagnoses for patients. The runtime of each dataset increases nearly linearly with respect to the increase in the size of the dataset. This result suggests that our method is scalable to large network datasets.

6 DISCUSSION AND FUTURE WORK

Data is an integral part of almost every industry, including healthcare. Data often contains explicit identifying information associated with personal data such as name, social insurance number, birth date, address, phone number, marital status, health record, and so on. A data custodian who holds person-specific information must be responsible for managing the use, disclosure, and privacy protection of collected data. Privacy is a fundamental human right [3], and for this several privacy legislation and regulations such as *Personal Information Protection and Electronic Documents Act (PIPEDA)* by Canada, *Health Insurance Portability and Accountability Act (HIPAA)* by the United States, and *General Data Protection Regulation (GDPR*) by the European Union, across the globe have been imposed for protecting personal data. GDPR is the most robust and influential data privacy law recognized globally. This standard has become the measuring stick for other regulations. It follows the micro-management model of privacy regulations and applies to all the sectors. The fundamental difference between the US and EU privacy laws is that the US is more concerned with data integrity as a commercial asset, while the EU, with the GDPR, is more centric on individual rights before the interest of businesses. For example, if a company fails to protect the privacy of EU data subjects, it will be accountable by the law.

Many organizations believe that enforcing regulatory compliance, such as the Gramm-Leach-Bliley Act (GLBA), which protects the privacy and security of individually identifiable financial information, or simply employing common de-identification methods, such as HIPAA Safe Harbor method, which involves removing 18 types of identifiers from personal data, is sufficient for privacy protection. However, HIPAA de-identification works when the purpose is solely to share data among the contractually bound partners. Yet, de-identification alone is not sufficient for privacy protection when data is required to be shared outside of authorized partners because the released data can be linked with an external source of information to identify an individual [20, 32, 56, 67]. That's why we need a rigorous anonymization method. It is worth to note that the terms de-identification and anonymization has been used in literature interchangeably [13]. However, anonymization is a more stringent standard of de-identification. Our proposed network data anonymization

solution does not limit a custodian to share with contractually bound partners since the purpose is to release anonymized data openly.

According to GDPR[3], anonymous data is defined as "information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable". If an anonymous data satisfies these criteria, it will be considered safe, and there will be no penalties or monetary fines applied. Yet, the provision of anonymous data defined by GDPR is ambiguous [51]. The Article 29 Working Party, an independent European advisory body for data protection, stipulated three reidentification risks of data subjects, including singling out, linkability, and inference [1]. Their guidelines suggested that an anonymization technique would be robust if it protects against the stated reidentification risks. Differential privacy provides strong privacy guarantees against reidentification attempts [50] and has been used to develop efficient anonymization methods over the last decade in research. Recently, US Census Bureau [52] discovered that traditional techniques for data de-identification are not sufficient to defend against new threats. They found that an adversary can identify victims from the released statistics using external data sources. They are modernizing privacy protections using differential privacy to disguise personal information in the data. Similar to other applications [33], differential privacy has also been adopted to protect health-related data [16, 37, 48]. It provides a control parameter ϵ that allows a data custodian to calibrate how much noise to add so that the results strike a better balance between privacy and accuracy for data utility.

Our proposed method aims to publish consolidated health network data in a differentially private manner to preserve patients' privacy. We employ *MIMIC*[4] data to form a heterogeneous network and subsequently preserve patient privacy by releasing an anonymized version of the heterogeneous network. However, our approach does not rely on this dataset. Any network data with similar properties can be used for our purpose. In contrast with blockchain-based health information exchange (HIE) solutions, data is securely exchanged between peers by maintaining integrity [7, 11], but it is not anonymized for a safe release to unauthorized parties. To illustrate the applicability of our solution, we model it with 4 types of nodes and 3 types of edges, where the intuition is to protect a patient's sensitive information. It is still applicable when the network grows either in the number of nodes, edges, or both. We compare our method with *DER* [10], which applies to homogeneous undirected networks. The results presented in Fig. 11(b) show that our method is more efficient in running time over the *DER* despite the fact that we are considering the direction of edges, which requires more noise to be added to suppress sensitive links. In Fig. 11(c), we measure the increase of runtime with respect to the size of the input dataset.

