DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Design and Implementation of a Data Utility Analysis Tool to Optimize the Application of De-Identification Techniques

Bhawna Saini

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

Design and Implementation of a Data Utility Analysis Tool to Optimize the Application of De-Identification Techniques

Entwurf und Implementierung eines Data Utility Analysetools zur Optimierung der Anwendung von Entidentifizierungstechniken

Author:	Bhawna Saini
Supervisor:	Prof. Dr. Florian Matthes
Advisor:	Gonzalo Munilla Garrido
Submission Date:	15.07.2021

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.07.2021 Bhawna Saini

Acknowledgments

First and foremost, I would like to thank my advisor Gonzalo Munilla Garrido, who has been a great support and a motivating force throughout the thesis. Without your guidance, constructive feedback, and friendly encouragements, this thesis wouldn't have been possible. Thank you so much for making my master's thesis journey extremely meaningful and rewarding.

Next, I would like to express my gratitude to Professor Dr. Florian Matthes for providing me with the opportunity to write this thesis at his chair for Software Engineering for Business Information Systems (SEBIS). His valuable feedback not only helped to define the thesis scope but brought professional insights that could often be overlooked.

I would also like to thank Emil Djerekarov for his immense support in the development of the application as well as for introducing me to the world of AWS. Your constant engagement and feedback made me a better software developer. Thank you so much for taking out time from your busy schedule and helping me with every obstacle I faced in the development of this application. Also, I would like to thank Rathje Ann Christin and Dr. Andre Luckow, for providing me the opportunity to pursue this thesis in their department.

Finally, I would like to thank my parents Shashi Saini and Krishan Saini, and my brother Archit, for providing me with constant love, care, and mental support from miles away in India. Last but not the least, thank you Christian for being the best support system I could have ever gotten and also for tolerating me.

This thesis journey was one of the most intense, highly engaging, and a beautiful experience, I have had in my life. It was filled with immense learning and personal growth. I am extremely grateful for doing this thesis and finishing my master's at the Technical University of Munich, one of the most renowned universities in the world. Thank you for being such a friendly university and being my home away from home!

Abstract

The current technological shift and the adoption of technologies by the masses have led to the enormous generation of data. This data, often consisting of personal information, is highly valuable to data-driven organizations to develop personalized products and services for the customers. However, the collection and processing of this data can only be done in accordance with privacy regulations around the world, resulting in a trade-off between ensuring user privacy and utilizing data to its full potential. Therefore, organizations resort to data de-identification, a privacy-enhancing process that maintains user privacy while at the same time preserves data utility.

Even though the data de-identification process enhances privacy, the application of de-identification techniques has a direct impact on data utility due to the resulting information loss. Data utility metrics provide an overview of this information loss by measuring the change in data utility. Thus, helping to understand the effects of de-identifications techniques on a dataset. In this thesis, we propose that to effectively and optimally de-identify the data, the data utility analysis process should be combined with the data de-identification process. Doing so would result in the adoption of a better de-identification strategy for the dataset.

The thesis tests this claim in the context of an automotive enterprise with the aim of enhancing its existing de-identification process. We develop a data utility analysis tool that allows the user to de-identify the data and then further assess its utility through various utility metrics. To implement this tool, various de-identification techniques and utility metrics are explored. Additionally, interviews are conducted with privacy experts at the industry partner to understand the process of de-identification and to derive the technical requirements for this tool in regards to a large enterprise. Finally, we evaluate the effectiveness of this tool by testing it with a real automotive dataset as well as with the privacy experts. From the feedback, we address the potential use cases of such a tool, the future enhancements, and the limitations.

Contents

1. Introduction

1.1. Motivation

The rapid development of technologies and their inclusion in our lives through social media, e-commerce, smart devices, etc., has led to the generation of enormous amounts of data. These data is of great value to organizations as they collect and analyze them to develop personalized products, services, and develop their business strategy. However, with collection of data comes the risks of data privacy. Various privacy legislations around the world advocate for consumer data protection and privacy. A few commonly known ones include the General Data Protection Regulation (GDPR) in Europe [1], the California Consumer Privacy Act (CCPA) [2], and the Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada [3]. These laws regulate the storage, processing, and the use of personal information and are designed to give people more control over their data. Organizations not only have to make sure that the data is collected and stored under strict and fair conditions as per user's consent but also protect it from leakage and misuse. Failure to do so can lead to hefty fines and penalties to the infringing organization. Thus, it is often difficult for organizations to utilize data to its maximum potential while at the same time maintaining user privacy and compliance with privacy laws [4].

Organizations therefore often use de-identification. De-Identification is a privacy enhancing process that enables to maintain data utility during analysis while at the same time preserving the privacy of the individuals to a certain extent[5]. De-identification removes, conceals, or replaces the personal identifiers of a dataset to prevent one's identity from being disclosed [6].

As mentioned by Garfinkel [7] and the ISO/IEC 20889:2018 [8], de-identification is not a single technique to de-identify data but *a "[...] collection of approaches, algorithms, and tools that can be applied to different kinds of data with differing levels of effectiveness"* [9]. Not only do these de-identification techniques enhance privacy but also result in information loss in the dataset. So, how does one decide which set of de-identification methods is appropriate for a given dataset in a way that the data utility is not significantly impacted? The goal of this thesis is to assist organizations in the automotive sector to select an appropriate set of de-identification techniques and understand the technical implication of the application of these techniques on the resulting de-identified dataset.

To help accomplish this goal, the thesis aims to design and implement a data utility

analysis tool that would help privacy experts to quantify the performance and optimize the application of de-identification techniques on datasets.

1.2. Problem Statement

The current shift in technologies has brought massive revolution in the way automobiles function. In-vehicle connectivity, that allows cars to communicate bidirectionally with other systems outside of the car by making use of internet, has transformed vehicles into more than just a traditional means of transportation [10]. Not only vehicles, but the drivers and passengers are becoming more and more connected. This move of connected cars into mainstreams have caused vehicles into becoming massive data hubs [10]. According to recent estimates, these cars could produce up to 25 gigabytes of data every hour, most of which is data about individuals, such as the driver of the vehicle or its passengers [11]. Since this data could directly or indirectly be linked to the individuals, organizations must therefore anonymise the data before they could make use of it in order to preserve the privacy of individuals and to adhere to privacy regulations around the world [12]. One of the ways to achieve this anonymization is through using data de-identification techniques that help to preserve the data utility to some extent [9]. As the choice of data de-identification techniques has a direct impact on the utility of data, therefore efforts should be made in understanding the peculiarities of application of different de-identification methods to structured textual/numerical data [13]. In his research [13] Tomashchuck mentions, since de-identification is usually a manual process and it is highly depended on the expertise of de-identification experts, not just systematization of methods is needed, but also automation of de-identification processes is desirable.

In order to achieve this, we propose that the data utility analysis process should be combined with the data de-identification process. Doing so would not only help privacy experts to understand the implication of their de-identification approach but at the same time come up with a optimal data de-identification strategy that could yield greater data utility.

In this thesis, we develop a data utility analysis tool as a proof-of-concept. This tool would help user to de-identify the data and then further assess the data utility through various utility metrics.

1.3. Research Questions

The goal of this thesis is to design and implement a data utility analysis tool that would help privacy experts to quantify the performance and optimise the application of de-identification techniques on datasets. Through the development of this tool, we aim to answer the following research questions.

RQ 1. What is the state-of-the-art of data utility metrics and data de-identification tools?

This research question will set the theoretical foundation for our data utility analysis tool. The objective of this research question is to identify the current state of research in utility metrics in the context of data privacy and discover widely used data de-identification tools in industries by conducting an extensive literature review.

RQ 2. How could the implementation of an enterprise level data utility analysis tool look like?

As a part of this research question, we develop a data utility analysis application as a proof of concept with our company partner. The answer to the research question comprise of our design and implementation choices made during the development of this application. In addition, it provides an overview of challenges and limitations faced during the development process.

RQ 3. Given the feedback during the application demonstration, in what ways could the tool be improved?

The effectiveness of this application will be evaluated by testing it on real datasets provided by the industry partner and with experts in software and privacy domain. Based on the useful insights and feedback provided during the demonstration, we derive a comprehensive list of ways in which the data utility analysis tool could be enhanced.

1.4. Thesis Structure

This section provides an overview of the structure of the thesis, explaining briefly the content of each chapter.

Chapter 2: Background provides the theoretical foundations of this master thesis. Concepts relating to the de-identification process and utility analysis are discussed.

Chapter 3: Related Work presents the scientific work and literature in the domain of data de-identification and utility analysis.

Chapter 4: Research Approach discusses the research methodology adopted to accomplish the goal of this thesis.

Chapter 5: Application of Data De-identification at an Automotive Enterprise discusses the process of data de-identification in the context of a large automotive industry. The chapter provides the overview of the entire process and presents the associated limitations with it. The chapter sets the basis for the proposed solution.

Chapter 6: Design of Data Utility Analysis Tool explains the proposed solution to address the limitations discovered in chapter 5.

Chapter 7: Implementation of Data Utility Analysis Tool explains the development of the proposed solution. The chapter explains the requirements and presents various software architectures of the application. Additionally, the technology stack adopted and the technical architecture are discussed as well.

Chapter 8: Evaluation of Data Utility Analysis Tool presents the results of the evaluation of the proposed solution against various criteria. In addition, the insights and feedback received from the experts in privacy domain are discussed.

Chapter 9: Conclusion summarizes the results of this thesis including the limitations of the proposed solution and the associated future work.

2. Background

This section sets the theoretical foundation of the thesis. Since de-identification is carried out based on the attribute's value type and identifier type, we first present the classification of the attributes. Then we briefly discuss the concept of de-identification followed by the popular approaches of de-identification. Then we present the de-identification techniques and finally the various adopted data utility metrics.

2.1. Classification of Dataset Attributes

In this thesis, we focus on the de-identification of tabular data consisting of rows (records) and columns (attributes). In the context of dataset, attribute defines a characteristic of any selected record also called entity [14]. It is essential to understand the properties of data before the application of de-identification techniques on it. Therefore, in this section, we discuss the classification of attribute based on its value type and the identifier type.

2.1.1. Types of attribute values

An attribute value is of either categorical or numerical (also referred as qualitative and quantitative) type [15]. Within these broad classifications exists further sub-classifications that are discussed below.

- *Categorical:* Categorical data is the data that could be allocated to possible categories [15].
  - *Nominal:* In case of nominal type, values are assigned to categories that have no natural ordering or any implicit rank [16]. Examples include, blood type(A,B,AB,O), gender types etc.
  - *Ordinal:* In ordinal data, the categories have a natural ordering and can be ranked [16]. For examples, the clothes sizing(S, M, L, XL), patients may classify their degree of pain on a scale of 0-5 etc.
- *Numerical:* Numerical data type is classified into discrete or continuous type [15]. Further on the basis of scale, a numerical data type could have interval scale or the ratio scale. This means that any numerical attribute whether continuous or discrete, can have a scale of either interval or ratio type.

- *Discrete:* Attribute values of discrete data types are represented as integers or whole numbers. They can be counted, are fixed in value and the in-between measures do not exist. An attribute of discreet type can take only limited values and not be further subdivided into smaller parts [16]. For example, a die can take values between 1-6 and one cannot get 3.2 on it. Other examples could be, the number of holidays in a year, number of workers in a company etc.

- *Continuous:* Continuous attribute values can take any numerical values and are represented as fractions. These value can be divided into smaller parts and have a sense of measure [16]. For example, the speed of a car or the height of a person can be measured on a precise scale.

- *Interval:* Data with interval scale can be both continuous or discrete, is ordered and the values are equally distant (there intervals are equal). The interval data can hold negative values and has no true zero [17]. For example, temperatures in Celsius and Fahrenheit are of interval type and their values can fall below 0. There exists no sense of ratio between interval values. One can count, order, add or subtract the interval data but not multiply or divide [18].

- *Ratio:* Like interval data, data with ratio type could be both continuous or discrete, is ordered and is equidistant. However, ratio scale has a meaningful zero and the data cannot hold negative values [17]. For example, the age, height, weight, distance is always zero and above. The ratio data can be added, subtracted, ordered, compared, multiplied or divided [18].

2.1.2. Identifier types

An identifier in a data set refers to set of attributes that can uniquely identify the data subject in the data set [8]. This data subject also called data principal could be an individual, a company or even a software application [8]. Based on identifiers type, attributes can be categorized into four categories as follows:

- *Directly Identifying Attributes:* Direct identifiers can uniquely identify the data principal either by themselves or in combination of other available attributes in the data set [19]. For example, a name attribute in the data set when combined with telephone number and email address attribute can lead to unique identification of the individual. Generally, the direct identifier is not relevant in the data analysis and if it is, it should be labeled as quasi-identifier [20].

- *Indirectly Identifying Attributes (Quasi-Identifiers):* An attribute is a quasi identifier if it satisfies the following three conditions: [21]

  - the attacker has background knowledge in regards to that attribute,

  - it can be used both individually or in combination to other attributes to re-identify the individuals
  - the attribute will be used for data analysis purposes.

  An attacker can gather background knowledge in multiple ways such as by having a relation to the targeted individual (neighbour, friend etc.), through other publicly disclosed data sets, social media platforms etc. Examples of quasi-identifiers are zip-codes, ethnicity, data of birth, gender, profession, locations etc.

- *Sensitive Attributes:* Sensitive attributes are those that comprise of sensitive data concerning the data subject. These attributes themselves do not lead to the re-identification of the individual but must be protected against any disclosure. An individuals health related diagnose is an example of sensitive attribute [19].
- *Other Attributes:* Attributes that do not belong to above mentioned types, belong to this category. Such an attribute are neither sensitive in nature, nor contributes in re-identification of data subject [19]. For example in a taxi data set, number of passengers or the mode of payment are of this type.

At times, it is difficult to determine whether an attribute is an indirect or direct identifier. Figure 2.1 depicts a simple rule based approach to classify the attributes.

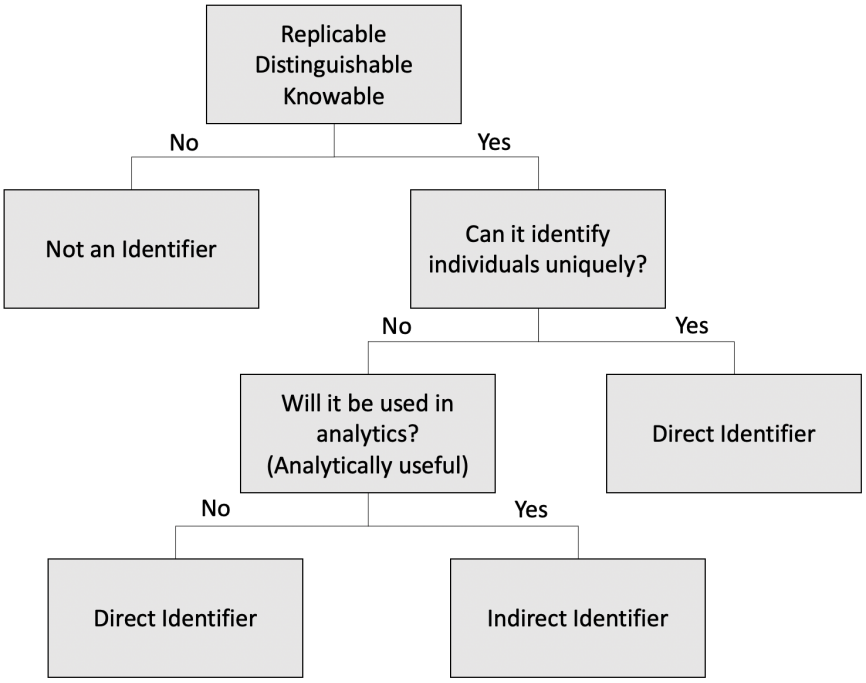


Figure 2.1.: Rules based classification between direct and indirect identifiers [21] (Figure 3)

2.2. Data De-identification

De-identification is a privacy-enhancing technique that protects the privacy of individuals by removing personally-identifying information (PII) from a record or a data set. As defined in [8], de-identification refers to a process of removing the association between a set of *identifying attributes* and the *data principal*. The term data principal, also called data subject, refers to whom the data set relates. The data principal could either be a person, company, organization, or software application [8]. An identifying attribute is an attribute in a data set that could either directly or indirectly contribute in the re-identification of the data principal. If a data set does not contain personal information, its use or disclosure cannot violate the privacy of individuals [20]. Organizations, therefore, often rely on the de-identification of the data to comply with the privacy laws [9]. It must be noted that de-identification does not completely eradicate the risk of re-identification but reduces it a to minimum in a data set if carried out effectively [20].

Often the terms *data anonymization* and *pseudoanonymization* are used interchangeably with *data de-identification*. In literature, there lacks a consistent definition and differentiation between these terms which makes their understanding and application at times confusing [9]. Since these terms would often be used in this thesis, it is therefore, important to establish a clear distinction among the terms. We stick to the combination of approaches of [7] and [13] to describe these terms as following:

- **Data de-identification:** De-identification refers to any process that removes the association between the data subject and a set of identifying attributes. It is a concept of higher-level that covers both *data anonymization* and *pseudonymization*.
- **Data anonymization:** Data anonymization is considered as a subset of de-identification methods which involves the condition of irreversibility. This means, that any data set after the de-identification process can by no means re-identified. This is very well reflected in the definition by ISO 25237/2017 [22], *"The process by which personal data is irreversibly altered in such a way that a data subject can no longer be identified directly or indirectly, either by the data controller alone or in collaboration with any other party"*.
- **Pseudonymization:** Pseudonymization is a type of de-identification technique where identifiers of the data principal are replaced by pseudonyms to hide the identity of the data principal [8].

2.3. Approaches to Data De-identification

The Health Insurance Portability and Accountability Act (HIPAA) [23] of 1996 that provides standards for the secondary use and disclosure of personal health information describes two key approaches for the de-identification of health data: *Safe Harbor Method* and *Expert Determination Method* [24]. Both of these approaches have been referred by

many privacy regulations around the world and sets a good foundation for deriving guidelines for effective de-identification.

2.3.1. Safe Harbor Method

As per the safe harbor method, for the data to be legally de-identified, the following two requirements should be met:

1. From the dataset, 18 types of identifiers related to the individual or of family members should be removed. These identifiers are provided in a form of a list stated in [24].
2. The entity with whom the data will be shared, should not have any external knowledge that when combined with the de-identified dataset could lead to re-identification of individual in the dataset. [24]

2.3.2. Expert Determination Method

Expert determination method, sometimes referred to as the statistical method appoints the use of an expert to perform the de-identification process by applying statistical and scientific principles. According to this method:

1. The de-identification should be carried out by the expert in a way that the risk of individual's identity being revealed is low both with or without the combination of any external information. [21]
2. The de-identification strategy, the reasoning behind the entire process as well as the results must be documented [21].

Expert determination method meets one of the major limitations of safe harbor method, that is, the strict removal of identifiers from the dataset in order to be considered de-identified. Such a removal could result in critical information loss and may render resulting dataset useless in some cases. Through the flexibility offered by expert determination method, datasets could be de-identified in a way that allows to preserve the utility of the data. Figure 2.2 summarizes these two approaches.

