Publishing Private Data: A Review

Emran altamimi
Department of Computer Science and Engineering
Qatar University, Qatar
ea1510662@qu.edu.qa

Abstract:

This review discusses and analyzes various models and methods employed to counter privacy threats in data publishing, including identity, membership, and attribute disclosure. Several models such as k-anonymity, l-diversity, and d-presence among others, have been discussed, each demonstrating the delicate balance between preserving data utility and ensuring privacy. Four distinct anonymization approaches for transaction databases, time-series data, multi-attribute data, and set-valued data are also scrutinized, highlighting the varied methodologies and algorithms employed to maintain this balance.

The first algorithm discussed is specifically designed for anonymizing transaction datasets for publication, which are characterized by high diversity and unstructured information. The second algorithm addresses the unique challenges of anonymizing time-series data, preserving inherent temporal correlations during the anonymization process. The third algorithm presents a new formulation to solve k-anonymity in an optimal way using dynamic programming and space mapping. Lastly, the fourth algorithm utilizes a recursive greedy approach to anonymize set-valued data, such as query logs, providing k-anonymity while preserving as much information as possible. This research highlights the need for distinct privacy preservation techniques that are adaptable to the characteristics and requirements of various data types.

Keywords:

Data Privacy, Anonymization, Identity Disclosure, Membership Disclosure, Attribute Disclosure

TABLE OF CONTENTS

1 INTRODUCTION 3
2 BACKGROUND AND RELATED WORK 4
3 RELATED WORK 5
3.1 Models against identity disclosure 6
3.2 Models against membership disclosure 7
3.3 Models against attribute disclosure 8
4 ALGORITHMS 9
4.1 Anonymizing Transaction Databases for Publication 9
4.2 Supporting Pattern-Preserving Anonymization for Time-Series Data 10
4.3 Fast Data Anonymization with Low Information Loss 10
4.4 Anonymization of Set-Valued Data via Top-Down, Local Generalization 11
5 CONCLUSION AND FUTURE WORK 12
6 REFERENCES 13

1 INTRODUCTION

The increasing availability of digital information has led to a new era of data accessibility, providing new insights and capabilities across various fields. The abundance of data frequently encompasses confidential personal information that necessitates safeguarding against any form of exploitation. The challenge of preserving privacy in data publishing necessitates a careful equilibrium between maintaining data utility for research or commercial objectives, and safeguarding the confidentiality of the individuals represented in the data. The use of conventional anonymization models, such as k-anonymity [1] or l-diversity [2], has been widespread in safeguarding structured relational data against re-identification attacks. However, their efficacy is limited when dealing with different data, for example, time-series [3] or transactional data [4]. Such efficacy loss is attributed to the different charecteristics exihibited by different datasets. For example, graph data must be anonymized in a very different way than transactional datasets [5]. The observed pattern loss has a notable impact on the feasibility of performing intricate data queries, thereby constraining its practicality.

The prevalence of sensitive data in the contemporary digital landscape highlights the importance of this issue. Time-series data obtained from sensor networks, RFIDs, and wireless positioning devices, as well as transaction data generated from web searches, purchase records, click streams, and emails, are playing an increasingly important role in a wide range of applications. The applications of data mining encompass a diverse range of areas, including financial analysis, social community tracking, association rule mining, user behavior prediction, recommender systems, and personalized web searches. Nevertheless, there is a significant possibility of misuse. Instances of privacy violations, wherein individuals were able to be identified through their data, have highlighted the urgent requirement for efficient anonymization techniques. The task at hand involves maintaining the semantic interpretability of published data, which refers to its comprehensibility and usefulness for the intended recipients, while simultaneously safeguarding the privacy of individuals. Therefore, this situation necessitates creative approaches to enable the secure dissemination of confidential data while preserving both the data's usefulness and the privacy of the individuals concerned.

This study examines privacy models and algorithms applicable to the non-interactive data sharing scenario, where the objective is to release an anonymized dataset. This methodology provides uninterrupted access to data, eliminates infrastructure expenses, and facilitates the testing of hypotheses. It is imperative for data owners to establish privacy and utility prerequisites prior to data sharing. It is important to note that once the data is disseminated, the owners relinquish control over it, rendering it vulnerable to potential, unanticipated security breaches.

Alternatively, in the interactive data sharing scenario, data is stored in a secure repository that can be queried by users, who receive protected responses rather than the entire dataset [6]. This situation enables data owners to oversee the usage of their data, implement access control regulations, and modify the safeguarding mechanism in response to emerging risks, providing continuous and up-to-date data protection. The system additionally enables the implementation of strong semantic privacy models and enables a suitable degree of privacy for responses. Nevertheless, the system may encounter difficulties in accommodating intricate queries, usually imposes a cap on the quantity of queries it can handle, and poses challenges for tasks that necessitate the use of individual records, such as visualization.

In the end, there exist both benefits and drawbacks to each scenario, and it is entrusted upon data publishers to carefully determine the most appropriate approach in accordance with their specific requirements. Certain algorithms that are intended for interactive scenarios maintain confidentiality by introducing randomization to every query response, allowing for precise, broad-level metrics to be computed. The statistical disclosure control community has extensively investigated methods for releasing statistics while preserving privacy. However, these techniques do not provide a rigorous model for ensuring privacy preservation.

