Radboud University Nijmegen

Faculty of Science

# Detecting privacy sensitive information in large file systems

Thesis project at Centraal Beheer Achmea

Thesis MSc Data Science

*Supervisors:*
Prof. M.A. Larson
Ir B.R. Stienstra

*Author:*
Niels Dekkers, BSc

*Second reader:*
Dr. Ir. F. Hasibi

November 2020

# Contents

## Abstract

As technology progressed a need was recognised for modern protections against misuse of data. In 2016, a new and improved data privacy law was passed, the EU General Data Protection Regulation (GDPR). Centraal Beheer Achmea manages a very large amount of insurance data. Some of the data contains personal client information whereas another part does not. Achmea would like to have tooling that is able to detect privacy sensitive information in large file systems. This method can be used as an additional method of ensuring that the company complies with EU legislation. In order to arrive at a solution, a few questions had to be answered. Which privacy-sensitive information is processed by Achmea? Is information directly traceable to a person or only in combination with other information? And finally, how can this information be searched in a scalable way?

Ultimately, 16 different types of privacy-sensitive information were found and classified as identifiers or quasi-identifiers. Using K-anonymity, a method to achieve differential privacy, it was possible to calculate when quasi-identifiers could be used to identify a specific person. In order to deploy this in a scalable way, iterators were used that can read and process datasets in them. Using this knowledge it is possible to develop a scalable solution that is able to detect privacy sensitive information which are being processed by Achmea. A possible implementation that meet these requirements is discussed. Two important recommendations are proposed. The first recommendation would be by finally improving the proposed solution by implementing it in PySpark. The second recommendation would be combining the results that are produced by the proposed tooling with admitted data governance requests. Combined this information can be used to determine whether processing of found privacy sensitive information present is allowed. If this is not the case, a data governance request can be submitted or the information be removed to ensure compliance with legislation.

## Acknowledgements

I would like to thank the following people for helping with this research project: First my internal supervisor (Prof. M.A. Larson) for providing guidance and feedback throughout this project. Second I would like to thank my external supervisor at Achmea (Ir. B.R. Stienstra) for helping me getting started. She guided me in asking questions to the right people. She also provided feedback on my thesis. Third, I would like to thank E.D. Bos., BSc. Working with him was very pleasant. He acted as client on behalf of Achmea. He set the terms and conditions for my assignment. In addition, all choices were coordinated with him during my research internship. Fourth, I would like to thank the manager of the Data Analytics L. M. Vink, BSc. He contributed by suggesting a suitable project for my thesis and linked me to the right people. During my research internship he stayed on top of my progress and assisted if I needed help. Last, I would like to thank all the other members of Achmea's Data Analytics team members who took the time to answer my questions and have taught me working in an Scrum (Agile) framework and to work with new techniques and platforms.

# 1 Introduction

Data is becoming more and more important. Combining data from various sources in order to obtain information regarding persons becomes more relevant and difficult. The main focus for this thesis is motivated and research questions are defined.

## 1.1 Motivation

As technology progressed and computing was invented, it was recognised the need for modern protections against misuse of data. The first law in the United States to foster information privacy of patients in the healthcare industry (Health Insurance Portability and Accountability Act, or short HIPAA) was enacted by former President Bill Clinton in 1996 [3]. California later introduced later the California Consumer Privacy Act (CCPA) which is a regulation that aims to enhance consumer privacy rights [7]. The first safeguard in the European Union was designed in 1995 when the EU passed the European Data Protection Directive, which established some data privacy [21]. Only recently in 2016, a new and improved data privacy law was passed, the EU General Data Protection Regulation (GDPR). As of May 25, 2018, it is enforced across all EU Member States. This law is a milestine in the European privacy framework. With six general data protection principles in mind and based on the concept that privacy is a fundamental human right, the GDPR will have a global impact [4].

The GDPR has six general data protection principles (fairness and lawfulness; purpose limitation; data minimisation; accuracy; storage limitation; and integrity and confidentiality) [5]. Data protection by design and data protection by default is at the core of the GDPR. Fundamental to GDPR is transparent handling of privacy sensitive information and by not complying with the law you can be held accountable. The implementation of the GDPR in the Netherlands occurred through the Dutch GDPR Implementation Act ('Uitvoeringswet Algemene Verordening Gegevensbescherming', UAVG).

Centraal beheer Achmea manages a very large amount of insurance data. Some of the data contains personal client information whereas another part does not. Achmea uses this data to improve their insurance services to their customers. But, processing data containing privacy sensitive information (PSI) must be in accordance with the UAVG. Managing large amount of data requires multiple big data tools such as SAS in case of Achmea. In order to be able to obtain and process these large amounts of data, it gets uploaded first to a SAS platform.$^{\text{link}}$ Within SAS statistics about the data are being created. However, the platform is getting overloaded due to the growing amount of data. Another problem is that when data of different sources are combined within the platform, information could be obtained that Achmea is not allowed to manage.

To solve these problems they want functionality which is able to detect privacy sensitive information. Because of the enormous amount of data, the solution should be scalable. While it seems logical that existing tooling can be used to detect PSI, this is not the case. After investigation by a SAS administrator, there was currently no tooling found available that is able to detect PSI in multiple file types, especially not for SAS datasets. Another problem is that data managed by Achmea is quite domain-specific. As a consequence the method to scan their data should not only be scalable, but also be specific in the way of detecting PSI.

## 1.2 Challenges

Data protection techniques must prevent each release of data which can be indistinguishably related to less than a certain number of a population of the general population. Regarding privacy, data can be divided in four different categories according to S. De Capitani Di Vimercati [9].

1. Identifiers: attributes that uniquely identify a natural person;
2. Quasi-identifiers: attributes that in combination (and possibly with external information) can be used to identify a natural person;
3. Confidential attributes: attributes that represent privacy sensitive information which can not be used to identify a natural person;
4. Non-confidential attributes: attributes that are not considered as privacy sensitive.

Detecting privacy sensitive information in systems where it should not reside, is urgently needed. There are already several examples where insurance companies were fined for not complying with the UAVG [1][2][22]. Detecting PSI in large file systems poses several challenges. One of the main challenges is the size of individual files. Where some files are only a few megabytes large, there are also files of several hundred megabytes or even a few gigabytes. In addition, different format files read in different ways. A Microsoft Excel (XLSX) file is zipped by itself, and therefore those files cannot be read in chunks. But other files, for example, CSV files do allow reading the file in chunks. Another challenge would be handling the huge amount of files. In large companies, such as Achmea, it often happens that there are several hundreds of thousands or more files on a file system. How do you make a solution scalable to such numbers? The final challenge is about how the data itself is presented in a dataset. Data is often entered in different ways by different people. Which format is used? For example: if something is not known, will the field be left empty or will say ‘n/a’ or maybe even ‘unknown’?