2.4. Data De-identification Techniques

The de-identification process is implemented through a range of de-identification techniques. The transformative nature of these techniques helps to meet the privacy requirements, but at the same time cause information loss in the data set [13]. Bondel et al. [9] performed extensive literature review on de-identification techniques and clustered them into *perturbative* and *non-perturbative* methods. Non-perturbative methods do not affect the truthfulness of the data but only reduce the accuracy. Whereas perturbative

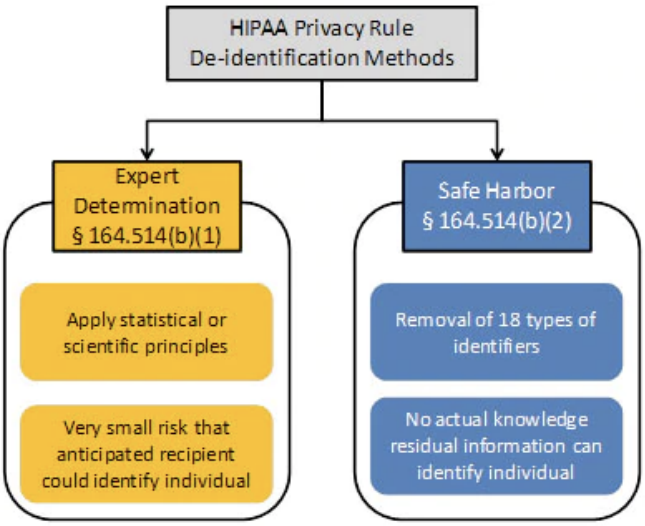


Figure 2.2.: De-identification Approaches in HIPPA Privacy Rule [24](Figure 1)

methods distorts the data but preserve the statistical properties. Their classification is depicted in Figure 2.3. Besides the perturbation category, there are four other sub categories: *Data Type Independent*, *Numerical*, *Deletion*, *Generalisation*.

- Data Type Independent de-identification techniques can be applied to any data irrespective of the data type.
- Numerical based de-identification techniques is only applicable to data of numerical types.
- Deletion refers to de-identification techniques that causes the removal of data.
- Generalisation based de-identification techniques reduces the accuracy of the data.

In the following subsections, we describe each of the de-identification techniques briefly with an example.

Sampling

In sampling, a smaller subset of data set(sample) is randomly selected from the larger data set. This creates uncertainty about the data principal being in the data set . Also, performing other de-identification techniques on a sample rather than the whole data set significantly reduces the risk of re-identification attacks [8]. Table 2.1 depicts an example of sampling where two out of four records are selected as sample.

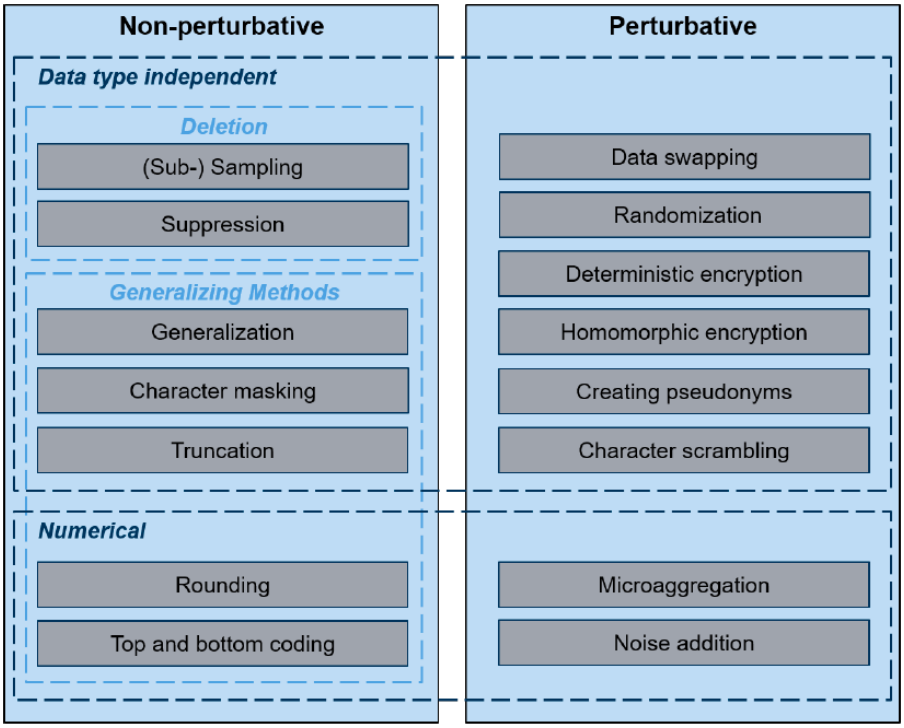


Figure 2.3.: Classification of de-identification methods [9](Figure 1)

Country	Cars-Sold
India	6500
China	8200
Germany	5300
US	9800

(a) Original Dataset

Country	Cars-Sold
Germany	5300
US	9800

(b) After sampling

Table 2.1.: Sampling

Aggregation

Data aggregation refers to the process of consolidating of information in a data set based on statistical functions and providing it in a summarised manner [8]. Within a data set, aggregation could be applied to a group of related attributes or values in an attribute.

- *Aggregation on group of related attributes:* In case of related attributes, aggregation is performed on the basis of a common branch of hierarchy or statistical functions. The end results represent combined values of related attributes in any individual record. In Table 2.2 (b) related attributes 'Q1', 'Q2', 'Q3', 'Q4' got combined together based on average function.
- *Aggregation on values of an attribute:* When performing it on the values of an attribute, the end result is a value that represent all records in the original data set. In Table 2.2 (c) the values of attributes 'Q1', 'Q2', 'Q3', 'Q4' are aggregated based on average function.

Year	Q1	Q2	Q3	Q4
2017	800	776	457	328
2018	676	567	678	411
2019	989	898	454	554

(a) Original dataset

Year	Avg-Sales
2017	590.25
2018	583
2019	723.75

(b) Related attributes

	Q1	Q2	Q3	Q4
Avg	821.6	747	529.6	431

(c) Values of attribute

Table 2.2.: Aggregation

Microaggregation

In microaggregation records are clustered into groups of atleast size 'k' such that the groups have similar values for a set of selected attributes. Then, the new values of each of these selected attributes is replaced by the aggregated value of that attribute's values in the group [25]. For example, in Table 2.3, the records of the vehicle data set are clustered into groups of size 3. The values of the attribute 'Km-Driven' is aggregated based on average function.

Suppression

Suppression refers to deletion of attribute values. It could be performed either for all records (global suppression) or for selected records (local suppression) [8]. Suppression

Car Model	Km-Driven	Selling Price
Twingo	30120	4500
A3	65221	11000
A170	32111	15000
Fiesta	72989	8000
320i	67887	13000
C200	29098	6000

(a) Original datatset

Car Model	Km-Driven	Selling Price
Twingo	30443	4500
A170	30443	15000
C200	30443	6000
A3	68699	11000
Fiesta	68699	8000
320i	68699	13000

(b) After microaggregation

Table 2.3.: Microaggregation

does not distort the data but causes loss in information. In Table 2.4, 'VehicleNumber' attribute is globally suppressed and the individual values of attribute 'Lat' and 'Long' are locally suppressed.

VehicleNumber	TripId	TimeStamp	Lat	Long
MGN212	tid11	2018-02-14 04:29:56	3.241	4.573
MGN212	tid11	2018-02-14 04:35:00	3.245	4.578
FG670	tid22	2018-02-14 17:12:00	7.241	8.444

(a) Original data set

TripId	TimeStamp	Lat	Long
tid11	2018-02-14 04:29:56	3.241	4.573
tid11	2018-02-14 04:35:00		
tid22	2018-02-14 17:12:00	7.241	8.444

(b) After local and global suppression

Table 2.4.: Suppression

Character Masking

Character masking is a simple technique where a number of characters in an attribute value is replaced by a special character from either direction [8]. This technique does not cause distortion in the data. In Table 2.5 (b), the attribute 'VehicleNumber' is masked. The special character is 'X', the number of characters masked is 3 and the masking direction is right.

Truncation

In truncation, a number of characters in an attribute value is removed from either direction [8]. This technique does not causes distortion in the data. In Table 2.5 (b), the attribute 'TimeStamp' is truncated. The number of characters truncated are 3 and the truncation direction is right.

Rounding

In rounding, the values of a numerical attribute is rounded either up and down depending on the rounding base [8]. Rounding only reduces the accuracy of the data and does not distort the data. In Table 2.5 (b), the attributes 'Lat' and 'Long' are rounded to base 1.

VehicleNumber	TimeStamp	Lat	Long
MGN212	2018-02-14 04:29:56	3.241	4.573
MGN212	2018-02-14 04:35:00	3.245	4.578
FG670	2018-02-14 17:12:00	7.241	8.444

(a) Original datatset

VehicleNumber	TimeStamp	Lat	Long
MGNXXX	2018-02-14	3.2	4.6
MGNXXX	2018-02-14	3.2	4.6
FGXXX	2018-02-14	7.2	8.4

(b) After masking, truncation and rounding

Table 2.5.: Character Masking, Truncation, Rounding

Generalisation

Generalisation is a type of de-identification technique where the attribute values are replaced with more general value based on a generalisation hierarchy [8]. This reduces the number of distinct values in an attribute as general value is shared by many records.

Generalisation does not distort the data but only reduces its granularity. Generalisation can be both local or global. In the case of global, all the values in an attribute are generalised based on the generalisation criteria. Whereas in local, only specific values of attributes in selected records are generalised. In Table 2.6 (b), attributes 'KM driven' and 'City' are generalised. The 'KM driven' attribute is an example of integer generalisation with interval size 1000.

Top/bottom coding

The top/bottom coding technique sets thresholds on the largest and smallest values that an attribute can take. When an attribute value exceeds these thresholds, the value is replaced by the threshold value [8]. In Table 2.6 (b), the attribute 'Selling Price' is top/bottom coded. The values lower than the threshold values of 1000 and 20000 are replaced by the threshold values

Name	KM driven	Selling Price	Place
Corsa	281767	3000	Munich
Punto	801123	800	Brussels
BMW320	5678	25000	Frankfurt

(a) Original datatset

Name	KM driven	Selling Price	Place
Corsa	280000-290000	3000	Germany
Punto	80000-810000	<1000	Belgium
BMW320	5000-6000	>20000	Germany

(b) After generalisation and top/bottom coding

Table 2.6.: Generalisation and top/bottom coding

Randomization

In randomization, an attribute values is replaced by some random value in a way that the format is preserved [8]. This technique distorts the data. The goal is to prevent an attacker from deducing the attribute values through pattern recognition. In Table 2.7, the attribute 'TimeStamp' is randomized.

Noise addition

In noise addition, random values called noise is added to attribute values of continuous data type in such a way that the statistical properties(mean, mode, median, variance

etc.) of the attribute is preserved [8]. In Table 2.7, some random noise is added to the values of the attribute 'Speed'.

Pseudonymization

In pseudonymization, personally identifying attribute values related to a data principal are transformed into artificial identifiers/ pseudonyms and later replaced by these specific pseudonyms [26]. This allows linking of records without revealing the identities of the data principal. In Table 2.7, the values of the attribute 'VehicleNumber' is replaced by the pseudonyms.

Permutation

Permutation, also referred to as data shuffling, involves swapping of attribute values among the records without causing modification of attribute values [8]. This process preserves the statistical properties of the attribute. However permutation should be carried out in a way that the resulting data set is logical and realistic. For example, if permutation is carried out on the attribute 'name', then the shuffling of values should correlate with the 'gender' attribute in the data set. In Table 2.7, the values of attribute 'Lat' is shuffled among all records.

VehicleNumber	TimeStamp	Lat	Long	Speed(km/h)
MGN212	2018-02-14 04:29:56	3.241	4.573	45.3
MGN212	2018-02-14 04:35:00	3.245	4.578	82.7
FG670	2018-02-15 17:12:00	7.241	8.444	105.4

(a) Original datatset

VehicleNumber	TimeStamp	Lat	Long	Speed(km/h)
fg678T	2009-03-14 03:12:00	7.241	4.573	40.3
fg678T	2001-07-14 18:45:00	3.241	4.578	85.7
bh778h	2007-12-15 13:17:23	3.245	8.444	107.4

(b) After randomization, noise addition, permutation and pseudoanonymization

Table 2.7.: Randomization, Noise addition, Permutation and Pseudonymization

Deterministic encryption

Deterministic encryption is a non-randomised symmetric encryption [8]. This means that same attribute values will always produce the same cipher-text if encrypted with the same encryption key. The data truthfulness is not affected. Equality and matching

operations as well as generating frequency distributions could still be performed on the encrypted attribute values. Some of the special types of deterministic encryption are order-preserving encryption and format-preserving encryption.

- *Order-preserving encryption:* This encryption works in a way that it preserves the numerical ordering of the plain-texts after encryption. This mean that the values of the encrypted attribute will have the same logical numerical order. Also, mathematical operations such as min/max queries, count, comparison operations and range queries can still be performed after encryption [27].

- *Format-preserving encryption:* This encryption encrypts in a way that the cipher-text has the same format and length as that of plain-text. Format-preserving encryption is useful in encrypting data in legacy systems that usually have length and storage limitations [28].

The Table 2.8 depicts the above mentioned encryption types.

Encryption	Plain Text	Cipher Text
Deterministic	password	vGHZ7/===67)/6ftFGC7W2
Order Preserving	10, 15, 20	4562, 7829, 9222
Format Preserving	123-ABC-789	876-BHT-245

Table 2.8.: Deterministic Encryption

2.5. Data Utility Metrics

The transformative nature of de-identification techniques cause the loss of information in the datasets [13]. The level of this transformation decides the impact on the data utility. The more the de-identification, the lesser is the data utility. However, data analytics could still be performed on the dataset provided that the de-identification is carried out optimally [19]. To perform the de-identification optimally, it is necessary to quantify the change in utility caused by the de-identification process. Data utility metrics are capable of quantifying the utility losses in de-identified data and for evaluating the performance of the de-identification process [13]. Variety of data utility metrics exists that measures different properties of data, however the choice of the metric is application specific and on the data type being anonymized [29]. In the following sections we discuss various of these utility metrics. At first, we describe the metrics that are implemented in the data utility tool and then the other utility metrics that are mathematically advanced or are based on the concept of domain generalisation hierarchy.

2.5.1. Utility metrics that are implemented in the application

Contingency Tables

A Contingency table is a matrix-formatted table that displays the multi-variate frequency between categorical variables. Usually, a contingency table depicts the relationship between two categorical variables [30]. There could also be more than two variables, but it is hard to display them visually. The contingency table is a practical utility measure to ensure if the analytical validity of the de-identified dataset is maintained after the de-identification process [31].

Summary Statistics

The de-identification process should aim to not alter the statistical characteristics of a dataset in order to maintain the analytical properties of the dataset. These statistical characteristics refer to *mean, mode, median, minimum, maximum, variance, standard deviation, range, kurtosis and geometric mean* . We briefly describe them below.

- *Arithmetic Mean:* Arithmetic Mean refers to the average of the selected attribute.
- *Mode:* Mode refers to the frequently occurring number in the selected attribute.
- *Median:* Median refers to the middle value in an ordered attribute. The median value divides the dataset such that half of the data points have a lower value to the median and the other half have a higher value.
- *Minimum:* The lowest value in a selected attribute.
- *Maximum:* The highest value in a selected attribute.
- *Variance:* Variance is the measure of variability that signifies how far the numbers are scattered in a given attribute from the arithmetic mean.
- *Standard deviation:* Standard deviation is a measure of the dispersion of a given attribute from its mean. It is calculated as the square root of the variance and has the same unit as the mean.
- *Range:* Range is one of the easiest ways to find out how varied a selected attribute is. Range refers to the difference between the highest and lowest value in a selected attribute. The value of the range is affected by the presence of outliers.
- *Kurtosis:* Kurtosis describes whether the distribution of a given attribute is heavily or lightly tailed compared to the normal distribution. The tails refer to the ends of the distribution. An attribute with positive and high kurtosis has excess data in the tails and more outliers. Whereas, an attribute with negative and low kurtosis has fewer data in the tails and less outliers. The kurtosis value of zero signifies that the distribution is the same as the normal distribution. [32].

- *Geometric Mean:* Geometric mean is technically defined as the nth root of the product of the n numbers. It is useful when the data points in a selected attribute have a multiplicative or exponential relationship [33].

Frequency Distribution

A frequency distribution shows how often distinct values occur for a selected attribute. Frequency distribution is usually depicted in the form of a table or histogram. Each entry in the table represents the number of occurrences of values within a particular group or interval [34].

Discernibility Metric

Discernibility metric is a utility metric that functions by penalizing the records in a dataset if they become indistinguishable in terms of their quasi-identifiers set (QIS) from other records or suppressed after the de-identification process [35]. A record is assigned a penalty 'e' if it is indistinguishable from 'e' other records in the dataset. The penalty of the entire group indistinguishable in terms of their QIS (an equivalence class) is e^2. If the record is suppressed, then it is assigned the penalty 'D', which is the size of the dataset. This is because the suppressed record cannot be distinguished from any other records in the dataset. The total discernibility score is calculated as the sum of all these penalties over the entire dataset as shown below [29]:

$$C_{DM} = \sum_{e \epsilon E} |e|^2 + |S||D|$$

where E is the set of equivalence classes in a dataset D. S refers to the set of suppressed records. Discernibility metric assigns penalty based on equivalence class and not on actual information loss [29]. It is assumed that equivalence classes are the results of the data de-identification process. Also, the discernibility metric disregards the data distribution in the original dataset [36]. This means, if there are existing equivalence classes in the original dataset, the discernibility metric will not ignore it in its calculation and can report a higher information loss.

Average Equivalence Class Size Metric

Another metric similar to the discernibility metric that measures the average size of groups of indistinguishable records is the average equivalence class size metric [30]. As proposed in [37], it is calculated as follows:

$$C_{AVG} = \left(\frac{total\_number\_of\_records}{total\_number\_of\_equivalence\_classes}\right) / (k)$$

where k is the minimum size of the equivalence class. Also, like the discernibility metric, the average equivalence class metric does not ignore the pre-existing equivalence classes in the original dataset. Both discernibility metric and average equivalence class size metric do not depend on generalisation hierarchies [38].

2.5.2. Other utility metrics

These utility metrics are not implemented in the application. Some of these are mathematically advanced and are based on the concept of domain generalisation hierarchy. In domain generalisation hierarchy, each value of an attribute is presented as a leaf in the tree. The higher nodes represent the higher levels of generalisation consisting of the generalised value of the leaf node [36][39]. For example, value 28 of attribute age is a leaf node. The parent node or the generalisation level 1 is 25-30, further the generalisation level 2 is 20-30, and so on in the domain generalisation hierarchy. Since the application does not support the creation of domain generalisation hierarchies, these metrics could not be implemented.

Number of missing values

This metric also referred to as 'missingness' is one of the simple metrics that computes information loss based on comparing the number of deleted values in the de-identified and the original dataset. The metric is an intuitive measure to generate information loss caused by the use of suppression technique and not generalisation [31] [29].

Number of records changed

It is a similar metric to the number of missing values, but it considers the values that have been altered due to the de-identification. Also, it includes the values that have been suppressed. This metric is a good indicator to understand the impact caused by the de-identification process [31].

