Publishing Privacy Sensitive Open Data

using An Automated Decision Support Tool

Andrei Manta

Publishing Privacy Sensitive Open Data

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

by

Andrei Manta
born in Bucharest, Romania

Multimedia and Signal Processing Group
Department of Intelligent Systems
Faculty EEMCS, Delft University of Technology
Delft, the Netherlands
www.ewi.tudelft.nl

Center for Advanced Studies
IBM Netherlands
Johan Huizingalaan 765
Amsterdam, the Netherlands
www.ibm.com/nl/nl

Publishing Privacy Sensitive Open Data

Author: Andrei Manta
Student id: 1520172
Email: manta.s.andrei@nl.ibm.com

Abstract

In recent years, the idea of Open Data has gained popularity, mainly due to the initiative of the president of the USA, Barack Obama. He started a promotion campaign for an Open Government and ordered government data to be made available as Open Data. In essence, Open Data is data published online which can be used and republished without restrictions from mechanisms of control. It is believed that most government data can be leveraged as fuel for innovation. A survey by TNO for the Dutch policymakers concluded that Open Data, including government data, has big economic value.

The motivation and intuition behind our research is that the publication of large sets of (linked) Open Data may lead to unforeseen breaches in data sensitivity and privacy. Therefore, we conducted interviews with policy makers who are responsible for the publication of Open Data. From the interviews, we conclude that there is no clear view on the possible issues surrounding the publication of Open Data. The data publishing task is usually delegated to institutions such as Statistics Netherlands(CBS). Moreover, a literature survey demonstrated that current research focuses solely on data privacy, ignoring other forms of unwanted publication such as data sensitivity.

The unclear view on possible issues and the focus of literature on data privacy have motivated us to propose a new data publishing process, supported by an automated decision support tool. For this system we present an architecture and a reference implementation. Furthermore, we propose a more extensive definition of data sensitivity. We also present definitions for privacy and utility metrics and we use these metrics to compare anonymization algorithms.

Keywords. *open data, privacy, privacy preserving data publishing, framework, publishing process*

Thesis Committee:

Chair:	Prof. Dr. Ir. R. L. Lagendijk, Faculty EEMCS, TU Delft
University supervisor:	Dr. Zekeriya Erkin, Faculty EEMCS, TU Delft
Company supervisor:	Ing. Bram Havers, IBM Netherlands, CEWM
Company supervisor:	Drs. Robert-Jan Sips, IBM Netherlands, CAS

Preface

It has been a long journey but it finally came to an end. And a good one, I might add. Getting started has not been easy. We started thinking about the privacy of data and eventually came to talk about safely publishing sensitive government data as Open Data.

This work is part exploratory and part outrageous. It is exploratory because it is a new direction for both TU Delft and IBM. In Delft, the experts use cryptography to keep data sets safe. At IBM, they have an interest for the domain of anonymized open data publishing. The funny thing is that this makes me – the one who needs to eventually be evaluated – the expert in the field for this group. It is outrageous, because it points fingers at big things already set in motion. Open Data, Linked Open Data and anything that is related to the previous two. I see the point of having enriched, inter-linked and freely accessible data. It's good for research and it's good for business. But, as always, we forget about things such as privacy and safety. We can just protect those afterwards – right?

Open Data and Co. are already here. A lot of governments now start to open up their data gates and flood the world with even more data. The only thing left for us to do is to protect whatever we can before it is too late. The goal of this thesis is to lend a helping hand in the process of sensitive data sanitization – through anonymization. We try to set the context of how governments currently open up their data. We then propose a publishing process which uses an automated decision support tool to help the data publisher mitigate the risks. We give the design details for this system and test to see how it performs.

It might be weird to say it here, but it is especially valid for this situation: the road to hell is paved with good intentions. Data which initially seems harmless and released for the good of the people, can eventually be combined and misused to commit burglary and other crimes. Everyone is so worried about privacy that they forget to stop and think if something worse exists. It does and it's called data sensitivity.

Acknowledgements

First of all I would like to thank my wife, Oana, for giving me the power to bring this project to a successful end. There have been moments when I no longer saw the finish line, but she managed to keep me on track. I would also like to thank my family for their trust and support.

I am grateful to Inald Lagendijk and Bram Havers for making this research project possible. It took months of searching to find the perfect match between IBM, TU Delft and myself. To this end, I would also like to thank Mathijs de Weerdt for pointing me in the direction of Inald.

I want to thank my daily supervisors, Robert-Jan Sips, Zekeriya Erkin and Bram Havers for their continuous feedback on my work. They guided me in the right direction and from time to time, whenever necessary, they took me out of the Mariana Trench and placed me on top of Mt. Everest - the big picture. We had some interesting discussions over the months that shaped the solution into what it is today. Thank you for your critical view on my work. I can honestly say I learned a lot.

Last but not least, I would like to thank all my colleagues at IBM and in Delft for creating a good and fun atmosphere to work in.

Andrei Manta
Delft, The Netherlands
November 29, 2013

Contents

List of Figures

Chapter 1

Introduction

In recent years, the idea of Open Data has gained popularity. In essence, Open Data is data published online which can be used and republished without restrictions from mechanisms of control. The movement started to gain popularity in the USA, as part of president's Barack Obama campaign for an Open Government. He wanted to increase government transparency by making the public information more easily accessible. He ordered government data to be made available as Open Data.

From [27] we can see that different countries have a different approach to the Open Data, each with their own motivation for adoption. For The Netherlands, the goal is to open up the Dutch Ministry of Economic Affairs and the Ministry of Interior and Kingdom Relations data by 2015 [1, 24]. The right of the citizens to be able to request information is mandated by the *Freedom of Information Law* (in The Netherlands this is called *Wet Openbaarheid van Bestuur*). Having that data published online means that it is easier to find, easier to access and should be computer readable.

The belief is that most government data can be leveraged as fuel for innovation [16]. A survey by TNO [27] for the Dutch policymakers concluded that Open Data, including government data, has big economic value. To further increase this value over time, we observe, from the Dutch Open Data website[1], that they both open up data and encourage people to use and to contribute to the existing database.

The motivation and intuition behind our research is that the increase in data availability, by means of publishing large data sets, will lead to unforeseen breaches in data sensitivity and privacy. To better understand the current situation, we conducted interviews with policy makers who are responsible for the publication of Open Data. These include Rijkswaterstaat (responsible for the roads, water and infrastructure in The Netherlands), Kadaster (responsible for parcel information), Amsterdam Economic Board (responsible for strategies for the economic development of the Amsterdam region) and Statistics Netherlands - CBS (supplier of most statistical information in The Netherlands). From the interviews, we conclude that most institutions delegate the data publishing process to CBS. This is a consequence of the fact that most institutions do not have a clear view of possible issues surrounding

[1] https://data.overheid.nl/

the publication of Open Data. But with a deadline for 2015 to open up a big part of the government data, this creates a bottleneck on the publisher's side (e.g. CBS).

Moreover, the literature survey we conducted [22] demonstrates that the current focus of research is on data privacy, mostly ignoring other forms of unwanted publication such as data sensitivity. We therefore propose the more extensive definition of data sensitivity. This covers data privacy and other types of issues that might occur regarding the privacy and security of legal entities or countries.

With the results of this thesis we lay the foundation to a new data publishing process, supported by automated decision support. This system assists policy makers in assessing the risk of publishing data sets.

1.1 Research Questions

Here we present our three main research questions. The first two questions can be answered based on our literature survey [22], while the third question is the main focus of this thesis.

RQ 1 *Why is privacy preserving data publishing necessary when dealing with Open Data?*

First, we would like to understand where publishing privacy sensitive data fits in the context of Open Data. This will help us understand what the challenges are and why extra precautions are necessary.

In order to properly open up the data, guidelines[2] and laws[3] have been created. As we have observed from the different interviews presented in Appendix B, the current rules are not enough to protect the data. The rules can only provide guidelines on how to act, but since they need to cover as many cases as possible, they lack the ability of precisely defining what should and what should not be published. Data sets vary greatly one from another, making it is impossible for the law to properly handle every single situation.

In our literature review [22] we identified two types of publishable data: *sensitive* and *non-sensitive data* (everything that is not sensitive). The type that require protection are the former.

Sensitive data is information that might result in loss of an advantage or level of security if disclosed to others. It may affect the privacy or welfare of an individual, trade secrets of a business or even the security of a nation.

The problems occur when data sets are released that are minimally anonymized to satisfy the legal requirements. The data set itself might then be safe, but the hidden threats may appear when more data sets are combined. Possible problems include the creation of new sensitive patterns in the data, re-identification of individuals by means of exclusion (every record but one exhibit the characteristics a person does *not* have) or by isolation (the record which has all the characteristics a specific individual has).

[2] Guidelines for The Netherlands: `https://data.overheid.nl/handreiking`

[3] In The Netherlands: Wet Openbaarheid van Bestuur

Whether talking about individuals or legal entities, one needs to ensure that their identifying characteristics are not published. The situation can become more complex when these entities and individuals can be grouped into hierarchical structures. Take for example individuals in households, in universities, in schools or employees in enterprises.

To better understand how the data can be combined in a harmful way, we present two examples. One related to sensitive data, the other to privacy sensitive data.

EXAMPLE 1. Our first example is about an app called "Makkie Klauwe"[4]. In essence it shows burglars which houses are easy to break into and give a nice profit. The interesting thing is that it combines relevant public data such as area value, reported problems in the area and how much does the municipality can spend to improve an area. For example it may suggest a house in a good neighbourhood where the streetlight is broken (easier for the thief to break in without being detected).

EXAMPLE 2. Our second example is related to privacy and is the motivation of the paper by L. Sweeney [26]. In Massachusetts the Group Insurance Commission collected medical data about state employees and their families. They assumed the data were anonymous and made them available for research. Sweeney obtained this data set and also bought the voter registration list for Cambridge Massachusetts; by combining these two data sets on ZIP code, birth date and gender, she managed to uniquely identify the medical information of the Governor of Massachusetts. These examples are meant to illustrate how data sets, which on their own pose no threat, can create risks and privacy breaches when combined.

It is interesting to observe that Open Data is not the only data or information related movement. The concern regarding data sensitivity is that most of these movements are complementary, i.e. they increase the amount of available information. Looking into the future, we can see that more data will be freely available, easier to access, and with interlinks and semantics already included (e.g. Linked Data movement [5]). This means that finding meaningful data set combinations becomes easier (to automate) due to the included semantics.

To the best of the author's knowledge, literature is currently focused on data privacy [22] and does not consider data sensitivity. Sensitive data is information that might result in loss of an advantage or level of security if disclosed to others. It may affect the privacy or welfare of an individual, trade secrets of a business or even the security of a nation. We also notice that most solutions only handle a single data set and do not consider what would happen if external information would be used. Breaches to data sensitivity and privacy can take place by combining different data sets. In some cases it has been shown that anonymizing just the current data set is enough, even if the adversary has external information; the only question now is whether the data is still useful or not since the anonymization process has an impact on data utility.

[4] http://www.bramfritz.nl/makkieklauwe/

RQ 2 *How are decisions taken when publishing sensitive data as Open Data?*

After the context has been set, we would like to understand how publishing privacy sensitive data is currently being done. We need to understand what policy makers are currently struggling with, how they overcome certain challenges and find the places where we can improve the process.

From several interviews with representatives of different public institutions of The Netherlands (Appendix B) we see that transforming a data set into Open Data is not easy. The current policy used for opening up data in The Netherlands is the "open tenzij" [24](tr. open unless) rule. This implies that one may publish everything, unless it does not adhere to the law or to the company regulations. We identify two reasons why this method is preferred above data sanitization:

1. No risks - it is easier to just not publish and avoid any risks.
2. Lack of experts - not every company or institution has access to the experts who can actually perform the cleaning and anonymization of the data.

So protection is assured, in most cases, through *secrecy* and not through *anonymization*. The advantage of this approach is that the privacy of the not published data set is guaranteed. When looking at data utility, we can conclude that it is equal to zero since the user has no access to the data and cannot use it. Here we see the inversely proportional relation between privacy and utility. Publish the raw data, then we have maximum possible utility and zero privacy. Do not publish at all, then we have complete privacy and no utility. In short, the law currently enforces secrecy. If the law would change, it might force the publishing of data sets that were previously prohibited under the "open tenzij" rule.

The responsibility to open up the data usually belongs to the department that owns the data, yet, as observed from our interviews (Appendix B), these departments do not always have the right knowledge to deal with the issues surrounding data publication. The main rules these institutions have to follow in The Netherlands are given by the Wet Openbaarheid van Bestuur (WOB)[5] and by the Wet Bescherming Persoonsgegevens(WBP)[6]. In some cases some internal guidelines/rules are also applied (e.g. Rijkswaterstaat) [22]. All in all, these rules only create a minimal safety boundary. As mentioned above, there is an increase in data availability. There are more sources of data which can be combined to create a breach in privacy.

Currently, publishing is based on a combination of rules, experience and intuition [22]. There are three components that make data publishing a challenge:

1. lack of specialists for data publishing.
2. knowledge gap on how to handle the data sanitization process properly within organisations.

[5]Open Data Law: `http://wetten.overheid.nl/BWBR0005252/geldigheidsdatum_29-05-2013`
[6]Privacy Law: `http://wetten.overheid.nl/BWBR0011468/geldigheidsdatum_29-05-2013`

3. the volume of data that needs sanitization. There are only a few institutions which have expertise in the domain (such as CBS, O&S). These institutions are already active with publishing and the Open Data initiative will only increase their workload.

Data Publishing Guidelines To get a better understanding of the thought process behind the current data publishing process promoted by the Dutch government[7], we will be giving a brief summary of each step below.

Step 1 - Why to start with Open Data The first step explains why to begin with Open Data and what the advantages are of Open Data. Tips are also given on how to start and promote this within the organization.

Step 2 - Data set selection The second step is about choosing the datasets to publish. It contains questions which should get the data publisher thinking about the data, whether it is worth publishing and what the risks surrounding the data might be.

Step 3 - The legal check The legal check is the next step. It is a very important step since it can have a big legal impact on the publisher if, for example, there is a breach in someone's privacy.

Step 4 - Organise the publication process The fourth step in the process gives tips on how to organize the publishing process. The point is that data publishing should not be an ad-hoc activity, but should be part of the company's workflow.

Step 5 - Make the data easy to find and accessible The last proposed step is about making the data easy to find and accessible. Preferably the data should be in a machine readable format, have an API and a contact person in case there are any questions about the dataset.

We observe that none of the steps above explain how to actually clean the data set. This leads to our next research question.

RQ 3 *How to anonymize the data?*

This research question is the main focus of this thesis. Because a sensitive data set cannot be released "as is", it first needs to be sanitized by means of anonymization. The problem is that different anonymizations provide different results. We need to investigate what these differences are. Because the question is very broad, we have divided it into three sub-questions.

A Which algorithms should be considered as candidates for anonymization for which type of data, with respect to applicability in practice?

[7] `https://data.overheid.nl/handreiking`, retrieved October 26, 2013

There are many algorithms for anonymization, each being able to sanitize a certain type of data. We first need to make a selection of suitable (in practice applicable) algorithms for each type of data

B How to interpret the measured values for privacy and utility and what guarantees do these values provide?

The side effects of data anonymization are mainly loss of utility to the users and increase in privacy levels. We need to measure these changes. To this end we need metrics that give an interpretable results.

C How does privacy / utility change when the data set is combined with external sources?

The challenge of publishing large data sets was the increase in risk of a breach in data privacy or sensitivity. We would like to be able to understand how this risk changes as more information is available out there.

1.2 Contributions

The area of safe data publishing is very broad. As such, the focus of this thesis will be on privacy. We define the base for a tool that can support the data publisher in the process of publishing information as Open Data. In other words, a tool that gives insight on the risks contained by the data and advice on how to anonymize such data, considering the desired privacy and utility levels. Below a more detailed overview of our contributions.

New Data publishing process We present an improved data publishing process which serves as the foundation towards automated data publishing. Automating the full process cannot be done, since data publishing retains the human factor, but many steps can and should be automated. We explain in the future work how to further automate parts of the process.

Automated decision support system We present the design, architecture and implementation of an automated decision support tool which helps the data publisher better understand the risks and gains. Instead of using one algorithm, the tool evaluates how many algorithms perform on the given data set and let the data publisher make an informed choice.

Analysis of utility and privacy metrics To measure the privacy gain and utility loss, we analyze various metrics and see whether they actually give useful values.

Comparison of anonymization algorithms We provide a more thorough analysis of the algorithms used in the experiments. We explain why certain metric values are achieved based on the inner workings of the algorithms.

1.3 Outline

The thesis is outlined as follows. Related work in privacy preserving data publishing is shown in Chapter 2. In Chapter 3, the steps of the data publishing process are described. The design and implementation details of our system are outlined in Chapter 4. Chapter 5 presents our experimental results. We conclude the thesis in Chapter 6 with a discussion, conclusions to the research questions and future work.

Chapter 2

Background

This chapter is dedicated to presenting background elements necessary for this thesis. For the complete literature survey, please refer to [22].

For our experiments, we chose to use four anonymization algorithms: *k*-anonymity [26], t-closeness [19], (n,t)-closeness [20] and (n,t)-closeness together with *k*-anonymity. These have been selected because they are widely known, used and referred to in the literature. *k*-anonymity has two implementations. One is based on the Incognito [17] algorithm while the other is based on the Mondrian [18] algorithm. The other algorithms have been implemented by extending either Incognito or Mondrian as follows. The t-closeness algorithm extends the Incognito implementation of *k*-anonymity, while (n,t)-closeness and (n,t)-closeness with *k*-anonymity both extend the Mondrian algorithm. Below we briefly present how these algorithms and their implementations work.

2.1 Incognito - How it works

The current implementation does a bottom-up global recoding of the QID combination space (the Cartesian product of the domains of every QID attribute - the full domain of possible QID values). Global recoding consists of replacing every value of an attribute by that value's one level higher generalisation. For example, given the taxonomy tree in Figure 2.1, every value of 0, 1 and 2 in the data set would be replaced by [0,2].

Initially a search lattice in created with the base root equal to $\{0, 0, \ldots, 0\}$ where the number of elements is equal to the QID size. The numbers represent the level to which a value is generalized in the taxonomy tree of the corresponding attribute. For example, given two attributes in the QID, the starting root node in the lattice is $\{0, 0\}$. The algorithm starts to check if first $\{0,0\}$, then $\{1,0\}$, then $\{0,1\}$, then $\{1,1\}$,... satisfy the privacy requirement. From all the anonymizations that satisfy the privacy requirements, the algorithm chooses the one that has the least number of generalisations.

Searching for all combinations is not very efficient. Because of that, heuristics exist that reduce the lattice search space. In the worst case scenario, the heuristics cannot be successful and the whole lattice needs to be searched.

Now, we discus the algorithms implemented using Incognito.

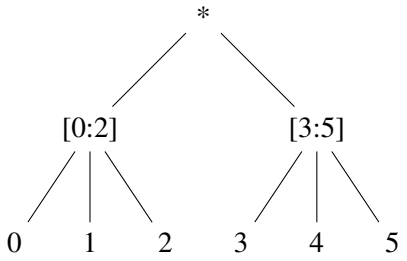


Figure 2.1: Example Taxonomy Tree of an Attribute's Values

Job	Sex	Age	Disease
Engineer	male	35	Hepatitis
Engineer	male	38	Hepatitis
Lawyer	male	38	HIV
Writer	female	35	Flu
Writer	female	35	HIV
Dancer	female	35	HIV
Dancer	female	36	HIV

Table 2.1: Patient table

Job	Sex	Age	Disease
Professional	male	[35-40)	Hepatitis
Professional	male	[35-40)	Hepatitis
Professional	male	[35-40)	HIV
Artist	female	[35-40)	Flu
Artist	female	[35-40)	HIV
Artist	female	[35-40)	HIV
Artist	female	[35-40)	HIV

Table 2.2: 3-anonymous patient table

2.1.1 *k*-anonymity

The algorithm works as follows. Let *qid* represent a QID value combination for a record in a data set. *k*-anonymity only requires that every single *qid* value appears at least *k* times in the data set. This means that the QID values of the records in the data set are generalised in such a way that grouping by QID values generates bins called equivalence classes(EC) of size at least *k*. We can see this in Table 2.1 and Table 2.2. For example {Engineer, male, 35} is generalised to {Professional, male, [35-40)}.

The effect of k-anonymity is that an attacker can link an individual to a record with a maximum probability of 1/k.

2.1.2 t-closeness

The anonymization algorithm t-closeness also uses generalisation of QID values to achieve its privacy requirement. But instead of requiring a minimum group size, it requires a maximum distance between two distributions. A data set is said to achieve t-closeness if for every equivalence class, the distribution of the sensitive values in the EC is within *t* of the distribution of sensitive values in the whole data set. The reasoning behind it is to limit the information gain from an individual EC, compared to the information already gained from the whole data set.

2.2 Mondrian - How it works

The name of the algorithm is inspired by a painting of Mondrian (Figure 2.2). The painting can be seen as a representation of how the algorithm works when clusters are creating in a two-dimensional space.

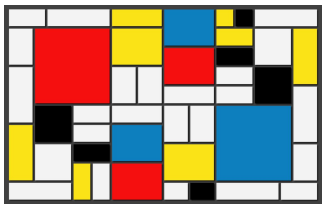


Figure 2.2: Painting by Mondrian

As with the painting, the algorithm partitions a high-dimensional space into regions and insures that a certain privacy requirement is guaranteed.

Initially, all the points(records) start off in one single bin (equivalence class). The algorithm does three things: it selects a dimension to partition on, it selects a value on that dimension to split on and it checks if the split results in a valid cut. A valid cut partitions a region into two smaller regions, both of which adhere to the privacy requirement. In our case, the value to split on is the median since it gives a more uniform partitioning [18].

2.2.1 (n,t)-closeness

(n,t)-closeness builds on top of t-closeness. The main advantage is that it distorts the data less. It requires the distribution of sensitive values for every equivalence class to be within t of a population of size at least n. This large enough population, of size at least n, needs to be a natural superset of its respective EC.

A natural superset of an equivalence class(E_1) is also an equivalence class(E_2) which contains the former by broadening the set value boundaries. Take for example E_1 to be the first EC in table 2.4. E_1 is thus defined as (zipcode='476**', age=[20,29]). A natural superset would then be (zipcode='476**', age=[20,39]) since it "naturally" contains it. If E_1 has a size greater than n, then it already respects the privacy requirement since it is its own natural superset.

Zipcode	Age	Disease	Count
47673	29	Cancer	100
47674	21	Flu	100
47605	25	Cancer	200
47602	23	Flu	200
47905	43	Cancer	100
47904	48	Flu	900
47906	47	Cancer	100
47907	41	Flu	900
47603	34	Cancer	100
47605	30	Flu	100
47604	36	Cancer	100
47607	32	Flu	100

Table 2.3: Patient table

Zipcode	Age	Disease	Count
476**	2*	Cancer	300
476**	2*	Flu	300
479**	4*	Cancer	200
479**	4*	Flu	1800
476**	3*	Cancer	200
476**	3*	Flu	200

Table 2.4: (1000,0.1)-closeness

2.2.2 (n,t)-closeness with k-anonymity

For this anonymization algorithm, Mondrian has been used as the base algorithm. The requirement for the cut step, explained above, has to respect both the k-anonymity and the (n,t)-closeness privacy requirements. This means that the EC have to be at least of size k and to within t distance of a natural superset of size at least n.

Chapter 3

Stages of the data publishing process

Searching for an official, government supported, publishing process which considers data sanitization did not yield any results. The only other data publishing process we have found, that has at its core data sanitization, is the process of Statistics Netherlands (CBS). This is a dutch institute responsible for reliable and consistent statistical information, that responds to society's demands in this respect. The data publishing process has been released in the book *Statistical Disclosure Control* [14, Ch. 3, p. 24-35]. We decided to use this as the foundation for our own Open Data publishing process because it is based on years of real-life experience of individual and legal entity microdata publishing. Microdata is data on the characteristics of units of a population, such as individuals, households, or establishments, collected by a census, survey, or experiment [6].

CBS makes data available under various formats B.3:

- Public Use Files - PUF. These are the data sets published online, freely accessible, constrained only by some type of General Public Licence (free, copyleft license for software and other kinds of works, intended to guarantee your freedom to share and change the concerned object).
- Microdata Under Contract - MUC. The user is constrained by contract to what he may do with the data. Various levels exist.
- Remote access - the user is allowed to query the data, but not allowed to see the data. A strict audit process is put in place.

In the case of Open Data we only have a more relaxed version of the PUF. The data is published online, is freely accessible, but there are no constrains on the data. The concepts presented in [14] are still viable and can be applied to Open Data. From a high level view, there are two parts to the process:

1. The user performs an in-depth manual or semi-automated data analysis to understand the characteristics of the data and the possible challenges.
2. The user select an appropriate method to sanitize the data and then publishes the anonymized data set.

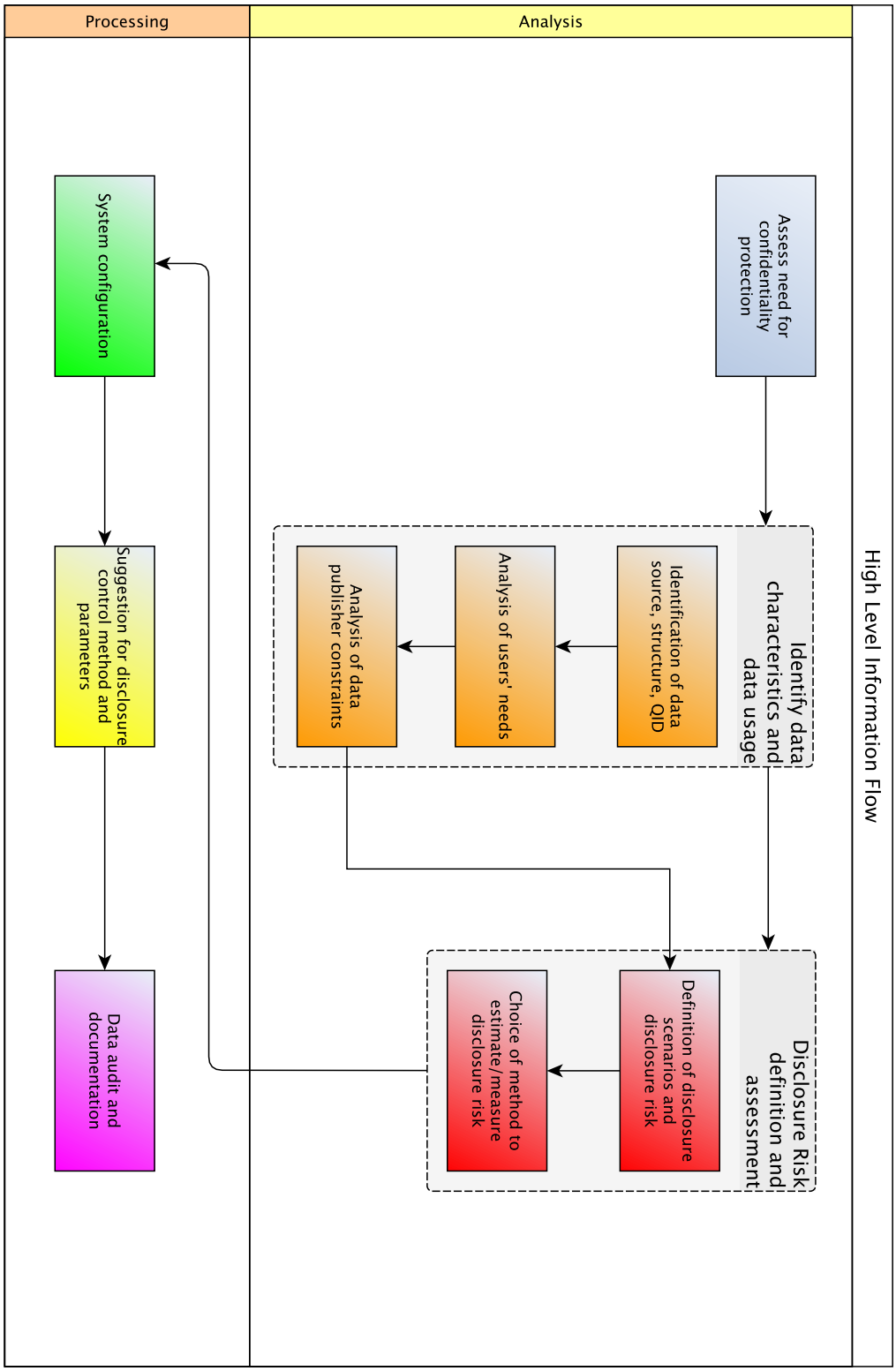


Figure 3.1: The 6 steps to anonymization

For the second part, CBS uses its own anonymization tool which only applies k-anonymity and suppression to sanitize the data. Unfortunately, as our experiments show, k-anonymity is not good enough to protect the data and applying suppression reduces utility by an unknown amount. Instead, other algorithms should also be tested to see if they can provide better privacy and utility guarantees. A data publisher should be able to choose between several such algorithms. This makes it possible to release better anonymizations, which can offer more utility with at least an equally good privacy level.

In the remainder of this chapter we will present a high level view of the data publishing process. Sections 3.1 to 3.3 and 3.6 present steps from the CBS process. It corresponds to point 1 mentioned above. Sections 3.4 and 3.5 represent our own replacement to the CBS software automated steps. They correspond to point 2 mentioned above.

