# Semantic Attack on Anonymised Transaction Data

**A thesis submitted in partial fulfilment**

**of the requirement for the degree of Doctor of Philosophy**

**Asma Shuhail AlShuhail**

**June 2021**

**Cardiff University**
**School of Computer Science & Informatics**

***To my hero,***

***to the one who always believed in me,***

***to whom I miss every day,***

***to whom I promised to dedicate***

***this dissertation before he left this world,***
***to my father, Shuhail AlShuhail***

# Abstract

Publishing data about individuals is a double-edged sword; it can provide a significant benefit for a range of organisations to help understand issues concerning individuals, and improve services they offer. However, it can also represent a serious threat to individuals' privacy. To overcome these threats, researchers have worked on developing anonymisation methods. However, the anonymisation methods do not take into consideration the semantic relationships and meaning of data, which can be exploited by attackers to expose protected data.

In our work, we study a specific anonymisation method called disassociation and investigate if it provides adequate protection for transaction data. The disassociation method hides sensitive links between transaction's items by dividing them into chunks. We propose a de-anonymisation approach to attacking transaction data anonymised by the disassociated data. The approach exploits the semantic relationships between transaction items to reassociate them.

Our findings reveal that the disassociation method may not effectively protect transaction data. Our de-anonymisation approach can recombine approximately 60% of the disassociated items and can break the privacy of nearly 70% of the protected itemets in disassociated transactions.

# Acknowledgements

I would like to express my deep gratitude and appreciation to my supervisor Dr. Jianhua Shao whose guidance, support and encouragement has been invaluable throughout my PhD study.

I would also like to say a heartfelt thank you to my family. Words can not express how grateful I am for all of the sacrifices they have made on my behalf, and their prayers for me were what sustained me thus far. A special thanks to my friend Raheel AlMarei for her unwavering support at every stage of my research.

# Contents

**Abstract** **iii**

**Acknowledgements** **iv**

**Contents** **v**

**List of Figures** **ix**

**List of Tables** **xii**

**1 Introduction** **1**

1.1 Data Privacy and its Protection . . . . . 1

1.2 Anonymising Transaction Data . . . . . 4

1.3 De-anonymising Transaction Data . . . . . 6

1.4 Research Hypothesis and Contributions . . . . . 7

1.5 Thesis Organisation . . . . . 9

**2 Background and Literature Review** **11**

2.1 Data Privacy . . . . . 11

2.2 Attacking Data Privacy and Privacy Models . . . . . 13
2.2.1 Attacker's Background Knowledge . . . . . 13
2.2.2 Linkage Attack . . . . . 15
2.2.3 Minimality Attack . . . . . 19
2.2.4 Inference Attack . . . . . 21
2.3 Data Anonymisation . . . . . 23
2.3.1 Generalisation . . . . . 24
2.3.2 Suppression . . . . . 26
2.3.3 Generalisation and Suppression for Transaction Data . . . . . 26
2.3.4 Anatomisation . . . . . 27
2.3.5 Disassociation . . . . . 28
2.3.6 Perturbation . . . . . 28
2.4 Semantic relationships and data privacy . . . . . 29
2.5 Summary . . . . . 34

**3 The Disassociation Method** **35**

3.1 Preliminaries . . . . . 35
3.2 Disassociation Method . . . . . 37
3.2.1 Horizontal Partitioning . . . . . 38
3.2.2 Vertical Partitioning . . . . . 39
3.2.3 Refining . . . . . 41
3.3 Improving the Original Disassociation Algorithm . . . . . 42

3.3.1 Horizontal Partitioning . . . . . 43

3.3.2 Hierarchical Clustering (HC) for Horizontal Partitioning . . . 50

3.3.3 Vertical Partitioning . . . . . 50

3.4 Comparison of Different Methods of Horizontal Partitioning . . . . . 53

3.5 Summary . . . . . 54

**4 Semantic Attack** **56**

4.1 Semantic Relationships . . . . . 56

4.1.1 Semantic Relatedness Measures . . . . . 57

4.1.2 Semantic Attack Measures . . . . . 59

4.1.3 The Limitations of NGD and WE . . . . . 63

4.1.4 Accuracy of Measures . . . . . 64

4.2 Semantic Attack Approach . . . . . 68

4.2.1 Scoring step: . . . . . 68

4.2.2 Selection step: . . . . . 70

4.3 Attacking methods . . . . . 73

4.3.1 Averaging-based attack (ABA) . . . . . 74

4.3.2 Related-group attack (RGA) . . . . . 79

4.3.3 Most-related attack (MRA) . . . . . 83

4.3.4 Vertical partitioning attack (VPA) . . . . . 85

4.4 Summary . . . . . 89

**5 Experiments and Results** **91**

5.1 Dataset Preparation and Experiment Setup . . . . . 91

5.1.1 Dataset properties . . . . . 91

5.1.2 Experiment Setup . . . . . 93

5.2 Performance Evaluation . . . . . 96

5.2.1 Privacy Breakage . . . . . 97

5.2.2 Reconstruction . . . . . 98

5.3 Results and Discussion . . . . . 100

5.3.1 Effect of $k$ value . . . . . 101

5.3.2 Effect of Data Density . . . . . 109

5.3.3 Record chunks attack . . . . . 112

5.3.4 Term chunks attack . . . . . 114

5.3.5 Effect of max cluster size . . . . . 116

5.3.6 Record chunks attack . . . . . 119

5.3.7 Term chunks attack . . . . . 121

5.4 Summary . . . . . 123

**6 Conclusion** **125**

6.1 Research Summary . . . . . 125

6.1.1 Future Research . . . . . 127

**Bibliography** **129**

# List of Figures

2.1 Re-identifying individuals by linking attack [96] . . . . . 16

2.2 Taxonomy tree for the 'Job' attribute . . . . . 24

2.3 An example of a semantic attack . . . . . 31

3.1 The disassociation method . . . . . 37

3.2 Comparison of the ratio of information loss for different methods . . . 54

4.1 A branch of WordNet's ontology for (eye disease) . . . . . 58

4.2 Word2Vec training models [66] . . . . . 60

4.3 An example of *f ('glaucoma')* by Google . . . . . 63

4.4 Different semantic measurements comparison . . . . . 66

5.1 Comparing the overall accuracy of the attacking methods . . . . . 102

5.2 Comparing the overall WMD of the attacking methods . . . . . 103

5.3 Comparing the overall transaction breakage of the attacking methods . 104

5.4 Comparing the overall $k^m$-anonymity breakage of the attacking methods 104

5.5 Comparing the accuracy of the attacking methods on record chunks . 105

5.6 Comparing the transaction breakage of the attacking methods on record chunks . . . . . 106

5.7 Comparing the $k^m$-anonymity breakage of the attacking methods on record chunks . . . . . 107

5.8 Comparing the accuracy of the attacking methods on term chunks . . 108

5.9 Comparing the transaction breakage of the attacking methods on term chunks . . . . . 108

5.10 Comparing the $k^m$-anonymity breakage of the attacking methods on term chunks . . . . . 109

5.11 Overall accuracy of the attacking methods with different data densities 110

5.12 Overall WMD of the attacking methods with different data densities . 111

5.13 Overall transaction breakage of the attacking methods with different data densities . . . . . 111

5.14 Overall $k^m$-anonymity breakage of the attacking methods with different data densities . . . . . 112

5.15 Accuracy of attacking record chunks with different data densities . . . 113

5.16 Transaction breakage of attacking record chunks with different data densities . . . . . 113

5.17 $k^m$-anonymity breakage of attacking record chunks with different data densities . . . . . 114

5.18 Accuracy of attacking term chunks with different data densities . . . . 115

5.19 Transaction breakage of attacking term chunks with different data densities . . . . . 115

5.20 $k^m$-anonymity breakage of attacking term chunks with different data densities . . . . . 116

5.21 Overall accuracy of the attacking methods with different max cluster sizes . . . . . 117

5.22 Overall WMD of the attacking methods with different max cluster sizes 118

5.23 Overall transaction breakage of the attacking methods with different max cluster sizes . . . . . 118

5.24 Overall $k^m$-anonymity breakage of the attacking methods with different max cluster sizes . . . . . 119

5.25 Accuracy of attacking record chunks with different max cluster sizes . 120

5.26 Transaction breakage of attacking record chunks with different max cluster sizes . . . . . 120

5.27 $k^m$-anonymity breakage of attacking record chunks with different max cluster sizes . . . . . 121

5.28 Accuracy of attacking term chunks with different max cluster sizes . . 122

5.29 Transaction breakage of attacking term chunks with different max cluster sizes . . . . . 122

5.30 $k^m$-anonymity breakage of attacking term chunks with different max cluster sizes . . . . . 123

# List of Tables

1.1 An Example of transaction data . . . . . 4

1.2 Disassociated data . . . . . 5

1.3 The semantic similarities . . . . . 7

2.1 An example of de-identified patient data . . . . . 13

2.2 An example of minimality attack . . . . . 20

2.3 An example of inference attack . . . . . 22

2.4 An example of generalisation . . . . . 25

2.5 An example of suppressed patient data . . . . . 27

2.6 The classification of attacking data privacy papers . . . . . 33

3.1 Horizontal partitioning (the first iteration) . . . . . 38

3.2 Horizontal partitioning (the resulting cluster) . . . . . 39

3.3 Disassociation transactions . . . . . 40

3.4 Original transactions . . . . . 41

3.5 Disassociated transactions without refining . . . . . 42

3.6 Disassociated transactions with refining . . . . . 43

3.7 The first method of vertical partitioning . . . . . 51

3.8 The second method of vertical partitioning . . . . . 52

4.1 Methods comparison . . . . . 67

4.2 Original transactions . . . . . 69

4.3 Disassociated transactions . . . . . 69

4.4 The semantic scores . . . . . 77

4.5 ABA results for example 4.3 . . . . . 78

4.6 Reconstructed transactions (ABA) . . . . . 79

4.7 RGA results for example 4.3 . . . . . 83

4.8 Reconstructed transactions (RGA) . . . . . 83

4.9 MRA results for example 4.3 . . . . . 86

4.10 Reconstructed transactions(MRA) . . . . . 86

4.11 Reconstructed transactions (VPA) . . . . . 89

# *Chapter 1*

# Introduction

The amount of data produced by people is increasing by the day. It is estimated that the amount of data that will be generated per day will reach 165 zettabytes by 2025 [77]. Such data can be collected from different sources, such as social networks, e-commerce websites or healthcare systems, and these data are often published to third-party research and business organisations to enable a wide range of data analyses. Although this type of data publishing can help improve service provisions by organisations and develop new solutions that are otherwise not possible, one issue must be addressed when practicing this: the protection of private and confidential information contained within the datasets to be published. Over the last two decades, much work has been carried out by the research community to understand how individuals' privacy can be protected when the data associated with them need to be published [30].

## 1.1 Data Privacy and its Protection

The need for protecting individuals' privacy has long been regarded as essential [60]. This is because privacy is the basis for maintaining a range of relationships and interactions among people, and we all need to have a control on who can access our personal lives and over our own information so as not to feel violated or be victimised [69]. For centuries, privacy protection has largely been achieved through laws [91]. Many regu-

lations and frameworks have been put forward, granting individuals their fundamental rights to privacy. The most recent example of a major legal framework for privacy protection is the General Data Protection Regulation (GDPR), which sets guidelines on how personal data may be collected and used by orgnisations [88].

Privacy laws have served the society well by deterring people from knowingly or accidentally violating individuals' privacy. However, this approach is becoming increasingly inadequate for protecting individuals' privacy as it pertains to data due to the fact that the information age has fast accelerated the amount and exchange of information [91], whereas privacy protection regulations have been struggling to keep pace with the accelerated technological development. In other words, data about individuals are being constantly generated, and advanced processing and publishing techniques have been developed and used to benefit from the data, but it is hard to guarantee that these techniques will not misuse the data accidentally or intentionally by third parties, thereby violating individuals' privacy. It is also challenging to predict how privacy may be violated. Therefore, it is impractical to rely just on legal systems to predict all the possible ways privacy may be breached by technology and provide protection against them [101]. Hence there is a need for privacy strategies to provide the required protection for data before publishing it.

One common and obvious method to protect data privacy is de-identification. This method removes any information from a published dataset that can be used to uniquely identify an individual, for example, one's national insurance number. However, removing such information may not be enough to protect an individual's privacy because the other information available in a de-identified dataset can still used to identify individuals. For instance, a combination of gender, marital status, date of birth and education, may be used to identify an individual [78]. This type of attribute can be employed by an adversary to identify individuals successfully. In 2006, after Netflix released

de-identified movie-ranking data, researchers from the University of Texas were able to re-identify some individuals associated with the de-identificated records by using IMDb users' movie ratings as auxiliary information [71]. Also, a recent study from the Imperial College proposed a model that successfully re-identified up to 99.98% of sampled anonymised datasets by using demographic attributes [81]. The study claims that even if a dataset is incomplete and sampled, it is possible to re-identify individuals by knowing a few attributes. Hence, removing or hiding unique identifiers is not sufficient for anonymity because there is still some information that can work as identifiers and that need to be anonymised as well.

To overcome this issue, researchers have studied and developed methods to protect data privacy through anonymisation. These methods aim to prevent the intentional or unintentional misuse of data by altering the data in such a way that individuals can no longer be identified directly or indirectly [83]. Anonymisation can be applied using different methods, such as generalisation, suppression or perturbation to protect individuals' identity and their sensitive information in different types of data (e.g., relation [25], text [43], graph [19] or transaction [98]).

The effectiveness of these methods in protecting privacy has also been analysed in previous studies to understand if they are adequate enough to protect data privacy against various types of attacks. In this thesis, we study the effectiveness of a particular anonymisation method called *disassociation* and investigate whether it can provide sufficient protection for transaction data.

## 1.2 Anonymising Transaction Data

Transaction data consists of a set of records, each containing a set of terms or items. One example of transaction data is given in Table 1.1, which contains four records or transactions, in which each describes a set of medical diagnoses and treatments for a patient.

**Table 1.1: An Example of transaction data**

| TID | Items |
|---|---|
| 1 | vessel, blood, treatment, lung, catheterisation |
| 2 | cancer, radiotherapy, lung, treatment |
| 3 | cancer, lung, blood, tumor, biopsy |
| 4 | cancer, blood, treatment, tumor, biopsy |

In Table 1.1, the data has been de-identified, but the privacy of the individuals could still be violated. For instance, if an attacker knows that Mary had a catheterisation and her record is in the dataset, the attacker would be able to find out which record belongs to Mary because there is just one transaction that has catheterisation, thereby allowing the attackers to identify other sensitive information about her in the transaction and violating her privacy.

Transaction data is difficult to protect. Using anonymisation methods such as generalisation or suppression to protect transaction data is likely to discard a lot of valuable information. This is because these methods require differentiation between the sensitive and non-sensitive items in a dataset, but transaction data can be considered sparse multidimensional data, making it difficult for the disassociation method to address this challenge [100].

The disassociation method achieves protection for transaction data by breaking the privacy threating associations among the items in the dataset, rather than by generalising or suppressing any items. It is built on the $k^m$-anonymity privacy model that states that if an attacker has knowledge up to $m$ items, they cannot match their knowledge to fewer than $k$ transactions. In other words, the disassociation method ensures that each combination of $m$ items appears at least $k$ times in the released dataset.

Using the disassociation method, items in transactions are protected by dividing them into groups such that the items in each group will satisfy the $k^m$-anonymity requirement, so the association between the group is broken. To illustrate the disassociation method, consider the example in Table 1.1. If we anonymise this example, Table 1.2 would be the resulting disassociated version of the dataset.

**Table 1.2: Disassociated data**

| | | | |
|---|---|---|---|
| 1 | blood, treatment, lung | | |
| 2 | cancer, lung, treatment | tumor, biopsy | vessel, catheterisation, radiotherapy |
| 3 | cancer, lung, blood | tumor, biopsy | |
| 4 | cancer, blood, treatment | | |

Here, we assume that the released data need to be $2^2$-anonymous. As can be seen, Table 1.2 has separated the terms into three groups. In the first and second columns, each tuple (line of items) is part of an original transaction, and together, they satisfy $2^2$-anonymity. However, because *tumor* or *biopsy* does not appear two times with *lung* or *treatment*, these terms are placed in different columns. Because the two groups are assumed to be disassociated, that is, each tuple in the first group can be paired with any tuple in the second group, an adversary cannot know the relationship regarding how *(tumor, biopsy)* belongs to which tuple in the first group. Therefore, the link between the two groups of tuples has been protected without removing or changing

any terms. The last column contains the terms that appear less than $k$ times in the dataset, and disassociation protects them by placing them in a separate column. For example, $catheterisation$ appears just once, and after disassociation, it is no longer possible to know to which transaction *catheterisation* belongs because any item/tuple in any group is considered to be linkable to any item/tuple in another group. So if an attacker knows that Mary had a catheterisation, they would not be able to know which transaction belongs to her in Table 1.2.

## 1.3 De-anonymising Transaction Data

The disassociation method assumes that the items in a transaction do not have semantic meanings, and it does not take into consideration the semantic relationship between any items in a transaction. However, if an attacker can use these semantic relationships to identify $m$ items in less than $k$ records, then the privacy protection offered by the disassociation method would have been broken.

In our work, we use semantic relationships to attack the protected links between the items in a transaction. First, we calculate semantic relationships among the items in a disassociated dataset, and then, based on this calculation, we reconstruct the dataset by re-establishing the 'broken' links. For instance, by applying our approach to Table 1.2, we obtain Table 1.3.

The semantic similarities in Table 1.3 illustrate the level of semantic relatedness between each item in the last column and the four tuples in the first column of Table 1.2. For *vessel*, its similarity to all the items in each tuple is calculated. The results show that the first tuple has the best semantic similarity score and that *vessel* is more likely to

**Table 1.3: The semantic similarities**

| | | **vessel** | **catheterisation** | **radiotherapy** |
|---|---|---|---|---|
| ***ID*** | blood, treatment, lung | **0.173** | **0.333** | 0.298 |
| 2 | cancer, lung, treatment | 0.151 | 0.317 | **0.442** |
| 3 | cancer, lung, blood | 0.154 | 0.276 | 0.308 |
| 4 | cancer, blood, treatment | 0.150 | 0.279 | 0.359 |

be associated with it. This is because semantically, *vessel* is closer to blood than the other items and they appear frequently together in the medical context, whereas *vessel* and *cancer* appear less frequently together. Similarly, *catheterisation* is considered a treatment for some cardiovascular conditions, so it is more likely to appear with blood and treatment than with cancer. Therefore, semantically, it is more similar to the items in the first tuple, whereas *radiotherapy* is used as a treatment for lung cancer, so it is the most appropriate to associate it with the second tuple. By finding these protected links and combining items based on semantic similarities, the privacy of the data will be breached and the reconstruction of the original dataset becomes feasible.

## 1.4 Research Hypothesis and Contributions

In this research, our hypothesis is that the disassociation method may not provide adequate protection for data because semantic relationships between terms may be exploited to reconstruct the original transactions, thereby breaking the privacy of the data. The main contributions of the thesis include the following:

- We propose a de-anonymisation approach that aims to expose the hidden links between items in a disassociated dataset by analysing semantic relationships

between items to reconstruct the original transactions. Our semantic attack approach consists of two stages: scoring and selection. The scoring stage is about finding and calculating the semantic scores between items in a disassociated dataset. We build the scoring stage based on two different semantic relationship measures: word embedding [39] and normalized Google distance [15]. After finding all the needed semantic scores, the selection stage will use these semantic scores to heuristically reconstruct the original dataset.

- We propose four methods to combine terms and reconstruct transactions: averaging-based attack (ABA), the most related attack (MRA), related group attack (RGA) and Vertical Partitioning Attack (VPA). The ABA uses the semantic similarities for the all items between two columns in the selection stage. Hence, it calculates the average of the semantic similarities between an item and tuples or two tuples and then reassociates the items and tuples based on the best semantic average. The MRA considers just one item that has the best semantic similarity in each tuple, then, based on the semantic similarities of these items, the tuple will be selected for the reassociation. In the RGA, the items in tuples will be divided into two groups related group and non-related group and just the items in the related group will be considered in the selection stage. In the VPA, the possible permutations between tuples can be found, and then, in the selection stage, it will use the $k^m$-anonymity condition to select tuples for reassociation.

- We evaluate our approach using some real-word datasets. We introduce two measures. The first measures how our approach may break privacy in two different ways: transactions breakage and $k^m$-anonymity breakage. In transactions breakage, we calculate how many transactions the approach can break by correctly reassociating at least one term to a transaction. The $k^m$-anonymity breakage calculates the breakage based on attacking protected infrequent item sets.

The infrequent item sets are combinations of $m$ terms that appear less than $k$ times in a dataset. The second measurement assesses how much of the original information can be correctly reconstructed from the disassociated dataset. We used accuracy and word mover's distance for this. The accuracy measures the proportion of correct reconstructions. Thus, it evaluates how many items in a transaction can be reconstructed by the approach. On the other hand, in the word mover's distance, we measure the reconstruction of information by finding the semantic distance between the original dataset and the reconstructed dataset. However, in the process of reconstruction, measures are calculated based on semantic scores, and the confidence of reconstruction has been left as future work.

Our experiments show that the transaction data may not be adequately protected by the disassociation method. About 60% of disassociated items can be reconstructed by our de-anonymisation approach. Also, this could break the privacy of almost 70% of protected itemsets in the disassociated transactions in our experiments.

## 1.5 Thesis Organisation

Chapter two discusses the general concept of data privacy and then reviews the related privacy protection techniques and privacy models. After that, we discuss three relevant types of attacks on data privacy. Also, it reviews the relationships between semantic similarity and data privacy. Finally, this chapter presents a classification for the reviewed work in attacking data privacy.

In chapter three, we review the disassociation method of transaction data anonymisation and propose three strategies for implementing the first step in the disassociation

method, which is called 'horizontal partitioning'. The strategies are: suppression, adding and remaining-list.

Chapter four introduces our approach to semantic attack. The attacking consists of two stages: scoring and selection. First, we present the scoring component, which is based on two different semantic measures: normalised Google distance and word embedding. Then, we go through the selection component and our proposed methods for this component. Afterwards, we illustrated how we have used these two stages to attack the two types of chunks in the disassociation method (record chunks and term chunks).

Chapter five starts by describing the datasets used in our experiments, along with the properties of these datasets. This is followed by an empirical evaluation to compare and test the different semantic attack methods in our approach.

In chapter six, we recap and conclude the thesis. We also discuss possible future work.

# *Chapter 2*

# Background and Literature Review

This research studies the risk associated with anonymised datasets. In order to understand how anonymised datasets may be de-anonymised, we must first understand how they are anonymised.

This chapter begins by reviewing the concept of data privacy and discussing how data privacy may be violated. Then, we consider the privacy risks associated with releasing individuals' data, and how the data may be protected by reviewing the relevant techniques for protecting data privacy. Finally, we discuss the related work on how semantic relationships have been used to attack data.

## 2.1 Data Privacy

The concept of privacy has been debated for decades and involves many aspects such as philosophical, social, and legal aspects [91]. Privacy is considered as a fundamental human right that is necessary for the protecting of human dignity and retaining autonomy and freedom. Historically, there have been numerous attempts to define privacy, but no unified definition of privacy has been produced yet. This is because privacy may be interpreted in many ways, and there are different aspects to the term 'privacy'. Also, significant differences between different societies and cultures could affect what people consider to be private. Nevertheless, with rapid technological developments, privacy

has been defined as the right of an individual to have control over who can have access to their personal information and how it is collected and used [97], [90].

Sharing personal information with organisations such as e-commerce enterprises and hospital services has become a requirement for using their services. For example, an order from a shopping website cannot be submitted without the name, address and credit card details of the purchaser. To register with a general practitioner (GP), even more private data are required, such as the patient's full name, date of birth and medical history. Such information is often released after some basic de-identification by these organisations to third parties for data analysis. These analyses can serve different purposes, such as to improve services, target advertising or develop new medicines. However, because this information may contain sensitive data about individuals, sharing it with other parties may violate individuals' privacy.

The simplest method used to prevent the violation of individuals' privacy in a published dataset is to de-identify the data. This involves removing or replacing explicit identifying data, such as names, before releasing the dataset to other parties. For example, in Table 2.1, the patients' information has been protected by removing their names. After de-identification, the dataset can then be released for medical research analysis. However, the age, gender and ZIP code of the patients in this released dataset can be used indirectly to identify individuals. This is because these attributes are considered as quasi-identifiers (QIDs), where each attribute of the QIDs is not a unique or direct identifier of an individual but these attributes can be combined to produce a unique identifier for an individual. For instance, if an attacker knows that John is in this dataset and that he is 50 years old, then the attacker can use the age and gender attributes to find out that the first record belongs to John.

Therefore, removing explicit identifiers is not enough protect data privacy. This is

**Table 2.1: An example of de-identified patient data**

| Age | Gender | ZIP code | Disease |
|---|---|---|---|
| 50 | Male | 43551 | Heart disease |
| 35 | Male | 43520 | Diabetes |
| 34 | Female | 43551 | Heart disease |
| 27 | Female | 43532 | Flu |
| 42 | Male | 43550 | HIV |

because privacy can be threatened if QIDs are published as they are. As a result, researchers in the area of privacy preserving data publishing (PPDP) have developed many anonymisation techniques to address this, that is, to make personal data usable without breaching individuals' privacy. These techniques protect data privacy by hiding (also referred to as sanitising) the direct and indirect identifiers that may lead to an individual being identified [30]. The next sections provide a review of some ways of attacking data privacy and privacy models, and we discuss some relevant anonymisation techniques for protecting data privacy.

## 2.2 Attacking Data Privacy and Privacy Models

### 2.2.1 Attacker's Background Knowledge

In general, data privacy protection aims to prevent an attacker from learning any additional information about any target victim in the published data. So if an attacker is able to connect an individual to a specific entry in a released dataset or expose sensitive attributes associated with him or her, then a violation of data privacy has occurred. Protecting data privacy becomes a challenging task when the attacker has some background knowledge obtained from different sources and uses their background know-

ledge to identify individuals from de-identified or otherwise protected data. For example, Wondracek et al. [107] provided an approach for attacking data by using extra information about a user's group membership to identify a user or, at least, to produce a list of possibilities. Narayanan and Shmatikov [72] also explained how an attacker can use the Internet Movie Database (IMDb) as background knowledge to identify anonymised Netflix records. Likewise, Frankowski and Cosley [29] illustarted the re-identification of protected data by using a public web movie forum to re-identify users in a private movie ratings dataset.

Background knowledge can also help an attacker learn more about individuals from the published data. For example, the attacker can exploit publicly available information such as marriages and birth announcements and use them to gain additional information about individuals, as illustrated by Griffith and Jakobsson [35].

However, the kind and amount of background knowledge an attacker may have or use will vary; may only be general knowledge obtained by the attacker from observation. Wang el al. [106] illustrated how some observed attributes can impact data privacy and data anonymization. For example, the attribute of *disease* can be considered a sensitive attribute but some diseases in this attribute, can be observed in an individual, such as flu.

Also, different releases of a dataset may be used by an attacker as an auxiliary knowledge to violate data privacy. Data privacy can be threatened if multiple sanitised versions of the same dataset are being published [52], [104], [112]. Ganta et al. [32] considered this type of background knowledge by studying the composition attack, wherein an attacker uses independently anonymised releases to attack data privacy.