The proposed method has some limitations that can be addressed in the future work. In our work, we dedicate $40\% \times \epsilon$ to explore vulnerable nodes in the first phase, $20\% \times \epsilon$ to generate noisy count for each vulnerable candidate in the second phase, and the remaining $40\% \times \epsilon$ to pick nodes that are less vulnerable (Refer to Sections 4.1.1, 4.1.2 for more details). The intention behind this distribution is to protect vulnerable nodes from identification attacks and to minimize information loss. To tackle this limitation, we intend to investigate the optimal allocation of the privacy budget ϵ, which is an open research question. This can be achieved under a less strict privacy model, such as Rényi Differential Privacy (RDP) [47]. RDP is a relaxation of the pure version of differential privacy. It shares several important properties with the standard definition of differential privacy. Our experiments have shown reasonably good results in information loss (Section 5.1) and link prediction (Section 5.2) with the more stringent version of different privacy. Consequently, we anticipate that relaxing the privacy requirement with RDP will only further reduce the information loss and improve the accuracy of link prediction. For another future research, we aim to investigate other vulnerabilities in the hybrid

[3]https://eur-lex.europa.eu/eli/reg/2016/679/oj
[4]Available at: https://mimic.physionet.org

setting (containing both directed and undirected links) of a heterogeneous network, which an adversary may exploit for privacy breaches.

7 CONCLUSION

In this article, we propose a practical solution to health information custodians (HICs) for publishing collected healthcare data to data recipients or researchers in a privacy-preserving manner. First, we model a complex de-identified healthcare dataset as a heterogeneous information network that consists of multi-type nodes and their multi-type edges. Then, we propose an edge-based differentially-private algorithm to protect the sensitive links of patients from inbound and outbound attacks in the heterogeneous health network. We evaluate the performance of our method in terms of information utility and efficiency on different types of real-life datasets that can be modeled as networks. The experimental results suggest that our method generally yields less information loss and is significantly more efficient in terms of runtime compared to existing network anonymization methods. It is also evident from the experiments that our method is scalable to large network datasets.

ACKNOWLEDGMENTS

This research is supported in part by the Discovery Grants (RGPIN-2018-03872) and CREATE Grants (CREATE-554764-2021) from the Natural Sciences and Engineering Research Council of Canada, Canada Research Chairs Program (950-230623), and the Research Incentive Funds (R18055 and R19044) from Zayed University, United Arab Emirates.

REFERENCES

[1] 2014. Article 29 Data Protection Working Party, Opinion 05/2014 on Anonymisation Techniques.
[2] 2019. Cost of a Data Breach Report. Ponemon Institute LLC. Sponsored by IBM Security.
[3] 2021. *Data Protection Laws of the World, Full Handbook.* https://www.dlapiperdataprotection.com
[4] Karim Abouelmehdi, Abderrahim Beni-Hessane, and Hayat Khaloufi. 2018. Big Healthcare Data: Preserving Security and Privacy. *Journal of Big Data* 5, 1 (2018), 1–18.
[5] Roland Assam, Marwan Hassani, Michael Brysch, and Thomas Seidl. 2014. (k, d)-Core Anonymity: Structural Anonymization of Massive Networks. In *Proceedings of the 26th International Conference on Scientific and Statistical Database Management.* ACM, Article 17, 12 pages.
[6] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. 2007. Wherefore Art Thou R3579x? Anonymized Social Networks, Hidden Patterns, and Structural Steganography. In *Proceedings of the 16th International Conference on World Wide Web.* ACM, 181–190.
[7] Sujit Biswas, Kashif Sharif, Fan Li, Zohaib Latif, Salil S. Kanhere, and Saraju P. Mohanty. 2020. Interoperability and Synchronization Management of Blockchain-Based Decentralized e-Health Systems. *IEEE Transactions on Engineering Management* 67, 4 (2020), 1363–1376.
[8] Christian Borgs, Jennifer Chayes, and Adam Smith. 2015. Private Graphon Estimation for Sparse Graphs. In *Advances in Neural Information Processing Systems 28.* 1369–1377.
[9] Jordi Casas-Roma, Julian Salas, Fragkiskos D. Malliaros, and Michalis Vazirgiannis. 2018. k-Degree Anonymity on Directed Networks. *Knowledge and Information Systems* 61, 3 (2018), 1743–1768.
[10] Rui Chen, Benjamin C. M. Fung, Philip S. Yu, and Bipin C. Desai. 2014. Correlated Network Data Publication via Differential Privacy. *The International Journal on Very Large Data Bases* 23, 4 (2014), 653–676.
[11] Zeng Chen, Weidong Xu, Bingtao Wang, and Hua Yu. 2021. A Blockchain-based Preserving and Sharing System for Medical Data Privacy. *Future Generation Computer Systems* 124 (2021), 338–350.
[12] James Cheng, Ada W. Fu, and Jia Liu. 2010. K-isomorphism: Privacy Preserving Network Publication Against Structural Attacks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data.* ACM, 459–470.
[13] Raphaël Chevrier, Vasiliki Foufi, Christophe Gaudet-Blavignac, Arnaud Robert, and Christian Lovis. 2019. Use and Understanding of Anonymization and De-Identification in the Biomedical Literature: Scoping Review. *Journal of Medical Internet Research* 21, 5 (2019).
[14] Wei-Yen Day, Ninghui Li, and Min Lyu. 2016. Publishing Graph Degree Distribution with Node Differential Privacy. In *Proceedings of the 2016 International Conference on Management of Data.* ACM, 123–138.
[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography.* Springer, 265–284.

