CONTRIBUTION TO PRIVACY-ENHANCING TECHNOLOGIES FOR MACHINE LEARNING APPLICATIONS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF TELEMATICS ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF UNIVERSITAT POLITÈCNICA DE CATALUNYA
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Ana Fernanda Rodríguez Hoyos
July 2020

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Jordi Forné Muñoz) Principal Co-Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(David Rebollo Monedero) Principal Co-Adviser

Approved for the University Committee on Graduate Studies.

A mi principio y mi fin, a mi aliento de vida. Gracias por tu amor incondicional y por tu gracia inmerecida, a ti sea todo el honor y gloria.

Abstract

For some time now, big data applications have been enabling revolutionary innovation in every aspect of our daily life by taking advantage of tons of data generated from the interactions of users with technology. Supported by machine learning and unprecedented computation capabilities, different entities are capable of efficiently exploiting such data to obtain significant utility. However, since personal information is involved, these practices raise serious privacy concerns.

Although multiple privacy protection mechanisms have been proposed, there are some challenges that need to be addressed for these mechanisms to be adopted in practice, i.e., to be "usable" beyond the privacy guarantee offered. To start, the real impact of privacy protection mechanisms on data utility is not clear, thus an empirical evaluation of such impact is crucial.

Moreover, since privacy is commonly obtained through the perturbation of large data sets, usable privacy technologies may require not only preservation of data utility but also efficient algorithms in terms of computation speed. Satisfying both requirements is key to encourage the adoption of privacy initiatives.

Although considerable effort has been devoted to design less "destructive" privacy mechanisms, the utility metrics employed may not be appropriate, thus the wellness of such mechanisms would be incorrectly measured. On the other hand, despite the advent of big data, more efficient approaches are not being considered. Not complying with the requirements of current applications may hinder the adoption of privacy technologies.

In the first part of this thesis, we address the problem of measuring the effect of k-anonymous microaggregation on the empirical utility of microdata. We quantify

utility accordingly as the accuracy of classification models learned from microaggregated data, evaluated over original test data. Our experiments show that the impact of the de facto microaggregation standard on the performance of machine-learning algorithms is often minor for a variety of data sets. Furthermore, experimental evidence suggests that the traditional measure of distortion in the community of microdata anonymization may be inappropriate for evaluating the utility of microaggregated data.

Secondly, we address the problem of preserving the empirical utility of data. By transforming the original data records to a different data space, our approach, based on linear discriminant analysis, enables k-anonymous microaggregation to be adapted to the application domain of data. To do this, first, data is rotated (projected) towards the direction of maximum discrimination and, second, scaled in this direction, penalizing distortion across the classification threshold. As a result, data utility is preserved in terms of the accuracy of machine learned models for a number of standardized data sets.

Afterwards, we propose a mechanism to reduce the running time for the k-anonymous microaggregation algorithm. This is obtained by simplifying the internal operations of the original algorithm. Through extensive experimentation over multiple data sets, we show that the new algorithm gets significantly faster. Interestingly, this remarkable speedup factor is achieved with no additional loss of data utility.

Finally, in a more applied effort, we propose a data privacy tool to protect privacy of individuals and organizations by anonymizing sensitive data included in security logs. We design different anonymization mechanisms to then implement them according to the definition of a privacy policy. We adapt said approach to the context of an EU project whose aim is to build a unified security framework. Since this framework collects and processes security-related data (logs, reports, events) from multiple devices of critical infrastructures, our work is devoted to protect privacy there by integrating our anonymization approach.

Acknowledgments

A mis padres por su amor y apoyo siempre. A mi esposo e hijo por su comprensión, paciencia y dedicación.

A mis tutores por su valiosa guía y soporte, por compartir sus conocimientos y, sobre todo, por su amistad.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In present times, sophisticated and powerful information systems are being implemented to achieve an unprecedented level of intelligent behavior and personalization. In a wide variety of fields, more utility can be mined from data to unveil qualitatively superior insight into challenges and opportunities that may otherwise remain undiscovered [3, 4]. This is now possible thanks to the combination of automatic learning algorithms and the increasing availability of data. Namely, vast quantities of detailed information, often referred to as big data, are made available to more sophisticated and powerful information systems.

Part of such sophistication involves machine-learning algorithms that are being developed to automatically discover useful "anomalies", e.g., in medicine, but they still require vast amounts of data to achieve actionable accuracy. Combining such technologies with big data may lead to truly remarkable scientific feats such as a better cancer detection ([5, 6]). In fact, human proficiency is being combined with machine-based mechanisms to provide augmented intelligence from large-scale databases.

An unquestionable product of this revolution is personalization. By adapting services to the specific needs of users, personalization has brought numerous benefits for people and big profits for companies. One of the reasons of its popularity has to do with the effectiveness of personalized services. In fact, personalization may be so accurate that it is currently applied to offer precision medicine or product

recommendation. Note that this is possible since there is a lot of personal information within the large amounts of data processed.

Consequently, the revolutionary advances accomplished in the big data era poses equally serious privacy risks for users. Although identifiers are typically suppressed from shared or published data, some demographic attributes, when combined, can still be used to re-identify individuals ([7–9]). Unfortunately, this re-identification might enable privacy attackers to link the identity of subjects with their corresponding sensitive attributes. Said disclosure might lead to harmful attitudes against subjects, e.g., discrimination [10].

Anonymization is commonly used to reduce this disclosure risk by perturbing demographic attributes to de-identify records. The privacy models enforced through user data perturbation, e.g., k-anonymity [7, 11] or ϵ-differential privacy ([12]), are usually conditioned by a privacy parameter that defines an upper bound on the re-identification risk. However, in practice, other parameters such as data utility and mechanism usability convolute the task of protecting privacy. Evidently, data perturbation comes at the cost of some loss in data utility. Additionally, finding a balance between privacy and utility, when big data is involved, might turn private data analysis unfeasible or unusable for some applications where, e.g., mechanisms must execute in a reasonable amount of time despite the size of data.

These penalties discourage the adoption of privacy protection so it is important to tackle them. First, an empirical metric of utility would help to determine the real impact of anonymization on the utility of data. Since said impact is relative to the application domain of data, its magnitude probably should be measured in similar terms. Second, preserving data utility while protecting privacy is another pending task. This is, in fact, the most valued parameter by an industry whose revenues are based on the exploitation of data. However, computational cost may be a metric as important, given the demanding requirements of current web applications. Unfortunately, providing privacy generates more distortion, which implies less data utility, while preserving utility usually entails more computing time. Addressing these issues is crucial for an accurate performance analysis of protection mechanisms and

it is fundamental for designing better approaches or choosing the best according to the context.

In this line, many privacy enhancing technologies have been proposed in the literature but there is not a consensus with respect, e.g., to the way to measure empirical utility. Moreover, these approaches include preserving data utility, but at a significant cost in computational cost. And, although some of them aim at reducing the execution time of privacy protection algorithms, the price in terms of distortion is high.

In order to have privacy implemented in practice, it is necessary to face this compromise.

1.1 Objectives

In this dissertation we tackle three main objectives. Firstly, we address the issue of evaluating the real impact of privacy protection on the empirical utility of data; first by performing a systematic study of a standard algorithm; and, secondly, extending this analysis to other, related, mechanisms. We use the accuracy of models learned from perturbed data as utility metric of privacy protection algorithms. On the other hand, we aim at tackling the problem of preserving utility when applying data-perturbative mechanisms. We address this problem by using a machine learning strategy to adapt the privacy protection mechanism to the application domain of data. Finally, we address the issue of computational cost of protection algorithms. For this, we resort to strategies of simplification to speed up their execution, particularly on large data sets.

The objectives of this thesis may be more precisely described as follows:

- **Impact on empirical data utility.** We systematically evaluate the impact of k-anonymous microaggregation on the empirical utility of data. To capture the practical degradation of data utility, we use a metric derived from a popular application domain of data: machine learning. To start, we evaluate the de facto microaggregation algorithm and then other approaches. Different scenarios are tested, including multiple machine learning algorithms and data sets, to

determine how data is affected and whether popular metrics are able to predict such impact.

- **Preservation of data utility.** We design a mechanism to preserve data utility empirically when a data-perturbative algorithm is applied. This mechanism is based on a machine learning technique to enable privacy protection to be adapted to this application domain of data. We try that this effort does not imply an increase in the execution time.

- **Runtime reduction.** We propose and evaluate strategies to significantly reduce the running time for k-anonymous microaggregation. This involves tuning the operations of the privacy protection algorithm to reduce its complexity. Also in this case, we concentrate on preventing additional distortion as a consequence of these approaches.

- As part of our collaboration on a European project, we describe the conception of a privacy protection tool oriented to anonymize cibersecurity data in critical infrastructures. We address the specific challenges of providing privacy in a context where unstructured data is involved.

1.2 Summary of contributions

Next, we give an overview of the major contributions of this dissertation.

- We investigate the impact on the performance of machine-learning tasks caused by data perturbation in the k-anonymous microaggregation process. We apply a rigorous methodology for evaluating the specific impact of microaggregated data on machine-learning tasks. Our methodology uses two standard measures of performance in machine learning and allow for the statistical dependence among quasi-identifiers. We conduct an extensive, thorough evaluation of a wide range of machine-learning algorithms amply used in classification tasks.