3.1 Assess need for confidentiality protection

The first step involves analysing the data set to see whether statistical units or variables are present which require protection. The decision on what needs to be protected and what not is based on the law and on common sense and experience. The legislation can vary greatly per country, so a thorough analysis needs to be carried out by a legal department. In the context of Open Data, publication of the data needs to adhere to several regulations: rules of the data publisher's company or institution (if any), the Dutch Law and any applicable European Union law.

3.2 Identifying data characteristics and data usage

This step involves gaining a better understanding of the characteristics of the data characteristics and how this data can and may be used by different parties. In our context of Open Data, knowing exactly how the data will be used, is not clear. One can only estimate the possible usage, but based on the open data principle, we would expect users to bring new insights by combining the data sources in a novel way.

Control vs. No Control
At CBS, besides highly aggregated PUF files, data can also be anonymized towards certain usage, based on who asks for the data. Statistical programs can be executed on the original, but with high security and audit in place. With open data, publishing is mostly data driven - ensuring no sensitive information about entities or individuals is leaked and providing utility towards what the data publisher considers to be the most feasible usage. No control mechanisms exist.

3.2.1 Identification of data characteristics

The first step is to look at how the data has been gathered (e.g. a questionnaire) and what the data source is for the analyzed data set. The goal is to identify the type of the data (i.e. is it administrative data, census data, operations data etc.), to understand the current level

of public availability of the information (e.g. what already can be requested at a town hall) and to understand what can be made public. Releasing just a sample of the data set might be necessary due to one of the following reasons:

- the law might require a sampling process.
- one considers the disclosure risk too high for the full data set.

The next step is to look at the data itself.

1. determine if identifiers are present (e.g. passport number, name etc.). These need to be removed since they provide a one-to-one identification.
2. determine which attributes are the quasi-identifiers (QIDs). These are aggregated into coarser categories.
3. determine if any working variables (flags, checks, values derived from original attributes) are present. These must be removed since they leak information.

The last step is the post-anonymization sanity check. This consists of checking whether pre-anonymization attribute relationships are still preserved after the anonymization has been applied. This is necessary to keep the data consistent.

3.2.2 Analysis of users' need

It is not clear, beforehand, to know how the published data will be used, but it is expected to provide value by combining different data sources in a novel way. This means that the data publisher can only estimate how the data might be used. There are two reasons why this step is necessary. First, it can give insight into how the data might be misused, discussed further in Section 3.3. Second, based on the estimated usage, the importance of attributes can vary. For example, if age would be considered very important, then this would be anonymized less (finer aggregation groups, less perturbed values etc.). There are many sources where one could obtain information about attribute importance:

- user groups are created to share information about certain types of data
- electronic libraries may contain papers which deal with the type of analyzed data
- project descriptions might contain information on how they use the data
- talking directly with the experts (statisticians, data analysts etc.) can give insight on how they use the data and what level of detail is required.

3.2.3 Analysis of data publisher needs

If the data publisher works for a company or public institution, then he needs to adhere to the internal rules for data publishing. He might be obliged to respect certain internal policies, as identified in Section 3.1. Such rules may enforce a minimum level of anonymization for certain pieces of information or that certain attributes should be removed completely.

Another aspect the data publisher needs to consider is *consistency*. If previous versions of a data set (or part of) have already been published, the current data needs to be consistent

and account for such history. As a general rule of thumb, a data set should be published only once, since multiple versions increase the risk of information leakage (e.g. the data is shared between users and inconsistencies might leak information). Finding and obtaining previous publications can be a manually intensive process.

Determining which attributes are QIDs is the responsibility of the data publisher. In the literature [22] we have seen that the QID choice has an impact on the risk level and on the utility of the data. More attributes means more anonymized data and less utility. Less attributes means a higher risk that a re-identification might occur based on the not anonymized attributes.

3.3 Disclosure risk, definition and assessment

In this section we will be giving a brief overview of disclosure, disclosure risk, disclosure scenario and methods to estimate the disclosure risk. For detailed information, please refer to [14, Ch. 3].

3.3.1 Disclosure, disclosure scenario and disclosure risk

Disclosure

Disclosure relates to re-identifying an individual or a legal entity in the published data and gaining sensitive or confidential information about them. There are several types of disclosure: identity disclosure, attribute disclosure, inferential disclosure, table linkage. These are defined in Appendix A.

Disclosure scenario

All the pieces of information about the data set, gathered by the data publisher up to this point, can now be used to create disclosure scenarios. These are scenarios which demonstrate how the data can be misused and how disclosures can occur. A disclosure scenario includes reasoning about the quasi-identifiers and the ways in which these could be used. In literature, there are two systems which can help with creating disclosure scenarios. First, Elliot et al. [11] present a prototype tool called *Key Variable Mapping System*, which can be used to identify the QID attributes. Second, Elliot and Dale [10] present a system for analysing disclosure scenarios. For disclosure scenario examples, please refer to [14, Ch. 3]

Disclosure risk

The disclosure risk is a quantification of how rare a statistical unit is, in the data set but also in the whole population. In general, disclosure risk is a function of the values of the QIDs. Furthermore, we distinguish between two types of risk - individual and global. The former refers to the risk of each individual unit in the data set. The latter is usually determined based on the individual risk and represents the risk of the whole data set. Based on the type of QIDs, three cases are possible: the QIDs are categorical, the QIDs are continuous or a

combination of the previous two.

1. CATEGORICAL QID VALUES
In the case of categorical QIDs, risk is defined as the probability that a statistical unit can be correctly re-identified from the data at hand. Three variations are possible for the definition of risk:

- when considering microdata - if the entity is considered to be in the data set, if the QID combination of the unit is too rare in the data, then the entity is at risk.
- when considering population characteristics - we look at QID combinations for the statistical units of the data, which are rare in the whole population (e.g. the 18 year old widow).
- when considering real external files - implies extensive tests (searching many databases) to actually link the data set at hand with external sources.

2. CONTINUOUS QID VALUES
This case is more challenging than the previous because we are moving away from the rareness of combinations and towards the rareness in the neighbourhood of the record (QID values of an entity). Again, there are three variations.

- risk based on outlier detection strategies - intervals or thresholds are determined for the QID values and the units that fall outside the interval or past the threshold are considered at risk.
- risk based on clustering techniques - various clustering techniques are used to group the data; a boundary is set and any unit outside the boundary is considered to be at risk.
- risk based on record linkage - the original data is perturbed and then a linkage procedure is applied between the original data and the perturbed data. The risk is defined as the number of correct matches.

3. BOTH CATEGORICAL AND CONTINUOUS QID VALUES
This is the most challenging case of the three. One way to solve this problem is to group the data using the categories of the categorical QIDs and then apply the continuous QIDs solutions to each group (sub-population).

3.3.2 Choice of method to measure/estimate disclosure risk

An overview of methods to estimate disclosure risk is presented bellow.

Threshold rule

This method requires that keys (QID value combinations) appear in the data set at least a given amount of times. If the number of occurrences is below a given threshold τ, then the unit is considered to be at risk. Having at least $k > \tau$ occurrences of each key ensures at least a matching probability of at most $\frac{1}{k}$.

Sampling weights

The general risk is formulated as $\frac{1}{F_k}$, where F_k is the population frequency of a certain combination of QID values. Since F_k is unknown, it has to be estimated using the sampling design weights. $f_1 \dots f_K$ represent the data set counts. Using F_k and f_k the individual risk $\hat{r}_k$ can be estimated. For further mathematical details please refer to [14, p. 44].

Because the sampling design is an important component of this method, if this is not known for the data set, this estimation method should *not* be used.

Defining a global risk measure for the whole data set can be done in the following manner. If r_k is the probability of re-identification, then $\sum_k r_k f_k$ is the expected number of re-identifications. The re-identification rate can be used for a data size independent measure: $R = 1/n \sum_k r_k f_k$.

Using heuristics

Another way of estimating the disclosure risk is based on heuristics. One such heuristic is the DIS-SUDA [12] method. It uses *minimal sample uniques* or MSUs for its computations. An MSU is an unique variable set without any unique subsets. The SUDA [12] component assigns a per record matching probability based on the size and number of MSUs contained by the record. The DIS [9] component gives the conditional probability of a correct match given a unique match. The combination of DIS and SUDA describes the confidence of an intruder that a match is correct.

This method has been extensively tested [14]. It gives a good estimate of the disclosure risk and does not require the assumption of the existence of an underlying statistical model.

Using record linkage

Disclosure risk can also be estimated using record linkage. This translates directly to the number of records linked, given the number of total records. It is a computationally intensive task since it requires thorough testing with various data. We distinguish between distance based record linkage [28, 3] and probabilistic based record linkage [13, 15, 7, 25].

3.4 Configuration of the automated decision support tool

In this step, the user needs to configure the system before it can run its analysis on the data.

Input/Output

The user chooses where to read the data from and where to store the intermediate and final results. Input can be anything from a file to a database stored in the cloud.

Data pre-processing

This step involves modifying the data, making it suitable for processing and visualisation. Examples of modifications include labeling of columns, data transformation - useful when the part of the data exists in a format which is not suitable for the algorithms (split full name in two columns or put date in another format or change the format of an interval) and original data visualisation.

Choose the domain

Based on the previous analysis, it should be clear to the user what type of data he is dealing with. Many types of data exist including relational data, tabular data, transaction data, location data etc. The type of the data is important because it will later provide suggestions on the algorithms and metrics to use.

Choose candidate algorithms

Here the user can either select from a list of existing algorithms (suggestions should be made based on the type of data to be analysed) or he can select his own implementation.

Choose metrics

The user either picks implemented algorithms for privacy and utility or his own implementation.

Select QID & sensitive attributes

Selecting the QIDs and the sensitive attributes is important since most of the algorithms require these (at least with respect to relational data algorithms). The attributes should have already been identified is Section 3.2.

3.5 Selecting the algorithm to be used for publishing

In this step the user runs the automated software to analyze the data set. When complete, he should receive information on all algorithms and how they performed given their Risk Utility maps [8]. This is a plot which shows an algorithm's performance, measured by the privacy and utility metrics, when executed using different parameter values.

The data publisher selects which algorithm he considers best and that anonymization will be selected for publishing. If needed, the user can execute post-anonymization data processing. Examples include suppressing certain values or changing the format of the data(e.g. if the date should follow a specific standard) before it is published. Finally, the data is written out to the configured location.

3.6 Data audit and documentation

This last step is required in order to create valid expectations on behalf of the future data users. The data publisher should choose which pieces of information can be released to the public. Two important pieces of information are the results of the utility metrics and the methods used to protect the data.

The first gives insight on how usable the data set can be towards certain tasks, depending on the information provided by the used metrics.

The second should be made public for reasons of transparency. The data can be checked by an external party for compliance with the regulations. The documentation needs to explain the legal or administrative reasons behind the data anonymization process. Furthermore, information about the anonymization process can help users understand what has been changed (e.g. which variables and the process applied - suppression, perturbation, generalisation etc) and what the impact could be on their data usage. This is important because it is possible, for example, to calibrate data mining algorithms to account for modifications made by an algorithm such as k-anonymity.

Chapter 4

System Architecture, Design & Implementation

4.1 Overview

This chapter presents the architecture and technical details of our framework. The system presented here corresponds in our process to the steps described in Sections 3.4, 3.5 and 3.6.

The goal of the system is to act as a decision support tool and an anonymization tool for the data publisher. In the next sections we describe the high level logical architecture of the system (4.2) and high level behaviour of data and information within the system (4.3). Due to time constraints we designed the full system, but only implemented a part thereof. The differences for the logical architecture are explained in Sections 4.2.2 and 4.2.3, respectively. The differences for the behavioral architecture are given in Sections 4.3.1 and 4.3.2.

4.2 Logical architecture

4.2.1 New code vs. code re-usage

Before starting the design of the current system, we analyzed if reusing existing tools could speed-up the prototyping process. We had limited time to design and develop with respect to the size of such a system. We found an existing tool [23] that was written in Java and for which the source code was publicly available. The GPL allowed for free usage, it already had several key algorithms already implemented. These can be seen in Figure 4.1. This represents a high level description of how the **reused** system looks like. As seen in the figure, several key characteristics exist:

- database abstraction layer of reading/writing the data
- existing implementations of algorithms based on Incognito [17] and Mondrian [18]
- good data abstraction (i.e. Anonymizer class)

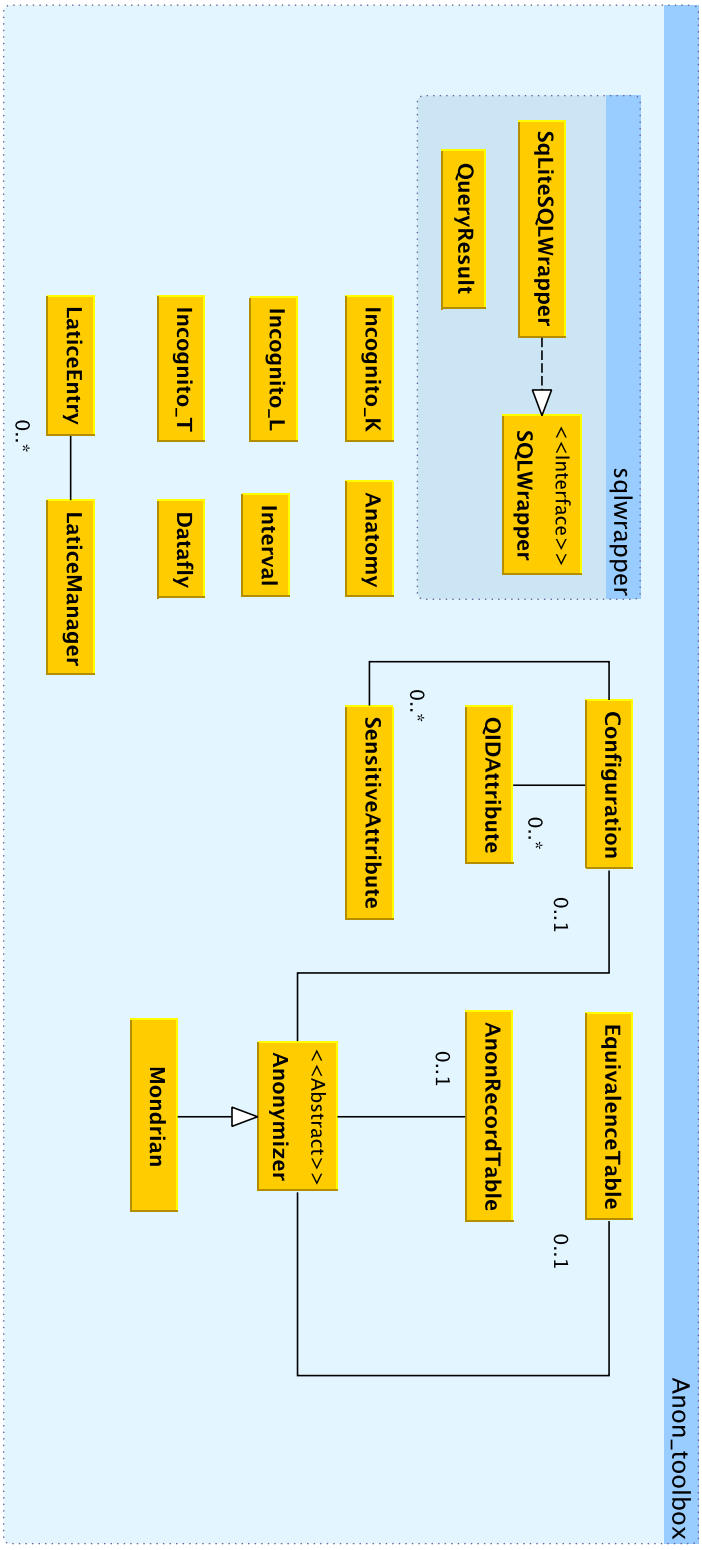


Figure 4.1: Original system components

- a configuration file
- QID and Sensitive Attribute representations

The existing algorithms are *Datafly*, *Anatomy*, *k-anonymity* implemented in both *Incognito* and *Mondrian*, ℓ*-diversity* and *t-closeness* implementations in *Incognito*. The *Lattice* and *LatticeManager* are necessary for *Incognito*.

EquivalenceTable and *AnonRecordTable* keep track of the created equivalence classes and of the QID/Senstive attribute generalisations, respectively. The *QIDAttribute* and *SensitiveAttribute* are representations of the previously mentioned attribute types.

Generally, there are two ways in which such a system works. It either treats everything as strings of characters or it treats everything as numbers. In our case, everything is seen as numbers or intervals. Categorical values are transformed to numbers by means specified in the configuration file.

The configuration file is essential. This is where everything receives its meaning. The following list describes the possibilities offered by the tool.

- input/output files - a database was not supported
- one algorithm to be executed on the data with required parameters
- categorical values translation to numbers (e.g. Male = 1, Female = 2)
- suppression values for different columns
- per column taxonomy tree
- definition of the QIDs
- definition of the sensitive attributes, though all existing algorithms only accept one such column

4.2.2 The full module design

Having the basics covered (e.g. reading,writing, data abstraction) helped in quickly developing a working system. The existing system required improvements in order to meet the requirements. The top priority requirements are the following:

- automated algorithm comparison and parameter variation
- can compare any number of algorithms
- can compute any number of utility and privacy metrics and display them
- modular
- easy to extend - new components, new algorithms, new metrics, new read/write locations
- GUI for the configuration

- GUI for data pre- and post-processing
- GUI for results(plot) visualisation
- GUI for data and anonymized data inspection

In Figure 4.2 we can see the full system design. Some components have been introduced only at concept level (the cloud-shaped components). These components require further analysis and research to determine the structure. For example, data pre-processing still needs to be researched to understand what users actually expect and need. Such research can start from investigating what other tools do wright. The *Anon_toolbox* component will mostly stay as is, since the elements are already implemented and working.

core

The *core* module has been extended with a *Runner*. This allows for execution of any number of algorithms and any number of parameter configurations. These configurations are parsed by the *Configuration* class. For each detected algorithm, a *RunConfig* is created. This stores the algorithm's full qualified name (i.e. including the package name) and the list of parameters (*RunConfigItem*) and how these should vary. It can also generate all possible parameter combinations for the algorithm.

mondrian

The *mondrian* module has been extended. First, in the original version, the *Mondrian* class was actually the full k-anonymity implementation using Mondrian. We have removed the specificity of k-anonymity from Mondrian into a separate class. In essence, a *Mondrian* based implementation needs to do three things: choosing a dimension on which to partition, choosing a value to split and checking if the split doesn't violate the privacy requirements. All these steps now can be modified by any subclass of *Mondrian*. Using these three entry points one can implement virtually any algorithm based on the original Mondrian.

Two contributions of this project to the algorithmic side are the implementations of (n,t)-closeness [20] (Mondrian_NT) and (n,t)-closeness combined with k-anonymity (Mondrian_KNT), both of which extend the base *Mondrian* class.

metrics

The *metrics* module has been introduced to allow the user to measure the privacy and utility levels of a given anonymization. There are two types of metrics represented by two classes: *PrivacyMetric* and *UtilityMetric*. They have the general methods *getName()* and *getPrivacyLevel()* or *getUtilityLevel()* respectively. The user can select from either existing metrics or implement his own. As long as the getter for the name and level is present, one can even execute system calls to let external tools compute the metrics.

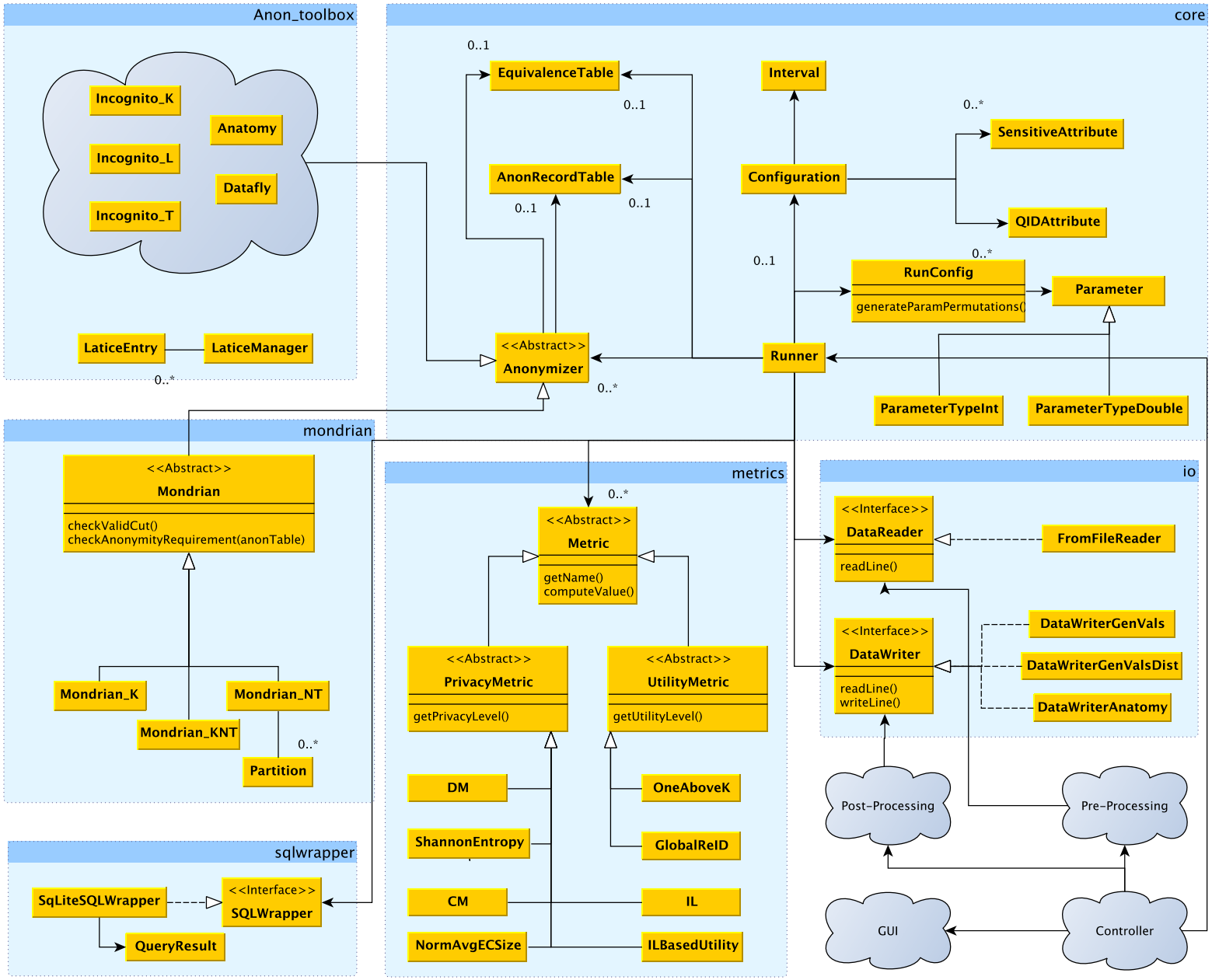


Figure 4.2: Full class diagram

io

The *io* component is used to read, write and, if necessary, transform data. The key elements are the interfaces *DataReader* and *DataWriter*. All they require is the ability to read in and write out line by line.

The *FromFileReader*, as the name suggests, takes the data from a file. This was also how the original tool read its data, but it was not easily extendable (e.g. replacing reading from file with reading from database).

DataWriteGenVals, *DataWriterGenValsDist* and *DataWriteAnatomy* are the representations of the data write possibilities of the original tool.

Pre-processing, Post-processing, GUI, Controller

As mentioned earlier, these components require further research to determine what the desired structure should be. This has not been done in the current project due to time constraints and due to the fact that they have no added value to the goal of this thesis.

4.2.3 The implemented design

In this section we will be going more into the details of what has been implemented and which improvements exist compared to the original tool. This can be seen in Figure 4.3 and regards mostly the *mondrian*, *metrics* and *core* components.

Anon_toolbox & sqlwrapper

This part of the system has remained mostly untouched. Improvements have been brought to all the implemented algorithms so that they could be executed many times. The most significant changes were made to how the data was read and stored, to make it compatible with the general approach which is currently implemented.

io

This part of the system has not been implemented. The data is still read from a file and written back to a file. The reading/writing responsibilities have been moved from the *Anonymizer* class (original tool) to the *Runner* class, which oversees all executions.

Internally, data changes are still done using a SQLite database. However, since some of the implemented metrics require external tools, the results of each anonymization is temporarily stored in a file accessible to the external tool.

mondrian

We chose to implement (n,t)-closeness (the *Mondrian_NT* class) based on several reasons. First, it is an algorithm that works for relational data. Second, there is no publicly available implementation of this algorithm. Third, literature criticizes t-closeness of significantly reducing the utility of a data set. Literature also states that (n,t)-closeness is an improved version of t-closeness w.r.t. utility. We decided to test this theory in our experiments.

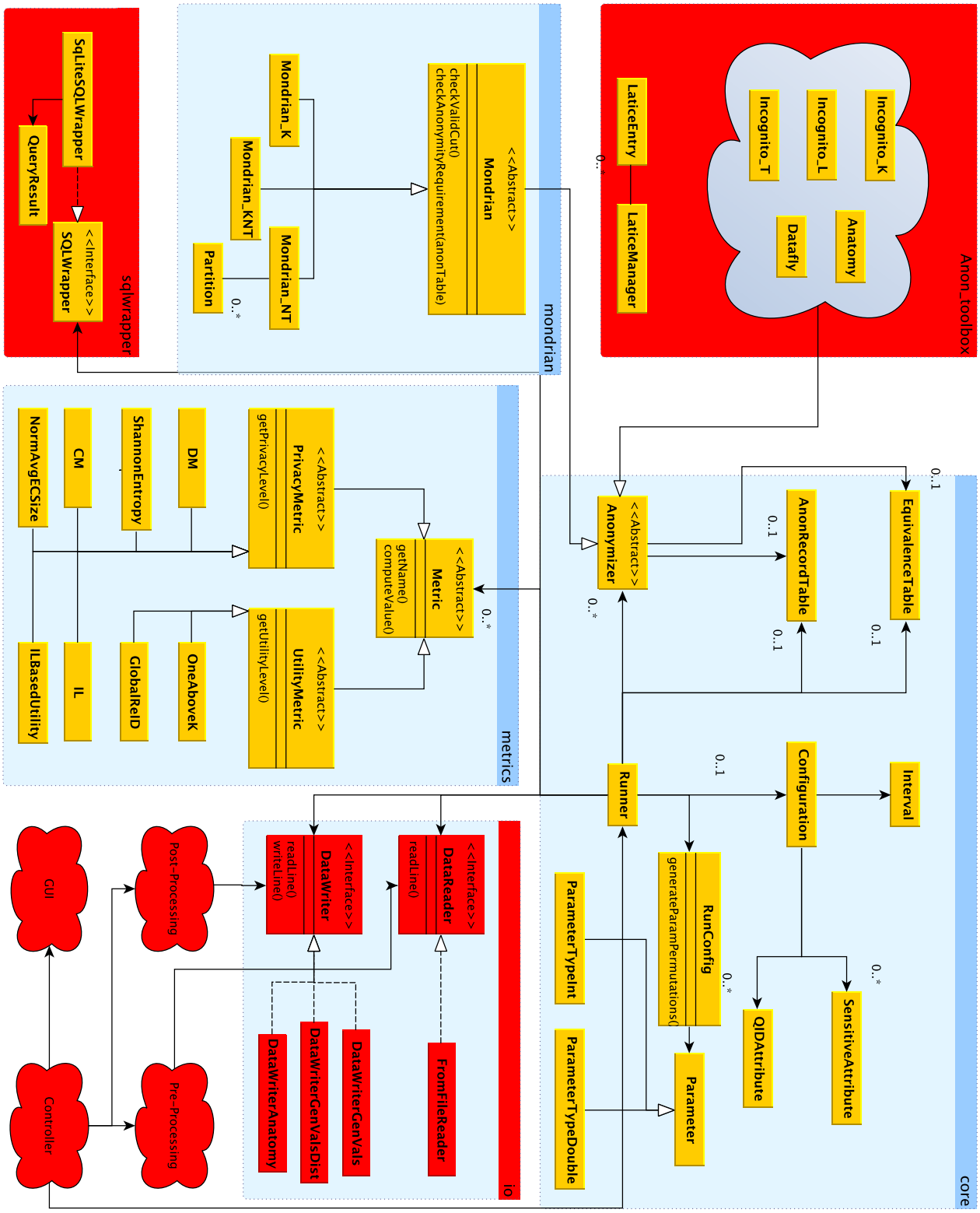


Figure 4.3: Class diagram of implemented features

Mondrian_KNT is our attempt to further improve the original (n,t)-closeness algorithm.

The implementations make full usage of the *Mondrian* abstraction, by modifying only how we choose the value to split on and how to check if the split satisfies the privacy requirements.

metrics

This module is a new piece of functionality added to the original tool. The metrics that have been implemented have been chosen based on the literature [22] - they are the most commonly used metrics to measure privacy and utility.

Utility

- CM - classification metric [4]
- DM - discernibility metric [4]
- ILBasedUtility - information loss metric [20]
- NormAvgECSize - normal average equivalence class size [21]
- ShannonEntropy - measure of uncertainty

Because CM and DM are dependant of the number of records in a data set, we decided to develop their normalized counterparts.