2 BACKGROUND AND RELATED WORK

Privacy Preserving Data Publishing (PPDP) is a set of techniques, approaches, and systems that enable the sharing of sensitive data with analysts and researchers while safeguarding the privacy of individuals. A significant body of research has been dedicated to the study of Privacy-Preserving Data Publishing (PPDP), with a particular emphasis on safeguarding user privacy across different stages of data analytics. Although data is often anonymized prior to sharing, it is still possible for adversaries to deduce user identities or sensitive information through the use of auxiliary data obtained from external sources. Numerous studies suggest the importance of safeguarding privacy throughout all stages of data handling, encompassing collection, storage, processing, release, and archival.

The PPDP conceptual process entails several stages, including data collection from individuals, identification of actors involved in the anonymization scenario, application of anonymization techniques, publication of anonymous data for analysis, and consideration of potential privacy breaches during data analytics. In the context of PPDP, it is common to identify five distinct categories of actors who are typically involved in the process. In certain instances, a single actor may undertake multiple roles. Data holders may choose to anonymize data prior to sharing it with analysts.

The PPDP process typically comprises five fundamental phases, namely: individual data collection, data storage and preparation for anonymization, user data anonymization, release or publication of anonymous data, and analysis of the published data to extract embedded knowledge.

During the initial stage of PPDP, data is gathered from individual users. The proliferation of technology has led to a significant rise in data production from a variety of sources, including sensor networks, social media platforms, healthcare software, online banking, third-party applications linked with social media, and other digital and non-digital entities. The data can be obtained either through direct communication with individuals or through their smart devices.

Instances of data collection encompass the acquisition of users' data. Data can be obtained from users through various means such as questionnaires, interviews, or online platforms that are initiated by service providers. Social networking platforms typically gather fundamental user information upon account creation, and in some cases, may acquire supplementary data without explicit authorization.

Once the data has been gathered, it is then processed and made ready for the purpose of anonymization. Comprehending data necessitates an examination of the various data types and their corresponding representations. Prior to anonymization, it is necessary to prepare the collected data for potential issues such as incorrect values, outliers, missing attribute values, or incomplete records. The procedure involves the elimination of anomalous data points and incomplete records, as well as the appropriate formatting of the data to meet the necessary criteria for subsequent analytical steps. The anonymization algorithm can be fed with the comprehensive and cleaned data of each user.

Data anonymization is a technique used to safeguard user privacy when publishing data. It involves modifying original values in tabular data to make them less specific and altering the graph structure in graph data. The process can be customized to meet the privacy requirements of the data owner and the objectives of data publishing. The process of anonymization involves four primary steps [7]:

1. To optimize computing power, any information that can directly or uniquely identify an individual, such as their name, social security number, email address, or phone number, is removed from the original data.
2. The selection of an anonymization technique is dependent on the way data is represented (in graphs or tables) and the specific goals of data publication. Anonymization techniques can be classified into two categories: structural anonymization for graph data and relational anonymization for tabular data.
3. The process of selecting an anonymization operation involves choosing the most suitable method to alter the original data values or graph structure. The choice of selection is determined by various factors such as the type of data, the method of anonymization, and the goals of privacy and utility.
4. Constraints can be enforced by data owners during the process of anonymizing data to ensure that they are adhered to. These constraints may relate to privacy and utility thresholds, the quantity of users in an equivalence class, the distribution of private values among users in a class or cluster, or the number of connections between users in a graph. Data owners can suggest constraints or derive them from data

statistics, taking into account the capabilities of potential adversaries and the sensitive nature of the information contained in the data.

Once the process of anonymization is complete, the resulting data (referred to as G 0 or T 0) is made ready for public release and subsequent analysis. Analysts, researchers, data miners, analytics firms, and third-party applications are able to access this data. Before releasing anonymous data, data owners conduct checks to ensure that the privacy and utility levels are maintained. Certain data owners choose to initially publish only certain portions of the anonymous data, and subsequently make the entire data set available. The data can either be shared with specific institutes that are relevant or published on the Internet for broader access. Once data is released, the owners no longer have control over how the data is used or distributed.

Once anonymous data is published, it can be collected by recipients for analysis. This may involve conducting research, such as examining disease factors in hospital data or evaluating loan repayment rates in bank data. Companies frequently mine social network (SN) data for scientific and business purposes. Data mining algorithms can be used to analyze published data and extract actionable insights. Sharing data can offer various advantages, including better decision-making, policy improvement, trend analysis, forecasting, and innovation. Although data publication has its benefits, it also presents a potential threat to user privacy. Adversaries can access the published data and leverage additional information from external sources to re-identify individuals. Privacy breaches can have adverse effects on individuals and result in a loss of trust from users towards data owners. The main objective is to create novel anonymization techniques that safeguard user privacy while having minimal impact on data utility and quality.

3 RELATED WORK

Privacy threats target three categories: direct identifiers, quasi-identifiers, and sensitive attributes. Direct identifiers have the ability to specifically identify individuals. The attributes encompassed are name, mailing address, phone number, social security number, other national identification numbers, and email address. Quasi-identifiers are a set of attributes that have the potential to reveal the identity of an individual. Examples of data elements may include demographic information such as gender, date of birth, and zip code, as well as diagnosis codes. Sensitive attributes are personal information that individuals wish to keep confidential, for example, diagnosis codes (e.g., psychiatric disorders, HIV, cancer) or genomic data. The combination of these attribute types results in diverse categories of privacy risks.