- Based on the methodology aforementioned, we evaluate the performance of other k-anonymous microaggregation techniques in terms of the loss in classification accuracy of the machine-learned models built from modified data. Extensive experimentation on four data sets allows us to compare the utility guarantees provided by the most popular microaggregation algorithms.

- We propose and analyze an anonymization method that draws upon a machine learning technique, with the aim of preserving the empirical utility of data. By transforming the original data records to a different data space, this technique enables k-anonymous microaggregation to adapt its operation to the application domain of data. To do this, the representation of data is changed. Interestingly, data utility is preserved without a price in running time.

- We develop five strategies to simplify the internal operations of the maximum distance to average vector algorithm, the de facto microaggregation standard. For the sake of its usability in large-scale databases, they, e.g., reduce the number of operations necessary to compute distances. Also, the complexity of sorting operations gets reduced. Through extensive experimentation over multiple data sets, we show that the new algorithm gets significantly faster. We get resulting algorithm four times faster than the original microaggregation mechanism. This remarkable speedup factor is achieved, literally, with no additional cost in terms of data utility, i.e., it does not incur greater information loss.

- Finally, we build a privacy preserving tool for obfuscating sensitive data from security logs to protect the privacy of the involved entities and individuals. In the context of the CIPSEC European project [13], our proposal includes a methodology to identify and perturb unstructured data generated by a cibersecurity system.

1.3 Related publications

Most of the research results presented in this dissertation have been published in journals. In this section we provide a list of such publications, together with their

bibliographic information. Further, we include other complementary articles that are not directly related with the research topic of this thesis, but in which the author has participated while performing her doctoral studies.

Journal publications:

1. A. Rodríguez-Hoyos, D. Rebollo-Monedero, J. Estrada-Jiménez, J. Forné, and L. Urquiza-Aguiar, "Preserving Empirical Data Utility in k-Anonymous Microaggregation via Linear Discriminant Analysis," accepted to be published in *Elsevier Engineering Applications of Artificial Intelligence*, May 2020. ISSN: 0952-1976. Impact factor 2019: 4.201 [14].

2. A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, J. Parra- Arnau, Ahmad Mohamad Mezher, and J. Forné, "The fast MDAV (F-MDAV) algorithm: An algorithm for k-anonymous microaggregation in big data," *Elsevier Engineering Applications of Artificial Intelligence*, vol. 90, no. 103531, April 2020. ISSN: Impact factor 2019: 4.201 [15].

3. A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, J. Parra-Arnau, and J. Forné, "Does k-anonymous microaggregation affect machine learned macrotrends?," *IEEE Access*, vol. 6, pp. 28 258–28 277, May 2018. ISSN: 2169-3536. Impact factor 2018: 4.098 [16].

4. E. Pallarès, D. Rebollo-Monedero, A. Rodríguez-Hoyos, J. Estrada-Jiménez, A. Mohamad Mezher, and J. Forné, "Mathematically optimized, recursive prepartitioning strategies for k-anonymous microaggregation of large-scale datasets," *Elsevier Expert Systems with Applications*, vol. 144, pp. 1–17, April 2020. ISSN: 0957-4174. Impact factor 2019: 5.452 [17].

5. J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "Online advertising: Analysis of privacy threats and protection approaches," *Elsevier Computer Communications*, vol. 100, pp. 32–51, March 2017. ISSN: 0140-3664. Impact factor 2017: 2.613 [18].

6. J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "On the regulation of personal data distribution in online advertising platforms," *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 13–29, June 2019. ISSN: 0140-3664. Impact factor 2019: 4.201 [19].

7. J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "Measuring Online Tracking and Advertising in Iberoamerica," submitted to *IEEE Access*. ISSN: 2169-3536. Impact factor 2019: 3.745.

Conference publications:

1. A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, J. Forné, R. Trapero, A. Álvarez, and R. Rodríguez, "Anonymizing cybersecurity data in critical infrastructures: The CIPSEC approach," in *Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, Valencia, Spain, May 2019 [20].

2. A. Rodríguez-Hoyos, J. Estrada-Jiménez, L. Urquiza-Aguiar, J. Parra-Arnau, and J. Forné, "Digital hyper-transparency: leading e-government against privacy," in *Proceedings of the 2018 International Conference on eDemocracy & eGovernment (ICEDEG)*, Ambato, Ecuador, June 2018. [21]

3. J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "Measuring Online Tracking and Privacy Risks on Ecuadorian Websites," *IEEE Fourth Ecuador Technical Chapters Meeting (ETCM)*, Guayaquil, Ecuador, November 2019 [22].

1.4 Outline of this thesis

The structure of this dissertation is in line with the research objectives defined in Sec. 1.1.

Chapter 2 describes some of the privacy risks derived from the era of big data as well as some privacy protection mechanisms proposed in the literature, but particularly k-anonymous microaggregation. Relevant concepts regarding the impact of privacy protection on data utility are also reviewed in this chapter.

Chapter 3 presents a systematic evaluation of the impact of MDAV, the de facto standard k-anonymous microaggregation, on the empirical utility of data. This methodology is also used in the following two chapters.

Chapter 4 follows a similar evaluation approach of k-anonymous microaggregation, but more algorithms are considered with the aim of detecting particular strategies that might preserve empirical data utility.

Afterwards, Chapter 5 proposes a mechanism to preserve empirical data utility when applying k-anonymous microaggregation. In this chapter we describe how we apply a machine learning technique to adapt the microaggregation process to the application domain of data.

Chapter 6 presents an approach to accelerate the execution of k-anonymous microaggregation for its application on large-scale data sets.

Lastly, Chapter 7 focuses on the design and implementation of a privacy protection tool for a system managing the security logs of critical infrastructures.

Chapter 2

Background and related work

2.1 Privacy issues in the era of big data

The exponential progress of computing is evident, not only in terms of capacity, including processing or storage, but also in terms of cost. Every five years computers become roughly 10 times more powerful (per constant dollar).

Such trend concurs these days with an equally exponential growth of data generation. Through an ubiquitous telecommunications infrastructure, full of sensors and activity monitors, millions of Internet users enable a massive collection of data, including theirs. But this is not only triggered by users browsing the Web; interactions of users with any single entity (hospitals, banks, social networks, Internet providers, etc.) are susceptible to feed big data.

The availability of big data and the capacity to process it have had a revolutionary impact on the world. Equivalently to a human's deeper observation, exploiting more data would not only enable us to "see" more but new, better, and different data [6]. In fact, as argued in [3], a massive wealth of data may significantly improve the effectiveness of a machine-learning algorithm to the point of turning a hopeless computer model into an expert system.

As expected, a lot of critical applications of big data have proved its positive impact, specially with the emergence of machine learning. One of such applications

has to do with health, where machine learning applied to large data sets may enable, e.g., the detection of subtle effects of some medicines, or even the personalized treatment of a given disease. Namely, based on big data, precision medicine could be implemented to identify which approaches will be effective for which patients based on genetic, environmental, and lifestyle factors.

Although other more commercial applications of big data are supporting billionaire businesses, including online advertising, currently, information is also raw material for other efforts such as scientific research or demographic studies. Due to the intrinsic value of data, currently, any information collected is expected to be released to some point and to some extent with the aim of being exploited.

The personalized nature of most big data applications that users consume implies that tons of *personal information* are required to get effectiveness. That is, the more data items are processed about users, the more accurate personalization services will be (i.e., the more utility could be extracted from information). Said data items might involve several kind of attributes that characterize users in a given context.

In addition, in the current era of big data, information flows involve several entities from different domains interested in extracting as much utility as possible. In fact, the means to process data have become so accessible that even small startups could actively participate in this revolution.

Given the multiple benefits of processing data, its application has spread to all areas and hundreds of services have been implemented to take advantage of it. Consequently, an intense exchange of (personal) information among entities has arisen, which has given rise to a very complex scenario where utility has always been the priority.

This complexity and the indiscriminate exploitation of personal data have lead to serious privacy concerns. Moreover, such a crowded environment has made data more prone to be shared, even openly, among third-parties. Thus, potential malicious "observers" could take advantage of sensitive information encoded within released data.

Unfortunately, the high speed of transactions when data is processed along with the real-time requirements of current web applications have left very little room for

facing privacy issues. In particular, the tight dependence of these applications on data discourages the implementation of privacy protection technologies that may significantly reduce its utility and, thus, the resulting revenues.

The issue, in this context, is that intended recipients of information are not fully trusted, thus conventional mechanisms such as confidentiality through cryptography are not suitable. Data is required to be usable (accessible, at some extent) while some protection for user privacy is provided, i.e., two opposing objectives.

A first approach to protect the privacy of individuals involves suppressing their identifiers, e.g., names, social security numbers, while releasing the rest of attributes. This way, the link between the subject and potential sensitive information, e.g., religion, income or political preference, is apparently broken. However, this strategy may not be enough to protect privacy. It was proven in [23] that three supposedly innocuous attributes (date of birth, gender and 5-digit ZIP code) were enough to unequivocally identify an 87% of the population in the United States in 1990.

Due to the discriminative potential of a few combined demographic attributes, more sophisticated approaches have been proposed to obscure the identity of the subjects represented in a released data set. Since less needs to be learned about users to be anonymized, said approaches usually require distorting the data. Sadly, such distortion of data implies reducing its utility.

Measuring the impact of privacy protection mechanisms on data utility is vital to determine their suitability in practice. If a mechanism is too destructive, it will not be applied on the industry, no matter how well it behaves in theoretical terms. Thus, empirical metrics to asses the expected degradation of utility are also necessary.