NORMCM
The classification metric adds a penalty to each row of 0, 1 or 2.

$$f(x) = \begin{cases} 1, & \text{if value is not majority in EC} \\ 1, & \text{if value is suppressed} \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

A penalty of 2 is given when in an EC class the majority of values are not suppressed. To normalize we thus consider only the rows which have received a penalty and divide the CM value by the number of rows with a penalty.

$$NormCM = \frac{CM}{\#\ rows\ with\ penalty} \tag{4.2}$$

NORMDM
In the case of the discernibility metric, the normalization is more straightforward. Every record in the dataset is penalized with a value equal to either the number of records in the equivalence class or the number of records in the whole data set.

$$NormDM = \frac{DM}{\#\ rows\ in\ data\ set} \tag{4.3}$$

Privacy

- OneAboveK - k is the size of the equivalence class. It finds $min(\frac{1}{k})$
- Global Re-identification risk (GlobalReID) - external call to R language script; based on the sdcMicro package implementation [12]

The flexibility of the system when it comes to metrics is more or less proved by the *Global Re-identification risk* metric. This has been implemented by placing an external system call to a script which takes the anonymized data file as input and returns the computed general privacy level.

core

Most of the new functionality lies in the *core* module. Initially, it was only possible to execute one algorithm with one set of parameters.

RUNABILITY

As previously mentioned, the *Runner* class allows for any number of algorithms to be executed with any number of parameter combinations (as long as memory allows it). The *Configuration* class has been modified to parse these changes by means of Java reflection. This makes adding a new algorithm easy. Simply implement and specify the full qualified name (package.className) in the *config.xml* (main configuration file). This improves on the previously hard-coded method.

When requesting a new algorithm, the user only needs to specify the algorithm name and between which values (and by how much) each parameter should be varied. The *Runner* then takes and generates a list of *RunConfig* with every possible combination for each specified algorithm.

DATA IO

Before, the data was read by the algorithm itself. Currently, the data is read by the *Runner* class and stored in the internal SQLite database. Then each algorithm simply receives a copy of the data. This method has been chosen to reduce from HDD reading time. We are aware that this might not be the best case when the data source is a database itself, but as previously mentioned, the data abstractization has not been implemented for this prototype.

PLOTTING

The *Runner* class is also responsible with calling the metrics and storing their results. Then it writes these values to several files. These files are then used to plot the Risk Utility Maps (RU Maps [8]). Each map contains all the algorithms and the results for one privacy and one utility metric for all parameter variations. Again, due to time constraints, we opted to use an external tool called *jgnuplot* in order to make plots in Java.

4.3 Dynamic behaviour of architecture

In this section we will explain the flow of data and information through the system, on a high level. As with the system design overview, we will first explain the idea for the full module concept and then go into the details of what was actually implemented. The flow described is related to the steps in sections 3.4 and 3.5.

4.3.1 The conceptual data flow

The high level data flow diagram can be seen in Figure 4.4. In essence, the system is quite simple. It consists of 6 elements:

- Configure - the system configuration that needs to be done before execution.
- Controller - the key element of the system: coordinates algorithms, measurements, plots, data I/O etc.
- Data - necessary for accessing writing and formatting of data.
- Anonymize - the sanitization process which cleans the data through anonymization.
- Measure - every anonymization needs to be measured in order to determine how the algorithm performed.
- Visualise - visualising the data, the configuration, the pre/post-processing, the measurements.

This general flow has been mapped to a more detailed flow, presented in Figure 4.5 and described below. The steps described contain between parentheses the name of the element that actually implements the component. An overview of this mapping is given in Figure 4.6.

Configure system. (GUI) First, the user needs to configure the system. This is done by means of a GUI to hide the complexity of the XML file from the user. This is the step where the data sources, data output, algorithms and parameters are set.

Parse configuration. (core.Configuration) Next, after the user starts the tool, the configuration will be parsed and all necessary internal values will be set (e.g. algorithms, QIDs, sensitive attributes etc).

Get input info. (core.Runner) The *Runner* class detects where the data should be read from.

Read data. (io.DataReader) The source location is passed onto the implementation of the data reader, which then retrieves and saves the data - depends on implementation. If necessary, data transformations are applied at this step.

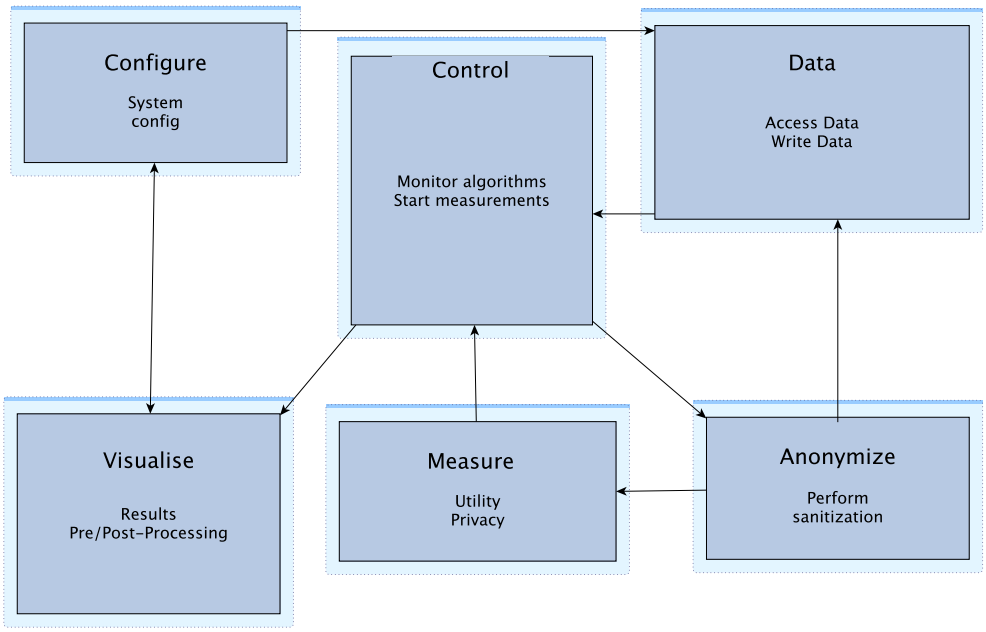


Figure 4.4: General information flow

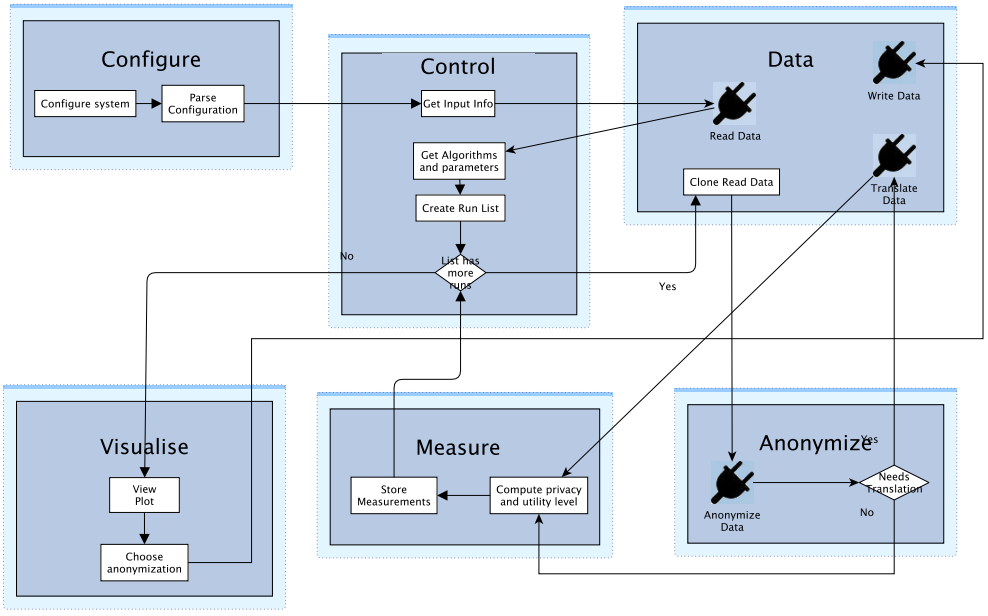


Figure 4.5: Functional data flow

Get information. (core.Runner) The *Runner* class retrieves from the configuration the information it needs to create the algorithm execution lists.

Create run list. (core.Runner) *Runner* adds for each algorithm and parameter combination one execution to the list.

Execute list (core.Runner) The *Runner* first clones the data and makes it available to the algorithm to be executed. The data is then anonymized. If the algorithm is executed externally, transformations might be necessary to format the data such that the system can read it.

Metrics. (metrics.*) After the anonymization is complete, the specified privacy and utility metrics are computed and stored.

Plot. (core.Runner, GUI) Once all the algorithms have been executed, the RU Maps are created and shown to the user.

Choose anonymization. (GUI) Based on the presented information, the user can choose the algorithm which he considers best for this data set. He might also require extra data post-processing (omitted due to reasons explained above).

Write Out. (io.DataWritter) Finally, the user selects for the anonymized data to be written to the specified destination.

4.3.2 The implemented data flow

The high-level data flow of the implemented system can be seen in Figure 4.7.

Data I/O. The first obvious difference lies in the fact that the *Runner* itself is doing the reading and writing of the anonymized data. There are no $DataReader$ and $DataWriter$ implementations.

Execute list. The run list behaves like two nested loops. The outer loop selects each algorithm in sequence. The inner loop first creates all parameter permutations for the current algorithm and then executes, in sequence, the algorithm with each parameter permutation. This behaviour can further be improved by executing each algorithm, and maybe each parameter permutation, in parallel.

Within each internal loop, the anonymized data set is first stored to a temporary location. This makes it possible to execute external metrics on the data. In our case the only external metric is GlobalReID.

Choose anonymization Currently, the system only allows the user to see the plot. The user cannot currently select an algorithm by clicking the plot (non-interactive) nor can he choose the final anonymization as this would require a GUI (omitted due to time constraints and

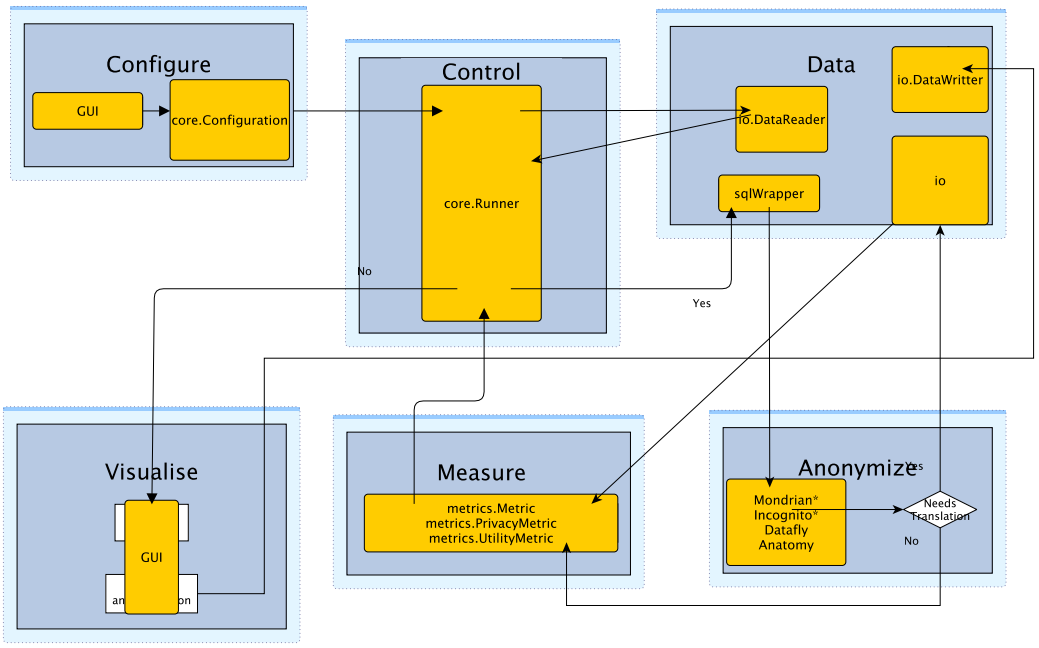


Figure 4.6: Functional to technical mapping

limited added value based on the goals of this thesis). The only way to do that, currently, is by restarting the program with one algorithm and one set of parameters.

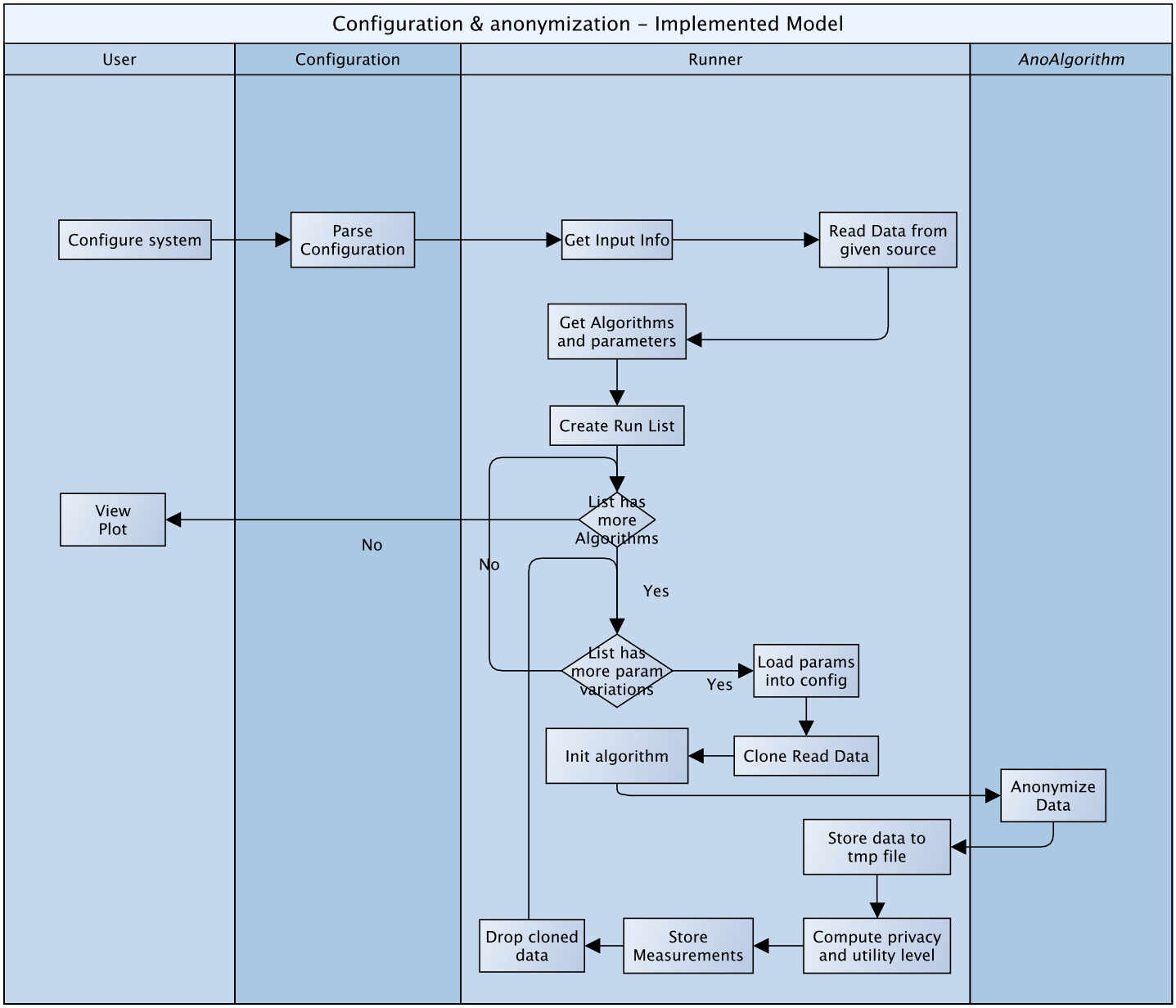


Figure 4.7: Information flow of implemented system

Chapter 5

Experiments & Results

In this chapter we present the results of applying the framework we have implemented, which is described in Section 4.2.3. Several algorithms have been selected for testing: *k*-anonymity [26], t-closeness [19], (n,t)-closeness [20] and (n,t)-closeness together with *k*-anonymity. These are all widely known algorithms for *relational data*. We also present the metrics used to evaluate the algorithms. This evaluation is motivated by the need to understand the meaning of the information presented to the data publisher and to understand how different algorithms behave in different scenarios.

We first look at each algorithm individually to understand how it behaves when the parameters and the quasi-identifiers change. In essence, we analyze the behaviour of the algorithm when more information becomes available. Afterwards, we compare the algorithms with each other, when more data, and hence more information, is added to the scenario.

In both cases we have used the following four algorithms to anonymize the data set.

1. (Incognito_K) An Incognito [17] based implementation of k-anonymity
2. (Incognito_T) An Incognito based implementation of t-closeness
3. (Mondrian_NT) A Mondrian [18] based implementation of (n,t-closeness)
4. (Mondrian_KNT) A Mondrian based implementation of (n,t-closeness) with the k-anonymity privacy constrain.

Of the four, we have implemented Mondrian_NT and Mondrian_KNT ourselves. The Incognito based algorithms have been reused *as is* from the existing toolbox of University of Texas at Dallas [23].

From the metrics mentioned in Section 4.2.3, we have measured utility using the normalized classification metric (NormCM), the normalized discernibility metric (NormDM) and the normalized average equivalence class size (NormAvgECSize). Privacy has been measured using only the global re-identification risk (Global ReID Risk).

We chose the utility metrics based on our experiments. The three utility metrics mentioned above were the only ones that gave interpretable results. For the other metrics it was hard to correlate the changes in the data to the metric values.

The privacy metric chosen for the experiments is also used in practice [14]. This motivated us to use it and to evaluate it.

5.1 Experimental Setup

5.1.1 The Data Set

For the experiments we have used the Adult data set[1]. The dataset is from the UC Irvine machine learning repository. It consists of data collected from the US census. Records with missing values have been removed, resulting in a data set consisting of 30162 records in total. We chose this data set because it is widely used in literature.

Research Question 3C is about understand the change of privacy and utility when data sets are combined. To simulate the effect of combining the data with external sources, we experimented with various QID sizes: 4, 7 and 13. The intuition is that having more attributes in the QID set increases the amount of available information. The same is valid when one combines the data set with an external source, by joining on some common attributes. The result is a bigger data set, with more columns and more information.

5.1.2 The Quasi-Identifiers

Attribute	Type	Nr Values	VGH height
Age	Numeric	74	4
Workclass	Categorical	8	3
fnlwgt	Numeric	20236	3
Education	Categorical	16	4
Marital status	Categorical	7	3
Occupation	Categorical	14	3
Relationship	Categorical	6	3
Race	Categorical	5	2
Gender	Categorical	2	2
Capital-gain	Numeric	118	3
Capital-loss	Numeric	90	4
Hours per week	Numeric	94	3
Native country	Categorical	41	5
Salary	Categorical	2	1

Table 5.1: Adult data set attribute characteristics.

The experiments have been concluded using three sets of QIDs of the following size: four, seven and thirteen. We shall represent these as QID_4, QID_7 and QID_{13} respectively. The attributes contained in the data set and their characteristics can be seen in Table 5.1. The three sets of QIDs are the following:

[1] http://archive.ics.uci.edu/ml/datasets/Adult

- {Age, Occupation, Race, Gender}
- {Age, Education, Marital status, Occupation, Race, Gender, Native country}
- All except salary

The original data set also contained an attribute called *Education-numeric*. This was however removed since it requires a post-anonymization consistency check with the attribute *Education*. It is the numerical representation of the education attribute.

The *Salary* attribute presented in Table 5.1 has been used as the *Sensitive Attribute*.

5.1.3 The Servers

Due to the long run times, the execution has been manually parallelized on several machines made available by the TU Delft. These can be seen in Table 5.2. The system is implemented in Java and makes use of an internal SQLite library for data storage and processing.

# servers	RAM(GB)	# procs	# cores	Proc name	Proc GHz
3	192	16	32	Intel(R) Xeon(R) CPU E5-2650	2.00
1	144	8	16	Intel(R) Xeon(R) CPU E5620	2.40
1	32	8	8	Intel(R) Xeon(R) CPU E5410	2.33

Table 5.2: Execution servers

Parallelization did improve the execution, time-wise, to a certain extent. However, it was not parallelization in the true sense, since each execution can currently only run on a single core. Some medium length executions took around 100 hours to complete. To overcome this we chose to do 100 random 1% samples of the original data set and use those values instead of single executions. This means that the runs were executed on 100 random samples of 301 records each.

5.1.4 Execution

By one execution we mean one run of the framework using one algorithm with one set of parameter values. Each type of algorithm has been executed using different parameter values and combinations. We briefly summarize these below:

Mondrian KNT n: 50 to 150 increments of 50; t: 0.1 to 0.3 increments of 0.1; k: 10

Mondrian NT n: 50 to 150 increments of 50; t: 0.1 to 0.3 increments of 0.1

Incognito K k: 5 to 50 increments of 10

Incognito T t: 0.05 to 0.3 increments of 0.05

Where more than one parameter was required, e.g. Mondrian KNT, we used all possible combinations of the listed parameters, e.g. {n=50,t=0.1}, {n=50,t=0.2}, {n=50,t=0.3}, {n=100,t=0.1}, {n=100,t=0.2}, ...

5.2 Discussion

5.2.1 Metric interpretation

Before we can look at the metric plots, we first need to understand what the metrics themselves measure. We will be explaining what NormCM, NormDM, NormAvgECSize and GlobalReIDRisk measure.

NORMCM
The classification metric is a measure of how well a classifier would operate on the data set. It gives a penalty to each row which can decrease the accuracy of the classifier. This happens when either a value in an equivalence class (or bin) is not part of the majority or when the value is suppressed.

NORMDM
The discernibility metric is a measure of how much the records in a data set are distinguishable one from another. The optimal result is achieved when no anonymization has taken place and no grouping exists. This implies that algorithms that aggregate records into bins decrease discernibility.

NORMAVGECSIZE
The normalized average equivalence class size is a measure of aggregation group size. It represents the value of the average bin size, normalized to a per record level. Normalization is useful when comparing different data sets or different data set samples.

NORMCM & NORMDM
Because *CM* and *DM* are values that depend on the number of records, we decided to use their normalized form in our experiments.

5.2.2 Baseline

Being able to interpret the results is very important. Giving meaning to the metric numbers was hard to achieve. The problem is that we did not have a fixed baseline to compare them against. To overcome this, we have defined a baseline for the metrics as the value of that metric when applied to the original data set (ODS). The baseline for the Adult data set and the used metrics can be seen in Table 5.3.

	QID-4	**QID-7**	**QID-13**
NormCM	1	1	1
NormDM	38.42	6.95	2.99
NormAvgECSize	9.4	2	1
GlobalReIDRisk	0.054	0.08	0.00003

Table 5.3: Baseline values per QID size

We observe that the original data set value for NormAvgECSize is 9.4 and 2 when the QID set sizes are four and seven, respectively. This happens due to duplicates based on only these four or seven attributes. On average, there are 9.4 identical record sets when the QID size is four and 2 when the QID size is seven. Using 13 attributes seems to be enough to create no duplicates.

We also see that the original data set has an initial re-identification risk of 5.4% and 8% for QID_4 and QID_7, respectively.

5.2.3 Individual algorithm analysis

Mondrian_KNT

NORMCM

In Figures C.1 to C.3 we can see that changing the size of the QID set or changing the parameters has a limited impact on the CM value. On the Y axis we have the normalized CM metric w.r.t. the ODS. We can see that the penalty of the anonymized data set can be reduced down to 80% of the penalty given to the ODS. This is somewhat expected since aggregation based anonymizations tend to remove any outliers or inconsistencies, improving, theoretically, the accuracy of a classifier.

NORMDM

In the case of the discernibility metric (Figures C.4 to C.6) increasing t does increase utility, since it relaxes the privacy requirements. When $n = 50$ we can see that values for $t \in [2,3]$ actually have a smaller penalty per record than the ODS. However, we notice that on this data set, increasing the QID size from 4 to 7 increases the DM penalty by a factor of 5-6, while increasing from 4 to 13 by a factor of 10-12. This is consistent with $n = 100$, $n = 150$, but also with the changes in utility seen in the NormAvgECSize metric. The relation between the increase factors should be quadratic, but since the NormAvgECSize for the baseline decreases when the QID set size increases, it becomes almost directly proportional.

NORMAVGECSIZE

We observe that for this metric (Figures C.4 to C.6), Mondrian_KNT never achieves a value less than 1. This means that the anonymized data set always has more record duplicates (w.r.t. the QID attributes) than the ODS. Increasing the value of n has a very big impact on the average EC size. Bigger values for n imply a stricter privacy requirement, which results in more records being grouped together. When n has a value equal the total number of records in the data set, Mondrian_KNT becomes Incognito_T.

PRIVACY

When looking at privacy, as expected, higher values for t mean that the risk of re-identification is also higher. But higher values for n decreases the risk, but also the scatter of the metric values. This can be explained by the fact that as the n value increases, the number of possible groups that have size greater or equal to n decreases, leading to less options when partitioning.

In (n,t)-closeness, the privacy guarantee states that for every EC g there exists a natural superset G, of size at least n, and that the distance between the distribution of g and that of G is at most t. The bigger G is, the more g has to anonymized in order to satisfy the distance requirement.

We observe that in all visualisations (except CM), if we choose a fix value for utility and then increase the QID size, that the privacy levels drop significantly. Take for example Figure C.6. We fix the DM penalty at 5 times that of the original. From a) we can see that this is possible for a re-identification rate of about 1%. From b) the risk increases to 3% and in c) to 8%.

More QIDs in the set imply that the grouping of values will be courser and will have a greater impact on utility. This is also known as the "Curse of dimensionality" [2]. So in order to preserve utility, one has to sacrifice privacy.

Montdrian_NT

NORMCM
Analyzing the normalized CM plots w.r.t ODS, Figures C.10 to C.12, we notice, again, that the value of n has limited impact on the classification metric. As in the case of Mondrian_KNT, a QID of size 7 and 13 makes it possible to anonymize the data such that the CM penalty is only 80% of the ODS penalty. However, for QID_4, the value for CM can go down to 20% of that of the ODS. This can be explained by looking at how Mondrian works. It looks on all possible dimensions to perform a slice of the range and selects the best slice. Having less dimensions to slice on implies that the focus is greater on the same dimensions. This results in a finer grained partitioning that is of higher quality, when the goal is classifier training.

NORMDM
Here we notice that higher QID group size does not necessarily improve privacy, it only decreases it's range of variation. As explained above, more options when choosing the dimension to slice on results in coarser bins. This means that privacy goes up and utility decreases.

As noted before by Mondrian_KNT, almost doubling the QID size results in doubling the DM penalty.

NORMAVGECSIZE
Because (n,t)-closeness does not have an inferior limit such that of (n,t)-closeness with k-anonymity, it can slice the data set into very small bins. This is especially true when n is small. A small n means that the privacy requirement only needs to hold w.r.t a small neighbourhood of records. This explains how it is possible to obtain a normalized average EC size of up to two times that of the ODS. In the QID_{13} scenario, this translates to an average EC size of two.

PRIVACY
On this data set, (n,t)-closeness can give good privacy guarantees, but at high utility costs.

If one would like to use the data set to train a classifier, very good privacy and utility can be achieved. If that is not the case, except for some rare situations where the distribution of the sample allows for both good privacy and good utility (e.g. Figure C.13-c), it would be very hard to achieve a privacy guarantee of less than 1% for small values for n, without the highest utility costs.

Incognito_T

Through our experiments for t-closeness we achieved three things. First, we have confirmed the findings in our literature review [22] about t-closeness. It has indeed good privacy for very bad utility. Second, we show how much the utility is degraded when compared to the original data set. Third, we observed a limit on QID size for which the algorithm can produce an anonymized data set without suppressing all the values.

As one can see in Figures C.19 to C.21, there are no plots for QID_{13}. This means that in our experiments, no reasonable value of t (less than 0.5) could find an anonymization for QID_{13}. The scenario involves a sample of 301 records and 13 attributes in the QID. We also tried to increase the random sample to 10% (3016 records) and still the algorithm did not manage to find a suitable anonymization. This implies that the QID space was too sparse for the strict privacy requirements of t-closeness.

NORMCM
The CM penalty is from 75% to 100% of that of the ODS. We notice that for QID_7, when the privacy requirement t is stricter (less than 0.15), that we obtain a CM penalty of 20% to 30% of ODS with a good privacy guarantee (less than 0.5% re-identification risk). This is caused by a more efficient grouping of the attributes when the goal is classifier training.

NORMDM
Where CM shows that the anonymized data is better than the ODS from a classification point of view, DM shows the opposite when it comes to a more general utility definition. In literature, a value of $t < 0.15$ is deemed acceptable [22]. Yet, with this data set, we notice that higher values for t may achieve the same or slightly worse privacy guarantees, but with much more utility.

NORMAVGECSIZE
Small QID set sizes produce acceptable ECs. Achieving a privacy guarantee of less than 1% for QID_4 implies an average bin size of 27 (ODS size of 9 times the factor 3 - Figure C.21). For QID_7 we have a size of about 120 for the same privacy guarantee. For QID_{13} we could extrapolate to a size of about 500 for the same privacy guarantee. This means that a random 1% sample of size 301 is too small to produce one bin. A 10% sample would only have room for about six bins. This, combined with the sparsity of the QID domain makes it impossible to find an anonymization.

PRIVACY
For small values of t (less than 0.2), high privacy guarantees can be achieved of 0.5% to 1%.

This comes, however, at a high utility cost when the data is not used for classifier training purposes.