Ambiguity Metric

The ambiguity metric quantifies the degree of uncertainty or ambiguity in the resulting de-identified dataset caused due to generalisation [40]. This metric quantifies this degree of ambiguity by calculating the possible combinations of the records of the original dataset, that the resulting de-identified dataset can represent. As proposed in [41], ambiguity metric is calculated as the average size of the Cartesian product of the generalised attributes in each record in the table.

$$C_{AM} = \frac{1}{n} . \sum_{i=1}^{n} \prod_{j=1}^{r} |\overline{R_i}(j)|$$

In the above equation, n refers to the total number of records and r refers to total number of attributes in the original dataset. It is assumed that after generalisation, a value in the dataset is replaced by a subset of values consisting of the original value. For example, the value 'male' of attribute 'gender' is generalised to the subset {'male', 'female', 'other'}. In the equation, $\overline{R_i(j)}$ is the subset values of the original value $R_i(j)$ (value of j^{th} attribute of record i in the dataset).

Classification metric

Classification metric is an apt metric when one needs to train classifier over the de-identified data [42]. In this, one attribute is considered as a class label. This metric then functions by penalizing records in an equivalence class based on quasi-identifiers set (QIS), if the value of their class label is different from the majority value of that class label of other records in the equivalence group or if the record is totally suppressed [43]. The classification metric is calculated as the average of penalties of all records [29].

$$C_{CM} = \frac{\sum_{i=1}^{n} penalty(row_i)}{n}$$

$$penalty(row_i) = \begin{cases} 1 & \text{if } row_i \text{ is suppressed} \\ 1 & \text{if } attribute(row_i) \neq majority\_class(e) \\ 0 & \text{otherwise} \end{cases}$$

In the above equation, a dataset consists of a total number of n rows. After de-identification, if a row is suppressed in the equivalence class, then the penalty for the row is 1. If the class label (outside of QIS attributes) has a different value than the majority value of that class label in the equivalence class e, then the penalty for the row is 1. Otherwise, there is no penalty for the row. The rationale behind the classification metric is that for accurate classification, it is preferable that all the attributes in an equivalence class have the same values [43].

Minimal Distortion

Minimal distortion metric [36] quantifies the information loss in the de-identified dataset by incrementing the distortion count every time by 1 if the value in a record is generalised. This metric is based on the concept of domain generalisation hierarchy in which each value of an attribute is a leaf node in the taxonomy tree. The higher nodes in the tree represent the higher generalisation levels consisting of generalised value for the leaf node. If an attribute in a dataset of 10 records is generalised by 2 levels, then the minimal distortion count of that attribute is 20. The problem with this approach is that all generalisations are treated equally. For example, generalisation of age attribute

from 27 to 25-30 may not result in much information loss than generalisation of 'City' attribute to 'Country'.

Loss Metric

Similar to minimal distortion, the loss metric targets the information loss generated by generalisation. It is defined as the number of nodes a record's value for a given attribute is made indistinguishable in comparison to a total number of leaf nodes in the domain generalisation hierarchy [43][36]. The loss metric is then calculated for each record and is then averaged to calculate the overall loss metric for the selected attribute.

$$C_{LOSS} = \frac{number\_of\_nodes\_indistinguishable\_from}{total\_number\_of\_leaf\_nodes - 1}$$

For example, Figure 2.4 depicts the generalisation hierarchy for the attribute 'Education' in a dataset. If the value 'Bachelors' is generalised to 'University', then it cannot be distinguished from the other 2 nodes 'Masters' and 'PhD'. Considering the total number of leaf nodes being 6, the loss metric for each records generalised to 'University' in regard to the 'Education' attribute is $2/5$.

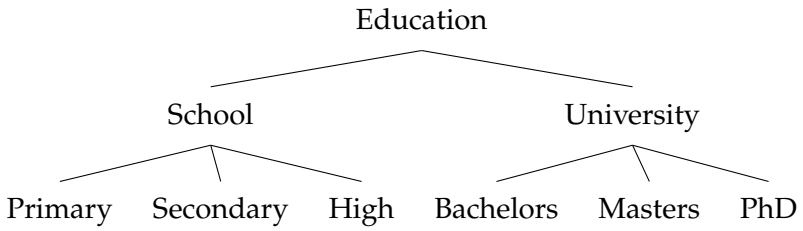


Figure 2.4.: Domain generalisation hierarchy for attribute 'Education'

Kullback-Leibler (K.-L.) Divergence

KL divergence metric measures the information loss by calculating the differences in the probability distributions of the original and the de-identified dataset [42].

Non-Uniform Entropy Measure

Non-uniform entropy measure, also referred to as mutual information utility metric, is an information-theoretic measure. It is based on the concept of mutual information which measures the amount of information that can be revealed by one attribute by providing the value of another attribute in the dataset [40].

3. Related Work

In this section, work reflecting the growing importance and the use of de-identification techniques is discussed. At first, we discuss the de-identification methods, then the data utility metrics, and afterwards, we present the frameworks or tools which use the de-identification techniques to achieve privacy.

3.1. De-identification Methods

In the book *"Guide to the De-Identification of Personal Health Information"* [19], concepts regarding the need and the application of the de-identification process in the sector of health data are explained in great detail. The book highlights de-identification as one of the most valuable privacy by design solution that protects individual's privacy while at the same time allows data to be used for secondary purposes.

Gloria Bondel et al. in *"The Use of De-identification Methods for Secure and Privacy-enhancing Big Data Analytics in Cloud Environments"* [5] argue that the de-identification is the aptest solution to achieve security and privacy for storing and processing big data in the cloud environments. Also, in their subsequent paper *"Towards a Privacy-Enhancing Tool Based on De-Identification Methods"* [9], they provide a thorough overview of de-identification methods and categorize them into perturbative and non-perturbative methods.

Both the *"ISO/IEC 20889"* [8], an ISO standard for data de-identification techniques, and *"NISTIR 8053"* [7], the official documentation by NIST, provide an overview of concepts relating to de-identification of data and establishes the standard terminology for their application. However, the information regarding how these particular techniques should be used in a particular use case is not provided.

In his master thesis *"Identification and Evaluation of Concepts for Privacy-Enhancing Big Data Analytics Using De-Identification Methods on Wrist-Worn Wearable Data"* [4], Kevin Baumer presents a comprehensive evaluation of the current state of the art of privacy enhancing approaches based on de-identification methods and an analysis of the privacy requirements and concepts in the use case of wrist-worn wearables.

3.2. Data Utility Metrics

Data utility metrics, also referred to as quality metrics, exist to measure the information quality of the resulting de-identified datasets in regards to a specific goal. O. Tomashchuk et al. in *"A Data Utility-Driven Benchmark for De-identification Methods"* [13] extensively discuss the importance of quantifying the data utility losses using utility metrics in order to evaluate and improve the de-identification process.

Fletcher et al. [36], and Podgursky [29] in their researches provides an overview of existing utility metrics in detail along with their usage and limitations. They point out that there exists no single utility metric that can robustly evaluate the quality of de-identified datasets in every scenario. Different scenarios require different approaches to data quality assessment. For example, statistical metrics (mean, mode, median, variance, etc.) are a great way to ensure that the analytical properties are preserved in de-identified data [31]. Or to understand how the distribution of values of attributes are affected, the discernibility metric is more suited[13].

Both ARX [30] and sdcMicro [31] supports variety of data utility metrics. ARX has categorized them into cell-oriented, attribute-oriented and dataset-oriented models. Whereas, sdcMicro has categorised the metrics based on the type of attribute value. Eicher et al. [40] in their research have compared the performances of several quality models and concluded that even though different models are suitable for different applications, Non-Uniform Entropy model has the potential to serve as a general purpose model.

3.3. Privacy tools based on De-identification Methods

In the domain of privacy-preserving open-source software, ARX Data Anonymization Tool [30] is a renowned cross-platform tool. It supports a wide variety of anonymization techniques comprising of a range of privacy models, de-identification methods, risk analysis tools, and data utility models. Also, it has been listed in the guidelines by the European Union Agency for Network and Information Security (ENISA) and UK Anonymization Network (UKAN) for implementing data privacy principles [44].

Other open-source tools, mainly from universities and research communities include CAT Anonymization Toolkit [45], UTD Anonymization Toolbox [46], TIAMAT [47], SECRETA [48], and Amnesia [49]. They are less comprehensive compared to ARX and implement a limited set of privacy models, and de-identification methods, which mostly comprise of generalisation and suppression.

Furthermore, μ-Argus [50] and sdcMicro [51][52] are two well-known tools originating from the statistics community to protect the disclosure of statistical microdata. These tools are highly comprehensive and offer features ranging from a wide variety of disclosure risk assessments, utility analysis, and data de-identification techniques. The

application of these privacy features is more manual, incremental, and iterative. At first, the risk is assessed, then, the de-identification is performed, and finally, the utility is evaluated until the desired outcome is achieved. The book *"Statistical Disclosure Control for Microdata"* by Matthias Templ [53] and the user manual of μ-Argus [54] are not only user guides for these tools, but it also explains the concepts related to these tools in a highly detailed manner.

Additionally, Novartis, a global healthcare company based in Switzerland has its own data de-identification tool to process sensitive clinical data [55]. The tool follows the Safe Harbor Approach, which is focused on de-identifying the data attributes that fall into one of the 18 categories of Personal Identifying Information (PII), as described by the Health Insurance Portability and Accountability Act (HIPAA) privacy rules. The tool applies 5 different types of de-identification methods categorised into 2 types: Masking and Removal. Masking includes translation, date offset, and age categorization methods. Removal includes dropping the data fields and setting the values to null.

Lastly, some other well known commercial tools that offer privacy solutions based on de-identification methods are IBM InforSphere Optim Data Privacy [56], Privitar: Enterprise Data Privacy Software [57], PHEMI: Data Privacy Manager [58], Aircloak [59], Thales: Tokenization and Dynamic Data Masking [60]. However, these tools are commercial and hence their implementations are private, so very little is known about how these tools operate and apply their de-identification techniques.

4. Research Approach

This section provides a brief overview of the research methodology used in this thesis.

The thesis is conceptualised and implemented on the principles of design science research methodology as defined by Hevner in *"A Three Cycle View of Design Science Research"* [61]. The methodology revolves around the creation and evaluation of innovative artifacts that aims to improve the processes in a given application domain. Additionally, the methodology are driven by both business needs and a thorough scientific knowledge base to develop these design artifacts. In our case, we are trying to optimise the data de-identification process at enterprises through data utility metrics. So it is essential to design a solution that caters to the business requirements of the enterprises.

Using the systematic approach defined in the design science research paradigm, Figure 4.1 depicts the research process followed in this thesis. There exists three main parts, *environment, design science research* and *knowledge base*. The interaction between these parts are defined through *relevance cycle, rigor cycle* and *design cycle*. In the subsequent paragraphs, we discuss these parts in relevance to our research.

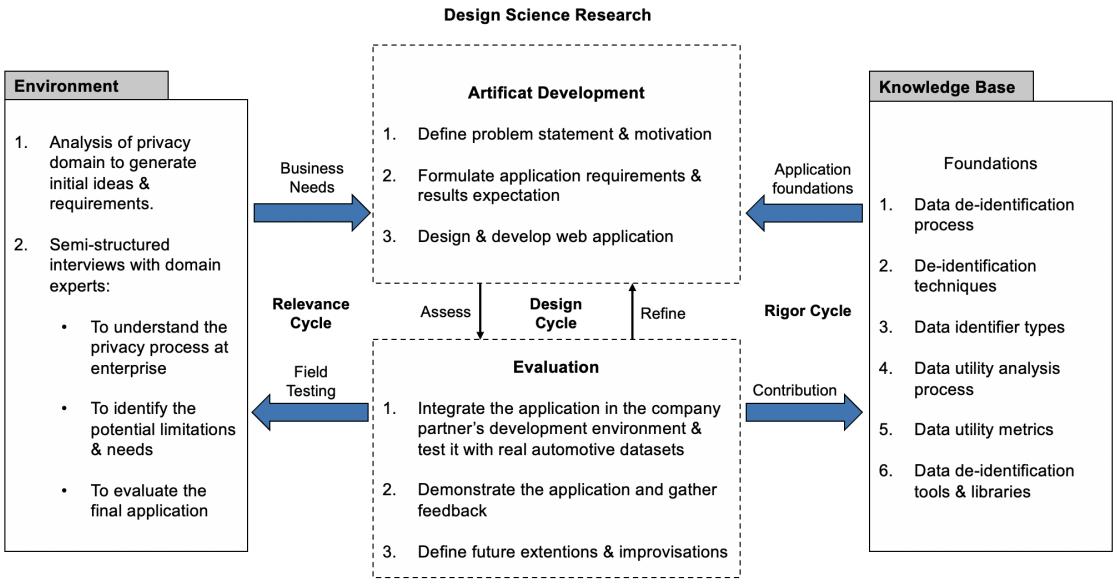


Figure 4.1.: Research Process Overview [61]

The environment represents the interest of the research. This is where one identifies

the scope of the problem and the requirements of the respective design solution. In this thesis, we aim to improve the privacy process at enterprises that use data de-identification techniques as a privacy-enhancing technology to comply with the privacy regulations. Even though the solution presented in this thesis could be applied to any enterprise in general, the problem statement and the solution requirements are derived in the context of our automotive industry partner. To do so, semi-structured interviews with the research partner are conducted to identify the potential opportunities and limitations in the existing privacy processes, thus, highlighting the relevance of the research project. The final evaluation of the design solution is based on the criteria defined in the application environment. Thus, a formal evaluation of the application is carried out by demonstrating the application to the experts in the privacy domain. Their feedback is highly critical in addressing the shortcomings of the research solution and the potential ways of improvement.

The knowledge base compromising of scientific literature, past knowledge and experiences, sets the foundation and guides the development of the design science research process. In particular, it provides an overview of concepts as well as state-of-the-art developments relating to data privacy. To achieve comprehensive knowledge in regards to data de-identification techniques and utility metrics, adequate sources from databases such as *IEEE Xplore Digital Library, ACM Digital Library* and *SpringerLink* are consulted. In particular, the data utility-driven benchmark presented by Tomashchuk et al. [13] and the ARX [30] data anonymization tool are the two main researches steering the development of this thesis.

The findings from the application domain and the foundations from the knowledge base facilitate the design and development of the data utility analysis tool as a web application in design science research. The design science activities begin with defining the problem statement, formulating the requirements, designing the potential solution, and further refining and finalizing them through continuous discussions with research partners. Additionally, an evaluation of the extensibility and usability of the web application is carried out through integrating the application in the company partner's development environment to test it with the real car datasets.

5. Application of Data De-identification at an Automotive Enterprise

This chapter introduces the use case of the application of data de-identification in the context of an automotive enterprise. We begin by describing a typical scenario where appropriate privacy measures are to be applied to the datasets. In the later sections, we present a detailed workflow of the data de-identification process in a large-scale enterprise and finally conclude the chapter on the limitations of the de-identification process.

5.1. Description of the use case

The automotive industry is leveraging the benefits of information-centric technologies such as Artificial Intelligence (AI), big data analytics, the Internet of Things (IoT), blockchain, and now even quantum computing [62] to enable modern innovations, ranging from efficiency to connectivity to autonomous driving to electrification and new mobility solutions [63]. These cars are connected now more than ever. Through the internet they are able to communicate with other cars, smartphones, networks, driver, and passengers [10]. Packed with numerous sensors ranging from cameras and GPS to accelerometers and event data recorders, these cars collect and record vast amounts of data [64]. Carmakers use this data to provide a variety of benefits and services of connected vehicles in the domain of road safety, fuel consumption, predictive maintenance, sustainable driving, etc. [65]. A connected car may generate up to 25 gigabytes of data per hour from at least 200 sensors within the vehicle [11]. However, this massive collection and use of vehicle data pose significant privacy risks. Vehicle data not only reveal the driver's footprint but can report on their driving style, their lifestyle, and locations of interest [64].

The connected car data is considered as personal data and must be processed in accordance with privacy regulations around the world [12] such as the General Data Protection Regulation (GDPR) [1] in Europe, California Consumer Privacy Act (CCPA) [2], and Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada [3]. This means organizations can only process the personal data for the purposes the user has consented to. Considering how automotive industries are transitioning into data-centric organizations, customer and vehicle data is of great value to the organizations to build their product and services. For example, this data is essential

to perform advanced analytics to improve vehicle performance, provide connectivity, mobility services, build customer loyalty, etc. [66].

To facilitate the secondary use and disclosure of data, companies resort to de-identification techniques, such as pseudonymization and anonymization [67]. As per the privacy regulations, principles of data protection do not apply to anonymous information namely, *"[...] information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable"* [10]. Therefore, data de-identification is a good strategy to address the GDPR compliance regulations while at the same time utilize the benefits of data to develop products and services.

Data de-identification is a privacy-enhancing process that makes the data private by removing, concealing, or replacing the personal identifiers of a dataset to prevent one's identity from being disclosed [6]. De-identification is carried out through various de-identification techniques as discussed in section 2.2 with varying levels of effectiveness. Even though these techniques help to achieve privacy, their application has a direct impact on data utility causing information loss in the data. Therefore, it is highly critical to understand the effects of these techniques on the data to achieve a perfect balance of privacy and utility.

Usually data de-identification is not just an independent process but rather dependent on the outcome of other processes such as identifier classification, risk assessment and utility assessment. In the next section, we discuss this overall process of de-identification along with these other processes.

5.2. Data De-identification Methodology

Based on the expert determination method of the Health Insurance Portability and Accountability Act (HIPAA), professors El Emam and Malin derived a methodology for de-identifying structured data [21]. Many institutions have adopted these guidelines to understand and apply the concepts related to de-identification [20] [7]. Additionally, this methodology sets the foundation of the de-identification framework developed by HITRUST Alliance [68]. Below we will discuss the important elements of the methodology[21].

- *Classification of attributes in the dataset based on their identifier types*

  An identifier is an attribute or a set of attributes that could either directly or indirectly identify the data subject in a dataset. As discussed in subsection 2.1.2, these identifiers are of four types: *direct*, *indirect*, *sensitive* and *insensitive*. Since de-identification is concerned with the removal of attributes that could contribute in re-identification, it is essential to classify the attributes based on their identifier types.

- *Masking or removal of direct identifiers*

  In order to comply with the privacy regulations, all the direct identifiers in a dataset should be removed or otherwise transformed using masking or pseudonymization techniques. After the removal of direct identifiers, there exists no risk of re-identification through direct identifiers.

- *Assessment of re-identification risks*

  Through use of appropriate risk metrics, the datasets are examined to identify any potential risks of re-identification and an acceptable risk threshold is determined. Additionally, risk mitigating strategies are defined in this step as well.

- *Application of de-identification techniques*

  If the risk associated with the dataset is higher than the risk threshold, then appropriate de-identification techniques are applied to the dataset. This step is repeated until the re-identification risk is lowered than the risk threshold.

- *Assessment of resulting data utility*

  Since de-identification often results in information loss, it is necessary to quantify this loss through the data utility metrics. If the resulting data utility is low or could be improved while at the same time respecting the re-identification risk threshold, the de-identification process is repeated.

To enhance the data de-identification process in a large enterprise, it is first essential to understand their privacy process. In the next section, we present a brief scenario where the application of privacy is considered on the datasets.

5.3. General Scenario of Privacy Application on Datasets

To adhere to the privacy regulations, enterprises employ various policies and processes to protect and block direct access to users data. For example, paying attention to administrative privileges, who has access to data both internally or externally, and for what purpose is the data being accessed [69]. Figure 5.1 depicts one such process.