In practice, it is evident that the utility of data is increasingly being obtained through the implementation of machine learning algorithms. By extracting intrinsic macrotrends from available data, these algorithms are being used massively to build models that predict outcomes from new data. Being these models the paradigm of data utility extraction, their performance parameters might be interesting indicators when measuring resulting data utility after applying distorting mechanisms.

Finally, in addition to data utility, computational complexity of privacy protection mechanisms is key to have usable privacy. Especially with the advent of big data and

the real-time requirements of web applications, data processing algorithms have to be faster than ever to satisfy such requirements. Thus, optimizing their implementation could be a great incentive for the adoption of privacy enhancing initiatives.

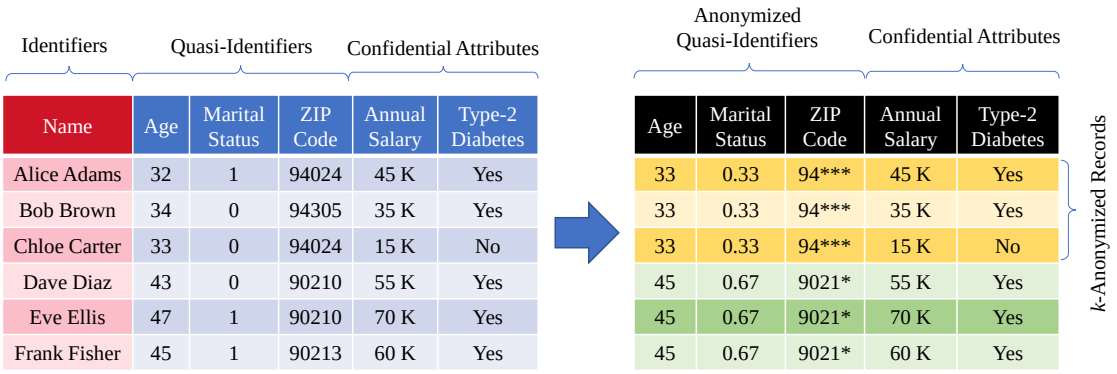
2.1.1 Data release and attacker models

When analyzing data privacy,it is important to define how data about individuals is represented. Standardizing such representation along with establishing privacy and utility metrics enables the construction of a common framework where different protection approaches are suitable to be evaluated and compared.

In general, privacy protection is applied on databases carrying information about individual respondents, e.g., from a survey or a census. The resulting databases (also known as microdata sets) contain a set of attributes that may be classified into identifiers, quasi-identifiers and confidential attributes.

Firstly, *identifiers*, such as full names or medical record numbers, can single out individuals from a data set, so are commonly removed in order to preserve the anonymity of respondents. Secondly, *quasi-identifiers* may include demographic attributes such as age, gender, address, or physical features, which combined and linked with other external information can be used to reidentify respondents [8, 9, 23]. Finally, a data set may contain *confidential attributes* with sensitive information on the respondents, such as salary, health condition, and religion. These sensitive attributes might be easily linked to the subjects to whom the disclosed information corresponds if quasi-identifiers are not adequately obfuscated; said disclosure might lead to discrimination, retaliation, and blackmail [10]. In Fig. 2.1, the table on the left illustrates an example of this representation and the different types of attributes here described.

With the aim of protecting privacy, then, only quasi-identifiers and confidential attributes should be released. Confidential attributes could be released as is since the link between them and the subject is supposed to be broken by the suppression of identifiers. However, quasi-identifiers, given its reidentification potential, have to be carefully obfuscated while preserving some of its utility.



Name	Age	Marital Status	ZIP Code	Annual Salary	Type-2 Diabetes
Alice Adams	32	1	94024	45 K	Yes
Bob Brown	34	0	94305	35 K	Yes
Chloe Carter	33	0	94024	15 K	No
Dave Diaz	43	0	90210	55 K	Yes
Eve Ellis	47	1	90210	70 K	Yes
Frank Fisher	45	1	90213	60 K	Yes

Age	Marital Status	ZIP Code	Annual Salary	Type-2 Diabetes
33	0.33	94***	45 K	Yes
33	0.33	94***	35 K	Yes
33	0.33	94***	15 K	No
45	0.67	9021*	55 K	Yes
45	0.67	9021*	70 K	Yes
45	0.67	9021*	60 K	Yes

Figure 2.1: Example of k-anonymous microaggregation of published data with k=3. Quasi-identifiers in the left table are anonymized on the right.

2.2 Privacy protection through k-anonymous microaggregation

2.2.1 Statistical disclosure control

Beyond the mere suppression of subjects' identifiers, *statistical disclosure control* (SDC) aims to allow useful inferences about subpopulations from a microdata set while at the same time protecting the privacy of the subjects who contributed their data.

Microdata are database tables whose records carry data concerning individual subjects. The typical scenario in microdata SDC is a data curator holding the original data set and perturbing the so-called *quasi-identifier* attributes (i.e., attributes that, in combination, may be linked with external information to reidentify individuals in the data set). The goal is to keep disclosure risk as low as possible, while ensuring that only useful statistics or trends are learned by the recipients of data. One of the most common strategies to keep this risk under control is the "privacy first" approach. Here, the data curator enforces a privacy model, which usually depends on a privacy parameter, to ensure an upper bound on the re-identification risk.

Some of the best-known privacy models comprise k-anonymity [11, 23] and ε-differential privacy [12].

Privacy models rely on a variety of anonymization mechanisms, the common denominator binding all them is *data perturbation*. Essentially, all such mechanisms modify the original data set to guarantee the chosen privacy model, inevitably at the cost of some loss in data utility [24]. Evidently, a balance between privacy and utility should be found so that protected data are useful in real practice, that is, they approximate well the original data. However, in a big data domain, privacy protection also requires mechanisms to execute in a reasonable amount of time, despite the size of the data.

Examples of privacy protection approaches include microaggregation, suppression, generalization and noise addition. Among them, k-anonymous microaggregation is a high-utility approach.

2.2.2 k-Anonymity

k-Anonymity guarantees the privacy of an individual by making her quasi-identifying attributes indistinguishable from those of other $k-1$ individuals in a microdata set. More specifically, k-anonymity is a privacy model that guarantees that each tuple of quasi-identifying values is identically shared by at least k records in a data set.

Thus, rather than making the original table available, a perturbed version of quasi-identifiers is published where aggregated records of quasi-identifying values are replaced by a common representative tuple. If every tuple shares quasi-identifying values with at least k records, the data set is considered k-anonymous [23].

Figure 2.1 depicts how a data set is transformed to satisfy k-anonymity. This way, a perturbed, more private, version of a data set is obtained to be published instead of the original one. In the figure, the original data set combines attributes common in census and medical surveys. It has three quasi-identifiers: age, marital status and ZIP code; and two confidential attributes: annual salary and type-2 diabetes condition. The figure at hand shows how, in order to preserve the privacy of respondents, k-anonymity is enforced by applying perturbation to quasi-identifiers. The technique applied here is called *microaggregation*. The result is a microaggregated data set that may prevent reidentification attacks.

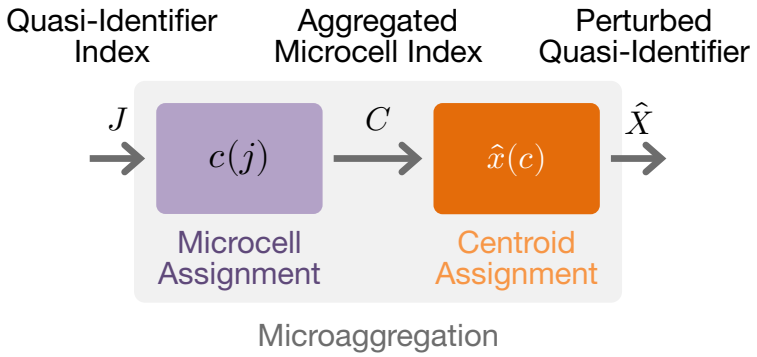


Figure 2.2: Block diagram of k-anonymous microaggregation as a two-step process [1].

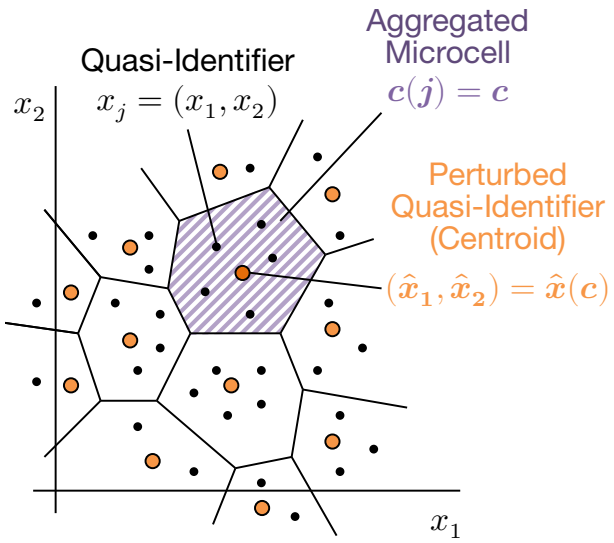


Figure 2.3: Illustration of the process of k-anonymous microaggregation as a minimum-distortion quantizer design problem [1].