Moreover, an attacker's background knowledge can cover how the data were sanitised

before publishing [41], [32], using this to attack the protected. Wong et al. [108] considered this type of knowledge to violate data privacy, where an attacker has knowledge about the partitioning algorithm used to protect data and use this knowledge to re-group individuals' data in order to violate privacy.

Therefore, even though data anonymisation has been developed to prevent individuals from being re-identified and sensitive information from being disclosed in published data, an attacker's background knowledge still can be used to attack the published dataset. In the following, we discuss three well-studied types of attacks: linkage attack, minimality attack and inferring attack.

### 2.2.2 Linkage Attack

The linkage attack is when an attacker links a record to the record owner or to a sensitive attribute in a published dataset by combining that data with an external dataset. In this attack, the attacker may use quasi-identifiers, such as ZIP code or gender that, are present in both datasets to establish linking connections. The linkage attack has been discussed in many studies [118], [3], [41], [29], [45], [73], [94].

One example illustrating linkage attacks was given by Sweeney [96]. She described a real-life record linkage attack by matching quasi-identifiers between a medical dataset that was published by the Group Insurance Commission in Massachusetts and the voter registration list for Cambridge, Massachusetts. Despite the fact that all the explicit identifiers in the medical dataset have been removed, she was able to re-identify the governor of Massachusetts, William Weld, by linking together his quasi-identifiers in both datasets. She used the combination of zip code, date of birth and sex that were in both the public voter list and the published medical database to re-identify individual

as shown in Figure 2.1. According to Sweeney [96], 87% of the US population is uniquely identifiable by their zip code, gender and date of birth.

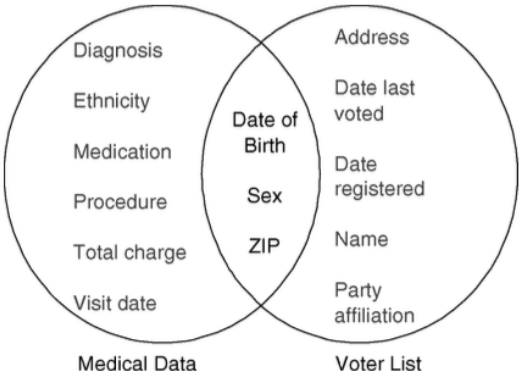



**Figure 2.1: Re-identifying individuals by linking attack [96]**

To illustrate re-identifying records in a linkage attack, we consider the de-identified patient data in Table 2.1. We assume that the attacker has knowledge about the victim's QIDs and that the victim's record is in this de-identified patient table. Hence, if the attacker knows that a person called Mary who lives in the zip code area of 43551 is in the released table, then the attacker can find out that the third record in Table 2.1 belongs to Mary.

To prevent the record linkage attack, privacy models such as $k$-anonymity, $l$-diveresity and $t$-closeness have been developed [85], [84], [61], [55].

- ***$k$-anonymity***

  A dataset is $k$-anonymous if no individual in the released data can be identified from a group of size less than $k$ based on its QID values. Hence, an attacker will not be able to identify an individual's particular record because there are $k$-1 records with the same QIDs [16], [74], [75]. This group of $k$ records with

the same QIDs is called an equivalent group. In this model, the $k$ value is used to determine the level of privacy protection: a higher $k$ value means it is more difficult to identify records, but it also means that the data are less useful [96] because more sanitisation must be done to achieve this level of protection. For example, the records in Table 2.2b are 2-anonymous, where age, gender and ZIP code are QIDs and disease is a sensitive attribute. With this anonymised data, even if an attacker knows that the dataset includes a 34-year-old female who lives at an address with a zip code 43551, he or she will not be able to identify which record is hers.

The simplicity of the k-anonymity model and availability of its algorithms have made it the most common privacy model used with generalisation and suppression. However, $k$-anonymity focuses on QIDs and does not take into consideration the values of sensitive attributes [114], which can expose the published dataset to an attribute linkage attack. Therefore, an attacker may be able to infer a sensitive value of a victim from the published data without identifying which record belongs to the victim. This is because of the lack of diversity in the sensitive attribute in equivalence groups created by $k$-anonymity. For example, in Table 2.2b, if the records of two females show that they have the same disease (heart disease), then an attacker could gain access to the female's sensitive information, even if they do not know which record belongs to her.

$k$-anonymity has a special form called $k^m$-anonymity for high-dimensional data. The key principle of $k^m$-anonymity is that an attacker is assumed to only know up to $m$ of QID values, and each combination of $m$ QIDs should appear $k$ times in the released dataset [99].

- ***l*-diversity**

$l$-diversity was proposed to address the privacy risk associated with an attribute linkage attack on sensitive attributes, which is not covered by k-anonymity. This model focuses on increasing the diversity of sensitive values within an equivalence group. In other words, each equivalence group should contain at least $l$ distinct values for the sensitive attribute [61], [33].

The principle of $l$-diversity requires that for each group of records (equivalence class), there are $l$ 'well-represented' sensitive values. This is because some sensitive values such as Flu naturally appear more frequently than others such as HIV. This can allow an attacker to deduce that a record in an equivalence class is very likely to have those frequent values even if QIDs are already $k$-anonymised. As a result, entropy $l$-diversity and recursive ($c$,$l$)-diversity have been proposed.

Entropy $l$-diversity ensures this for each block of records:

$$-\sum_{s \in S} P(qid, s) \log(P(qid, s)) \geq \log(\ell) \tag{2.1}$$

where $s$ is a sensitive attribute, and $P(qid, s)$ is the fraction of records that have this sensitive attribute in one records group. The left-hand side is referred to as the sensitive attribute's entropy. So, a larger value of $l$ indicates that inferring a certain sensitive value in a records group is more difficult.

Recursive ($c$,$l$)-diversity ensures that the most frequently occurring value does not appear too frequently, and that the less frequently occurring values do not appear too infrequently.

However, $l$-diversity does not take into consideration the problem of the skewed distribution of those values, assuming that each sensitive attribute's value is distributed equally over the dataset.

- ***$t$-closeness***

  To avoid the disclosure of sensitive attributes from a skewed distribution, Li et al. [55] proposed the $t$-closeness model. This model requires that the distribution of a sensitive attribute in any equivalence group be close to the corresponding distribution in the original dataset [1]. To put this in another way, the distance between the distribution of a sensitive attribute in an equivalence group and the distribution of the same attribute in the original dataset needs to be less than or equal to $t$. Although $t$-closeness helps overcome the problem of skewed distribution, it could degrade data utility because it requires the same distribution of sensitive values in all equivalence groups.

### 2.2.3 Minimality Attack

In the minimality attack, in addition to background knowledge about the victim's QIDs and that the victim's data is in the published dataset, an attacker is assumed to have knowledge about the anonymisation mechanism used and privacy requirements for the published dataset. The attacker may examine the published dataset or its documentation and learn about the mechanism behind the anonymisation algorithm, which then allows them to identify the privacy requirements that have been used. The attacker can then break the anonymity by using this knowledge [30], [108], [18], [115].

To illustrate the minimality attack, consider the patient dataset in Table 2.2 (a), where each record belongs to one patient. Although explicit person identifying attributes such as names have been removed, the dataset still can cause privacy issues. If an attacker knows that a patient called Andy is in this dataset and that he is a 36-year-old, then the attacker can infer that the second record belongs to Andy. To protect the dataset, it is converted to a 2-anonymous dataset in Table 2.2 (b), so there are at least two identical

**Table 2.2: An example of minimality attack**

(a) Patient dataset

| Age | ZIP code | Gender | Disease |
|---|---|---|---|
| 30-40 | 43551 | Female | HIV |
| 30-40 | 43551 | Male | HIV |
| 50-60 | 43551 | Female | Flu |
| 50-60 | 43551 | Male | Heart disease |
| 50-60 | 43551 | Male | Flu |

(b) k-anonymity patient dataset

| Age | ZIP code | Gender | Disease |
|---|---|---|---|
| 30-40 | 43551 | * | HIV |
| 30-40 | 43551 | * | HIV |
| 50-60 | 43551 | * | * |
| 50-60 | 43551 | * | * |
| 50-60 | 43551 | * | * |

records for each patient.

However, if the attacker knows that $k$-anonymity has been used and the gender attribute has either 'male' or 'female' in it, then the attacker can infer that the first and second records must contain both 'male' and 'female'. This is because if the gender attribute value is the same in both records, then the value will not be anonymised. As a result, the attacker can infer that either the first or second records belong to Andy and that Andy has HIV. As such, anonymity is broken.

The minimality attack exploits the principle that any anonymisation mechanism needs

to define a minimum requirement or a limit where beyond this limit, the anonymisation model should not generalise, suppress, or distort the data [54]. Therefore, a minimality attack is applicable to most privacy models that define a minimum requirement, such as $l$-diversity [61], $t$-closeness [55], $k^m$-anonymity [99], $(a, k)$-anonymity [109] and $(k, e)$-anonymity [116],[64].

### 2.2.4 Inference Attack

The inference attack occurs when an attacker can deduce sensitive information that they do not have access to from accessible non-sensitive information by using common knowledge and authorised query results [27]. Analysing the tools can lead to an inferring attack by finding information not expected to be found in a published dataset. For example, data analysing or data mining tools can be used to discover sensitive patterns or correlations within data that violate the privacy of individuals [102], [17].

To illustrate the inference attack, let us consider the example given in Table 2.3 for an employee dataset. This dataset contains name, age and salary information. The salary information is protected by limiting access via queries. So for this data, a query for the sum of salaries of multiple employees can be answered, while a query about a specific employee's salary is prohibited. However, the attacker can use the age attribute to infer the salary of Andy, for example. Andy is the only employee who aged 36 in this dataset. Therefore, by submitting the two queries $q1$ and $q2$, where one is for the sum of salaries for employees aged between 36 and 42 and the other query for the sum of salaries for employees aged between 37 and 42, the attacker can find out the salary of Andy's salary:

$q1$ : select sum(SALARY) from EMPLOYEE where AGE $\geq$ 36 and AGE $\leq$ 42

$q2$ : select sum(SALARY) from EMPLOYEE where AGE $\geq$ 37 and AGE $\leq$ 42

**Table 2.3: An example of inference attack**

| Name | Age | Salary |
|---|---|---|
| Andy | 36 | 2900 |
| John | 40 | 3600 |
| Alice | 37 | 3200 |
| Mary | 42 | 3400 |
| Fred | 41 | 3900 |

There are a number of studies that illustrate the inference attack on anonymised data[26], [79]. For example, Kifer [51] illustrated how to use the non-sensitive attributes of one individual to attack the data anonynised by the anatomy method, hence being able to learn the correlations between attributes.

- ***$\varepsilon$-differential privacy***

  Differential privacy guarantees no privacy leakage when sharing summary statistics from datasets by protecting individuals against inference attacks [24], [10]. This model ensures that releasing the aggregate results does not disclose too much information about any individual who contributes to these results. The privacy risks in statistical disclosure control focus on the possibility of an attacker correlating different published statistics to recover sensitive data. Hence, in addition to an attacker's background knowledge, multiple queries into published statistics can provide the attacker with extra information that can help in predicting sensitive information [30].

  To overcome this risk, differential privacy has been proposed. It offers a mathematically proven assurance of privacy protection against various privacy attacks

that are defined as attempts to learn personal information from a published dataset, for example re-identification and record linkage. Differential privacy focuses on the concept that the exclusion or inclusion of a single individual from the database should not (significantly) impact the results of a given query [23], [110]. In other words, whether or not that individual's private data is included in the dataset, an attacker viewing the output of a differentially private analysis will essentially draw the same conclusion regarding that individual's private data.

Differential privacy provides a privacy loss parameter ($\varepsilon$) to control how much noise will be added to the data. The value of $\varepsilon$ determines the balance between the desired protection and the accuracy of the query results. The lower the value of $\varepsilon$, the stronger the privacy, but this also means more noise is applied to the results, leading to less accurate results. For example, if $\varepsilon$ value is equal to $0$ (0-differential privacy), then this means highest protection privacy and the lowest accuracy, so the data would be useless, because the queries results will just be the added noise. This may be an issue when there is a high diversity in the dataset and the value of $\varepsilon$ is too low to provide a required protection.

## 2.3 Data Anonymisation

As previously mentioned, protecting data privacy by anonymisation aims to make the data available to the public without breaching privacy or disclosing the identity or sensitive information of an individual. Anonymisation is the process of sanitising data in a way that it is no longer possible to identify individuals. Data anonymisation techniques include generalisation, suppression, anatomisation, disassociation and perturbation.

### 2.3.1 Generalisation

Generalisation is an anonymisation method. This method replaces specific values with general values to prevent attackers from violating data privacy. For example, the categorical attribute value are replaced with superordinate values, whereas numerical attribute values can be replaced with range values. Generalisation changes the values of QID attributes into more general but equivalent values to make an individual indistinguishable from a group of individuals [42], [105], [31], [58].

Generalisation usually uses taxonomy trees to describe categorical attribute hierarchies [95], [31]. For example, Figure 2.2 illustrates a taxonomy tree for the attribute 'job'. Based on this taxonomy, 'lawyer' and 'engineer' can be generalised to 'professional'. To illustrate generalisation, consider the example of a patient dataset in Table 2.4a. To anonymise the dataset, the attributes 'Job' and 'Age' will be generalised. For the job attribute, the 'Job' taxonomy in Figure 2.2 is used, while for the age attribute, the range values are used; for example, 52 and 59 are replaced by [50-60].

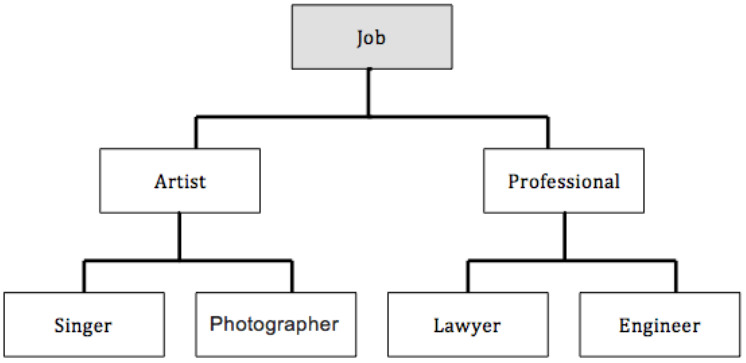



**Figure 2.2: Taxonomy tree for the 'Job' attribute**

**Table 2.4: An example of generalisation**

(a) Patient dataset

| Age | Job | Gender | Disease |
|---|---|---|---|
| 30 | Lawyer | Male | Heart disease |
| 35 | Lawyer | Male | Diabetes |
| 52 | Lawyer | Female | Heart disease |
| 59 | Engineer | Female | Flu |
| 34 | Engineer | Male | HIV |

(b) Generalised patient dataset

| Age | Jobe | Gender | Disease |
|---|---|---|---|
| 30-40 | Professional | Male | Heart disease |
| 30-40 | Professional | Male | Diabetes |
| 50-60 | Professional | Female | Heart disease |
| 50-60 | Professional | Female | Flu |
| 30-40 | Professional | Male | HIV |

### 2.3.2 Suppression

With the suppression method, a value is not released at all. Suppression is applied to QID attributes to prevent the re-identification by removeing values from the released data [57], [47]. For example, Table in 2.5 shows the anonymised patient dataset by generalising the job attribute and suppressing the age attribute, which is done by replacing values with a special value (an asterisk) to prevent the identification of an patient through the combination of age and job attributes.

Suppression can be considered a special form of generalisation. Also, both suppression and generalisation achieve protection by replacing the original values of the QID attributes. However, these methods introduce different degrees of distortion, and to ensure the balance of data utility and privacy, the values need to not be excessively distorted.

### 2.3.3 Generalisation and Suppression for Transaction Data

Unlike the relational data, in the transaction data, it is hard to distinguish between quasi-identifiers and sensitive items. However, to generalise or suppress transaction data, several works have been proposed to address their anonymisation. For example, global generalization applies the $k^m$-anonymity on transactions where each subset of no more than $m$ items appears in at least $k$ transactions and if the dataset does not satisfy $k^m$-anonymity requirement, then precise items will be replaced with more generalised ones [99]. For suppression, an sensitive item will be deleted from all transactions or a whole transaction can be deleted from the released datset. To control information loss in suppression, $(h, k, p)$-coherence has been proposed [113].

**Table 2.5: An example of suppressed patient data**

| Age | Job | Gender | Disease |
|---|---|---|---|
| * | Professional | Male | Heart disease |
| * | Professional | Male | Diabetes |
| * | Professional | Female | Heart disease |
| * | Professional | Female | Flu |
| * | Professional | Male | HIV |

### 2.3.4 Anatomisation

A key advantage of using anatomisation is that it allows for a more accurate data analysis, because it anonymises data by disassociating the correlation between QIDs and sensitive attributes without changing any data. This is in contrast to generalisation and suppression, which change data to anonymise it. Xiao and Tao [111] showed that anatomised tables outperformed the generalised tables in answering aggregate queries involving QID and sensitive attributes because the original data were not modified.

Anatomisation anonymises a dataset by releasing data in two separate tables: one table contains QID attributes while the other table contains sensitive attributes. This particular method links these two tables by adding GroupID in both. Thus, the records in one group will have the same GroupID value in the QID table and sensitive attributes table. Each group in QIDs table will link to $l$ distinct sensitive values. As a result, if each distinct sensitive value happens just once in a group, the probability to link a distinct sensitive value to a record by using the GroupID value will be $1/l$.

However, anatomisation is not sufficient for anoymised data in the case of continuous data publishing. Because this method does not modify any attribute values, the records for the same individuals may remain the same across all releases. In addition,

the impact of applying data mining tools such as classification or clustering on the anatomised tables is not clear [30].

Moreover, this method tends to be used with relational datasets that usually contain demographic information such as age, gender, income, education and employment. This type of information does not have the semantic property that makes anatomisation less vulnerable for the semantic attack.

### 2.3.5 Disassociation

The anatomisation method is applied based on the possibility to classify attributes into QIDs and sensitive attributes. However, it may be difficult to differentiate the two types of data, especially in sparse multidimensional data. One example is web search query logs, which can contain millions of terms, so classifying these terms into sensitive or non-sensitive is difficult. Also, applying suppression and generalisation to this data would distort of the some most valuable information. Because of this, the disassociation method has been proposed for anonymising transaction data. Disassociation anonymises data by splitting a record's items into groups to hide the correlation between them. This method uses the $k^m$-anonymity privacy model to split record items [100], [59]. This method will be expanded on in detail in chapter 3.

### 2.3.6 Perturbation

Perturbation is used primarily as a part of statistical disclosure control to protect individuals' confidential data in statistical information [11], [48]. Perturbation modifies the data in such a way that it would allow for a summary statistical information to be

released without disclosing individuals' confidential data.

Additive noise, data swapping and synthetic data generation are commonly used perturbation techniques. The additive noise method is designed to protect numerical sensitive attributes such as income [9], [2]. It adds a randomised value that is chosen from some distribution to the original sensitive numerical values to distort them. However, if the correlation between attributes is high, the original values can be retrieved from randomised data by using noise removal techniques [48], [12], [49]. Also, Sramka et al. [92] illustated how the protection offered by this technique can be violated; they proposed a fusion technique to remove noise from published anonymised data, where they used the combined results of multiple data miners to give an estimation for the original data.

Another paerturbation method is data swapping. This method was designed to protect numerically and categorically sensitive attributes by switching the sensitive attributes values between records while preserving the underlying statistics of the data [20], [70], [103].

Synthetic data generation is another technique of perturbation. It protects data privacy by substituting synthetic data values for the original data values in a way that the difference between the statistical information derived from the perturbed data and statistical information derived from original data is insignificant [80], [38].

## 2.4 Semantic relationships and data privacy

The term 'semantic relationship' refers to the associations that exist between words, phrases or sentences, such as synonym or hyponyms. However, most data privacy

protection techniques focus on the distribution of data values without considering the meaning of the data and semantic relationships that may exist among them. Exploring such meanings and semantic relationships that exist among the terms can raise privacy threats to data protected by methods that do not consider them. In other words, the semantic relationships among the items in a dataset can allow for indirect semantic inferences; for example, some drugs are only related to a given disease, and some customs can be linked only to certain religions [7], [13], [63], [65].

In general, anonymisation approaches depend on formal privacy models such as $k$-anonymity, which uses privacy parameters to guarantee privacy by making records indistinguishable. However, these parameters cannot adequately capture semantic relationships. Therefore, semantic inferences can be used to disclose protected items, hence leading to re-identification or exposing sensitive information.

Figure 2.3 illustrates an example of exploiting semantic relationships in attacking anonymised data [89]. The original medical transactions in Figure 2.3 (a) have been anonymised by a set-based generlisation [58] to produce the result shown in Figure 2.3 (b). In set-based generlisation method, data is protected by replacing a single item with a set of items. Assume that insulin is a sensitive item that needs to be hidden, as are sneezing and petechiae in transactions (2) and (3). Set-based generalisation is applied by grouping insulin, sneezing and petechiae in a set (Figure 2.3 (b)) to protect these items.

However, semantic relationships can be used to find the associations between items in a set and the rest of the terms in a transaction. Thus, an attacker would find that the strongest relationship for transaction (1) is between diabetes and insulin (Figure 2.3 (c)). This type of semantic relationship allows an attacker to reduce the number of members in the generalised set by removing fake items, hence increasing the likelihood

to determine the original transaction, and violating the individual's privacy (Figure 2.3 (d)).

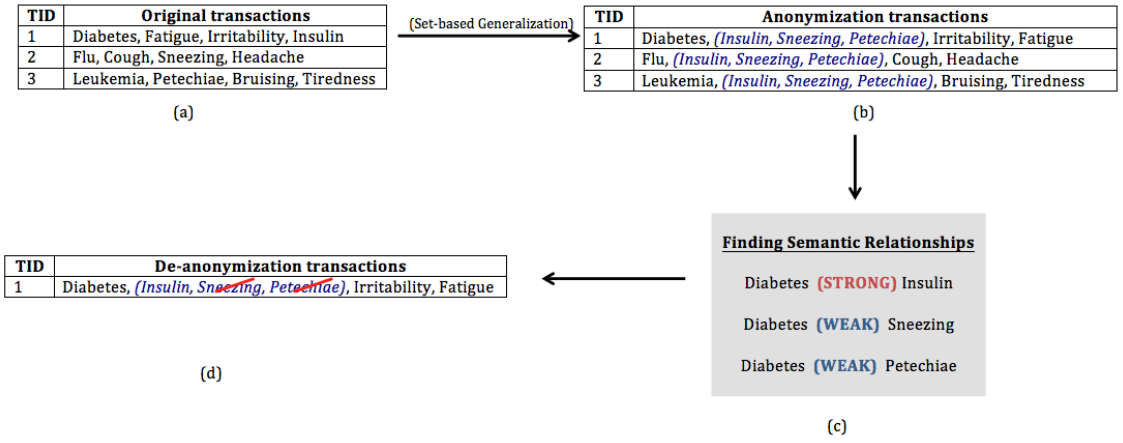



**Figure 2.3: An example of semanitc attack**

Ong and Shao proposed a method for attacking set-generalised transactions based on semantic relationships [89]. They assumed that the items in a transaction occur in a coherent manner; consider transaction (2) in table (a) of Figure 2.3 (flu, cough, sneezing and headache), for example. They used the neighbouring items of the generalised set as contextual items. That is, flu is used as a contextual item to determine which items in the generalised item is likely to be a fake item.

The type of semantic relationship in Table 2.3 explains a semantic inference where insulin is considered as a related treatment to diabetes. This type of inference depends on the co-occurrence of two items in a context. In other words, insulin and diabetes appear often together in the medical context; therefore, the semantic relationship between them is strong. However, a number of tools in natural language processing (NLP) can be used to understand and interpret semantic relationships, and attackers can employ these tools to attack anonymised datasets. For example, Sanchez et al. [87] measured the semantic distance between terms using point-wise mutual information (PMI)

and used the World Wide Web (WWW) as a corpus to find related terms [8], [86], [93], [14]. They used these calculations to determine semantically related terms that can lead to the disclosure of sensitive information. Chow et al. [13] used word co-occurrences on the web as a part of their inference detection model to predict what an adversary can infer and to detect undesired inferences that may be derived from text.

Table 2.6 presents a summary and classification of the papers reviewed here on risks and their protections. Compared to previous work, our proposed approach is the first work that exploits semantic relationships in attacking disassociated data.

**Table 2.6: The classification of attacking data privacy papers**

| Ref No | Privacy Model | | | | Anonymisation method | | | | Attack | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $k^m - anonymity$ | K-anonymity | l-diversity | Differential privacy | Generalization | Suppression | Disassocation/ Anatomization | Perturbation | Linkage | Minimality | Inferring | Semantic |
| [108] | | X | X | | X | | | | | X | | |
| [3] | | | | | | X | | | X | | | |
| [94] | | | | | | X | | | | | X | |
| [51] | | X | | | | | X | | | | X | |
| [92] | | | | X | | | | X | | | X | |
| [18] | | | X | | X | | | | | X | | |
| [41] | | X | X | | X | | | | X | | | |
| [106] | | X | | | X | X | | | X | | | |
| [32] | | X | X | | X | | | | X | | | |
| [72] | | X | | | X | X | | X | X | | X | |
| [29] | | X | | | | X | | X | X | | X | |
| [45] | | | X | | | X | | | X | | | |
| [118] | | X | | | X | X | | | X | | | |
| [64] | | X | | | | | X | | | | X | |
| [89] | X | | | | X | | | | | | | X |
| The proposed approach | X | | | | | | X | | | | | X |

## 2.5 Summary

In this chapter, we discussed data privacy and how individuals' privacy in published datasets may be protected. We started by reviewing the concept of data privacy and then illustrated how background knowledge can be used to assist attackers in attacking privacy. After that, we discussed some of the most common privacy risks associated with data publishing by reviewing a number of well-known attacks. In addition, we discussed privacy models and reviewed some anonymisation techniques, including generalisation, suppression, anatomisation, disassociation, and perturbation. Finally, we showed how the concept of semantic relationships can be used to violate data privacy.

Although most anonymisation techniques that protect data depend on their distribution in the dataset, the semantic meaning of data can be exploited by attackers to break anonymity. Most of existing works have focused on considering term relationships to violate the privacy of generalised data. However, to the best of our knowledge, our work is the first to consider semantic relationships to de-anonymise transactions data that were anonymised by the disassociation method. We use two semantic similarity measures to infer semantic relationships between items and then exploit these relationships to break the privacy of the anonymised transactions.

# *Chapter 3*

# The Disassociation Method

This chapter illustrates the original disassociation method, which is essential for understanding our semantic attack. We first give some necessary definitions and then explain some limitations associated with the original method. To address these issues, we propose new strategies for horizontal partitioning: *suppression, adding* and *remaining-list*. These strategies aim to handle small clusters that have less than $k$ transactions in them. Also, we explain two different methods to perform the vertical partitioning.

## 3.1 Preliminaries

The disassociation method is an anonymised method that is designed to protect the identities and sensitive information of individuals in published transaction datasets [100]. Disassociation preserves the original terms, but hides the fact that two or more different infrequent terms appear in the same transaction. In other words, it protects the individuals' privacy by disassociating the transaction's terms that participate in identifying combinations to prevent an attacker from using those infrequent combinations to identify individuals within a published dataset.