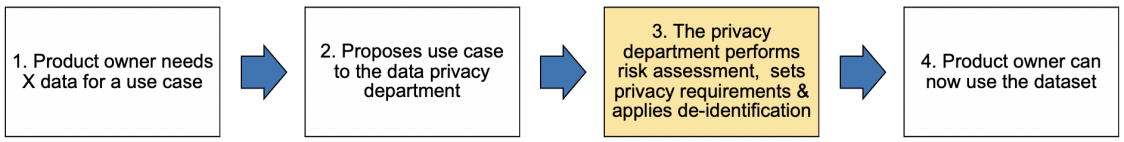


Figure 5.1.: General privacy process in an enterprise

Typically, if a product owner requires datasets for a particular project, they submit a dataset request to the privacy department of the enterprise. The privacy department then

processes this request. The department performs various safety checks and decide on a privacy strategy to reduce any privacy risks. They then apply necessary de-identification techniques to achieve the required privacy level. The resulting de-identified datasets are then provided to the product owner.

5.4. Overview of Data De-identification Process

Figure 5.2 depicts the workflow diagram of the above mentioned privacy scenario. There are 3 primary roles in this workflow: *Product Owner, Privacy Experts* and *Data Engineers*. As explained before, the product owner submits a request to the privacy department to get access to the datasets required for a project. This request generally contains the description of the project, information regarding the datasets, tasks associated with the datasets, etc. This request is then processed by the privacy department.

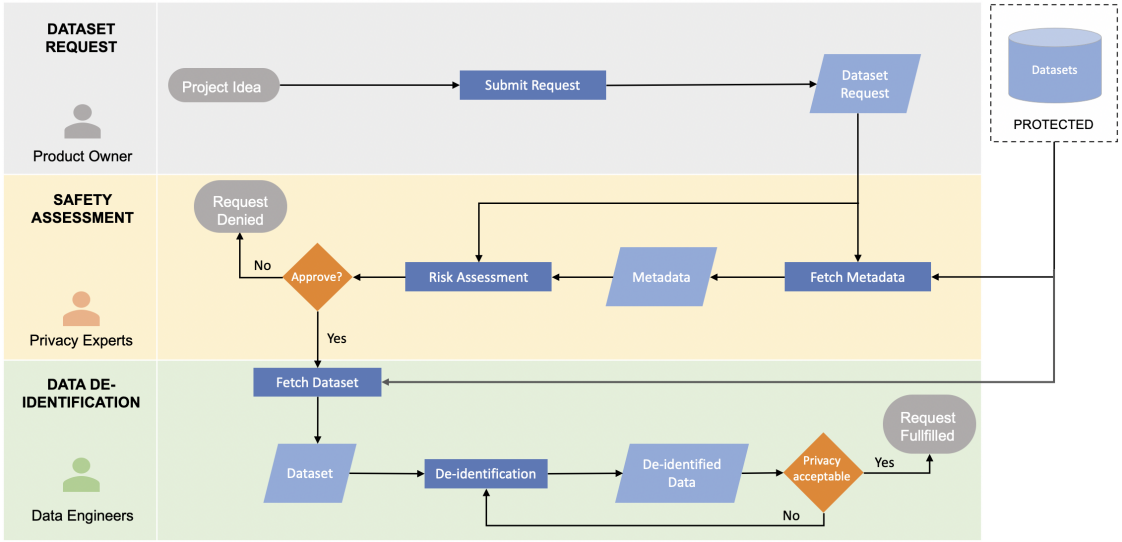


Figure 5.2.: Overview of Data De-identification Process in an enterprise

The privacy department consists of privacy experts and data engineers. The privacy experts are responsible for performing various safety checks based on the information provided in the dataset request and the metadata of the requested datasets. The metadata refers to the data description of the datasets. The request for the datasets could be rejected depending on the rules surrounding the datasets as well as re-identification risks associated with the datasets. For example, if the request is to access a highly classified dataset or if the combination of datasets could result in the re-identification of users, the privacy department can refuse such a request. The privacy experts perform tasks such as determining whether the proposed project conforms to the official privacy regulations and data protection rules at the enterprise, assessing re-identification risk

associated with the datasets, and identifying any privacy loopholes with the project in general. If the request passes all the safety checks, this request is then forwarded to the data engineers.

The team of data engineers are responsible for preparing the datasets. Based on the safety assessment done by the privacy experts, they determine the appropriate privacy strategy that addresses any privacy and re-identification related risks in the requested datasets. They then apply necessary de-identification techniques to achieve the required privacy level. In this process any information that could lead to re-identification of users is removed or altered. The resulting de-identified dataset is then provided to the product owner for the project.

In the following section, we discuss the potential shortcomings of this data de-identification process and later reflect on the way it could be enhanced.

5.5. Limitations of Data De-identification Process

The privacy process comprising of safety assessment, creation of respective privacy strategy and application of de-identification techniques is in general a time intensive and highly iterable process. The privacy experts are required to make decisions regarding the acceptance of the request. Often while deciding on the privacy strategy, it is hard to estimate whether the resulting utility of dataset after de-identification will be adequate for the project or not. Two major limitations that were discovered while understanding the process of de-identification in a large enterprise setting are as follows:

- *Data de-identification is often a manual process*: Considering the vast variety and sheer scale of automotive data generated from various sources and processed in a highly complex environment, it is a hard to automate the process of de-identification. The data has to be first individually analysed, personal attributes needs to be classified and only then can the de-identification be performed.
- *De-identification is carried out based on the rules decided by privacy experts and data engineers*: Often there are pre-established rules based on which de-identification is carried out. For example, latitude and longitude should be rounded to 2 decimal points, the speed should be generalized with interval size 100, the vehicle number should be pseudonymized, etc. These rules are often described by privacy experts and data engineers. However, this approach can sometimes generate privacy loopholes or cause unnecessary information loss in the dataset. For example, even after de-identification there may exists unique records in a dataset consisting of sensitive information, which can lead to an information leak or sometimes re-identification, or if essential attributes in a dataset are de-identified, the data utility may not be enough for a particular project.

6. Design of Data Utility Analysis Tool

In this chapter we discuss our proposed solution to enhance the data de-identification process as discussed in chapter 5. First, we present the proposed solution and later discuss its design and components.

6.1. Proposed Solution

The data de-identification process depicted in section 5.4 and the fact that the de-identification process is always accompanied by the loss of information, raise 2 major questions:

1. How does one decide whether the resulting utility of dataset after de-identification is adequate for a given project?
2. How does one decide which set of de-identification techniques is suitable for the requested datasets in a way that the data utility is not significantly impacted?

Different de-identification techniques affect the data differently and could result in different levels of information loss [7]. For example, rounding reduces the accuracy of data, noise addition affects the truthfulness of data, character masking conceals the part of the data or the information loss caused by the integer generalisation of step size 10 is less than the step size 30, etc. To understand the impact of the de-identification process on the utility of the data, we suggest the use of data utility metrics. These metrics as discussed in section 2.5 quantify the changes in utility/information-loss in de-identified data with regards to a specific goal [13]. These utility metrics help in understanding the implications of the de-identification techniques on the data by providing an overview of the resulting utility of the de-identified dataset to the data engineers. Based on this, they can decide on a better and optimal de-identification strategy that could yield better data utility for the requested datasets while at the same time not compromising with the privacy.

We propose the solution of combining the data utility analysis process with the process of data de-identification. Figure 6.1 depicts the proposed alteration to the de-identification process. In this, the requested datasets by the product owner provided to the data engineers team after the safety assessment by the privacy experts, are first de-identified. Then, utility analysis of the de-identified dataset is performed through various metrics. For the utility assessment, both the de-identified dataset and the original dataset is

required to compare the change in the utility. Afterward, a utility acceptable check is performed, in which the data engineers decide if the resulting utility is acceptable. If not, then the application of de-identification techniques is repeated unless the required utility for the dataset is achieved. Then, a check for privacy requirements are made, where it is assessed that for a given utility, the privacy requirements are met or not. If yes, then then an effective de-identification is achieved, else the process is repeated until a balanced level of privacy and utility is achieved.

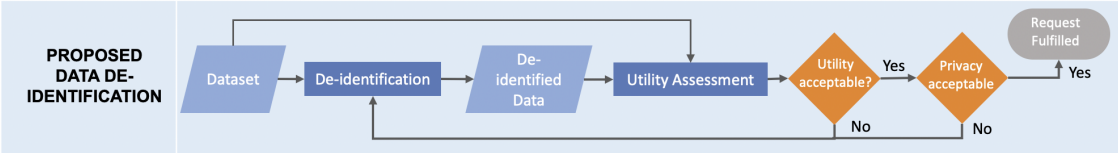


Figure 6.1.: Proposed De-identification Solution

6.2. Design of the Proposed Solution

Figure 6.2 is the detailed representation of our proposed solution. Knowledge base and the input are the 2 main components. In the following sub-sections, we discuss these components and the workflow.

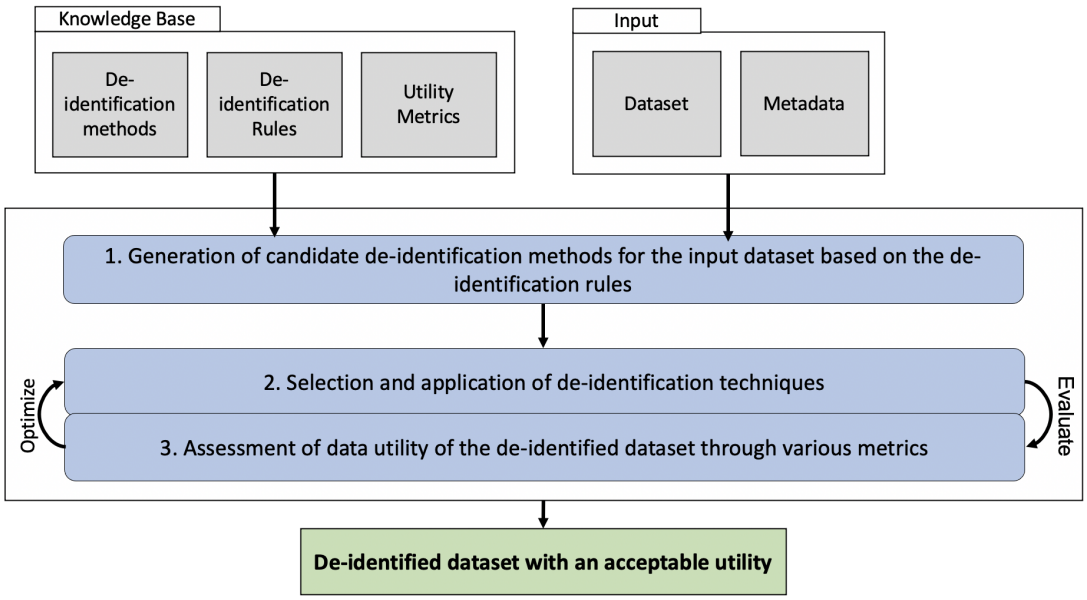


Figure 6.2.: Design of the Proposed De-identification Solution

Knowledge Base

Knowledge Base represents the information that is required to execute the proposed process. It consists of the *de-identification methods, utility metrics* and the *de-identification rules*.

- *De-identification Methods*: Set of de-identification techniques available in the tool to de-identify the data. More the number of these techniques, the more the options data engineers have to de-identify the data. An overview of these techniques is provided in section 2.2.
- *Utility Metrics:* Set of utility metrics available in the tool to measure various aspects of information loss. An overview of these metrics is provided in section 2.5
- *De-identification Rules:* Refers to the set of rules either derived from scientific literature, privacy regulations or defined by the enterprise itself based on which de-identification techniques are applied to the dataset. For example, all directly identifying attributes should be suppressed or pseudonymized. Or for indirectly identifying attributes, with float data type the possible de-identification techniques are rounding, noise addition and suppression. In our application, we have derived these rules based on attribute's data type (string, float, integer, boolean) and identifier type (direct, indirect, sensitive, insensitive) [8].

Input

The input comprises of the dataset to be de-identified and it's corresponding metadata. The metadata is required to get information regarding the attribute's data type. Using this information and the de-identification rules, de-identification techniques could be suggested for the attributes in the dataset.

Workflow

Here we describe the 3 main processes in our enhanced de-identification approach.

1. *Generation of candidate de-identification Methods*:

   For each attribute in a given dataset, de-identification methods as potential candidates are generated. They are generated based on the data de-identification rules. To execute these rules, information from the metadata of the dataset such as attribute data type or identifier type is required.

2. *Selection and application of de-identification techniques:*

   From the generated list of de-identification techniques, data engineers first select for each attribute a de-identification technique and set its appropriate parameters. Once the selection is done, these techniques could be then applied to the dataset.

3. *Utility assessment of the de-identified data*:

   Both original and de-identified datasets are compared and their information losses are evaluated through various utility metrics. For example, a statistical summary or frequency distribution of a particular attribute can be generated and compared. If the data utility of the de-identified dataset is unsatisfactory, then the de-identification process is repeated. Either other de-identification methods are chosen from the generated candidate methods, or the parameters of the de-identification techniques are changed. The data utility is then assessed again. This process is repeated until a de-identified dataset with an acceptable utility is generated.

Considering how variable the effects of de-identification techniques could be on the dataset, it becomes essential to measure their implications. Since their application directly affects the data utility, efforts should be made to assess the change in utility. Combining the process of de-identification and utility assessment, would help privacy experts to quantify the performance and optimise the application of de-identification techniques on datasets. Our proposed solution to some extent can help estimate whether the resulting utility of the de-identified dataset would be useful for the project.

7. Implementation of Data Utility Analysis Tool

This chapter describes the development process of the data utility analysis tool. The use case description in chapter 5 and the design of the utility tool described in chapter 6 serves as the basis for the development process. At first, we present the requirements of the tool, then we discuss the application architecture of the data utility analysis tool and finally we discuss the choice of technology and the technical architecture adopted for the implementation of this tool.

7.1. Application Requirements

In this section, the requirements for the development of the utility analysis tool are described. These requirements are derived from the interviews conducted with privacy experts. These requirements are categorized into functional and non-functional requirements.

7.1.1. Non Functional Requirements

- *The tool should be able to process large amounts of data.*

  In our use case, the datasets are requested by the product owners for their projects. These datasets are of large sizes as advanced analytics is performed on them. Therefore, in order for our tool to be useful in this context, it should have no limitations in de-identifying and performing utility analysis on large amounts of data.

- *The tool should offer an easy to use interface.*

  Deciding on the de-identification strategy for a dataset is in itself a complex procedure. Moreover, most of the third-party de-identification tools available usually offer a complicated user interface. For the tool to stand out and be accepted, it should offer a simple design and interface. All the relevant information necessary to perform a proper de-identification should be displayed neatly and concisely.

7.1.2. Functional Requirements

- *Data de-identification and utility analysis should be done on the server-side.*

  To enable faster processing of large amounts of data and to prevent the exposure of data anonymization logic, it is suitable to perform the de-identification and utility analysis on the server-side.

- *The application should support data to be uploaded from the cloud storage.*

  In large enterprises, the data is stored, processed, and analyzed in clouds. Even the connected cars directly pump their data to the cloud and from there it is shared with other processes. Therefore, the application should allow the uploading of data from cloud storage.

- *The data should not be allowed to be downloaded.*

  In order to prevent any re-identification risks by combining different de-identified datasets, the user should not be allowed to store data locally. Additionally, only a small portion of the data that needs to be de-identified should be displayed in the application so that the user can choose the appropriate de-identification techniques for the dataset. Based on the choice of the de-identification strategy, the requested dataset should be processed and de-identified on the server-side and then stored in the cloud. Further, only a small portion of the resulting de-identified dataset should be displayed in the application.

- *The de-identified data should be stored in the cloud.*

  Since the resulting data is not allowed to be downloaded and is required in the data utility analysis process to compare the amount of information loss, it should be stored in the cloud. After the successful de-identification process, this de-identified data should be shared with the product owner.

7.2. Application Architecture

Application architecture plays an important role in the development of the application as it describes the application components, their relationships, their interaction with each other and provides concrete guidelines that help to evolve the application [70]. In this section, we present the architecture of the data utility analysis tool starting from a higher perspective with a use case diagram, then depicting the application process through a sequence diagram and then depicting the lower level and detailed architecture through a component diagram.

7.2.1. Use Case Model

The use case model is essential in making the architectural decisions about the application by providing the context of the application, the way the external actors interact with the application, and the flow of functionalities in the application [71]. Figure 7.1 depicts the use case model of our data utility analysis tool.

The main users of the data utility analysis tool are the data engineers as they are the ones responsible for performing data de-identification as discussed in section 5.4. The application provides them with 3 major functionalities: *loading of the dataset*, *de-identification of the dataset*, and *utility assessment of dataset*.

- *Load dataset*: The dataset that is required to be de-identified is first loaded into the system from some external cloud data storage. Upon loading, the dataset and its respective metadata are displayed in the application.

- *De-identify dataset*: Based on the metadata and the original dataset provided in the application, the data engineer then defines the data de-identification strategy for the dataset. This data de-identification strategy comprises the de-identification technique along with its respective parameters for each attribute. After the de-identification, the de-identified data is rendered back in the application.

- *Analyze data utility*: The application provides the data engineer with various utility assessment metrics. Based on the de-identified data and the original data as displayed in the application, through these utility metrics, the data engineer can then visualize the change in data utility.

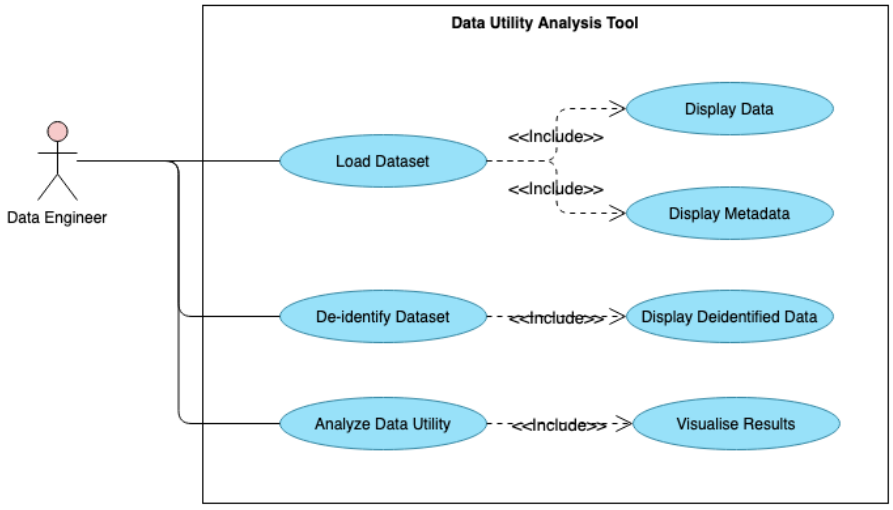


Figure 7.1.: Use Case Model of Data Utility Analysis Tool

7.2.2. Sequence Diagram

The sequence diagram of the application is illustrated in Figure 7.2 depicting the sequential interactions between the important components of the application. As displayed in the diagram, the application comprises 4 main components *the app, the server, the cloud storage (protected dataset)*, and *the cloud storage (requested dataset)*. The app is the user interface of the application, the server consists of the de-identification and utility analysis logic, the cloud storage (protected dataset) consists of the protected datasets, and the cloud storage (requested dataset) serves as storage for de-identified datasets. Below we explain the sequence diagram briefly.