2.2.3 k-Anonymous microaggregation

Microaggregation is a technique aimed to protect the privacy of those individuals whose personal records are included in a released microdata set. With microaggregation, distortion is applied to quasi-identifying attributes to satisfy the k-anonymity privacy model [11, 23]. The original formulation of k-anonymity as a privacy criterion was modified into the microaggregation-based approach in [25–28].

In Fig. 2.2, a block diagram describes microaggregation as a two-step process including microcell assignment and centroid assignment. Accordingly, each record (its

quasi-identifying tuple) is first grouped in a cluster with other $k-1$ records. Then, within each cluster of size k, a centroid, representative of such cluster, is calculated and assigned to each record. The version released of the data set involves the values of the centroids calculated for each quasi-identifying tuple.

This process is graphically illustrated in Fig. 2.3. If tuples of quasi-identifiers in a data set could be represented as points in the Euclidean space, k-anonymous microaggregation would consist in partitioning these points in cells of size k, and quantizing each cell and its elements with a representative point. Perturbed key attributes would be characterized by the set of representative points.

2.2.4 Maximum distance to average vector

The maximum distance to average vector algorithm (MDAV) is the de facto standard for numerical microaggregation. It was proposed in [29] as a practical evolution of a multivariate fixed-size microaggregation method and conceived in [26]. We provide, in Algorithm 1, a simplified version of that given in [27] and termed "MDAV generic".

Seen a data set as a list of points in $\mathbb{R}^n$, MDAV is an iterative process that starts by finding the centroid C (calculated as the mean) of the points not yet assigned to a microcell. Then, points P and Q are found as the furthest point from C and the furthest point from P, respectively. Two corresponding microcells are built by grouping P and Q with their $k-1$ nearest points. This process is repeated while $2k$ points or more in the data set remain to be assigned to microcells.

Finally, to be released, the data set is reconstructed replacing the quasi-identifying values of each record with the centroid of the microcell they belong to. This centroid is calculated as the mean of the quasi-identifiers of the microcell.

2.2.5 Reconstruction mechanisms

The way of representing records for each resulting microcell is also important to preserve the utility of data. For MDAV, when having numerical microdata, we have chose to use the average of the quasi-identifying tuples as centroid and, thus, as representative tuple.

Algorithm 1 MDAV "generic", functionally equivalent to Algorithm 5.1 in [27]

function MDAV
input k, $(x_j)_{j=1}^n$ ▷*Anonymity parameter k, quasi-ID portion $(x_j)_{j=1}^n$ of a data set of n records*
output q ▷*Assignment function from records to microcells $j \mapsto q(j)$*
while $2k$ points or more in the data set remain to be assigned to microcells **do**
  find the centroid (average) C of those remaining points
  find the furthest point P from the centroid C, and the furthest point Q from P
  select and group the $k-1$ nearest points to P, along with P itself, into a microcell, and do the same with the $k-1$ nearest points to Q
  remove the two microcells just formed from the data set
if there are k to $2k-1$ points left **then**
  form a microcell with those and finish
else ▷*At most $k-1$ points left, not enough for a new microcell*
  adjoin any remaining points to the last microcell ▷*Typically nearest microcell*

However, besides numerical microaggregation, other anonymization mechanisms can be used to implement data perturbation. These mechanisms include suppression, generalization and noise addition. Those could be used indistinctly, depending on the type of data (e.g., numerical, categorical, ordinal, string), and on its expected utility. In our work, we mostly deal with numerical data although for some data sets we transform some textual or ordinal to numerical attributes.

2.2.6 Other privacy criteria

Although k-anonymity is a very popular privacy criterion, it is not flawless. Since the criterion strictly operates with quasi-identifying attributes, the statistical properties of confidential attributes (and thus their disclosure potential), both in the data set and in the entire population, are neglected. In general, k-anonymity overlooks the knowledge a potential attacker may already have or obtain about the data set, giving rise to similarity, skewness or background-knowledge attacks [30–32].

In spite of its shortcomings, the application of k-anonymous microaggregation does not only concern the publication of databases but also some variants thereof like search engine querying, online data collection and data streaming [33–35].

Additional criteria have been proposed that refine k-anonymity and prevent some of the above-mentioned attacks. The former, *p-sensitive* [36, 37], requires that each group of k-anonymized records contains at least p different values of each confidential

attribute. In the same but broader spirit, *l-diversity* proposes that each group have at least l well-represented confidential values. None of these criteria assures complete protection against skewness attacks, nor against similarity attacks when confidential attributes within a group are semantically similar.

Other privacy criteria dealing with similarity and skewness attacks pose requirements in the distribution of confidential attributes within groups. The aim is that confidential attributes in each group of anonymized records are stratified according to their distribution in the original data set. Depending on the discrepancy allowed between the within-cluster and overall distributions, these privacy criteria yield t-closeness [38], δ-disclosure [39], and average privacy risk [31, 40].

To cope with the NP-hardness of multivariate microaggregation, several heuristic algorithms have been proposed. These algorithms can be classified as fixed-size and variable-size. Among the former ones, we find the maximum distance [26] (MD) and its variation, maximum distance to average vector [26, 27] (MDAV). Variable-size algorithms include, on the other hand, the μ-Approx [28], the minimum spanning tree [41] (MST), the variable MDAV [42] (V-MDAV) and the two-fixed reference points algorithms (TFRP).

In general, the implementations of microaggregation have been oriented to reduce the inherent information loss [43–45] due to perturbation, which commonly derives in more sophisticated and significantly costlier implementations in terms of computational time [1].

2.3 Impact of microaggregation on data utility

k-Anonymous microaggregation, as any data perturbation mechanism, implies distortion or information loss on the data since original data is modified. Measuring such impact is fundamental for assessing the performance of this and other privacy protection mechanisms.

2.3.1 A syntactic metric based on mean squared error

The usual criterion to quantify the distortion of microaggregated data is the mean-squared error (MSE) for numerical quasi-identifying attributes representable as points in the Euclidean space. MSE can be computed as

$$\text{MSE} = \sum_{j=1}^{n} \|x_j - \hat{x}_j\|^2 ,$$

where n is the number of records of the data set, m is the number of attributes of each record, $x_j \in \mathbb{R}^m$ is the j^{th} record, and $\hat{x}_j$ is the tuple representative of the j^{th} record.

As can be seen, MSE measures the numerical variation of records after data perturbation is applied. Although such variation may provide an idea about the magnitude of utility degradation, it does not consider the global macrotrends within the data set and, more important, neglects the application domain of data where its utility is exploited. This is the reason why we refer to MSE, or distortion, as a syntactic measure of data utility.

2.3.2 Machine learning parameters as a semantic metric

While MSE, as a measure of data distortion, is the general metric of the degradation of data utility after microaggregation, probably, other more practical metrics are required to evaluate the real impact of this privacy protection mechanism. A more empirical approach necessarily involves the application domain of data, i.e., the process by which utility is extracted in practice. Besides, it is reasonable to think about adapting privacy protection mechanisms to the application domain of data such that more utility is preserved while offering similar levels of privacy.

Undoubtedly, one of the most common ways of data exploitation currently is machine learning. Learning algorithms are widely used to build models (from data) capable of predicting an outcome when applied to new data.

In an effort to tailor anonymization mechanisms to the application domain of data (e.g., building classifiers to predict someone's health condition), some previous

research work has used empirical utility metrics. One of such metrics is the *accuracy* of machine-learned macrotrends built using anonymized data.

The logic is simple: a learning model built with perturbed data would be less accurate than another built with original data. Accordingly, a higher degree of anonymization would result in less accurate models. Surprisingly, to the best of our knowledge, this metric has not been used to systematically evaluate microaggregation-based anonymization algorithms, but other anonymization algorithms based on generalization and suppression of records, such as Incognito, Mondrian and DataFly.

In previous work, classification accuracy has been used to evaluate the utility of (or, equivalently, the distortion introduced to) anonymized data, just to compare the performance of adapted classifiers or anonymization mechanisms. One of these works is [46], where the effects of four microaggregation algorithms on the estimation of a linear regression is compared, when solely applied to simulated data sets. Other works propose improvements on machine learning algorithms and methodologies, to obtain higher utility (classification accuracy) from anonymized data. This is the case of [47], where the authors develop a method to increase the level of utility obtained from support vector machine (SVM) and k-nearest neighbor (kNN) machine learning algorithms, when data are anonymized with the DataFly algorithm. By feeding these algorithms with statistics from original data, in addition to anonymized data, greater utility ensues from the latter. In the same line, [48] describes an adjustment to logistic regression that provides differential privacy [12]. Furthermore, decision tree learning methods are developed in [49] and [50] that enforce l-diversity and differential privacy, respectively, as privacy criteria and whose accuracy levels approach those of a non-private decision tree. Using a different focus, [51] and [52] address the privacy risk resulting from the release of SVM and the anonymized data. Privacy preserving versions of SVM are proposed and their classification accuracies are used to compare them with the original SVM.

A great deal of research has also investigated adaptations of anonymization algorithms that generate private data of "higher quality". In that context, the utility of anonymized data is evaluated in terms of classification accuracy of machine learning models [53], [54], and [55]. The cited works rely on generalization and suppression as

perturbation techniques and include preprocessing steps such as selective anonymization of attributes, to adapt the released data to machine learning applications, and hence preserve their utility. On the other hand, [56] proposes publishing synthetic microdata generated from differentially private models applied on original data. For that, machine learning techniques are integrated to improve utility.