Incognito_K anonymity

The k-anonymity algorithm had the same problem as t-closeness. It could not find a suitable anonymization for QID_{13} (Figures C.22 to C.24). This relates back to the sparsity of the QID space. From literature we know that increasing the QID size also increases the sparsity of the QID space. This is called the curse of dimensionality [2].

NORMCM
For a small QID size (four) we obtain big improvements on the classification penalty. These vary between 20% and 50% of the ODS penalty. This shows that good aggregations are created when few attributes are analyzed. When we increase the number of attributes to 7, the same pattern of 80% to 100% of ODS emerges, as seen for the previous algorithms.

NORMDM
Again, we see that a QID of size four can actually obtain an on par utility with that of the ODS, iff one considers a re-identification risk of 4% acceptable.

NORMAVGECSIZE
Opposed to the other algorithms, the values for the normalized average equivalence class size are quite stable. The normalization occurs by dividing the average bin size by the value of k. For example, in Figure C.24-a we see that $k = 50$ gives a factor of about 5.3 for utility w.r.t ODS. Since 9.4 is the average ODS bin size, this means that the average set size for $k = 50$ is approximately 50. Because of how k-anonymity works, we know that the average bin size will always be at least k.

PRIVACY
From the interviews B.5 we have learned that in practice, in the medical domain, a value of $k = 5$ is considered to be a good choice for anonymizing a data set with k-anonymity. From Figure C.23-a we understand why this is a desired value. The utility is in about 50% of the cases on par with that of the ODS and only 2 to 2.5 times worse in the rest of the cases. But looking at the privacy guarantee, we see that the chance for re-identification varies between 2% and 4%. It comes down to the data publisher to decide what is a good threshold for the risk. Given that there are 30162 records, a 2% to 4% re-identification rate translates to one re-identification for every 25 to 50 records. This enforces the need to actually suppress values, degrading the utility by an unknown factor. An acceptable identification rate, depending on the dataset, could be between 1 in 500 to 1 in 1000 records (a 0.2% to 0.1% global re-identification risk, respectively).

From our point of view, $k = 5$ is not the best choice for anonymization. k should either be higher or a different algorithm should be used.

The spread of the $k = 5$ points also confirms the fact that it is really necessary to analyze the data, since the same algorithm with the same parameter can perform significantly

different for different data samples.

5.2.4 Comparison of Algorithms

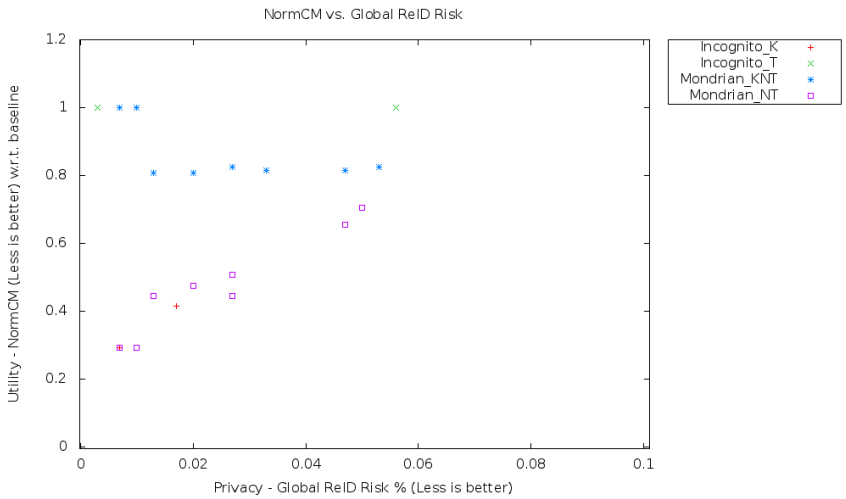


Figure 5.1: Example Comparison

The same data that was used for the plots discussed in the previous section has been merged in order to be able to compare the algorithms head to head. This means that 100 1% samples have been used, which gives a good idea of how the algorithm behaves. Normally, the data publisher would see significantly less points, as the execution of one single pass for all the algorithms with their parameter variations over the data set. To show an example we have plotted one run on a random 1% sample for all the algorithms. This can be seen in Figure 5.1. As previously mentioned, from the three QID set sizes, the Incognito implementations could only find a valid anonymization for sizes four and seven.

NORMCM
In Figure C.25 we observe that for QID_4, Incognito_K and Mondrian_NT offer the best utility and second best privacy. Incognito_T, as expected, offers the best privacy but at a high utility cost. In this scenario, the worst case value for utility is on par with that of the ODS.

For a QID set of size seven it becomes clear that Incognito_T is the best choice. Because Mondrian tries to slice the QID space as uniformly as possible, it does not provide optimal aggregation of values and incurs a higher classification penalty for it's two implementations. k-anonymity is limited by the value of k to the minimum bin size. This makes it possible for mixed values to be grouped together and incur a higher penalty. Incognito_T manages to achieve a grouping of values into smaller bins and yet preserve privacy.

In the third scenario (Figure C.25-c) we can see that (n,t)-closeness without k-anonymity provides, in the majority of configurations, a better anonymization.

NORMDM

In both cases where the QID set size was four and seven, all algorithm anonymizations have, for a given parameter value, a global re-identification rate of 0.7% for approximately the same utility value. We can only distinguish them when the QID set size is 13. First, the Incognito implementations could not find any anonymization. Second, the Mondrian based algorithms managed to find anonymizations with a better utility level than when the QID set size was equal to four. KNT managed a factor of 5 w.r.t to the ODS while NT a factor of 1.3 to 2. The reason why NT outperforms KNT is that the former is not limited by a minimum EC size of k.

NORMAVGECSIZE

We observe in Figure C.27 that Incognito_T offers the best anonymization possible for QID_4. In QID_7, Incognito_T is on par with Mondrian_KNT. Incognito_T requires a taxonomy tree for every attribute in order to work. Having a better result than Mondrian_KNT means that the user defined taxonomy tree for QID_4 is better than the Mondrian self generated partitioning. In QID_7 we see that Mondrian_KNT is able to find a similar partitioning to that of Incognito_T. In the last case, that of QID_{13}, Mondrian_NT outperforms Mondrian_KNT again. The reason is the same as for the *DM* metric: there is no lower bound on the EC size for Mondrian_NT.

PRIVACY

If one would need to choose, then Incognito_T would be the best choice for QID_4 and equally good as the Mondrian implementations for QID_7. It seems that in this case, the strategy for the balanced taxonomy trees for the Incognito algorithm performs better than the median partitioning strategy for the Mondrian algorithms. In the case of QID_{13} however, the only distinction that can be made between Mondrian_NT and Mondrian_KNT will be based on the utility offered, since both algorithms achieve a global re-identification rate of less than 0.1%.

GENERAL REMARKS

We also noted, thought not specifically tested for, that Mondrian_NT and Mondrian_KNT have a run-time that is 6 to 10 times faster than that of Incognito_K and Incognito_T, with Mondrian_KNT being the fastest. We believe that Mondrian is generally faster than Incognito implementations since it stops when no more cuts can be performed along any dimension. On the other hand, Incognito could, in the worst case scenario, continue until it has exhausted all the lattice search tree nodes. Mondrian_KNT is also faster than Mondrian_NT since the k-anonymity requirement forces the algorithm to stop earlier with the partitioning.

Chapter 6

Discussion and Future Work

In this chapter we discuss the answers to our research questions, significant findings and the directions for future work.

6.1 Conclusions

We now present how our investigations and experiments give an answer to the proposed research questions mentioned below.

RQ 1 *Why is privacy preserving data publishing necessary when dealing with Open Data?*

From our literature survey [22] we have identified two types of data encountered when publishing: *sensitive* and *non-sensitive* data. Sensitive data is information that might result in loss of an advantage or level of security if disclosed to others. It may affect the privacy or welfare of an individual, trade secrets of a business or even the security of a nation. It is necessary for such data to be protected when a data set is released.

To the end of protecting sensitive data when publishing, guidelines[1] and laws[2] have been created. These are, however, not enough. From interviews, presented in Appendix B, we have learned that these rules only set boundaries meant to cover all possible scenarios. The regulations lack the ability of precisely defining what should and what should not be published.

This means that a special process in required when dealing with sensitive data, a process which requires expert level knowledge of the possible problems that can arise.

RQ 2 *How are decisions taken when publishing sensitive data as Open Data?*

In The Netherlands, publishing data is done using the "open tenzij" [24](tr. open unless) rule. If the data set contains sensitive data, as defined by law or the company regulations, then this data is not published. We identify two advantages to this approach:

[1] Guidelines for The Netherlands: `https://data.overheid.nl/handreiking`

[2] In The Netherlands: Wet Openbaarheid van Bestuur

- unpublished data presents no risks.
- not many experts are required - it is also the fact that there is a lack of experts who can actually clean and anonymize the data before publishing.

This guarantees that the sensitive data is safe through secrecy. The biggest disadvantage of this method is that unpublished data offers no value to anyone but the institution who owns the data.

The alternative requires the data to be anonymized before release. This publishing process is based on a combination of rules, experience and intuition [22]. The challenge can be summarized as lack of knowledge. There are not enough specialists who can sanitize the data. Most institutions do not have such specialists. The people who have the expertise are already busy with such tasks. Forwarding the sanitization process to them would only create a bottleneck.

RQ 3 *How to anonymize the data?*

A Which algorithms should be considered as candidates for anonymization for which type of data, with respect to applicability in practice?

B How to interpret the measured values for privacy and utility and what guarantees do these values provide?

C How does privacy / utility change when the data set is combined with external sources?

RQ 3A

To answer the first sub-question, *Which algorithms should be considered as candidates for anonymization for which type of data, with respect to applicability in practice?*, we first have a look at our literature survey [22]. There we have learned that the best candidates for our data category type, namely *relational data*, are the following five algorithms: k-anonymity, ℓ-diversity, t-closeness, (n,t)-closeness and (n,t)-closeness with k-anonymity.

From our experiments we have concluded that there is no best algorithm. As expected, it depends on the data set what the best anonymization is.

From our observations for the given data set, we conclude that Mondrian_NT and Incognito_K have extremely good results for the CM metric in the case of QID_4, while Incognito_T has extremely good results for the QID_7 scenario.

RQ 3B

The second research sub-question, *How to interpret the measured values for privacy and utility and what guarantees do these values provide?*, is concerned with the metrics. We first tried to give the values of the metrics - classification metric (CM), discernibility metric (DM) and the normalized average equivalence class size (NormAvgECSize) - a tangible meaning. This was only possible for NormAvgECSize since it measures the average bin size. The other required an extra step.

In our second attempt we normalized the CM and DM metrics, NormCM and NormDM, respectively. This gave data set size independent values. Though an abstraction to the initial concept, it was still not possible to correlate it to the changes in the data.

In order to make the values meaningful we used the original data set (ODS) as a baseline for the metrics. The plots report the value of the utility metric applied to the anonymized data set, divided by the value of the same utility metric applied to the ODS. This factor can have any positive value. A value less than 1 implies an improvement over the ODS, while a value greater than 1 a decrease in utility. For example, in some cases we have a NormCM value for some anonymization that is 0.2 times that of the value for the ODS. This means that a classifier would have an increased accuracy using the anonymized data set, than when it would be using the ODS.

NormDM translates to how different the records are after the anonymization than before. A factor of one to two w.r.t. ODS would mean that the anonymization preserves, more or less, the record diversity and hence their utility.

The used privacy metric represents the global risk that a record in the data set might be re-identified. In our case, a re-identification rate of 1% translates to 301 records that could be at risk, given that the data set has 30162 records.

Instead of trying to find meaning in the utility metric alone, we have also expressed the meaning of the metric in terms of what happens when the anonymized data set is used instead of the original one.

RQ 3C

We observed a rather surprising result for our third sub-question, *How does privacy / utility change when the data set is combined with external sources?*. We expected that one can find a balance between sacrificing utility and sacrificing privacy, without actually achieving good values for both at the same time. From the experiments, however, it seems this is possible, at least to the extent this can be expressed by the metrics.

The metrics used follow a *less is better* value ordering. This means that the best anonymizations have points plotted close to the origin point (0,0). As one can see in the plots in Appendix C, there are many points which have this property.

6.1.1 Other experimental findings

From the interviews, Appendix B.5, we have learned that in practice a value for $k = 5$ for Incognito_K is considered good enough. They further eliminate any remaining risks by means of suppression. In our case, from Figure C.23 we can see that utility varies, but always has a bad privacy guarantee. We consider 5-anonymity as a risky choice and would recommend either a much higher value for k (e.g. > 10 for QID_4 or > 50 for QID_7), but which would decrease utility significantly, or use a different algorithm.

What is a good value for the privacy metric? From [14] we conclude that it is data set specific. It is up to the data publisher to decide what is an acceptable risk. This can vary, for example, from 1 re-identification every 100 records to 1 re-identification every 1000 records. This is equivalent to a risk between 1% and 0.1%, respectively.

6.2 Future Work

6.2.1 Extending the framework

There are many directions for future work. The first step should be extending the current proposed framework. In our case, we have developed a solution for dealing with relation data. But many other data types exist including, and not limited to the following: transactional, location, social, graph.

The current framework is only a prototype. It lacks many modules including: a graphical user interface (GUI), a data pre- and post-anonymization processing capabilities, more modules for data I/O (reading and writing to and from a database or to and from other sources). It also requires new algorithm implementations for other data types (transactional, location etc), but also for relational data. In the future, we would like to add ℓ-diversity to the framework and compare it to the existing algorithms.

6.2.2 Correlation between CM and classifier accuracy

We are also interested in finding whether a correlation exists or can be automatically determined (in case it varies on a per data set basis) between classifier accuracy and the value given by NormCM plot w.r.t. to ODS. We are interested, for example, to understand how a 0.8 value given by NormCM w.r.t ODS translates to classifier accuracy percentage.

6.2.3 Further automate the process

There are still many manual steps in the process presented in Chapter 3. We would like to further investigate other tools which can be used to further automate this process. As a starting point we would consider the following systems. Elliot et al. [11] present the Key Variable Mapping System which can be used to identify the QID attributes in a data set. Furthermore, Elliot and Dale [10] present a system which can be used for analysing disclosure scenarios - the intruder's perspective.

6.2.4 More on Sensitive Data

We consider that sensitive data is not researched thorough enough. We would like to investigate this direction and identify ways in which sensitive data can be identified, how sensitive data can be quantified and how sensitive data can be ranked - why is this data more sensitive than the other. Another interesting question would be to investigate which *combinations* of individually harmless data (i.e. neighbourhood welfare, problems on the street that need repair, placement of trash bins etc.) can lead to the creation of sensitive data. See "Makkie Klauwe" example in Section 1.1.

Acronyms

EC Equivalence Class

EMD Earth Mover Distance

ODS Original Data Set

PPDP Privacy Preserving Data Publishing

QID Quasi-IDentifier

RQ Research Question

Bibliography

[1] The dutch open government draft action plan. http://www.opengovpartnership.org/sites/www.opengovpartnership.org/files/country_action_plans/Draft%20Action%20Plan%20The%20Netherlands_0.pdf. Online; visited May 2013.

[2] *On k-Anonymity and the Curse of Dimensionality*, 2005.

[3] Johann Bacher, Ruth Brand, and Stefan Bender. Re-identifying register data by survey data using cluster analysis: An empirical study. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):589–608, 2002.

[4] Roberto J Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 217–228. IEEE, 2005.

[5] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, 2009.

[6] United Nations. Statistical Commission, United Nations. Economic Commission for Europe, and Conference of European Statisticians. *Terminology on statistical metadata*. Number 53. United Nations Statistical Commission and Economic Commission for Europe, 2000.

[7] Josep Domingo-Ferrer and Vicen Torra. Disclosure risk assessment in statistical microdata protection via advanced record linkage. *Statistics and Computing*, 13(4):343–354, 2003.

[8] George T. Duncan, Sallie A. Keller-mcnulty, and S. Lynne Stokes. Disclosure risk vs. data utility: The r-u confidentiality map. Technical report, Chance, 2001.

[9] Mark Elliot. Dis: A new approach to the measurement of statistical disclosure risk. *Risk Management*, pages 39–48, 2000.

[10] Mark Elliot and Angela Dale. Scenarios of attack: the data intruders perspective on statistical disclosure risk. *Netherlands Official Statistics*, 14(Spring):6–10, 1999.

[11] Mark Elliot, Susan Lomax, Elaine Mackey, and Kingsley Purdam. Data environment analysis and the key variable mapping system. In Josep Domingo-Ferrer and Emmanouil Magkos, editors, *Privacy in Statistical Databases*, volume 6344 of *Lecture Notes in Computer Science*, pages 138–147. Springer, 2010.

[12] Mark J Elliot, Anna Manning, Ken Mayes, John Gurd, and Michael Bane. Suda: A program for detecting special uniques. *Proceedings of the Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality*, pages 353–362, 2005.

[13] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64:1183–1210, 1969.

[14] Anco Hundepool, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric Schulte Nordholt, Keith Spicer, and Peter-Paul de Wolf. *Statistical Disclosure Control*. Wiley, 2012.

[15] M. A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. Journal of the American Statistical Association 84:414–420, 1989.

[16] Nellie Kroes. The big data revolution. `http://europa.eu/rapid/press-release_SPEECH-13-261_en.htm#PR_metaPressRelease_bottom`, March 2013. Online; visited May 2013; EIT Foundation Annual Innovation Forum /Brussels.

[17] 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. ACM, 2005.

[18] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In Ling Liu, Andreas Reuter, Kyu-Young Whang, and Jianjun Zhang, editors, *International Conference on Data Engineering*, page 25. IEEE Computer Society, 2006.

[19] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In Rada Chirkova, Asuman Dogac, M. Tamer zsu, and Timos K. Sellis, editors, *International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[20] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. Closeness: A new privacy measure for data publishing. *IEEE Trans. Knowl. Data Eng.*, 22(7):943–956, 2010.

[21] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):146, 2007.

[22] Andrei Manta. Literature survey on privacy preserving mechanisms for data publishing, May 2013. pages 1-54.

[23] University of Texas at Dallas. Anonymization toolbox. http://cs.utdallas.edu/dspl/cgi-bin/toolbox/index.php, October 2013.

[24] Rijksoverheid. Innovatie-estafette 2011: baaierd aan innovatieve initiatieven. http://www.rijksoverheid.nl/ministeries/ienm/nieuws/2011/10/04/innovatie-estafette-2011-baaierd-aan-innovatieve-initiatieven.html, October 2011. Retrieved May 2013.

[25] Chris Skinner. Assessing disclosure risk for record linkage. In Josep Domingo-Ferrer and Ycel Saygin, editors, *Privacy in Statistical Databases*, volume 5262 of *Lecture Notes in Computer Science*, pages 166–176. Springer, 2008.

[26] L. Sweeney. k-anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):557–570, 2002.

[27] Thijs van den Broek, Noor Huijboom, Arjanna van der Plas, Bas Kotterink, and Wout Hofman. Open overheid. http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/rapporten/2011/01/14/open-overheid-internationale-beleidsanalyse-en-aanbevelingen-voor-nederlands-beleid/tno-rapport-open-overheid-creative-commons.pdf, January 2011. Retrieved May 2013.

[28] William E. Winkler. Re-identification methods for masked microdata. In Josep Domingo-Ferrer and Vicen Torra, editors, *Privacy in Statistical Databases*, volume 3050 of *Lecture Notes in Computer Science*, pages 216–230. Springer, 2004.

Appendix A

Terminology and Definitions

In this section we define terminology necessary to understanding the contents of this thesis. We start by defining what privacy protection is.

Privacy protection Access to the published data should not enable the attacker to learn anything extra about any target victim compared to no access to the database, given that the attacker has only a *limited* amount of background knowledge.

If privacy protection fails, then a disclosure takes place. There are several types of disclosure possible.

Disclosure types Disclosure as a consequence of a *linkage attack* and as a consequence of a *probabilistic attack*.

Linkage attack types: **Identity disclosure** or **re-identification** occurs when an individual's or entity's record is identified based on matching attributes. *Attribute disclosure* occurs when only the sensitive attributes belonging to an individual or entity are re-identified. For example, if everyone in a group has the same sensitive attribute. *Table linkage* occurs when an individual or entity is re-identified as being part of or missing from the data set.

Probabilistic attack types: also known as **inferential disclosure**, this implies that after the data set has been published, an intruder can now infer some sensitive value of an individual or entity with a higher probability than otherwise possible.

Below we present the type of possible attributes in a data set. These are *identifiers*, *quasi-identifiers*, *sensitive attributes* and *non-sensitive attributes*.

Identifiers These are attributes, or a set there-of, that fully and non-ambiguously identify a person (also referred to as "victim") to some pieces of sensitive information in a data set. Examples include SSN, passport number and name.

Quasi-identifiers(QID) Represent a set of attributes used for linking with external information in order to uniquely identify individuals in a given anonymized table. These include attributes which at first glance may seem harmless - postcode, gender, age.

Sensitive attributes (S) These attributes contain values that are considered to be sensitive to the victim. Examples of such attributes are salary and disease.

Non-sensitive attributes (NS) These attributes are composed of column in the table that do not fall under any of the previously mentioned categories.

The order is which the attributes are determined to belong to a certain category is the same as above: first identifiers, then QIDs, then sensitive attributes. The rest are then considered non-sensitive attributes. One might ask that, since these attributes are sensitive, why publish them at all. The problem lies in the fact that these values are most of the time the reason why such a data set is published. Thus, the solution must rely on hindering an attacker's ability to link an individual to sensitive information.

All the anonymization algorithms used in this thesis rely on data generalisation to perform the anonymization. The attributes that are generalized are the QIDs. Through generalization, groups of records are created that have the same QID values. Such a group is referred to as an **equivalence class (EC)**.

Throughout this thesis we will be referring to the utility and privacy of data. We define these as follows.

Data utility refers to the level of actual or perceived usefulness of a data set, from the point of view of the user of that data.

Data privacy refers to how much guarantee can a data set offer w.r.t. not leaking any private or sensitive information about an individual or entity. In other words, what is the offered level of privacy protection.

Appendix B

Interview transcripts

This chapter summarizes the discussions carried out with different people at public institutions in the Netherlands.

B.1 Rijkswaterstaat (RWS)

This interview has been carried out with Aart van Sloten at RWS. Rijkswaterstaat is the institution that handles the practical execution of public works and water management. They have a lot of data in their databases which is either sensitive or non-sensitive. Very few datasets are known to be somewhere in the middle.

The approach they use to publish data is very simple. If there is the slightest proof that a data set contains some sensitive data, then do not publish (e.g. the data contains addresses of individuals). Examples of types of data that do not get publish include company and competition data, country security, information about prisons, information on Defense, full building layouts, technical details of tunnels, environment information (to prevent rare specie hunting). What they usually publish and are not worried about is general geographical data (e.g. roads / waterways). One of the gray area data sets is about the NAP bolts. These are copper bolts throughout the Netherlands which have a known (pre-measured) altitude. They are not sure whether to release the exact positions of these bolts due to several simple reasons: copper can be stolen for money, some of the bolt lie on private properties and are recorded as such (address, names).

Due to the nature of the institution, they do not have a lot of data about people. When information needs to be published about such data sets (e.g. information on road accidents) they ask CBS to correlate the data with other statistics (e.g. how many people actually went to the hospital) and release general statistics about the incident (region level, big city level).

From the discussion we have noticed that RWS does not have a complex process for reasoning about privacy. As such, there is no process to anonymize the data. Privacy is protected through secrecy - non-disclosure. A framework could help in setting up a process which deals with the anonymization of the data locally, instead of delegating the task to third parties.

B.2 Kadaster

At Kadaster we have interviewed Dick Eertink. Kadaster is the institution which manages parcel information within the Netherlands. Each parcel has its own number, location, size, type (building space / agricultural space). To this there are transaction documents linked which define the last sell event. Who was the buyer, who was the seller, for what amount was it sold and what mortgage has been used. From these documents one can retrieve the name, address, postal code and city of the buyer/seller. The BSN is also stored, but that is not released to the public.

The access to the data is allowed through the online interface. One can request information about one parcel at a time. The only limitations in place are the licence agreement (use the information just for yourself) and the fact that it costs 3.50 euro per inquiry. This for example does not protect an individual if an attacker acts in a targeted manner.

They also give big datasets for different purposes, mainly internally to the government. There are strict rules on usage (such as WBP - dutch privacy law) and they try to verify that people respect these rules, but in practice it's not that easy.

There are three types of data: parcel information, transaction information and owner/former owner information. Access is given only to the newest information (no history given such as past owners). Yet, with some effort, most of the history can be reconstructed. The CBS (college bescherming persoonsgegevens) is currently debating on how to handle these categories. The desire is to eventually remove the costs of inquiry and make this Open Data.

Other information they manage includes:

- national topography
- address and building/land type for that address (mostly already public)
- act as an intermediary for information about cables and underground pipes

There is currently no process in place to anonymize the data and deciding on how to publish sensitive data sets is not easy. The laws are not yet very specific w.r.t. types of data they handle. They expect this year (2013) new versions for the Dutch and European privacy laws.

B.3 Statistics Netherlands (CBS)

At CBS we had an interview with Peter-Paul de Wolf and Anco Hundepool. The goal of CBS is to publish relevant and trustworthy statistics. They gather a lot of information from individuals and companies and thus must handle it with great care. Most of the time they deal with microdata of people and sometimes of companies. They said protecting company microdata is not possible, in general, since companies are too easy to identify in the crowd.

They took part in the development of several tools in collaboration with other statistical departments in Europe. What they mostly use for anonymizing data here in the Netherlands are the generalization and suppression techniques (present in the μ-argus tool). Other methods include PRAM (post randomization method), which they tried a few times. The

problem with this method is the sanitized data. It is very hard to use and one needs all sorts of corrections to any statistical operation performed, in order to compensate for PRAM.

Their microdata can be released in three formats where data is aggregated into bins. k represents the minimum size of the bins.

- public use files - accessible to all, the rules for this include having no less than 200000 respondents per region, and a k parameter of 10000.
- under contract data - given for research purpose only - somewhat less anonymized (k is in the range of 100 to 1000)
- data that stays on CBS - again, for research only - even less anonymized if at all.

The third type is very interesting. Researchers either come on-site and use the dataset based on the tools available on the premises (e.g. SPSS) or they access the data remotely, but only get to see what the tool outputs on the screen. There is no dataset transferred. This is becoming more and more popular and the request of anonymized datasets is becoming less popular. To manage this, a strict screening process has been put in place. The results are inspected and one must be able to show the steps performed to achieve those results. Transparency is key.

Upon requesting some data sets, it was suggested that it would be easier to work with synthetic data. The US has a reputation of generating quality synthetic data. It would be easier since requesting data is too complex for my goal - requests need to be filed, then approved, and at most, Type 2 data would be provided, which is not very useful for our research goal.

The Anonymization Process

CBS uses the notion of key attributes (QIDs) which can be used to re-identify individuals. Three categories of attributes can be distinguished: identifiable, more identifiable, the most identifiable. Based on these three categories, they try to make combinations (2-3 up to 10-20 in mu-argus) that meet a certain non-uniqueness criterion (e.g. no less than 100 per combination). It falls onto them to decide which attributes fall under which category. There are some standard attributes and others are simply agreed upon within CBS - experience/gut feeling plays a big role here.

Risk decision making is based on combination frequency. Usually, the data is based on a representative population sample. Sometimes combining this with information from GBA or some other administrative institution (not always possible) is required to be able to reason about the sample. If the sample is not big enough, they use different techniques to estimate the population. Once the population is known, it can be checked whether a certain combination is frequent or rare.

During the generalisation process, choosing which column to generalize first is done based on experience - their statistics department knows which columns are more important, in general, to researchers. Even so, researchers are never happy with the data they get (the data is always anonymized towards certain purposes). To prevent privacy breaches by sequential release, they only anonymize a dataset once.

Data types, thresholds and measurements

The data does not have to be numerical, since their program only looks at frequency of combinations. Threshold value, k is usually 100. It has been determined based on experimentation with the data. 10 is too little, 1000 is too much for the second type data.

When reasoning about utility, they do this more on feeling than on measurements. It is hard to measure utility if you do not know the purpose of the data. One possible idea would be to generate several utility measurements, aimed at different tasks (data mining, query answering etc).

One topic they have interest in is if there exists a better way to determine record/table risk, other than combination frequency?

B.4 Amsterdam Economic Board (AEB)

At the Amsterdam Economic Board we had an interview with Ron van der Lans en Jasper Soetendal. AEB tries to improve economic growth by bringing together people from different institutions / organisations (CEO's, managers, scientists, researchers etc).

From the discussion it was clear that the most person related data that they have is in the Dienst Basisinformatie (DBI - basic information service). As we have observed from previous interviews, they rely on an external party, this time O&S (a research and statistics department of Amsterdam), to publish their data by means of aggregation. The aggregation levels differ from regions to city regions to neighborhood combinations. Other rules that apply are for example that there must be at least 3 to 5 people in every aggregation. The only publicly available data sets are the ones about electricity and gas. The data in this case is aggregated on building level.

They are in the process of opening up their data, but most of the time, one can simply access this data by requesting it at the local town hall.

In determining what data has privacy issues, they rely on common sense, experience and whether or not the data is about people. Usually, the data that AEB has is not so sensitive (e.g. trashbins, lamp posts etc). There are currently about 160 datasets published[1]. In the future, they will probably have more than thousands of datasets that will be published. Some examples for which some security has been taken is data on fire alarms - the street and approximate geo-coordinate has been released. They are also looking into how data on public works should be released. It contains information (phone number, address and other information) of the person in charge of the works.