Identity disclosure or re-identification [8] is a significant threat when releasing medical data. It is considered one of the most severe threats. The threat arises when an assailant can establish a connection between a user and their record in a publicly available dataset, despite the absence of direct identifiers. An individual can be re-identified by an attacker using only quasi-identifiers such as birth date and zip code.

The second type pertains to membership disclosure [9], wherein an attacker can deduce with a high level of certainty that an individual's record is present in the published data. This presents a substantial privacy risk by potentially disclosing confidential information such as a patient's medical condition. Membership disclosure can occur despite the safety of data from identity disclosure.

The third threat pertains to the disclosure of sensitive or attribute information [2]. This refers to the situation where an individual's sensitive attributes are exposed, potentially revealing confidential information such as genetic data or medical expenses. The significance of efficient anonymization and data privacy mechanisms is underscored by these privacy breaches.

Privacy models are computational strategies developed to prevent privacy threats when sharing datasets. They serve two key purposes: firstly, they identify potential privacy threats that may occur when data is shared or published; secondly, they propose methods to protect against these threats [10]. These models can be categorized based on the privacy threats they guard against. In this section, we delve into well-established privacy models that are designed to combat identity disclosure and attribute disclosure, both of which are prominent threats in medical data publishing. We further break these down into models for demographics and diagnosis codes, providing an extensive overview of the diverse range of privacy models and their specific applications. It's important to note that the purpose of these models is to strike a balance between ensuring data privacy and preserving the utility of the data for research or other purposes.

Privacy models also can be categorized on another independent axis, which is the type of data they are meant to protect. Some privacy models for focus on protecting information such as age, gender, zip code, race, and other demographic data. Demographic data is often referred to as quasi-identifiers, as, in combination, these data points could potentially lead to the identification of an individual. For example, the combination of age, zip code, and gender could be unique enough to identify a specific person in a data set. Hence, the privacy models for demographics are designed to prevent the possibility of combining these quasi-identifiers to re-identify individuals.

On the other hand, there are models that are concerned with protecting sensitive information, for example, diagnosis codes. Diagnosis codes can reveal sensitive health information about individuals, such as specific diseases or conditions they may have been diagnosed with. Given the sensitive nature of this information, privacy models in this category focus on ensuring that such data cannot be linked to an individual, thus preventing attribute disclosure or sensitive information disclosure.

In summary, while both categories aim to protect individual identities, the key difference lies in the type of data they focus on: demographic models focus on general demographic information (quasi-identifiers), while sensitive attributes models aim to secure more sensitive information present in the data.

The privacy models are classified by the type of attack they aim to protect against, therefore, the following subsections review the privacy models based on the attack they protect against. It's worth noting that privacy models also include the algorithms that enforce them. The subsequent section will also explain the common transformation (for example, removing some data points or generalize several attributes' values into one attribute value), utility objectives (for example, minimizing information loss) and heuristic strategies used by the algorithms to enforce these privacy models.

That being said, such grouping of strategies is in no means exclusive to the privacy models. All strategies can be applied in any of these privacy models as they are merely tools to enforce the privacy models. However, finding that a particular strategy is more popular in particular privacy models may indicate that they are more advantageous in enforcing the privacy model.

3.1 Models against identity disclosure

Privacy models, such as k-anonymity [1] and k-map [11], aim to reduce the likelihood of identity disclosure by safeguarding demographic information. The k-anonymity model requires that each record in a dataset has identical values in the quasi-identifier attributes (QIDs) set with at least k-1 other records. QIDs are seemingly harmless attributes that, when aggregated, can establish connections between external data sources and the dataset that has been made public. The k-anonymity model minimizes the risk of identity disclosure by limiting the probability of connecting an individual to their record based on QIDs to 1/k. The parameter kregulates the level of privacy safeguarding. The k-map model, proposed for demographics, incorporates linking based on larger datasets or population tables that underlie the published dataset. This model enables the publication of more comprehensive user information, thereby maintaining data utility, but offers less protection compared to k-anonymity. The assumption underlying this statement is that attackers lack knowledge of a record's inclusion in the published dataset, and that data publishers possess the population table. Other privacy models, such as (1, k)-anonymity, (k, 1)-anonymity, and (k, k)-anonymity [12], are variations of k-anonymity that assume different attacker capabilities and generally offer greater data utility but weaker privacy protection.

As mentioned earlier, its possible to also focus on protecting the sensitive attributes and achieve the same goal. While the idea is similar, there are some assumptions made that make subtle but significant differences. For example, the complete k-anonymity model postulates that identity disclosure could result from any combination of sensitive attributes and requires that at least k records in the shared data have the same sensitive attributes. This model, however, may unnecessarily limit the usefulness of the data because it is very difficult for attackers to be aware of every sensitive attribute in the data.

K^m-anonymity [13], a more flexible model, enables the control of the greatest possible number of sensitive attributes that an attacker might be aware of. It is helpful when data publishers are unable or unwilling to specify specific sets of sensitive attributes that could result in identity disclosure attacks because it requires each combination of m sensitive attributes to appear in at least k records of the released dataset. A more recent model of anonymity called privacy-constrained anonymity [14] works with sets of sensitive attributes that may be known to an attacker as the privacy constraints. It does so by requiring that the set of sensitive attributes in each privacy constraint appear in the dataset at least k times (or not at all), which reduces the probability of performing identity disclosure to no more than 1/k.