Ironically, although enhancements in the utility of anonymized data are reported, it is not clear what the overall impact of original anonymizing mechanisms in the first place is. Some approaches do attempt to evaluate the tradeoff between privacy gain and information loss (measured as accuracy reduction) due to anonymization. However, various considerations should be done for such evaluation. To start, there is a variety of anonymization algorithms. For example, [57] focuses on a proprietary anonymization algorithm whereas [58] examines a non-standard one.

Other caveat is the variable application domain of the data. While classification is the most popular workload for anonymized data, machine learning algorithms would perform differently depending on the particular data set used, so the utility would vary accordingly. This also applies to the number of records, or the size of the data set, which may affect the performance of anonymization algorithms, e.g., when k-anonymity is applied, a given value of k shall affect the utility of small data sets more than the utility of bigger ones.

A last limitation has to do with the baselines to measure privacy gain and utility loss. Utility, measured as the accuracy of machine learning models, reaches its lower bound when all the key attributes are discarded; or, for k-anonymity, when k equals the number of records of the data set. Utility's upper bound is attained when no anonymization is applied[(a)].

Even in this variable scenario, one thing is certain about how machine-learned trends are affected by anonymization: simultaneously satisfying various privacy criteria, e.g., k-anonymity, l-diversity, and t-closeness, may make the data completely useless, as reported by [39], a study where not only syntactic but also semantic requirements of privacy are evaluated. Those privacy criteria, together with differential privacy, are out of the scope of this work, since our target application is that of

[(a)] Further considerations regarding baseline performance can be found in [59].

data release for general statistical analysis with a focus on data utility. Recall that differential privacy is conceived for online querying on predefined computations, and that in general it imposes stringent restrictions, both in terms of usability and data utility. Those restrictions, introductorily explained also in [60], render it useless for our purposes.

The review of the state of the art in this section has been conducted from a strictly technological perspective. Legal and socioeconomic aspects are covered, for instance, in [61, 62]. Table 2.1 summarizes some of the main contributions where machine learning performance parameters are used to measure the utility degradation of data after applying privacy protection mechanisms.

Table 2.1: Contributions using machine learning performance as metric of privacy protection impact.

Reference	Anonymization algorithm	Type of attributes used	Application domain	Max size of data sets	Max value of k	Main focus
Inan et al, 2009 [47]	DataFly	Hybrid	Classification	5,000	128	Comparing classifiers on anonymized data
LeFevre et al, 2006 [53]	Mondrian, TDS	Hybrid	Classification	49,657	1,000	Algorithms to anonymize data while preserving utility
Chaudhuri and Monteleoni, 2008 [48]	Differential Privacy	Numeric	Classification	N/A	N/A	Improving ML algorithm to work with anonymized data
Lin and Chen, 2010 [51]	DataFly	Numeric	Classification	270-49,990	128	Improving ML algorithm to work with anonymized data
Kisilevich et al, 2010 [55]	kACTUS, TDS, TDR, Mondrian, kADET	Hybrid	Classification	42,244	1,000	Building an algorithm to protect privacy in classification tasks (comparing accuracy with others)
Jaffer et al, 2014 [54]	Mondrian	Hybrid	Classification	1,000	50	Building an algorithm to protect privacy in classification tasks (comparing accuracy with others)
Malle et al, 2016 [58]	SaNGreeA	Hybrid	Classification	42,244	19	Showing the destructive effect of an anonymization algorithm on classification tasks
Gursoy et al, 2017 [57]	k-Map	Hybrid	Classification	42,244	5	Evaluating an anonymization algorithm based on differential privacy
Brickell and V Shmatikov, 2008 [39]	Mondrian	Hybrid	Classification	42,244	1,000	A methodology to measure the tradeoff between loss of privacy and gain of utility

2.4 Impact of microaggregation on data usability

Although the extraction and preservation of data utility are certainly important when designing applications either to exploit or protect data, there are other requirements to make sure such applications are suitable in practice. Evidently, such requirements are commonly inherited from the technological applications to which privacy protection mechanisms support.

Although data is the new oil in the big data era, applications of big data are currently possible just because the algorithms that process it can be executed much faster than in the past. However, the execution time is still a bottleneck for some highly demanding applications, which does not favor the implementation of further processing privacy routines. Consequently, we could say that accelerating the execution of privacy protection mechanisms is a fundamental approach to encourage its adoption in the era of big data.

Recent works have shown to follow approaches to increase the efficiency of privacy protection algorithms, not only in terms of runtime [63–65], but also in terms of resulting data utility. For instance, [41] developed an efficient clustering mechanism to deal with large databases while preserving the data utility through a partitioning method of a modified minimum spanning tree. In the same line, [66] designed an efficient and effective microaggregation mechanism based on calculating the distance among records as the mutual information (entropy) among them. Finally, [67] introduced fast data-oriented microaggregation (FDM), a method capable of getting multiple protected versions of a large data set (for different values of k) in a single load.

Note that all these approaches offer a reduction of runtime for privacy protection mechanisms at a cost in data distortion. Thus, there is another trade off that should be actively tacked to guarantee that privacy protection can be implemented in real scenarios.

Chapter 3

Impact of MDAV on the empirical utility of data

3.1 Introduction

The permanent and increasing interactions of people (both conscious and unconscious) with the Internet trigger the disclosure of tons of personal data. Besides, as discussed in Sec. 2.1, the great utility that can be extracted from data encourages its exploitation by thousands of third-parties. Naturally, serious privacy concerns arise from such practice.

To face the potential privacy threats, several protection mechanisms have been proposed in the literature; one of them is k-anonymous microaggregation, whose basic elements were described in 2.2.3 . Although the level of privacy protection it offers is clearly defined, there is an issue with measuring its real impact on data utility. In this line, we discussed in Sec. 2.3 the problem that merely syntactic metrics may not reflect the impact of data perturbation, in terms of utility, if these metrics are not tightly linked with the mechanism through which data utility is extracted. Therefore, an exploration to find an empirical metric of data utility would enable a more accurate evaluation of the impact on utility, and would assist researchers in building utility-preserving privacy protection mechanisms.

As any perturbative mechanism, anonymization comes at the cost of some information loss that may hinder the ulterior purpose of the released data, which very often is building machine-learning models for macrotrends analysis.

In this chapter we propose to assess the impact of microaggregation on the utility of anonymized data by calculating the resulting accuracy of said models. In particular, we address the problem of measuring the effect of k-anonymous microaggregation on the empirical utility of microdata. For this, we quantify utility as the accuracy of classification models learned from microaggregated data, and evaluated over original test data. In a nutshell, our approach seeks to validate whether this impact is major and, accordingly, whether the metric of distortion (based on MSE) is concordant with a more empirical vision of data utility.

We apply a rigorous methodology for evaluating the specific impact of microaggregated data on machine-learning tasks. Our methodology uses accuracy and F-measure as utility metrics. The two are standard measures of performance in machine learning and allow for the statistical dependence among quasi-identifiers. The impact of microaggregation on the utility of anonymized data is quantified, accordingly, as the resulting accuracy (or F-measure) of a machine-learning model trained on a portion of microaggregated data and evaluated on a different portion of original data.

Since the utility extracted from data could depend on the learning algorithm used, the results of utility we present correspond to the algorithms that obtain the greatest accuracy from each anonymized data set. Among others, our experiments investigate naïve Bayes, logistic regression, SVM, bagging and C4.5. As for microaggregation algorithms, we focus on MDAV, the SDC de facto standard for k-anonymous microaggregation. The evaluation of MDAV and all those machine-learning algorithms is conducted in four data sets, three real and one synthetic.

Note that this analysis focuses on high-utility SDC, which involves plain k-anonymous microaggregation using *numerical* microdata. Although more strict privacy criteria exist, e.g., in the domain of syntactic microaggregation (such as t-closeness or l-diversity), or in the domain of semantic privacy (such as differential privacy), only

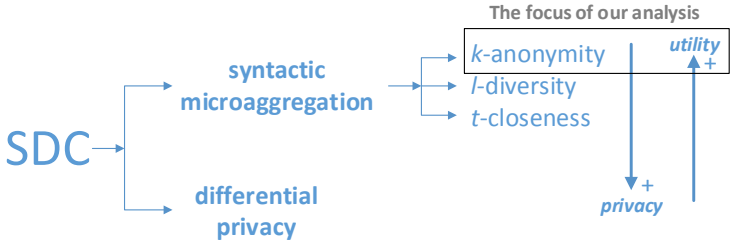


Figure 3.1: Our work focuses on high-utility SDC, involving k-anonymous microaggregation, which has a direct application, e.g., in the health domain.

privacy mechanisms offering greater utility guarantees for anonymized data are examined, which may be highly desirable in domains like health. This analysis context is illustrated in Fig. 3.1.

The work presented in this chapter was published in [16].

Chapter outline

The rest of this chapter is organized as follows. Section 3.2 describes our experimental methodology. Section 3.3 shows the experimental results obtained for a variety of data sets and machine-learning algorithms. Lastly, conclusions are drawn in Section 3.4.

3.2 Methodology of evaluation

When assessing a privacy protection mechanism, defining the assumptions considered is fundamental to provide a systematic and repeatable analysis. In addition, the details of personal data release, as well as the applications used to exploit it, may vary from case to case, even more in the changing technological world we live in now. Then, it is convenient to clarify the particular scenario for which our approach is valid.