[16] Amalie Dyda, Michael Purcell, Stephanie Curtis, Emma Field, Priyanka Pillai, Kieran Ricardo, Haotian Weng, Jessica C. Moore, Michael Hewett, Graham Williams, and Colleen L. Lau. 2021. Differential Privacy for Public Health Data: An Innovative Tool to Optimize Information Sharing while Protecting Data Confidentiality. *Patterns* 2, 12 (2021), 100366.

[17] R. A. Finkel and J. L. Bentley. 1974. Quad Trees a Data Structure for Retrieval on Composite Keys. *Acta Informatica* 4, 1 (1974), 1–9.

[18] Tao-Yang Fu, Wang-Chien Lee, and Zhen Lei. 2017. HIN2Vec: Explore Meta-Paths in Heterogeneous Information Networks for Representation Learning. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 1797–1806.

[19] Benjamin C. M. Fung, Yan'an Jin, Jiaming Li, and Junqiang Liu. 2015. *Recommendation and Search in Social Networks*. Springer, Chapter Anonymizing Social Network Data for Maximal Frequent-Sharing Pattern Mining, 77–100.

[20] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. 2010. Privacy-Preserving Data Publishing: A Survey of Recent Developments. *ACM Computing Survey* 42, 4, Article 14 (2010), 53 pages.

[21] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[22] Huan Gui, Jialu Liu, Fangbo Tao, Meng Jiang, Brandon Norick, Lance Kaplan, and Jiawei Han. 2017. Embedding Learning with Events in Heterogeneous Information Networks. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2428–2441.

[23] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. 2009. Accurate Estimation of the Degree Distribution of Private Networks. In *Proceedings of the 9th IEEE International Conference on Data Mining*. IEEE Computer Society, 169–178.

[24] Rebecca Herold and Kevin Beaver. 2014. *The Practical Guide to HIPAA Privacy and Security Compliance* (2nd ed.). Auerbach.

[25] Jing Hu, Jun Yan, Zhen-Qiang Wu, Hai Liu, and Yi-Hui Zhou. 2019. A Privacy-Preserving Approach in Friendly-Correlations of Graph Based on Edge-Differential Privacy. *Journal of Information Science and Engineering* 35, 4 (2019), 821–837.

[26] Ming Ji, Jiawei Han, and Marina Danilevsky. 2011. Ranking-Based Classification of Heterogeneous Information Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1298–1306.

[27] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, A Freely Accessible Critical Care Database. *Scientific Data* 3 (2016), 160035.

[28] Zach Jorgensen, Ting Yu, and Graham Cormode. 2016. Publishing Attributed Social Graphs with Formal Privacy Guarantees. In *Proceedings of the International Conference on Management of Data*. ACM, 107–122.

[29] Kevin Judd, Michael Small, and Thomas Stemler. 2013. What Exactly are the Properties of Scale-Free and Other Networks? *EPL (Europhysics Letters)* 103, 5 (2013), 58004.

[30] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2013. Analyzing Graphs with Node Differential Privacy. In *Proceedings of the 10th Theory of Cryptography Conference on Theory of Cryptography*. Springer, 457–476.