## 1.3 Research questions

These challenges combined result in the main research question of this thesis: How to detect privacy sensitive information in large file systems containing structured data? To be able to answer this question, we need to answer most if not all of the aforementioned challenges. This will be the first sub-research question: How to ensure scalability in a solution scanning large file systems?

Besides these challenges, we need to know which information is regarded as privacy sensitive.A further distinction can be made between the different types of data that are marked as privacy sensitive. Which information falls under identifiers and which information does fall under quasi-identifiers as stated by S. De Capitani Di Vimercati? This will be the second sub-research question: What is the stakeholders definition of privacy sensitive information? Which of those types of information is on its own enough to be seen as PSI and which information needs to be combined with other information to be seen as PSI?

Knowing privacy sensitive information and classifying it as either identifier or quasi-identifier is not enough. Although it is clear that an identifier can be used to identify a person. But it s not clear when a quasi-identifier is able to identify a natural person. How to determine when a combination of quasi-identifiers results in such a small remaining group of persons that they together can be seen as identifiers? For example, knowing an age is not enough to determine a specific individual who is involved. After all, several people were born on the same day. Knowing in which municipality someone lives is also not very specific. Usually a few thousand to a few hundred people live in a municipality. However, if you know both aspects, the group of people can be reduced to a smaller number, after that it may be possible to identify the correct individual. A study from Samarati found that 87 percent of the U.S. population can be identified using a combination of their gender, birth date and zip code [24]. This will be the third sub-research question: which approach can we use to determine when privacy sensitive information can be used to identify a natural person and what would be the limitations of such an approach?

Summarised, the following four question will be addressed in this thesis:

1. Main RQ: How to detect privacy sensitive information in large file systems containing structured data?

2. 1st sub RQ: How to ensure scalability in a solution scanning large file systems?

3. 2nd sub RQ: What is the stakeholders definition of privacy sensitive information? Which of those types of information is on its own enough to be seen as PSI? Which information needs to be combined with other information to be seen as PSI?

4. 3th sub RQ: How to determine when privacy sensitive information can be used to identify a natural person and what would be the limitations of such an approach?

### 1.4 Structure of thesis

The thesis is build up of 6 chapters where in the first chapter the main challenges and research questions are defined. In chapter 2 relevant literature will be discussed. In chapter 3 the approach and proposed solution is explained and in chapter 4 the results are discussed. In chapter 5 the advantages of the proposed solution will be highlighted and a conclusion will be made whether the proposed solution meets its requirements. Last, the results are discussed and limitations highlighted in chapter 6. Finally, an advice for the stakeholder will be given and subsequently lessons learned during this research internship will be discussed.

## 2 Related Work

This chapter discusses related work in four relevant directions. First, we dive a little deeper into the legal basis of this research. In addition, we look at existing methods to detect specific data. Methods that are specific to privacy-sensitive information are discussed and more general methods for big data quality checks and validation are discussed. Finally, a number of methods are mentioned which can be used to measure differential privacy.

### 2.1 GDPR and UAVG

Since May 25, 2018 the EU General Data Protection Regulation (GPDR) applies to processing activities of personal data which have a link to the EU's territory or market. The Dutch implementation of the GDPR (UAVG) can be found here (in Dutch). In short, personal data is information about a specific individual or a small group of people. This only concerns information about 'natural persons' and not about 'legal persons'. A natural person is someone of flesh and blood; a legal person is, for example a company or an organisation. The GDPR concerns about information of living persons. Information about deceased persons does not fall under the scope of the GDPR [4]. In addition, 'identifiable' data also fall under the term 'personal data'. Identifiable information means that you are not able to identify at this time, but will be able to in the future (with or without additional information). For example, if you have two separate datasets, each of which separately cannot individually identify anyone, but if they are merged, then those datasets may need to be viewed as containing personal data even before they are merged. This will also be an important aspect of this thesis. As a final note, it is important that even though encryption or pseudonymisation are recommended as security measures but this does not mean that no personal data is processed, as the data still refers to a unique individual. It can be concluded that a lot of data can be seen as personal data. However, it is too simple to state that all data is regarded as personal data. It requires a lot of effort to distinguish whether some data should or should not be seen as 'personal data' [28].

## 2.2 Methods of detecting privacy sensitive information in big data

A summary of different approaches to reveal PSI is written by W.B. Tesfay et al. The authors discuss the challenges in detecting privacy revealing information using ontologies, Natural language processing (NLP)-methods and machine learning (ML)-methods in unstructured text [26]. Tesfay and colleagues propose in another paper a machine learning method based on natural language processing to detect privacy sensitive information (PSI) in unstructured texts [27]. Just like Tesfay, Y. Nan et al. uses NLP in combination with keyword detection to identify sensitive user inputs in mobile apps [18]. Another type of machine learning used in finding personal health data is proposed by P. Jindal and colleagues. In their paper they use a novel semi-supervised technique for finding privacy-sensitive events of patients in clinical texts [15]. A completely different method is proposed by V. Menger and colleagues. Their method relies on lookup tables, decision rules and fuzzy string matching. In their paper they propose a method that detects protected health information (PHI) in nursing notes and treatment plans [17]. Their methods to detect PSI will be used as inspiration for detection methods in this thesis.

## 2.3 Methods of big data validation and quality checks

A. Zamperoni and P. Löhr-Richter describe a few statistical data validation methods which are used on a dataset with information related to cheese production. These methods are used to detect outliers in the data [14]. A. Wiggins and colleagues describe most commonly used methods to validate data obtained by surveys [29]. Strategies and techniques to check data quality in general are described in several papers [6][8][25]. Since every organisation specifies their own requirements to determine data quality P. Woodall and colleagues propose and evaluate a hybrid approach to assessing data quality. They show how to dynamically configure an assessment technique while they are still making use of the best practices from existing assessment techniques [30]. Last, a paper written by A.F. Haryadi and colleagues give an overview of important data quality aspects in financial service organisations and a few commonly used methods by those organisations to preserve a good data quality [13]. This paper provided some great background knowledge about big data problems in financial organisations, which was really helpful in understanding the underlying problem at Achmea and writing this thesis.