Let $W = (w_1, ....., w_m)$ be a finite set of words called terms. A transaction $T$ over $W$ is a set of terms $T$=($t_1$,$t_2$,.....,$t_k$), where each $t_i, 1 \leq i \leq k$, is a distinct term in $W$. A transaction dataset $D$ = $\{T_1, T_2, ....., T_v\}$ is a set of transactions over $W$.

**Definition 1.** *($k^m$-anonymity).*

*If an adversary knows up to $m$ terms of a record, they cannot use this knowledge to identify less than $k$ candidate records in an anonymised dataset. In other words, the $k^m$-anonymity model guarantees that each combination of $m$ terms appears at least $k$ times in the anonymised dataset.*

For example, if an adversary knows that a person suffers from *cancer and diabetes* and this person's record is released in a $3^2$-anonymous dataset, then the adversary will not be able to identify this person's record form less than three records.

**Definition 2.** *(Disassociated transactions).*

*Let $(T_1, T_2, ....., T_v)$ be transactions in the original dataset $D$, and disassociation takes as an input dataset D and results in the anonymised dataset $\hat{D}$. The anonymised dataset $\hat{D}$ groups transactions in clusters $\hat{D} = (P_1, ....., P_z)$. Each cluster divides the terms of the transactions into a number of record chunks $(C_1, ....., C_n)$ and a term chunk $C_T$. The record chunks contain the terms in an itemset form called sub-record $(SR_1, SR_2, ....., SR_v)$ that satisfy $k^m$-anonymity, while the term chunk contains the rest of the terms of the transactions.*

## 3.2 Disassociation Method

The disassociation method performs three steps to anonymise a dataset: *horizontal partitioning, vertical partitioning* and *refining* (Figure 3.1).

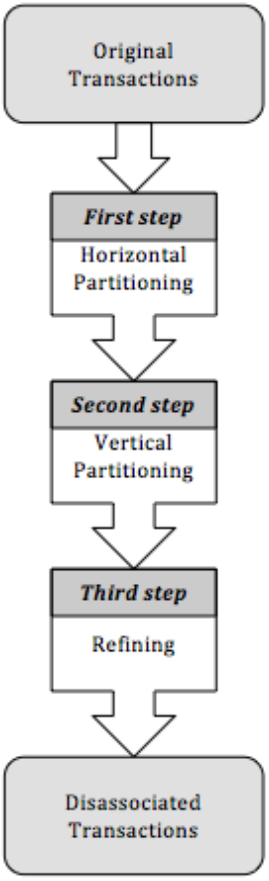



**Figure 3.1: The disassociation method**

First, horizontal partitioning groups transactions into clusters. After that, vertical partitioning separates infrequent term combinations in a cluster by placing them into different groups. Finally, the refining step is implemented to limit information loss and increase data utility.

### 3.2.1 Horizontal Partitioning

Horizontal partitioning is the first step of the disassociation method. In this step, transactions are separated into groups called clusters. Horizontal partitioning uses a recursive method to perform binary partitioning of the data into groups based on the frequency of term occurrence in the dataset. In other words, the algorithm finds the most frequent term and then uses it to divide the transactions into two groups: one for transactions with the term and the other for transactions without the term. After that, the method finds the next most frequent term for each group, and divides transactions again based on it. The partitioning process will be performed on each group of transactions until the clusters reach the required size which should not be smaller than $k$.

For example, if $k$ is equal to 2 and we need to horizontally divide the dataset in Table 3.1, the algorithm would picks *lung* as the most frequent term in the first iteration. Horizontal partitioning then divides the transactions into two clusters, the first cluster contains that the transactions have the term *lung* in them, and the second cluster the rest of the transactions as follows:

**Table 3.1: Horizontal partitioning (the first iteration)**

| ID | Transactions |
|---|---|
| *P1* | {cancer, radiotherapy, lung, treatment}<br>{cancer, lung, blood, tumor, biopsy}<br>{vessel, blood, treatment, lung, catheterisation} |
| *P2* | {cancer, blood, treatment, tumor, biopsy} |

Table 3.1 illustrates the result of the first iteration of horizontal partitioning. Although the cluster $P1$ satisfies the size condition, cluster $P2$ is smaller than $k$. As a result, horizontal partitioning will abandon the division returning $P2$ and $P1$ undivided as one

cluster (Table 3.2).

**Table 3.2: Horizontal partitioning (the resulting cluster)**

| ID | Transactions |
|---|---|
| *P1* | {cancer, radiotherapy, lung, treatment} |
| | {cancer, lung, blood, tumor, biopsy} |
| | {cancer, blood, treatment, tumor, biopsy} |
| | {vessel, blood, treatment, lung, catheterisation} |

The aim of the horizontal partitioning step is to minimise anonymisation transactions to the anonymisation of small and independent multiple groups of transactions by having as few transactions and as many similar terms as possible in a cluster. This will lead to less disassociation among the terms and enhance data utility. However, abandoning the dividing step due to one cluster being too small could produce large clusters. As shown in Table 3.2, all transactions have return in one cluster without any partitioning. This may affect the effectiveness of the horizontal partitioning step.

### 3.2.2 Vertical Partitioning

In general, this step leaves term combinations that occur frequently intact while separating terms that generate infrequent combinations. The purpose of vertical partitioning is to hide the links between the terms of infrequent combinations by disassociating them. So, after horizontal partitioning, the disassociation method performs vertical partitioning for each cluster. The clusters are divided vertically into two types of chunks: record and term chunks. The record chunks contain sub-records of the original transactions where the terms in these sub-records satisfy the $k^m$-anonymity condition. This means that each $m$-sized combination of terms needs to appear at least $k$ times in a

record chunk. The term chunks contain the rest of the terms that have not been placed in record chunks. Each cluster can have a number of record chunks but only one term chunk.

To illustrate the vertical partitioning step, let us consider the example in Table 3.2, if $m$ equals 2 and $k$ equals 2, then the terms of transactions will be disassociated into chunks, ensuring that all resulting record chunks are $2^2$-anonymous.

Table 3.3 shows the first iteration of vertical partitioning. It takes the first term *blood* and checks its support. If its support is equal to or larger than $k$, the vertical partitioning will extend it with the next term *treatment* to check $2^2$-anonymous and so on. In transactions 3 and 4, the terms *tumor* and *biopsy* create a $2^2$-anonymous sub-record, but both terms have not appeared enough with *lung* or *treatment*, so vertical partitioning moves them to the second record chunk. In addition, vertical partitioning moves any term that has not appeared in the record chunks to the term chunk. Therefore, *vessel, catheterisation* and *radiotherapy* are placed in the term chunk (Table 3.3).

**Table 3.3: Disassociation transactions**

| | ***Record Chunks*** | | ***Term Chunk*** |
|---|---|---|---|
| ***ID*** | ***C1*** | ***C2*** | ***CT*** |
| 1 | {blood, treatment, lung} | | vessel, catheterisation, radiotherapy |
| 2 | {cancer, lung, treatment} | {tumor, biopsy} | |
| 3 | {cancer, lung, blood} | {tumor, biopsy} | |
| 4 | {cancer, blood, treatment} | | |

The vertical partitioning step depends mainly on the $k^m$-anonymity condition to create record chunks. However, if we consider the second cluster, $P2$, in example 3.5, the original transaction for this cluster is illustrated in Table 3.4. Cluster $P2$ has three record

chunks, where $C2$ contains *surgery* and $C3$ contains *failure* and both record chunks satisfy the $k^m$-anonymity requirement. In the original transactions, the term *surgery* never appears with the term *failure*. Therefore, if vertical partitioning includes *surgery* and *failure* in the same record chunk, then the $k^m$-anonymity requirement will still be satisfied. However, the vertical partitioning in the original disassociation method does not provide a detailed explanation of how sub-records are chosen to create record chunks.

**Table 3.4: Original transactions**

| ID | Transactions |
|---|---|
| 1 | {kidney, infection, failure, sepsis} |
| 2 | {kidney, surgery, catheterisation} |
| 3 | {kidney, infection, surgery} |
| 4 | {infection, failure, dialysis} |

### 3.2.3 Refining

The aim of the refining step is to enhance the utility of published data while preserving anonymisation. This step targets term chunks and examines the possibility of reducing the number of terms in the term chunks by introducing *joint* clusters. These clusters will be created for every two adjacent clusters if they have shared terms; subsequently, they will be moved to a shared chunk. To illustrate this stage, we use an extended example below.

The refining step provides more flexibility by allowing different clusters to share record chunks. In Table 3.5, the term *catheterisation* has enough support in the original dataset but after horizontal partitioning, the term *catheterisation* does not pass the $k^m$-anonymity requirement. However, if we consider the support of *catheterisation* in both

**Table 3.5: Disassociated transactions without refining**

| *ID* | ***Record Chunks*** | | | ***Term Chunks*** |
|---|---|---|---|---|
| | *C1* | *C2* | *C3* | *CT* |
| *P1* | {cancer, lung, treatment}<br>{cancer, lung, blood}<br>{cancer, blood, treatment}<br>{blood, treatment, lung} | {tumor, biopsy}<br>{tumor, biopsy} | | vessel, catheterisation,<br>radiotherapy |
| *P2* | {kidney, infection}<br>{kidney}<br>{kidney, infection}<br>{infection} | {surgery}<br>{surgery} | {failure}<br>{failure} | catheterisation,<br>dialysis<br>sepsis |

clusters $P1$ and $P2$, it has sufficient support. As a result, the refining step create a joint cluster for $P1$ and $P2$, and *catheterisation* will be moved to a shared chunk in this joint cluster (Table 3.6).

## 3.3 Improving the Original Disassociation Algorithm

As illustrated in the previous section, the original method of disassociation has some limitations. In this section, we introduce three strategies for horizontal partitioning: *suppression, adding* and *remaining-list*. In addition, we illustrate two possible ways to perform vertical partitioning.

**Table 3.6: Disassociated transactions with refining**

| *Record Chunks* | | | *Term Chunks* | *Shared* |
|---|---|---|---|---|
| P1 Cluster | | | | |
| {cancer, lung, treatment}<br>{cancer, lung, blood}<br>{cancer, blood, treatment}<br>{blood, treatment, lung} | {tumor, biopsy}<br>{tumor, biopsy} | | vessel,<br>radiotherapy | {catheterisation} |
| P2 Cluster | | | | |
| {kidney, infection}<br>{kidney}<br>{kidney, infection}<br>{infection} | {surgery}<br>{surgery} | {failure}<br>{failure} | dialysis,<br>sepsis | {catheterisation} |

### 3.3.1 Horizontal Partitioning

To address the issue of cluster size in the original disassociation algorithm, we develop our new algorithms and introduce an improved solution with three new strategies to handle the clusters with a size smaller than $k$. In our algorithms, we add a check step to horizontal partitioning. The check step uses two parameters: the max cluster size and the $k$ value. The max cluster size determines the largest size of a cluster that requires no further partitioning where the $k$ value will be used to determine the smallest size acceptable for a cluster. There are three possible cases that need to be considered for cluster size:

1. As we analysed previously, the termination condition of the recursive partitioning of a dataset into a cluster depends on two parameters: the max cluster size and $k$. Therefore, if the cluster size is larger than the max cluster size, then the cluster will continue to be partitioned.

2. If the cluster size is between the max cluster size and the $k$ value, then no further partitioning is required and it will go through vertical partitioning in the next step.

3. If the cluster size is smaller than the $k$ value, we will not simply abandon the partitioning but will apply one of following strategies:

**Suppression Strategy**

In this strategy, clusters with a size smaller than $k$ are not released in the published disassociated dataset. To disassociate transactions in the vertical partitioning step, an itemset needs to appear at least in $k$ number of transactions to satisfy the $k^m$-anonymity requirement. Transactions in a small-sized cluster as not having enough frequent itemsets and all the terms will end up in the term chunk. This means that these transactions may not be beneficial to data analysis, so the strategy will suppress them from the published data.

Algorithm 1 shows how the suppression strategy performs for transactions. The algorithm first check the size of a cluster (Line 4). If the size is less than the max cluster size, checks if the size of a cluster is greater than or equal to $k$ (Line 5). If a cluster is smaller than the $k$ value, then it will be deleted (Line 6). Otherwise, the algorithm uses the most frequent term to divide a cluster into two groups: one with the records containing the most frequent term and one with the remaining records (Lines 11 to 14). These steps are recursively applied to each cluster in the cluster queue $Q$ until all the clusters have a suitable size. Terms previously used for partitioning are saved in the set $ignore$ and will not be used in subsequent splitting (Line 12).

**Algorithm 1:** Suppression method

**Input:** Dataset D,$MaxClusterSize$, $k$

**Output:** Horizontal partitioning of D

$ignore \leftarrow \{\}$ , $Q \leftarrow D$;
**while** $Q \neq \{\}$ **do**
    {D}$\leftarrow$ head(Q);
    **if** $|D| < MaxClusterSize$ **then**
        **if** $|D| < k$ **then**
            Delete $D$
            **else**
            Save D
        **end**
        **else**
        $T \leftarrow$ be the set of terms of D
        Find the most frequent term x in (T $-$ ignore)
        $D1 \leftarrow$all records of D having term x
        $D2 \leftarrow D - D1$
    **end**
    **return** *(D1; ignore* $\cup$ *x) and (D2; ignore) to* $Q$
**end**
**return** *Horizontal partitioning of D*

### Adding Strategy

Instead of suppressing small clusters or abandoning the partitioning, this strategy distributes small clusters to other clusters in the cluster queue. this is because the infrequent terms in one small cluster can be frequent in other large clusters. Hence, this strategy adds small clusters to another clusters to increase the frequency of terms. This will avoid the issue of partitioning being abandoned in the original algorithm.

In Algorithm 2 of the adding strategy, the same cluster size checking as in suppression is carried out. If the cluster size is between the max cluster size and $k$ value, then the cluster does not need more partitioning (Line 12). If the size is less than $k$, then the small cluster will be added to the second cluster in the clusters queue $Q$ (line 7). However, if there are no more clusters in the cluster queue, then the small cluster will be added to the last cluster in the queue (Line 9). The algorithm uses the most frequent term to divide a cluster into two clusters (Lines 15 to 18). these steps are recursively applied to each cluster in the cluster queue $Q$ until all the clusters have a suitable size. At the end, the horizontal partitioning of the dataset is returned.

**Algorithm 2:** Adding method

**Input:** Dataset D,$MaxClusterSize$, $k$

**Output:** Horizontal partitioning of D

$ignore \leftarrow \{\}$ , $Q \leftarrow D$
**while** $Q \neq \{\}$ **do**
    {D}$\leftarrow$ head(Q)
    **if** $|D| < MaxClusterSize$ **then**
        **if** $|D| < k$ **then**
            **if** $|Q| > 1$ **then**
                Add {D} to the second {D} on $Q$
                **else**
                Add {D} to the last {D} on $Q$
            **end**
            **else**
            Save D
        **end**
        **else**
        $T \leftarrow$ be the set of terms of D
        Find the most frequent term x in (T $-$ ignore)
        $D1 \leftarrow$all records of D having term x
        $D2 \leftarrow D - D1$
    **end**
    **return** *(D1; ignore* $\cup$ *x), (D2; ignore) to* $Q$
**end**
**return** *Horizontal partitioning of D*

**Remaining List Strategy**

This strategy creates a list called the 'remaining list'. After checking the size of a cluster, the remaining list strategy moves all the clusters with a size less than $k$ to the remaining list. After reaching the end of the horizontal partitioning of all transactions, the remaining list will be moved back as a one big cluster, and the horizontal partitioning will be applied to it again.

Algorithm 3 shows how the remaining list strategy performs for transactions. The remaining list $L$ is created as a first step in this strategy (Line 1) and the remaining list will be moved to the clusters queue (Line 3). Then, the algorithm checks the size of a cluster as before (Lines 5 and 6). If the size of a cluster is less than $k$, then the cluster will be moved to the remaining list (Line 7). Otherwise, the algorithm uses the most frequent term to divide the large cluster into two smaller clusters. After horizontal partitioning all the clusters in the cluster queue, the remaining list will be moved to the cluster queue and the process is repeated. However, if the size of the remaining list $L$ is less than $k$, then the transactions in the remaining list will be added to the last cluster (Line 21).

**Algorithm 3:** Remaining list method

**Input:** Dataset D,$MaxClusterSize$, $k$

**Output:** Horizontal partitioning of D

```
1  ignore ← {} , Q ← D, L ← Q
2  while L ≠ {} do
3      if L > k then
4          Q ← L L ← {} while Q ≠ {} do
5              {D}← head(Q)
6              if |D| < MaxClusterSize then
7                  if |D| < k then
8                      L ← L ∪ {D}
9                      else
10                     Save D
11                 end
12                 else
13                 T ← be the set of terms of D
14                 Find the most frequent term x in (T − ignore)
15                 D1 ←all records of D having term x
16                 D2 ← D − D1
17             end
18             return (D1; ignore ∪ x), (D2; ignore) to Q
19         end
20     end
21     else
22     tail(Q) ← tail(Q) ∪ L
23 end
24 return Horizontal partitioning of D
```

### 3.3.2 Hierarchical Clustering (HC) for Horizontal Partitioning

Hierarchical clustering is a clustering approach aiming to group objects into clusters based on their similarity. There are two types of hierarchical clustering strategy: agglomerative strategy and divisive strategy. Agglomerative strategy follows the bottom-up approach that starts with many little clusters and then combines them to form larger clusters. Divisive strategy uses top-down approach. It starts with one large cluster and then divides it into smaller clusters.

However, to determine which clusters should be joined (for agglomerative) or which cluster should be divided (for divisive), HC needs to use an appropriate metric to measure the similarity such as Euclidean distance, between objects. However, in the disassociation method, the horizontal partitioning needs to be based on the frequency of terms to achieve the $k^m$-anonymity requirement. So, to be able to use the HC approach for horizontal partitioning, the similarity metric should be based on the frequency of $m$-itemsets to form clusters, not the distance between transactions.

### 3.3.3 Vertical Partitioning

Vertical partitioning aims to divide each cluster into record and term chunks. Thus, it groups terms that frequently occur in one record chunk and separates infrequent combinations over the record chunks based on the $k^m$-anonymity condition. In vertical partitioning the support of terms in a $m$ combination of sub-records is not illustrated in detail in the original disassociation algorithm. Hence, we illustrated two possible ways to apply the $k^m$-anonymity condition:

- In a cluster, when disassociating terms of transactions into record chunks, each group of $m$ terms need to satisfy the $k^m$-anonymity condition. So, to create a

record chunk, all possible $m$ combinations of terms of one group need to appear at least $k$ times in a cluster to place them in the same record chunk. In other words, all combinations of sub-records' terms in one record chunk need to satisfied the $k^m$-anonymity condition. This method will produce more record chunks and may produce more empty sub-records inside record chunks. For example, if we have a group of terms *(surgery, failure)* from Table 3.4, first, the vertical partitioning checks the support of *surgery* then add *failure*. However, the itemset of *(surgery, failure)* is not appear $k$ times together. Therefore, the *failure* will not be added to the same record chunks with *surgery*.

The disassociated transaction in Table 3.7 follows this method. To disassociate transactions into record chunks, terms in cluster $P1$ are divided to three groups that satisfy the $k^m$-anonymity condition. The first group contains *kidney, infection*, the second group contains *surgery* and the third group contains *failure*. Each group creates a record chunk, therefore, there are three record chunks for cluster $P1$.

**Table 3.7: The first method of vertical partitioning**

| ***ID*** | ***Record chunks*** | | | ***Term chunk*** |
|---|---|---|---|---|
| | ***C1*** | ***C2*** | ***C3*** | ***CT*** |
| ***P1*** | {kidney, infection}<br>{kidney}<br>{kidney, infection}<br>{infection} | {surgery}<br>{surgery} | {failure}<br>{failure} | catheterisation<br>dialysis<br>sepsis |

- In a cluster, when disassociating terms of transactions into record chunks, each record chunk can have more than one group of terms. In other words, all sub-records from different groups of terms can be placed in the same record chunk if

all $m$ combinations in each group have already satisfied the $k^m$-anonymity condition. Taking this into account, there are no shared $m$ combinations between two different groups that have support less than $k$, then all the sub-records of the two groups can be placed into one record chunk. This will reduce the number of record chunks, allowing for more different sub-records to be in the same record chunk without violating the $k^m$-anonymity requirement. For example, if we have the two groups *surgery* and *failure* and both of them satisfy the $k^m$-anonymity requirement. Also there is no shared $m$ combinations between them, then they can share the same record chunk.

Table 3.8 illustrates the second method of vertical partitioning. As with the first method, the terms of cluster $P1$ will be separated into three groups that satisfy the $k^m$-anonymity condition. However, groups can share one record chunk if there are no shared $m$ combinations between them. Therefore, the second group *surgery* and the third group *failure* are placed in the same record chunk. This is because both groups satisfy the $k^m$-anonymity requirement and they do not appear together in the original transaction so there is no shared $m$ combinations between them. Therefore, vertical partitioning will include *surgery* and *failure* in the same record chunk, and cluster $P1$ will be disassociated to two record chunks instead of three.

**Table 3.8: The second method of vertical partitioning**

| ***ID*** | ***Record chunks*** | | ***Term chunk*** |
|---|---|---|---|
| | ***C1*** | ***C2*** | ***CT*** |
| ***P1*** | {kidney, infection}<br>{kidney}<br>{kidney, infection}<br>{infection} | {failure}<br>{surgery}<br>{surgery}<br>{failure} | catheterisation<br>dialysis<br>sepsis |

## 3.4 Comparison of Different Methods of Horizontal Partitioning

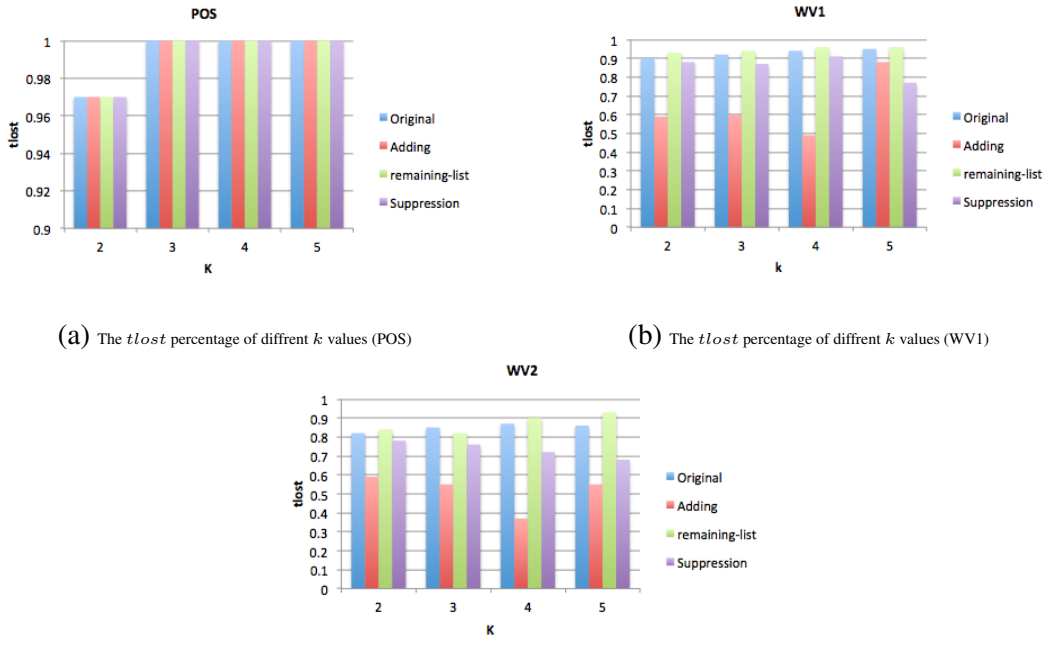
In this section, we compare our proposed methods for horizontal partitioning with the original method. Our methods aim to control the clusters size and avoid producing clusters with size larger than the max cluster size. This is because larger clusters may affect the utility of disassociated data. To evaluate the performance of our methods, we use three real datasets that were introduced in [117]. The first two datasets are WV1 and WV2, which contain click-stream data from two e-commerce websites that were collected over a period of many months. The third dataset is POS that is a transaction log from an electronics retailer.

We use $tlost$ as an evaluation metric to calculate the amount of information loss incurred by different methods. The $tlost$ metric gives the percentage of terms that support more than $k$ in the original dataset $D$ but they are placed in term chunks in the disassociated dataset.

Figure 5.21 illustrates the amount of information loss in the POS dataset over different $k$ values. However, all methods have the same $tlost$ percentages, which can be considered as high percentages, with different k values. This is because the level of frequency of terms is too high in this dataset. So the chance of moving a frequent term to the term chunk in some clusters is higher.

Figures 5.23 and 5.24 illustrate the amount of information loss in the WV1 and WV2 datasets over different $k$ values. We can see the effectiveness of the adding method in achieving the lowest $tlost$ percentages over all $k$ values while the remaining-list has similar percentages as the original method. The suppression method has a low percentage of information loss, but this is due to the removal of some terms from the

disassociated dataset.




(a) The $tlost$ percentage of diffrent $k$ values (POS)

(b) The $tlost$ percentage of diffrent $k$ values (WV1)

(c) The $tlost$ percentage of diffrent $k$ values (WV2)

**Figure 3.2: Comparison of the ratio of information loss for different methods**

## 3.5 Summary

In this chapter, we have described the original disassociation method. Also, we have given definitions of key concepts that are important to the understanding of this method. We have discussed the three steps of the disassociation method and explained the limitations that are associated with horizontal partitioning and vertical partitioning. One limitation of horizontal partitioning in the original disassociation method is that it abandons the dividing of a small cluster resulting from the partitioning and returns the cluster without any division. Abandoning the dividing could produce large clusters, leading to more disassociation among the terms, therefore diminishing the data utility.

To overcome this issue, we have introduced three strategies to handle the clusters with a size smaller than $k$: suppression, adding and remaining-list. The suppression strategy removes small clusters, while the adding strategy adds small clusters to another large cluster. The remaining-list strategy uses a list to hold all the small clusters that were the result of horizontal partitioning. Then, we applied horizontal partitioning again on this list. We illustrated how our proposed methods of horizontal partitioning achieved better $tlost$ level than the original method. This is because our methods provide a stronger control on max cluster size in disassociating a dataset. In other words, no horizontal partitioning will be abandoned in our methods; therefore, clusters will not exceed the allowed max cluster size as in the original disassociation method. Also, for the vertical partitioning, we illustrated two different methods to apply the $k^m$-anonymity condition on clusters to disassociate infrequent combinations into record chunks and term chunks.

# *Chapter 4*

# Semantic Attack

In this chapter, we propose our approach to semantic attacks on anonymised transaction produced by the disassociation method. We illustrate two types of semantic similarity measures and then explain how these measures are used in our attacking approach. Afterwards, we explain the semantic attack approach. We illustrate the two stages of this approach: scoring and selection. In the scoring stage, we explain how the semantic similarity measures are used to find semantic relationships between terms. In the selection stage, we propose four methods to exploit the semantic relationships to reconstruct original transactions.

## 4.1 Semantic Relationships

Semantics is the study of meaning and interpretation in a language. Semantics refer to the relationship between words and how humans derive meaning from these words. A semantic relation refers to any possible semantic relationship between the meaning of words, defining how two terms are related [6], [34]. For example, *gem* and *jewel* are related semantically because they share the same meaning, while *world* and *cup* do not share the same meaning, but they are related semantically in a specific context (*e.g. football competition*). In our semantic attack, we use this type of semantic relationship to reveal the hidden links between items in different chunks and reconstruct original

transactions.