[31] Rashid Hussain Khokhar, Rui Chen, Benjamin C. M. Fung, and Siu Man Lui. 2014. Quantifying the Costs and Benefits of Privacy-Preserving Health Data Publishing. *Journal of Biomedical Informatics* 50 (2014), 107–121. Special Issue on Informatics Methods in Medical Privacy.

[32] Rashid H. Khokhar, Benjamin C. M. Fung, Farkhund Iqbal, Dima Alhadidi, and Jamal Bentahar. 2016. Privacy-Preserving Data Mashup Model for Trading Person-specific Information. *Electronic Commerce Research and Applications* 17 (2016), 19–37.

[33] Rashid H. Khokhar, Farkhund Iqbal, Benjamin C. M. Fung, and Jamal Bentahar. 2021. Enabling Secure Trustworthiness Assessment and Privacy Protection in Integrating Data for Trading Person-specific Information. *IEEE Transactions on Engineering Management* 68, 1 (2021), 149–169.

[34] Daniel Kifer and Johannes Gehrke. 2006. Injecting Utility into Anonymized Datasets. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*. ACM, 217–228.

[35] Daniel Kifer and Bing-Rong Lin. 2010. Towards an Axiomatization of Statistical Privacy and Utility. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. ACM, 147–158.

[36] Daniel Kifer and Ashwin Machanavajjhala. 2011. No Free Lunch in Data Privacy. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 193–204.

[37] Jong Wook Kim, Kennedy Edemacu, and Beakcheol Jang. 2019. MPPDS: Multilevel Privacy-Preserving Data Sharing in a Collaborative eHealth System. *IEEE Access* 7 (2019), 109910–109923.

[38] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. 2009. On Social Networks and Collaborative Recommendation. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 195–202.

[39] Saurabh Kumar and Pradeep Kumar. 2021. Privacy Preserving in Online Social Networks Using Fuzzy Rewiring. *IEEE Transactions on Engineering Management* (2021), 1–9.

[40] Andrea Landherr, Bettina Friedl, and Julia Heidemann. 2010. A Critical Review of Centrality Measures in Social Networks. *Business and Information Systems Engineering* 2, 6 (2010), 371–385.

[41] Jingquan Li. 2014. Data Protection in Healthcare Social Networks. *IEEE Software* 31, 1 (2014), 46–53.

[42] David Liben-Nowell and Jon Kleinberg. 2007. The Link-Prediction Problem for Social Networks. *Journal of the American Society for Information Science and Technology* 58, 7 (2007), 1019–1031.

[43] Zijie Lin, Liangliang Gao, Xuexian Hu, Yuxuan Zhang, and Wenfen Liu. 2019. Differentially Private Graph Clustering Algorithm Based on Structure Similarity. In *Proceedings of the 2019 the 9th International Conference on Communication and Network Security*. ACM, 63–68.

[44] Kun Liu and Evimaria Terzi. 2008. Towards Identity Anonymization on Graphs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 93–106.