## 2.4 Methods of measuring privacy

If only quasi-identifiers are present in a dataset, this does not mean that these are directly personal data. They must still be traceable to a natural person or a small group of natural persons. In these cases, methods can be used that calculate the chance an individual can be identified. In case of implementing algorithms that ensure privacy you will need to make sure the algorithm complies with the requirements of differential privacy. *"Differential privacy is a mathematical definition of privacy in the context of statistical and machine learning analysis."* [20] An algorithm is said to be differentially private if you are not able to whether an individual's data was included in the original dataset or not. Several studies have been conducted into the processing of data in which a certain degree of privacy is a must. A. Friedman and A. Schuster consider the problem of data mining with formal privacy guarantees. Here they delve deeper into the trade-off of privacy and the value of information. They demonstrate an improved algorithm that can achieve a high level of accuracy and preserve privacy [12]. Cynthia Dwork proposes a new approach to privacy-preserving data analysis in which there is no risk incurred by joining a statistical database, which means ensuring differential privacy [11]. K-anonymity is a property that is possessed by certain datasets. An algorithm which is able to change a dataset such that it complies with the k-anonymity property ensures differential privacy. The concept of k-anonymity was first introduced by Latanya Sweeney and Pierangela Samarati in a paper published in 1998. They attempted to solve the problem: *"Given person-specific field-structured data, produce a release of the data with scientific guarantees that the individuals who are the subjects of the data cannot be re-identified while the data remains practically useful."* [23] A release of data is said to have the k-anonymity property if the information for each person contained in the dataset cannot be distinguished from at least $k - 1$ individuals whose data also appear in the the same dataset. Another possible method like K-anonymity is L-diversity, developed by Ashwin Machanavajjhala and colleagues [16]. This method looks a like K-anonymity, but has as additional requirement that the values of the sensitive attributes are well represented in each group. However, this method has the disadvantage that it is slower. It was decided to use K-anonymity in this thesis because a faster runtime is valued above the advantage of this additional requirement.

# 3 Approach

This chapter discusses the approach to answer the four research questions.

## 3.1 Defining personal information

The first step to answer the main research question (How to detect privacy sensitive information in large file systems containing structured data?) is to determine which privacy sensitive information is used by the stakeholder.

The guardians of these sensitive information are so called data stewards within Achmea. Data stewards have various duties and are responsible for the quality, use and significance of data within the organisation. The data stewards should be familiar with the data, know how data can be applied, and understands which (privacy) issues may arise in this area. Furthermore, the data steward must ensure correct metadata, monitor data quality requirements and metric data standards, with the aim of ensuring correct decision-making. Achmea's policy is that when working with personal data, a data governance request must be submitted to a data steward. The data steward has then to determine whether the processing of personal data is done properly, whether only the necessary persons who need it have access to it and whether its use is proportionate. A data governance request needs to be submitted for each (group of) dataset(s) you would like to use and needs to describe why you would like to use it, what for, how long, who will have access and a number of other practical matters.

To get an overview of which personal information is present in data managed by Achmea, a data steward and SAS administrator are consulted. In combination with Achmea's guidelines regarding data usage, privacy sensitive information in this overview will be divided in identifiers and quasi-identifiers. In order to decide which information fits into which category, it is examined whether the information is about precisely one person or whether the information can concern several people. These decisions are based on the most negative scenarios. For example, knowing a particular street will normally not be enough to identify a person, because multiple people live in a street. However, there are streets in the Netherlands with only one address. In this case, a street name can still be seen as an identifier.

## 3.2 K-anonymity

The second step is to determine when quasi-identifiers should be a privacy-concern. This is important to the stakeholder to be able to determine whether it should be acted upon. One possibility to calculate how difficult it is to identify a specific individual is to use k-anonymity. K-anonymity is a property that is possessed by certain datasets. K-anonymity was first introduced by Latanya Sweeney and Pierangela Samarati in 1998. A release of data is said to have the k-anonymity property if the information for each person contained in the data cannot be distinguished from at least $k - 1$ individuals whose data also appear in the release. The stakeholder can simply set a limit value, $k$. If the $k$ is lower than the set limit value (and the chance of identifying someone increases), the same security conditions must apply as with datasets which contain identifiers. K-anonymity is often used as a method to alter a datasets such that it has a certain K-value. In this research, the goal is to detect privacy sensitive information. The goal is not to alter data such that it is not sensitive anymore. In addition, using K-anonymity to make suggestions to alter data will make a proposed solution slower because more steps need to be taken to calculate such suggestions.

How difficult it is to achieve k-anonymity is dependent on the cardinality of the data. Cardinality specifies the number of different values a feature can have in a dataset. A high cardinality dataset is more prone to generalisation techniques on its features than a low cardinality one, because there are more possible generalisations to choose. K-anonymity assumes it is possible to pinpoint a specific individual to a data entry in the dataset. K-anonymity tries to prevent three situations. First, it tries to prevent a situation that an attacker can tell that a given person is in the dataset. Second it prevents that an attacker is able to tell that a given person has a certain sensitive attribute and third it prevents that an attacker is able to tell which record corresponds to a given person. To be able to know which data corresponds to which individual, you should have some prior knowledge about the people who are ‘described’ in the dataset.

## 3.3 Ensuring scalability

The third step is to ensure that the solution will be scalable. Because there is an enormous number of datasets available, whereby the datasets themselves can also be enormous, it is important that the solution is scalable. The datasets processed by the stakeholder varies from a few thousand rows to many millions of rows. In addition, there is a total of 900,000 files containing datasets. Because of this variation there is a need for a specific solution. If the variation would have been different, for example large amounts of data obtained in a stream, a different solution would likely perform better. The reason of this is because additional/other assumptions can be made about the format of the data. Scalability for this particular situation will be measured both in memory and in time.

First, in order to be able to scale in the size of the datasets, it should be guaranteed that a dataset will always fit in memory. One way to solve this is to divide datasets which are too large into pieces and process them separately from each other. The results must subsequently be combined.

Secondly, in order to be able to scale in the number of datasets, it is important that not all datasets are read in memory at once. If this is the case, it is possible that there will be a shortage of working memory. So a memory-friendly solution will have to address this issue.

Finally regarding runtime, in order to be able to scale both in large datasets and in large amounts of datasets, it is important that the algorithm is worst case linearly dependent on the input. If the solution becomes slower with more input, an unworkable situation will arise eventually. In addition, a number of features should be implemented in order to save time whenever possible. An example would be using a filter that is able to detect non-sensitive columns based of their properties (length, type et cetera). Due to the use of these kinds of features, runtime will depend not only on the size of the dataset, but also on the content.

## 3.4 Detecting privacy sensitive information

To be able to accurately implement a solution to detect PSI, some preconditions and assumptions about the data need to be made. These preconditions and assumptions have been made in consultation with the client. The first precondition is about data formats.

Knowing how the data is formatted will increase the effectiveness of the script accurately by detecting more privacy sensitive information. Everyone within Achmea who uses the SAS platform has to follow some basic guidelines specifying how data should be formatted. Most employees even followed training in which naming columns and formatting data is discussed. The first rows contains the header. If there is no information available or not applicable, a field is left empty. Because of this, it was possible to make some assumptions about the datasets. For example: all numbers will be assigned to a numeric field (integer, double etc.), dates will have specific date-time format and column names should accurately say which data can be found in that column. In practice this means, we do not have to try all possible formats (dates have a lot of common formats. For example: 01-01-2020, or 1 January 2020, or even 01JAN20). All datasets that were encountered during this research met these requirements.