B.5 IBM Ireland

Aris Gkoulalas-Divanis is one of IBM's researchers that are currently working on privacy. Other topics covered by his research include Dublinked and mobility data. Currently, he is doing a postdoc on medical data (anonymizing medical data). For Dublinked, they try improve on how to decide on the vulnerability of a data set. Currently, this is done based on

[1] `amsterdamopendata.nl`

experience. They are manually inspecting data and the only automated tasks are generating histograms for attributes and identifying unique combinations (e.g. if {sex=male,age=55} is unique). On a macro level, he is researching "knowledge hiding", which is essence is preventing people to understand patterns in data by reducing frequency of the patterns.

He has been experimenting with different kinds of data: relational, transactional, sequential data, each requiring a different approach to protect.

Regarding Open Data, it only makes the problem more complex. With the opening of all the new data sets, the data could be used in unforeseen ways (combination with other data sets).

They looked at the The Health Insurance Portability and Accountability Act (HIPAA) and other similar regulations, but they only provides minimum requirements; it is not enough to actually protect the data.

Talking about anonymization techniques, we have learned that an efficient anonymization technique will only lead to less utility since it will cut corners on utility to finish faster; the focus of such algorithms is on privacy. A good anonymization algorithm seems to be Mondrian (does k-anonymity) by recursively partitioning the space. It achieves a good balance between privacy and utility.

As far as run time goes, he noted that slower algorithms take hours. From his experience, a k value of 5 is enough for medical data. Most approaches can be parallelized which reduces the overall computation time.

Anonymity levels - how they decide on parameters and data sensitivity:

- measure re-identification risk by studying which elements are unique in the dataset
- identify outliers
- reason based on this how sensitive the data is

Three types of utility measures that they use:

- IL (information loss): each generalisation increases the IL (general measure)
- based on workload (for which goal is the data anonymized): provides more utility for specific tasks
- average aggregation query - what is the error on these queries (general measure)

Selecting the QID is done based on type of data. This reflects the need for prior experience. In the case of medical data, selecting a QID is relatively easy because the data and its sensitivity is well defined. In other areas this is more difficult.

For visualizing the trade-off between risk and utility, R-U confidentiality maps can be used. The metrics to be used to measure risk and utility vary, depending on the data type and publication goal. In the case of Mondrian, one measure for risk can be for example 1/k (most risky individual).

There are also two IBM products which handle data sanitization.

- Infosphere Optim: implements masking approaches to protect sensitive data. It uses auxiliary data dictionaries (for example to replace names). In essence, it generates a new data set. Data masking is not the same as anonymization.
- Infosphere Guardium works with reduction. It automatically identifies data, identifies sensitive words (patient names) and removes them. This may turnout to decrease utility too much.

Appendix C

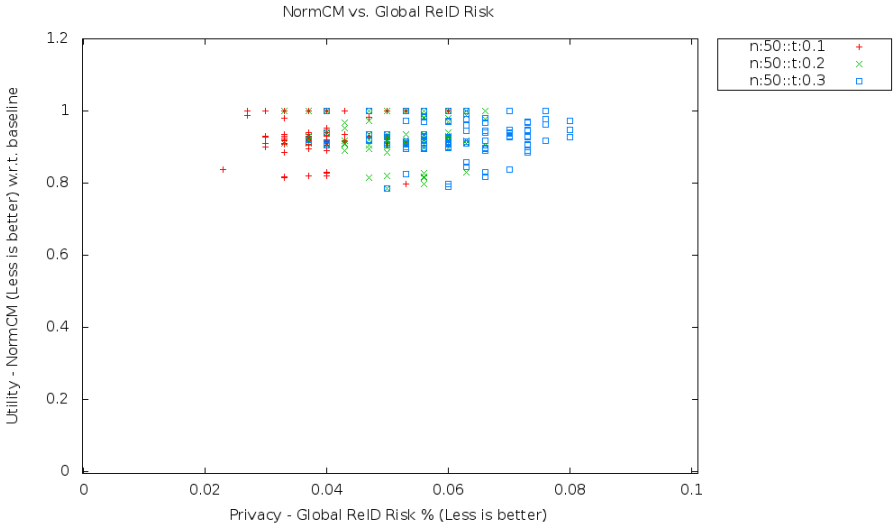
Experiment plots

In this chapter one can find most experiment plots. We placed them here to accommodate for the large number of figures.

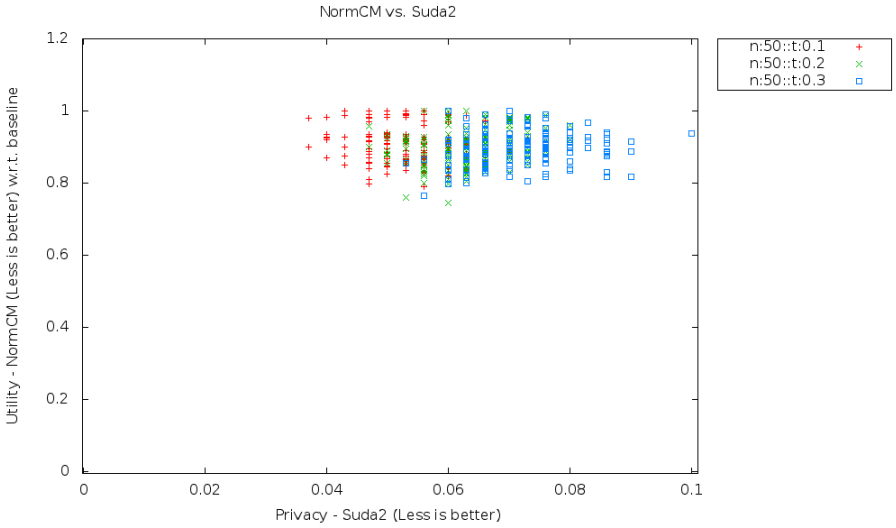
- Mondrian_KNT figs. C.1 to C.9
- Mondrian_NT figs. C.10 to C.18
- Incognito_T figs. C.19 to C.21
- Incognito_K figs. C.22 to C.24

The comparisons between algorithms are structured based on the metric used:

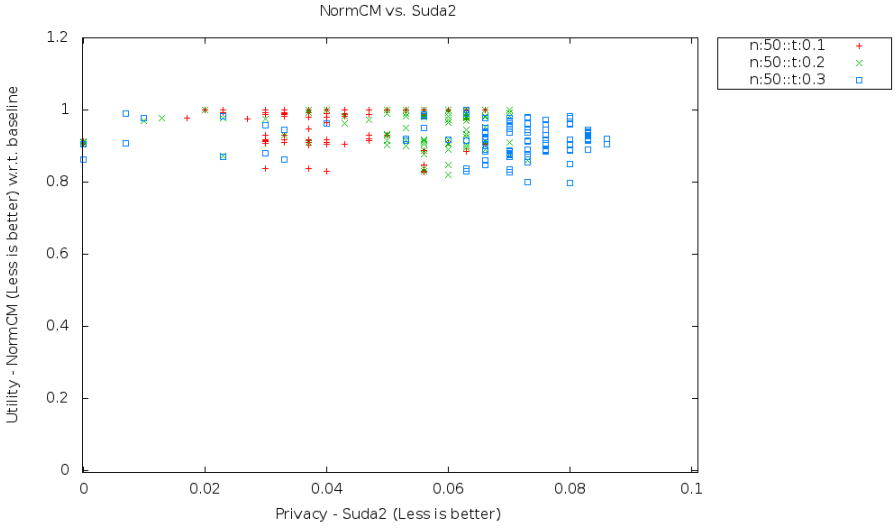
- NormCM - fig. C.25
- NormDM - fig. C.26
- NormAvgECSize - fig. C.27



(a) QID-4

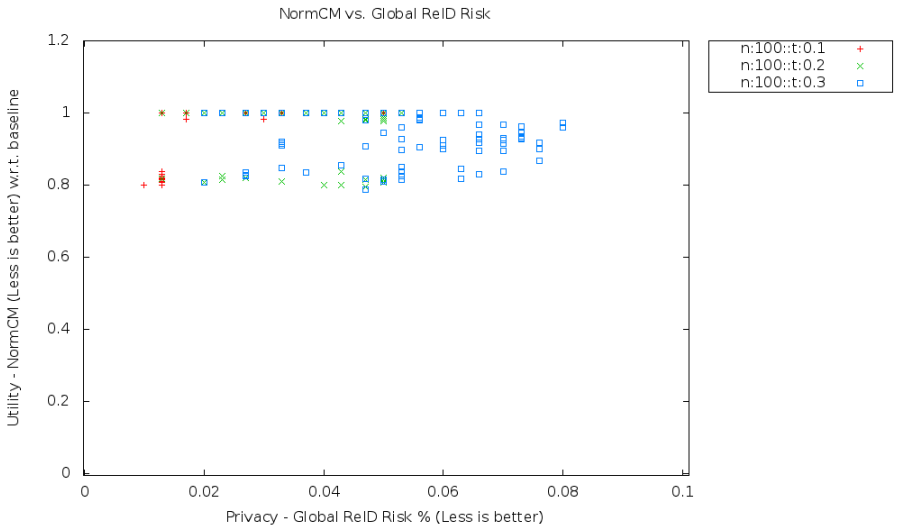


(b) QID-7

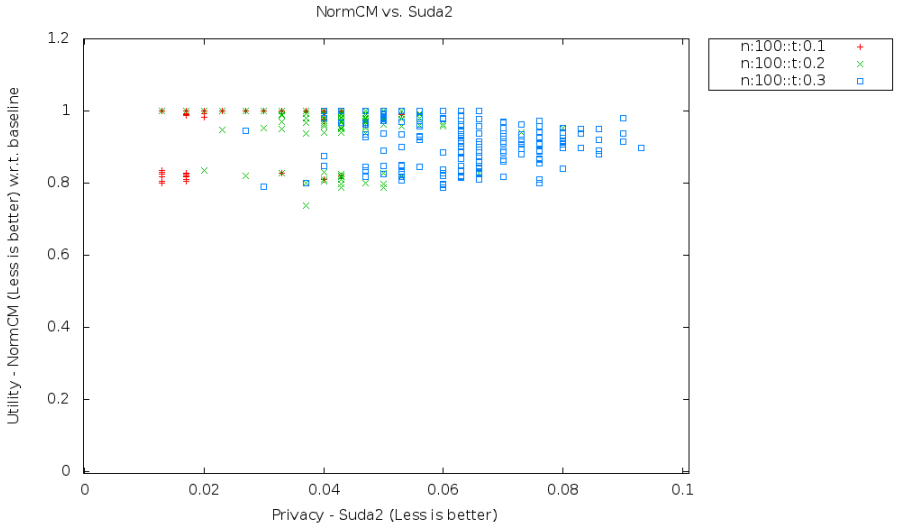


(c) QID-13

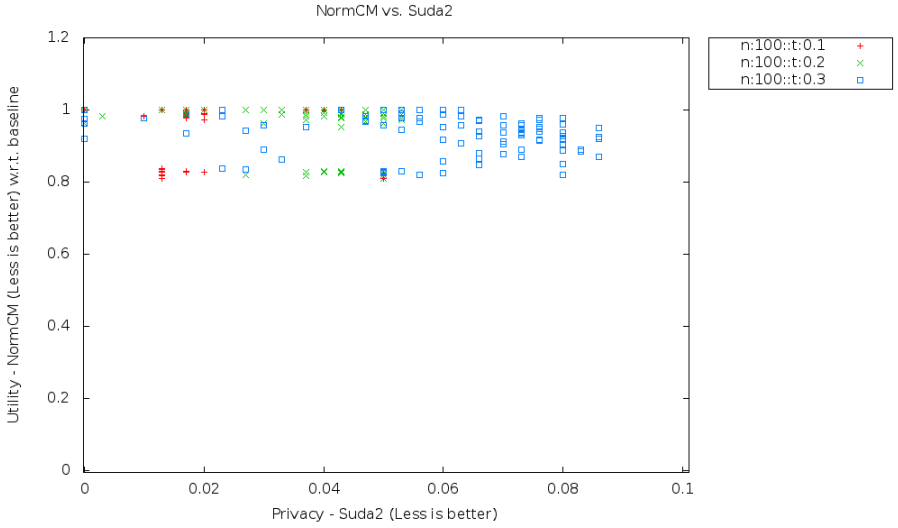
Figure C.1: Mondrian KNT: normalized CM (n=50)



(a) QID-4

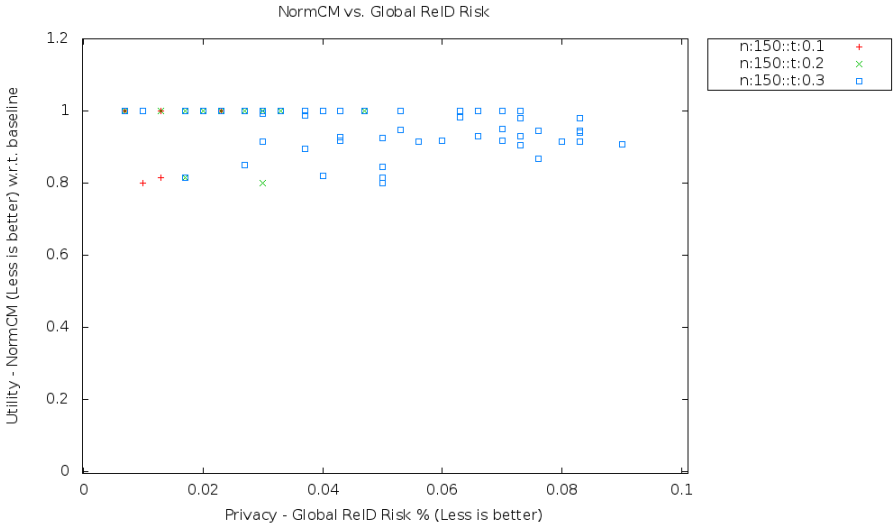


(b) QID-7

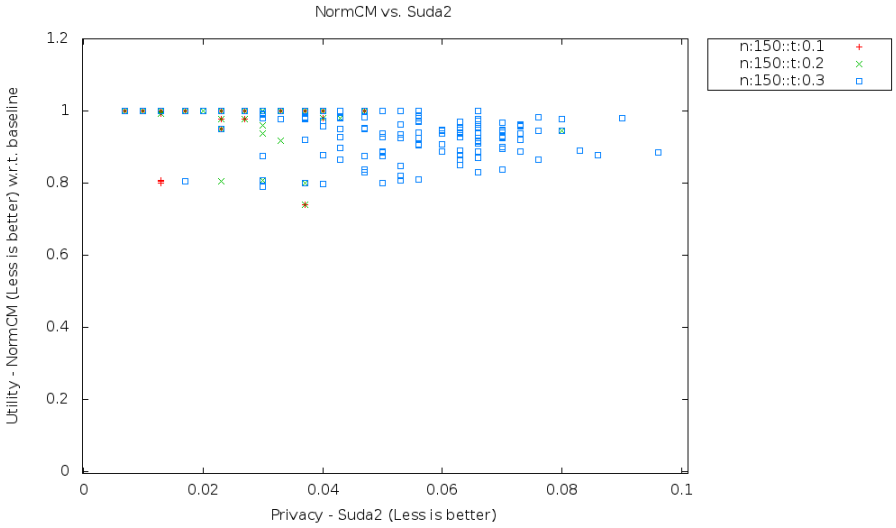


(c) QID-13

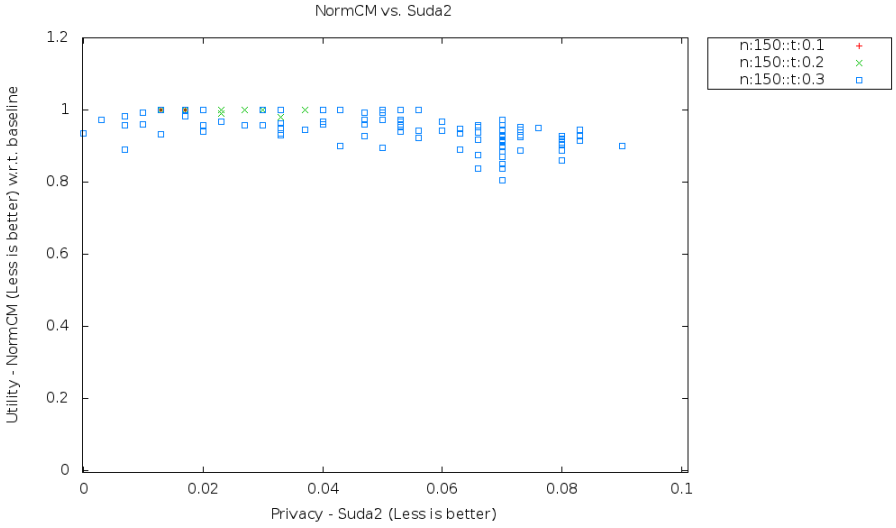
Figure C.2: Mondrian KNT: normalized CM (n=100)



(a) QID-4

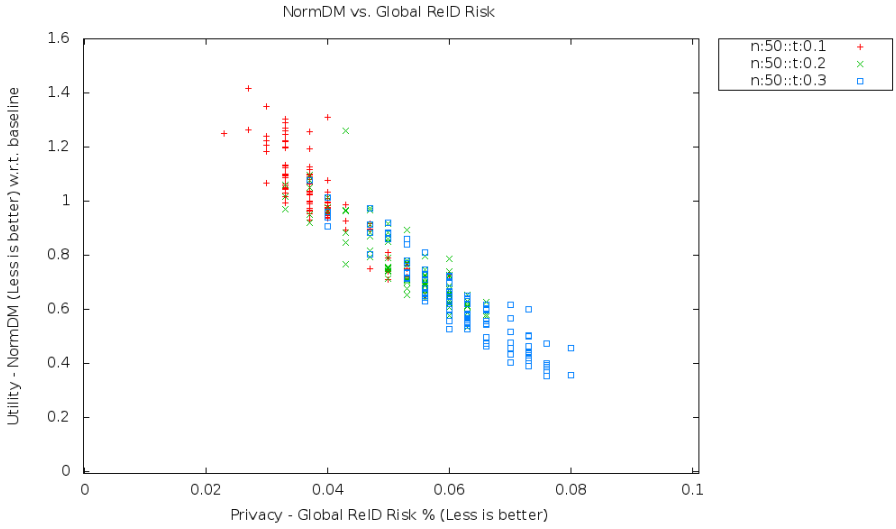


(b) QID-7

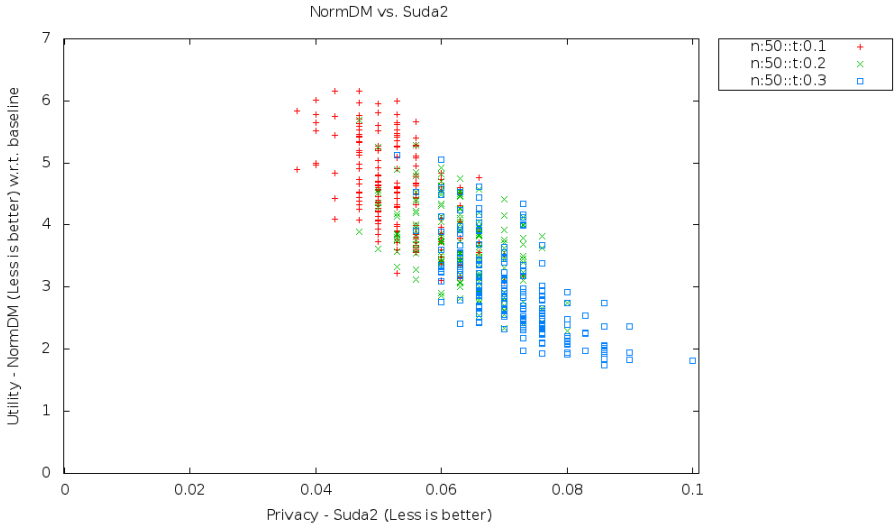


(c) QID-13

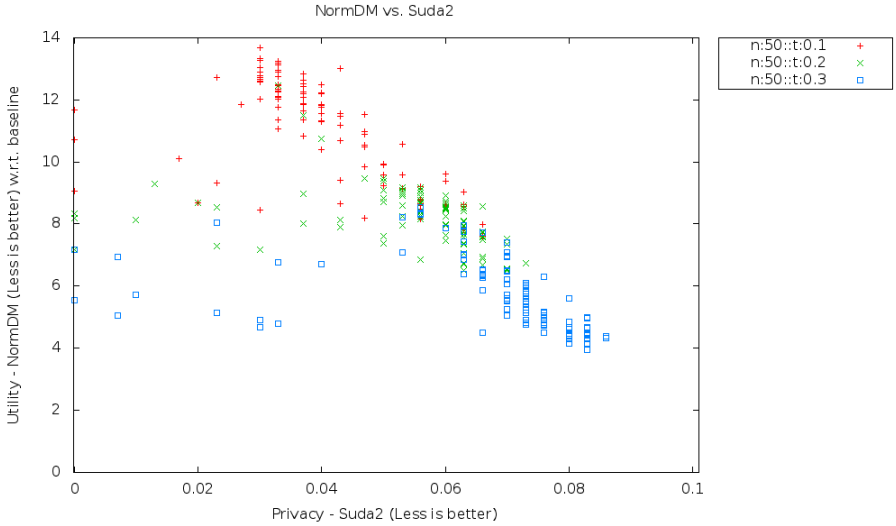
Figure C.3: Mondrian KNT: normalized CM (n=150)



(a) QID-4

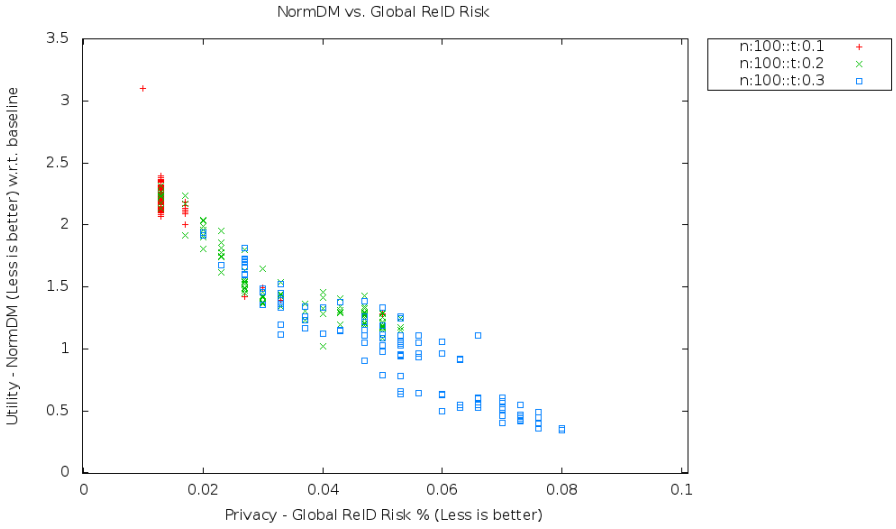


(b) QID-7

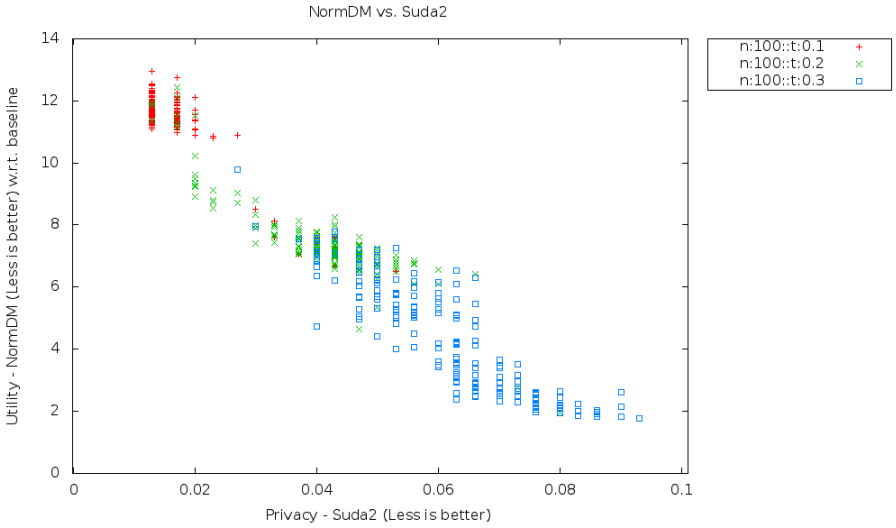


(c) QID-13

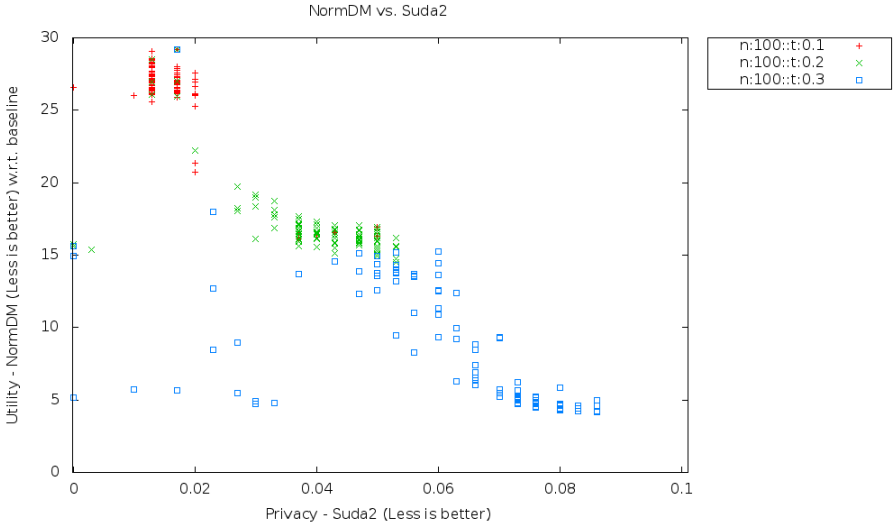
Figure C.4: Mondrian KNT: normalized DM (n=50)



(a) QID-4

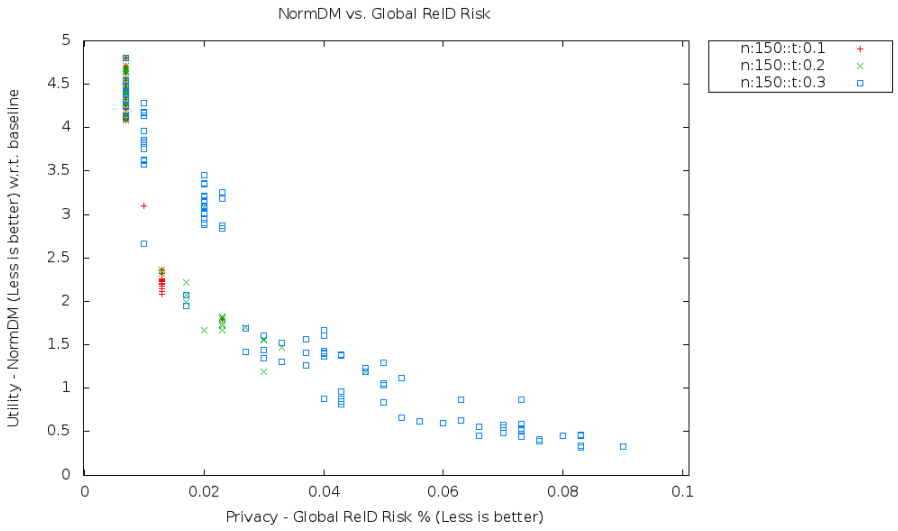


(b) QID-7

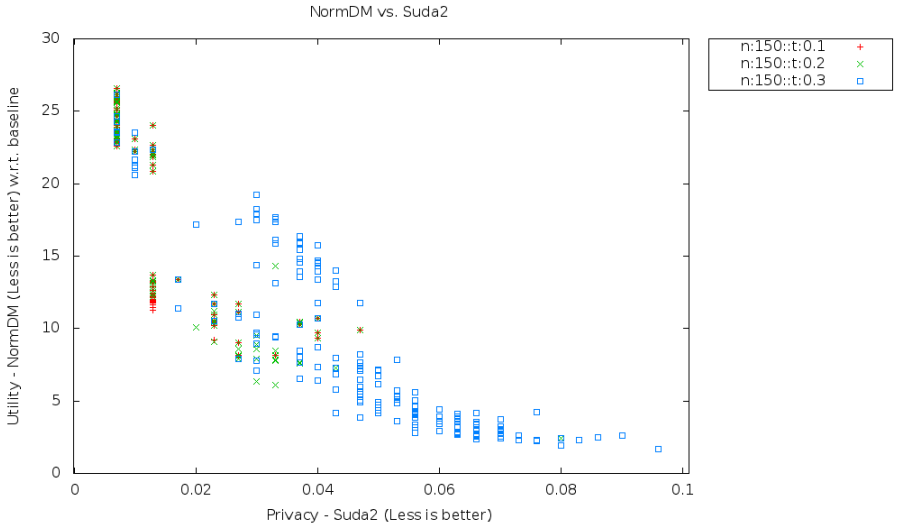


(c) QID-13

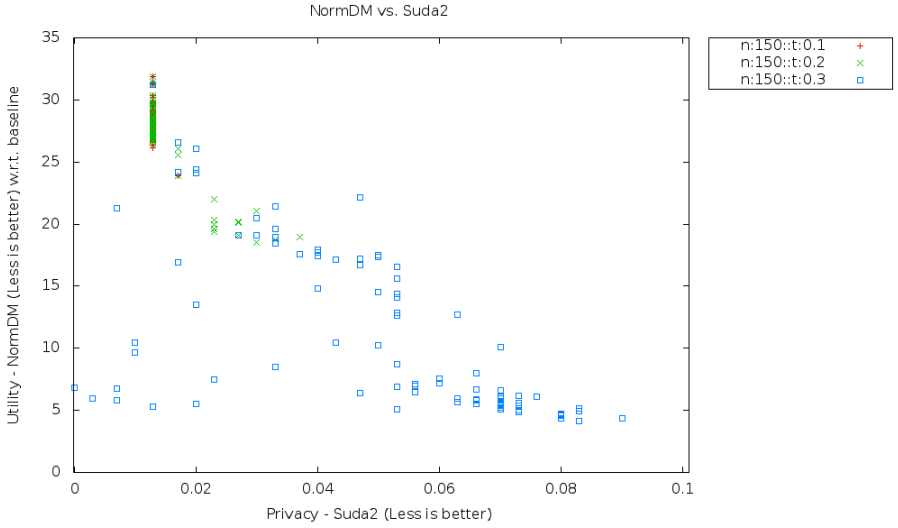
Figure C.5: Mondrian KNT: normalized DM (n=100)