These models are built under the presumption that the attackers are aware of whether a user's record is present in the dataset that has been made public. Potentially, lowering this supposition could provide greater utility at the expense of privacy. Sets of sensitive attributes that could be used in identity disclosure attacks are specifically protected by privacy-constrained anonymity according to the privacy policy. This method fixes a significant flaw in both complete k-anonymity and K^m -anonymity, which have a tendency to overprotect data. As a result, data utility is better preserved.

The two most common algorithm strategies used in this group are data partitioning and clustering-based strategies. The strategies here also need a heuristic, which is commonly minimizing information loss. Both are similar in that they iteratively form groups however they do it in different ways. Namely, Data partitioning works by dividing the dataset into groups or "partitions" based on the values in a specific attribute or set of attributes, typically quasi-identifiers. The aim is to create partitions that are similar with respect to the chosen attribute(s), thereby anonymizing the data. An example of data partitioning could be separating a medical database into two groups based on age: one group for patients under 50, and another for patients 50 and over. This process is repeated iteratively for different attributes, creating a multi-dimensional partitioning. The selection of attributes and the number of partitions can significantly influence the utility of the anonymized data.

The advantage of data partitioning is its efficiency: it requires O(n * log(n)) time complexity where n is the number of records in the dataset, making it significantly faster than clustering-based strategies. However, it may lead to higher utility loss compared to clustering-based methods, and its performance can be negatively impacted if the dataset is skewed.

While Clustering-based strategies, work by merging groups of records based on the values of all quasi-identifier attributes together. Unlike partitioning strategies that split records based on a single attribute, clustering takes into account the multi-dimensional similarity of records. The algorithm iteratively merges the most similar groups of records until a certain condition is met, often a predefined number of clusters or a certain level of similarity within clusters.

For example, consider a medical database with attributes such as age, gender, and location. A clustering algorithm could merge all records of individuals aged 20-30, male, living in a specific city into one cluster, and individuals aged 50-60, female, living in another city into another cluster, and so on.

While clustering-based methods can result in lower utility loss compared to partitioning methods, and are less sensitive to the choice of attributes or skew in the dataset, they require significantly more computational resources. Specifically, they have a time complexity of O(n^2), making them less practical for large datasets.

3.2 MODELS AGAINST MEMBERSHIP DISCLOSURE

In order to prevent membership disclosure, privacy models work to restrict an attacker's ability to infer that a person's record is present in a particular dataset. The first of these models, d-presence [9,15], limits the likelihood that an individual's record is included in a dataset given a version of the dataset intended for publication and a public population table, which is assumed to contain all information that is generally known to exist, including direct and quasi-identifiers for every member of the population. This model offers a range of allowable probabilities (dmin; dmax) for the inference of a person's record as a component of the dataset.

The d-presence model's applicability is constrained by the requirement that data owners possess thorough knowledge of the population. A modified model [16], called c-confident d-presence, was put forth to get around this problem. This model makes use of a set of population distribution functions rather than a complete population table. In other words, it is known that a person has a certain probability of being associated with one or more values, rather than one or more attributes. The model protects against membership disclosure by ensuring that a record is d-present in relation to the population with an owner-specified probability (c).

Most of these models employ a top-down lattice search strategy. The way they work is by generalizing quasi-identifier attributes in a systematic way. To explain it in detail, the strategy starts with a taxonomy for each attribute that represents the various levels of generalization for that attribute. Each node in the taxonomy is a potential generalization of the data at a lower level. For instance, consider the attribute "Age". A taxonomy for age might look like this:

1. At the most specific level (leaf nodes), you have the exact ages: e.g., 25, 26, 27, 28, etc.
2. The next level up might generalize the ages to ranges: 20-29, 30-39, etc.
3. At the root level, all ages could be generalized to "Any Age".

Similarly, taxonomies can be built for other quasi-identifiers.

Now, these taxonomies are combined to create a lattice structure. Each node in the lattice represents a combination of generalizations for all the quasi-identifiers. For instance, a node might represent the combination {Age: 20-29, Gender: Any}.

The top-down lattice search begins at the most generalized node (i.e., the top of the lattice) where all attributes are generalized to their most unspecific form. It then descends the lattice, exploring less generalized nodes, hence the term "top-down".

In every stage of the search, the algorithm verifies if the current generalization level meets the required privacy model, in other words, if the data is sufficiently anonymized. If the level of generalization doesn't satisfy the privacy model, the algorithm prunes the search; it doesn't need to explore the descendants of the current node further as they will likely not meet the privacy criteria. This approach makes the search efficient as it avoids the need to traverse the entire lattice.

Consider the quasi-identifiers Age and Gender as an example. The lattice's top represents the most generalized case, {Age: Any Age, Gender: Any}. This level of generalization typically lacks utility even though it guarantees privacy. We proceed to less generalized nodes such as {Age: Any Age, Gender: Male}, {Age: Any Age, Gender: Female}, {Age: 20-29, Gender: Any}, etc. At each step, we check whether the privacy criteria are satisfied. This process continues until the algorithm identifies a level of generalization that optimally balances the data utility while ensuring privacy or until all options have been exhausted.