Another assumption is be made is that within a column, only one type of data will be present. It should not be possible that in column 'A' the first 10 rows are costs, and following rows contain addresses. Another important premise is that it is not necessary to actually know every instance. What is important is to recognise enough to flag a column that is full of certain PSI. An example: it is okay if a few unique personal names are not recognised, as long as enough are recognised to be able to decide whether the column is about personal names or not. The stakeholder prefers to have a fast algorithm that is able to recognise 99% rather than a complete algorithm that is capable of recognising 100% but will not be practical in terms of runtime. A dataset contains several columns. At Achmea this often involves a few dozen columns. In addition, the dataset contains a number of rows. At Achmea this often varies from a few tens of thousands to a few million rows. The datasets used for the experiment contained on average 3 to 4 columns with (quasi-)identifiers.

A basic conceptual solution, which takes aforementioned scalability aspects into account would look like this:

**Algorithm 1:** Detecting privacy sensitive information - conceptual algorithm

```
Result: Documents containing PSI
set parameters;
list all files;
results = [ ];
for document in all documents do
    intermed_results = [ ];
    if size document < threshold then
        chunks = whole document;
    else
        chunks = divide document in multiple chunks;
    end
    for columns in all chunks do
        for column in all columns do
            if column contains PSI then
                intermed_results append column;
            else
                continue;
            end
        end
    end
    k = calculate K-anonymity based on intermed_results;
    results append (document, k, intermed_results)
end
Save results in file;
```

Calculating K-anonymity can be done by combining all columns in *intermed_results* into one column and subsequently using Pandas *value_counts*() function. The value that occurs least in the dataset will be the K value. To calculate a certainty, 1 gets divided by *k*. A certainty value is calculated for each dataset. Using native Pandas functions that are highly optimised is preferred. Even for a lot of columns and lots of rows; the time it takes to calculate K-anonymity is a small fraction of the time it takes to detect PSI and does not have a significant effect on the time complexity.

The proposed solution above is basic but slow. To ensure a faster solution, smarter algorithms are needed. One way to make a solution faster is by using functions which identify columns of which it is easy to see that they do not contain any privacy sensitive information. Smart filter functions will be utilised in the final proposed solution. There are several ways to detect PSI. A simple way is by using regular expressions. Another way is to use a checksum (in the example of Dutch BSN numbers). Depending on the type of PSI, a specific method will be preferred. An example of a PSI test which could be performed on a column in Algorithm 1 is shown below in Algorithm 2:

**Algorithm 2:** Detecting Dutch BSN number

```
Result: True or False on Dutch BSN test
Arguments: column of document
initialisation;
for field in all rows do
    if field complies with 11 checksum then
        return True;
    else
        Continue;
    end
end
return False;
```

## 3.5 Evaluation

To know which privacy sensitive information is being processed by the stakeholder we depend on the datasets provided by the client and data stewards. Furthermore, an existing solution, K-anonymity, is used to determine when privacy sensitive information is used to identify a natural person.

To determine whether this approach is suitable regarding scalability, we need to take the following aspects into account:

1. Is it able to handle large files (where large is defined by stakeholder as millions of records and/or larger than 100MB up to a few GB)?

2. Is it able to handle a lot of files (where a lot is defined by stakeholder as many of thousands of files)?

3. Is it able to accurately detect specified privacy sensitive data?

4. Is the script fast enough to be practical?

Data used for this research is provided by the stakeholder. The server on which the script should ultimately run contains approximately 54 TB of files. of the 54TB, 50TB consists of datasets in the form of sas7bdat (49TB), CSV (800GB) and xls(x) (200GB). Altogether it contains about 900,000 files. However, although it was possible to view all data, for technical and privacy reasons it was not possible to transfer a lot files in a place where the script could be run. In total 10 large production datasets (more than 10,000 lines) were used in combination with some dummy datasets for testing of the script. The client at Achmea has ensured that these datasets are representative of all datasets which are present in the SAS environment.

To determine whether the solution was able to handle large files, it was tested on several files of approximately 100 MB large, and a single dataset which is almost 1 GB large. These files were provided by the stakeholder. Because large files are read as an iterator, with chunks of 10.000 rows, it was not expected to be a problem since only a single chunk is read in memory at a time. To ensure that the datasets are large enough to reveal any memory issues, they are each read multiple times, making up to take 100 GB of memory in total.

To determine whether the solution is able to handle a lot of files, a few small dummy files have been used. These files have been read multiple times until 50.000 files have been read in total. What the above tests actually tried to prove is that the memory usage did not increase to be too large to handle. Since the solution must be scalable, the only way to achieve this is to have a space complexity of O(1). A solution is implemented in Python and tests were run on a server running RStudio. The server used is a L5 Linux workstation.

In terms of time complexity, the script should run in $O(i * n)$, where $i$ is the amount of privacy tests, and $n$ is the amount of documents which should be tested. Because $i$ is a fixed number, there are 16 privacy tests in this case, time complexity should be $O(n)$. It is difficult to determine exactly how fast the solution will be. This depends on many factors. For example, SAS7BDAT and CSV files can be read faster than XLS(X) files. Files with many empty rows are also faster just like files with a lot of PSI. If PSI are found in a column, there is no need to process remaining rows of that column.

# 4 Results

In this chapter the privacy sensitive information processed by the stakeholder is addressed and categorised as identifier or quasi-identifier. This information was obtained by talking to the SAS administrator and a data steward. Afterwards the implementation of K-anonymity is discussed. Subsequently a proposal for an implementation is explained and its scalability is tested using various datasets. Finally, a number of problems are discussed that are encountered while getting the results.

Design sprint
Sprint 1
Sprint 2
Sprint n
Retro-
spective

Figure 1: Working agile

At Achmea we worked in an a more or less Agile way. The main focus was to produce working functionality in short periods of time (sprints). This meant that in the first sprint the focus was placed on clearly defining the overall assignment, determining which steps had to be taken. Finally, evaluation criteria have been agreed with the client. During the following sprints a solution was made, improved iteratively and tested thoroughly by datasets provided by the stakeholder. At the end of each sprint, the choices made were discussed with the client. In the final sprint demo, the integrated solution was presented. In addition a retrospective was carried out for establishing the lessons learned in order to improve the next sprints or release.

## 4.1 Definition and scope privacy sensitive information

The conversation with the SAS administrator and a data steward resulted in a list of 16 (quasi-)identifiers. Sensitive information is classified as identifier or quasi-identifier. With this classification it is possible to determine how sensitive something is. If a dataset contains an identifier, there is a 100% chance to identify a person. If a dataset contains one or more quasi-identifiers, K-anonymity will be used to calculate the chance someone will be identified. A list of all privacy sensitive information that is being processed by the stakeholder is listed below.