### 4.1.1 Semantic Relatedness Measures

To measure the semantic relatedness between two terms, there are two major approaches that have been developed: the knowledge-based approach, which compares words by analysing the structured sources of information (ontologies), and the corpus-based approach, which compares words by analysing unstructured or semi-structured texts [40].

#### Knowledge-based approach

This approach uses an ontology to determine the semantic relationship between terms where the concepts (terms) within a domain are organised in a hierarchical way and the relationships between the terms are described in a specific number of relational descriptors [36]. For example, knowledge-based methods use WordNet (a lexical database) as an ontology to find the relationships between terms. The semantic relations in this database include synonyms, hypernyms, meronyms and antonymys [28], [67]. Figure 4.1 illustrates WordNet's ontology of *Eye disease*, where terms are semantically related by hyponymy. For example, *cortical cataract* and *nuclear cataract* have the same parent node of *cataract*, so they are considered to be more related than *Normal tension glaucoma* and *acute glaucoma*.

The knowledge-based approach depends on the availability of ontologies such as WordNet [28] and MeSH [44]. Although these ontologies have a huge number of entities, they may not cover all the subjects. Also, it can be time-consuming and challenging to build an ontology manually because doing so requires expert knowledge.

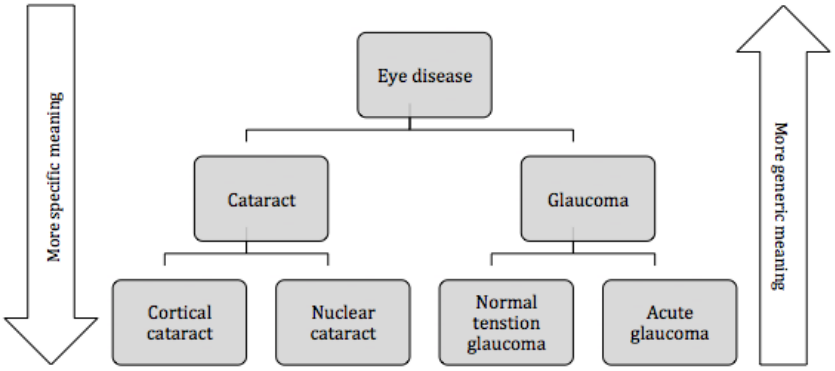



**Figure 4.1: A branch of WordNet's ontology for (eye disease)**

Moreover, a given term can have multiple definitions in an ontology, so finding the appropriate meaning can be complicated. For example, *bank* and *depository* are related if they are considered a storing place, but if *bank* is interpreted as the land along the edge of a river, then *bank* is not related to *depository*.

### Corpus-based approach

This approach uses a large collection of texts as a knowledge source; this is often referred to as a (corpora). The relationship between two terms can be determined by how often they occur in the same context in a corpora [37] [40]. For example, *cataract* is considered to be more related to *glaucoma* in a context about vision loss than other diseases such as *double vision*. This is because *cataract* frequently appears with *glaucoma* because both are serious eye diseases that can result in blindness. However, because this approach relies on the co-occurrence of information between terms, the size of a corpora will affect the measurement. The larger the corpora is, the more accurate the measurement will be.

### 4.1.2 Semantic Attack Measures

Transactions such as patient discharge reports, where each transaction contains information about patient's diseases and symptoms, can be protected by disassociating the information into chunks to preserve patient's privacy. However, an adversary can still associate symptoms that commonly occur together or symptoms with diseases because they would naturally appear together. This type of semantic relationship between a symptom and disease is usually not covered by ontologies such as synonyms or hypernyms because this relationship depends on how often a symptom and a disease appear together. So in our work, we argue that an attacker can benefit from the fact that two terms are related or have a semantic relationship, even if they do not share a similar meaning. Hence, we followed the corpus-based approach to measure the semantic relatedness between terms in disassociated transactions where the relationship is defined by the likelihood of their co-occurrence. We adopt two methods from the corpus-based approach: normalised Google distance (NGD) and word embeddings (WE). The next section will explain how these methods have been used in our semantic attack approach.

#### Word Embedding (WE)

WE is a type of latent representation of words where similar words have similar representations. In other words, it is the process of taking a large corpus of words and mapping each word to a vector in a vector space where similar words are assigned in nearby points (vectors). Being able to successfully do this depends on training a neural network on a chosen dataset in which two words are considered similar if they appear in a similar context; then, this trained neural network can be used to determine the relatedness of new input words. There have been different neural embedding techniques developed, but the most popular techniques are Word2vec and GloVe [76], [5].

Word2vec relies on assuming that the context in which a word apears can be used to infer the meaning of this word effectively. There are two models to learn word embedding in Word2vec: continuous bag-of-words (CBOW) model and continuous Skip-Gram model. CBOW predicts a target word based on its context, while the Skip-Gram model predicts the target context words given a current word [66]. The Skip-Gram model is better for training a small amount of data and infrequent words, while CBOW is faster and more accurate for frequent words. Figure 4.2 illustrates the training architectures of these two models.

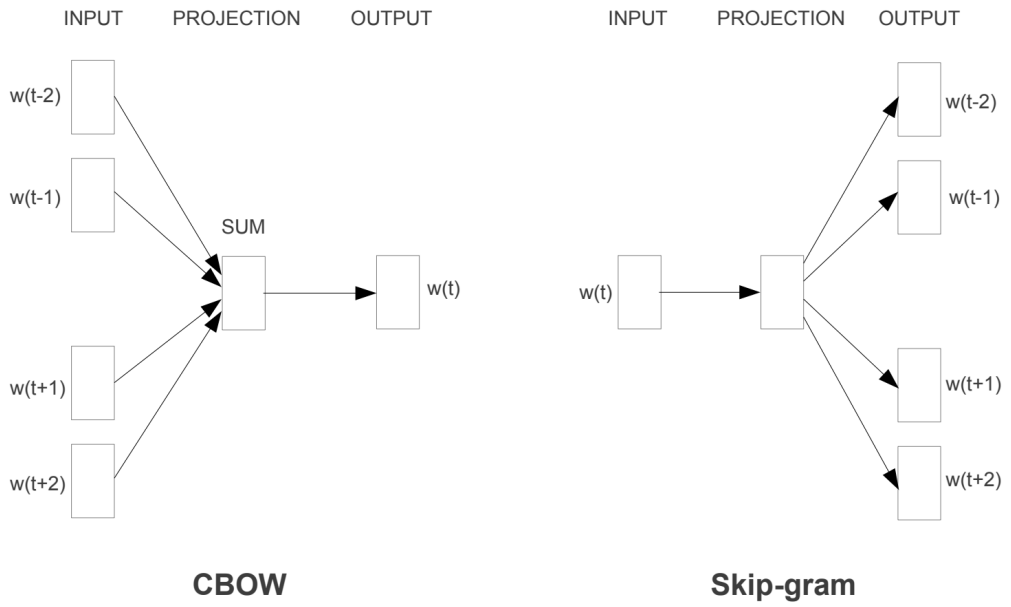



**Figure 4.2: Word2Vec training models [66]**

The other technique used for WE is GloVe. It learns WE by using a global matrix factorisation method (word-word co-occurrence matrix) [76]. In this matrix, each row presents a word and each column a context, and this matrix uses statistics to count the co-occurrence between the word and context across the whole corpus.

In our work, we use WE to measure the semantic similarity between the terms in dif-

ferent chunks. One advantage of using this method is that there are many pre-trained models that have been developed by researchers and that can be used in different domains; hence, they are available for use in any project. This means that there is no need to build a model from the scratch. Indeed, using word embedding to find semantic relationships is fast and simple, which makes it a reasonable measure to be used by an attacker. In our approach, we used pre-trained Wikipedia GloVe embeddings trained by the GloVe model [76]. The corpus has 400k unique words in the vocabulary from a snapshot of English Wikipedia in 2014 and English Gigaword fifth edition [1].

The main limitation of using WE is the out-of-vocabulary words issue. If a word is not included in the training phase of an embedding model, then the model will not be able to interpret this word or know how to assign a vector to it [4]. Also, the effectiveness of this method depends on the domains covered in the corpus.

**Normalised Google Distance (NGD)**

The second corpus-based method in our work uses the entire World Wide Web as a corpus [15]. The semantic relationship between terms is measured based on how frequently these terms are used in the same page on the web. Unlike word embedding, NGD uses a Google repository that contains numerous domains; hence, there is no need to build a customised corpus or update it to include new terms.

This method measures the semantic similarity based on the number of pages that are returned by Google's search engine. The theory behind NGD is that if two terms have a semantic relationship, then the likelihood of both terms appearing together on a large number of pages will be high [50]. In other words, similar words that frequently occur together tend to be close in Google distance, while dissimilar words tend to be farther

[1] `https://nlp.stanford.edu/projects/glove/`

apart. To find the semantic relationship between the two words $x$ and $y$, the following formula will be used:

$$NGD(x,y) = \frac{max\{\log f(x), \log f(y)\} - \log f(x,y)}{\log N - min\{\log f(x), \log f(y)\}}, \tag{4.1}$$

In Equation 4.1, $N$ is the total number of web pages searched by Google; $f(x)$ is the number of web pages with word $x$ and $f(y)$ the number of web pages with word $y$; and $f(x,y)$ is the number of web pages with both $x$ and $y$. If $x$ and $y$ occur separately but never appear together on the same web page, then the Google distance between $x$ and $y$ is infinite. If $x$ and $y$ always occur together, then $x$ and $y$ are viewed as similar as possible and their Google distance equals zero.

For example, to find the similarity between *glaucoma* and *cataract*, and *glaucoma* and *double vision*, we use Google's search engine to find the number of pages or documents that contain each term. Hence, by applying Equation 4.1 to *glaucoma* and *cataract*, $f(x)$ and $f(y)$ are the number of pages for *glaucoma* and *cataract*, and $f(x,y)$ is the number of web pages on which both *glaucoma* and *cataract* occur together. So, for example, take $f('glaucoma') = 67400000$, as shown in Figure 4.3. The result of the NGD for both pairs is as follows:

$$NGD(glaucoma, double\_vision) > NGD(glaucoma, cataract) \tag{4.2}$$

where $NGD(glaucoma, cataract) = 0.18$ and $NGD(glaucoma, double\_vision) = 0.32$, which suggests that in general, *glaucoma* is more related to *cataract* than to *double vision*.

However, unlike the approaches that use a local corpus to calculate semantic relationships, an NGD query to a Google repository is relatively slow and limited for a vary large-scale dataset. This is because Google's search engine only allows for a limited

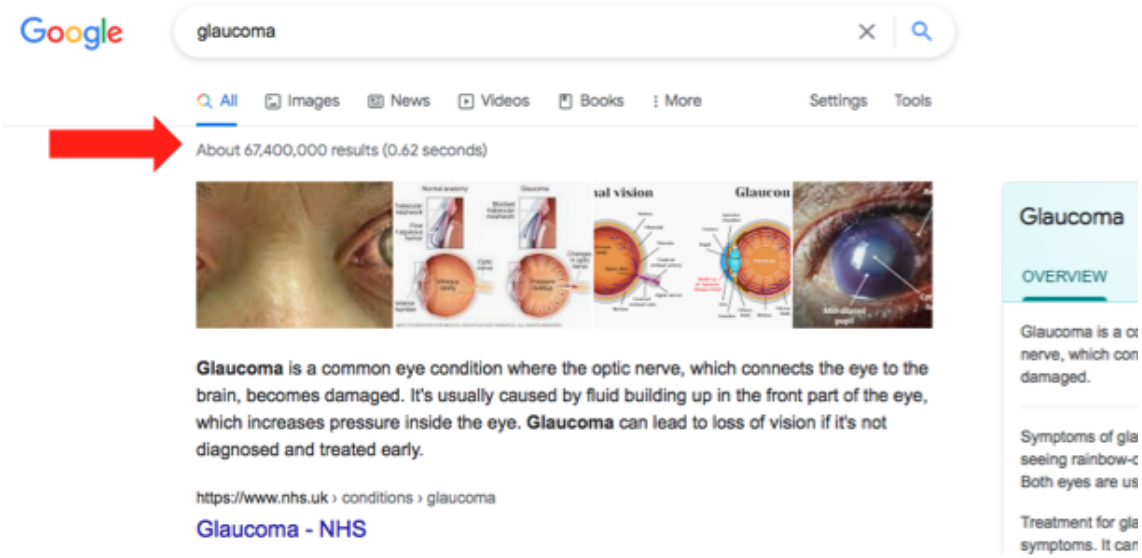



**Figure 4.3: An example of *f ('glaucoma')* by Google**

number of queries for a given period of time.

### 4.1.3 The Limitations of NGD and WE

The WE measurement calculates the semantic relationships based on semantic vectors that can be faster than the NGD measurement. However, the major limitations in WE are that it may not cover all the terms in the dataset and there may be a need to train your own word embedding on the original transactions to include all the terms that will be not available in the real attack for an attacker. In contrast, the NGD measurement does not have this problem, because it uses the Google repository. However, a limitation in NGD is that because the Google repository is continuously updated, the relationships between the terms can change. However, given the size of Google repository, such change will not affect the calculation of semantic score. For example, the relationship between the term *fever* and the term *isolation* has been changed recently, and their co-occurrence in the same page on the web is increased due to the coronavirus. The

NGD score for *fever* and *isolation* is 0.33, which is considered semantically related. However, this change in the relationship between fever and isolation does not affect the semantic relationship between *fever* and other terms, e.g. *flu*. Hence *fever* and *flu* are still semantically related and frequently appear in the same page on the web with NGD score equal to 0.34.

### 4.1.4 Accuracy of Measures

In our attacking approach, we mainly rely on the semantic relationships among the terms in different chunks to reconstruct the original transactions. Although there are other methods that use the corpus-based approach, such as Li's method [56] and Islam's method [46], NGD and WE are more available for attackers. In addition, WE is efficient, while NGD covers various domains. Therefore, in this section, we illustrate the accuracy of these two measures in identifying semantic relationships among the terms. We use a well-established benchmark to test the accuracy of WE and NGD, comparing them with other related methods for measuring semantic relationships.

#### Correlation between measures

Our test is based on a comparison between the semantic relationships produced by Li's method [56], Islam's method [46], WE [5], NGD [15] and human judgement. Li's method was developed to measure semantic similarity between sentences or very short texts based on semantic and word order information implied in the sentences. In Li's method, the semantic similarity is derived from the lexical knowledge base WordNet [67] and corpus statistics [56]. In Islam's method, the similarity of two short texts is determined using a combination of semantic and syntactic information. This method considered two main functions (string similarity and semantic word similarity) as well

as an optional function (common-word order similarity). Both Li and Islam used human judgment of 30 sentence pairs in their works to evaluate their similarity measures. So, to conduct this comparison, we use the same human judgment sets from the study by Miller and Charles [68] with 30 word pairs. The 30 pairs were extracted from the 65 pairs given in Rubenstein and Goodenough [82], which include synonymy pairs (e.g., boy : lad) and completely unrelated pairs (e.g., cord : smile). Human subjects evaluated each pair in this set using a scale from 0 to 1, giving these pairs semantic similarity scores. We use the set as a benchmark and calculate the NGD and WE for the same 30 pairs. For Li's method and Islam's method, we use their resulting finds of comparing their methods with the same human judgement set in their papers. After doing this, we calculate the correlation by using $COV(X;Y) = \frac{\sum_{i=1}^{n}(X_i-\overline{X})(Y_i-\overline{Y})}{n-1}$, where $X$ and $Y$ are the measurement result and the human estimation, and $\overline{X}$ and $\overline{Y}$ are their means. $n$ is the total number of pairs in this testing comparison. Table 4.1 shows the comparison of the different methods used for measurement.

Figures 4.4 illustrates the comparison between human judgement and Islam's method, Li's method, NGD and WE. All methods, except for NGD, have the same scale. Therefore, we convert NGD to NGD' to have a scale from 0 to 1 by using $1 - \frac{NGD}{MAX(NGD)}$, where a higher NGD score means a stronger relationship between terms.

Table 4.1 shows that WE has the highest correlation by 0.75 and that NGD correlates with human estimates by 0.72. Although Li's method has a close correlation to NGD, NGD uses the Google repository as a corpus, so it does not need to prepare or train any corpus to perform similarity measures. As a result, we consider NGD and WE to be reliable methods for determining the semantic relationships between terms. Although, the scores for NGD and WE do not exactly match human scores, they still indicate whether or not two terms are related. In Figure 4.4, in the comparisons of NGD and WE with human judgement, all pairs with ID 10 and a higher have higher similarity

than the pairs below ID 10. Although there are still small errors when measuring the relationships in WE and NGD, we consider these methods to have a reasonable accuracy for our attacking approach.

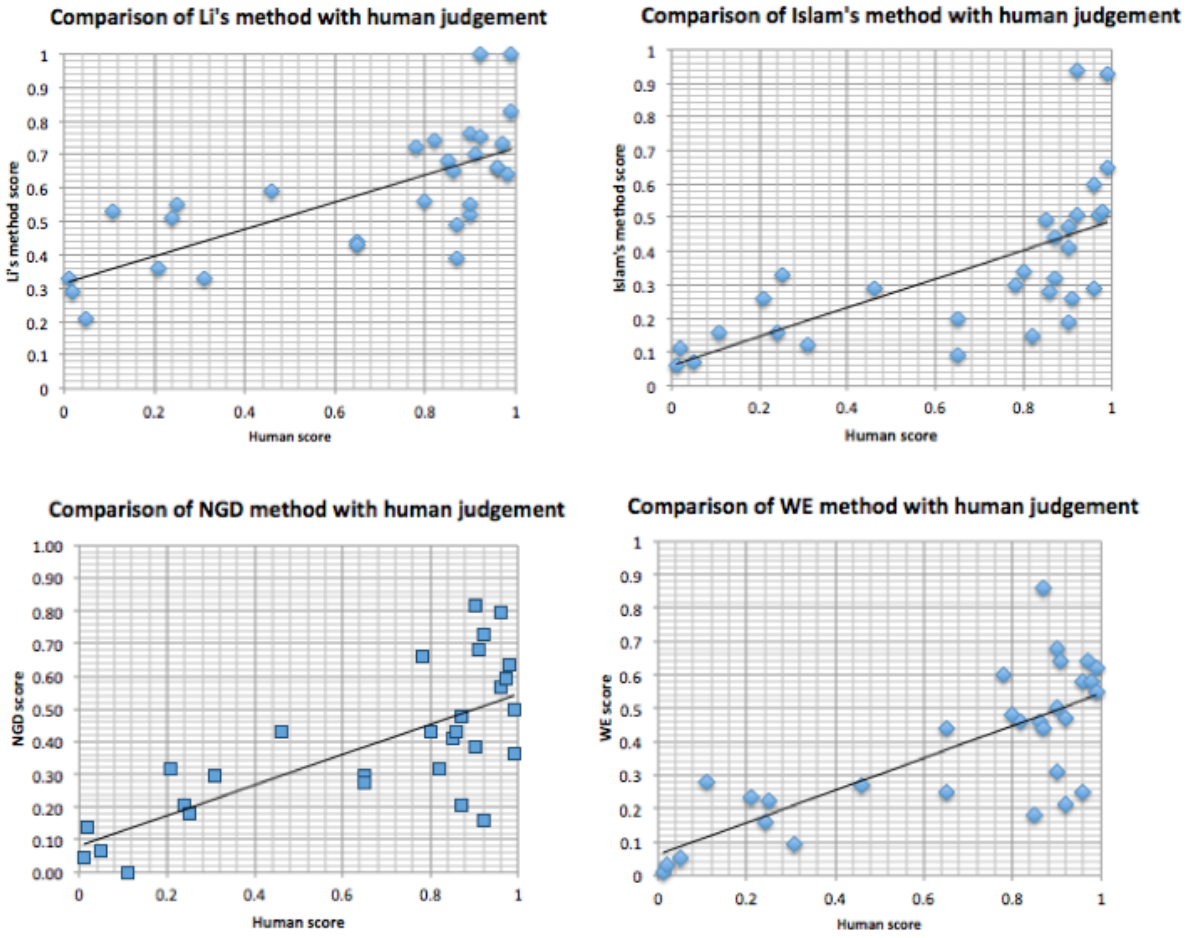



**Figure 4.4: Different semantic measurements comparison**

**Table 4.1: Methods comparison**

| | Word pair | Human | NGD' | Word Embedding | LI [56] | islam [46] |
|---|---|---|---|---|---|---|
| 1 | cord-smile | 0.01 | 0.05 | 0.01 | 0.33 | 0.06 |
| 2 | autograph-shore | 0.02 | 0.14 | 0.03 | 0.29 | 0.11 |
| 3 | asylum-fruit | 0.05 | 0.07 | 0.05 | 0.21 | 0.07 |
| 4 | boy-rooster | 0.11 | 0.00 | 0.28 | 0.53 | 0.16 |
| 5 | coast-forest | 0.21 | 0.32 | 0.23 | 0.36 | 0.26 |
| 6 | boy-sage | 0.24 | 0.20 | 0.16 | 0.51 | 0.16 |
| 7 | forest-graveyard | 0.25 | 0.18 | 0.22 | 0.55 | 0.33 |
| 8 | bird-woodland | 0.31 | 0.30 | 0.09 | 0.33 | 0.12 |
| 9 | hill-woodland | 0.46 | 0.43 | 0.27 | 0.59 | 0.29 |
| 10 | magician-oracle | 0.65 | 0.30 | 0.25 | 0.44 | 0.2 |
| 11 | oracle-sage | 0.65 | 0.27 | 0.44 | 0.43 | 0.09 |
| 12 | furnace-stove | 0.78 | 0.66 | 0.6 | 0.72 | 0.3 |
| 13 | magician-wizard | 0.8 | 0.43 | 0.48 | 0.56 | 0.34 |
| 14 | hill-mound | 0.82 | 0.32 | 0.46 | 0.74 | 0.15 |
| 15 | cord-string | 0.85 | 0.41 | 0.18 | 0.68 | 0.49 |
| 16 | glass-tumbler | 0.86 | 0.43 | 0.46 | 0.65 | 0.28 |
| 17 | grin-smile | 0.87 | 0.48 | 0.86 | 0.49 | 0.32 |
| 18 | serf-slave | 0.87 | 0.20 | 0.44 | 0.39 | 0.44 |
| 19 | journey-voyage | 0.9 | 0.39 | 0.68 | 0.52 | 0.41 |
| 20 | autograph-signature | 0.9 | 0.39 | 0.31 | 0.55 | 0.19 |
| 21 | coast-shore | 0.9 | 0.82 | 0.5 | 0.76 | 0.47 |
| 22 | forest-woodland | 0.91 | 0.68 | 0.64 | 0.7 | 0.26 |
| 23 | implement-tool | 0.92 | 0.73 | 0.21 | 0.75 | 0.51 |
| 24 | cock-rooster | 0.92 | 0.16 | 0.47 | 1 | 0.94 |
| 25 | boy-lad | 0.96 | 0.57 | 0.58 | 0.66 | 0.6 |
| 26 | cushion-pillow | 0.96 | 0.80 | 0.25 | 0.66 | 0.29 |
| 27 | cemetery-graveyard | 0.97 | 0.59 | 0.64 | 0.73 | 0.51 |
| 28 | automobile-car | 0.98 | 0.64 | 0.58 | 0.64 | 0.52 |
| 29 | midday-noon | 0.99 | 0.36 | 0.55 | 1 | 0.93 |
| 30 | gem-jewel | 0.99 | 0.50 | 0.62 | 0.83 | 0.65 |
| | | **Correlation** | **0.72** | **0.75** | **0.72** | **0.65** |

## 4.2 Semantic Attack Approach

Our semantic attack approach consists of two stages. The first stage is to find the semantic relationships among the terms in a disassociated dataset, which is called *scoring*. The second stage uses these semantic relationships to determine which terms should be associated for reconstructing the original transactions, which is called *selection*. The input dataset of our semantic attack is the transactions anonymised by the disassociation method.

### 4.2.1 Scoring step:

In the first step of our semantic attack, we use two measures, NGD and WE, to find the semantic relationship scores. In our approach, we use the sub-records in the first record chunk as a basis to establish the semantic relationships, and we refer to this chunk as an *anchoring chunk* in our work. Therefore, in each cluster, this step finds the semantic relationship scores between disassociated terms in other chunks and terms in the anchoring chunk.

The pseudocode for the scoring step is provided in Algorithm 4. The algorithm is performed for each cluster $P$ in the disassociated dataset $\hat{D}$ (Step 1). There are two different types of chunks in a disassociated dataset: record chunks ($C_1$,$C_2$,...,$C_n$) and a term chunk ($C_T$), as shown in Table 4.3, which shows the $2^2$-anonymous disassociated transactions for the original transactions in Table4.2. In a cluster, each record chunk contains a number of sub-records ($SR_1, SR_2, ....., SR_v$), and the term chunk contains terms ($t_1$,$t_2$,....,$t_k$).

In Algorithm 4, for each sub-record $SR$ in the record chunks from $C_2$ to $C_n$ or for each term $t_i$ in the term chunk $C_T$, the algorithm uses NGD or WE to calculate the semantic

**Table 4.2: Original transactions**

| ID | Transactions |
|---|---|
| 1 | {vessel, blood, treatment, lung, catheterisation} |
| 2 | {cancer, radiotherapy, lung, treatment} |
| 3 | {cancer, lung, blood, tumor, biopsy} |
| 4 | {cancer, blood, treatment, tumor, biopsy} |

**Table 4.3: Disassociated transactions**

| | ***Record Chunks*** | | ***Term Chunk*** |
|---|---|---|---|
| ***ID*** | ***C1*** | ***C2*** | ***CT*** |
| 1 | {blood, treatment, lung} | | vessel, catheterisation, radiotherapy |
| 2 | {cancer, lung, treatment} | {tumor, biopsy} | |
| 3 | {cancer, lung, blood} | {tumor, biopsy} | |
| 4 | {cancer, blood, treatment} | | |

relationship between pairwise of $SR$ or $t_i$ and each sub-record $ASR$ in the anchoring chunk ($C_1$) (steps 2 and 3). In steps 5 and 6, the algorithm calculates the semantic scores for each term in the term chunks. All the semantics scores between disassociated sub-records or terms and all sub-records in the anchoring chunk in a cluster are stored in $scores_p$ to use them in the next step (step 8). For example, for Table4.3, the first pairwise consists of *(tumor, biopsy)* from $C_2$ and *(blood, treatment, lung)* from the anchoring chunk in step 2. If WE is chosen as the semantic measure in step 3, the semantics scores for this pairwise would be [0.47, 0.61, 0.84]. This step will be repeated for the other three sub-records in $C_1$. Then, in steps 5 and 6, the algorithm will find the semantics scores between each term in $C_T$ *vessel, catheterisation, radiotherapy* and the four sub-records in $C_1$ . All the resulting scores will be stored in $scores_p$.