3.3 MODELS AGAINST ATTRIBUTE DISCLOSURE

Demographic privacy models in these privacy models are designed to prevent the disclosure of sensitive attributes, similar to what’s discussed previously. L-diversity [2] is a popular model that requires each anonymized group in a dataset to have at least 'l' well-represented sensitive attributes values. Although k-anonymity is not a prerequisite for l-diversity, this strategy can be applied even in k-anonymous groups. Each anonymized group must contain at least 'l' distinct sensitive attributes values according to a popular implementation known as distinct l-diversity.

The probability distribution of sensitive attributes values within each anonymized group is constrained to mirror the distribution in the entire dataset by models like t-closeness [17], which takes this restriction a step further. This strategy stops intruders from discovering specific details about a person's sensitive attributes value that they couldn't deduce from the overall dataset. For instance, if the sensitive attribute value "Disease" for 75 percent of the records in both the entire dataset and a specific anonymized group is "COVID," an attacker can infer that someone in the group has COVID using this information about the entire dataset. Additionally, models have been created to stop the disclosure of private numerical attribute value ranges. These include using a Worst Group Protection model to prevent range disclosure or limiting the maximum range or variance of sensitive attributes values in a group. The subtleties and difficulties involved in protecting against attribute disclosure are highlighted by these sophisticated techniques.

On the other hand, in order to prevent the association of an individual with sensitive attribute, privacy models are created to guard against attribute disclosure. These models limit the likelihood of assigning a person to a specific sensitive attribute. For instance, the q-uncertainty model [18] sets a lower bound on this probability of q or less.

Other models, like the "h-k-p"-coherence model [19], are made to guard against the disclosure of both sensitive information and identities. This model limits the possibility of inferring sensitive diagnosis codes by treating non-sensitive attributes similarly to K^m -anonymity. Another model, the PS-rule based anonymity model, uses association rules to thwart the disclosure of both sensitive information and identities. The model demands that at least k records from the published dataset contain the set of diagnosis codes in the antecedent part of the rule, and that at most a percentage c of those records also contain the sensitive attribute in the consequent. Unlike some of the other models, this one is flexible and lets data publishers specify specific privacy requirements. By placing limitations on attribute disclosure when dealing with delicate sensitive attributes.

The algorithms developed here typically follow an Apriori-like search strategies. Apriori-like lattice search strategies are techniques used for data anonymization, particularly for demographics. They're based on the concept of 'lattice', which represents all possible ways to generalize quasi-identifier attributes in a data set. Quasi-identifier attributes are non-identifying on their own but when combined with other quasi-identifiers, can potentially be used to re-identify individuals.

The Apriori-like lattice search strategy is named after the Apriori algorithm, which is widely used in association rule mining for frequent itemset mining. It is based on the principle that all non-empty subsets of a frequent itemset must also be frequent.

In the context of data anonymization, a lattice of generalizations is built on the quasi-identifiers. Each node in this lattice represents a different set of generalized values for quasi-identifiers. For example, consider two quasi-identifiers: Ethnicity (with levels English, Welsh, and British) and Gender (with levels Male, Female, and Any). We can create a lattice with nodes representing various combinations like {English, Male}, {English, Any}, {British, Any}, etc.

In the Apriori-like lattice search strategy, the search begins with the most specific nodes (i.e., the least generalized nodes). Then it gradually moves to more generalized nodes, in an incremental fashion. If a node fails to satisfy the privacy model (it allows identity disclosure), the algorithm prunes this node and its descendants, eliminating a large portion of the search space, because it assumes that if a node does not satisfy the privacy model, then its more generalized nodes (its ascendants) won't either (Apriori principle).

The strategy here is to find the least generalized version of the data that still meets privacy requirements, preserving maximum data utility.

4 Algorithms

In this section, four algorithms will be discussed in details that preserve privacy and enforce privacy models. Each algorithm will be described in it's own subsection. The subsections will focus primarily on the challenges and motivation of each specific field and a detailed describtion of the algorithm.

4.1 Anonymizing Transaction Databases for Publication [19]

Transaction datasets are significantly different from traditional relational datasets. Transaction data is a collection of items that represent the activities or behavior of an individual. It includes data like web browsing histories, purchase records, search queries, and healthcare service records, among others. These datasets can be highly diverse and unstructured, and the relationship between items within a transaction is not as straightforward as it is in a relational dataset. Each transaction can vary in length, and each individual can have multiple transactions, making it a complex, multidimensional problem.

The nature of transaction datasets introduces unique privacy challenges and requirements that traditional privacy models like k-anonymity and l-diversity might not sufficiently address. Traditional models generally focus on generalizing or suppressing certain attributes to make individuals indistinguishable within a group. However, in transaction data, there may be no obvious attributes to suppress or generalize, and it may not be clear how to group transactions together without losing important information. Further, the specific combination of items within a transaction may itself be unique and identifying. Also, transaction data often has high-dimensional attributes, and traditional anonymization methods can lead to high information loss when applied to such data. Therefore, there is a need for privacy models and algorithms specifically designed for transaction data, to preserve privacy while retaining the richness and utility of the data for research purposes.