Next we describe the elements of this evaluation scenario. While in Section 2.1.1 we briefly introduced attacker and data release models, here we also include the usability model in Section 3.2.1 where we illustrate by example the practical context where our evaluation has sense. Moreover, the privacy and utility metrics we use in this chapter and along the rest of this work are defined in Section 3.2.2.

Finally, the tools, data sets used, and the methodology followed are described in Sections 3.2.3 and 3.2.4.

3.2.1 Attack and usability model

In this chapter, and in the next ones, we assume the standard attack model of the SDC literature [68]. When a microdata set is released, it is assumed to be available to any privacy attacker. For research and statistical purposes, released microdata contains key attributes (basically, demographic data) that are correlated with another, probably confidential, attribute. In the k-anonymity model, besides, the attacker knows that a target individual's record –although microaggregated– is in the released data set.

To protect that individual's privacy, an anonymized version of the microdata set is released. To keep the information usable, i.e., "truthful" [11], microaggregation is applied to key attributes, while the confidential attribute keeps unperturbed. Researchers may leverage the key attributes by building classifiers on the microaggregated data, for example to predict a given condition. Recall that classification is a machine learning task that aims to predict the class, or label, of a tuple of information. To do so, it requires learning a model from a group of labeled input samples. In our case, we can assume a large anonymized data set of patients that is publicly released so that researchers can build classifiers.

As another example of this model, suppose that the taxation authority publishes a microaggregated data set with 3 key attributes: gender, age, and marital status; additionally, a confidential binary attribute is published without being modified, specifying whether a respondent has paid taxes or not. Both perturbed key attributes and the confidential attribute could be used by researchers to develop algorithms that predict the propensity of other people to pay taxes. At the same time, the privacy of a specific individual would be preserved as a result of microaggregation. However, as commented in previous sections, the macrotrends embedded in the original data, which are necessary to get more accurate classifiers, might be affected by the perturbation of the key attributes values caused by microaggregation.

3.2.2 Measuring privacy and utility

To evaluate the impact of anonymization on the utility of a released microdata set, quantifiable metrics of privacy and utility are required. Since our experiments focus on microaggregation as anonymization mechanism, we assume k-anonymity as privacy criterion. In this manner, the identity of a respondent will be protected in a group of k tuples sharing the same key attribute values. Higher values of k will imply more anonymity and then more privacy, although, eventually, less utility.

To measure the utility of anonymized data, we must decide the application domain of such data. We choose *binary classification* since it is a very popular workload for released microdata sets. Accordingly, we measure utility through the performance of a binary classifier, when executed on anonymized data. Several metrics exist that measure the performance of binary classification tests. Next, we elaborate on them with a medical example.

Let D be a binary random variable (r. v.) representing whether a patient has a given condition ($D = 1$) or not ($D = 0$). Let T be a binary r. v. representing the outcome of a medical test, being $T = 1$ a positive detection, and $T = 0$ a negative detection. By the law of total probability,

$$\mathrm{P}\{T = D\} = \mathrm{P}\{T = D \mid D = 0\}\,\mathrm{P}\{D = 0\} + \mathrm{P}\{T = D \mid D = 1\}\,\mathrm{P}\{D = 1\},$$

and thus,

$$\mathrm{P}\{T = D\} = \mathrm{P}\{T = 0 \mid D = 0\}\,\mathrm{P}\{D = 0\} + \mathrm{P}\{T = 1 \mid D = 1\}\,\mathrm{P}\{D = 1\}.$$

Specificity (true negative rate) and *sensitivity* (true positive rate) are two metrics of the performance of a binary classifier and can be defined as $\mathrm{P}\{T = 0 \mid D = 0\}$ and $\mathrm{P}\{T = 1 \mid D = 1\}$, respectively. In our evaluation, we follow the same approach as [53, 55, 58] and measure utility as the *accuracy* of a binary classifier. In our example, accuracy can be defined as the probability that the test and disease coincide, that is $\mathcal{A} = \mathrm{P}\{T = D\}$. Accuracy can also be expressed in terms of specificity and sensitivity as the convex combination

$$\mathcal{A} = (1 - \text{prevalence}) \times \text{specificity} + \text{prevalence} \times \text{sensitivity}$$

weighted by the prevalence, that is, the a priori probability of having a disease.

Although accuracy is a very popular metric, when the class of the data is significantly unbalanced this metric might incorrectly measure the goodness of a classifier. Fortunately, other stricter indicators are available such as F-measure, ROC curve and area under the ROC curve (AuC).

Accuracy quantifies how well a binary classifier performs, in terms of the rate of correctly classified (as positive or negative) samples in a test set. For example, a binary classifier constructed to predict diabetes would be 100% accurate if, when applied on a test set of 600 samples, it correctly identifies the class of the 500 samples labeled with "no diabetes" and the class of the 100 samples labeled as "diabetes".

F-Measure (or F_1 score) is a machine learning metric that combines other metrics, particularly recall and precision. In fact, F-Measure is defined as the harmonic mean of precision and recall. Furthermore, another composed metric is the ROC curve, which measures the performance of a classifier based on the graphical representation of the sensitivity in function of the specificity.

For our application domain (binary classification), we first measure the utility of a microdata set before being microaggregated. Since no perturbation is applied to the data, the classifier built from that data set would yield the highest accuracy. The data would therefore give the best achievable utility, but the worst privacy.

In our experiments, we generate several microaggregated versions of a data set, by varying the value of the privacy parameter k incrementally for a wide range. For each of these versions, we compute the corresponding classification performance to observe the progressive degradation of data utility due to microaggregation. We use accuracy and F-measure to assess the performance of classifiers built with microaggregated data. Naturally, as k increases, we expect a lower data utility, but obviously in exchange for higher privacy. Note that, for binary classifiers computed over a set of data samples and their corresponding labels, the lowest possible accuracy is not zero. To see this, suppose that "positive" is the majority class (more than 50% of the training samples are labeled as "positive"). Accordingly, the simplest classifier would classify any new input as "positive". Then, interestingly, a binary classifier should not have accuracy values lower than 50%.

3.2.3 Experimental setup

Next, we describe the algorithms, tools and data we use to quantify the impact of k-anonymous microaggregation on the performance of machine-learned classifiers.

3.2.3.1 Algorithms

With regard to microaggregation, our experiments employed MDAV [69], the de facto standard protocol described in Section 2.2.4.

With the aim of constructing classifiers from microdata, we used the *Weka toolkit* [70], a collection of algorithms extensively employed by the machine learning community. In the interest of fairness when measuring the impact of microaggregation, we assigned each data set the machine learning algorithm that extracts the greatest utility from it. Accordingly, we measured said impact with respect to the highest achievable utility. In order to find the corresponding algorithm for a data set, we tried on it a range of machine learning algorithms, including naïve Bayes, logistic regression, SVM, bagging, and C4.5. The reasons for choosing this set is manifold. First, we included different algorithms to observe whether the effects of microaggregation are consistent along different utility extraction techniques. Moreover, we selected naïve Bayes and SVM since in several previous works [53, 55, 56, 58] they have been adapted to obtain more utility from anonymized data. Additionally, logistic regression, C4.5 and bagging were considered to represent the main families of machine learning classifiers, i.e., regression, decision tree, and ensemble algorithms, respectively. For each data set, we chose the algorithm showing the best performance in the classification task, i.e., the highest accuracy. This way, we tested the impact of microaggregation in the different utility contexts or domains defined by a variety of data sets and machine learning algorithms.

3.2.3.2 Data

For the purpose of illustration, we evaluated the impact of microaggregation first on a synthetic data set. The effect of microaggregation on real scenarios was assessed afterwards in data sets satisfying these four properties. First, we require data sets

containing demographic attributes so that they reflect the typical characteristics of microdata. Secondly, we considered only data sets whose potential key attributes are correlated with a given sensitive (label) attribute, so the latter could be effectively predicted (classified). Thirdly, we needed a relatively large number of records (e.g., more than 500) to have a better view of the overall effect of microaggregation, using an incremental value of the privacy parameter k. Finally, we used standardized or already tested data sets so that our results could be easily reproduced. It is worth noting that predictive demographic data turned out to be a very restrictive condition when we searched for data sets to carry out the tests.

For the sake of simplicity and ease in its graphic representation, we built the synthetic data set with only two numerical attributes (x_1, x_2) resembling quasi-identifiers, and a binary attribute (y) as the confidential attribute. The data set was generated so that y is to some extent predictable from x_1 and x_2 and had 30,000 records. In Sec. 3.3.2, we describe in greater detail the process by which the synthetic data set was generated and show a preliminary experiment to illustrate the effects of microaggregation.

Regarding the experiments on real data sets, we first employed the standardized "Adult" data set from the UCI Machine Learning Repository [71], described in Table 3.1. The data set in question has been widely used to evaluate binary classifiers and privacy preserving mechanisms. Its 45,222 records are already split into two parts, for training (2/3) and testing (1/3) purposes. The data set contains 15 input demographic attributes and a binary label attribute, the salary, which is the attribute the machine learning algorithm tries to predict. In particular, the attribute specifies whether a person makes over 50K a year or not. The attributes we use as quasi-identifiers are age, education-num, marital-status, sex, capital-gain, and hours-per-week.