**Algorithm 4:** Scoring Step

**Input:** Disassociated transactions

**Output:** Semantic relationships scores

**for** *each cluster* $P$ **do**
  **for** *each sub-record* $SR$ *in* ($C_2$ *to* $C_n$) **do**
    Calculate the semantic score between $SR$ and all sub-records in $C_1$ by NGD / WE
  **end**
  **for** *each term* $t_i$ *in* $C_T$ **do**
    Calculate the semantic score between $t_i$ and all sub-records in $C_1$
  **end**
  $scores_p$= semantic relationships scores for sub-records and terms in a cluster $P$
**end**
**return** *Semantic relationships scores of the disassociated transactions*

### 4.2.2 Selection step:

The disassociation method protects the terms of the transactions by dividing them into record and term chunks. For each disassociated sub-record or term, this step aims to find their original transaction and reassociate the terms and sub-records based on the semantic relationships that exist among them. Hence, having derived all the semantic relationship scores in the scoring step, the selection step determines how the disassociated terms and sub-records are combined with the sub-record in the anchoring chunk to reconstruct transactions. We propose four methods for this step, and these methods will be discussed in detail in the following section. Note that the sub-records in the anchoring chunk are the first parts of the transactions, and these sub-records usually contain the largest sub-record of each disassociated transaction. Therefore, these large sub-records can be used to provide more semantic information for the reconstruction. For example, in Table 4.3, $C_1$ is the anchoring chunk that contains the largest sub-

records in the cluster. Each sub-record *(e.g., blood, treatment, lung)* is an incomplete transaction, so there is a need to find its disassociated sub-records or terms in $C_2$ and $C_T$ to reconstruct the original transaction.

Algorithm 5 shows how the selection step is performed. Having determined the method that will be used in the selection step for both record chunks and the term chunk, the algorithm is applied on each cluster $P$ in the disassociated dataset $\hat{D}$. For each record chunk from $C_2$ to $C_n$ in a cluster $P$, the attacking method is preformed for each sub-record $RS_i$ in a record chunk (steps 3 and 4). The attacking method is then executed to find the best related $ASR_i$ in $C_1$ to the current $RS_i$ and return it (step 5), where $ASR_{i1}$ is the chosen sub-record that the sub-record $RS_i$ will be appended to. For example, Table 4.3, describes a set of medical diagnoses and treatments that belong to a patient. The attacking method will be applied to the sub-record *tumor, biopsy* $SR$ to return the best related $ASR_i$ in the four sub-records in $C_1$. After that, the sub-record in $C_1$ will be updated by adding the sub-record $RS_i$ (step 6). For the term chunk $C_T$ in a cluster $P$, the algorithm performs the selection step in one term for each iteration (step 10). In step 12, the chosen sub-record $ASR_i$ from $C_1$ will be returned in order to add the term to it (step 12). After all the terms in the term chunk have been processed, the transactions are deemed to be reconstructed and the algorithm will store the reconstructed cluster in $Rec_P$ (step 14). The algorithm will return the reconstructed transactions of the disassociated transactions in step 10.

**Algorithm 5:** Selection Step

**Input:** Disassociated transactions, Semantic relationships scores

**Output:** Reconstructed transactions

**for** *each cluster* $P$ **do**
  **for** *each record chunk in (*$C_2$ *to* $C_n$*)* **do**
    **for** *each sub-record* $SR_i$ *in a record chunk* **do**
      Execute an attacking method
      $ASR_i$, $RS_i$ = attacking method($SR_i$)
      Update $ASR_i$ in $C_1$
    **end**
  **end**
  **for** *each Term* $t_j$ *in* $C_T$ **do**
    Execute an attacking method
    $ASR_i$, $t_i$ = attacking method($t_i$)
    Update $ASR_i$ in $C_1$
  **end**
  $Rec_P =$ The reconstructed transactions of a cluster $P$
**end**
**return** *The reconstructed transactions of the disassociated transactions*

## 4.3 Attacking methods

In this section, we illustrate how both types of chunks; record and term chunks, can be attacked. We propose four strategies to use semantic scores to associate terms and sub-records in disassociated transactions to reconstruct the original transactions. The strategies are : averaging-based attack (ABA), most-related attack (MRA), related-group attack (RGA) and vertical partitioning attack (VPA). The ABA, MRA and RGA utilise the semantic relationships scores to accomplish the attack, while the VPA employs the vertical partitioning from the disassociation method to attack the disassociated transactions.

In disassociated transactions, each sub-record $ASR_i$ in the anchoring chunk needs to be completed by combining its terms from other chunks to reconstruct the original transaction. Hence, the terms in the sub-records in the anchoring chunk are used as a base to reassemble the transactions. In the following, we explain how the record and term chunks will be attacked.

- ***Attacking record chunks***

  In a cluster, each record chunk has at least $k$ sub-records that satisfy the $k^m$-anonymity requirement. Hence, to break this requirement, different sub-records need to be reassociated to find the protected $m$ combinations of terms that have support of less than $k$. In general, to perform the attack on record chunks, the scoring step is executed first for each cluster $P$ in the dataset, where a semantic relationship calculation is performed on the anchoring chunk $C_1$ and a chunk from $C_2$ to $C_n$. After that, the selection step is applied. To attack record chunks, only the ABA and VPA strategies are used. This is because the sub-records in record chunks usually have more than one term. Each term in one sub-record could have different levels of semantic relatedness with another sub-record. Therefore,

using the MRA and RGA strategies may not capture the semantic score properly between two sub-records. As a result, the MRA and RGA strategies are not used in attacking record chunks.

- ***Attacking term chunk***

  Unlike record chunks, the term chunk of a cluster contains single terms, not sub-records. These terms have support of less than $k$, and they are protected by placing them in the term chunk so that no terms can be linked to fewer transactions than the size of the cluster. To perform the attack on the term chunk, the scoring step is first executed for each cluster $P$ in the disassociated dataset between each term in the term chunk and all sub-records in the anchoring chunk. After that, the selection step is applied. For term chunks that are attacked, all strategies are used except for the VPA strategy. This is because each term in the term chunk never needs to be a part of any record chunks, so, the VPA strategy is not useful.

### 4.3.1 Averaging-based attack (ABA)

This strategy assumes that all the terms in one transaction are about the same context, which means they would have similar semantic relatedness. Therefore, all the terms in the sub-records from the anchoring chunk should be included in the selection step. In other words, to find the correct sub-record $ASR$ in $C_1$ for a sub-record $SR$ or term $t$ in other chunks, the semantic scores for all terms in $ASR_i$ are considered. That is, this strategy selects the best semantically related sub-record based on the average of the $ASR$ terms.

The pseudocode of the ABA strategy is provided in Algorithm 6. The algorithm is run for each input sub-record or term that needs to be reassociated. For each sub-record $ASR$ in the anchoring chunk, the average score of all the semantic relationships scores between the terms in $SR$ or $t$ and all the terms in $ASR$ is calculated (steps 1 and 2). After doing this, based on the averages, the sub-records in the anchoring chunk are arranged from the most to least related in a list 'N' (step 4). If the input is a sub-record $SR$, then the algorithm calculates the count of how many sub-records there are in a record chunk (step 6). Based on the count, the algorithm stores the most related sub-records $ASR$ in $R_{ASR}$(step 8). If the input is a term $t$, then the algorithm will store the most related sub-records $ASR$ in $T_{ASR}$ from the arranged list (steps 11 to 13). The algorithm returns the chosen sub-records from anchoring chunk for each input sub-record or term that needs to be recombined.

**Algorithm 6:** ABA

**Input:** $C_1$, $SR$ or $t$, $k$ value

**Output:** Chosen $ASR_i$

**for** *each sub-record* $ASR_i$ *in* $C_1$ **do**
    Calculate the average score of the total semantic relationships scores for $SR$ or $t$
**end**
Arrange sub-records of $C_1$ based on the average in list $N$
**if** *the input is* $SR$ **then**
    Find the $SR$ count
    **for** $i = 1$ *to count* **do**
        $R_{ASR}$←R$_{ASR}$ + $N_i$
    **end**
**end**
**if** *the input is* $t$ **then**
    **for** $i = 1$ *to* $k - 1$ **do**
        $T_{ASR}$←T$_{ASR}$ + $N_i$
    **end**
**end**
**return** $R_{ASR}$, $T_{ASR}$

**Table 4.4: The semantic scores**

| *Terms in C1* | *Terms in C2* | | *Terms in CT* | | |
|---|---|---|---|---|---|
| | tumor | biopsy | vessel | catheterisation | radiotherapy |
| blood | 0.20 | 0.27 | 0.17 | 0.25 | 0.08 |
| treatment | 0.27 | 0.34 | 0.16 | 0.37 | 0.48 |
| lung | 0.48 | 0.36 | 0.18 | 0.36 | 0.33 |
| cancer | 0.63 | 0.44 | 0.11 | 0.20 | 0.51 |

To illustrate this type of attack, consider the example of disassociated transactions in Table 4.3. The example in Table 4.3 contains one cluster with two record chunks and a term chunk. To attack this cluster, the sub-records $SR$ in the second record chunk $C2$ need to be reassociated with $C_1$, which is the same for the terms in the term chunk. As a first step, the attack applies the scoring step to obtain all the semantic relationship scores between the terms in different chunks by the WE semantic measure. Table 4.4 is the resulting semantic scores from the scoring step for the cluster in Table 4.3.

The ABA method considers the terms in a transaction to be semantic related to each other, for example, the terms in a transaction describing one disease. Therefore, the ABA calculates the average of the semantic relationship scores between a term or sub-record from different chunks and all the terms of the sub-records in the anchoring chunk by applying Equation 4.3.

$$ABA(ASR, SR) = \frac{\sum_{i=1}^{n} \frac{\sum_{i=1}^{x}(SC)}{|x|}}{|n|} \tag{4.3}$$

where $SC$ is the semantic scores between $ASR$ and $SR$, $x$ is the number of terms in $SR$, and $n$ is the number of terms in $ASR$.

For example, to find the the average semantic score between the two sub-records $ASR$

*(blood, treatment, lung)* and $SR$ *(tumor, biopsy)*, ABA will calculate its semantic relatedness, as illustrated in Equation 4.4. The similarity scores obtained by the ABA distances for all chunks are shown in Table 4.5.

$$ABA(ASR, SR) = \frac{((0.20 + 0.27)/2) + ((0.27 + 0.34)/2) + ((0.48 + 0.36)/2)}{3} = 0.316 \tag{4.4}$$

**Table 4.5: ABA results for example 4.3**

| | Record Chunks | | Term Chunk | | |
|---|---|---|---|---|---|
| ***ID*** | ***C1*** | ***C2*** | ***CT*** | | |
| | | tumor, biopsy | vessel | catheterisation | radiotherapy |
| 1 | blood, treatment, lung | 0.316 | **0.170** | **0.326** | 0.296 |
| 2 | cancer, lung, treatment | **0.416** | 0.150 | 0.310 | **0.440** |
| 3 | cancer, lung, blood | **0.393** | 0.153 | 0.270 | 0.306 |
| 4 | cancer, blood, treatment | 0.353 | 0.146 | 0.273 | 0.356 |

As a result of this attack, the reconstructed transactions are produced in Table 4.6. As can be seen, ABA reconstructed the original transactions correctly, except for the last transaction. The transactions are considered as not holding the $k^m$-anonymity privacy requirement. This means that the attacker can reassociate protected $m$ combinations terms such as (vessel, catheterisation). Therefore, the attacker can reconstruct the original data. Although, the ABA is effective in reconstructing transactions, the assumption that all terms in a single transaction are semantically connected to each other at the same level may not hold true for all transactions.

**Table 4.6: Reconstructed transactions (ABA)**

| ID | Transactions |
|---|---|
| 1 | {blood, treatment, lung, **vessel**, **catheterisation**} |
| 2 | {cancer, lung, treatment, **tumor**, **biopsy**, **radiotherapy**} |
| 3 | {cancer, lung, blood, **tumor**, **biopsy**} |
| 4 | {cancer, blood, treatment} |

### 4.3.2 Related-group attack (RGA)

The ABA strategy preforms the attack based on the assumption that the transaction terms are semantically related. However, in some datasets, a transaction may contain more than one context; for example, a patient's record may describe two unrelated diseases. Therefore, considering all the terms of $ASR$ from $C_1$ may include the unrelated terms in the semantic calculation, which can affect the accuracy of the final score, resulting in the term $t$ or sub-record $SR$ being added to the wrong transaction.

The RGA strategy considers a situation where terms may come from multiple contexts in the selection step. In other words, a term $t$ or sub-record $SR$ from different chunks can be close, semantically to some terms in a sub-record $ASR$ in the anchoring chunk but not to other terms. This makes it unreasonable to treat all terms equally to determine which transaction is the best for combination in the selection step.

In this strategy, we assume that the terms of one sub-record $ASR$ in the anchoring chunk can be divided into at least two contexts. Therefore, after applying the scoring step, the RGA strategy finds the median semantic relationship score between each $t$/$SR$ that needs to be associated and the sub-record $ASR$ in the anchoring chunk, using this value as a division indicator. Based on this division indicator, the terms in each $ASR$ in the anchoring chunk are divided into two groups. The first group is the *related*

*group*, which contains the terms that are semantically close to the disassociated $t$ or $SR$, while the other group is the *unrelated group*, which contains the rest of the terms. After that, only the semantic relationship scores for the terms in the related group will be considered when conducting the selection step.

The pseudocode for the RGA strategy to attack disassociated transactions is illustrated in Algorithm 7. The algorithm is executed to recombine disassociated terms or sub-records. For each sub-record $ASR$ in the anchoring chunk, the division indicator value of the semantic relationships between terms in $SR$ or $t$ and all the terms in $ASR$ are calculated (steps 1 and 2). Based on the division indicator value, the terms in $ASR$ in the anchoring chunk are divided into two groups: related group $RG$ and unrelated $NG$ (line 3). Only the terms in the related group $RG$ are included in the semantic calculation for $ASR$, and the average of the semantic relationships scores for terms in $RG$ will be calculated (step 4). After that, based on the averages, the sub-records in the anchoring chunk are arranged from the most to least related in a list $N$ (step 6). For inputting the sub-records $SR$, based on the count, the algorithm stores the most related sub-records $ASR$ in $R_{ASR}$(step 10). For input term $t$, the algorithm will store the most related sub-records $ASR$ in $T_{ASR}$ from the arranged list (steps 13 to 15). For each input sub-record or term that needs to be reassociated, the algorithm returns the most related sub-records from the anchoring chunk.

To illustrate the RGA strategy, we show how it works by applying it to example 4.3. After finding the division indicator value for each disassociated $t$ or $SR$ and considering the terms in the related group, the resulting average semantic scores between chunks are shown in Table 4.7.

To find the division indicator value, Equation 4.5 is used. The $SC$ is the list of the se-

mantic relationship scores between a sub-record from the anchoring chunk ($ASR$) and a sub-record or term ($SR$ or $t$) from a different chunk. For example, to find the division indicator value of the semantic scores $SC$ (0.08, 0.33, 0.48) of the first $ASR$ *(blood, treatment, lung)* and $t$ *(radiotherapy)*, the RGA calculates as illustrated performs the calculation shown in Equation 4.6.

$$\mathrm{Div_i}(SC) = \begin{cases} SC\left[\frac{n+1}{2}\right] & \text{if } n \text{ is odd} \\ \frac{\left(SC\left[\frac{n}{2}\right]+SC\left[\frac{n+2}{2}\right]\right)}{2} & \text{if } n \text{ is even} \end{cases} \tag{4.5}$$

where $SC$ is the ordered list of semantic scores of terms of $ASR$ and $n$ is the number of terms in $ASR$.

$$\mathrm{Div_i}(SC) = SC\left[\frac{3+1}{2}\right] = 2 \tag{4.6}$$

The division indicator for the first $SR$ is *0.33*. Hence, the term *blood* is excluded from the semantic score because the semantic score between *blood* and *radiotherapy* is *0.08*, which is less than the division indicator value. Consequently, *blood* is placed in the unrelated group.

As a result of the RGA attack, the reconstructed transactions can be produced, as shown in Table 4.8. As can be seen, the RGA reconstructed the original transactions correctly, except for transaction four. The RGA is considered to be an effective strategy and terms in reconstracted transactions are not holding the $k^m$-anonymity privacy requirement. Although, the RGA is effective in reconstructing transactions, more than one sub-record in $C1$ can have the same semantic score for a term $t$, so the strategy chooses the first $ASR$ with the best semantic score. This is because the four sub-records are somewhat similar to each other. This can affect the effectiveness of this

**Algorithm 7:** RGA

**Input:** $C_1$, $SR$ or $t$, $k$ value

**Output:** Chosen $ASR_i$

**for** *each sub-record* $ASR_i$ *in* $C_1$ **do**
    Calculate the division indicator value for $SR$ or $t$
    Divide terms into $RG$ and $NG$ based on the division indicator value
    Calculate the average semantic score for $RG$
**end**
Arrange sub-records of $C_1$ based on the average in list $N$
**if** *the input is* $SR$ **then**
    Find the $SR$ count
    **for** $i = 1$ *to count* **do**
        $R_{ASR}$←R$_{ASR}$ + $N_i$
    **end**
**end**
**if** *the input is* $t$ **then**
    **for** $i = 1$ *to* $k - 1$ **do**
        $T_{ASR}$←T$_{ASR}$ + $N_i$
    **end**
**end**
**return** $R_{ASR}$, $T_{ASR}$

strategy. Therefore, the strategy would work better with more distinct sub-records in the first record chunks. This is because it will give more distinctive scores that can be used in the selection step.

**Table 4.7: RGA results for example 4.3**

| | Record Chunks | | Term Chunk | | |
|---|---|---|---|---|---|
| ***ID*** | ***C1*** | ***C2*** | ***CT*** | | |
| | | tumor, biopsy | vessel | catheterisation | radiotherapy |
| 1 | blood, treatment, lung | 0.360 | **0.175** | **0.365** | 0.405 |
| 2 | cancer, lung, treatment | **0.475** | 0.170 | **0.365** | **0.495** |
| 3 | cancer, lung, blood | **0.475** | **0.175** | 0.305 | 0.420 |
| 4 | cancer, blood, treatment | 0.415 | 0.165 | 0.31 | 0.490 |

**Table 4.8: Reconstructed transactions (RGA)**

| ID | Transactions |
|---|---|
| 1 | {blood, treatment, lung, **vessel**, **catheterisation**} |
| 2 | {cancer, lung, treatment, **tumor**, **biopsy**, **radiotherapy**} |
| 3 | {cancer, lung, blood, **tumor**, **biopsy**} |
| 4 | {cancer, blood, treatment} |

### 4.3.3 Most-related attack (MRA)

The MRA strategy focuses on the strongest semantic relationship between two sets of terms. In the RGA strategy, the terms in the related group may not have the same strength of the semantic relationship for a term or sub-record. This is because that transaction's terms can describe more than one context. Hence, the MRA strategy finds the term with the strongest semantic relationship to choose which $ASR$ is the most related for combining a term $t$ or sub-record $SR$. In highly sparse datasets, the semantic relationships between terms become more distinct, increasing the chance to have more distinct semantic scores. Therefore, for each term $t$ or sub-record $SR$, the MRA strategy arranges the terms of $ASR$ from the most semantic related term to the least related in a list. Then, it will include only the most related term in each $ASR$.

After that, the strategy will add $t$ or $SR$ to the $ASR$, which have the best semantic score.

The pseudocode for the MRA strategy is provided in Algorithm 8. For each sub-record $ASR$ in the anchoring chunk, the MRA finds the best score from all the semantic relationships between the terms in $SR$ or $t$ and all the terms in $ASR$ (steps 1 and 2). After that, based on the best score in each sub-record in the anchoring chunk, the sub-records are arranged from the most to least related in a list (step 4). For the input sub-records $SR$, the algorithm stores the most related sub-records $ASR$ in $R_{ASR}$(step 8). If the input is a term $t$, then the algorithm will store the most related sub-records $ASR$ in $T_{ASR}$ (steps 11 to 13). The algorithm returns the most related sub-records from anchoring chunk for each input sub-record or term that needs to be reassociated.

To illustrate the MRA, Table 4.9 shows the semantic scores between chunks after applying it to example 4.3. The MRA strategy includes just the term with the closeest semantic relationship to determine the best $ASR$ for combining the term $t$ or sub-record $SR$. For example, by applying the MRA on the term *radiotherapy* from the term chunk, the term *treatment* in $C1$ has the strongest semantic relationship with *radiotherapy*. Then, the sub-records that contain *treatment* will be considered for adding *radiotherapy* to them.

As a result of the MRA attack, the reconstructed transactions have violated the $k^m$-anonymity privacy requirement, as shown in Table 4.10. Most or original transactions have been reconstructed correctly by the MRA. Similar to the RGA strategy, in the MRA, more than one sub-record $ASR$ in $C1$ can have the same semantic score for $t$ or $SR$, and the MRA chooses the first $ASR$ with the best semantic score. The MRA strategy can be more effective with more sparse dataset. This is because the relationships between terms will be more variant, which can mean more distinctive scores that

**Algorithm 8:** MRA

**Input:** $C_1$, $SR$ or $t$, $k$ value

**Output:** Chosen $ASR_i$

**for** *each sub-record* $ASR_i$ *in* $C_1$ **do**
    Find the best score in the semantic relationships for $SR$ or $t$
**end**
Arrange sub-records of $C_1$ based on the best scores in list $N$
**if** *the input is* $SR$ **then**
    Find the $SR$ count
    **for** $i = 1$ *to count* **do**
        $R_{ASR} \leftarrow \mathrm{R}_{ASR} + N_i$
    **end**
**end**
**if** *the input is* $t$ **then**
    **for** $i = 1$ *to* $k - 1$ **do**
        $T_{ASR} \leftarrow \mathrm{T}_{ASR} + N_i$
    **end**
**end**
**return** $R_{ASR}$, $T_{ASR}$

can be used in the selection step.

### 4.3.4 Vertical partitioning attack (VPA)

The idea of this strategy is to use the vertical partitioning step of the disassociation method to check if the reconstruction is valid. The disassociation method uses $k^m$-anonymity as a basis to divide terms vertically into chunks. Unlike previous strategies,

**Table 4.9: MRA results for example 4.3**

| | Record Chunks | | Term Chunk | | |
|---|---|---|---|---|---|
| ***ID*** | ***C1*** | ***C2*** | ***CT*** | | |
| | | tumor, biopsy | vessel | catheterisation | radiotherapy |
| 1 | blood, treatment, lung | 0.42 | **0.18** | **0.37** | 0.48 |
| 2 | cancer, lung, treatment | **0.53** | **0.18** | **0.37** | **0.51** |
| 3 | cancer, lung, blood | **0.53** | **0.18** | 0.36 | **0.51** |
| 4 | cancer, blood, treatment | **0.53** | 0.17 | **0.37** | **0.51** |

**Table 4.10: Reconstructed transactions(MRA)**

| ID | Transactions |
|---|---|
| 1 | {blood, treatment, lung, **vessel**, **catheterisation**} |
| 2 | {cancer, lung, treatment, **tumor**, **biopsy**, **radiotherapy**} |
| 3 | {cancer, lung, blood, **tumor**, **biopsy**} |
| 4 | {cancer, blood, treatment} |

this strategy does not use the semantic relationships to attack the disassociated transactions. Instead, the VPA strategy finds possible combinations of chunks by finding all the combinations between the sub-records or terms for each two chunks. Then, the VPA strategy associates sub-records and terms based on one combination at a time. After that, the VPA strategy tests this reconstruction by applying vertical partitioning. If the resulting chunks are similar to the chunks in the disassociated transaction, then the associated chunks are considered to be correct. Otherwise, the VPA will move to the next combinations and check the vertical partitioning again until finding the correct partitioning.

This strategy is executed independently for each cluster $P$ in the disassociated dataset. Each iteration will be run on every two chunks from $C_1$ to $C_n$ by finding all the possible

combinations between the two record chunks (steps 2 and 3). After that, this method will temporarily add the two chunks based on one possible combination each time before applying vertical partitioning to the temporarily reconstructed records (steps 5 to 7). If the vertical partitioning of the reconstructed record chunks produces similar record chunks to the disassociated transaction, then the strategy adds sub-records permanently, and the reconstructed records will be saved (steps 8 to 10). Otherwise, the temporarily reconstructed record chunks will be discarded (step 13), and the method will check the next possible combination between the current two record chunks (step 15). This step will be repeated until it passes the vertical partitioning for all record chunks. The algorithm will store the reconstructed cluster in $Rec_P$ (step 17).

To illustrate this strategy, we apply it to example 4.3; there are two sub-records of *(tumor, biopsy)* in $C2$ *(tumor, biopsy)*. Based on the possible combinations between $C1$ and $C2$, *(tumor, biopsy)* can be added based on one of the following combinations of the four sub-records in $C1$: (1,2), (1,3), (1,4), (2,3), (2,4), and (3,4). For example, by using the first combination, the VPA strategy will add *(tumor, biopsy)* to the first and second sub-records in $C1$. After that, vertical partitioning is applied to this combination. If it produces similar record chunks as the disassociated transactions in 4.3, then the combination would be considered to be correct.

As a result of the VPA attack, the reconstructed transactions are shown in Table 4.11. The VPA strategy aims to reconstruct transactions. However, there is usually more than one valid combination for combining two record chunks, and the number of these valid combinations is affected by the number of transactions in a cluster. Also, the number of these valid combinations decreases when more chunks are combined. Therefore, even if reconstructed transactions pass the vertical partitioning step, the combination may be not the correct one. Also, the VPA strategy is not applicable for terms in the term chunk; this strategy adds them randomly to transactions, which can affect the ef-

**Algorithm 9:** VPA

**Input:** Disassociated dataset

**Output:** Reconstructed record chunks of transactions

**for** *each cluster* $P$ **do**
  **for** *Every two record chunks in* ($C_1$ *to* $C_n$) **do**
    Find all the possible combinations between($C_i$ and $C_{i+1}$)
  **end**
  **for** *each combination* **do**
    Add sub-records in $C_i$ and $C_{i+1}$ based on current combination
    Apply VP to the current reconstructed record chunks
    **if** *the current reconstructed record chunks pass the VP* **then**
      Save the current reconstructed record chunks
      Move to the next record chunk $C_i$
      **else**
        Discard the current reconstructed record chunks
      **end**
      Check next combination
    **end**
  **end**
  $Rec_P$=The reconstructed record chunks of cluster $P$
**end**
**return** *The reconstructed record chunks of transactions*

fectiveness of this strategy.

**Table 4.11: Reconstructed transactions (VPA)**

| ID | Transactions |
|---|---|
| 1 | {blood, treatment, lung, **tumor**, **biopsy**, **catheterisation**} |
| 2 | {cancer, lung, treatment, **tumor**, **biopsy**, **radiotherapy**} |
| 3 | {cancer, lung, blood, **vessel**} |
| 4 | {cancer, blood, treatment} |

## 4.4 Summary

We started this chapter by discussing the semantic similarity concept and reviewed two types of semantic similarity measurementes. After doing this, we discussed two semantic measures used in our attacking approach: NGD and WE. We then proposed our semantic attacking approach, which consists of two steps: scoring and selection. Scoring calculates the semantic relationships between sub-records and terms by using NGD and WE. The result of this step is then used by the selection step to combine the sub-records and terms to reconstruct the original transactions. In the selection step, we introduced different strategies to determine how the scoring results are used to choose the best sub-record for combination.

The averaging-based attack (ABA) assumes that in one transaction, all the terms are semantically related. Therefore, all the terms in the anchoring chunk are considered when selecting the most related sub-records for reconstruction. However, this assumption may not apply to all datasets, which can affect the effectiveness of this strategy.

The related-group attack (RGA) is proposed based on the idea that the terms in a transaction may not always be related. Therefore, we divide terms into two groups: related and unrelated. The effectiveness of this strategy can be affected by the similarity between related groups in the sub-records in the anchoring chunk.