[45] Frank McSherry. 2010. Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis. *Commun. ACM* 53, 9 (2010), 89–97.
[46] Frank McSherry and Kunal Talwar. 2007. Mechanism Design via Differential Privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 94–103.
[47] Ilya Mironov. 2017. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*. 263–275.
[48] Noman Mohammed, Xiaoqian Jiang, Rui Chen, Benjamin C. M. Fung, and Lucila Ohno-Machado. 2013. Privacy-Preserving Heterogeneous Health Data Sharing. *Journal of the American Medical Informatics Association* 20, 3 (2013), 462–469.
[49] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2007. Smooth Sensitivity and Sampling in Private Data Analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*. ACM, 75–84.
[50] Kobbi Nissim, Thomas Steinke, Alexandra Wood, Mark Bun, Marco Gaboardi, David R. O'Brien, and Salil Vadhan. 2017. Differential Privacy: A Primer for a Non-technical Audience. Privacy tools for sharing research data project at Harvard University.
[51] David Peloquin, Michael DiMaio, Barbara Bierer, and Mark Barnes. 2020. Disruptive and Avoidable: GDPR Challenges to Secondary Research Uses of Data. *European Journal of Human Genetics* 28, 6 (2020), 697–705.
[52] Samantha Petti and Abraham Flaxman. 2020. Differential Privacy in the 2020 US Census: What Will It Do? Quantifying the Accuracy/Privacy Tradeoff. *Gates Open Research* 3 (2020), 1722.
[53] Sofya Raskhodnikova and Adam Smith. 2016. Lipschitz Extensions for Node-Private Graph Statistics and the Generalized Exponential Mechanism. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. 495–504.
[54] Leonardo F.R. Ribeiro, Pedro H.P. Saverese, and Daniel R. Figueiredo. 2017. Struc2vec: Learning Node Representations from Structural Identity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 385–394.
[55] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y. Zhao. 2011. Sharing Graphs Using Differentially Private Graph Models. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. ACM, 81–98.
[56] Pierangela Samarati. 2001. Protecting Respondents' Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering* 13, 6 (2001), 1010–1027.
[57] Soumitra Sengupta, Neil S. Calman, and George Hripcsak. 2008. A Model for Expanded Public Health Reporting in the Context of HIPAA. *Journal of the American Medical Informatics Association* 15, 5 (2008), 569–574.
[58] Chuan Shi, Yitong Li, Jiawei Zhang, Yizhou Sun, and Philip S. Yu. 2017. A Survey of Heterogeneous Information Network Analysis. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2017), 17–37.
[59] Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. 2018. Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2190–2199.
[60] Shuang Song, Susan Little, Sanjay Mehta, Staal A. Vinterbo, and Kamalika Chaudhuri. 2018. Differentially Private Continual Release of Graph Statistics. *CoRR* abs/1809.02575 (2018).
[61] Yizhou Sun, Charu C. Aggarwal, and Jiawei Han. 2012. Relation Strength-Aware Clustering of Heterogeneous Information Networks with Incomplete Attributes. *Proceedings of the VLDB Endowment* 5, 5 (2012), 394–405.
[62] Yizhou Sun, Rick Barber, Manish Gupta, Charu C. Aggarwal, and Jiawei Han. 2011. Co-author Relationship Prediction in Heterogeneous Bibliographic Networks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*. 121–128.
[63] Yizhou Sun and Jiawei Han. 2012. *Mining Heterogeneous Information Networks: Principles and Methodologies*. Morgan & Claypool Publishers.
[64] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. Pathsim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proceedings of the VLDB Endowment* 4, 11 (2011), 992–1003.
[65] Yizhou Sun, Jie Tang, Jiawei Han, Manish Gupta, and Bo Zhao. 2010. Community Evolution Detection in Dynamic Heterogeneous Information Networks. In *Proceedings of the 8th Workshop on Mining and Learning with Graphs*. ACM, 137–146.
[66] Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-Based Clustering of Heterogeneous Information Networks with Star Network Schema. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 797–806.
[67] Latanya Sweeney. 2011. Patient Identifiability in Pharmaceutical Marketing Data. Data Privacy Lab Working Paper 1015.
[68] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. 2013. Exploiting Homophily Effect for Trust Prediction. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. ACM, 53–62.
[69] C. Lee Ventola. 2014. Social Media and Health Care Professionals: Benefits, Risks, and Best Practices. *Journal of Pharmacy and Therapeutics* 39, 7 (2014), 491–520.
[70] Paul Voigt and Axel Von Dem Bussche. 2017. *The EU General Data Protection Regulation (GDPR): A Practical Guide* (1st ed.). Springer.
[71] Yue Wang and Xintao Wu. 2013. Preserving Differential Privacy in Degree-Correlation Based Graph Generation. *Transactions on Data Privacy* 6, 2 (2013), 127–145.
[72] Cort J. Willmott and Kenji Matsuura. 2005. Advantages of the Mean Absolute Error (MAE) over the Root Mean Square Error (RMSE) in Assessing Average Model Performance. *Climate Research* 30, 1 (2005), 79–82.
[73] Xiaotong Wu, Wanchun Dou, and Qiang Ni. 2017. Game Theory Based Privacy Preserving Analysis in Correlated Data Publication. In *Proceedings of the Australasian Computer Science Week Multiconference*. ACM, Article 73, 10 pages.
[74] Bin Yang, Issei Sato, and Hiroshi Nakagawa. 2015. Bayesian Differential Privacy on Correlated Data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*. ACM, 747–762.

[75] Xiaobo Yin, Shunxiang Zhang, and Hui Xu. 2019. Node Attributed Query Access Algorithm Based on Improved Personalized Differential Privacy Protection in Social Network. *International Journal of Wireless Information Networks* 26, 3 (2019), 165–173.