The second standardized data set was "Pima Indians Diabetes" [72] which contains 768 records and 9 demographic attributes. Available at the UCI Machine Learning Repository, this data set has been used in [51, 54, 55]. The 8 attributes we selected allows predicting whether an individual will be diagnosed with diabetes or not. The third real data set we considered in our experiments was the "Irish Census" [73], a synthetic version of the data from the 2011 Irish Census, which has been used in [74]

and [75] to evaluate and compare k-anonymization algorithms. It contains 100,000 records and 10 demographic attributes. Originally, it was not built with a predictive task in mind, but 5 of its attributes could be used to predict an individual's economic status (employed or unemployed).

Table 3.1 describes the main characteristics of the data sets tested in our experiments, and Table 3.2 shows the machine learning algorithms employed for each data set.

Table 3.1: Description of the Data sets Used to Evaluate the Impact of k-Anonymous microaggregation

Data set	**# of instances**	**# of attributes**	**Selected key attributes**	**Confidential (label) attribute**
Synthetic	30,000	2	x_1, x_2	y
Adult [71]	45,222	15	Age, education-num, marital-status, sex, capital-gain, hours-per-week	Salary (>50K?)
Pima Indians Diabetes [72]	768	9	Number of pregnancies, glucose concentration, blood pressure, triceps skin fold thickness, serum insulin, body mass index, diabetes pedigree function, age	Health condition (diabetes?)
Irish Census [73]	100,000	10	Gender, age, marital status, highest education completed	Economic status (employed?)

Table 3.2: Machine learning algorithms used in our experimental evaluation

Data set	**ML algorithm used**		**ML algorithm description**
	Type	**Name**	
Synthetic	Classification tree	C4.5	It builds decision trees from training data, where attribute nodes are selected based on their information gain (mutual information).
Adult [71]	Ensemble	Bagging	Bootstrap aggregation is an ensemble of decision trees that improves classification tasks by combining the classification results of randomly (bootstrap) generated training data sets obtained from the original data set.
Pima Indians [72]	Regression	Logistic Regression	It is a regression model that probabilistically estimates a binary response (binary classification) based on a set of predictors. It is based on the logistic or sigmoid function.
Irish Census [73]	Classification tree	C4.5	It builds decision trees from training data, where attribute nodes are selected based on their information gain (mutual information).

3.2.3.3 Additional Tasks

Since our implementation of MDAV only operates with numerical attributes, we conducted some preprocessing tasks on the data sets described in the previous subsection. Specifically, we converted some useful categorical attributes to numeric, where possible, and binarized the sensitive attribute, where necessary, so that the application domain of data was binary classification.

3.2.4 Experimental methodology

The steps we follow to evaluate the impact of microaggregation on the utility of microdata are in line with the attack and utility models described at the beginning of Section 3.2.1 and are illustrated in Fig. 3.2. As a first data preprocessing step, we extract the quasi-identifiers of our interest from each data set, according to the guidelines described in the previous subsection. Moreover, from the selected attributes, we "numerize" the categorical data so that they are compatible with MDAV. Finally, we identify the quasi-identifiers that are then used as input samples and the sensitive label attribute that will serve as the class to be predicted by the classification model.

The next step splits each microdata set into training and test sets. As is common in the evaluation of machine learning algorithms, a model is constructed from a training subset of the data and is evaluated on the test subset. Following such methodology, we use two-thirds of the data for training and one-third for testing. The splitting is done in such a way that the class attribute is stratified in each subset, according to its original distribution in the data set.

After splitting the data into training and test sets, the microaggregation process is performed using MDAV over the latter set. To this end, previously we followed the common practice of normalizing each column of the data to have zero mean and unit variance.

With the microaggregated versions of each (training) data set, we then construct a classification model over each of those versions using Weka and 5-fold cross validation. The learning algorithms we use for each data set are listed in Table 3.2. Finally, we evaluate the accuracy of the resulting classification models over the non-anonymized

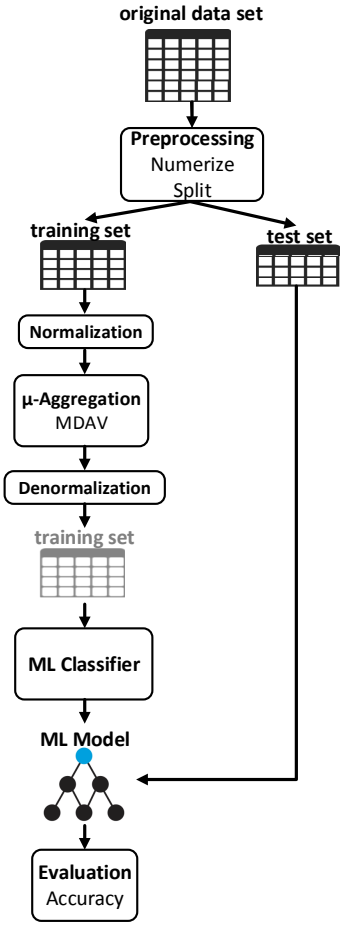


Figure 3.2: Experimental methodology followed to evaluate the impact of MDAV-based k-anonymous microaggregation on the empirical utility of microdata.

test subset, reproducing the application scenario where a database user would use the classification model to classify their original samples of data.

3.3 Experimental results

3.3.1 Preliminary experiment

To get some intuition about the impact of microaggregation and its clustering capability on the empirical utility of anonymized data, we next make an analogy with the operation of some machine learning algorithms. Consider the k-nearest neighbors

algorithm (kNN), a simple classifier, and assume a data set with n training tuples, each one assigned to a binary class label. kNN classifies a new tuple according to a majority vote of its k closest "neighboring training tuples" in the feature space. Note that, in this context, k has nothing to do with anonymity. A small k implies considering few neighboring samples for classification, which would be the most representative ones, being the closest, but would not be so reliable in terms of predictability. On the other hand, a large k implies taking more (and not so close) neighboring samples, being demographically less representative, but predictably more reliable. This trade-off is illustrated in Fig. 3.3, where we measure the accuracy of kNN on the original UCI Adult data set for several values of k. As depicted in Fig. 3.3, the classification accuracy of kNN improves as groups rather than individual samples are considered to robustly infer what would effectively constitute a macrotrend.

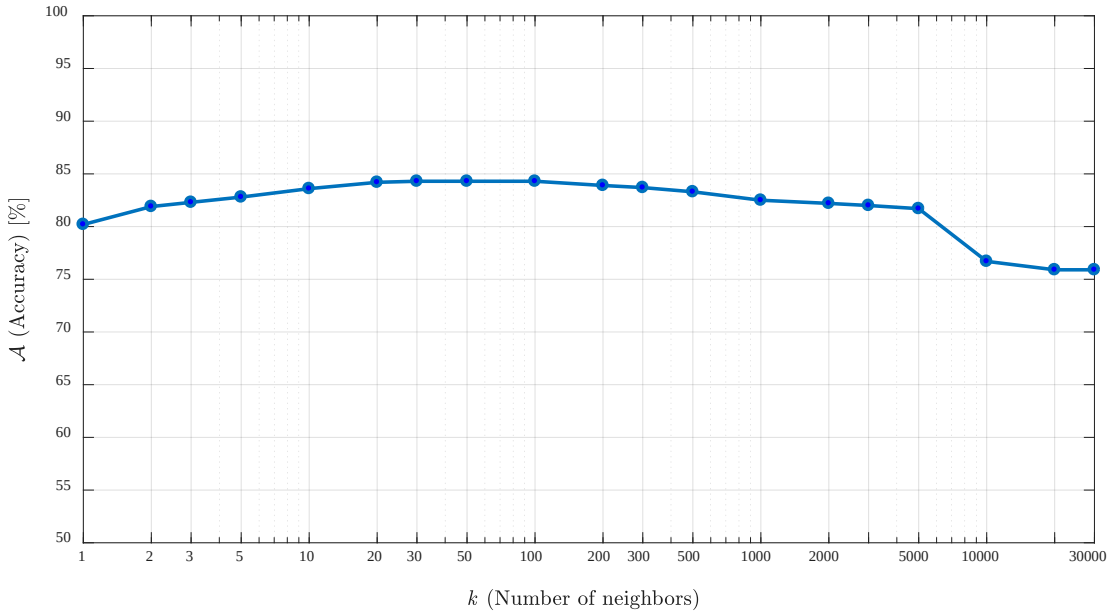


Figure 3.3: Accuracy of the kNN machine learning algorithm applied on the UCI Adult data set, for different values of k (here, k is not related with k-anonymity).

We argue that microaggregation would be acting analogously to kNN when aggregating neighboring data points to construct cells, and computing averages to get representative centroids for each cluster. Such clustering could be regarded as a denoising process. In fact, the benefit of preprocessing data with unsupervised techniques based

on clustering, prior to supervised learning, is known in the machine-learning literature. Therefore, it seems reasonable to expect k-anonymous microaggregation to have a minor (and sometimes even positive) impact on the empirical utility of data, measured as the accuracy of machine learning models when deriving macrotrends.

3.3.2 Measuring the impact of microaggregation on a synthetic data set

We begin our experiments by analyzing the effect of microaggregation on synthetic data. To this end, we generate 30,000 samples of 3-dimensional Gaussian data. The first two dimensions are assumed to be quasi-identifiers, and the third dimension represents a binary confidential attribute. Since we require that the quasi-identifiers be predictors of the confidential attribute (as would be, e.g., the weight and height predictors of the existence or not of a disease in an individual), we introduce in the data a learnable macrotrend or dependence among the quasi-identifiers and the confidential attribute.