(a) QID-4

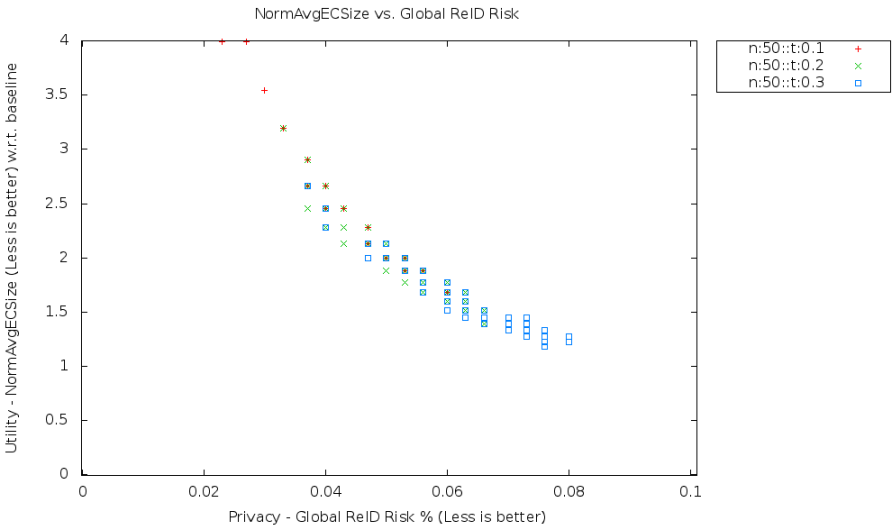


(b) QID-7

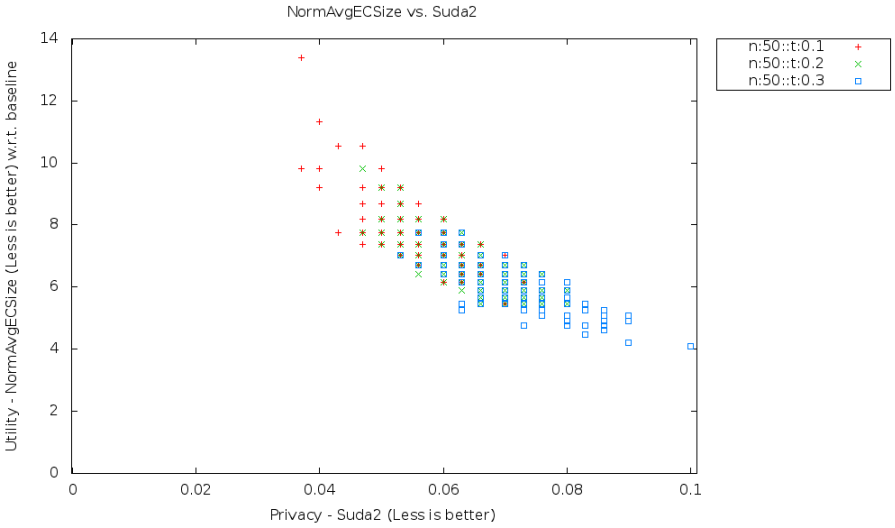


(c) QID-13

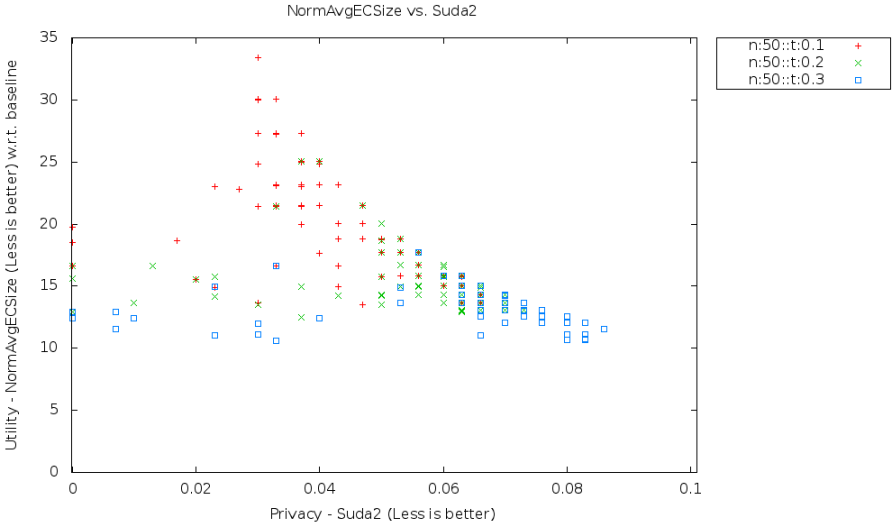
Figure C.6: Mondrian KNT: normalized DM (n=150)



(a) QID-4

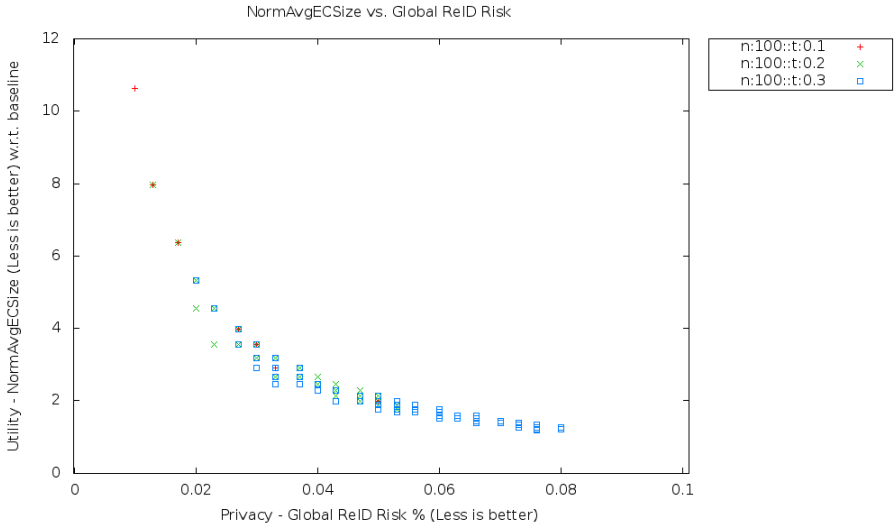


(b) QID-7

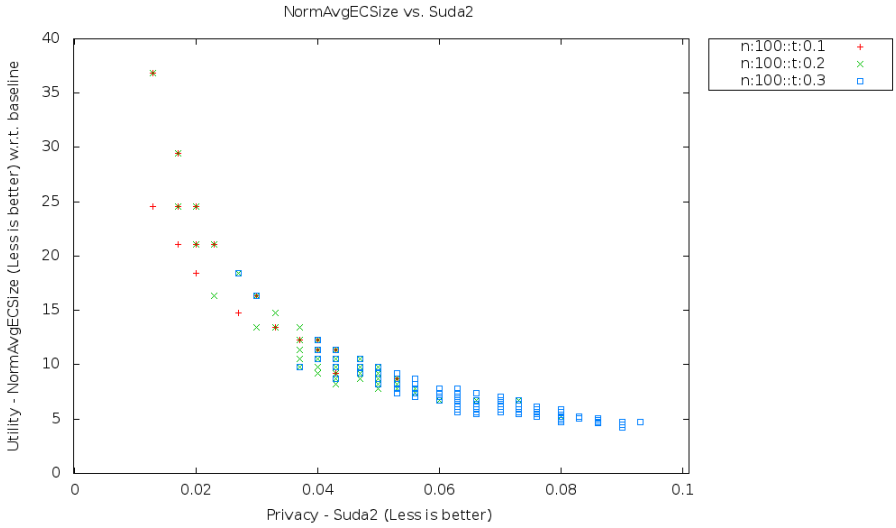


(c) QID-13

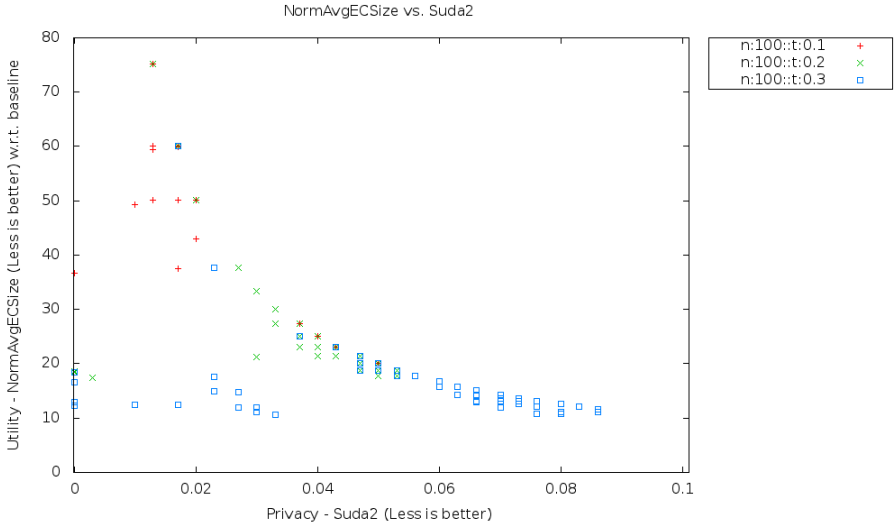
Figure C.7: Mondrian KNT: normalized avg. EC size (n=50)



(a) QID-4

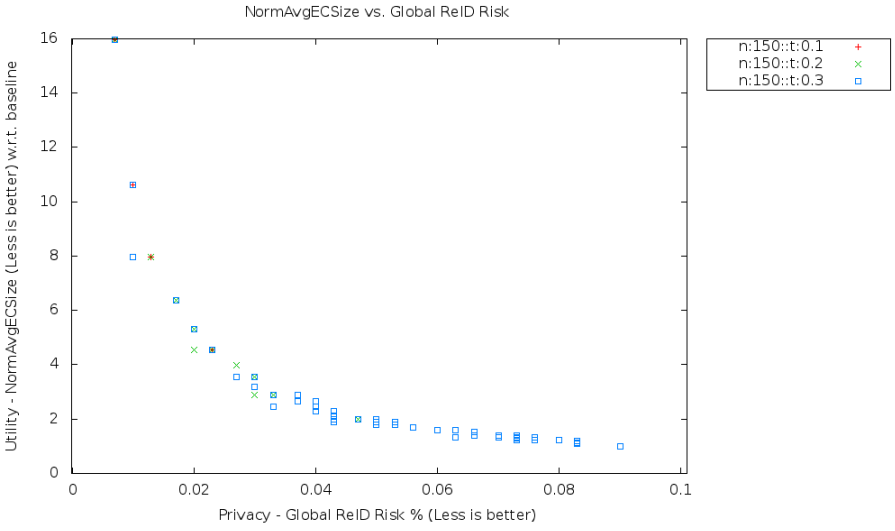


(b) QID-7

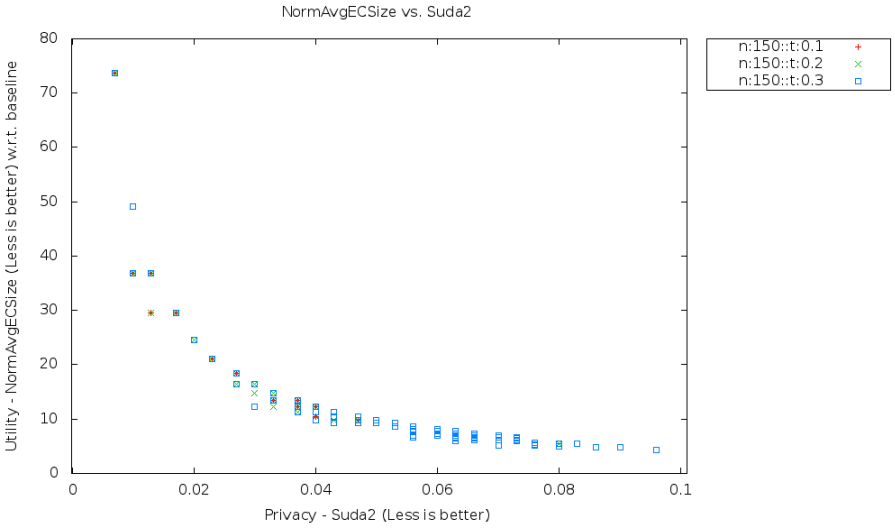


(c) QID-13

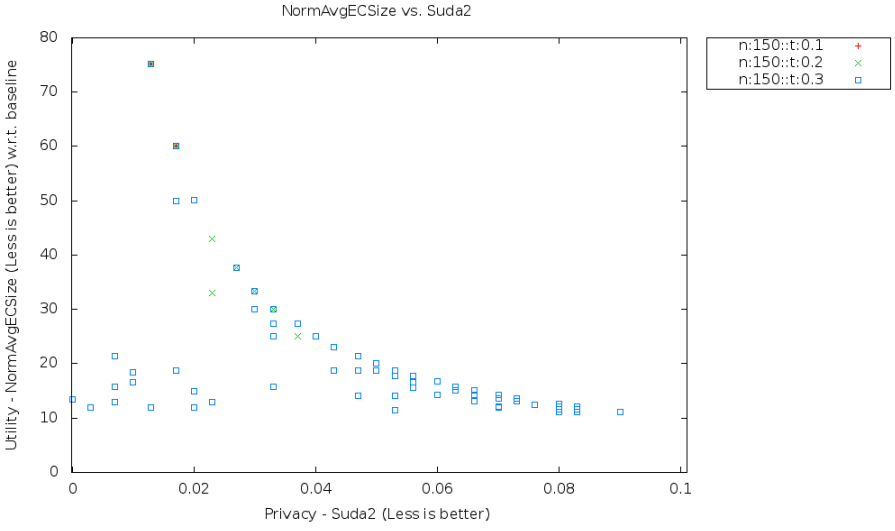
Figure C.8: Mondrian KNT: normalized avg. EC size (n=100)



(a) QID-4

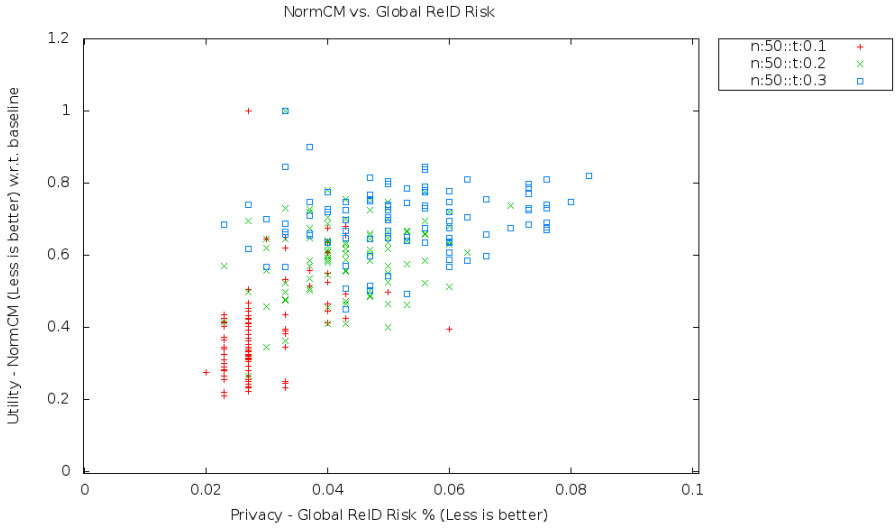


(b) QID-7

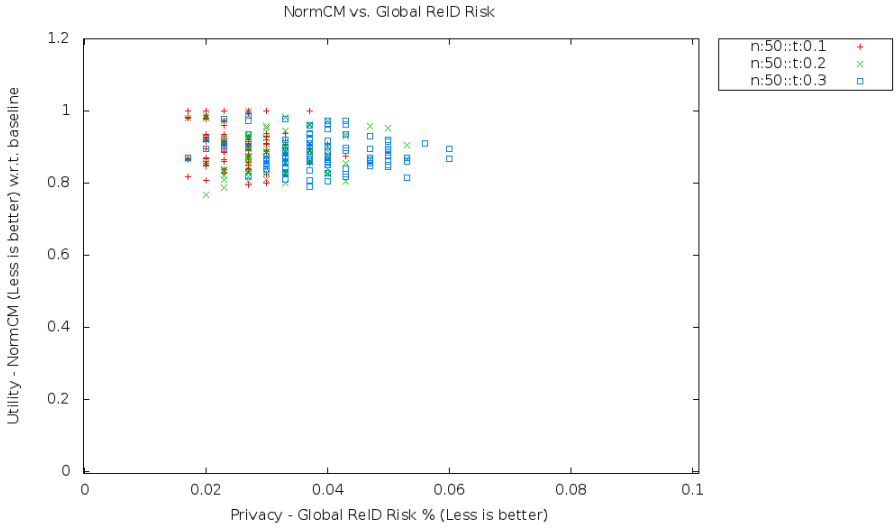


(c) QID-13

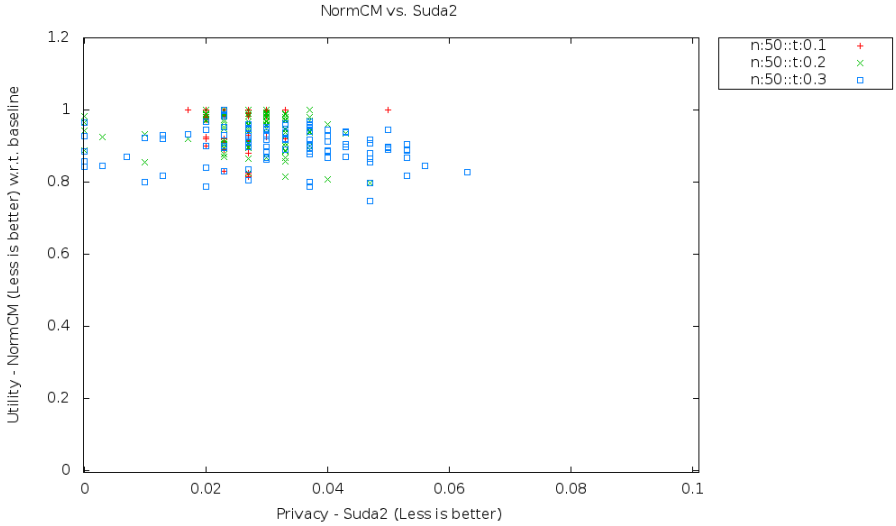
Figure C.9: Mondrian KNT: normalized avg. EC size (n=150)



(a) QID-4

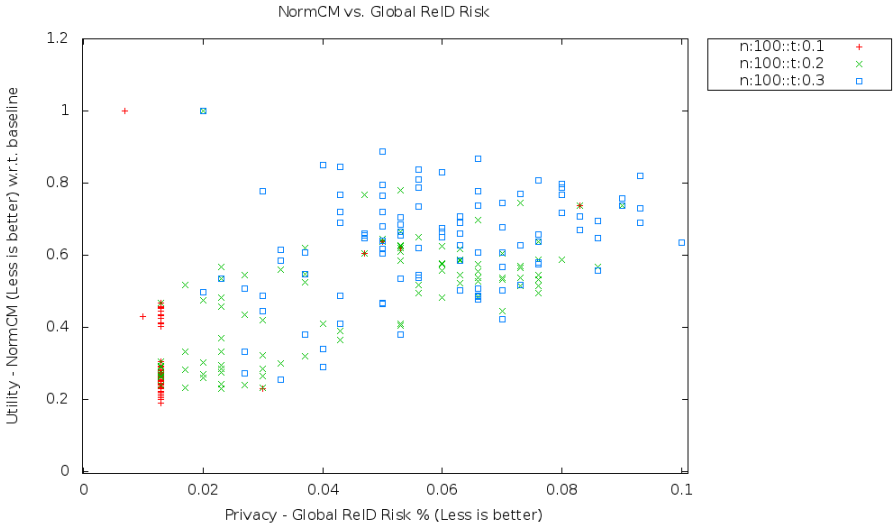


(b) QID-7

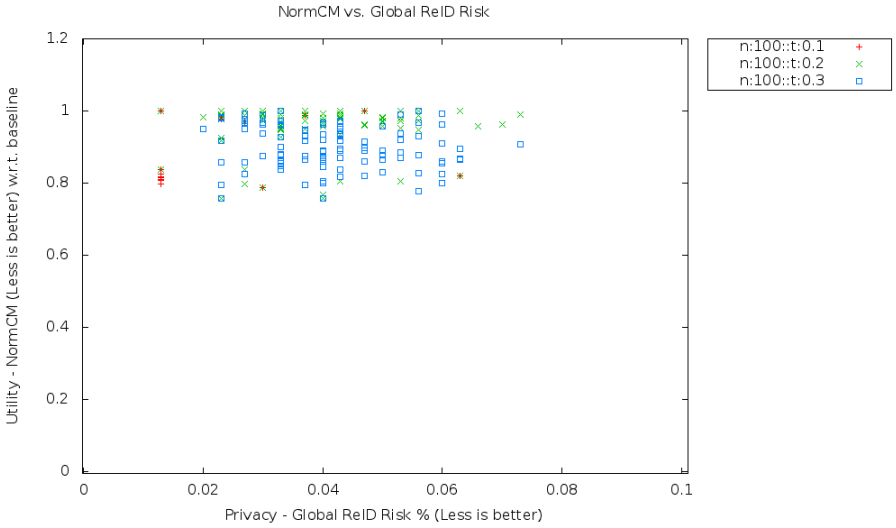


(c) QID-13

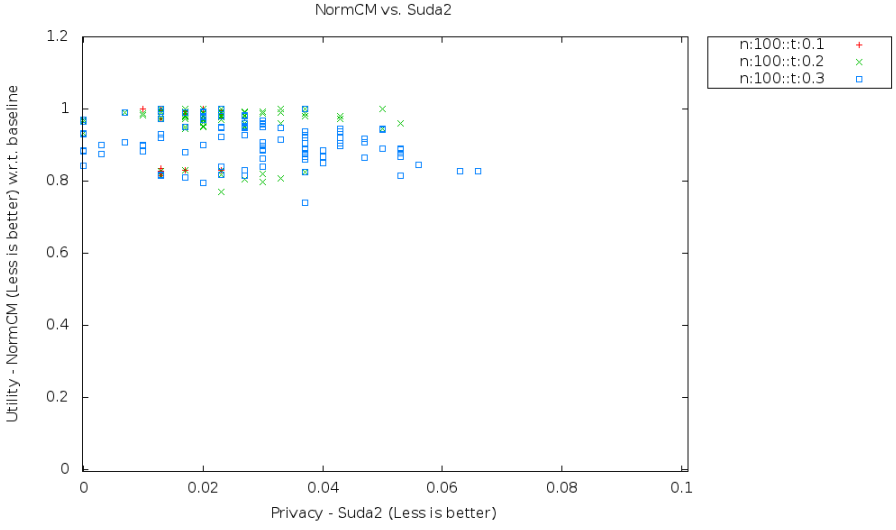
Figure C.10: Mondrian NT: normalized CM (n=50)



(a) QID-4

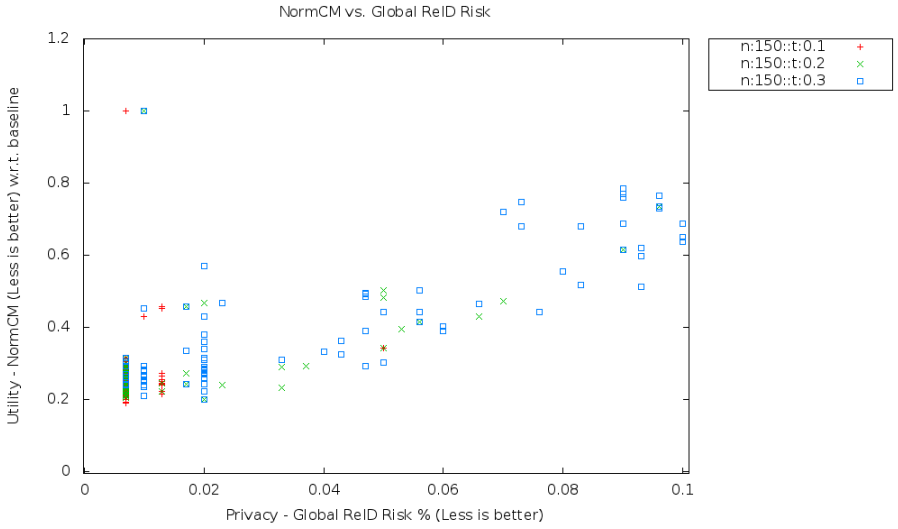


(b) QID-7

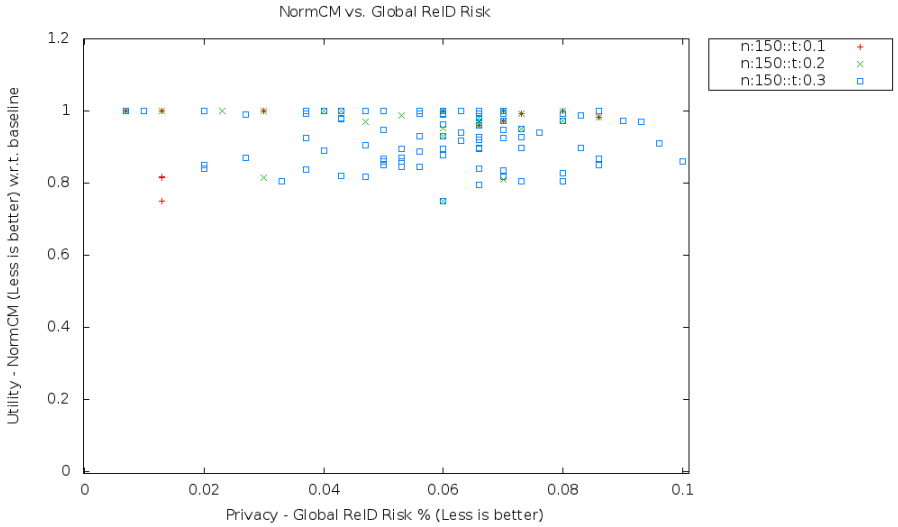


(c) QID-13

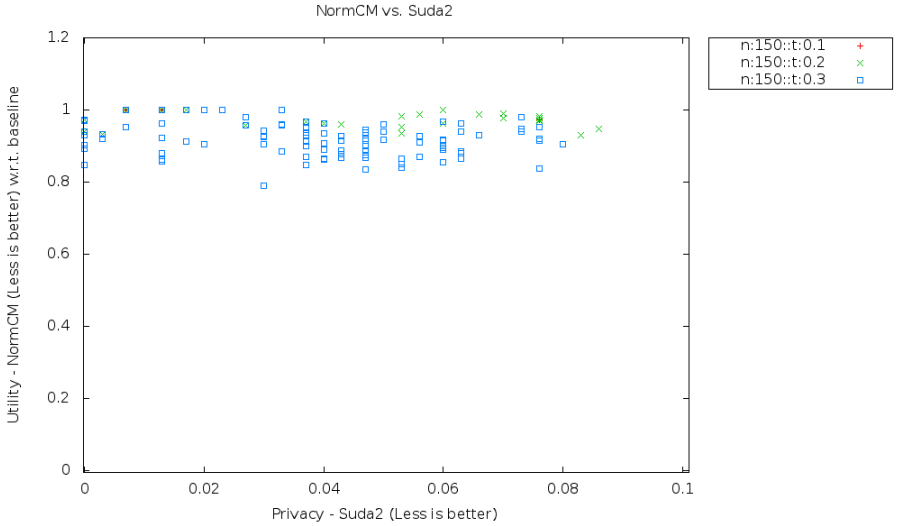
Figure C.11: Mondrian NT: normalized CM (n=100)