The authors approach in solving the problem is in 4 main steps.

a. First they have given and defined the problem statement. They defined the problem as publishing the data (which has public and private items) such that given an attacker that can know P items about any individual, they may not:

i. Have a high probability of associating any private item with the set P (called breach probability).

ii. They may not narrow down the P items to a small enough number of records.

b. They gave several definitions of the problem and terminology used and put forward a theorem explaining that their problem is NP-hard by formulating it as a vertex cover problem.

c. They proposed the greedy algorithm by growing their set of publishable or safe items starting from an empty set.

d. They then evaluated their algorithm based on the three parameters they have in their algorithm (p- power of attacker, k which is defined in (ii) previously, and breach probability defined in (i). They have also evaluated their model on the percentage of private items in the dataset.

Their greedy algorithm works in Apriori-like search, by starting with a single public and iteratively finding more item subsets that will cause privacy breaches. Their unique approache takes advantage of a concept they call a minimal mole set. The idea is that they find all minimal moles such that removing any item from that mole will make it adhere to the privacy model. Therefore, they can hearusticaly remove items that minimize information loss by removing the item that appears in a high number of minimal moles. Intuitevely this approach, finds public items that are the most useful for attackers and remove them from the dataset.

4.2 Supporting Pattern-Preserving Anonymization for Time-Series Data [20]

The temporal dependencies found in time series datasets create unique challenges, making traditional anonymization or privacy models insufficient.

Firstly, time-series data inherently has high dimensionality. Each time point in the series can be considered as a separate dimension, which leads to the curse of dimensionality when attempting to use traditional privacy preservation techniques. Secondly, the sequential nature of time-series data introduces an order sensitivity that is not found in static (non-time-series) datasets. This sensitivity means that simple alterations in the data to provide privacy can drastically change the underlying patterns and correlations in the data, rendering it useless for further analysis. Therefore, preserving the inherent temporal correlations between data points during the anonymization process is critical. Lastly, time-series data often requires specialized queries, such as range queries and pattern matching, which demand data utility preservation while ensuring privacy. Thus, the development of unique privacy models and algorithms that can cater to these specific characteristics and requirements is necessary.

In this article they employ two strategies and compare them. The first strategy is the top down clustering approach, the idea is that they utilize any privacy model to form groups on the quasi identifiers of the data. This approach will form a number of groups but does not provide any privacy for the time series data itself. Then for each group, they represent the time series data in patterns. The patterns are represented in levels, such higher levels are more detailed but has fewer examples and lower levels are less detailed but has more examples. This allows the algorithm to form grouping similar to Apiori-like search on each group and make sure that each pattern represents atleast P time-series users.

The other approach is a bottom-search where they start by forming the P-groups in all of the datasets and form the K groups in a greedy manner. The idea is that each K group must have atleast K records and they are formed by assigning p-groups to the k-groups in a way that ensures that the privacy of the quasi identifiers are enforced.

This is done by merging each p-group with another group in such a way that it minimizes the information loss (information loss occurs by generalizing the quasi-identifiers). This is a greedy bottom-up approach to grouping the P-groups and it was found to be more effective at preserving the utility of the data.

4.3 Fast Data Anonymization with Low Information Loss [21]

The approach proposed in this article pioneered a new formulation and approach of this problem with a space mapping approach. Their article starts by laying the background on k-anonymity and l-diversity, then they solve the problem of k-anonymity for only one attribute.

They solve the optimal solution of k-anonymity for one attribute usinig dynamic programming. The way they do it is positing a lemma that they can always find an optimal solution without having any group that has more than 2k -1 records. Then they establish the following recursive property:

$\mathrm{Opt}(i) = \min_{i-2k<j<= I - k} (\mathrm{Opt}(j) + \mathrm{Opt}([j+1, i]))$

Therefore The recursive relation Opt(i) = min (Opt(j) + OptI ([j + 1, i])) for i-2k < j ≤ i-k is the heart of the algorithm. For every i (the current record we're considering), it checks all possible previous group boundaries j (falling between i-2k and i-k) to find the one that gives the minimum total information loss. It sums up the minimum information loss Opt(j) up to the boundary record j, and the information loss OptI ([j + 1, i]) when grouping the records from j+1 to i. The purpose of checking between i-2k and i-k is to ensure that each group has between k and 2k-1 records.

The process begins by directly computing the optimal solutions for all j-prefixes of R where k ≤ j ≤ 2k - 1. This serves as the base cases for the dynamic programming solution.

Then, it systematically proceeds with increasing i (2k ≤ i ≤ N), updating the minimum total information loss using the recursive relation for each i.

After the process is complete, the optimal partitioning P is generated and the Generalized Cluster-based Perturbation (GCP) of P is calculated as Opt(N)/N, which represents the average information loss per record in the optimal partitioning.

In essence, the dynamic programming solution is systematically exploring all possible groupings within the given constraints to find the one that minimizes the total information loss.

The time complexity of the solution is O(kNω), this means that the algorithm's grows linearly with the number of records N and the computation time of OptI ([j + 1, i]) represented by ω, and also linearly with the parameter k which defines the size of groups.

- N is the total number of records. For each record (i), the algorithm checks k possible previous boundaries (j), so the computation is done N*k times.
- For each pair (i,j), the computation of OptI ([j + 1, i]) takes some time. This is represented by ω.