The most-related attack (MRA) depends on the strongest semantic relationship when it comes to reassociating chunks. However, this strategy would be more effective with a sparse dataset where the relationships between terms can be more distinct.

The vertical partitioning attack (VPA) does not use semantic relatedness to reconstruct the dataset. Instead, the chunks are combined based on the chosen combination between two adjacent record chunks, and it employs vertical partitioning to check the validity of this chosen combination.

In general, the difference between the selection strategies depends mainly on the level of density of datasets. The ABA strategy is be more effective with denser datasets, whereas MRA and RGA strategies are more effective with less dense datasets. Also, the cluster size may affect the performance of these methods if the $k^m$-anonimity requirement is very strong. However, the VPA strategy's effectiveness is more affected by the size of clusters, whereas smaller cluster size is more vulnerable to VPA strategy.

In the following chapter, we provide our experimental findings and evaluate the accuracy of our proposed algorithms.

# *Chapter 5*

# Experiments and Results

The goal of the experimental evaluation is to demonstrate the performance and effectiveness of our proposed methods for attacking disassociated transaction data. This chapter evaluates how the exploitation of semantic relationships between terms can be used to reassociate terms and sub-records from different chunks, hence reconstruct the original transactions.

We start this chapter by providing a discussion on the datasets that have been used in our experiments and how we prepared them. We test different properties to evaluate our methods in a range of conditions. After that, we discuss and compare the effectiveness of our algorithms. Finally, we analyse the experimental results using different measures.

## 5.1 Dataset Preparation and Experiment Setup

### 5.1.1 Dataset properties

To conduct our experiments, we have prepared our datasets with different properties we also vary a number of parameters when anonymising data to ensure that our semantic attack is evaluated under different conditions. We have used real datasets construc-

ted from EzineArticles (general articles)[1]. The articles have been written by experts and cover many specialized fields. Also they are in the form of short and informative articles that makes them a suitable source of our transactions. After extracting the transactions, we anonymise them using the disassociation method with the properties and parameters:

- ***Dataset density***

  A transaction dataset $D$ consists of transactions $(T_1, T_2, ....., T_n)$, where each transaction $T$ consists of a number of terms. Hence, to measure the dataset density in our experiments, we used the type-token ratio (TTR) [62], which is the number of distinct terms divided by the total number of all terms in the dataset over all the transactions $(T_1, T_2, ....., T_n)$ in a dataset $D$. It is defined as follows:

$$TTR = (\frac{distinct\ terms}{All\ terms}) \tag{5.1}$$

  The TTR illustrates how often terms occur in the dataset, and it will affect the number of terms in both the record and term chunks in the resulting anonymised dataset. Therefore, the density level can affect the effectiveness of our attack, so we use different levels of density to evaluate our methods.

- ***Semantic property***

  The semantic property refers to the meaning or interpretation of terms and how to draw meaning from the relationships between terms. In our experiments, we select the datasets by ensuring that they have this property, so the results of the attack reflect the hypothesis stated in chapter one.

[1] `https://ezinearticles.com`

This property may not exist in some datasets, such as a shopping basket dataset, where the relationship between items is based on items frequently bought together. For example, if we have two sub-records from two different records chunks [ Milk, Eggs, Coke] and [Bread], we can infer that the two sub-records are related because people often buy bread, milk, eggs and coke together. However, they may not be associated by semantic relationships or have similar meaning, but rather based on a pattern that can be mined in a shopping basket dataset. Our methods are not designed to this types of datastes.

- ***Quality of data***

  This indicates if the data contain many abbreviations and errors in their spelling. The more of these are in a dataset, the harder it is for extracting terms, and this can also affect the semantic measurement because NGD and WE may not understand the terms.

- ***The*** $k$ ***and*** $MaxClusterSize$ ***parameters***

  The $k$ variable needs to vary to evaluate its impact on the attacking performance. We test $K$=2, 3, 4 and 5. Also, the $MaxClusterSize$ parameter determines the largest size allowed as a cluster, and the value of this parameter cannot be less than $k$ value. In our experiments, we fix the value of $k$ at 2 and test the $MaxClusterSize$ value from $k^2$ to $k^6$.

### 5.1.2 Experiment Setup

To execute the experiments, a dataset needs to be processed across three stages. First, data is collected and extracted. In this stage, data need to be pre-processed

in order to form the transactions. Second, these transactions need to be anonymised. In this stage, the disassociation method is applied. After disassociating the transactions, the dataset will be ready to implement the semantic attack as the last stage. In the scoring step of our approach, we use Spyder software tool[2] to program and apply NGD and WE semantic measures on terms. Then the selection methods will be implemented as explained previously in Chapter 4.

**Data pre-processing**

In our experiments, we use real-world datasets collected from Ezinearticles.com. This source contains hundreds of thousands of articles. The reason that we have chosen Ezinearticles is because the articles cover a wide range of topics, allowing us to evaluate our approach in different domains and to have datasets with different levels of density.

To construct our datasets, we have chosen around 1,000 articles in different topics with a varying number of keywords to form transactions. Our experiment's main goal is to anonymise transactions using the disassociation method and then attack them using our semantic attack. Therefore, we follow some steps to transfer the articles (free text form) to transactions:

– Articles can contain information, such as: titles, references and external links. We concentrate on extracting the main content and removing these parts as a first step to prepare our datasets.

– In the next step, we use tokenisation to process the raw text. In natural language processing (NLP), tokenisation is performed by chopping text up into smaller

[2] https://www.spyder-ide.org

units called tokens. In our datasets, we split the text into words.

– To analyse these tokens properly, we reduce the inflectional forms of the words to their common base forms by using lemmatisation. This task resolves a word to its dictionary form, which is known as a lemma by considering its meaning and context. For example, the words studies, studying and study's will be resolved to 'study'.

– Then, we remove tokens if: 1) the token is a stop word; 2) the token is a number; 3) the token is a punctuation; and 4) the token is a single characters. 5) We also remove duplicated terms.

– To control topics and keywords, we use an unsupervised technique called non-negative matrix factorisation (NMF) for topic modelling [22] to identify the topics that occur in a collection of articles; then, we cluster them based on the topic models. NMF is a statistical technique for reducing the dimension of the input text. It converts articles into a term-articles matrix, which is a collection of all the terms in the given articles. Then, it assigns a weight to each term in the articles. To illustrate NMF in more detail, there are W and H matrices in the original matrix V. W represents the topics it found, and H represents the weights associated with those topics. In our case, V is the original the articles by terms, H represents articles by topics, and W represents the topics by terms.

– To construct transactions from the articles, based on the resulting topics from the previous step, we classify and select the articles. Then, we construct two hundred transactions with around 4,000 terms from more than 35 different topics that can be attacked.

**Data Anonymisation**

In this section, we illustrate how we use the disassociation method to anonymise transactions. To disassociate a dataset, we perform the following steps:

1) We apply horizontal partitioning by using the algorithm given in chapter three to disassociate data. The transactions are grouped into clusters with a size between $k$ and $MaxClusterSize$. To handle clusters with a size smaller than $k$, we use the Adding strategy illustrated in chapter three.

2) Then, we apply vertical partitioning to each cluster. We apply the $k^m$-anonymity requirement by following the second method of the vertical partitioning illustrated in chapter three.

## 5.2 Performance Evaluation

To evaluate our proposed semantic attacking methods, a random attack on disassociated transactions will be used as a baseline method. In a random attack, an attacker randomly combines record chunks and term chunks with no information other than the dataset that has been released. To measure the attack on the privacy of disassociated transactions for record chunks and term chunks, we use two different types of measurement. And these are given in the sections below.

### 5.2.1 Privacy Breakage

The disassociation method protects the data privacy of a dataset $D$ by partitioning transactions based on $k^m$-anonymity. Hence, a privacy breach occurs when an adversary can connect the terms from different partitions and break the $k^m$-anonymity requirement. To measure privacy breakage, we introduce two methods: transactions breakage and $k^m$-anonymity breakage.

#### Transactions breakage

This measure calculates the breakage privacy of each transaction. If an adversary is able to link at least one term to its transaction, then the privacy for this transaction will be considered broken. The attack aims to break the privacy of the protected links between the sub-records $SR$ that come from different record chunks, or between the sub-records $SR$ from a record chunk and terms $t$ from a term chunk $C_T$ in a cluster $P$. Note that, the terms in the anchoring chunk $C_1$ for each transaction will always be the original. Hence, to measure the breakage in a more accurate way, the first chunk will be excluded from the calculation for each transaction ($T^-$ = T - $C_1$), as well for each reconstructied transaction ($\overline{T^-}$ = $\overline{T}$ - $C_1$).

To evaluate transaction breakage, we need to find out how many reconstructed transactions $\overline{T}_i$ break the privacy of its original transaction $T_i$. So after excluding $C_1$, the privacy of a transaction $T^-_i$ has been broken if $Br(T_i)$ in Equation 5.2 does not equals $0$.

To find the total percentage of transactions breakage, the total number of $Br(T_i)$ will be divided by the total number of transactions in the reconstructed dataset $|\overline{D}|$.

$$Br(T_i) = |\overline{T^{-}}_i \cap T^{-}_i| \tag{5.2}$$

$$Br(D) = \frac{\sum_{i=1}^{n} Br(T_i)}{|\overline{D}|} \tag{5.3}$$

#### $k^m$-anonymity breakage

The $k^m$-anonymity requirement protects infrequent itemsets that have support less than $k$ by dividing an infrequent itemset's terms into different chunks. Hence, this measure calculates the breakage based on how many of these protected infrequent itemsets are re-discovered. Therefore, if an adversary can recover a disassociated itemset, then a breach of privacy has occurred. To calculate the total percentage of $k^m$-anonymity breakage, the number of recovered protected infrequent itemsets in the reconstructed dataset $\overline{D}$ is divided by the total number of protected infrequent itemsets in the original dataset $D$.

### 5.2.2 Reconstruction

In this type of measurements, we measure the effectiveness of our semantic attack by calculating how similar the reconstructed transactions are to the original transactions. Although the attacker will not have the original transactions to measure the validity

of his attack, the results of the reconstruction process will indicate how serious is the privacy threat of using the semantic attack on disassociated datasets. This can affect the reliability of using the disassociation method to protect data.

The reconstruction process takes the disassociated dataset $\hat{D}$ as an input and outputs de-anonymised transactions $(\overline{T}_1, ....., \overline{T}_n)$ to produce the reconstructed dataset $\overline{D}$. To calculate the correctness of reconstruction, we use two measures: accuracy and word mover's distance.

### Accuracy

The accuracy measure calculates the percentage of correct reconstruction of transactions. To reconstruct a transaction $T_i$, its disassociated terms in different chunks need to be correctly reassociated to produce the reconstructed transaction $\overline{T}i$.

Equation 5.4 calculates the percentage the correct reconstruction for each transaction. This is performed by finding how many disassociated terms of this transaction are reassociated correctly in the reconstructed dataset $\overline{D}$ divided by the number of its disassociated terms. The value of $Rec(T_i)$ is between 0 and 1, if $Rec(T_i)$ is equal to 1 for a single transaction, if the whole transaction is reconstructed successfully.

$$Rec(T_i) = \frac{|\overline{T\text{-}_i} \cap T\text{-}_i|}{|T\text{-}_i|} \tag{5.4}$$

To evaluate the reconstruction for a dataset $D$, the accuracy finds the sum of $Rec(T_i)$ from equation 5.2 for all transacctions $\overline{T}$ in $\overline{D}$; this sum is divided by the total number

of transactions in the original dataset $D$.

$$Ac(D) = \frac{\sum_{i=1}^{n} Rec(T_i)}{|D|} \tag{4}$$

**Word mover's distance**

The word mover's distance (WMD) is a method for calculating the semantic similarity between two documents; it measures the minimum distance that the words in one document need to travel to reach the words in another document in a semantic space [53], [21].

This is calculated by using the word embeddings of the words in two documents to measure the minimum distance. If the distance is small between the two documents, then the words in the two documents are similar to each other.

In our experiments, we aim to measure the WMD between the original dataset and the reconstructed dataset. To do this, we calculate the WMD of each original transaction against its reconstructed one in a Word2vec space and then find the average of all distances.

## 5.3 Results and Discussion

In this section, we evaluate our semantic attacking methods on disassociated datasets. Afterwards, we discuss the results of attacking record chunks and attacking term

chunks. We evaluate the reconstruction's accuracy by calculating the percentage of terms and sub-records that are correctly combined for both record chunks and term chunks. Also, we evaluate how well the semantic attacking methods reconstruct transactions by finding the WMD between the original transactions and reconstructed transactions. We next consider the breakage percentage of each algorithm in terms of the transaction breakage and $k^m$-anonymity breakage. Finally, we evaluate how the density will affect an algorithm's performance.

### 5.3.1 Effect of $k$ value

In Figure 5.1, we investigate the efficacy of our algorithms with varying $k$ values and fixing the max cluster size value at $5^2$. The $k$ value is used as a privacy constraint that needs to be satisfied in the disassociated dataset. Increasing the $k$ value in the disassociation method means increasing the protection level, which usually results in pushing more terms to term chunks, and sub-records in the record chunks become more indistinguishable. In terms of accuracy, the effect of increasing $k$ is positive regarding our algorithm's performance. This is because of the following two possible situations: first, because the number of transactions in a cluster is increased to satisfy the $k^m$-anonymity requirement, this causes the number of sub-records in the anchoring chunks that have the same semantic relationship scores to increase as well. This will reduce the chance of choosing the wrong sub-record when associating a term. The other possible situation is when the number of identical sub-records in the anchoring chunk is high and with no semantic differences. As a result, any sub-record that is chosen for adding a term to it will be correct. However, with increasing $k$, the difference between our methods and the random attack becomes smaller. This is because the sub-records in the anchoring chunk become almost identical, so the difference between semantic relationships scores becomes insignificant.

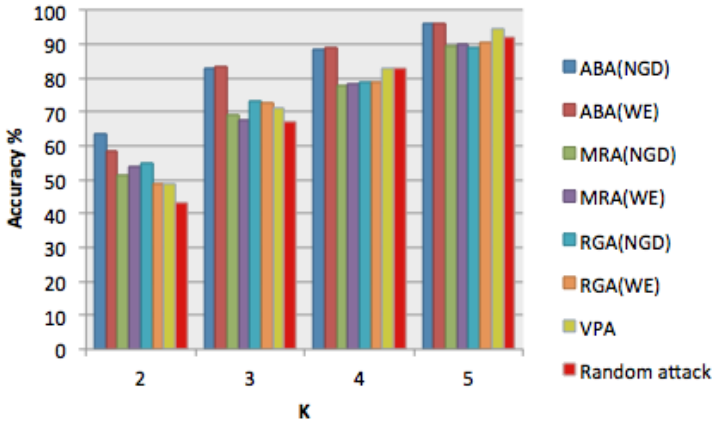



**Figure 5.1: Comparing the overall accuracy of the attacking methods**

Overall in Figure 5.1, we can see a clear upward trend in accuracy percentage, reaching over 90% of the reconstructed sub-records and the terms are correct as well. Hence, the algorithm's effectiveness of finding the combined sub-records and terms in the original datasets increases with an increase in $k$, even for a random attack. Also, it can be seen in Figure 5.1 that ABA with both NGD and WE measures has the best performance with different $k$ values; this is because of the density level of the dataset. The density level is fixed at 0.30 in this experiment, which is considered to be a dense dataset. So considering all the terms from the anchoring chunk in the selection step is better in determining the semantic distances between chunks. However, continuing to increase $k$ value can lead to decrease in the accuracy. This is because the difference between the $k$ value and the $MaxClusterSize$ value will decrease. Also, this may lead to an increase in the number of identical sub-records in the anchoring chunk, which will affect the accuracy of our semantic attack.

Figure 5.2 illustrates the reconstruction extent in terms of reconstructing the entire transactions correctly from original dataset. In general, the semantic distance between the reconstructed and the original transaction is increased with an increasing $k$ value

for VPA and random, while it is slightly decreased for the semantic attack methods after $k$ equals 4. However, the number of terms in the anchoring chunk affects the WMD, so larger numbers means less terms in different chunks that need to be reassociated and that less semantic distance is needed between the terms in both the original and reconstructed transactions. Therefore, with increasing $k$, the number of terms in the anchoring chunk decreased and the semantic distance started to increase. However, semantic attack methods maintained a low WMD with increasing $k$ compared with the random attack.

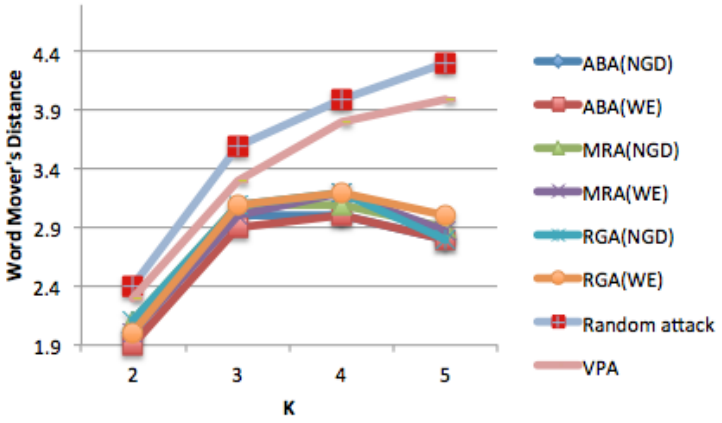



**Figure 5.2: Comparing the overall WMD of the attacking methods**

In Figure 5.3, we can see the effectiveness of our algorithm's performance on privacy breaking of transactions. Overall, the breakage rate is around 25% for all $k$ values. However, an increasing value of $k$ has a different effect on attacking record chunks and term chunks, which we analyse in detail in the next sections. However, attacks on record and term chunks have opposite trends for transactions privacy breakage with increasing $k$. This explains the fluctuating trend of the overall transactions' privacy breakage at different $k$ values.

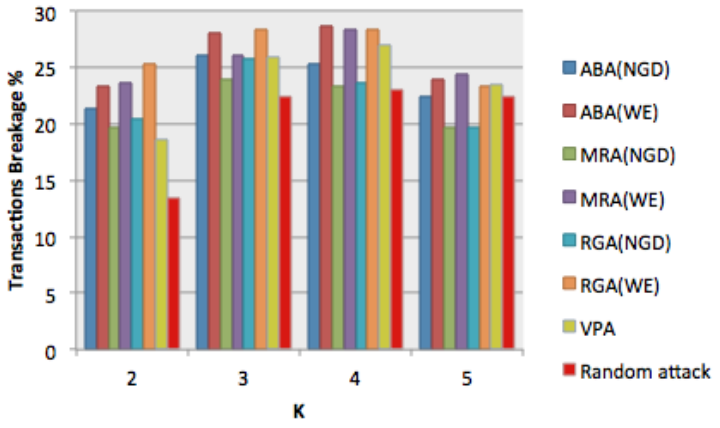



**Figure 5.3: Comparing the overall transaction breakage of the attacking methods.**

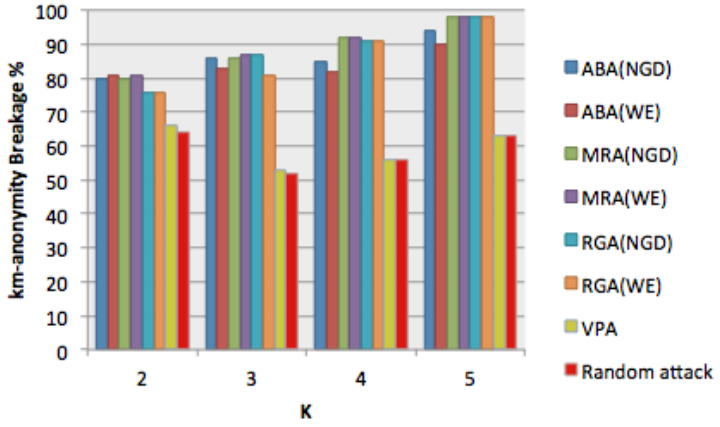



**Figure 5.4: Comparing the overall $k^m$-anonymity breakage of the attacking methods.**

Figure 5.4 shows the impact of increasing $k$ values on attacking the protected infrequent itemsets in the disassociated dataset. It can be seen that the breakage percentage increases with $k$. In general, a higher $k$ means more infrequent itemsets that have been protected. However, because we associate terms based on the semantic relationships in our semantic attack methods, the increase in the number of protected itemsets means a greater chance of finding more infrequent itemsets.

### Record chunks attack

The value of $k$ affects the number of record chunks and sub-records inside the record chunks. Figure 5.5 shows the accuracy of attacking record chunks over different values of $k$. Increasing $k$ will decrease the number of record chunks in a cluster, which means that fewer sub-records need to be added and that the chance of constructing the wrong sub-records decreases. Therefore, the accuracy percentage increases with an increasing $k$ value.

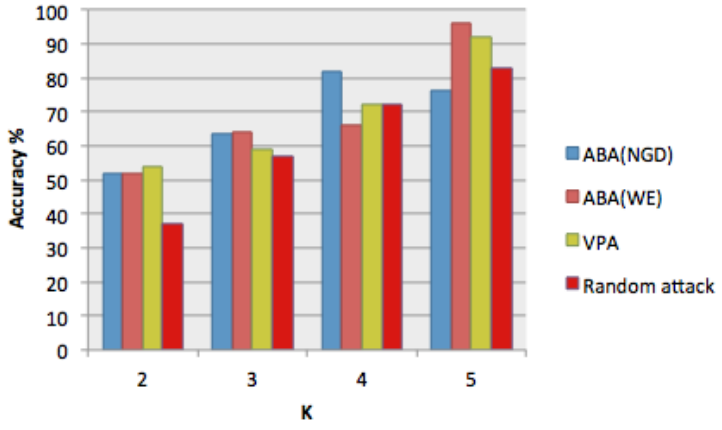



**Figure 5.5: Comparing the accuracy of the attacking methods on record chunks**

The effect of $k$ on the transactions breakage measure is shown in Figure 5.6. The percentage of broken transactions decreases with increasing value of $k$. As mentioned earlier, there is a negative relationship between the number of record chunks in a cluster and the $k$ value. This means the number of transactions that can be broken into higher $k$ values is less. In other words, the terms of transactions will be divided between the anchoring chunk and the term chunk. Hence, there are no more sub-records that need to be associated to reconstruct a transaction from the record chunks. However, at some points, NGD performance is better than WE performance, while at other points, it is the opposite. This is because of the semantic nature between the terms and how the

semantic measure captures this relationship. For the VPA method, because the number of identical sub-records in a record chunk is increased, the number of possible ways of vertical partitioning that can pass the $k^m$-anonymity requirement becomes larger. Hence, this means the chance of choosing a non-original partitioning and associating the sub-record with the wrong transaction is greater.

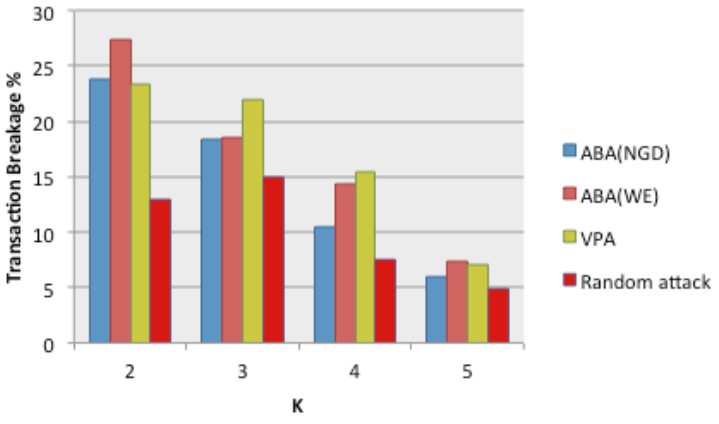



**Figure 5.6: Comparing the transaction breakage of the attacking methods on record chunks.**

In Figure 5.7, we illustrate that even with a higher $k$ value, an adversary will be able to reconstruct almost all the protected itemsets. This is because the total number of protected itemsets that has been separated into the record chunks has decreased, so the error rate will be low. Hence, even a random attack will have a high chance of reconstructing many protected itemsets correctly.

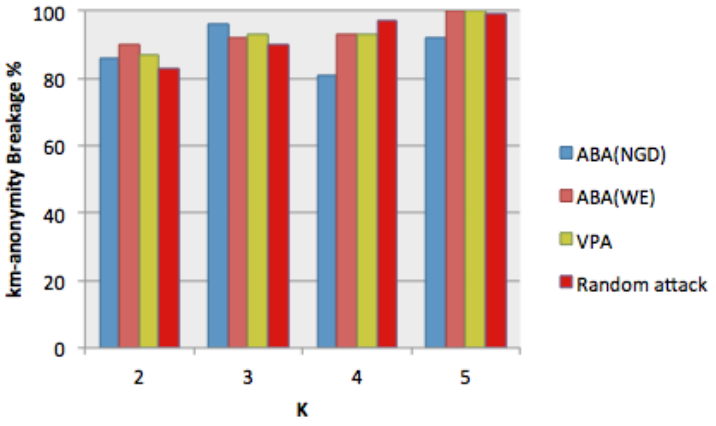



**Figure 5.7: Comparing the $k^m$-anonymity breakage of the attacking methods on record chunks.**

**Term chunks attack**

The size of term chunks will be affected by the $k$ value. A higher $k$ will impose higher protection in terms of the disassociated dataset, and as a result, more terms will be moved to term chunks. Figure 5.8 shows the accuracy of our attacking algorithms on term chunks. The accuracy increases for all the methods when the $k$ value is increased. This is because of the increase in number of indistinguishable sub-records in the anchoring chunks, which means that the chance to successfully reconstruct the transaction and find this reconstructed transaction in the original dataset becomes higher. This also explains the decrease in the difference between the methods' performance. Overall, the ABA method with both NGD and WE shows the best performance across different values of $k$. This can be related to the density level of the dataset. The dataset used in this experiment is very dense; hence, it relies on one or a few terms from the anchoring chunk to find the semantic relationships, which is not as effective as considering all the terms.

In Figure 5.9, we investigate the efficacy of the attacking algorithms on term chunks in

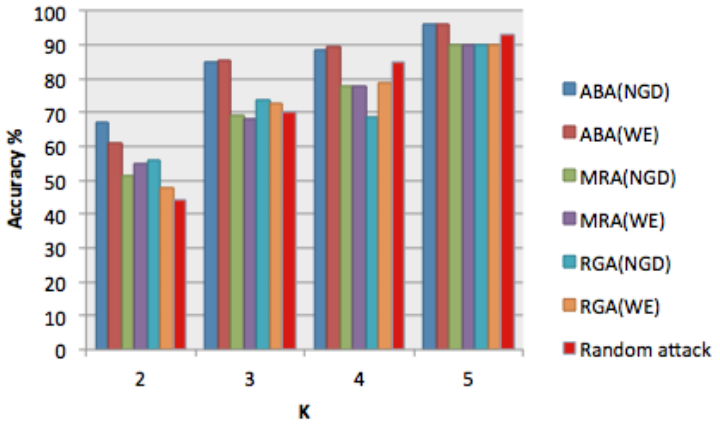



**Figure 5.8: Comparing the accuracy of the attacking methods on term chunks**

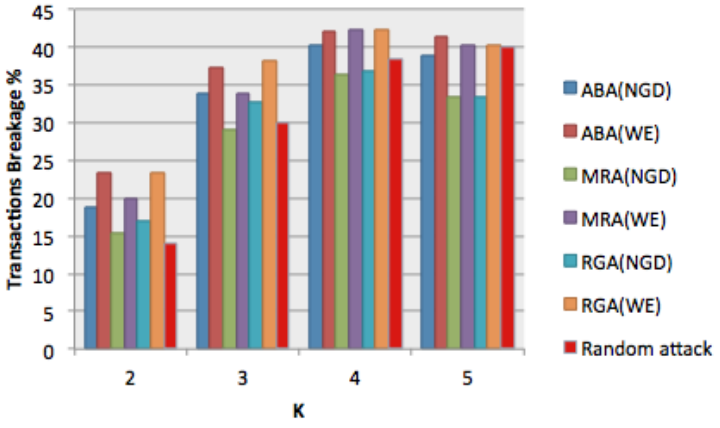



**Figure 5.9: Comparing the transaction breakage of the attacking methods on term chunks.**

terms of the transaction breakage. Unlike record chunks attacking, the performance of attacking term chunks improves with an increasing $k$ value. This is because the number of distinct sub-records in the anchoring chunk decreases, so the number of transactions to choose from drops.