(a) QID-4

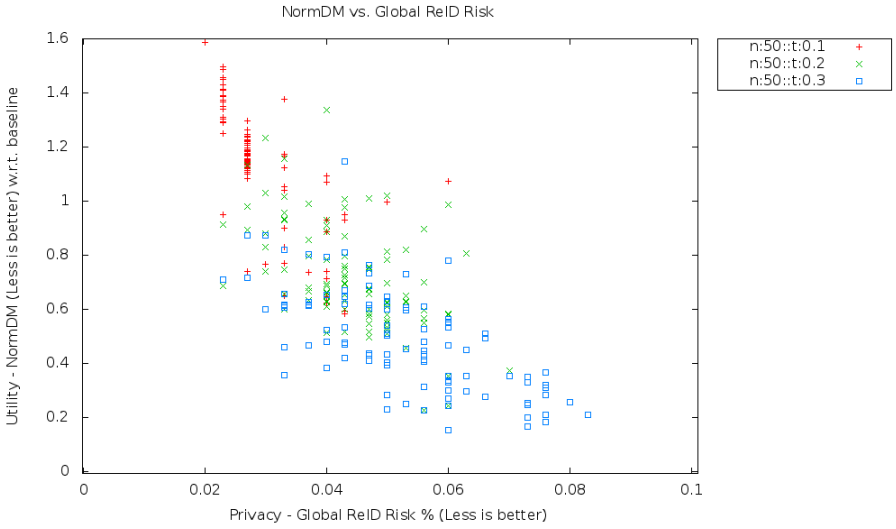


(b) QID-7

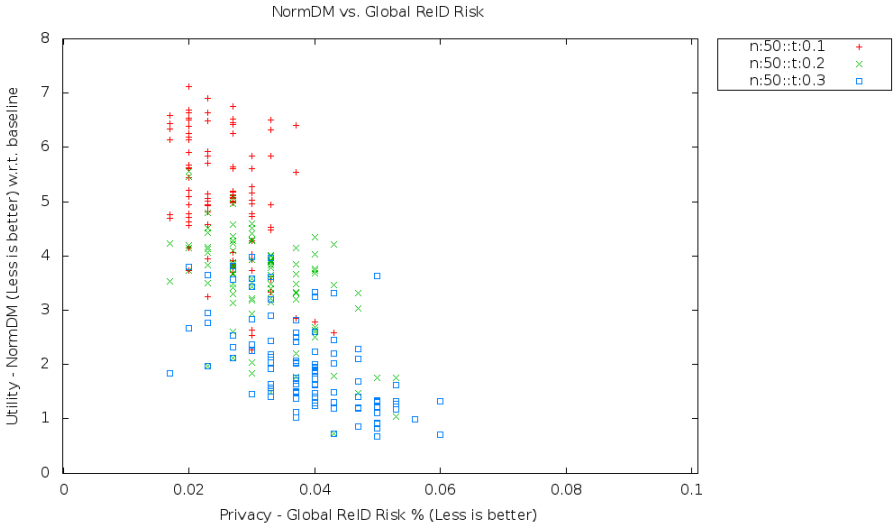


(c) QID-13

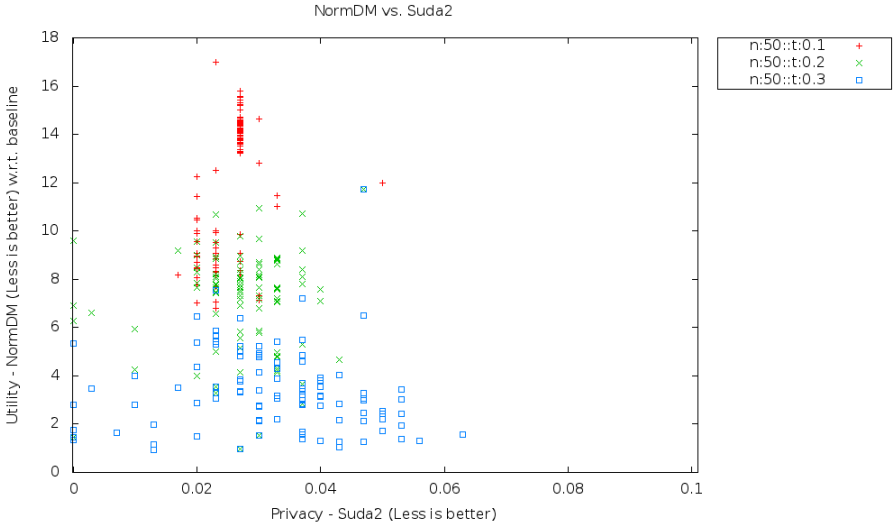
Figure C.12: Mondrian NT: normalized CM (n=150)



(a) QID-4

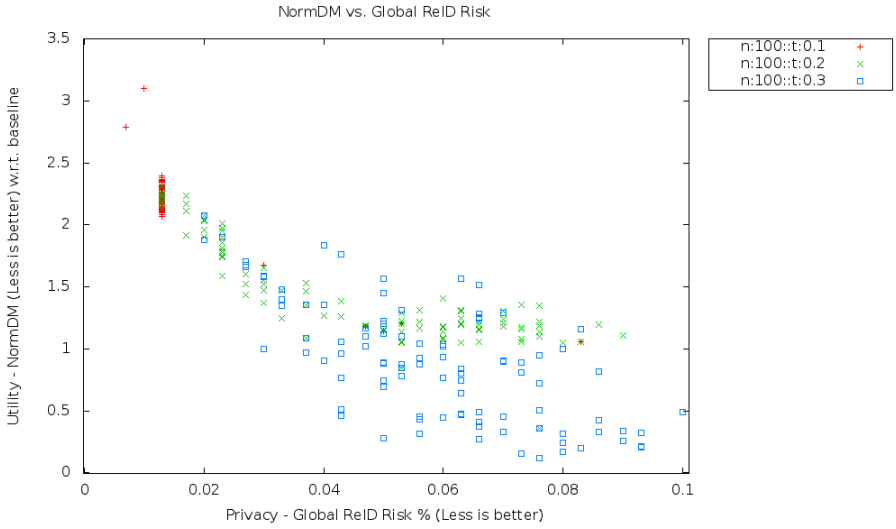


(b) QID-7

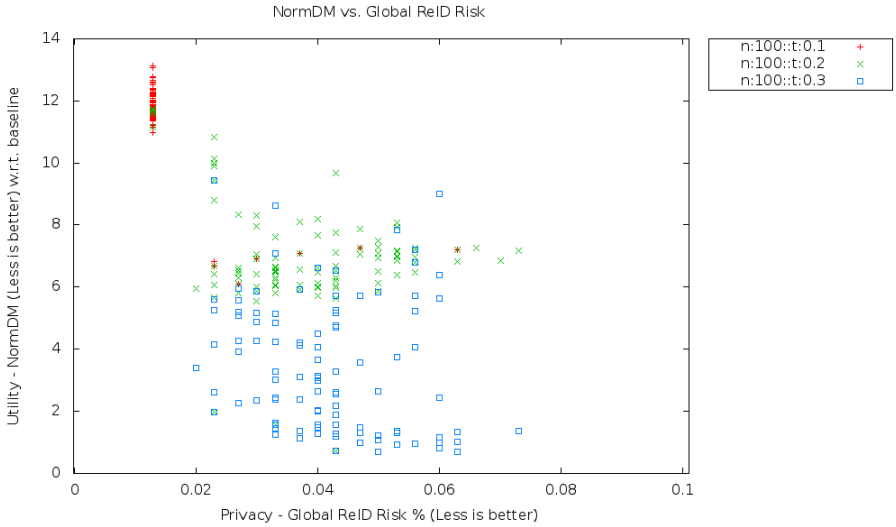


(c) QID-13

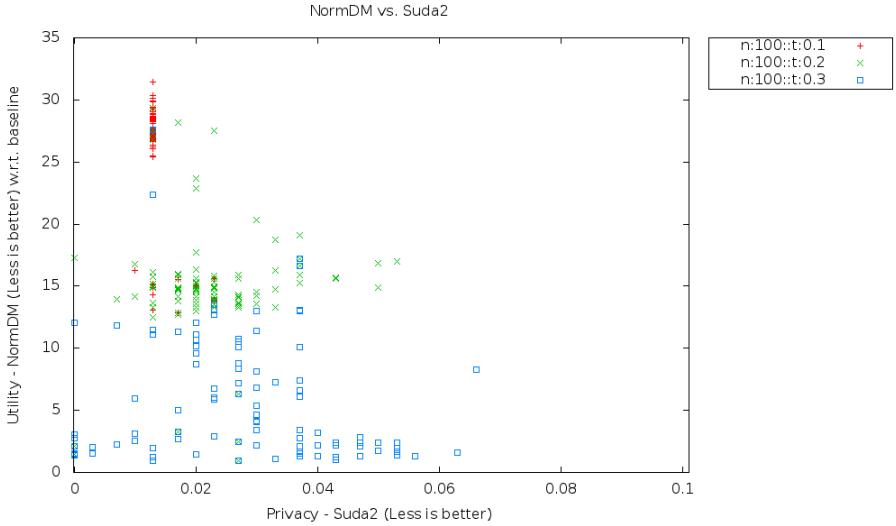
Figure C.13: Mondrian NT: normalized DM (n=50)



(a) QID-4

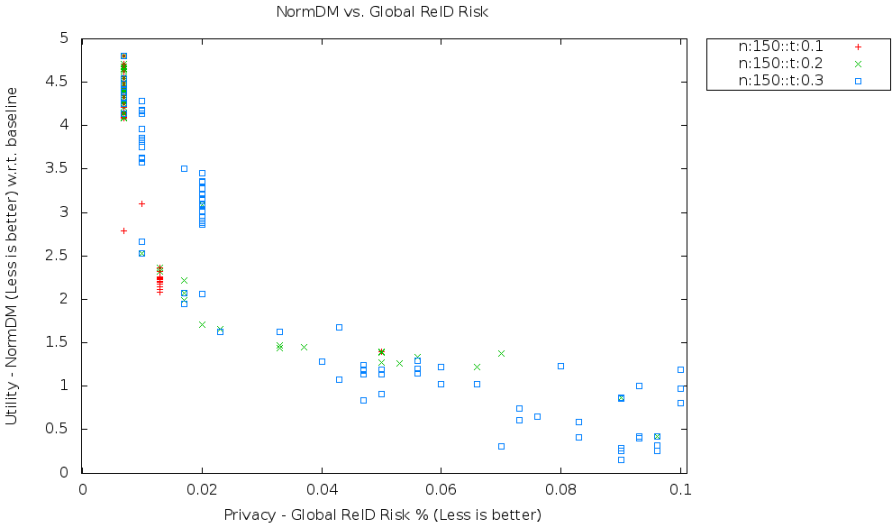


(b) QID-7

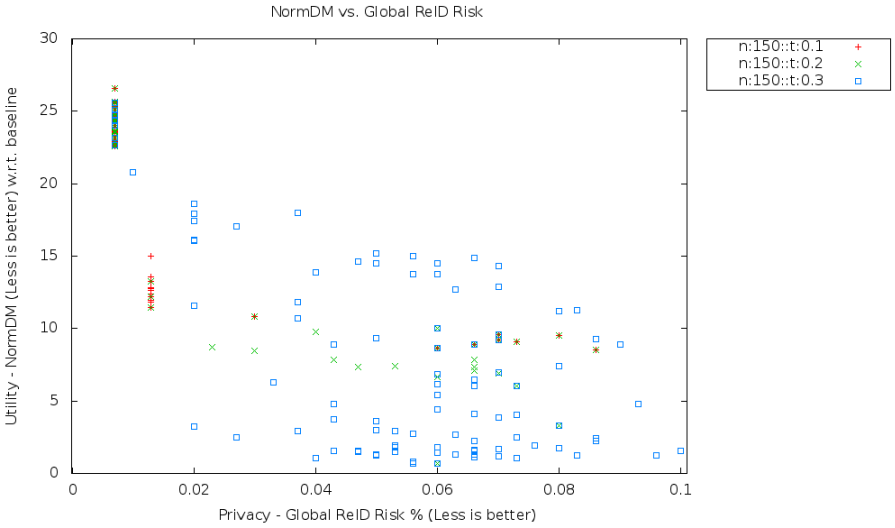


(c) QID-13

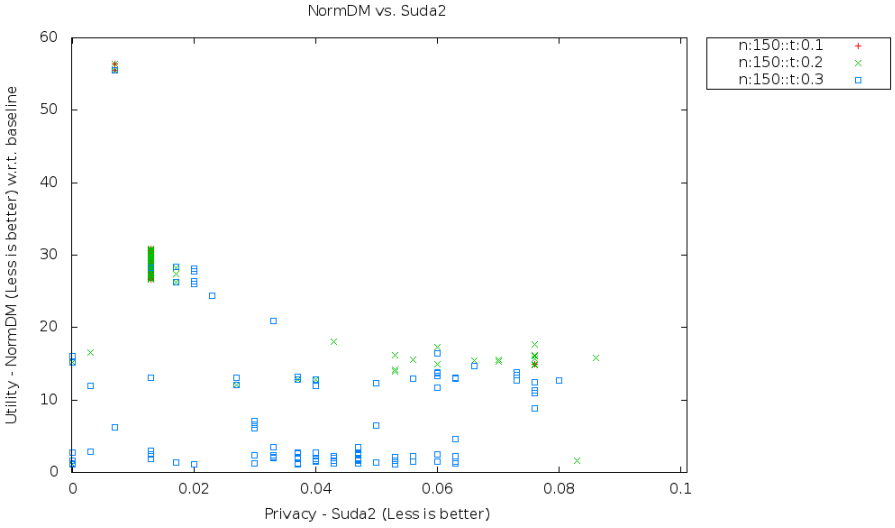
Figure C.14: Mondrian NT: normalized DM (n=100)



(a) QID-4

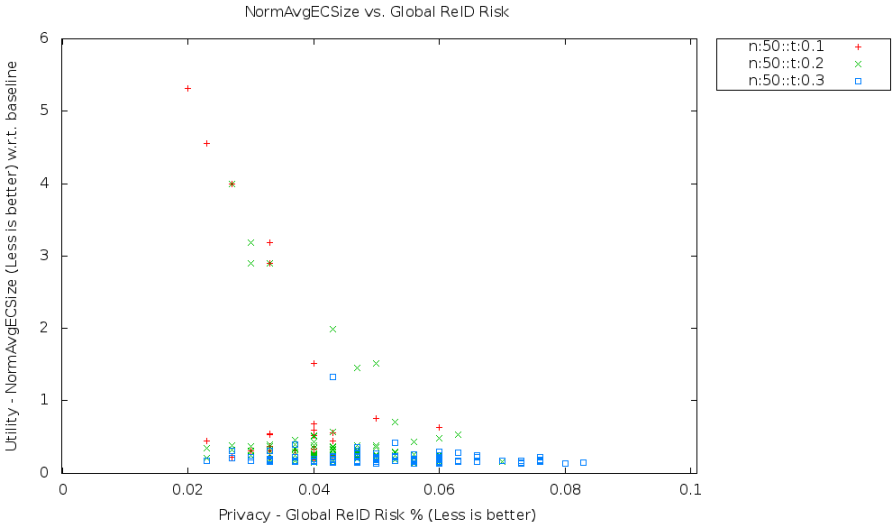


(b) QID-7

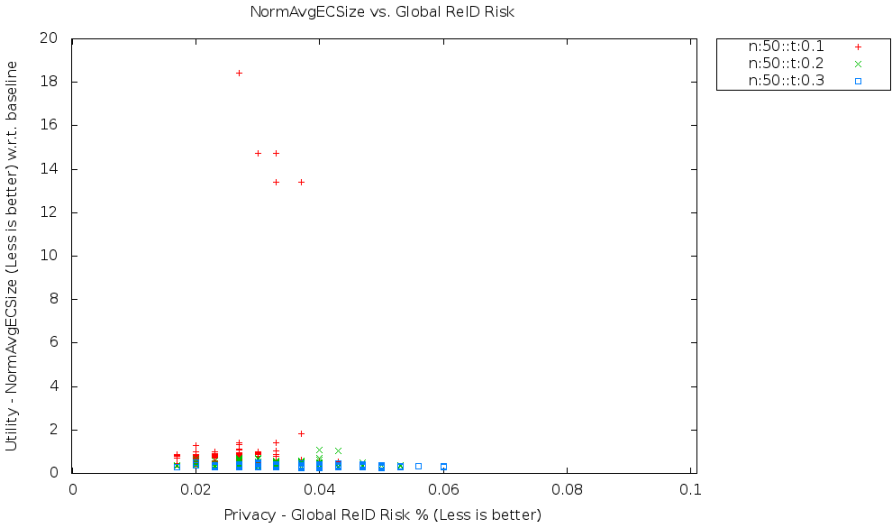


(c) QID-13

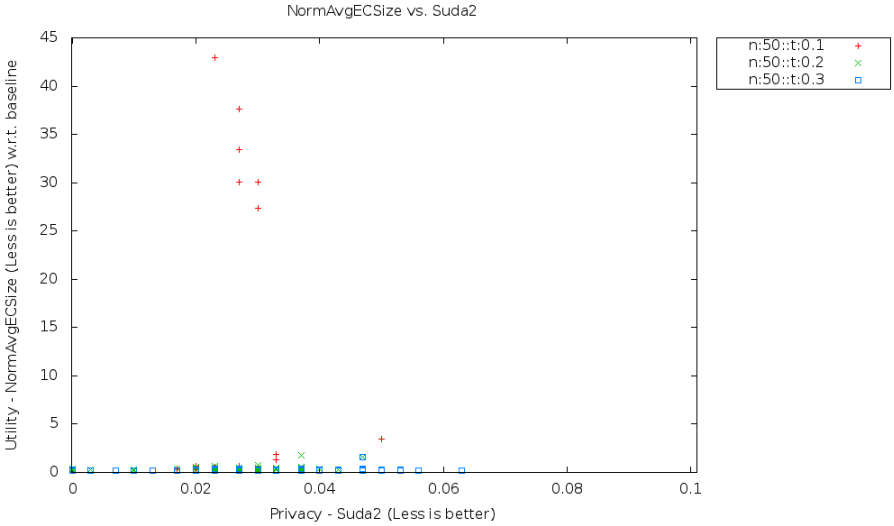
Figure C.15: Mondrian NT: normalized DM (n=150)



(a) QID-4

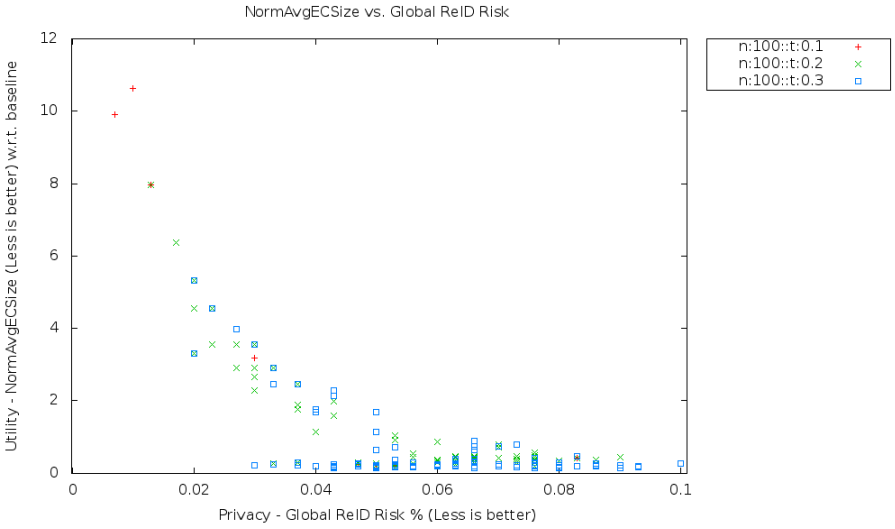


(b) QID-7

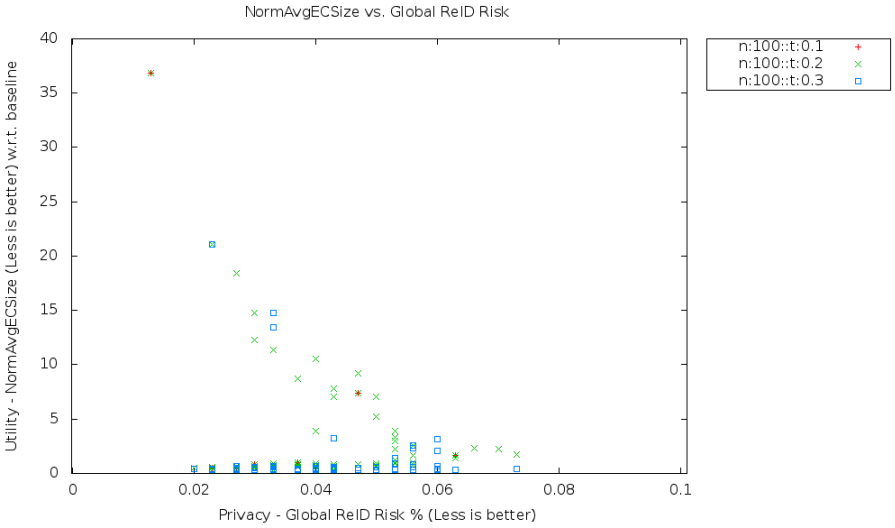


(c) QID-13

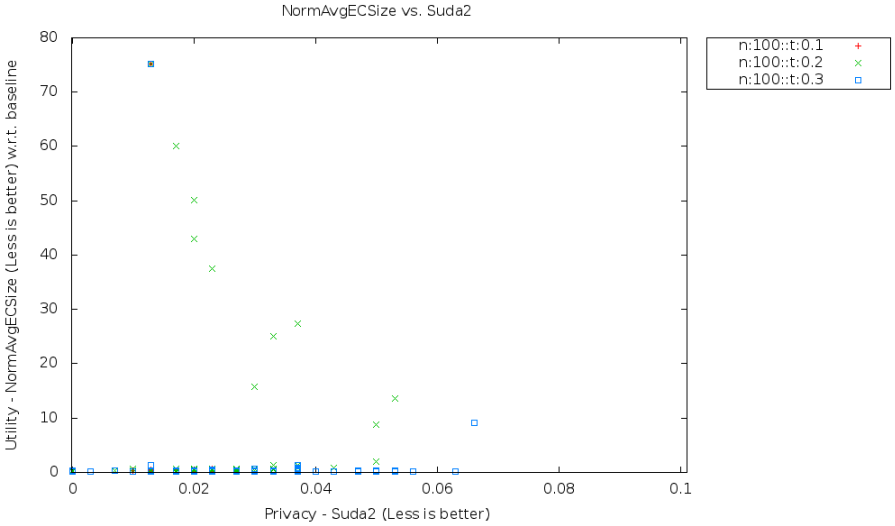
Figure C.16: Mondrian NT: normalized avg. EC size (n=50)



(a) QID-4

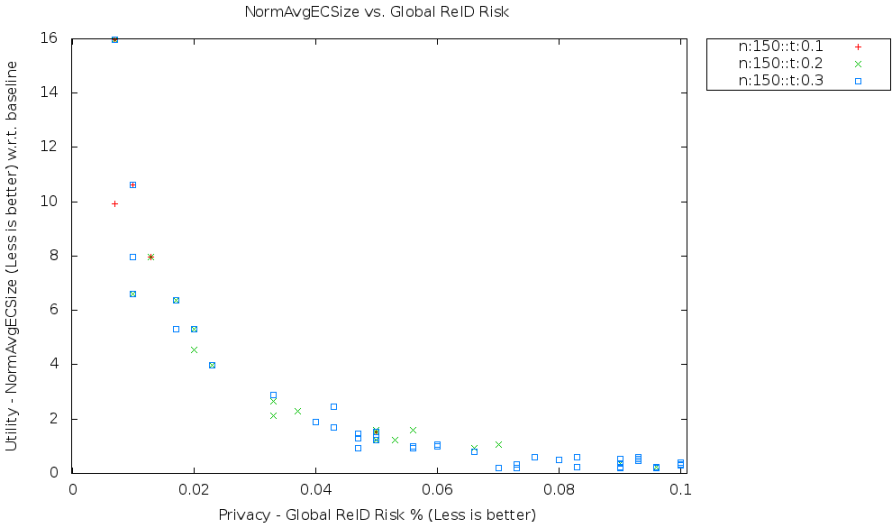


(b) QID-7

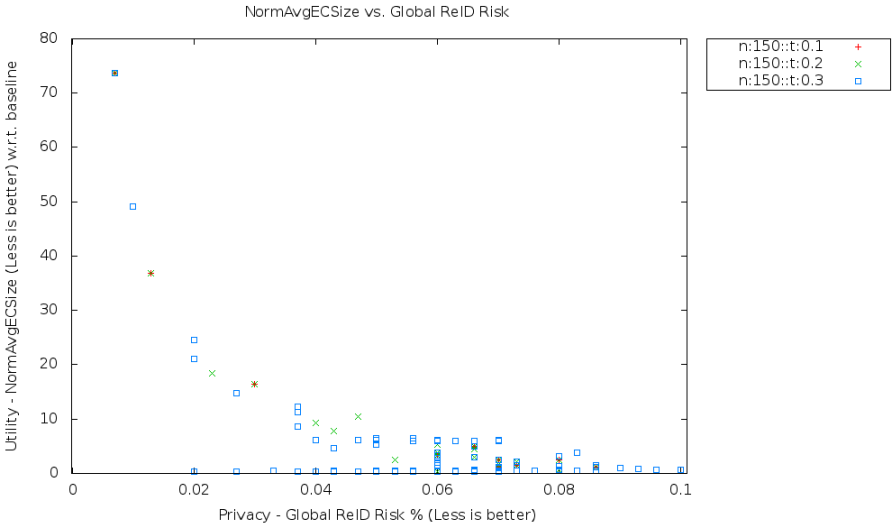


(c) QID-13

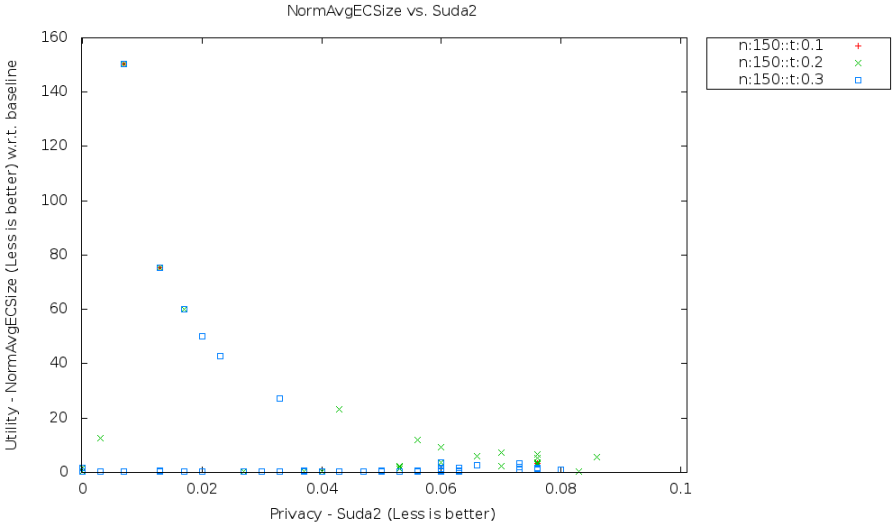
Figure C.17: Mondrian NT: normalized avg. EC size (n=100)



(a) QID-4

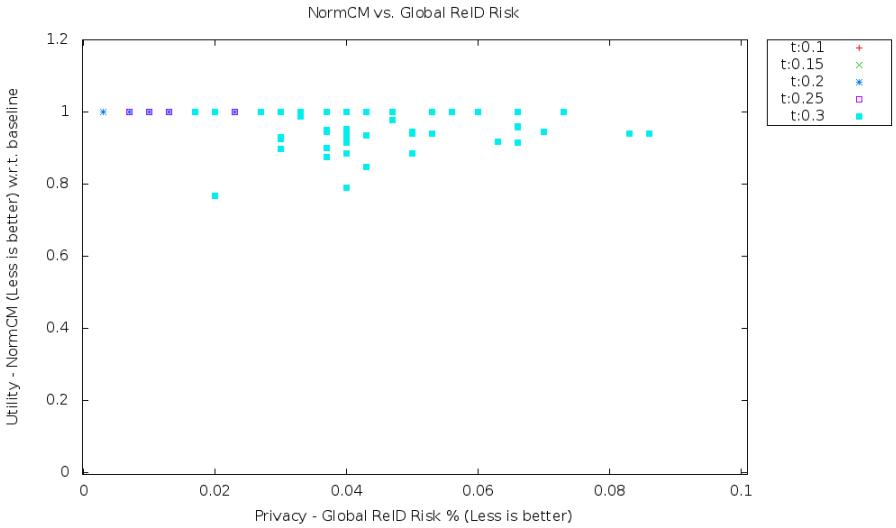


(b) QID-7

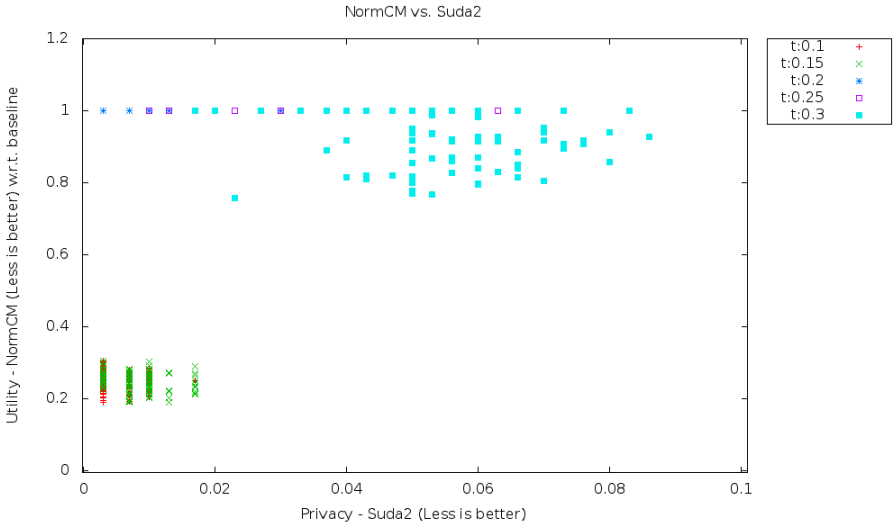


(c) QID-13

Figure C.18: Mondrian NT: normalized avg. EC size (n=150)