After they proposed a solution that grows linearly in time for one property using dynamic programing they extend their solution to multiple attributes using space mapping.

Their space mapping solution utilizes Hilbert curve, The Hilbert curve is a continuous fractal curve that fills up space, and it can be used to map a multi-dimensional space to a one-dimensional space. It works by traversing each region of the multi-dimensional space and assigning it a unique integer in the one-dimensional space. One key property of the Hilbert curve is locality preservation: if two points are close together in the multi-dimensional space, they will also be close together after being mapped to the one-dimensional space using the Hilbert curve.

For instance, the article mentions a transformation from a 2-D to a 1-D space for an 8 x 8 grid. The Hilbert curve passes through each point in the grid, creating a one-dimensional ordering of all the points. By mapping multi-dimensional data to a one-dimensional space using the Hilbert curve, the authors can apply their one-dimensional k-anonymity to multi-dimensional quasi-identifiers. Although the solutions obtained through this mapping are not optimal, the authors argue that the good locality properties of the Hilbert curve mean that the information loss is still relatively low.

4.4 Anonymization of Set-Valued Data via Top-Down, Local Generalization [13]

Query logs are a type of set-valued data record that document the history of all queries performed over a particular system, such as a database or a search engine. They record details about the specific search queries made, including the keywords used, the timestamp of the query, and possibly additional information such as the user's IP address or other identifiers. Query logs are crucial for various purposes including system troubleshooting, optimization, personalization, and trend analysis.

Anonymizing query logs poses more challenges than other datasets for several reasons. Firstly, unlike conventional databases that contain structured records, query logs are often unstructured or semi-structured, which makes the application of traditional anonymization techniques difficult. Furthermore, the context or semantics of the queries might reveal sensitive information about the users even when their identities are anonymized.

The problem this article aims to tackle is similar to the transactions datasets in section 3.1. However, this article utilizes a recursive greedy approach to solve the problem. In essence, this algorithm is a technique for protecting users' privacy in a database by anonymizing set-valued data. The process is iterative and starts with a base state in which all data points are generalized to the top of a hierarchy (for example, by designating all items as "food"). The algorithm then divides this initial group into sub-partitions repeatedly, dividing on a different node in the hierarchy each time, and then performing the anonymization process recursively on the newly created sub-partitions. This procedure is repeated until there is no more way to divide the data.

Let's imagine that each partition has a "hierarchy cut" to indicate the degree of generalization that has been used in order to provide a detailed description. This cut is made up of a number of hierarchy nodes, and as partitions are divided, these nodes are expanded to provide a more detailed representation of the transactions in that partition.

Only one node in the hierarchy cut is expanded at a time when splitting a partition, resulting in various possible buckets (or sub-partitions) depending on which node was expanded. A greedy heuristic that seeks to maximize "information gain," which is defined as the reduction in information loss achieved by splitting on this node, chooses which node to expand.

The algorithm then divides each transaction into a suitable bucket based on its generalization after splitting. Next, the algorithm makes sure that every partition has at least 'k' transactions, maintaining k-anonymity. If a partition has fewer transactions than 'k', it is balanced by transferring data from other buckets or by taking the least amount of 'information gain' transactions from overfilled buckets.

Each new sub-partition in the hierarchy cut shows the growth of a parent node to its child nodes. The transactions contained in the partition and this updated cut are passed on to the following recursive call of the anonymization procedure.

By using this method, the algorithm anonymizes the dataset while preserving as much information as possible and maintaining k-anonymity. This is accomplished by choosing the best course of action based on information gain at each stage of the partitioning process.

5 Conclusion

In our extensive review, we explored a myriad of models, methods, and algorithms designed to tackle privacy threats in data publishing, with specific focus on identity disclosure, membership disclosure, and attribute disclosure. We discovered that these threats often pertain to direct identifiers, quasi-identifiers, and sensitive attributes of datasets, and have consequently been addressed by distinctive models devised for each category of threat.

To counteract identity disclosure, models such as k-anonymity, k-map, and privacy-constrained anonymity play a pivotal role. On the other hand, membership disclosure threats are addressed by models like d-presence and c-confident d-presence. Lastly, to prevent attribute disclosure, models such as l-diversity, t-closeness, and h-k-p-coherence are typically used.

In parallel, we also examined the anonymization of various types of data, namely transaction databases, time-series data, multi-attribute data, and set-valued data such as query logs. Each of these methods demonstrated their ability to balance privacy preservation with data utility, albeit through markedly different approaches. For instance, a greedy algorithm was employed for transaction databases, top-down clustering and a greedy bottom-up method for time-series data, dynamic programming for multi-attribute data, and a recursive greedy approach for set-valued data.

In conclusion, both these exploration tracks provided compelling evidence that effective anonymization and privacy preservation strategies can indeed be developed for different types of data and privacy threats. The choice of model, method, and algorithm should align with the specific requirements of the use case, the data type, and the potential severity of the privacy threats.

6 FUTURE WORK

While current models and methods demonstrate considerable promise, the ever-evolving data landscape and increasing complexity of data structures point towards several potential areas for future research.

First, there's the challenge of novel data types such as Geospatial Data, Genomic Data, and Cyber Physical Systems Data. These data forms present unique privacy preservation obstacles and call for the development of innovative privacy models and algorithms capable of tackling these challenges effectively.