[76] Aston Zhang, Xing Xie, Kevin Chen-chuan, Carl A. Gunter, Jiawei Han, and Xiaofeng Wang. 2014. Privacy Risk in Anonymized Heterogeneous Information Networks. In *Proceedings of the 17th International Conference on Extending Database Technology*. 595–606.

[77] Bin Zhou and Jian Pei. 2008. Preserving Privacy in Social Networks Against Neighborhood Attacks. In *Proceedings of the 24th IEEE International Conference on Data Engineering*. IEEE Computer Society, 506–515.

[78] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. 2009. Predicting Missing Links via Local Information. *The European Physical Journal B* 71 (2009), 623–630.

[79] Lei Zou, Lei Chen, and M. Tamer Özsu. 2009. K-Automorphism: A General Framework for Privacy Preserving Network Publication. *Proceedings of the VLDB Endowment* 2, 1 (2009), 946–957.

A TYPES OF INFORMATION NETWORKS

Generally, an information network is a representation that models the real world, focusing on objects and the interactions between objects [63]. These interactions in the network can be *symmetric* and *asymmetric*. In a symmetric interaction the relationship between objects can be in both directions, whereas asymmetric represents a one-way relationship. Typical examples of information networks are social networks, collaboration networks, health networks, and communication networks.

Definition A.1. (*Homogeneous information network*) *[63]. Given a network,* $G = (V, E)$ *with an entity type-mapping function* $\varphi : V \rightarrow \mathcal{E}$ *and a relation type-mapping function* $\psi : E \rightarrow \mathcal{R}$*, it is called a homogeneous information network if there exists only one type of entities and relations (i.e.,* $|\mathcal{E}| = |\mathcal{R}| = 1$*).* ■

Definition A.2. (*Heterogeneous information network*) *[63]. The information network is called a heterogeneous information network if the types of entities* $|\mathcal{E}| > 1$ *or the types of relations* $|\mathcal{R}| > 1$*.* ■

The *network schema* describes the meta structure of a heterogeneous information network, in which type constraints on the set of objects and relationships are specified. Many complex networks are modeled by heterogeneous networks to capture rich semantics. Traditional mining methods [38, 68] are designed for homogeneous networks, which cannot be directly applied to solve the problems of heterogeneity in many real-world networks. Various mining methods have been proposed to tackle the problem of heterogeneity for network analysis, such as ranking-based classification and clustering [26, 66], meta-path-based similarity search [64], relationship prediction and relation strength learning [61, 62], and community evolution [65]. Recently, advanced embedding methods for homogeneous networks [21, 54] and heterogeneous networks [18, 22, 59] have gained increasing attention for large-scale network analysis. On the one hand, these mining and embedding methods for heterogeneous networks serve different requirements of network analysis, but on the other hand, the privacy of an individual is at stake unless proper protection measures are deployed.

B NETWORK MEASURES

Below are some widely adopted graph metrics [40]. These measures contribute to the analysis of the structural properties of a network.

B.1 Betweenness centrality

The intuition of this measure is to determine the importance of a node in connecting other nodes. The betweenness of a node v_i in the network is computed by

$$CB(v_i) = \sum_{j \neq i \neq k \in V} \frac{\sigma_{v_j, v_k}(v_i)}{\sigma_{v_j, v_k}} \tag{5}$$

where $|V|$ is the number of nodes in the network, σ_{v_j,v_k} is the total number of shortest paths from node v_j to node v_k, and $\sigma_{v_j,v_k}(v_i)$ is the number of those paths that pass through v_i. To normalize the betweenness centrality, divide the metric in Eq. (5) by $(|V|-1)(|V|-2)$ for directed graphs and by $(|V|-1)(|V|-2)/2$ for undirected graphs.

B.2 Degree centrality

A node is in the "central" if it has many direct neighbors. For a directed network, *indegree* is the number of incoming links representing the popularity of a node, whereas *outdegree* is the number of outgoing links representing the sociability of a node. In an undirected network, the *degree* of a node is simply the number of directly connected neighbors ignoring edge directions. The normalized degree centrality CD for a node v_i is computed by

$$CD(v_i) = \frac{d(v_i)}{|V|-1} \tag{6}$$

where $d(v_i)$ is the degree of node v_i.