1. **Step 1:** The dataset that needs to be de-identified is loaded into the application. The user provides the details of the dataset in a form of query parameters. These query parameters are used to query the dataset from the cloud storage (protected dataset).

2. **Step 2-4:** The server forms the query from the query parameters it received. Through this query, the server then fetches the requested dataset from the cloud storage (protected dataset).

3. **Step 5-9:** After receiving the dataset, the server stores this dataset in the cloud storage (requested dataset) for future processing. The server then reads 20 records from the dataset and returns them to the app along with the metadata. The app then renders these 20 records and the metadata. For privacy reasons the application only renders at most 20 records of a dataset.

4. **Step 10-19:** Using the metadata and the dataset that is displayed in the application, the user selects the de-identification technique for each attribute. This de-identification information is sent to the server. The server first fetches the dataset from the cloud storage (requested dataset), applies the relevant de-identification technique on each attribute, and then stores the de-identified data back in the cloud storage (requested dataset). The server then reads the first 20 records of the de-identified dataset and sends them to the app which then renders them.

5. **Step 20-28:** After receiving the de-identified data, the user has an option to analyse the data utility through various utility metrics. The user specifies the parameters for a utility metric and sends it to the server. Then server first fetches both the original and de-identified dataset. Then it performs the requested utility analysis using the original and de-identified dataset and sends the results back to the application.

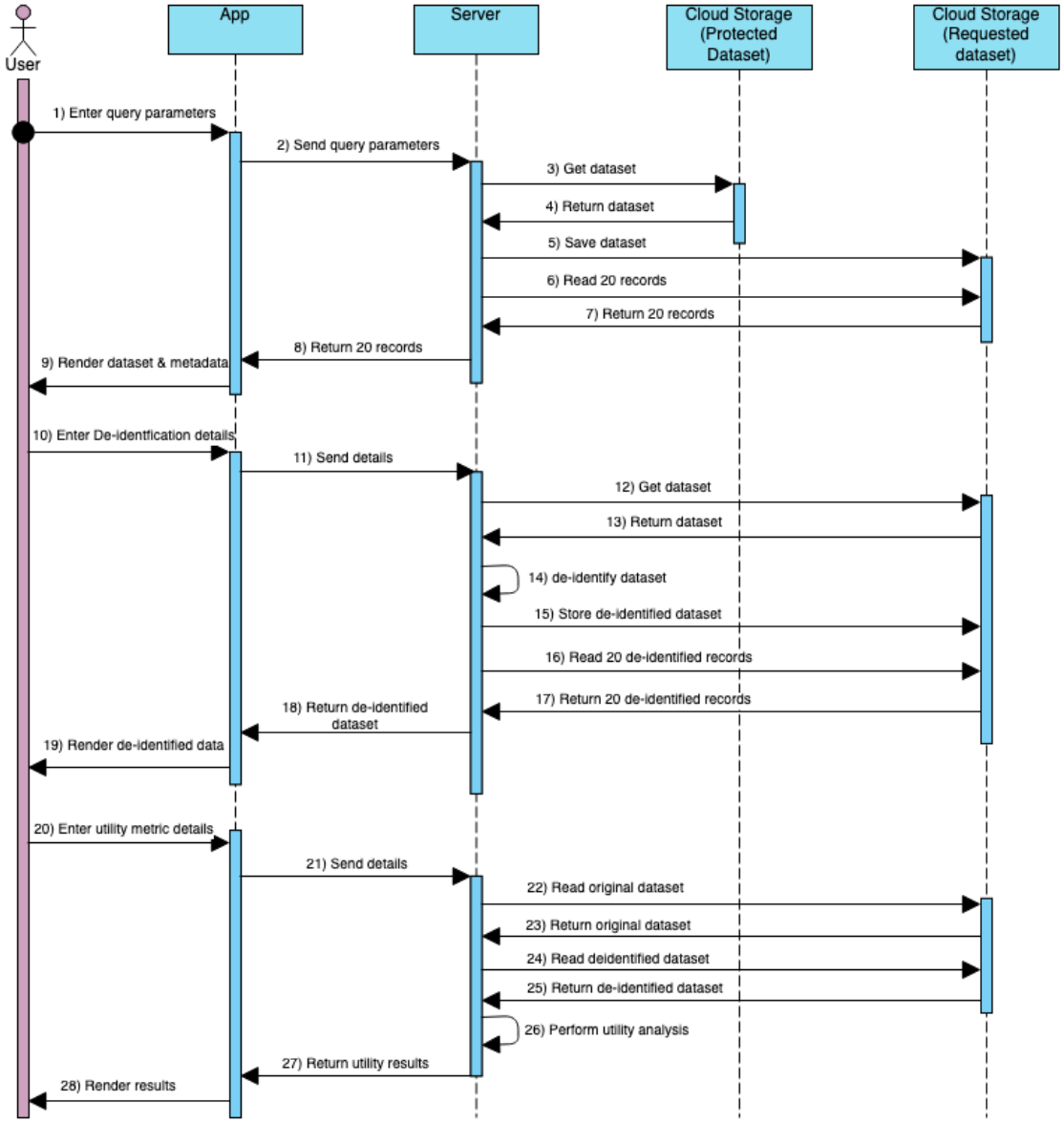


Figure 7.2.: Sequence Diagram of Data Utility Analysis Tool

7.2.3. Component Diagram

In this section, we discuss the structure of the data utility analysis tool. A component diagram is a great way to visualize the physical structure of the application as they provide a static view of the application components, their interaction with other components, their relationships, and their dependencies. Figure 7.3 depicts the component diagram of the data utility analysis tool. Below is the list of these components and their detailed descriptions.

- *Front-end*: The front-end component provides the user interface of the data utility analysis tool and communicates with the back-end and its sub-components through the REST API. The front-end component allows the user to query the dataset, de-identify the queried dataset and then perform the utility analysis based on the de-identified dataset. These functionalities are interdependent to each other and are represented as separate components.
- *Back-end*: The back-end component contains all the logic of the data utility analysis tool. It consists of sub-components that implement the data querying, de-identifying, and utility analysis functionalities. It communicates with the front-end through the REST API and is responsible for accessing data from the cloud storage.
- *QueryData*: The QueryData component is where the application begins. It provides the interface to the user to get the details of the dataset that needs to be de-identified. The component is called queryData because it takes in the query parameters that are required to query the data from the data storage. The query parameters include the database name, table name, attributes and the number of records. These details are then forwarded to the data controller component through the REST API.
- *Data Controller*: The data controller component is responsible for fetching the requested dataset from the cloud data storage and then storing this data in other cloud storage. Furthermore, the component passes a portion of this data as well as the metadata to the front-end.
- *Cloud Storage (Protected Datasets)*: The cloud storage (Protected Datasets) component is the primary source of all the datasets in an organization.
- *Cloud Storage (Requested Datasets)* : The datasets that are required by the product owner and need to be de-identified, are stored in this cloud storage. After the de-identification process, the de-identified data is also stored in this storage.
- *Deidentification*: The de-identification component in the front-end provides the interface to the user to select the identifier type and the de-identification technique and respective parameters for each attribute. Through the REST API, this information is passed to the de-identification controller component in the back-end.
- *Deidentification Controller*: Based on the information supplied by the de-identification component from the front-end, the de-identification controller de-identifies the dataset. The dataset is first fetched from the cloud storage, then de-identified using the de-identification techniques components and then stored back in the cloud storage. A part of the de-identified data is sent back to the front-end.
- *Deidentification Techniques*: The de-identification techniques component comprises all the available de-identification techniques in the application and is responsible for performing those techniques.

- *Utility Assessment*: The utility assessment component in the front-end provides the interface comprising of various utility metrics to the user. The user selects a metric and its required parameters and sends it to the utility analysis controller in the back-end through the REST API.

- *Utility Analysis Controller*: The utility analysis controller component is responsible for performing the utility analysis based on the information supplied by the utility assessment component. The utility analysis controller fetches the de-identified dataset as well as the original dataset from the cloud storage and evaluates the change in utility through the utility metrics component.

- *Utility Metrics*: The utility metrics component consists of all the available utility metrics in the application. Utility Metrics component utilizes the third-party ARX library component to provide these metrics.

- *ARX Library*: ARX is an open source data de-identification tool [30]. In addition to providing data transformation methods, the ARX library provides support for various data utility metrics.

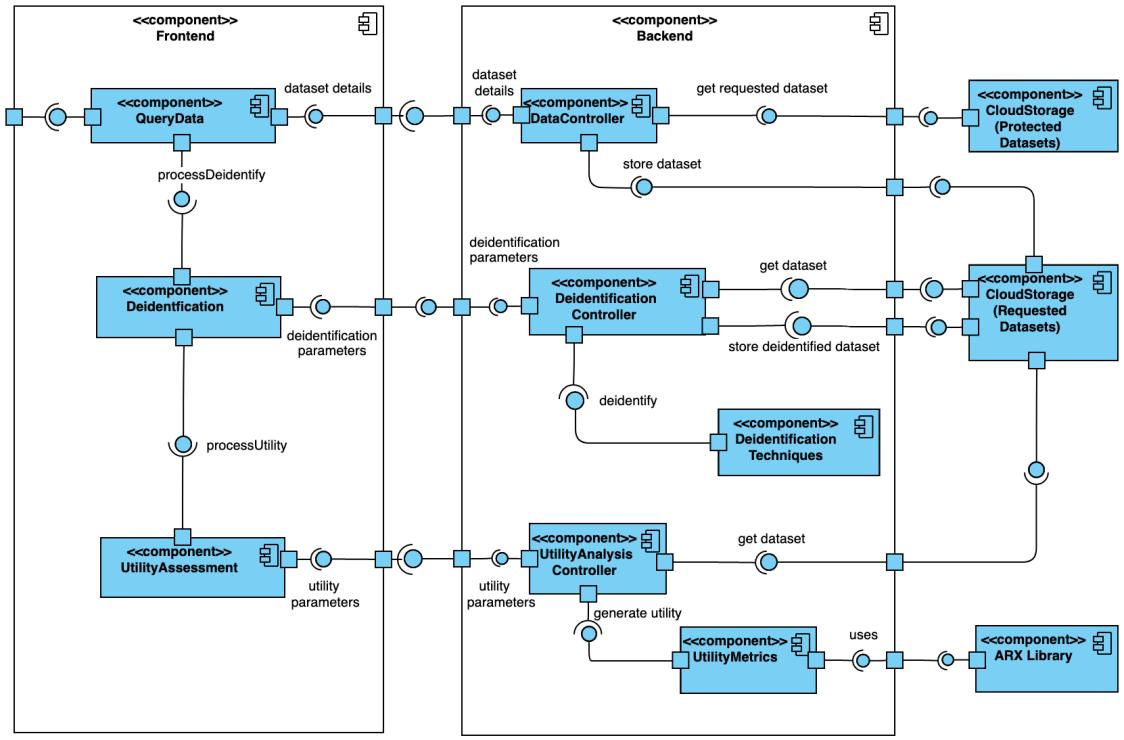


Figure 7.3.: Component Diagram of Data Utility Analysis Tool

7.3. Application Implementation

This section describes the technology stack that is used to build the data utility analysis tool, followed by the technological architecture of the application.

7.3.1. Technology Stack

To understand the technical architecture of the application, it is important to have an overview of the technologies. This section briefly introduces the technologies that are used to develop the application.

React

The front-end of the data utility analysis tool is based on React framework. React is an open-source powerful JavaScript front-end library used to develop highly responsive user interfaces in the form of UI components for web applications and is maintained by Facebook [72]. Being lightweight and offering features such as server-side rendering, virtual DOM, component-based development, etc. React has a huge developers community. For beginners with basic knowledge of HTML, CSS, and JavaScript, React is easy to grasp with a low learning curve as its syntax is very similar to that of HTML.

Amazon API Gateway

Application Programming Interface (API) acts as an intermediary between software components and allows the applications to access functionalities implemented in the back-end. Amazon API Gateway is a server-less service from Amazon Web Services (AWS) that allows creating REST APIs that enable the web application to have two-way communication with the components implemented in the back-end. Ranging from features such as strong authentication mechanism, traffic management, API usage monitoring, Amazon API Gateway allows developers to easily create, publish and manage secure REST APIs [73]. In the data utility analysis tool, the API gateway exposes the API endpoints through which the user interface is able to communicate with the back-end services and data resources.

Amazon Simple Cloud Storage (S3)

Simple Storage Service (S3) [74] is a highly scalable, secure, and durable object storage service offered by Amazon allowing to store any type and amount of data and providing the ability to retrieve it from anywhere. S3 enables the developers to create applications that are based on cloud storage. Additionally, S3 can be used to host static websites or as a file sharing solution. One can even perform analytics and run queries on the data stored in S3 using services such as S3 Select, Athena and Lambda functions. In the data

utility analysis tool, all the datasets are stored and accessed from S3. The data utility tool queries and processes the data stored in S3 using Amazon Athena and Lambda functions. In S3, files are stored in buckets. The files are referred to as the objects of the buckets. Using security policies, one can control who and which applications have the right to access and process the buckets and to what extent.

Amazon Athena

Amazon Athena [75] is an interactive server-less big data analysis tool, allowing you to process complex data queries in relatively less time without worrying about the infrastructure and associated costs. One can query this data stored in Amazon S3 using standard SQL. Automatic parallel execution of queries in Athena enables faster processing of the data. To start querying the data using Athena, one needs to simply have the data loaded in S3. Then, a database schema of the data has to be defined in Athena either manually or automatically. Once this is set up, one can now query this data through Athena using SQL queries. The queries are executed asynchronously and the results are stored in the S3.

Amazon DynamoDB

Amazon DynamoDB [76] is a fully managed NoSQL server-less database allowing one to store data in the form of key-value pair and offers low latency data retrieval. It is fast, reliable, has high performance, and can scale on-demand to support more than 20 million read/write requests per second. The data is organized in the form of tables consisting of several items and each item has its own unique key and attributes. Considering that the tables are schema-less, neither the attributes nor their data types need to be defined beforehand. In the data utility analysis tool, dynamoDB is used to maintain the data de-identification requests and their corresponding status.

Amazon Lambda

Amazon Lambda [77] is a server-less compute service that lets you run your code in a form of functions, on-demand without servers. Adopting the server-less approach by using Amazon Lambda saves time and effort of provisioning and managing the infrastructure required for the back-end. Amazon Lambda supports many different programming languages enabling you to virtually run any type of application or back-end service. A server-less application consists of 2 or 3 components, *event, function* and sometimes *services*. An event refers to anything that can invoke a lambda function such as a request to API endpoint, upload of data to S3, etc. This event then triggers the lambda function that starts running in its own container and allocates resources as required. After the execution of the function, it either returns a result to the invocation source or makes changes to any connected service (such as writing to the database). The

data utility analysis tool's back-end is entirely developed using lambda functions that are triggered by gateway endpoints. These lambda functions are written in Java and interact with data sources in S3 and dynamoDB as well as trigger query processing in Athena.

Amazon Cloud Development Kit (CDK)

Manual provisioning and managing of cloud resources becomes challenging and prone to error when the application grows in size. Amazon provides a solution to this problem through the Cloud Development Kit (CDK). Amazon CDK [78] allows developers to define and deploy the cloud application infrastructure using mainstream programming languages that through flexible and dynamic classes encapsulates the whole bunch of AWS resources. The high level components in CDK called constructs comes with pre-configured defaults helps to build the cloud infrastructure quickly that in turn accelerates the development process of any cloud application. In data utility analysis tool, all the Amazon Web Services (AWS) resources are created and build using CDK software development framework. Through just one command, all the necessary AWS resources are created and the application is easily deployed.

7.3.2. Technical Architecture

The data utility analysis tool has a server-less architecture and is based on AWS infrastructure consisting of Lambda functions, Gateway API, S3, DynamoDB, and Athena. The front-end of the application is designed using the ReactJS framework. Figure 7.4 depicts the architecture of the data utility analysis tool. In this section, we discuss this architecture in detail.

The front-end of the application, based on the ReactJS framework offers 3 main functionalities to the user: *Load Dataset, De-identify Dataset* and *Analyse Data Utility*. The front-end is only responsible for rendering the data and accepting user inputs. The back-end is built using various Amazon web services as depicted in Figure 7.4. Gateway API exposes the endpoints that allow the front-end to communicate with the back-end services. All the application logic regarding the access of data sources, data de-identification, and utility analysis are implemented in Java programming language as Lambda functions. The DynamoDB table 'Request Status' maintains the status of the dataset requests. Lambda functions update the status of the request as necessary. Through AWS Athena, datasets are queried which are further stored in the S3 bucket called DUT. The Lambda layer consists of all the Java and ARX dependencies that are required to build this application. This Lambda layer is shared by all the Lambda functions to provide access to these libraries. Finally, all these AWS resources are defined, created, and deployed through a CDK script.

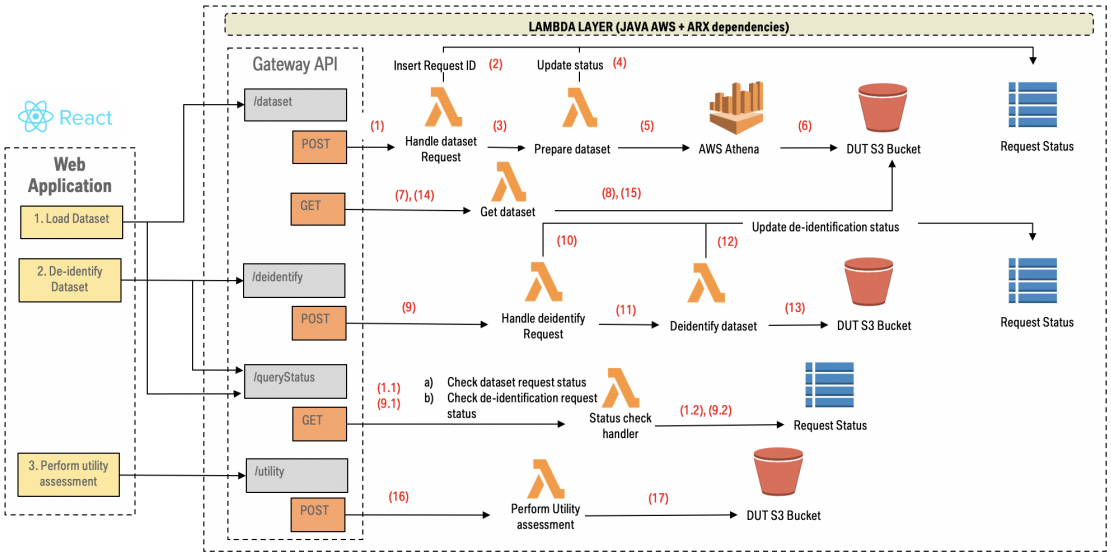


Figure 7.4.: Technical Architecture of Data Utility Analysis Tool

In the subsections below, we describe the functionalities of the application with respect to the technical architecture displayed in Figure 7.4. The description would help understand the process flow and the role of each service in the back end.