For the $k^m$-anonymity breakage of the term chunks, Figure 5.10 shows the effectiveness of our algorithms with a varying $k$. There is a positive relationship between the

size of term chunk and the value of $k$. As a result, there are more protected itemsets from the term chunks when $k$ is larger. However, because of the decrease in the number of distinct sub-records in the anchoring chunk, the chance of adding the itemset's terms to a transaction correctly increases.

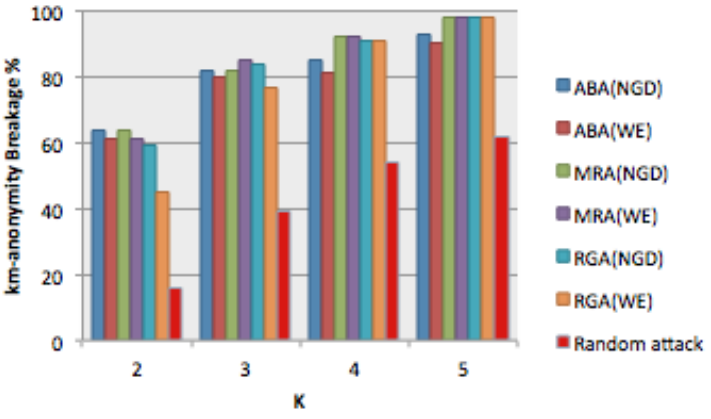



**Figure 5.10: Comparing the $k^m$-anonymity breakage of the attacking methods on term chunks.**

### 5.3.2 Effect of Data Density

Because one of the important properties that can influence the results of our algorithms is the density of a dataset, this section focuses on how our algorithms work at different density levels. To conduct this experiment, subsets of documents have been selected from the chosen articles with average density levels ranging from 0.2 to 0.7.

The result in Figure 5.11 compares the accuracy of our algorithms at different density levels. In general, the accuracy greatly increases as the density level increases for most attacking methods. This is because increasing the sparsity level means that there are more distinct terms, meaning that there will be more distinct terms and more varied

semantic relationships in a dataset.

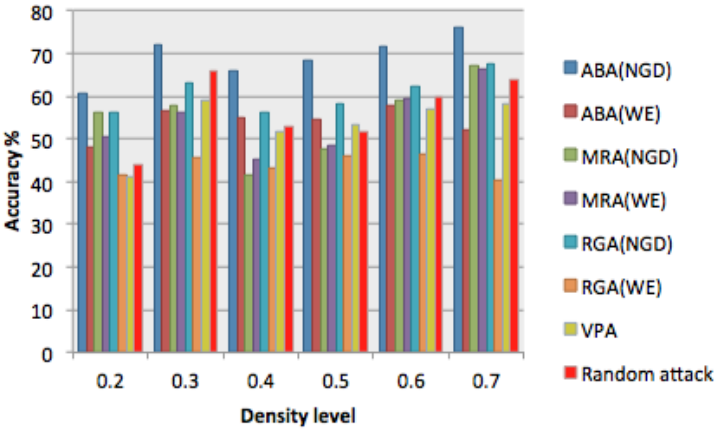



**Figure 5.11: Overall accuracy of the attacking methods with different data densities.**

As explained in the previous chapters, NGD uses the WWW as a corpus to find the semantic relationships. Therefore, the NGD measure can find the semantic score for any term in a dataset. On the contrary, the WE measure will be limited by the trained corpus. This shows that when increasing the sparsity level, the methods using NGD as a semantic measure perform better than the same methods that using the WE measure in Figure 5.11.

Figure 5.12 describes the reconstruction of correct transactions. For the semantic attack methods, the density level does not have a strong effect on the full reconstruction, so the results fluctuated between 2.5 and 3.

The results of transactions breakage are presented in Figure 5.13. The breakage level for all attacking methods improves by increasing the sparsity level until 0.5 has been

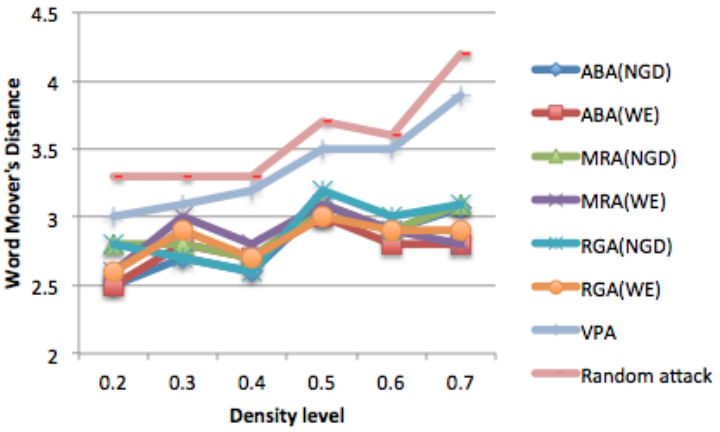



**Figure 5.12: Overall WMD of the attacking methods with different data densities.**

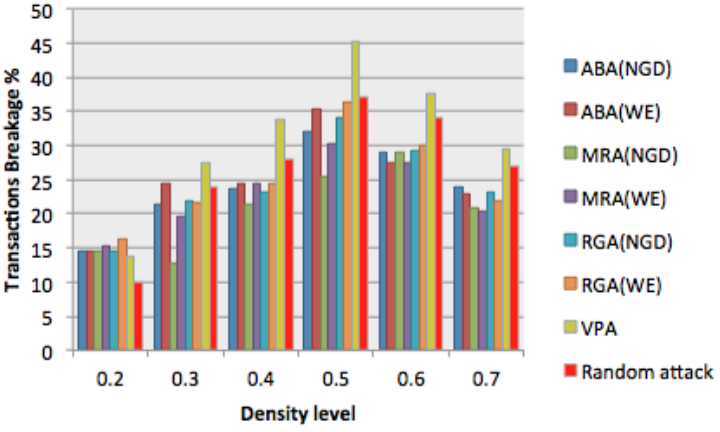



**Figure 5.13: Overall transaction breakage of the attacking methods with different data densities.**

reached, at which point it starts to decrease. This is because after 0.5, the number of sub-records or terms that have a frequency greater than $k$ drops. In other words, the number of terms inside the record chunks will decrease.

Figure 5.14 shows the overall $k^m$-anonymity breakage of the attacking methods with different data density levels. In general, the difference between the performances of attacking methods becomes clearer when the density is higher. This is because more dense datasets have more diversity of semantic relationships; therefore, this helps to

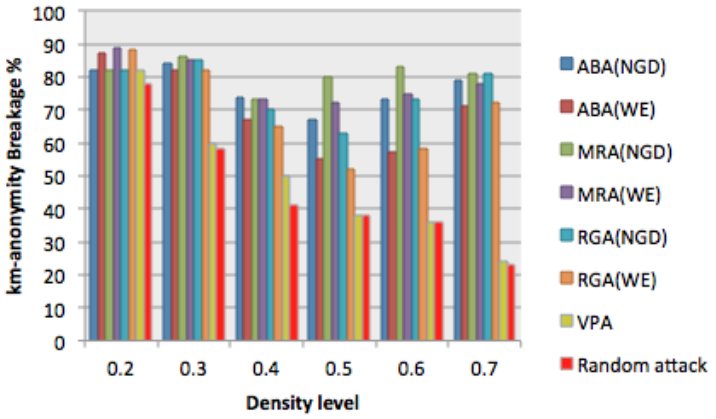



**Figure 5.14: Overall $k^m$-anonymity breakage of the attacking methods with different data densities.**

determine which terms need to be included from anchoring chunks, and which has an effect on the total semantic scores, hence affecting the reconstruction.

### 5.3.3 Record chunks attack

Figure 5.15 illustrates the impact of density on the accuracy of attacking record chunks. Here, a sparser dataset has more distinct sub-records in the record chunks, meaning that there are more distinct semantic relationships scores. This explains the improvement in the accuracy results for the semantic attacking methods. However, an excessive increase in density level will harm this distinct level because the number of terms with enough frequency to be included in the record chunks will be very low. In our datasets, anything after the 0.5 level is considered to be excessive for anonymising by disassociation. For the VPA, the accuracy is not affected by an excessive increase in density. Therefore, the performance will continue to improve.

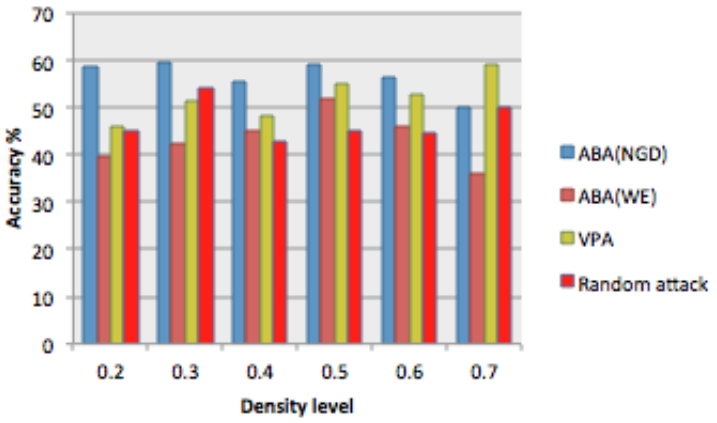



**Figure 5.15: Accuracy of attacking record chunks with different data densities**

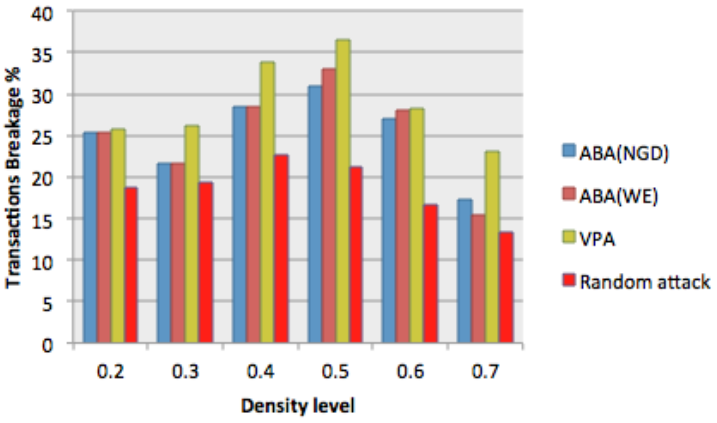



**Figure 5.16: Transaction breakage of attacking record chunks with different data densities.**

For the transactions breakage, when attacking record chunks, Figure 5.16 shows the performance of our algorithms at different density levels. The breakage reaches its highest level at 0.5 for all attacking methods. In general, the VPA breaks more transactions than the semantic attack methods. This is because VPA depends on $k^m$-anonymity. Therefore, it benefits from distinct sub-records and a decrease in the number of transactions within a cluster when the density level increases.

In Figure 5.17, we illustrate the impact of density level on the $k^m$-anonymity breakage. In general, when increasing the density, there are no significant changes in the number of protected itemsets that have been successfully attacked.

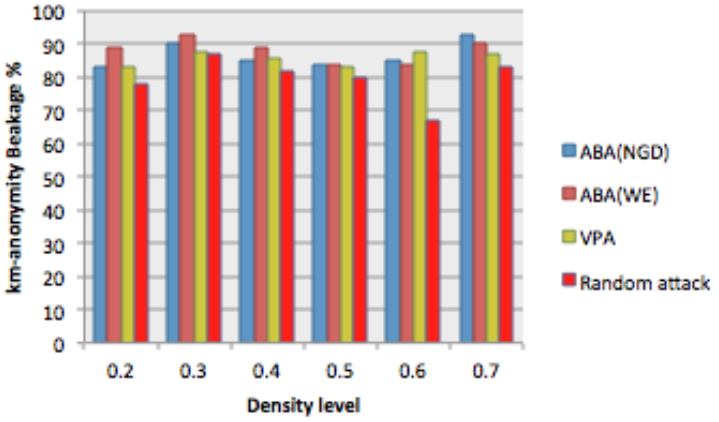



**Figure 5.17: $k^m$-anonymity breakage of attacking record chunks with different data densities.**

### 5.3.4 Term chunks attack

Figure 5.18 shows the accuracy of our attacking algorithm on term chunks with different density levels. The accuracy is slightly increased after increasing the density for some methods that use the NGD measure, while the trend is the opposite for some methods using the WE measure. This is because with increasing density levels, more terms are included in the dataset, so the likelihood of not finding a term in the corpus increases; consequently, the WE measure becomes less accurate.

In general, increasing the density level will result in pushing more terms to term chunks. The breakage level in terms of transactions privacy breakage is illustrated

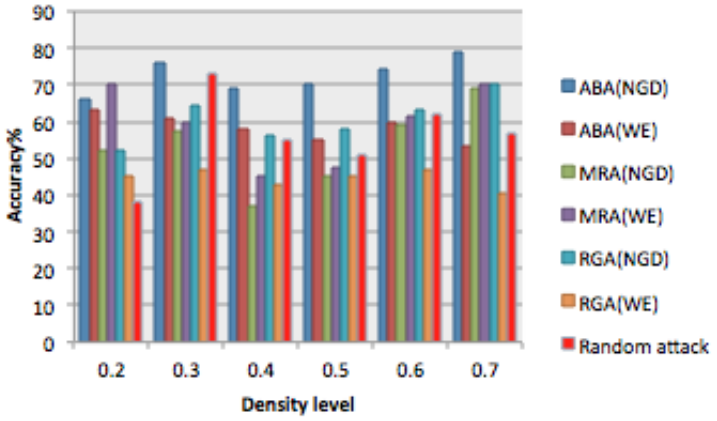



**Figure 5.18: Accuracy of attacking term chunks with different data densities**

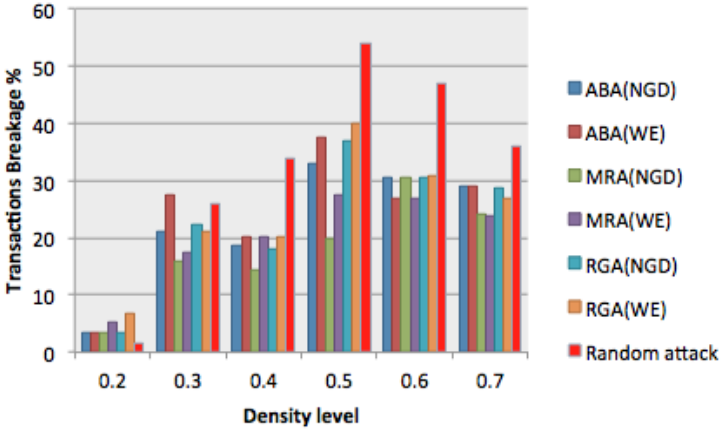



**Figure 5.19: Transaction breakage of attacking term chunks with different data densities.**

in Figure 5.19. The breakage extent for all attacking methods improves as the density level increases. This is because a higher density level will include more distinct terms and more diverse semantic relationships. Hence, the performance of semantic attack methods will improve. However, after reaching 0.5, the breakage levels start to decrease because of the decrease in the number of distinct sub-records in the anchoring chunk. Also, because of the drop in the number of transactions in a cluster and increase in the number of terms in term chunks as a result of increase in the density, the chance of randomly assigning terms to the correct sub-records becomes greater; this can be

explained by the marked improvement in the random attack's performance.

The results in Figure 5.20 shows the impact of different sparsity levels on attacking the protected itemsets from term chunks. In general, the performance of most attacking methods fluctuates. However, MRA with the NGD semantic measure has the best breakage level for most density levels. This is because as mentioned before, NGD can find the semantic relationships for all the terms because it uses the WWW as a corpus. Also, focusing on the term with the best semantic relationship in the selection stage is better than including all the terms that could be unrelated and have affected the total semantic relationship. This is because in denser datasets the terms in one transaction would not have similar semantic relatedness.

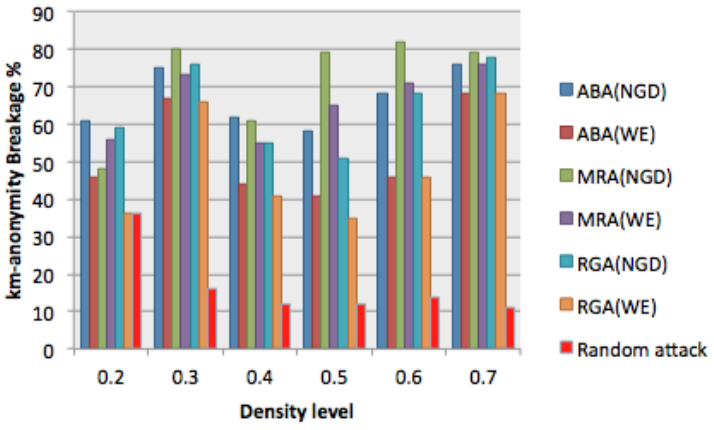



**Figure 5.20: $k^m$-anonymity breakage of attacking term chunks with different data densities.**

### 5.3.5 Effect of max cluster size

Because the max cluster size is one of the parameters used by the disassociation, this section illustrates how our algorithms work on the anonymised transactions produced

by the disassociation method with various max cluster sizes. To evaluate the impact of the max cluster size on attacking performance, we test the max cluster size from $k^2$ to $k^6$ with the $k$ value fixed at 2 and density level at 0.30.

Figure 5.21 compares the accuracy of our algorithms at various max cluster sizes. In general, larger sizes allow for more transactions in a cluster, and this negatively affects the accuracy of all attacking methods. This is because the chance to associate terms with the wrong sub-records increases.

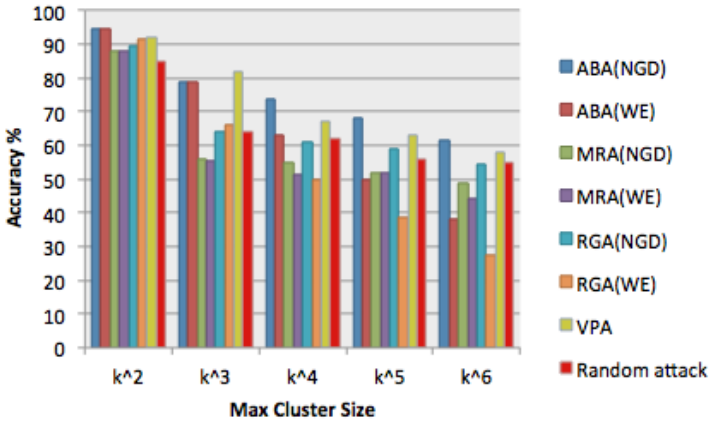



**Figure 5.21: Overall accuracy of the attacking methods with different max cluster sizes.**

The extent of reconstruction in terms of proportion of reconstruction for the original transactions is illustrated in Figure 5.22. With an increase in the size of clusters, the reconstructed transactions become semantically less similar to the original transactions. In larger clusters, the number of transactions is large, increasing the possibility of incorrectly combining terms into sub-records.

In Figure 5.23, we evaluate the effectiveness of our attacking methods on the transac-

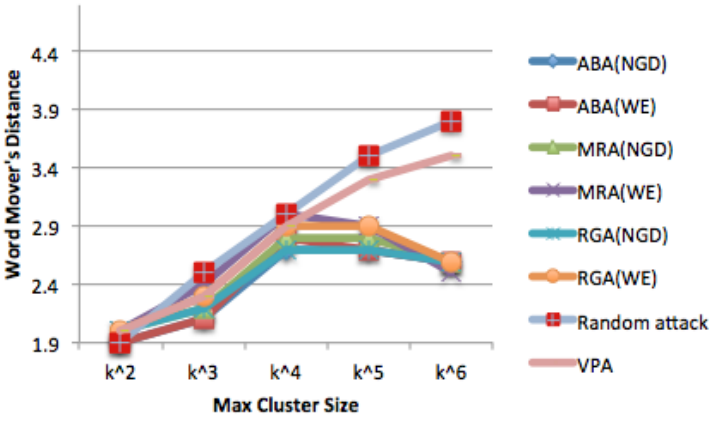



**Figure 5.22: Overall WMD of the attacking methods with different max cluster sizes.**

tion breakage of transactions. Increasing clusters sizes has different impacts on attacking record chunks and term chunks, which we discuss further in the next sections.

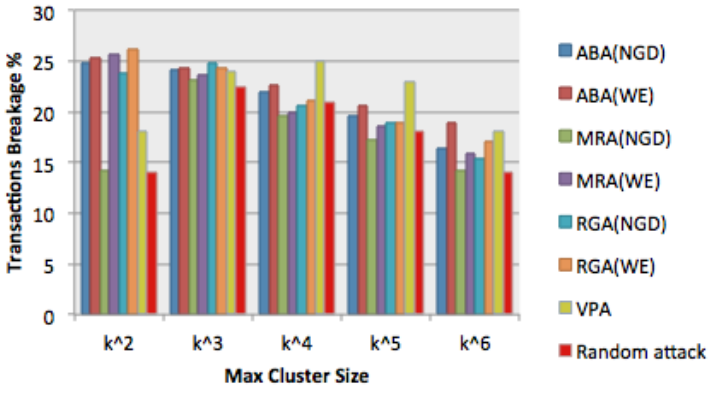



**Figure 5.23: Overall transaction breakage of the attacking methods with different max cluster sizes.**

Figure 5.24 illustrates the overall $k^m$-anonymity breakage of the attacking methods with different max cluster sizes. As mentioned earlier, larger sizes allow for more transactions in a cluster, hence affecting the performance of all methods, which means

the breakage percentages decrease slightly as the size increases.

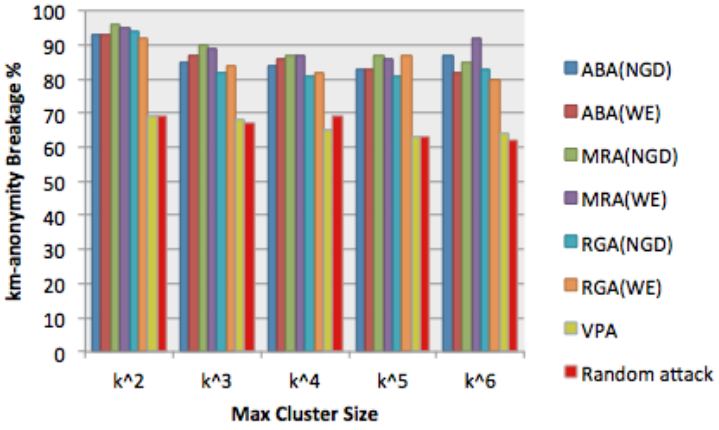



**Figure 5.24: Overall $k^m$-anonymity breakage of the attacking methods with different max cluster sizes.**

### 5.3.6 Record chunks attack

Figure 5.25 shows the impact of increasing max cluster size on the accuracy of attacking record chunks. Increasing the size of cluster leads to more record chunks, which makes attacking them more difficult. This explains the drop in accuracy for all methods.

For the transactions breakage for attacking record chunks, Figure 5.26 shows how the different max sizes of clusters affect the breakage. With more transactions in a cluster, the percentage of having similar sub-records in the anchoring chunk decreases. Hence, the breakage percentages improve as the sizes become larger.

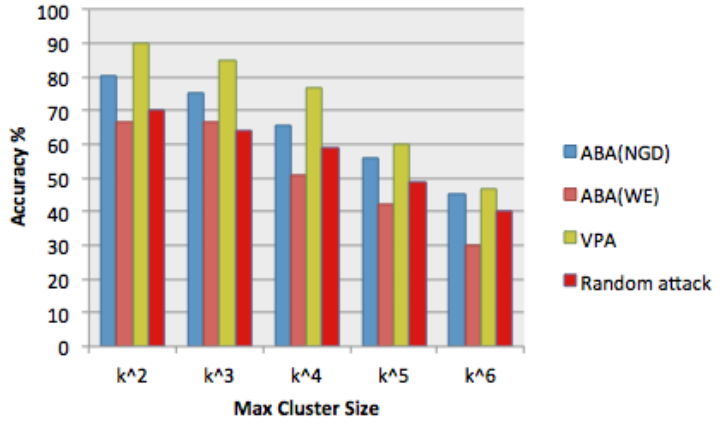



**Figure 5.25: Accuracy of attacking record chunks with different max cluster sizes.**

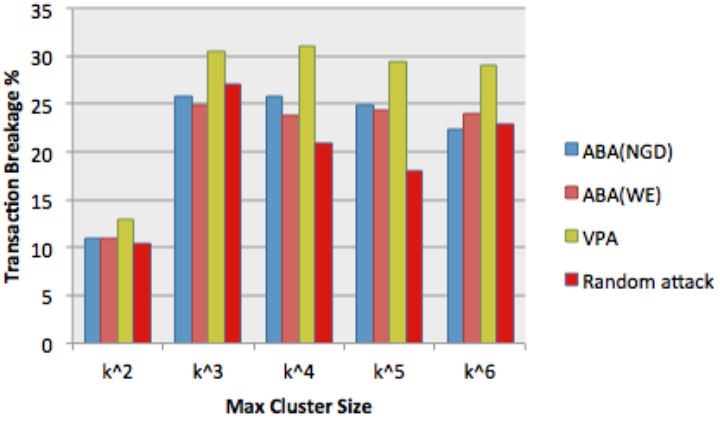



**Figure 5.26: Transaction breakage of attacking record chunks with different max cluster sizes.**

In Figure 5.27, we evaluate the impact of max cluster size on the $k^m$-anonymity breakage. As the size of clusters increases, the number of attacked protected itemsets is slightly reduced.

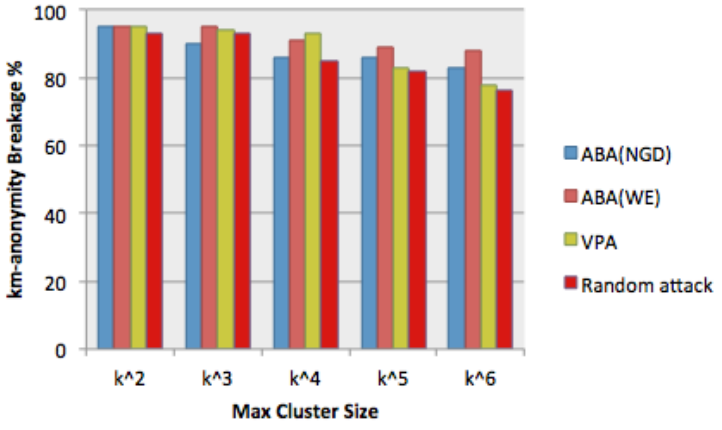



**Figure 5.27: $k^m$-anonymity breakage of attacking record chunks with different max cluster sizes.**

### 5.3.7 Term chunks attack

Figure 5.28 shows the accuracy of our attacking algorithms on term chunks. In general, the accuracy decreases with increasing clusters sizes. Also, because a larger cluster allows for more topics to be included in one cluster, the difference between the performance for different semantic measures in one method is obvious. When the sizes increase, the attacking methods with NGD have a better accuracy than the attacking methods with WE.