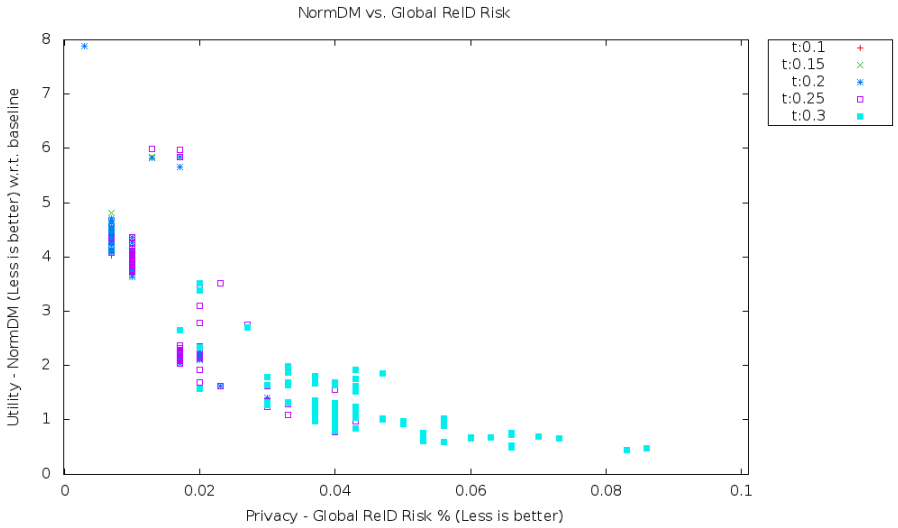


(a) QID-4

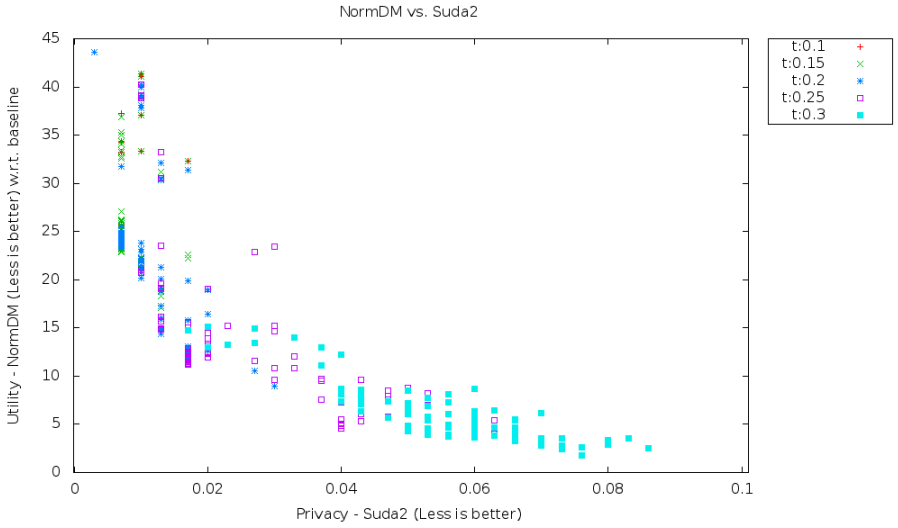


(b) QID-7

Figure C.19: Incognito T: normalized CM

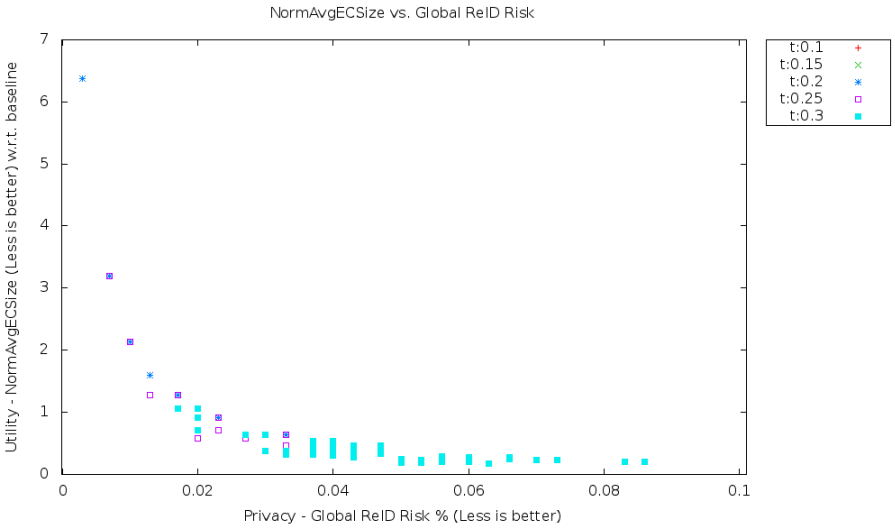


(a) QID-4

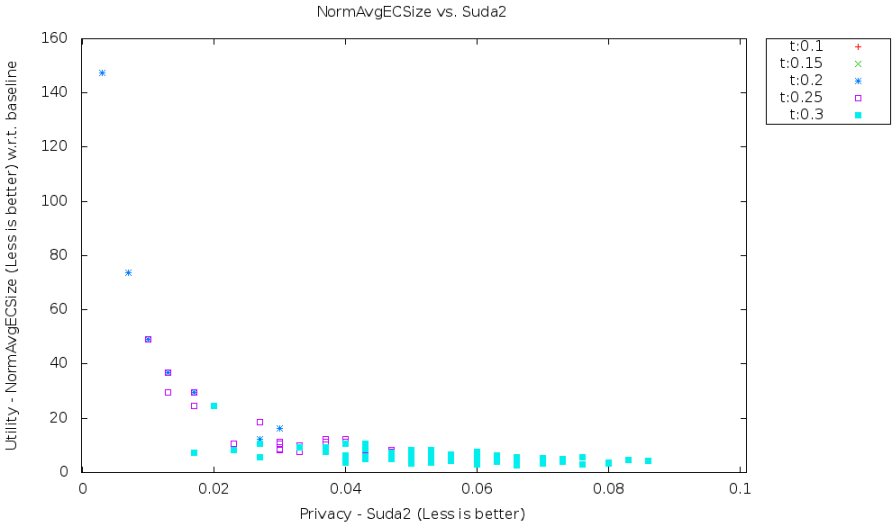


(b) QID-7

Figure C.20: Incognito T: normalized DM

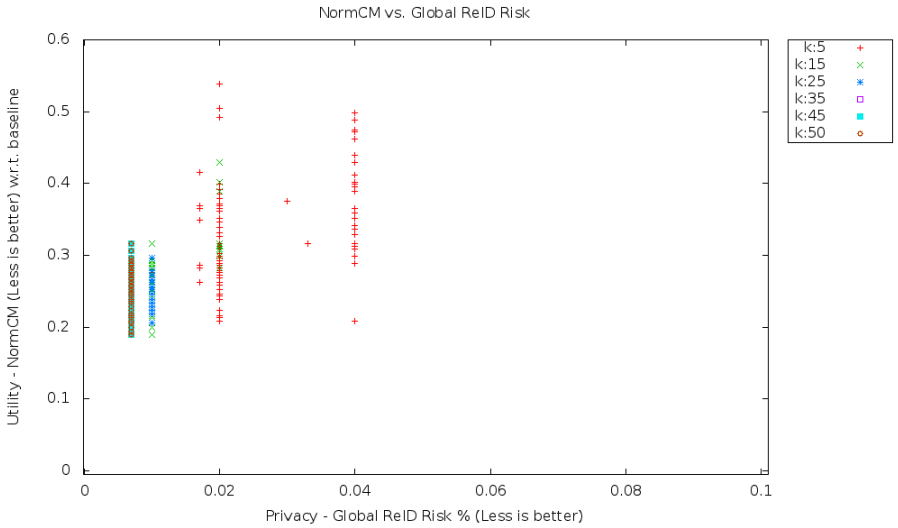


(a) QID-4

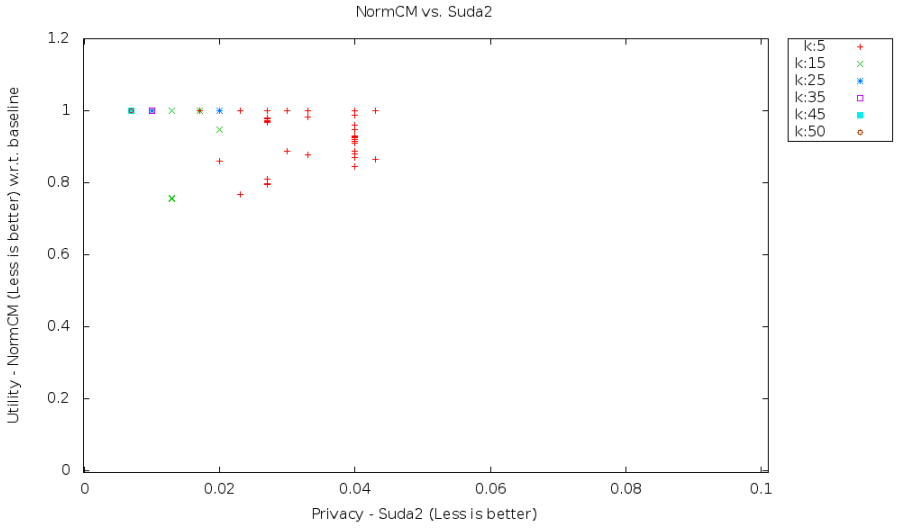


(b) QID-7

Figure C.21: Incognito T: normalized avg. EC size

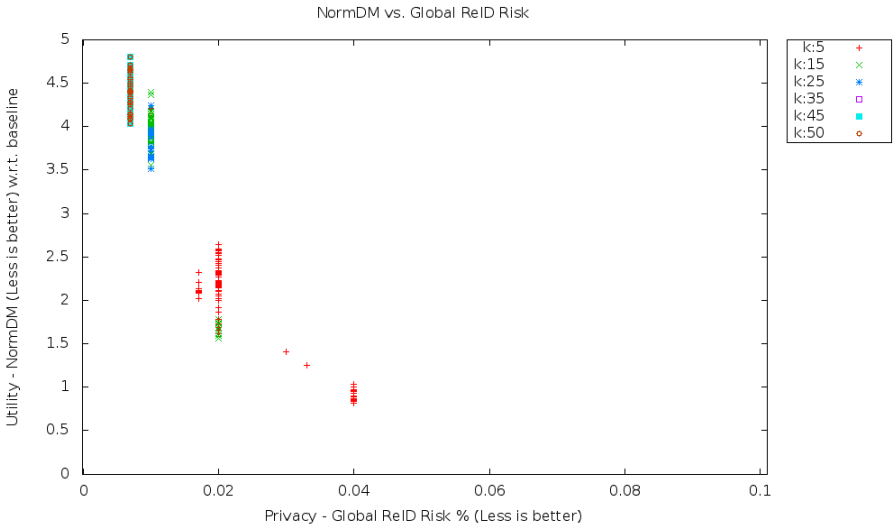


(a) QID-4

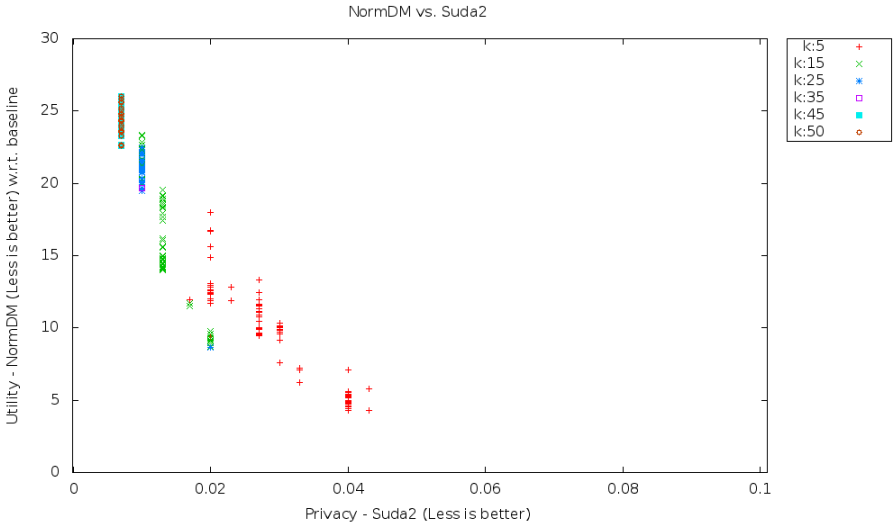


(b) QID-7

Figure C.22: Incognito K: normalized CM

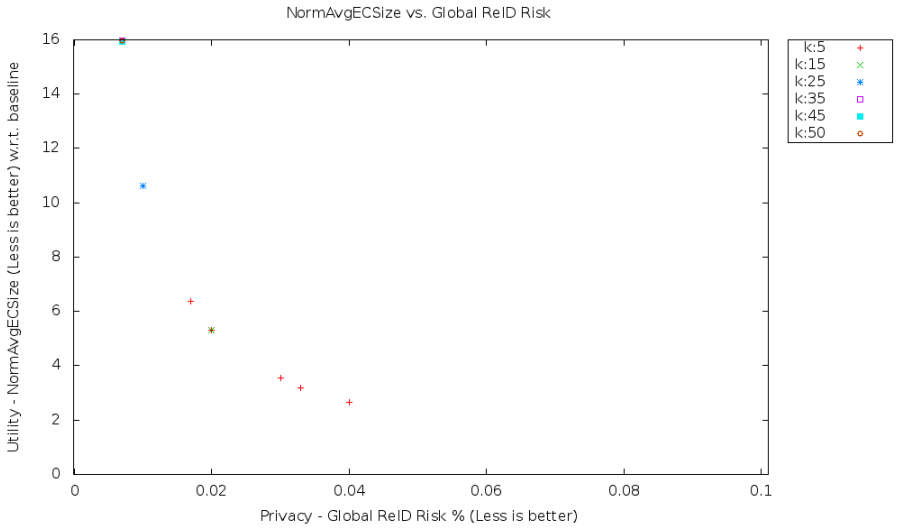


(a) QID-4

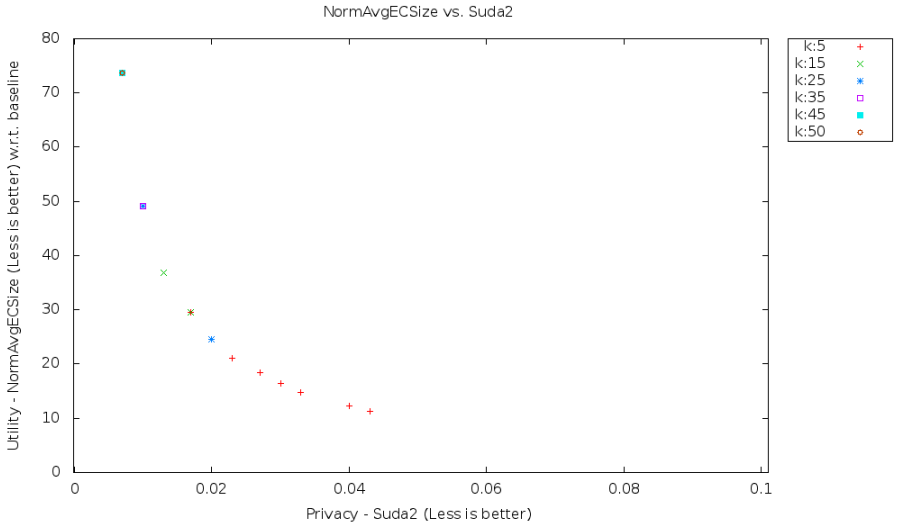


(b) QID-7

Figure C.23: Incognito K: normalized DM

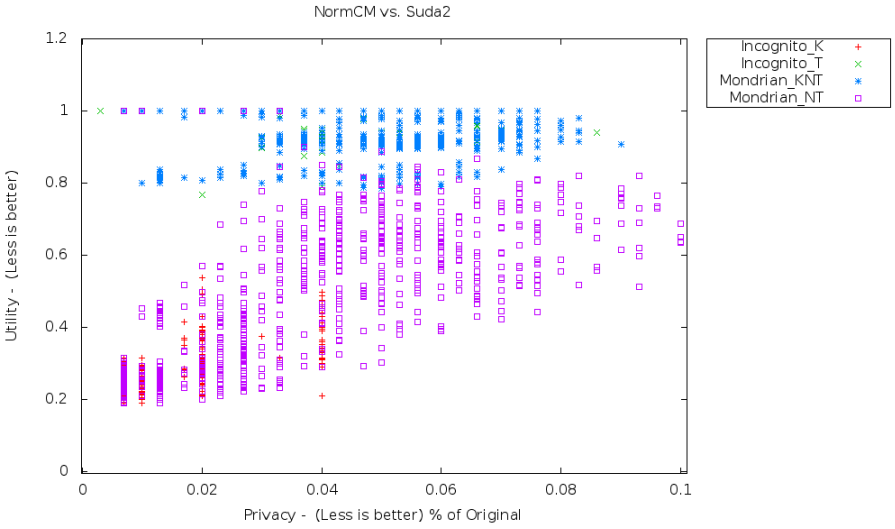


(a) QID-4

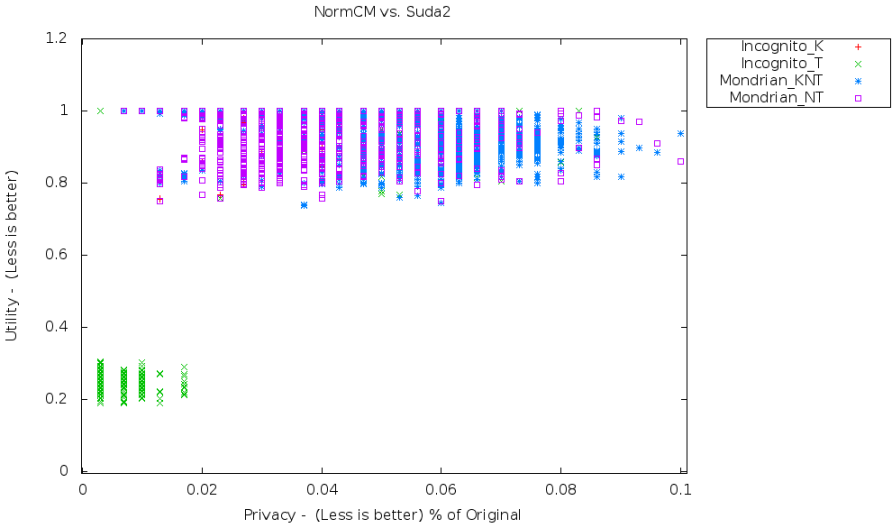


(b) QID-7

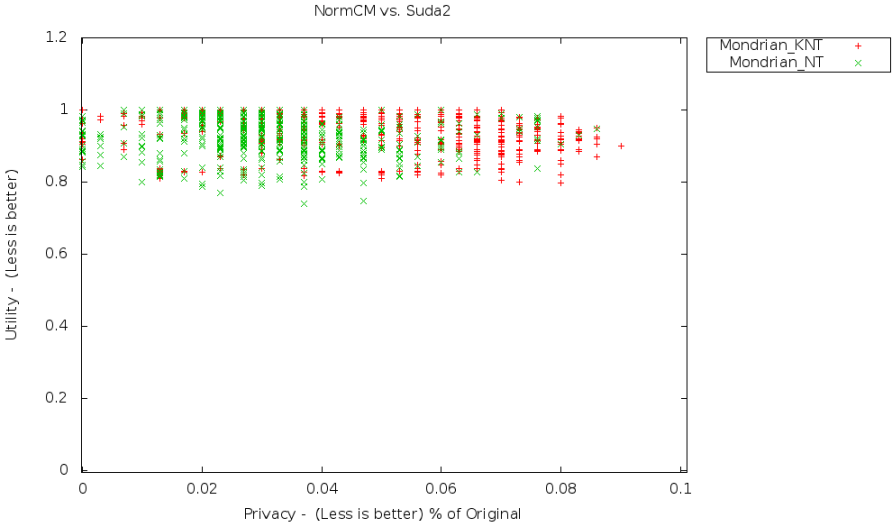
Figure C.24: Incognito K: normalized avg. EC size



(a) QID-4

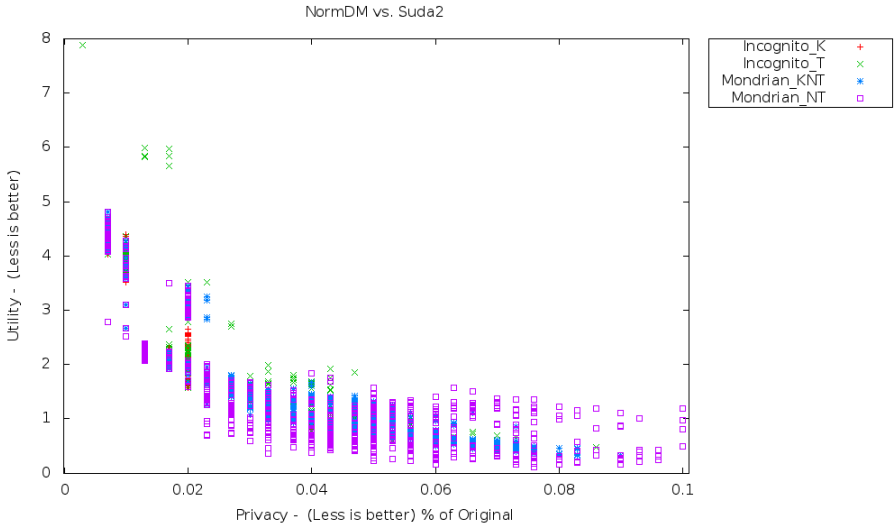


(b) QID-7

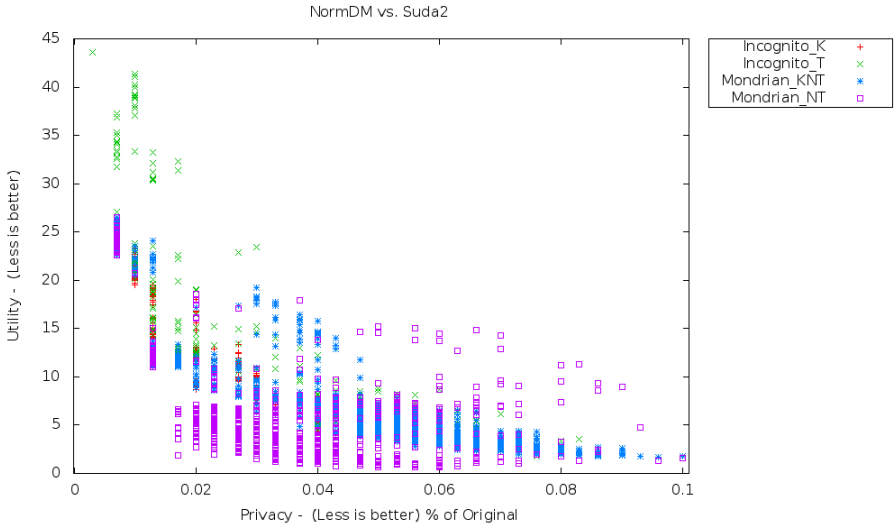


(c) QID-13

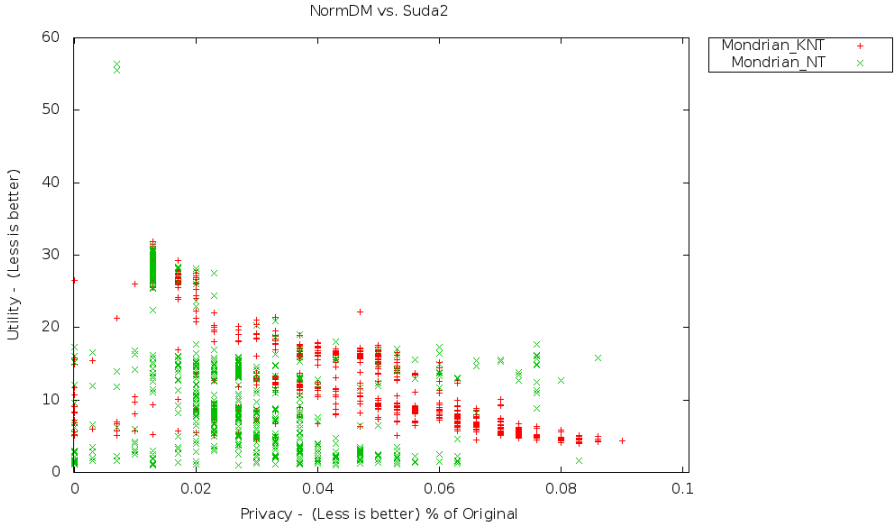
Figure C.25: Normalized CM comparison



(a) QID-4

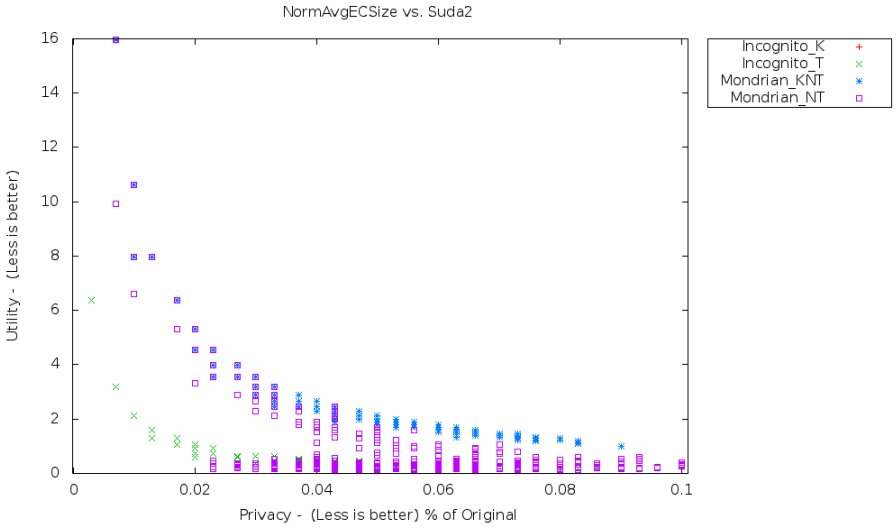


(b) QID-7

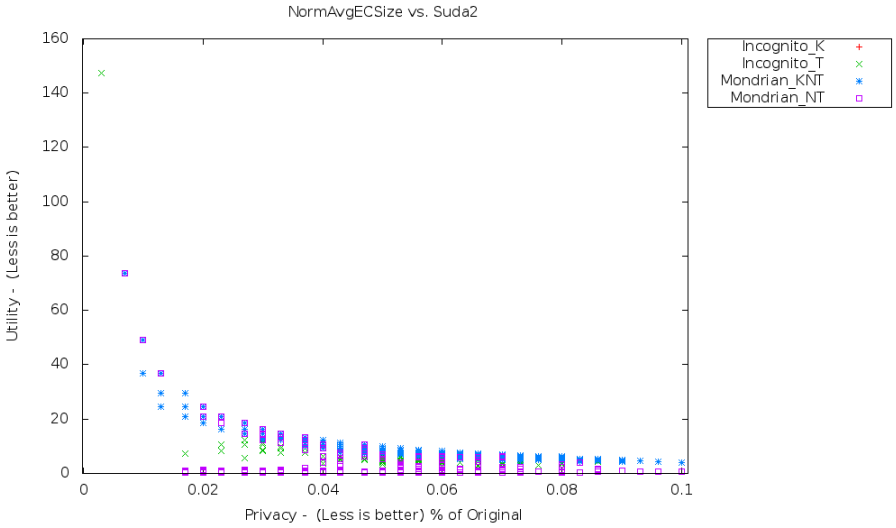


(c) QID-13

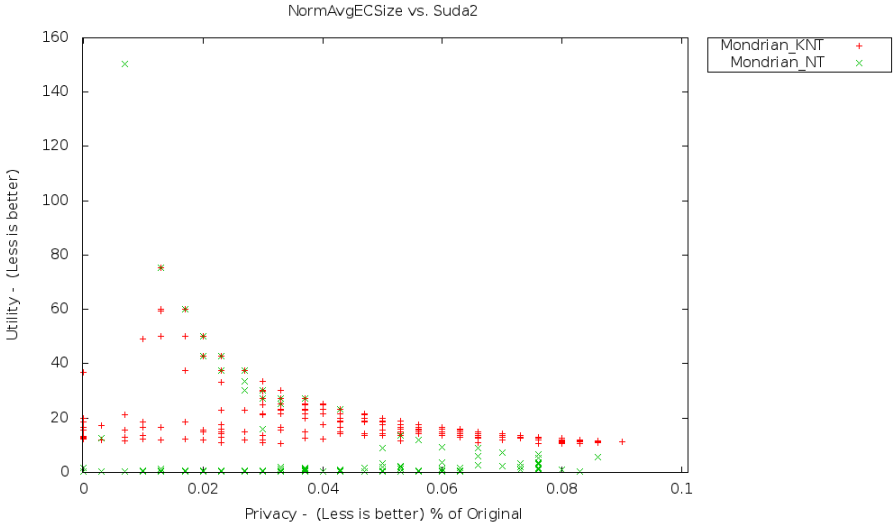
Figure C.26: Normalized DM comparison



(a) QID-4



(b) QID-7



(c) QID-13

Figure C.27: Normalized Avg. EC Size comparison