Load Dataset and Metadata

The dataset that needs to be de-identified must be first loaded. The user loads this dataset by providing the dataset details in the form of query parameters in the 'Submit Dataset Query' form. Once the form is submitted, a POST request is performed on the '/dataset' API endpoint. This request invokes the lambda function A that handles this dataset request. Considering that the API gateway timeout is 29 seconds, and the processing time for the request is more than that, lambda function A responds back to the application immediately with the message 'Request is submitted' and asynchronously invokes the lambda function B that is responsible for preparing the requested dataset. Additionally, Lambda function A makes the entry of this request in the form of its ID and status in the dynamoDB table 'Request Status'.

Lambda function B processes the dataset request by querying it from Athena. Also, Lambda function B correspondingly updates the status of this request in the 'Request Status' table. After processing the query, Athena stores the resulting dataset as well as its metadata in the DUT S3 bucket.

The application gets notified about the dataset request completion through the pull notification mechanism. Every 5 seconds the application checks the status of the dataset through the GET request at '/queryStatus' endpoint. Once the status of the request is set to 'Done', the application issues the GET request at '/dataset' endpoint. This GET

request invokes the lambda function C, which reads this dataset from the DUT S3 bucket and sends 20 records of this dataset as well as the metadata of the dataset as a response to the request. The application then renders these records along with the metadata.

Figure 7.5 depicts the loading of NYTaxi dataset from 'thesisDB' database comprising of all the attributes and 10,000 records.

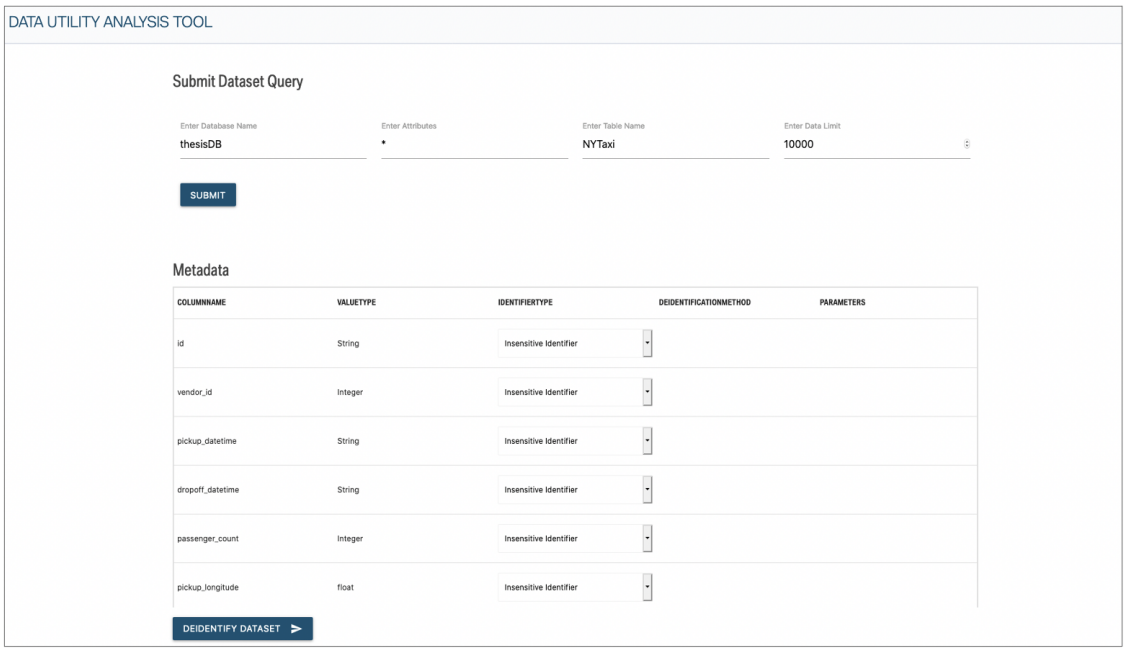


Figure 7.5.: Loading of Dataset and Metadata

De-identify Dataset

Once the dataset is loaded into the application, the user is provided with the metadata comprising of the de-identification form. After specifying the identifier type and de-identification technique for the attributes, the user submits a data de-identification request. On submission, a POST request is performed on the '/deidentify' API endpoint. This triggers the lambda function D, which handles the de-identification request by responding back to the application with the message 'de-identification request submitted' and asynchronously invoking the lambda function E, responsible for de-identifying the dataset. Like the previous lambdas function, lambda function E simultaneously updates the status of the de-identification request in the request status table.

The lambda function E reads the dataset that needs to be de-identified from the DUT S3 bucket. Then based on the user's request, the lambda function de-identifies this dataset, stores this dataset back in DUT S3 bucket and updates the de-identification status of the request. Upon getting notified about the de-identification status through the pull notification mechanism that triggers the GET request of '/queryStatus' endpoint, the

application then triggers the GET request at '/dataset' endpoint to fetch the de-identified dataset. This GET request invokes the lambda function C, which gets the 20 records of the de-identified dataset and sends it as the response to the request. The application then displays this de-identified dataset. Figure 7.6 depicts the de-identification process of NYTaxi dataset in the application.

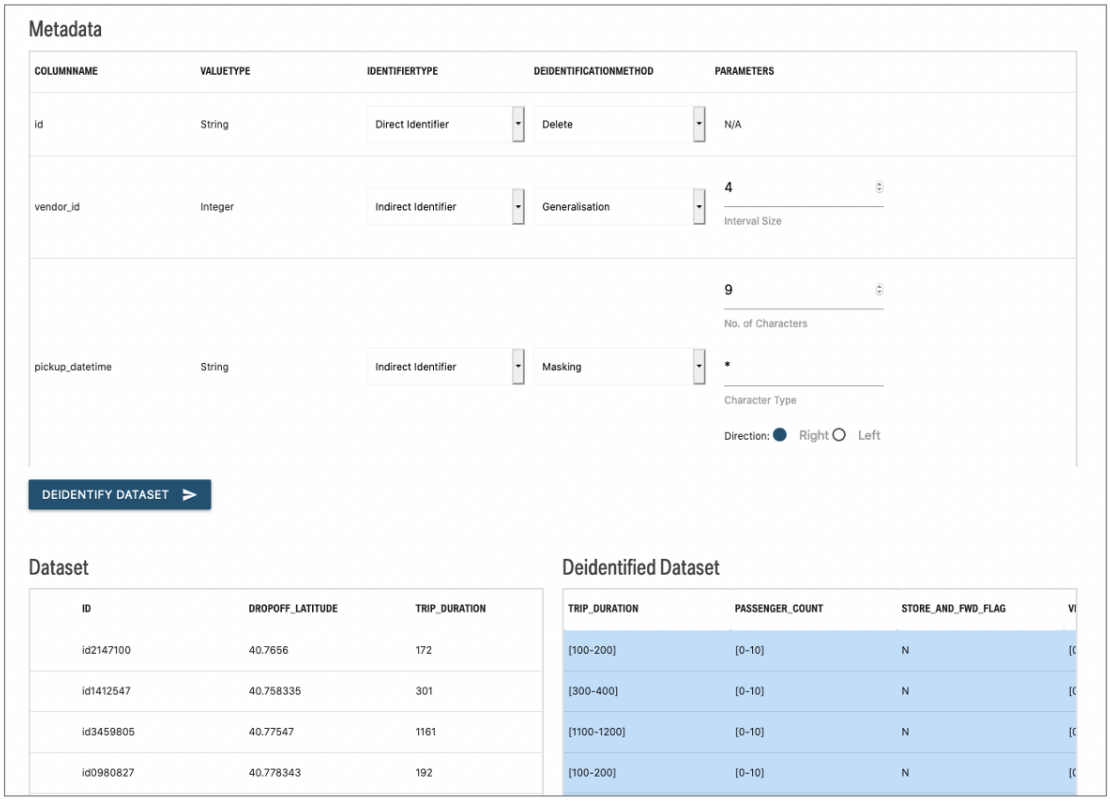


Figure 7.6.: Data De-identification

Analyse Data Utility

After the de-identification of the data, the user can analyse the information loss through various data utility metrics provided in the application. Each utility metric request triggers the POST method of '/utility' API endpoint. This further invokes the lambda function G, responsible for executing the utility metric logic. The lambda function reads both original and de-identified datasets from the DUT S3 bucket and then performs the utility analysis. The result of the analysis is then sent as the response to the request. The result is then visualized in the front-end. Figure 7.7 depicts the utility analysis view of the application.

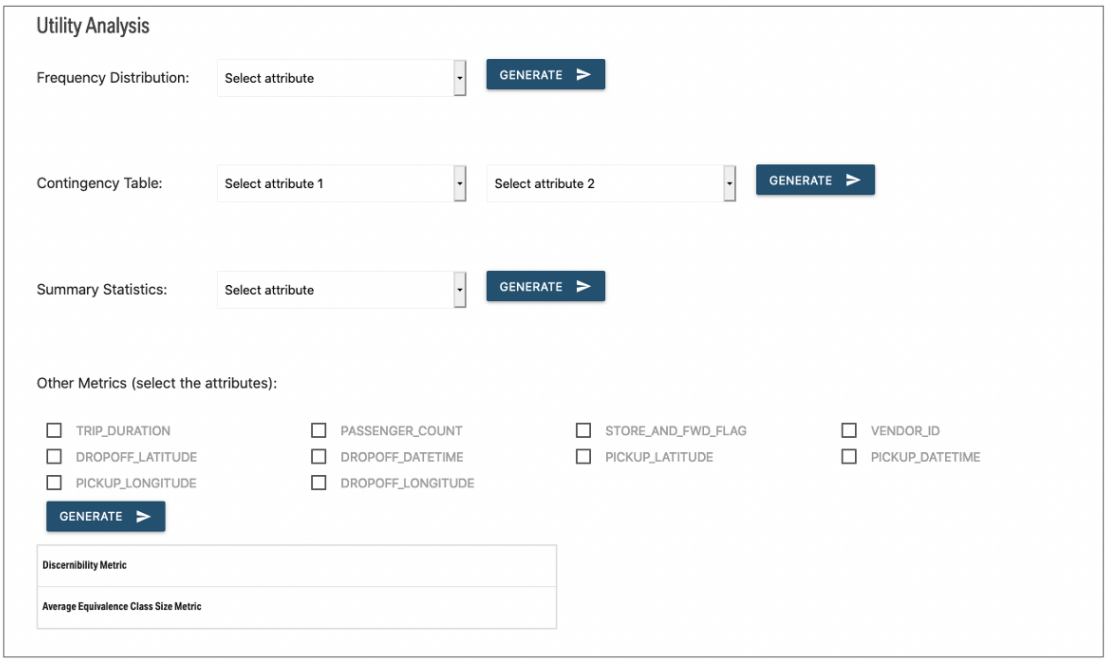


Figure 7.7.: Utility Analysis

8. Evaluation of Data Utility Analysis Tool

In this chapter, the data utility analysis tool is evaluated against various criteria. First, the tool is evaluated based on the application requirements as discussed in section 7.1. Then, the tool is tested with the real automotive dataset provided by industry partner. The final evaluation of the tool is conducted based on the feedback received during the demonstration of the tool to the experts in the software and privacy domain.

8.1. Evaluation based on Application Requirements

One of the goal of this thesis was to built an enterprise level application. Keeping this in mind, the functional and non functional for the data utility analysis tool as discussed in section 7.1 were derived. This section lists those requirements and describes how the tool fulfills them.

8.1.1. Non Functional Requirements

1. *The tool should be able to process large amounts of data.*

   The data utility analysis tool is developed in a way that it can support the processing of large datasets. One can simply increase the memory size of lambda functions to support datasets of larger size. The maximum memory size that could be allocated to a lambda function is 10240MB. The table below represent the time required to process the dataset of size 200MB.

Size of dataset	200MB
Total number of rows	1,458,644
Total time to fetch dataset from S3	13 seconds
Total time to de-identify entire dataset	50 seconds
Memory Size of lambda function	2048MB

Table 8.1.: Time required to process 200MB of dataset

2. *The tool should offer an easy to use interface.*

   The data utility analysis tool is developed as a single-page application. The entire process starting from loading of the dataset to utility assessment is performed

on this single page, making sure that all the relevant information in regards to de-identification is right in front of the eyes. Additionally, the datasets and metadata are displayed in neatly organized tables. The results of utility metrics are visualized through interactive charts.

8.1.2. Functional Requirements

1. *The application should support data to be uploaded from the cloud storage.*

   Since Amazon S3 serves as the primary storage for the datasets, the application supports the uploading of the dataset from the S3 cloud storage. To access these protected datasets, the application only requires the necessary permission to access the S3 bucket containing these datasets. The application then queries these datasets from the S3 bucket using Athena. It is assumed that the tables for the datasets are defined in the Athena. Then the user can query this dataset from the application by providing the database name where the dataset's table resides, table name, the attributes, and the number of records of the dataset. This queried dataset is then stored in another S3 bucket for future processing and a part of this dataset is rendered in the application.

2. *Data de-identification and utility analysis should be done on the server-side.*

   To prevent the exposure of anonymization logic as well as the faster processing of the data, it was recommended to perform the de-identification and the utility analysis in the back-end. The data utility analysis tool supports this requirement by implementing all the de-identification techniques and the utility metrics in the form of lambda functions in the back-end. These lambda functions get invoked whenever the user submits the de-identification or utility analysis request. They then are responsible for reading and processing the dataset from the S3 bucket to apply the user's selected de-identification techniques or the utility metric.

3. *The data should not be allowed to be downloaded.*

   The data utility analysis does not have the functionality to download the data. After de-identification, the de-identified data is stored in the S3 bucket. Only people who have access rights to that bucket can access the de-identified data. This protects the dataset and prevents any re-identification attack from occurring. Additionally, at the most 20 records of the dataset is displayed in the application and not the entire dataset.

4. *The de-identified data should be stored in the cloud.*

   As explained above, the de-identified data is stored in S3 cloud storage and only people having permission to the S3 bucket can access it.

8.2. Evaluation based on Test with Real Automotive Dataset

For fair evaluation of the tool and to get unbiased feedback, the application was tested with real automotive datasets provided by the industry partner. This section briefly describe the application deployment experience in industry partner's application development environment.

To access these datasets, the application had to be integrated into the application development environment of the industry partner. With some added security measures that are enterprise dependent, the application was successfully deployed in the industry partner's development environment. Using the CDK script, the application was easily deployed using just one command. After getting the required permission to access the datasets, the application was successfully able to load real automotive datasets in the data utility analysis tool. Subsequently, data de-identification and utility assessment could be performed on the dataset.

This proves that the data utility analysis tool can be effectively deployed in any development environment and be ready to use without any additional changes to the code provided the application has the necessary access rights.

8.3. Evaluation based on the Feedback of Application Testing

The application was demonstrated to a group of privacy experts and the software developers at the industry partner. Multiple meetings were conducted to get a comprehensive feedback on the data utility analysis tool. The feedback comprised of discussions on various aspects of the application such as the use cases and scenarios the tool can fit in, future enhancements as well as the usability of the application. Additionally some potential research topics were also derived from this feedback. We discuss the feedback below by categorizing it into 4 categories *use cases of the application*, *suggested enhancements*, *potential research topics* and *application usability*.

8.3.1. Use cases of the Application

The major goal of data utility analysis is to help privacy experts to quantify the quality and optimize the application of de-identification techniques on dataset. In addition to this goal, two potential use cases for the tool were discovered during the feedback session.

1. *The tool can help to define data de-identification thresholds and standards.*

   The tool can be used to conduct data de-identification experiments with various automotive datasets. Through the use of utility metrics, one can define standards and rules for the application of data de-identification techniques on datasets. For example: the geo-coordinates should be rounded off to base 3 and not less. Or the

speed should be generalised with minimum interval size of 50. These standards make sure that in the goal of achieving acceptable or high data utility, the privacy requirements are not violated.

2. *The tool can be used to build a public database of de-identified datasets*

   Often requesting datasets for projects through privacy department is a time consuming process as described in section 5.3. At times the datasets are required for quick research analysis or experimental projects. This tool can assist in building a public repository of safe to use de-identified datasets having low risk of re-identification to speed up the dataset access process. For example, the datasets could be de-identified in a way that they are strongly de-identified but have average data utility.

8.3.2. Suggested Enhancements

Through the demonstration of tool, some potential features were discovered that could be added to the data de-identification tool in order to enrich the data de-identification process.

1. *The application should support generalisation for the date/time data type.*

   Currently the application supports integer only generalisation. However it was suggested to have a date data/time type generalisation as it is one of the most common data type. For example, the generalisation should convert any given date to week, month or year. Of course the application should take into account different types of available date formats.

2. *The application should display description and units of attributes of a dataset*

   Considering the vast variety of datasets, often it is not clear what the attribute means in the context of the dataset. This missing information can sometimes cause either improper de-identification or can lead to ignoring of critical and risky attributes. Additionally, information regarding the unit of the attribute is essential to understand the data values. For example, whether mileage is represented in Kms, miles or nautical miles. Both description and units are required for effective de-identification otherwise de-identification is carried out on just a guess.

3. *The application should automatically classify direct identifiers in a dataset*

   It would be a great enhancement if the data utility tool can automatically identify whether an attribute is a direct identifier or not in a dataset.

4. *The application should support caching of dataset to reduce latency and costs*

   Currently, the application reads and writes the dataset directly from the cloud storage. Moreover, for every change in de-identification technique, the entire dataset

is de-identified again. This results in higher processing time and increased costs if using external cloud provider services [79]. Caching of the dataset that allows to handle every change in de-identification technique until the de-identification is finalized could help to reduce the latency and costs resulting in increased application response time.

8.3.3. Potential Research Topics

The feedback discussion on the data utility analysis tool lead to the identification of two potential research topics in the field of data utility assessment and data de-identification.

1. *Determination of acceptable levels for data utility metrics*

   As discussed before, the data utility metrics quantify the information loss in the de-identified dataset caused due to de-identification process in regards to a particular goal [13]. For example, the discernibility metric measures the indistinguishability introduced in the dataset after de-identification. However, what is the acceptable level or threshold of this metric? What is considered a good metric score or the minimum level below which the data utility becomes useless? This research question deals with defining a scientific scale for each utility metric that can help privacy experts to determine the acceptable level for the utility metrics.

2. *Automatic classification of attributes in a dataset based on their identifier types*

   An identifier is an attribute or a set of attributes that can uniquely identify the data subject in a dataset [8]. As discussed in subsection 2.1.2, direct identifiers can alone identify an individual whereas indirect identifiers are the attributes that can in combination with other attributes can lead to identification. This research question deals with determining ways to automatically identify direct and indirect identifiers in a dataset.

8.3.4. Application Usability

In terms of usability, the data utility tool was well perceived. We describe the application's usability feedback in regards to following factors:

- *Intuitive design*: The application was appreciated for its easy to use simple user interface. The data utility analysis tool is built as a single page web application. In terms of design, the application was mostly liked for displaying all the necessary information one is working with in a neat and concise manner.
- *Ease of learning*: The data de-identification and utility analysis process performed in the application was easily grasped by even the non-privacy experts. The application performs this process in a sequence of steps which prevents the user from getting overwhelmed with the information in the beginning.