1. Address: addresses consists of a street name and a house number. An address refers to only one house. Therefore it is regarded as an identifier.

2. Birthday: a birthday consists of a year, month and day when someone is born. Because many people have the same birthday, it is regarded as a quasi-identifier.

3. Birthplace: a birthplace is a city. A lot of people are born in the same city and therefore it is regarded as a quasi-identifier.
4. BSN number: in the Netherlands everyone has its own unique BSN number. Therefore it is regarded as an identifier.
5. City: same as birthplace, it is regarded as an quasi-identifier.
6. Email address: although an email address in itself is not privacy sensitive, a lot of people use their name (and often their birthday) as their email address (for example john_doe1990@hotmail.com). This information can be used to identify a specific individual. Since a email address is unique, it is regarded as an identifier.
7. Employee's ID: everyone who works at the company has their own employee number. Therefore it is regarded as an identifier.
8. First name and surname: While many people share a name, there are also plenty of people with a unique name or combination of first and last name. Therefore names are considered as an identifier.
9. Gender: people have their gender in common and therefore it is regarded as a quasi-identifier.
10. IBAN number: IBAN numbers are unique and traceable to a natural person. Therefore it is regarded as an identifier.
11. KVK number: in sole proprietorships this number can be tracked to a natural person. Therefore it is regarded as an identifier.
12. License plate: just as IBAN, license plates are unique and traceable. Therefore it is regarded as an identifier.
13. Marital status: Everyone falls within the scope in one of seven possible options and therefore it is regarded as a quasi-identifier.
14. Old BTW number: the old dutch BTW number can be a combination of a BSN, a 'B' and two digits. Because it contains a BSN, it is regarded as an identifier. New BTW number have 9 random digits instead of a BSN and is therefore not regarded as a (quasi-)identifier.
15. Phone number: just as IBAN, phone numbers are unique and traceable. Therefore it is regarded as an identifier.
16. Postal code: there are postal codes in the Netherlands that contains only a single household. Therefore it is regarded as an identifier.

Other examples of quasi-identifiers are: race, education, work class or native-country. These examples are not processed by the stakeholder and therefore not included.

## 4.2 K-anonymity

As described before, K-anonymity will be used in the last step of the solution as a method to indicate whether quasi-identifiers can be regarded as identifiers. This will be represented by the chance the PSI could be used to identify a person. K-anonymity was chosen because it can be calculated quickly and therefore has little influence on the total runtime. In the proposed solution all files containing at least one column PSI that is classified as an identifier will get a score of 1 (100%). Of all files that contain only quasi-identifiers and attributes a regular k-anonymity score is calculated. Using this method a worst-case probability could be calculated. Only the columns containing PSI will be used in the calculation of the k-anonymity score. The chance will be $1/k$ where k is the K-anonymity score. Tests on Achmea's production data sets often show that if they contain personal data, there is often a score of 1. Sometimes there is a direct identifier. It is more often the case that there are several quasi-identifiers, so that the final result often ends up with a chance of 100%.

## 4.3 Implementation privacy sensitive information detection

With the preconditions defined, as discussed in chapter 3.4, a script has been developed that is able to detect the aforementioned sixteen categories of privacy sensitive information. Scalability is ensured by reading datasets in chunks and by processing it one by one.

The first step to detect privacy sensitive documents in files is to read the headers and only the first row of each file. The files will be read into a Pandas dataframe. With this information, the algorithm tries to determine whether a file could contain interesting information, without the need of reading each file completely in memory. Only structured data is taken into consideration. Files with XLS(X), CSV or SAS7BDAT extensions will be read, files with other extensions will be skipped.

The second step is to check whether the headers of the read files match with a list of predefined interesting keywords (examples of interesting keywords could be gender, city, age or first name). A list of keywords was obtained by speaking with the SAS administrator and a few end-users. If a column has a header that matches with one of the keywords, the name of the column will be saved in a list together with the corresponding file. If the header of a column does not match a keyword, the first row will be inspected. It tries to determine whether this row has characteristics that could match characteristics of the PSI. An example of this would be matching the characteristics of the first field with the characteristics of a BSN number (without doing the expensive calculations to determine whether it is an actual BSN number which in this case is doing a checksum). It would look if that field contains an integer with a length of 9 digits. If this is the case (or it matches with the characteristics of another PSI), this column will also be added to the list. If the first field is empty (or produces an error), it will also be added to the list to be sure nothing will be missed. This tells us which columns of which file might contain PSI. Using this method it is possible to reduce the amount of files (or at least reduce the columns of a file) that needs to be read.

The third step is to read all rows of the listed columns of corresponding files. XLSX files are read completely because excel files are compressed and therefore not suitable to read line by line. CSV and SAS7BDAT files will be read as an iterator. Iterators are able to split the file in chunks and read them separately. This reduces the memory usage drastically and tackles the problem of handling large files. Data in Excel files can not be read as an iterator due to their nature of being zipped. For that reason data in the Excel files need to be read in as a complete set. A complete search for PSI is subsequently performed per file and per column. The column will be filtered of null values before being searched through. Search is based on string matching (example: gender female / male), based on regular expressions (example: telephone numbers and postcodes) or based on lookup lists (example: names and cities). When something is found, the corresponding category is saved. To be saved a certain percentage of the number of fields, chosen by the user, should test positive. If this is not done, a lot of false-positives could occur. A category where false-positives occur is when searching for BSN numbers. For example amounts, it is possible one of them will coincidentally match a BSN number. Now it knows for each file which categories of PSI it contains.

These steps are done consecutively for each document. When a dataset has gone through all the steps, the result is immediately written to an Excel file. Should the script be terminated prematurely, the results of the datasets that are finished will at least be saved. A simple overview how this script work is presented in figure 2.

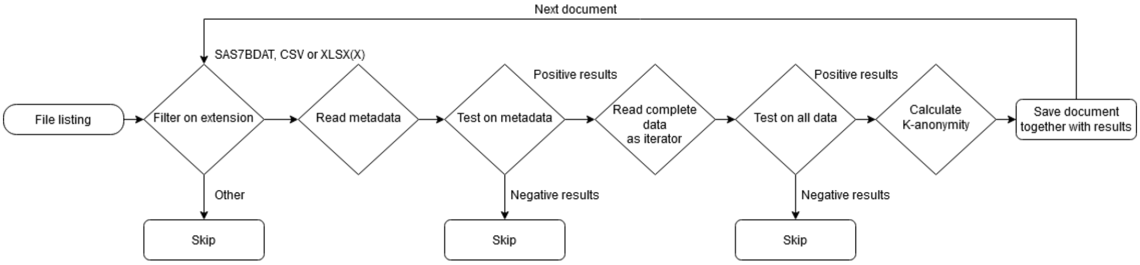

Figure 2: Diagram steps proposed solution

## 4.4 Ensuring and testing scalability

Two experiments were performed to test the scalability of the solution. The first experiment looked at scalability in the number of files. Dummy files have been used for this purpose. The dummy files were constructed in such a way that they resemble datasets such as those that can be found at Achmea. The datasets are quite small, from a few hundred to a few thousand rows to save time. It has been agreed with the client that a test of up to 50,000 files is sufficient to demonstrate scalability.
The results of the first experiment can be seen in figure 3.