Second, with the scale and complexity of datasets skyrocketing in the big data and AI era, the demand for more scalable and efficient privacy-preserving methods becomes apparent. The time complexities of some of the current algorithms, particularly for high-dimensional and large datasets, necessitate improvements and could benefit from the integration of advanced computational techniques such as machine learning. A particularly interesting approach is generating synthetic datasets from the real world datasets. Therefore, we argue that, the confluence of privacy preservation and machine learning holds intriguing possibilities. Exploring how machine learning techniques can be used to enhance the effectiveness and efficiency of privacy preservation techniques is a compelling research direction.

Lastly, the anonymization of dynamic data sets is a promising area for exploration. The current solutions may not always be optimal, and therefore, there's a need for more research on dynamic data anonymization. Dynamic datasets, continually updated over time, pose significant privacy preservation challenges in data publishing. Traditional anonymization methods, designed for one-time data release, can fall short as changes in the dataset could reveal previously anonymized information. Therefore, advanced techniques that allow real-time or near real-time anonymization are urgently needed. These techniques must balance privacy preservation and data utility, and could utilize differentially private algorithms or machine learning methods.

7 REFERENCES

[1] L. Sweeney, “k-anonymity: A model for protecting privacy,” International journal of uncertainty, fuzziness and knowledge-based systems, vol. 10, no. 05, pp. 557–570, 2002.
[2] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, “l-diversity: Privacy beyond k-anonymity,” ACM Transactions on Knowledgen Discovery from Data (TKDD), vol. 1, no. 1, pp. 3–es, 2007.
[3] H. Wang and Z. Xu, “Cts-dp: publishing correlated time-series data via differential privacy,” Knowledge-Based Systems, vol. 122, pp. 167–179, 2017.
[4] G. Ghinita, P. Kalnis, and Y. Tao, “Anonymous publication of sensitive transactional data,” IEEE Transactions on Knowledge and Data Engineer-ing, vol. 23, no. 2, pp. 161–174, 2010.
[5] X. Ding, C. Wang, K.-K. R. Choo, and H. Jin, “A novel privacy preserving framework for large scale graph data publishing,” IEEE transactions on knowledge and data engineering, vol. 33, no. 2, pp. 331–343, 2019.
[6] C. Dwork, “Differential privacy: A survey of results,” in Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi’an, China, April 25-29, 2008. Proceedings 5. Springer, 2008, pp. 1–19.
[7] A. Zigomitros, F. Casino, A. Solanas, and C. Patsakis, “A survey on privacy properties for data publishing of relational data,” IEEE Access, vol. 8, pp. 51 071–51 099, 2020.
[8] X. Xiao and Y. Tao, “Personalized privacy preservation,” in Proceedings of the 2006 ACM SIGMOD international conference on Management of data, 2006, pp. 229–240.
[9] M. E. Nergiz, M. Atzori, and C. Clifton, “Hiding the presence of individuals from shared databases,” in Proceedings of the 2007 ACM SIGMOD international conference on Management of data, 2007, pp. 665–676.
[10] A. Gkoulalas-Divanis, G. Loukides, and J. Sun, “Publishing data from electronic health records while preserving privacy: A survey of algorithms,” Journal of biomedical informatics, vol. 50, pp. 4–19, 2014.
[11] K. El Emam and F. K. Dankar, “Protecting privacy using k-anonymity,” Journal of the American Medical Informatics Association, vol. 15, no. 5, pp. 627–637, 2008.
[12] A. Gionis, A. Mazza, and T. Tassa, “k-anonymization revisited,” in 2008 IEEE 24th International Conference on Data Engineering. IEEE, 2008, pp. 744–753.

[13] M. Terrovitis, N. Mamoulis, and P. Kalnis, “Privacy-preserving anonymization of set-valued data,” Proceedings of the VLDB Endowment, vol. 1, no. 1,pp. 115–125, 2008.

[14] G. Loukides, A. Gkoulalas-Divanis, and B. Malin, “Anonymization of electronic medical records for validating genome-wide association studies,” Proceedings of the National Academy of Sciences, vol. 107, no. 17, pp. 78987903, 2010.
[15] M. E. Nergiz and C. Clifton, “Thoughts on k-anonymization,” Data & Knowledge Engineering, vol. 63, no. 3, pp. 622–645, 2007.
[16] “δ-presence without complete world knowledge,” IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 6, pp. 868–883, 2009.
[17] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond k-anonymity and l-diversity,” in 2007 IEEE 23rd international conference on data engineering. IEEE, 2006, pp. 106–115.
[18] J. Cao, P. Karras, C. Ra ̈issi, and K.-L. Tan, “ρ-uncertainty: inference proof transaction anonymization,” Proceedings of the VLDB Endowment (PVLDB), vol. 3, no. 1, pp. 1033–1044, 2010.
[19] Y. Xu, K. Wang, A. W.-C. Fu, and P. S. Yu, “Anonymizing transaction databases for publication,” in Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 2008, pp. 767–775.
[20] L. Shou, X. Shang, K. Chen, G. Chen, and C. Zhang, “Supporting pattern preserving anonymization for time-series data,” IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 4, pp. 877–892, 2011.
[21] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, “Fast data anonymiza tion with low information loss,” in Proceedings of the 33rd international conference on Very large data bases, 2007, pp. 758–769.