- *Efficiency of use*: The application is built upon AWS architecture that enables the user to quickly accomplish the de-identification task. So far one can effectively de-identify all the attributes of a dataset of size 200MB in just under 1 minute. The application code can always be optimized more to faster process the data.
- *Memorability*: The intuitive design of the application enables any user to effectively use the application without any external assistance.
- *Error frequency*: Currently, the application supports basic client-side error handling in regards to user inputs or server issues. The application still needs to implement server-side error handling. For example, in case the dataset doesn't exist, or if the parameters of de-identification techniques are wrong, the user should be adequately informed.
- *Subjective Satisfaction*: Overall the application was well accepted in terms of functionality, implementation as well as design.

9. Conclusion

In the final chapter, we summarize the results of this thesis by answering the research questions that were proposed in section 1.3. Additionally, we discuss the limitations that were faced in developing the data utility analysis tool and the future work related to this thesis.

9.1. Summary and Thesis Results

The goal of this thesis was to enhance the data de-identification process through the use of data utility metrics. To achieve this goal, development of a data utility analysis tool was proposed. Through the development of this tool, the thesis aimed to answer the three research questions that were proposed as a part of the thesis. In this section, the answer to these research questions are presented along with the contribution of the thesis.

RQ 1. What is the state-of-the-art of data utility metrics and data de-identification tools?

Through an extensive literature review, we identified and compiled a list of various available data utility metrics, prescribed in scientific literature. These utility metrics that quantifies the loss of information with regards to a specific goal after the de-identification process [13] are presented in section 2.5. Few of these metrics are implemented in the data utility analysis tool. In addition to the utility metrics, the concepts relating to data de-identification, the de-identification approaches and standards and the available de-identification techniques are discussed, as they set the theoretical foundation of the data utility analysis tool. Examples are presented for each de-identification technique to understand their working. Also, various frameworks and tools based on data de-identification are discussed in section 3.3. Some of these tools are open source and emerging from scientific and statistical communities, while others are commercially available tools. One of the most renowned and highly comprehensive tool is the ARX [30] data anonymization tool. The utility metrics implemented in data utility analysis tool has been implemented using the ARX API [80]. Most of these tools implemented a limited set of de-identification techniques such as suppression and generalisation and were highly privacy model and risk assessment focused. This limitation is addressed in the data utility analysis tool, as it is more data de-identification-centric implementing various data de-identification techniques.

RQ 2. How could the implementation of an enterprise level data utility analysis tool look like?

The thesis aimed to built an enterprise level data utility analysis tool. The tool combined the process of data de-identification with that of data utility analysis to capture the implications of data de-identification process and improve the application of de-identification techniques. To built this tool, semi-structured interviews were conducted to understand the de-identification process at the industry partner and to identify the requirements of the data utility application tool. The description of the privacy process and the overview of a typical data de-identification process is discussed in great detail in chapter 5. Also, limitations with this de-identification process were discovered. One major limitation was the rule based de-identification without utility assessment that could either result in privacy loopholes or cause extra information loss. Based on these limitations, the design of the data utility analysis tool was proposed that is discussed in chapter 6. The functional and non functional requirements of the data utility analysis tool that caters to the needs of an enterprise level application are listed in chapter 7 followed by the application's architecture. The architecture of the data utility analysis is designed in a way that the tool is not just industry partner requirements specific but rather that it could be used in any enterprise setting. Even though any technology could be adopted to develop the tool, the back-end and front-end of tool is based on AWS architecture and ReactJS respectively. Both of these technologies are widely known, often used in industries and backed by huge developers community. The technical architecture based on these technologies and the application workflow is presented in section 7.3.

RQ 3. Given the feedback during the application demonstration, in what ways could the tool be improved?

The formal evaluation of the data utility analysis tool was carried out by demonstrating it to the experts in privacy and software domain. Multiple meetings were conducted and a thorough feedback was collected. In chapter 8, this feedback is discussed in a comprehensive manner. From this feedback, list of ways data utility tool could be enhanced, the additional use cases of the application and potential research topics were identified. In addition to minor enhancements such as the support of data/time generalisation in the tool, it was discovered that the descriptions and units of attributes of a dataset play an important role in effective de-identification of the data as sometimes the meaning of attributes are not clear resulting in improper de-identification. So to enrich the de-identification process, descriptions of the attributes of dataset should be displayed in the application. To reduce latency and costs associated with processing of data in the data utility analysis tool, caching of dataset was recommended. Finally, automatic identification of direct identifiers in a dataset was suggested to improve the de-identification process.

9.2. Limitations of the Data Utility Analysis Tool

In this section we discuss the two limitations of the implemented data utility analysis tool.

Firstly, due to lack of time not all data de-identification techniques presented in chapter 2 are implemented in the tool. Currently, the de-identification techniques the tool supports are *integer generalisation, suppression, truncation, rounding* and *character masking*. Similarly only limited utility metrics are supported by the tool that includes *frequency distribution, contingency table, summary statistics, discernibility* and *average equivalence class size metric*. As discussed in subsection 2.5.2, other advanced utility metrics could not be implemented because they are based on the concept of domain generalisation hierarchy which is not supported by the application. Creation of these generalisation hierarchy is one of the future works of the application.

Secondly, in the data analysis tool whenever the de-identification technique of an attribute is altered (through change in parameter or a different de-identification technique), all the attributes of the dataset are de-identified again. This is not optimal in longer run as it results in higher latency and increased costs due to direct read and write operations in the cloud storage. De-identification should be performed in a way that not all the dataset is de-identified again for every change but only the altered attribute. Caching of the dataset until the de-identification is finalised is one solution that was recommended in the feedback session. Caching prevents the direct reading and writing of dataset in the cloud storage. The code could be optimised to only de-identify the attribute for which the de-identification technique is altered.

9.3. Future Work

The limitations of the data utility analysis and the suggestions of the feedback session proposed ideas for application extension and research topics. These ideas are briefly discussed in this section.

Firstly, all the de-identification techniques and utility metrics should be implemented in the data utility analysis tool to provide the user with a wide selection. Similarly the concept of domain generalisation hierarchy [36] should be implemented in the tool to support other utility metrics. Additionally, as suggested in the feedback, the tool should support the generalisation of date/time data type.

Secondly, to enrich the de-identification process, the application can be extended to display the description and units of attributes in the dataset. This description would help the privacy experts to understand the meaning of each attribute which would help in effective de-identification of the attributes. Additionally, the data utility tool can enable the caching of dataset to reduce latency and costs.

Thirdly, two potential future research topics were discovered during the feedback session. One of the topic is regarding defining a scientific scale that can help to determine the acceptable level or a good metric score for the utility metrics. The other research question is about determining ways to automatically classify direct identifiers in a dataset.

Finally, the data utility analysis tool sets the basis for developing advanced privacy models [13]. Currently, efforts are being made to integrate the k-anonymity algorithm implemented by Sharada Sowmya in her master thesis *"Implementation of K-Anonymity for the Use Case of Automotive Industry in Big Data Context"* [81] with the tool. The utility metrics implemented in the tool could help in bench-marking the performance of the privacy models and their algorithms.

A. Tutorial of Data Utility Assessment Tool

In this chapter, we provide a step by step guide on how to de-identify dataset and assess the data utility in the data utility assessment tool.

1. *Load the dataset in the data utility analysis tool that needs to be de-identified.*

   To load the dataset, specify the name of the Athena database that consists the dataset, the attributes of the dataset (in a form of comma separated list or '*' in case all attributes are required), the name of the table that contains the dataset and the number of records that need to be de-identified. And click on 'Submit' button as depicted in Figure A.1. The name of the database is 'bhawnathesisdb', to select all the attributes in the dataset we use '*', the name of the table is 'trains' and the number of records are '100'.

Figure A.1.: Loading dataset that needs to be de-identified

   Upon loading the dataset, metadata and the dataset would be displayed in the application as depicted in Figure A.2 and Figure A.3. Only 20 records of the dataset are displayed and not the entire dataset.

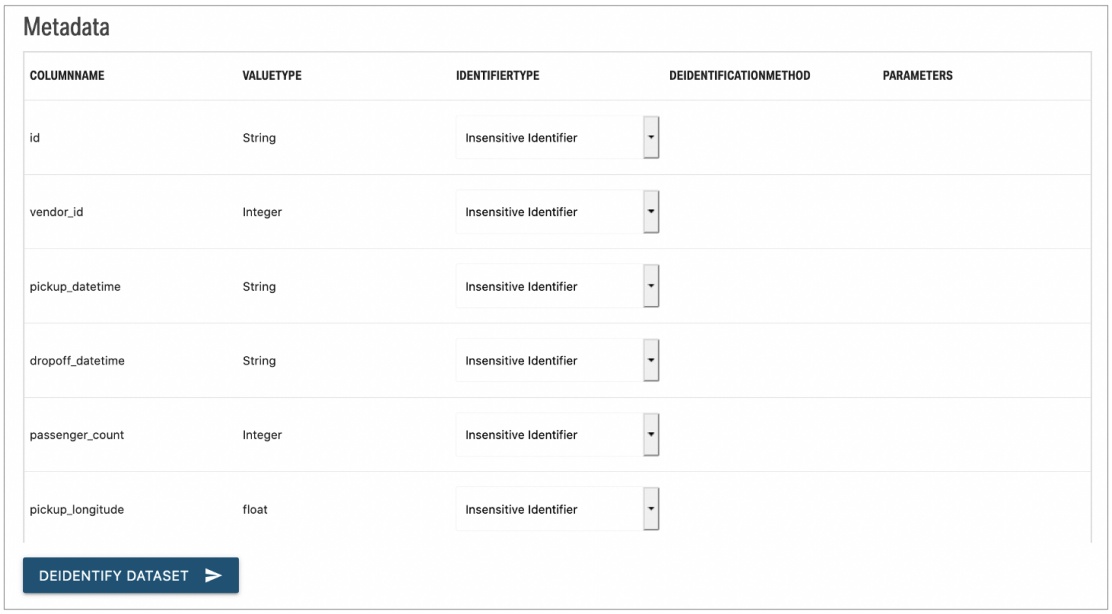


Figure A.2.: Metadata of the dataset

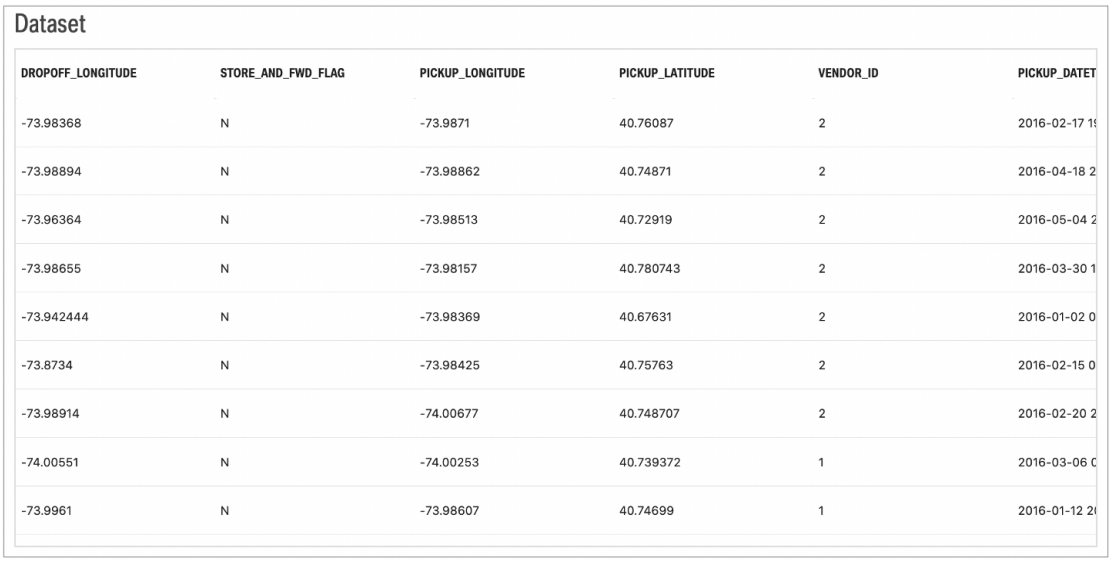


Figure A.3.: Dataset that needs to be de-identified

2. *Select the identifier type for each attribute and its corresponding de-identification technique*

   In the metadata form, select the identifier type of the attribute and then select a de-identification technique as depicted in Figure A.4. Based on the identifier type, de-identification techniques are suggested in the application [8]. By default, the

identifier type is 'Insensitive identifier' for which no de-identification techniques are suggested. The recommendations of de-identification techniques are depicted in Table A.1. After selecting the identifier type and de-identification techniques for each attribute, click 'Deidentify Dataset'.

Identifier Type	**De-identification Techniques**
Insensitive	-
Sensitive	Delete, Generalisation, Truncate, Masking, Rounding
Direct	Delete
Indirect	Generalisation, Truncate, Masking, Rounding

Table A.1.: Identifiers and suggested de-identification techniques

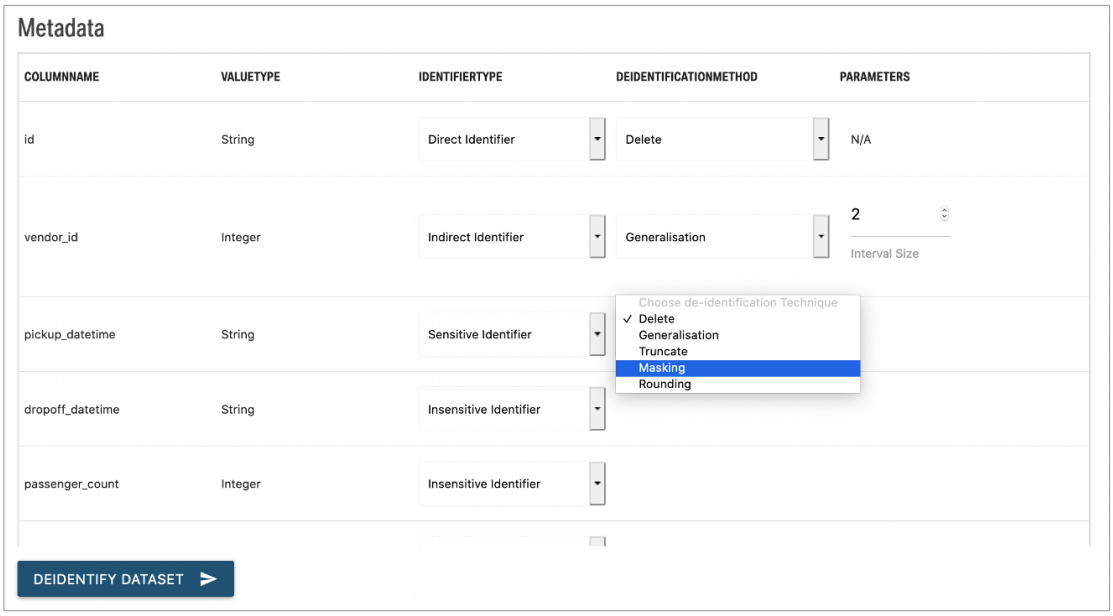


Figure A.4.: Selecting the identifier type and de-identification technique

The de-identification process takes a few seconds depending on the number of rows selected. After the de-identification, both the original and the de-identified dataset are displayed as depicted in Figure A.5.

3. *Perform the data utility assessment using the available utility metrics*

   To perform utility analysis, the application allows the user to generate frequency distribution, contingency table and summary statistics. In addition to this, the application supports discernibility metric and average equivalence class size metric. Discernibility and average equivalence class size metric generate a score between 0 and 1. A higher score represents that the utility in regards to goal of these

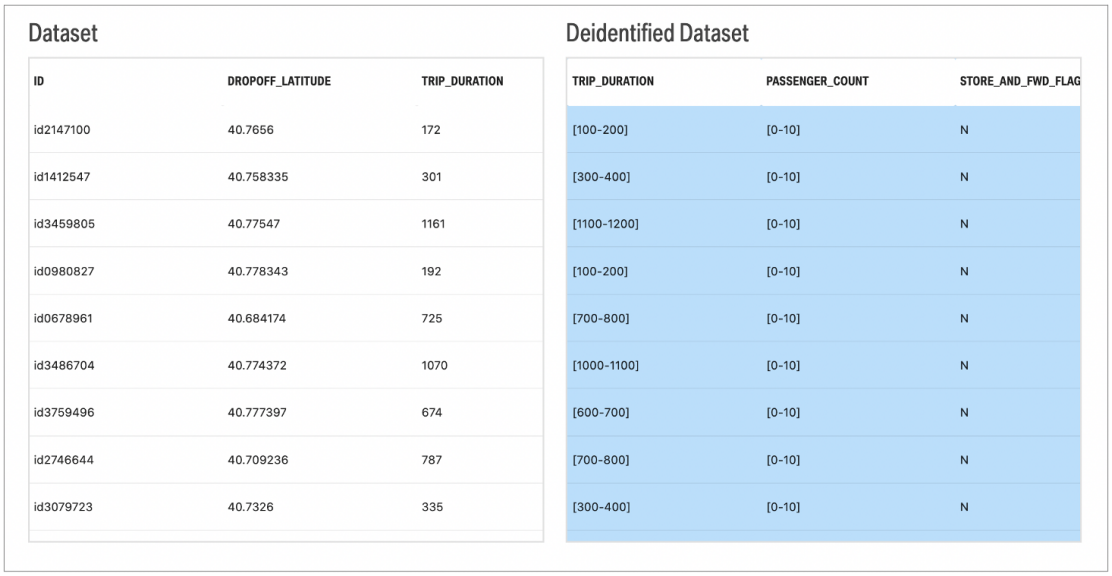


Figure A.5.: De-identified Dataset

metrics is preserved and lower score indicates that the utility is affected. For eg: a discernibility metric score of 1.0 means that no indistinguishability is introduced in the dataset after de-identification.

The utility analysis module of the application is depicted in Figure A.6. In the subsequent Figure A.8, Figure A.7, Figure A.9, utility analysis is performed for some attributes.