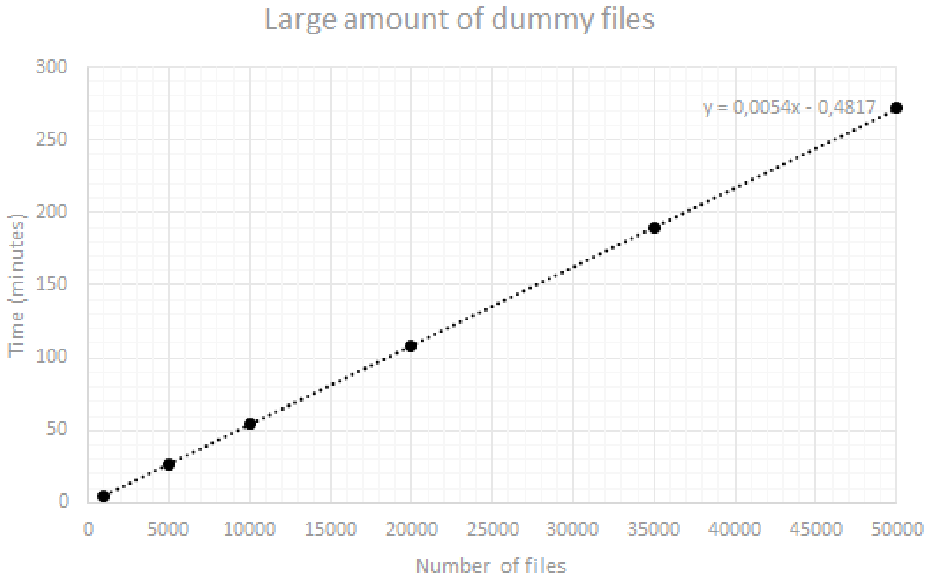

Figure 3: Experiment with a lot of small dummy datasets

The results show that proposed solution is able to handle 50,000 files and runtime is linearly dependent on the input. Thus it complies with the preconditions set for the solution. The results do not give a representative indication of the runtime since the input consist of artificially created dummy sets.

For the second experiment, datasets were actually used that are produced by Achmea. To test whether the solution is also scalable in terms of memory, it was tested with large files (from several hundreds of thousands rows to several million rows). Here too it is important that the time is linear, meaning that results from one document do not affect the results of another document. Results of the second experiment are showed in figure 4.

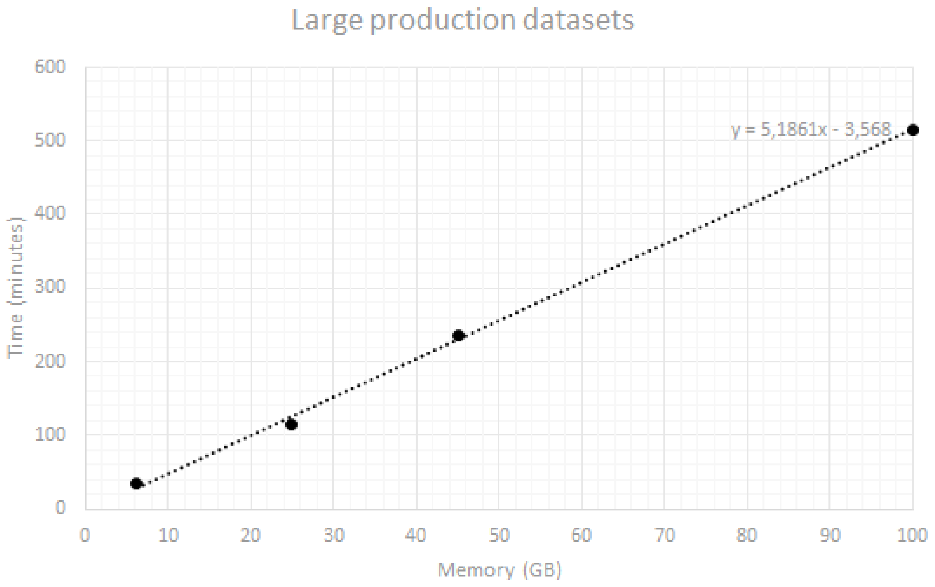

Figure 4: Experiment with some large production datasets

The results show that the solution is able to handle many large data sets. The time looks linearly dependent on the input. In this case, the runtime is representative because the input is representative of the datasets that are being processed at Achmea. Although the runtime in figure 4 is linear, this does not mean that the proposed solution will always be linearly dependent. There will be minor differences in each dataset, depending on what data is present in the dataset. Small differences can be caused because there is for example a lot of personal data (the algorithm stops searching a column when something is found) or because many columns are already dropped in the test-on-metadata phase. However, in practice it appears that, there are few such columns in proportion to the total amount of columns. Because of this, for large numbers of datasets the result will look like linear.

## 4.5 Problems

A number of problems were encountered while collecting the results.

First, it was no longer possible to get large amounts of data sets to a network drive where they could be scanned. This problem was caused by a combination of technical and privacy sensitive reasons.

In addition to being a challenge to transfer files to another platform, this platform also had to meet the same security requirements as the platform on which the data was already stored. Unfortunately, it was not possible to find a platform within the specified time that met the high security requirements and was technically able to access the SAS data so that the script could be run over it.

In addition, the plan was to implement the solution in PySpark. This would distribute and process the datasets across several nodes, speeding up the overall solution. Due to several factors, of which time was the most important, it was decided to implement the solution in native Python. With relatively few adjustments it is possible to convert this into a solution in PySpark in the future.

# 5 Conclusions

First, conclusions regarding the additional challenges (that where identified in the introduction) will be drawn. Subsequently, the advantages and disadvantages of the proposed solution and K-anonymity are discussed. Finally, a conclusion is presented on the research questions.

## 5.1 Challenges

In addition to the research questions there are 4 additional challenges stated in section 1.2 that needed to be addressed. The first challenge was the size of the individual files. Although some files consist of several thousand rows, others may consist of many millions rows. This challenge has been addressed by dividing the large files into chunks. Those chunks are read and processed separately and the results will be combined. However, this solution was not possible for Excel files. Since the number of Excel files is relatively small compared to the number of SAS files, it is expected that this will not be a problem. However, this could be a limitation in other applications. Perhaps it is also possible to find a way to solve this problem specifically for Excel files.