B.3 Closeness centrality

In this measure, a node is in the "central" if it is close to many other nodes, and of which the closeness can be measured by the shortest paths for reaching those nodes. The normalized closeness centrality CC for a node v_i is computed by

$$CC(v_i) = \frac{|V|-1}{\sum_{j\neq i}^{|V|} d(v_j,v_i)} \tag{7}$$

where $d(v_j,v_i)$ is the shortest-path distance between v_j and v_i. If the direction between nodes v_i and v_j is not specified, then the total number of nodes $|V|$ is used in Eq. (7) instead of the path length.

B.4 Harmonic centrality

It is a variant of closeness centrality that deals with the scenario of unconnected networks. It is the sum of the reciprocal of the shortest path distances from all other nodes to a given node. The normalized harmonic centrality CH for a node v_i is computed by

$$CH(v_i) = \frac{1}{(|V|-1)} \times \sum_{j\neq i}^{|V|} \frac{1}{d(v_j,v_i)} \tag{8}$$

If there is no path from v_j to v_i, then $1/d(v_j,v_i)$ becomes 0.

C DIFFERENTIAL PRIVACY FOR NETWORK DATA

Differential privacy [15] is a widely known privacy model with an assumption that all the records in the database are independent of each other. A line of research [28, 36, 73, 74] indicates that differential privacy may not guarantee privacy against adversaries with arbitrary background knowledge when data records are correlated. To tackle this issue, a notion of correlation parameter k is proposed by [10] that provides a similar differential privacy guarantee when releasing network data. The intuition of their solution is to add extra Laplace noise in the anonymization process to compensate for the effect of correlation.

Definition C.1. (*ϵ-differential privacy under correlation*) *[10]. A sanitization mechanism $\mathcal{M}$ provides* ϵ-differential privacy *if for any two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ with a correlation parameter k that differs on* at most *one record (i.e., symmetric*

difference $|\mathcal{D}_1 \triangle \mathcal{D}_2| \leq 1$), and for any possible sanitized dataset $\hat{\mathcal{D}}$, we have

$$Pr[\mathcal{M}(\mathcal{D}_1) = \hat{\mathcal{D}}] \leq e^{\frac{\epsilon}{k}} \times Pr[\mathcal{M}(\mathcal{D}_2) = \hat{\mathcal{D}}],$$

where the probability is taken over the randomness of $\mathcal{M}$. ■

In the literature, node-differential privacy [8, 14, 30] and edge-differential privacy [10, 25, 71] are the most prevalent formulations for anonymizing network data. In node-DP, two graphs G and G' are neighboring graphs if they differ by *at most* one node and, by extension, all its edges. Whereas in edge-DP, two graphs G and G' are neighboring graphs if they differ by *at most* one edge or an *isolated* node (a node that has no edges). The following definitions define two types of neighboring graphs under node- and edge-differential privacy, respectively.

Definition C.2. (*Neighborhood under node-differential privacy*) *[23]. Given graph $G = (V, E)$, where V is a set of nodes and E is a set of edges, two graphs G and G' are neighbors if $|V \oplus V'| = 1$ and $E \oplus E' = \{(u, v)|u \in (V \oplus V') \text{ or } v \in (V \oplus V')\}$.* ■

Definition C.3. (*Neighborhood under edge-differential privacy*) *[23]. Given graph $G = (V, E)$, where V is a set of nodes and E is a set of edges, two graphs G and G' are neighbors if $|V \oplus V'| + |E \oplus E'| = 1$.* ■

The *Laplace mechanism* [15] and *exponential mechanism* [46] are the two most common mechanisms for achieving ϵ-differential privacy. These mechanisms depend on the privacy parameter ϵ and the sensitivity [15] of a function that maps the input database to real values. The *sensitivity* of the function f is defined as follows:

*Definition C.4 (*Sensitivity*). For any function $f : G \rightarrow \mathbb{R}^d$, the sensitivity of f is*

$$\Delta f = \max_{G,G'} ||f(G) - f(G')||_1 \tag{9}$$

for all G, G' differing at most *by one edge or node (including all its adjacent edges).* ■

Laplace mechanism was introduced by Dwork et al. [15]. It is appropriate when the output of function f is a real value, and f should return a noisy answer to preserve privacy. The noise is calibrated based on the privacy parameter ϵ and the sensitivity of the utility function Δf. Formally, the Laplace mechanism takes as inputs a network dataset G, the privacy parameter ϵ, and a function f and outputs $f(\hat{G}) = f(G) + \mathsf{Lap}(\lambda)$, where $\mathsf{Lap}(\lambda)$ is a noise drawn from the Laplace distribution with probability density function $\Pr(x|\lambda) = \frac{1}{2\lambda}\exp(-|x|/\lambda)$, where $\lambda = \frac{\Delta f}{\epsilon}$. The variance of this distribution is $2\lambda^2$, and the mean is 0.