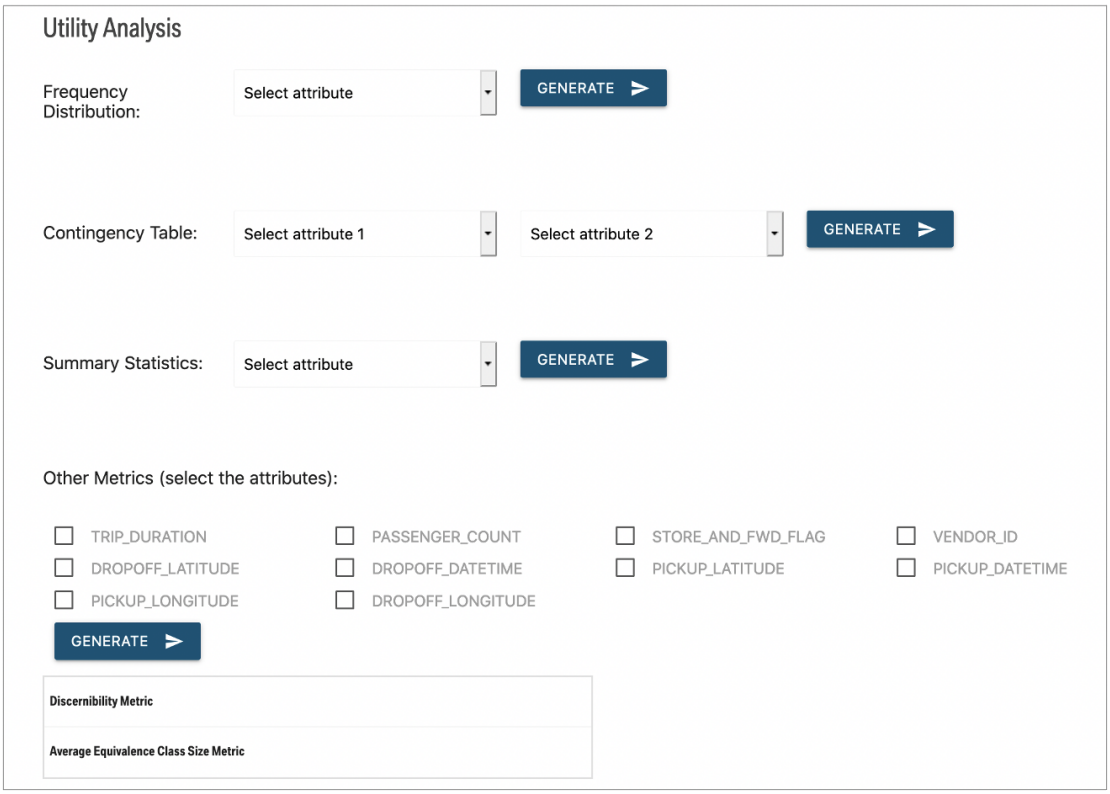


Figure A.6.: Utility Analysis supported by the application

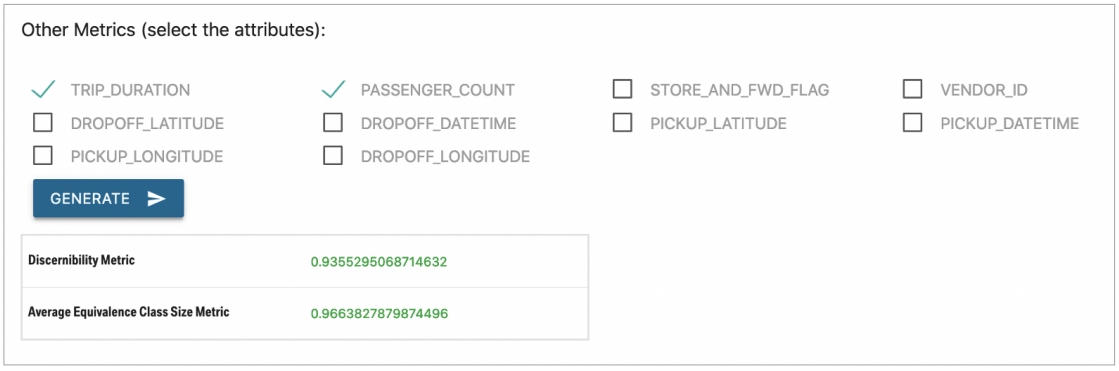
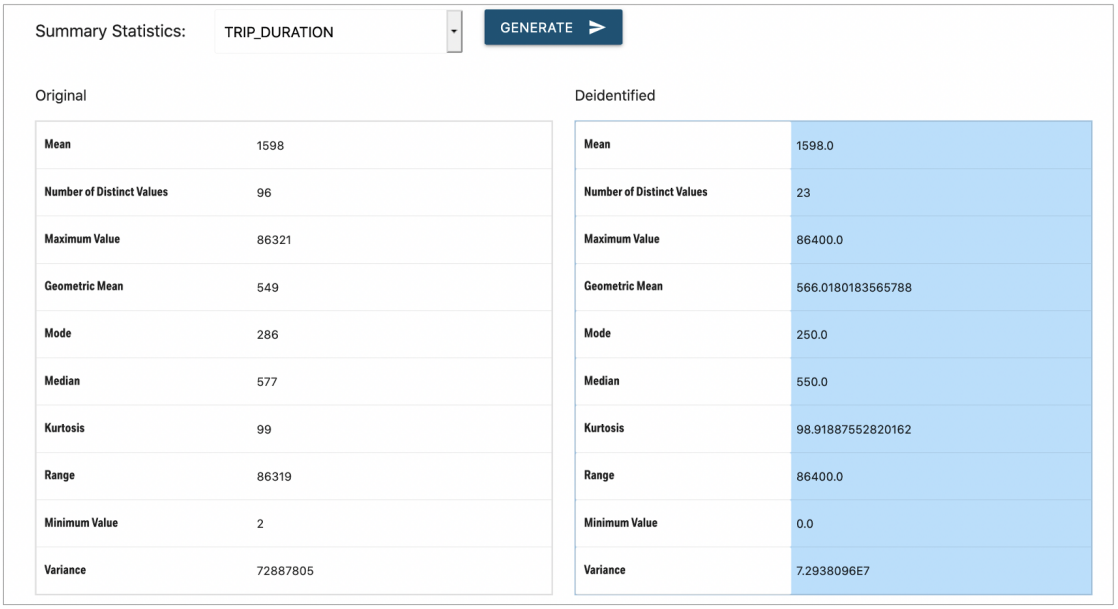


Figure A.7.: Discernibility and Normalised equivalence Class Size Metric

Figure A.8.: Frequency Distribution and Contingency Table



Original		Deidentified	
Mean	1598	Mean	1598.0
Number of Distinct Values	96	Number of Distinct Values	23
Maximum Value	86321	Maximum Value	86400.0
Geometric Mean	549	Geometric Mean	566.0180183565788
Mode	286	Mode	250.0
Median	577	Median	550.0
Kurtosis	99	Kurtosis	98.91887552820162
Range	86319	Range	86400.0
Minimum Value	2	Minimum Value	0.0
Variance	72887805	Variance	7.2938096E7

Figure A.9.: Summary Statistics

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 29

API Application Programming Interface. 46

AWS Amazon Web Services. 46, 48

CCPA California Consumer Privacy Act. 29

CDK Cloud Development Kit. 48

GDPR General Data Protection Regulation. 29

HIPAA Health Insurance Portability and Accountability Act. 8, 30

IoT Internet of Things. 29

PIPEDA Personal Information Protection and Electronic Documents Act. 29

S3 Simple Storage Service. 46

Bibliography

[1] P. T. AG. *GDPR*. https://gdpr.eu/. 2020.

[2] S. of California Department of Justice Office of Attorney General. *CCPA*. https://oag.ca.gov/privacy/ccpa. 2020.

[3] O. of The Privacy Commissioner of Canada. *PIPEDA*. https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/. 2020.

[4] K. Baumer. "Identification and Evaluation of Concepts for Privacy-Enhancing Big Data Analytics Using De-Identification Methods on Wrist-Worn Wearable Data". MA thesis. Technische Universität München, 2020.

[5] G. Bondel, G. M. Garrido, K. Baumer, and F. Matthes. "The Use of De-identification Methods for Secure and Privacy-enhancing Big Data Analytics in Cloud Environments." In: *ICEIS (2)*. 2020, pp. 338–344.

[6] S. Ribaric, A. Ariyaeeinia, and N. Pavesic. "De-identification for privacy protection in multimedia content: A survey". In: *Signal Processing: Image Communication* 47 (2016), pp. 131–151.

[7] S. L. Garfinkel. "De-identification of personal information". In: *National institute of standards and technology* (2015).

[8] I. 20889:2018. *Privacy enhancing data de-identification terminology and classification of techniques*. Standard. International Organization for Standardization, 2018.

[9] G. Bondel, G. M. Garrido, K. Baumer, and F. Matthes. "Towards a Privacy-Enhancing Tool Based on De-Identification Methods." In: *PACIS*. 2020, p. 157.

[10] T. E. D. P. Board. *Guidelines 1/2020 on processing personal data in the context of connected vehicles and mobility related applications*. https://edpb.europa.eu/sites/default/files/consultation/edpb_guidelines_202001_connectedvehicles.pdf. 2020.

[11] T. E. D. P. Supervisor. *TechDispatch 3: Connected Cars*. https://edps.europa.eu/data-protection/our-work/publications/techdispatch/techdispatch-3-connected-cars_en. 2019.

[12] G. Kermorgant and M. Guilbot. "The GDPR and Its Application in IoT and Connected Cars Opportunities for Business and Competitivity". In: *Electronic Components and Systems for Automotive Applications*. Springer, 2019, pp. 255–267.

[13] O. Tomashchuk, D. Van Landuyt, D. Pletea, K. Wuyts, and W. Joosen. "A data utility-driven benchmark for de-identification methods". In: *International Conference on Trust and Privacy in Digital Business*. Springer. 2019, pp. 63–77.

[14] P. J. Pratt and M. Z. Last. *Concepts of database management*. Cengage Learning, 2014.

[15] D. G. Altman. *Practical statistics for medical research*. CRC press, 1990.

[16] S. Valcheva. *6 Types of Data in Statistics and Research: Key in Data Science*. `http://www.intellspot.com/data-types/`.

[17] QuestionPro. *Interval scale Vs Ratio scale. What is the difference?* `https://www.questionpro.com/blog/ratio-scale-vs-interval-scale/`. 2020.

[18] L. Baker. *4 Types of Data*. `https://www.chi2innovations.com/blog/discover-data-blog-series/data-types-101/`. 2020.

[19] K. El Emam. *Guide to the de-identification of personal health information*. CRC Press, 2013.

[20] Information and P. C. of Ontario. *De-identification Guidelines for Structured Data*. `https://www.ipc.on.ca/wp-content/uploads/2016/08/Deidentification-Guidelines-for-Structured-Data.pdf`. 2016.

[21] K. El Emam and B. Malin. "Appendix B: Concepts and methods for de-identifying clinical trial data". In: *Sharing clinical trial data: Maximizing benefits, minimizing risk* (2015), pp. 1–290.

[22] I. 25237:2017. *Health informatics — Pseudonymization*. Standard. Geneva, CH: International Organization for Standardization, 2017.

[23] O. for Civil Rights (OCR). *The HIPAA Privacy Rule*. `https://www.hhs.gov/hipaa/for-professionals/privacy/index.html`. 2020.

[24] O. for Civil Rights (OCR). *Guidance Regarding Methods for De-identification of Protected Health Information in Accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule*. `https://www.hhs.gov/hipaa/for-professionals/privacy/special-topics/de-identification/index.html`. 2015.

[25] J. Domingo-Ferrer. "Microaggregation for database and location privacy". In: *International workshop on next generation information technologies and systems*. Springer. 2006, pp. 106–116.

[26] B. Riedl, V. Grascher, and T. Neubauer. "A secure e-health architecture based on the appliance of pseudonymization." In: *JSW* 3.2 (2008), pp. 23–32.

[27] A. Boldyreva, N. Chenette, Y. Lee, and A. O'neill. "Order-preserving symmetric encryption". In: *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer. 2009, pp. 224–241.

[28] M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. "Format-preserving encryption". In: *International workshop on selected areas in cryptography*. Springer. 2009, pp. 295–312.

[29] B. T. Podgursky. “Practical k-anonymity on large datasets”. PhD thesis. 2011.

[30] ARX. *ARX Data Anonymization Tool*. `https://arx.deidentifier.org/`. 2020.

[31] S. P. Guide. *Measuring Utility and Information Loss*. `https://sdcpractice.readthedocs.io/en/latest/utility.html`. 2019.

[32] N. Sematech. *Measures of Skewness and Kurtosis*. `https://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm`. 2020.

[33] S. H. To. *Geometric Mean: Definition, Examples, Formula, Uses*. `https://www.statisticshowto.com/geometric-mean-2/`. 2021.

[34] Wikipedia. *Frequency distribution*. `https://en.wikipedia.org/wiki/Frequency_distribution`. 2021.

[35] R. J. Bayardo and R. Agrawal. “Data privacy through optimal k-anonymization”. In: *21st International conference on data engineering (ICDE'05)*. IEEE. 2005, pp. 217–228.

[36] S. Fletcher and M. Z. Islam. “Measuring information quality for privacy preserving data mining”. In: *International Journal of Computer Theory and Engineering* 7.1 (2015), p. 21.

[37] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. “Mondrian multidimensional k-anonymity”. In: *22nd International conference on data engineering (ICDE'06)*. IEEE. 2006, pp. 25–25.

[38] J.-W. Byun, Y. Sohn, E. Bertino, and N. Li. “Secure anonymization for incremental datasets”. In: *Workshop on secure data management*. Springer. 2006, pp. 48–63.

[39] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. “Incognito: Efficient full-domain k-anonymity”. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 2005, pp. 49–60.

[40] J. Eicher, K. A. Kuhn, and F. Prasser. “An experimental comparison of quality models for health data de-identification”. In: *MEDINFO 2017: Precision Healthcare through Informatics*. IOS Press, 2017, pp. 704–708.

[41] J. Goldberger and T. Tassa. “Efficient anonymizations with enhanced utility”. In: *2009 IEEE International Conference on Data Mining Workshops*. IEEE. 2009, pp. 106–113.

[42] D. Kifer and J. Gehrke. “Injecting utility into anonymized datasets”. In: *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. 2006, pp. 217–228.

[43] V. S. Iyengar. “Transforming data to satisfy privacy constraints”. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 279–288.

[44] F. Prasser, J. Eicher, H. Spengler, R. Bild, and K. A. Kuhn. “Flexible data anonymization using ARX—Current status and challenges ahead”. In: *Software: Practice and Experience* 50.7 (2020), pp. 1277–1304.

[45] C. D. Group. *CAT Cornell Anonymization Toolkit*. https://sourceforge.net/projects/anony-toolkit/. 2014.

[46] U. D. D. Security and P. Lab. *UTD Anonymization ToolBox*. http://www.cs.utdallas.edu/dspl/cgi-bin/toolbox/index.php. 2012.

[47] C. Dai, G. Ghinita, E. Bertino, J.-W. Byun, and N. Li. "TIAMAT: a tool for interactive analysis of microdata anonymization techniques". In: *Proceedings of the VLDB Endowment* 2.2 (2009), pp. 1618–1621.

[48] G. Poulis, A. Gkoulalas-Divanis, G. Loukides, S. Skiadopoulos, and C. Tryfonopoulos. "SECRETA: A system for evaluating and comparing relational and transaction anonymization algorithms". In: (2014).

[49] OpenAIRE. *Amnesia*. https://amnesia.openaire.eu/. 2020.

[50] S. D. Control. *-ARGUS*. http://research.cbs.nl/casc/mu.htm. 2018.

[51] sdcMicro: Statistical Disclosure Control Methods for Anonymization of Data and R. Estimation. *-ARGUS*. https://cran.r-project.org/web/packages/sdcMicro/index.html. 2020.

[52] A. Hundepool and L. Willenborg. "ARGUS, Software Packages for Statistical Disclosure Control". In: *COMPSTAT*. Springer. 1998, pp. 341–345.

[53] M. Templ. "Statistical disclosure control for microdata". In: *Cham: Springer* (2017).

[54] J. B. Anco Hundepool Peter-Paul de Wolf, A. R. L. F. S. Polettini, r. s. Alessandra Capobianchi (Risk models) Josep Domingo (Numerical micro aggregation, and synthetic data). *μ-ARGUS: User Manual version 5.1*. http://research.cbs.nl/casc/Software/MUmanual5.1.3.pdf. 2014.

[55] B. Vernay. "A Standardized Approach to De-Identification". In: (2016).

[56] IBM. *IBM InfoSphere Optim Data Privacy*. https://www.ibm.com/products/infosphere-optim-data-privacy. 2020.

[57] P. LTD. *Privitar: Enterprise Data Privacy Software*. https://www.privitar.com/. 2020.

[58] P. S. Corporation. *PHEMI*. https://www.phemi.com/data-privacy-management//. 2020.

[59] A. GmbH. *aircloak*. https://aircloak.com/. 2020.

[60] THALES. *THALES: Streamlined Tokenization and Dynamic Data Masking*. https://cpl.thalesgroup.com/encryption/vormetric-application-crypto-suite/vormetric-application-encryption/tokenization-data-masking. 2020.

[61] A. R. Hevner. "A three cycle view of design science research". In: *Scandinavian journal of information systems* 19.2 (2007), p. 4.

[62] N. M. Ondrej Burkacky and L. Pautasso. *Will quantum computing drive the automotive future?* https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/will-quantum-computing-drive-the-automotive-future. Last accessed 1 July 2021. 2020.

[63] D. Alvarez-Coello, D. Wilms, A. Bekan, and J. M. Gómez. "Towards a data-centric architecture in the automotive industry". In: *Procedia Computer Science* 181 (2021), pp. 658–663.

[64] Otonomo. *A Privacy Playbook for Connected Car Data*. https://fpf.org/wp-content/uploads/2020/01/OtonomoPrivacyPaper.pdf. Last accessed 1 July 2021. 2020.

[65] F. Vallet. "The GDPR and Its Application in Connected Vehicles—Compliance and Good Practices". In: *Electronic Components and Systems for Automotive Applications*. Springer, 2019, pp. 245–254.

[66] H. L. Publications. *More data, more risk: The automotive industry rethinks its privacy strategies*. https://www.hoganlovells.com/en/publications/the-automotive-industry-rethinks-its-privacy-strategies. 2020.

[67] M. Hintze and K. El Emam. "Comparing the benefits of pseudonymisation and anonymisation under the GDPR". In: *Journal of Data Protection & Privacy* 2.2 (2018), pp. 145–158.

[68] HITRUST. *A Consistent Methodology for the De-Identification of Data*. https://hitrustalliance.net/product-tool/de-identification/. 2018.

[69] K. E. Todt. "Data Privacy and Protection". In: *The Cyber Defense Review* 4.2 (2019), pp. 39–46.

[70] D. Garlan and D. E. Perry. "Introduction to the special issue on software architecture". In: *IEEE Trans. Software Eng.* 21.4 (1995), pp. 269–274.

[71] A. Gemino and D. Parker. "Use case diagrams in support of use case modeling: Deriving understanding from the picture". In: *Journal of Database Management (JDM)* 20.1 (2009), pp. 1–24.

[72] F. O. Source. *React*. https://reactjs.org/. Last accessed 29 June 2021. 2021.

[73] A. W. Services. *Amazon API Gateway*. https://aws.amazon.com/api-gateway/. Last accessed 29 June 2021. 2021.

[74] A. W. Services. *Amazon S3 FAQs*. https://aws.amazon.com/s3/faqs/. Last accessed 29 June 2021. 2021.

[75] A. W. Services. *What is Amazon Athena?* https://docs.aws.amazon.com/athena/latest/ug/what-is.html. Last accessed 29 June 2021. 2021.

[76] A. W. Services. *Amazon DynamoDB*. https://aws.amazon.com/dynamodb/. Last accessed 29 June 2021. 2021.

[77] A. W. Services. *AWS Lambda*. https://aws.amazon.com/lambda/. Last accessed 29 June 2021. 2021.

[78] A. W. Services. *AWS Cloud Development Kit*. https://aws.amazon.com/cdk/. Last accessed 29 June 2021. 2021.

[79] J. Wood and H. O. Prasath. *Caching data and configuration settings with AWS Lambda extensions*. https://aws.amazon.com/blogs/compute/caching-data-and-configuration-settings-with-aws-lambda-extensions/. Last accessed 7 July 2021. 2021.

[80] ARX. *API*. https://arx.deidentifier.org/development/api/. Last accessed 7 July 2021. 2021.

[81] S. Sowmya. "Implementation of K-Anonymity for the Use Case of Automotive Industry in Big Data Context". MA thesis. Technische Universität München, 2021.