The second challenge was about the number of files. This challenge has been solved by reading and processing the files separately. This means that not everything has to be stored in memory at once.

The third challenge was handling different formats to store information in a file. Making assumptions about formats was necessary to be able to detect privacy-sensitive data more quickly. The challenge has been resolved by discussing with the client which assumptions can be reasonably made and which assumptions can not be made.

The fourth and final challenge was processing different types of files, namely SAS7BDAT, CSV and XLS(X). By using different Python implementations for the different types of files, it was possible to read them as optimally as possible.

## 5.2 Benefits and limitations detection method

Although the proposed solution in 4.3 and 4.4 is able to solve almost all challenges, it has some drawbacks. The first and most important is that there is no way to guarantee that it is able to detect all sorts of privacy sensitive information. Some assumptions have been necessarily made in order to come up with a scalable solution that is able to detect privacy sensitive data within a reasonable time. Although these assumptions are supported by the stakeholder, it is always possible that there are datasets that do not comply with these assumptions and therefore will not be detected.

The second shortcoming is the runtime. Although the solution is scalable, it is not yet fast enough to be practical. It currently takes roughly 8 hours to scan 100 GB of data. The plan is to run a scan every month. If only newly added and modified files are scanned, results should be obtained within a day. This roughly amounts to a few TB per month. In the current way, a scan could then take up to a week. However, a solution to this would be to implement the solution in PySpark. This makes it possible to process documents across multiple nodes in a Spark cluster and the solution could be up to 10 times faster. This means that it can scan 3 TB of data per 24 hours. Unfortunately, due to the time, it was no longer possible to make this adjustment within this research project. With this adjustment, the solution would meet all the requirements. However, this will not be a problem for the majority of the file.

The solution is able to accurately detect personal information and will help the administrators of the platform to gain insight into the personal data that is stored on the SAS platform.

## 5.3 Benefits and limitations K-anonymity

There are some limitations to the use of k-anonymity. One problem is that the value calculated when using K-anonymity is not always representative of the problem. An example: propose a dataset in which only the quasi-identifiers city and gender occur. In a small dataset it is quite possible that one person comes from Amsterdam, for example, and is a woman. In this case is $1/k = 1$. However, the information that the person is a woman from Amsterdam does not help much to identify the person. This would be accurate in the case you are able to determine which people are present in the dataset. In that case, knowing a person named Anna which you know lives in Amsterdam is enough information to determine which record is about her.

Another limitation of K-anonymity is its susceptibility to de-anonymisation techniques. Such techniques include Isolation Attacks, where an individual's record from the dataset can be recovered (isolated from the other records) using information found in an anonymised public dataset. If for example Bob occurs in both datasets (and this is known); one dataset contains record with {A, B, C}, and the other dataset contains a record with {Bob, A, B}. By joining these records it can be concluded that information C will also be about Bob. This does only apply if there is no other record present which matches. Another technique is the Information Amplification Attack. In this case the adversary learns additional information about the individual, although he may not be able to uniquely identify a person. This can be achieved by exploiting distributions of sensitive information if sensitive values in an equivalence class lack diversity or if the attacker has background knowledge [19][10].

Despite its limitations, K-anonymity is certainly not worthless. Datasets from Achmea will not be made public, not even if the datasets are anonymised. This makes the aforementioned attacks such as Isolation Attacks and Information Amplification Attacks a lot more difficult to perform. An advantages of K-anonymity is that the score is easy to interpret. In addition, the score can be calculated very quickly, even for large data sets.

Sixteen different types of privacy sensitive information are being processed by the stakeholder. By defining PSI as identifier or as quasi-identifier in combination with using K-anonymity you are able to determine whether actions need to be taken to ensure compliance with legislation and company policy. This answers the second sub question *(What is the stakeholders definition of privacy sensitive information? Which of those types of information is on its own enough to be seen as PSI? Which information needs to be combined with other information to be seen as PSI?)* and third sub question *(How to determine when privacy sensitive information can be used to identify a natural person and what would be the limitations of such an approach?)*. To answer the first sub question, how to ensure scalability, methods are proposed and tested to ensure a solution will be able to handle growing amounts of data.

# 6 Discussion

Finally, conclusions are discussed and are used to provide advice to the stakeholder. In addition, a number of important lessons learned are addressed and finally an outlook of this research project is discussed.

## 6.1 Advice for stakeholder

A number of results have emerged from this research project. With these results it is possible to recommend a few important follow-up steps which could benefit the stakeholder. There are two types of follow-up steps. First type are possible ideas about improving the proposed solution. The second type is about applying the solution to make an impact on the business. First a few possible improvements will be discussed.

The first recommendation would be by improving the proposed solution by implementing it in PySpark in combination with Pandas user defined functions (UDF). Over the past few years, Python has become the default language for data scientists. But, at the same time, Apache Spark has become the default language in processing big data. To enable data scientists to use the best of both, Spark added a Python API with support for user-defined functions. With this API you have the ability to define low-overhead, high-performance User Defined Functions (UDFs) entirely in Python with the scalability of Spark. Another way to reduce running time is to scan only the changed and newly added files (after the initial scan). This feature is already implemented in the script.

Another possible improvement could be supporting more file formats. Every file which is able to be transferred in a dataframe could be suitable. If more file formats are supported, this solution could be used in more diverse cases. This could also result in more impact on the business.

Last, in order to make optimal use of this research project, a follow-up step need to be done. This solution aims to provide insight into which personal data is present in a document. Achmea's policy is that when working with personal data, a data governance request must be submitted. A data governance request is submitted to a data steward. He will determine whether the processing of personal data is done properly, whether only the persons who need it have access to it and whether its use is proportionate. The next step is to compare the results of the script with all those requests. With this information combined it can be used to determine whether the data present is actually allowed to be stored according to the data steward and whether the right persons have access to it. If this is not the case, follow-up steps can be taken to meet these requirements. By submitting a data governance request, modifying the data governance request or by simply removing the data.

## 6.2 Lessons learned

The research project resulted in achieving much more experience working within a large organisation with large datasets. My programming skills have improved. I have become familiar with new programming methods and programming packages. Being able to critically assess which of the possible methods is the best was also very important during this project. While the results of these methods were often the same, it could have a lot of impact on run time. When this research project started, it was expected machine learning was used as a possible method of detecting PSI. However, after gaining more insight into the data and the various personal data that had to be detected, this no longer seemed the most practical solution. Much of the personal data would be easily identifiable through regular expressions. These are inherently faster than using Machine Learning. In addition, for other categories of personal data, the challenge was to come up with suitable features to make an accurate classification. What features does a date have? A municipality? A name? Other methods seemed more suitable for this.

Another point where I learned a lot is working remotely. Due to the Corona virus my research project was done entirely from home. In the beginning this was quite difficult. It took longer to get to know your colleagues and thus it also took longer before you knew where to turn to for a specific question. The Achmea team where I did my research project did their utmost to get me started as quickly as possible despite the limitations and partly for that reason the Corona virus will have had little impact on the final result.