Theorem C.5. *For any function $f : G \rightarrow \mathbb{R}^d$, the algorithm $\mathcal{M}$ that adds independently generated noise with distribution $\mathsf{Lap}(\Delta f/\epsilon)$ to each of the d outputs satisfies ϵ-differential privacy.*

Exponential mechanism was proposed by McSherry and Talwar [46]. It is appropriate when it is desirable to choose the best response, because adding noise directly to the count can destroy its value. Given an arbitrary range $\mathcal{T}$, the exponential mechanism is defined with respect to a utility function $\mathsf{u} : (G \times \mathcal{T}) \rightarrow \mathbb{R}$ that assigns a real-valued score to every output $t \in \mathcal{T}$, where a higher score means better utility. The exponential mechanism induces a probability distribution over the range $\mathcal{T}$ and then samples an output t.

Theorem C.6. *Given a utility function $\mathsf{u} : (G \times \mathcal{T}) \rightarrow \mathbb{R}$ with sensitivity $\Delta\mathsf{u} = \max_{\forall t,G,G'} |\mathsf{u}(G, t) - \mathsf{u}(G', t)|$, an algorithm $\mathcal{M}$ that chooses an output t with probability proportional to $\exp(\frac{\epsilon \mathsf{u}(G,t)}{2\Delta\mathsf{u}})$ satisfies ϵ-differential privacy.*

Sequential composition and *parallel composition* are the two important composition properties of differential privacy [45]. The first property stipulates that if a sequence of differentially private computations take place in isolation on the same input data, then the entire sequence gives the accumulated privacy guarantee. The second property stipulates that if differentially private computations take place on each chunk separately over the split dataset, where chunks are disjoint, then the privacy cost does not accumulate, but it depends only on the worst guarantee of all computations.

Theorem C.7 (*Sequential composition* [45]). *Let each $\mathcal{M}_i$ provide ϵ_i-differential privacy. A sequence of $\mathcal{M}_i(G)$ over the network G provides $(\sum_i \epsilon_i)$-differential privacy.*

Theorem C.8 (*Parallel composition* [45]). *Let each $\mathcal{M}_i$ provide ϵ-differential privacy. A sequence of $\mathcal{M}_i(G_i)$ over a set of disjoint networks G_i provides ϵ-differential privacy.*

D INFORMATION LOSS MEASURES

Below are some generic measures to quantify the information loss when releasing anonymized network G'. The general goal is to minimize information loss and to improve data utility.

D.1 Mean absolute error

This measures the absolute error by comparing the degree centrality score of a node v_i in the anonymized network G' with respect to the original network G. The mean absolute error (*MAE*) [72] for all the nodes in the network is computed as follows:

$$MAE(G, G') = \frac{1}{|V|} \times \sum_{i=1}^{|V|} |CD(G', v_i) - CD(G, v_i)| \tag{10}$$

D.2 Average relative error

This measures the relative error of a node v_i in the anonymized network G' with respect to the original network G [31]. The average relative error (*ARE*) for all the nodes in the network is computed as follows:

$$ARE(G, G') = \frac{1}{|V|} \times \sum_{i=1}^{|V|} \frac{|CD(G', v_i) - CD(G, v_i)|}{CD(G, v_i)} \tag{11}$$

D.3 Kullback–Leibler divergence

Degree distribution captures the important structural properties of a network. This one computes the frequency count of the occurrence of each degree to differentiate the number of connections between nodes in a network. For a directed network, the frequency counts for the indegree and outdegree of a node are computed based on the type of degree direction. Given the degree distributions of the original network and the anonymized network, $DD(G)$ and $DD(G')$, we measure their difference by *Kullback–Leibler divergence* [34] as follows:

$$KLDiv(DD(G)||DD(G')) = \sum_{i=0}^{|V|-1} DD(G)[i] \cdot \ln\left(\frac{DD(G)[i]}{DD(G')[i]}\right) \tag{12}$$