The breakage level in terms of transactions breakage is illustrated in Figure 5.29. Increasing cluster size allows for more terms to be in record chunks, which also means fewer terms are in term chunks. This is because larger clusters have more terms meaning that the chance of term appearing in a cluster is likely to increase. As a result, the frequency for some terms may increased, leading to them being moved from term chunks to record chunks. Therefore, the chance to break transactions from term chunks decreases with larger max cluster sizes for all methods.

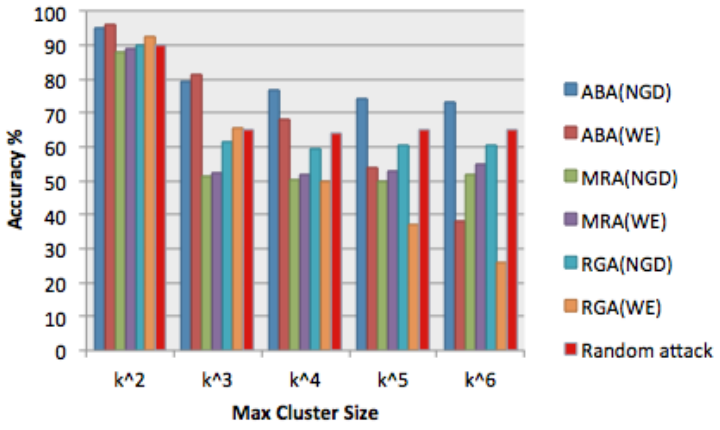



**Figure 5.28: Accuracy of attacking term chunks with different max cluster sizes**

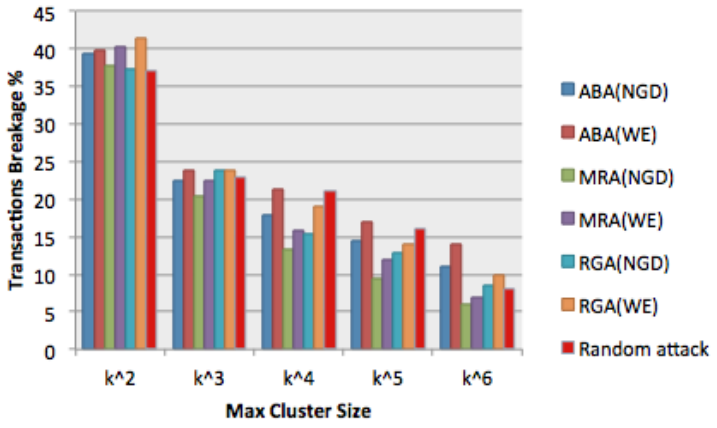



**Figure 5.29: Transaction breakage of attacking term chunks with different max cluster sizes.**

The results in Figure 5.30 show the impact of different max cluster sizes on attacking the protected itemsets from the term chunks. The performances of most attacking methods fluctuate. However, increasing cluster size means more diverse semantic relationships can be found in a cluster. NGD depends on the WWW as a corpus, and as a result, it has better coverage for all terms. This explains why the methods with the NGD perform better in larger clusters than the methods with WE. Also, in larger clusters, concentrating on the term with the best semantic relationship in the selection

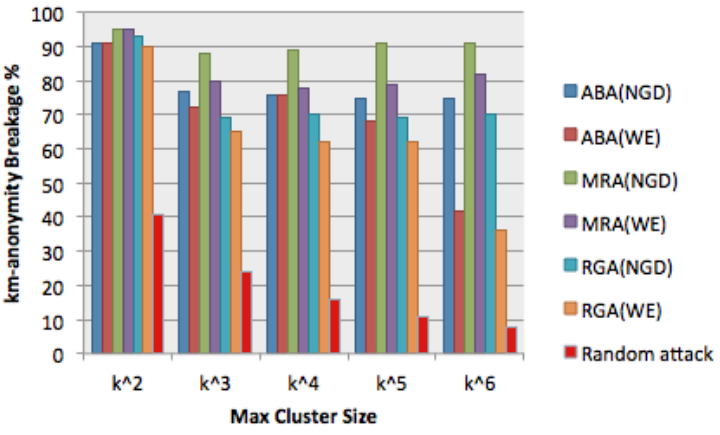



**Figure 5.30: $k^m$-anonymity breakage of attacking term chunks with different max cluster sizes.**

stage performs better than including all the terms which may be unrelated. This explains the outstanding performance for the MRA method for both NGD and WE in larger clusters.

## 5.4 Summary

This chapter presented an evaluation of our attacking methods by using real-world datasets. We studied the effectiveness of our algorithms in two ways. The first focused on how the algorithms break the $k^m$-anonymity requirement in the disassociated dataset by measuring how many transactions and protected itemsets for which their privacy has been broken. In the second, we examined the effectiveness of our attacking methods by measuring how correct our reconstructions are.

As a conclusion, when the dataset is dense, the ABA method using both NGD and WE produced better results for reconstruction accuracy and WMD. Also, it has better

results regarding transaction breakage than other methods. With higher densities, all semantic attacking methods that used the NGD measure have a higher accuracy for attacking record and term chunks than the same methods that used WE. This is because WE may not cover all the terms. Hence, when the density is higher, the methods with NGD outperform the methods with WE in terms of accuracy. In general, even with a high density or high $k$ value, using semantic relationships between terms helps breach the privacy of a disassociated dataset. In addition, the performance of the VPA in most measure gives better results when the number of transactions in a cluster is small and the density is high. However, this method is only effective in attacking record chunks. Also, the attacking performance for all methods, even a random attack, is improved when increasing the $k$ value.

However, our semantic attack is very effective when the disassociated dataset has a high level of sparsity and the size of clusters is small. Therefore, to protect the privacy of disassociated dataset and prevent the threat of semantic attack, a data owner needs to evaluate the effect of the chosen values of disassociation parameters and determine if they provide the promised protection before releasing data. As a suggested solution, this can be achieved by including our proposed approach in the anonymisation process to improve the disassociation method. So, before publishing and after disassociating transactions, a semantic attack can be applied on the disassociated transactions. If the semantic attack can still be a threat to the privacy of data, then the values of disassociation parameters need to be changed. This process can be repeated until the data owner is satisfied with the balance between the protection and data utility. Otherwise, another anonymisation method can be applied or the data owner can decide not to publish the data, because it is not protected adequately.

# *Chapter 6*

# Conclusion

In this chapter, we summarise and conclude the thesis, and then we discuss possibilities for future research.

## 6.1 Research Summary

Driven by mutual benefit, data concerning individuals has been collected and published extensively by a range of organisations. However, individuals' privacy can be violated when data is published without being anonymised. One method for anonymising data is the disassociation method. This method anonymises data by dissociating the links between data items that are vulnerable to attacks but does so without changing any data so that the utility of the data is preserved. In this research, we studied how safe the released data is, when it is anonymised by the disassociation method.

In chapter one, we examined privacy issues with disassociation and saw that although this method can be used to protect data privacy, it does not consider the meaning of terms in a transaction and the semantic relationships that may exist between them. Our hypothesis is that disassociation may not provide adequate protection for transactional data and that an adversary could connect partitioned terms over chunks based on the semantic relationships that exist between the terms in a transaction.

In chapter two, we discussed data privacy and how an attacker can use various types of background knowledge to compromise data privacy, focusing on linkage, inference, and minimality attacks. Then, we studied some related techniques for protecting data privacy as well as some privacy models. We discussed a number of methods for anonymising data: generalisation, suppression, anatomisation, disassociation and perturbation. After that, we illustrated how the concept of semantic relationship can be used to breach data privacy, and we presented a classification of existing works.

We examined the disassociation method in chapter three. First, we illustrated an issue of how clusters smaller than the required size should be handled in the horizontal partitioning. Then, we proposed three strategies for implementing horizontal partitioning to deal with this issue: suppression, adding and remaining-list.

In chapter four, we introduced our approach to semantic attack. We explained the two steps of our attacking approach: scoring and selection. In the scoring stage, we used two semantic measures: normalised Google distance and word embedding, to find the semantic relationships among the terms in disassociated datasets. The semantic scores that resulted from this stage were used in the selection stage.

In the selection stage, we proposed four methods to choose how to add sub-records and terms to each other. The averaging-based attack (ABA) takes into account the semantic scores of all the terms in the anchoring chunk to choose the right transaction to adding terms to. The most-related attack (MRA) considers just the term with the closest semantic score in the anchoring chunk to determine the best sub-record it should add terms to. The terms are divided into two groups based on the relative semantic scores in the related group attack (RGA): there is a related group and non-related group, and only the terms in the related group are considered to select the sub-record to add terms to. The vertical partitioning attack (VPA) uses the possible permutations between

record chunks instead of semantic scores then tests each permutation by applying the $k^m$-anonymity condition to execute the vertical partitioning. Based on the results of the vertical partitioning, the record chunks are combined with each other.

In chapter five, we presented an evaluation of our attacking methods. Our results showed that the semantic relationships that exist among the terms could be exploited by an attacker, hence threatening the privacy of disassociated datasets. The approach can reconstruct different chunks with around 60% accuracy and can break over 70% of protected itemsets. This illustrates that the disassociation method is not safe in terms of data privacy protection when the semantic relationships among the terms are exploited.

### 6.1.1 Future Research

The proposed approach in this research can be considered as a real privacy threat on the disassociated dataset. Therefore, if an attacker is able to find and exploit the semantic relationships between terms in disassociated transactions, then he may reconstruct the transactions correctly with an accuracy of 60%. This potential privacy threat can affect the reliability of using the disassociation method to anonymise data. Therefore, this may instigate the data owner to avoid the disassociation method or individuals will avoid sharing their real data with organisations that use the disassociation method. This proposed approach has a number of future directions in which to expand this work and improve it.

- Improving the scoring techniques:

  In the scoring stage, we used normalised Google distance and word embedding to score semantic relationships among the terms. However, each measure has some limitations that can affect the accuracy of semantic relationship scoring. For normalised Google distance, because it depends on page counts, rare terms

will be more likely to return a smaller number of pages than other terms. So even if these rare terms are closely related, their NGD score will be smaller than some that are less related. As for word embedding, it is not affected by the rare terms issue, but to find the semantic relationships for a pair of terms, the two terms should already be included in the training corpus. This measure could be improved by training the corpus on specific topics to ensure that all terms are included.

- Improving term chunks attack:

  The disassociation considers infrequent terms as vulnerable terms and protects them by placing them in term chunks, In our attacking approach, we consider the semantic relationships between a term from a term chunk and the terms in the anchoring chunks. One possible way to improve attacks on term chunks is to develop a clustering process for the terms in the term chunk by using the semantic relationship then associating the clustered terms to the reconstructed record chunks.

- Measuring confidence:

  Our approach depends on the semantic relationships between terms to perform the semantic attack. However, these relationships are approximated based on how the semantic measure calculates the distance between two terms. Hence, when a term is associated with a sub-record, it would be more useful to determine the degree of certainty that this sub-record is the correct transaction for this term.

# Bibliography

[1] Charu C Aggarwal and S Yu Philip. A general survey of privacy-preserving data mining models and algorithms. In *Privacy-preserving data mining*, pages 11–52. Springer, 2008.

[2] Dakshi Agrawal and Charu C Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 247–255, 2001.

[3] Esma Aïmeur, Gilles Brassard, and Paul Molins. Reconstructing profiles from information disseminated on the internet. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 875–883. IEEE, 2012.

[4] Dzmitry Bahdanau, Tom Bosc, Stanisław Jastrzębski, Edward Grefenstette, Pascal Vincent, and Yoshua Bengio. Learning to compute word embeddings on the fly. *arXiv preprint arXiv:1706.00286*, 2017.

[5] Amir Bakarov. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*, 2018.

[6] Tanmay Basu and CA Murthy. Semantic relation between words with the web as information source. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 267–272. Springer, 2009.

[7] Montserrat Batet and David Sánchez. Semantic disclosure control: semantics meets data privacy. *Online Information Review*, 2018.

[8] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40, 2009.

[9] Ruth Brand. Microdata protection through noise addition. In *Inference control in statistical databases*, pages 97–116. Springer, 2002.

[10] Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener. *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part I*, volume 4051. Springer, 2006.

[11] Keke Chen and Ling Liu. Privacy preserving data classification with rotation perturbation. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 4–pp. IEEE, 2005.

[12] Keke Chen and Ling Liu. Geometric data perturbation for privacy preserving outsourced data mining. *Knowledge and information systems*, 29(3):657–695, 2011.

[13] Richard Chow, Philippe Golle, and Jessica Staddon. Detecting privacy leaks using corpus-based association rules. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 893–901, 2008.

[14] Richard Chow, Ian Oberst, and Jessica Staddon. Sanitization's slippery slope: the design and study of a text revision assistant. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–11, 2009.

[15] Rudi L Cilibrasi and Paul MB Vitanyi. The google similarity distance. *IEEE Transactions on knowledge and data engineering*, 19(3):370–383, 2007.

[16] Valentina Ciriani, S De Capitani Di Vimercati, Sara Foresti, and Pierangela Samarati. $\kappa$-anonymity. In *Secure data management in decentralized systems*, pages 323–353. Springer, 2007.

[17] Chris Clifton and Don Marks. Security and privacy implications of data mining. In *ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, pages 15–19. Citeseer, 1996.

[18] Graham Cormode, Divesh Srivastava, Ninghui Li, and Tiancheng Li. Minimizing minimality and maximizing utility: analyzing method-based attacks on anonymized data. *Proceedings of the VLDB Endowment*, 3(1-2):1045–1056, 2010.

[19] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. Anonymizing bipartite graph data using safe groupings. *the VLDB Journal*, 19(1):115–139, 2010.

[20] Tore Dalenius and Steven P Reiss. Data-swapping: A technique for disclosure control. *Journal of statistical planning and inference*, 6(1):73–85, 1982.

[21] Tom De Nies, Christian Beecks, Wesley De Neve, Thomas Seidl, Erik Mannens, and Rik Van de Walle. Towards named-entity-based similarity measures: Challenges and opportunities. In *Proceedings of the 7th International Workshop on Exploiting Semantic Annotations in Information Retrieval*, pages 9–11, 2014.

[22] Inderjit S Dhillon and Suvrit Sra. Generalized nonnegative matrix approximations with bregman divergences. In *NIPS*, volume 18. Citeseer, 2005.

[23] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.

[24] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.

[25] Khaled El Emam and Fida Kamal Dankar. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, 15(5):627–637, 2008.

[26] Alexandre Evfimievski, Johannes Gehrke, and Ramakrishnan Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222, 2003.

[27] Csilla Farkas and Sushil Jajodia. The inference problem: a survey. *ACM SIGKDD Explorations Newsletter*, 4(2):6–11, 2002.

[28] Christiane Fellbaum. Wordnet. In *Theory and applications of ontology: computer applications*, pages 231–243. Springer, 2010.

[29] Dan Frankowski, Dan Cosley, Shilad Sen, Loren Terveen, and John Riedl. You are what you say: privacy risks of public mentions. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 565–572, 2006.

[30] Benjamin CM Fung, Ke Wang, Rui Chen, and Philip S Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (Csur)*, 42(4):1–53, 2010.

[31] Benjamin CM Fung, Ke Wang, and Philip S Yu. Top-down specialization for information and privacy preservation. In *21st international conference on data engineering (ICDE'05)*, pages 205–216. IEEE, 2005.

[32] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273, 2008.

[33] Olga Gkountouna. A survey on privacy preservation methods. Technical report, Technical Report, Knowledge and Database Systems Laboratory, NTUA, 2011.

[34] Jorge Gracia and Eduardo Mena. Web-based measure of semantic relatedness. In *International Conference on Web Information Systems Engineering*, pages 136–150. Springer, 2008.

[35] Virgil Griffith and Markus Jakobsson. Messinâwith texas deriving motherâs maiden names using public records. In *International Conference on Applied Cryptography and Network Security*, pages 91–103. Springer, 2005.

[36] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[37] Kristina Gulordava and Marco Baroni. A distributional similarity approach to the detection of semantic change in the google books ngram corpus. In *Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics*, pages 67–71, 2011.

[38] Songtao Guo and Xintao Wu. On the use of spectral filtering for privacy preserving data mining. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 622–626, 2006.

[39] Abdelkader Hameurlain, Josef Küng, and Roland Wagner. *Transactions on Large-Scale Data-and Knowledge-Centered Systems I*, volume 5740. Springer, 2009.

[40] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. Semantic similarity from natural language and ontology analysis. *Synthesis Lectures on Human Language Technologies*, 8(1):1–254, 2015.

[41] Yeye He, Siddharth Barman, and Jeffrey F Naughton. Preventing equivalence attacks in updated, anonymized data. In *2011 IEEE 27th International Conference on Data Engineering*, pages 529–540. IEEE, 2011.

[42] Yeye He and Jeffrey F Naughton. Anonymization of set-valued data via top-down, local generalization. *Proceedings of the VLDB Endowment*, 2(1):934–945, 2009.

[43] Steffen Hedegaard, Søren Houen, and Jakob Grue Simonsen. Lair: A language for automated semantics-aware text sanitization based on frame semantics. In *2009 IEEE International Conference on Semantic Computing*, pages 47–52. IEEE, 2009.

[44] Angelos Hliaoutakis. Semantic similarity measures in mesh ontology and their application to information retrieval on medline. *Master's thesis*, 2005.

[45] Danesh Irani, Steve Webb, Kang Li, and Calton Pu. Large online social footprints–an emerging threat. In *2009 International conference on computational science and engineering*, volume 3, pages 271–276. IEEE, 2009.

[46] Aminul Islam and Diana Inkpen. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(2):1–25, 2008.

[47] Vijay S Iyengar. Transforming data to satisfy privacy constraints. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 279–288, 2002.

[48] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Third IEEE international conference on data mining*, pages 99–106. IEEE, 2003.

[49] Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. Random-data perturbation techniques and privacy-preserving data mining. *Knowledge and Information Systems*, 7(4):387–414, 2005.

[50] Tom Kenter and Maarten De Rijke. Short text similarity with word embeddings. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pages 1411–1420. ACM, 2015.

[51] Daniel Kifer. Attacks on privacy and definetti's theorem. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 127–138, 2009.

[52] Daniel Kifer and Johannes Gehrke. Injecting utility into anonymized datasets. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 217–228, 2006.

[53] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966. PMLR, 2015.

[54] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60, 2005.

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

[56] Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 18(8):1138–1150, 2006.

[57] Junqiang Liu and Ke Wang. Anonymizing transaction data by integrating suppression and generalization. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 171–180. Springer, 2010.

[58] Grigorios Loukides, Aris Gkoulalas-Divanis, and Bradley Malin. Coat: Constraint-based anonymization of transactions. *Knowledge and Information Systems*, 28(2):251–282, 2011.

[59] Grigorios Loukides, John Liagouris, Aris Gkoulalas-Divanis, and Manolis Terrovitis. Disassociation for electronic health record privacy. *Journal of biomedical informatics*, 50:46–61, 2014.

[60] Adrienn Lukács. what is privacy? the history and definition of privacy. 2016.

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

[62] David Malvern and Brian Richards. Measures of lexical richness. *The encyclopedia of applied linguistics*, 2012.

[63] Sergio Martı, Aida Valls, David Sánchez, et al. Semantically-grounded construction of centroids for datasets with textual attributes. *Knowledge-Based Systems*, 35:160–172, 2012.

[64] David J Martin, Daniel Kifer, Ashwin Machanavajjhala, Johannes Gehrke, and Joseph Y Halpern. Worst-case background knowledge for privacy-preserving data publishing. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 126–135. IEEE, 2007.

[65] Sergio Martínez, David Sánchez, and Aïda Valls. Towards k-anonymous non-numerical data via semantic resampling. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 519–528. Springer, 2012.

[66] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[67] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[68] George A Miller and Walter G Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

[69] Norman Mooradian. The importance of privacy revisited. *Ethics and information technology*, 11(3):163–174, 2009.

[70] C Nalini and AR Arunachalam. A study on privacy preserving techniques in big data analytics. *International Journal of Pure and Applied Mathematics*, 116(10):281–286, 2017.

[71] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

[72] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large datasets (how to break anonymity of the netflix prize dataset). *University of Texas at Austin*, 2008.

[73] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *2009 30th IEEE symposium on security and privacy*, pages 173–187. IEEE, 2009.

[74] Guillermo Navarro-Arribas, Vicenç Torra, Arnau Erola, and Jordi Castellà-Roca. User k-anonymity for privacy preserving data mining of query logs. *Information Processing & Management*, 48(3):476–487, 2012.

[75] Hyoungmin Park and Kyuseok Shim. Approximate algorithms for k-anonymity. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 67–78, 2007.

[76] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[77] Christo Petrov. 25+ big data statistics - how big it actually is in 2020? urlhttps://techjury.net/blog/big-data-statistics/, year=2021, month=Jan.

[78] C Christine Porter. De-identified data and third party data mining: The risk of re-identification of personal information. *Shidler JL Com. & Tech.*, 5:1, 2008.

[79] Vibhor Rastogi, Dan Suciu, and Sungho Hong. The boundary between privacy and utility in data publishing. In *Proceedings of the 33rd international conference on Very large data bases*, pages 531–542. Citeseer, 2007.

[80] M Reza and Somayyeh Seifi. Classification and evaluation the ppdm techniques by using a data modification-based framework. *IJCSE, Vol3. No2 Feb*, 2011.

[81] Luc Rocher, Julien M Hendrickx, and Yves-Alexandre De Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature communications*, 10(1):1–9, 2019.

[82] Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[83] Ira S Rubinstein and Woodrow Hartzog. Anonymization and risk. *Wash. L. Rev.*, 91:703, 2016.

[84] Pierangela Samarati and Latanya Sweeney. Generalizing data to provide anonymity when disclosing information. In *PODS*, volume 98, pages 275487–275508. Citeseer, 1998.

[85] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. 1998.

[86] David Sánchez, Montserrat Batet, and Alexandre Viejo. Detecting sensitive information from textual documents: an information-theoretic approach. In *International Conference on Modeling Decisions for Artificial Intelligence*, pages 173–184. Springer, 2012.

[87] David Sánchez, Montserrat Batet, and Alexandre Viejo. Detecting term relationships to improve textual document sanitization. In *PACIS*, page 105, 2013.

[88] Government Digital Service. Data protection, Sep 2015.

[89] Jianhua Shao and Hoang Ong. Exploiting contextual information in attacking set-generalized transactions. *ACM Transactions on Internet Technology (TOIT)*, 17(4):40, 2017.

[90] H Jeff Smith, Tamara Dinev, and Heng Xu. Information privacy research: an interdisciplinary review. *MIS quarterly*, pages 989–1015, 2011.

[91] Rhys Smith and Jianhua Shao. Privacy and e-commerce: a consumer-centric perspective. *Electronic Commerce Research*, 7(2):89–116, 2007.

[92] Michal Sramka, Reihaneh Safavi-Naini, and Jörg Denzinger. An attack on the privacy of sanitized data that fuses the outputs of multiple data miners. In *2009 IEEE International Conference on Data Mining Workshops*, pages 130–137. IEEE, 2009.

[93] Jessica Staddon, Philippe Golle, and Bryce Zimny. Web-based inference detection. In *USENIX Security Symposium*, 2007.

[94] Jessica Su, Ansh Shukla, Sharad Goel, and Arvind Narayanan. De-anonymizing web browsing data with social networks. In *Proceedings of the 26th international conference on world wide web*, pages 1261–1269, 2017.

[95] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):571–588, 2002.

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

[97] David G Taylor, Donna F Davis, and Ravi Jillapalli. Privacy concern and online personalization: The moderating effects of information control and compensation. *Electronic commerce research*, 9(3):203–223, 2009.

[98] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. Anonymity in unstructured data. In *Proc. of International Conference on Very Large Data Bases (VLDB)*, 2008.

[99] Manolis Terrovitis, Nikos Mamoulis, and Panos Kalnis. Privacy-preserving anonymization of set-valued data. *Proceedings of the VLDB Endowment*, 1(1):115–125, 2008.

[100] Manolis Terrovitis, Nikos Mamoulis, John Liagouris, and Spiros Skiadopoulos. Privacy preservation by disassociation. *Proceedings of the VLDB Endowment*, 5(10):944–955, 2012.

[101] Frank M Tuerkheimer. The underpinnings of privacy protection. *Communications of the ACM*, 36(8):69–73, 1993.

[102] Muhamed Turkanovic, Tatjana Welzer Druzovec, and Marko Hölbl. Inference attacks and control on database structures. *TEM Journal*, 4(1):3, 2015.

[103] Vassilios S Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *ACM Sigmod Record*, 33(1):50–57, 2004.

[104] Ke Wang and Benjamin CM Fung. Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 414–423, 2006.

[105] Ke Wang, S Yu Philip, and Sourav Chakraborty. Bottom-up generalization: A data mining solution to privacy protection. In *ICDM*, volume 4, pages 249–256, 2004.

[106] Ke Wang, Yabo Xu, Ada WC Fu, and Raymond CW Wong. ff-anonymity: When quasi-identifiers are missing. In *2009 IEEE 25th International Conference on Data Engineering*, pages 1136–1139. IEEE, 2009.

[107] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *2010 ieee symposium on security and privacy*, pages 223–238. IEEE, 2010.

[108] Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, and Jian Pei. Minimality attack in privacy preserving data publishing. In *Proceedings of the 33rd international conference on Very large data bases*, pages 543–554, 2007.

[109] Raymond Chi-Wing Wong, Jiuyong Li, Ada Wai-Chee Fu, and Ke Wang. ($\alpha$, k)-anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 754–759, 2006.

[110] Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David R O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21:209, 2018.

[111] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and effective privacy preservation. In *VLDB*, volume 6, pages 139–150, 2006.

[112] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, pages 229–240, 2006.

[113] Yabo Xu, Ke Wang, Ada Wai-Chee Fu, and Philip S Yu. Anonymizing transaction databases for publication. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 767–775, 2008.

[114] Chao Yao, X Sean Wang, and Sushil Jajodia. Checking for k-anonymity violation by views. In *Proceedings of the 31st international conference on Very large data bases*, pages 910–921. Citeseer, 2005.

[115] Lei Zhang, Sushil Jajodia, and Alexander Brodsky. Information disclosure under realistic assumptions: Privacy versus optimality. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 573–583, 2007.

[116] Qing Zhang, Nick Koudas, Divesh Srivastava, and Ting Yu. Aggregate query answering on anonymized tables. In *2007 IEEE 23rd international conference on data engineering*, pages 116–125. IEEE, 2007.

[117] Zijian Zheng, Ron Kohavi, and Llew Mason. Real world performance of association rule algorithms. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 401–406, 2001.

[118] Athanasios Zigomitros, Agusti Solanas, and Constantinos Patsakis. The role of inference in the anonymization of medical records. In *2014 IEEE 27th International Symposium on Computer-Based Medical Systems*, pages 88–93. IEEE, 2014.