Finally, measuring the quality of the software was less trivial than expected at the outset. The most difficult question was how it can be proven that the solution finds everything. The solution was able to find everything in the dataset provided by the stakeholder. However, this was only a small number compared to the amount of datasets they have in their possession. The only way to say something useful about this was to go through the tests with the client so that he could provide insight into whether it will work or not and to test on a number of representative datasets.

## 6.3 Concluding summary

In this thesis an answer was presented on how a scalable solution can be created that is able to detect privacy sensitive information in datasets processed by Achmea. Several challenges have been identified in the introduction and resolved one by one in later chapters.

First it was needed to ensure scalability. To ensure a scalable solution, a number of requirements must be met. For example, the memory usage may not depend on the number of datasets to be scanned, and the runtime may at most increase linearly with more datasets. Additional savings in runtime can be achieved by using filter methods based on meta-data. Second, a definition of privacy-sensitive information processed by the stakeholder was required. With the help of a data steward and a SAS administrator, 16 different privacy-sensitive information types were defined that had to be detected and classified as quasi-identifier or as an identifier. Third, by using methods that are able to qualify privacy, such as K-anonymity, an insight can be gained whether quasi-identifiers should be seen and acted upon the same as is the case with identifiers.

The proposed solution can be improved by implementing it in PySpark, which will make the proposed solution even faster. With the results of this thesis the insight of the stakeholder is improved. More important, it will make it easier to comply with the GDPR/UAVG legislation because this solution can be used to check if available privacy sensitive information on a SAS platform meet Achmea's requirements for data processing.

# References

[1] EUR114 million in fines have been imposed by European authorities under GDPR | News | DLA Piper Global Law Firm.

[2] Sancties voor Menzis en VGZ voor overtreding van de privacywet, november 2020. Publisher: Autoriteit Persoonsgegevens.

[3] Accountability Act. Health insurance portability and accountability act of 1996. *Public law*, 104:191, 1996.

[4] Jan Philipp Albrecht. How the GDPR Will Change the World. *European Data Protection Law Review (EDPL)*, 2:287, 2016.

[5] Navneet Arora and Prashant Khanna. Gdpr: Six principles of gdpr. 2019.

[6] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. Methodologies for data quality assessment and improvement. *ACM Computing Surveys*, 41(3):16:1–16:52, July 2009.

[7] Ronald Camhi and Scott Lyon. What is the california consumer privacy act? *Risk Management*, 65(9):12–13, 2018.

[8] Corinna Cichy and Stefan Rass. An Overview of Data Quality Frameworks. *IEEE Access*, 7:24634–24648, 2019.

[9] Sabrina De Capitani Di Vimercati, Sara Foresti, Giovanni Livraga, and Pierangela Samarati. Data privacy: definitions and techniques. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 20(06):793–817, December 2012. Publisher: World Scientific Publishing Co.

[10] X. Ding, L. Zhang, Z. Wan, and M. Gu. A Brief Survey on De-anonymization Attacks in Online Social Networks. In *2010 International Conference on Computational Aspects of Social Networks*, pages 611–615, September 2010.

[11] Cynthia Dwork. Differential Privacy: A Survey of Results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, Lecture Notes in Computer Science, pages 1–19, Berlin, Heidelberg, 2008. Springer.

[12] Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*, page 493, Washington, DC, USA, 2010. ACM Press.

[13] Adiska Fardani Haryadi, Joris Hulstijn, Agung Wahyudi, Haiko van der Voort, and Marijn Janssen. Antecedents of big data quality: An empirical examination in financial service organizations. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 116–121, December 2016.

[14] S. A. Jimenez-Marquez, C. Lacroix, and J. Thibault. Statistical Data Validation Methods for Large Cheese Plant Database. *Journal of Dairy Science*, 85(9):2081–2097, September 2002.

[15] Prateek Jindal, Carl A. Gunter, and Dan Roth. Detecting privacy-sensitive events in medical text. In *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, BCB '14, pages 617–620, New York, NY, USA, September 2014. Association for Computing Machinery.

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

[17] Vincent Menger, Floor Scheepers, Lisette Maria van Wijk, and Marco Spruit. DEDUCE: A pattern matching method for automatic de-identification of Dutch medical text. *Telematics and Informatics*, 35(4):727–736, July 2018.

[18] Yuhong Nan, Zhemin Yang, Min Yang, Shunfan Zhou, Yuan Zhang, Guofei Gu, Xiaofeng Wang, and Limin Sun. Identifying User-Input Privacy in Mobile Applications at a Large Scale. *IEEE Transactions on Information Forensics and Security*, 12(3):647–661, March 2017.

[19] A. Narayanan and V. Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, May 2008. ISSN: 2375-1207.

[20] An Nguyen. Understanding differential privacy. *Medium*, Jul 2019.

[21] Neil Robinson, Hans Graux, Maarten Botterman, and Lorenzo Valeri. Review of the european data protection directive. *Rand Europe*, 2009.

[22] Thijs Rösken. 'Zorgverzekeraars overtreden privacywet AVG', june 2019. Publisher: Telegraaf.

[23] Pierangela Samarati and Latanya Sweeney. Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression. page 19, 1998.

[24] Latanya Sweeney. k-ANONYMITY: A MODEL FOR PROTECTING PRIVACY. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, October 2002. Publisher: World Scientific Publishing Co.

[25] Ikbal Taleb, Mohamed Adel Serhani, and Rachida Dssouli. Big Data Quality: A Survey. In *2018 IEEE International Congress on Big Data (BigData Congress)*, pages 166–173, July 2018.

[26] Welderufael B. Tesfay, Jetzabel Serna, and Sebastian Pape. Challenges in Detecting Privacy Revealing Information in Unstructured Text. In *PrivOn@ ISWC*, 2016.

[27] Welderufael B. Tesfay, Jetzabel Serna, and Kai Rannenberg. PrivacyBot: Detecting Privacy Sensitive Information in Unstructured Texts. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 53–60, October 2019.

[28] W. Gregory Voss. European Union Data Privacy Law Reform: General Data Protection Regulation, Privacy Shield, and the Right to Delisting. *The Business Lawyer*, 72(1):221–234, 2016. Publisher: American Bar Association.

[29] Andrea Wiggins, Greg Newman, Robert D. Stevenson, and Kevin Crowston. Mechanisms for Data Quality and Validation in Citizen Science. In *2011 IEEE Seventh International Conference on e-Science Workshops*, pages 14–19, December 2011.

[30] Philip Woodall, Alexander Borek, and Ajith Kumar Parlikad. Data quality assessment: The Hybrid Approach. *Information & Management*, 50(7):369–382, November 2013.