Next, we describe how we generate this synthetic data set. Let X be a bidimensional continuous r.v. representing the two quasi-identifiers $(x_1,\ x_2)$, and let Y be a binary r.v. indicating whether an individual has a disease ($Y = 1$) or not ($Y = 0$). The data set is generated in two parts, each matched to a different value of Y. Accordingly, X is distributed as a unit-variance Gaussian distribution with mean μ for $Y = 1$, and with mean $-\mu$, for $Y = 0$. In Fig. 3.4, we represent this data set by plotting the values of X for each record as coordinates of a point in a plane, coloring each point according to the class to which it belongs. As expected, two clouds of points are obtained (the red one, for $Y = 1$, slightly on the right; and the blue one, for $Y = 0$, on the left) where we can guess the optimal threshold to estimate the class $\hat{Y}$ of each point.

Let $\mathrm{P}\{Y = 1|x\}$ be the discriminative model of this problem. The prevalence p of a disease in this data set is the proportion of records matched to the class $Y = 1$. It is routine to represent this model, using logarithmic odds, as

$$\mathrm{L}\{Y = 1 \mid X = x\} = 2\mu x + \ln\frac{p}{1-p}.$$

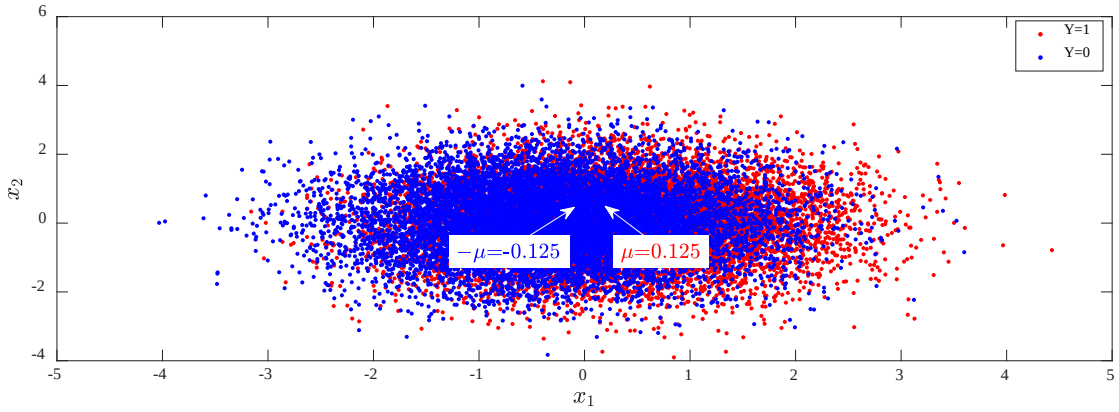


Figure 3.4: Depiction of the quasi-identifiers (x_2 *vs* x_1) of our synthetic data set. Samples are colored according to their class, y; blue for $y = 0$ and red for $y = 1$.

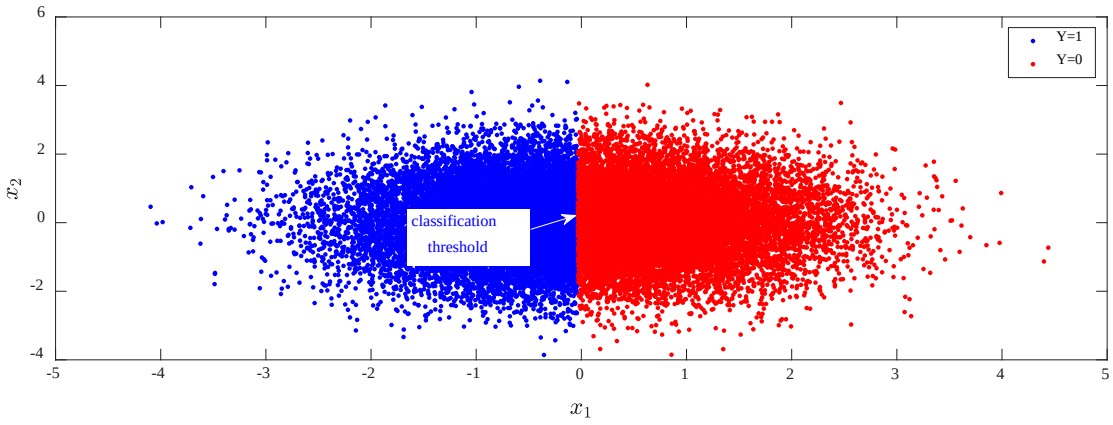


Figure 3.5: Depiction of the quasi-identifiers (x_2 *vs* x_1) of our synthetic data set. Samples are colored according to their class, y; blue for $y = 0$ and red for $y = 1$.

We denote the cumulative distribution function (CDF) of the zero-mean, unit-variance Gaussian distribution as Φ. The accuracy $\mathcal{A}$ of our model to find the estimated class $\hat{Y}$ can be expressed as

$$\mathcal{A} = \mathrm{P}\{Y = \hat{Y}\} = (1-p)\Phi(\theta+\mu) + p\,\Phi(\mu-\theta),$$

for a given threshold $x = \theta$. It is straightforward to derive the optimal threshold θ^* for maximum accuracy of our discriminative model, which is

$$\theta^* = -\frac{1}{2\mu}\ln\frac{p}{1-p}.$$

In order to have a balanced data set, we use $p = 0.5$, thus half of the samples are matched to each class. Consequently, the optimal threshold to classify both parts of the data set is $\theta^* = 0$. Additionally, we choose $\mu = 0.125$ so that the distribution of both groups of samples are close; evidently, the more overlapped the two groups are, the more difficult the classification task.

Next, we train a machine learning model over a stratified part of the synthetic data, using the C4.5 algorithm. Since μ is low, the accuracy obtained from the classifier is 60%. Based on this model, we predict the class using the quasi-identifiers. Then, in Fig. 3.5, we plot the same clouds of samples of Fig. 3.4, but now we color them according to the *predicted* class. Accordingly, the classification threshold is evident.

To analyze the impact of microaggregation, we apply MDAV to the training set of this data set with $k = 3000$, which is a very large value of cluster size. Accordingly, we get 7 cells that we plot in Fig. 3.6 with distinct colors; the classification threshold is also plotted. Notice in the figure that, after the clustering applied by MDAV, the samples of 3 out of 7 cells might be misclassified with a higher probability since such samples are distributed on both sides of the classification threshold. However, the remaining 4 cells, which account for about 57% of the data, are clearly defined on one side of the classification threshold, so they would be correctly classified. Hence, even after microaggregation, machine-learned macrotrends might not suffer a significant impact, i.e., the accuracy obtained from original data is not harshly reduced, even for high values of k.

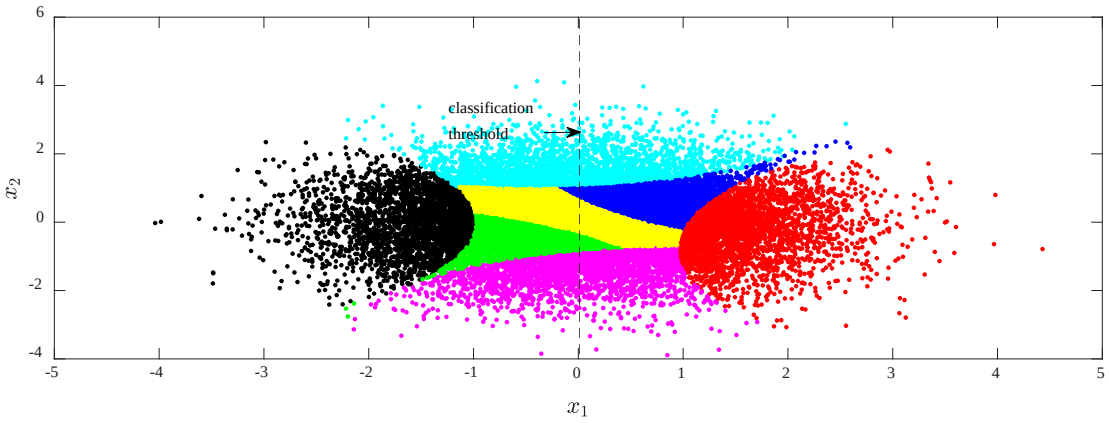


Figure 3.6: Cells of samples obtained after k-anonymous microaggregation with MDAV on the quasi-identifiers of our synthetic data set ($k = 3000$).

To illustrate more systematically this effect on data utility, we plot in Fig. 3.7 the accuracy and F-measure of the learning model obtained from our synthetic data, after anonymizing it with different values of k. Consistently with the previous experiment, none of these utility metrics is drastically affected by the influence of microaggregation, for practical values of k. Another metric of the impact of microaggregation (not necessarily in terms of utility degradation) is also depicted in Fig. 3.8. Here, we observe that distortion, measured in terms of MSE, increases with k. However, distortion starts soon to increase significantly from $k = 100$. This divergence between accuracy and distortion is evidenced in Fig. 3.9, where the connection between both seems nonspecific and nonlinear. A more detailed discussion regarding these results is presented in the next section, where real data is considered.

3.3.3 Results from real data sets

We begin our first series of experiments by computing the relevance of the number of predictive attributes in each data set. The aim is to analyze how the accuracy of the classification task varies with the number of predictive attributes. To determine the order of the attributes employed, we used sequential forward selection, which consists in sequentially adding attributes to an empty set until the addition of further attributes does not decrease the accuracy of the classification task. Figure 3.10

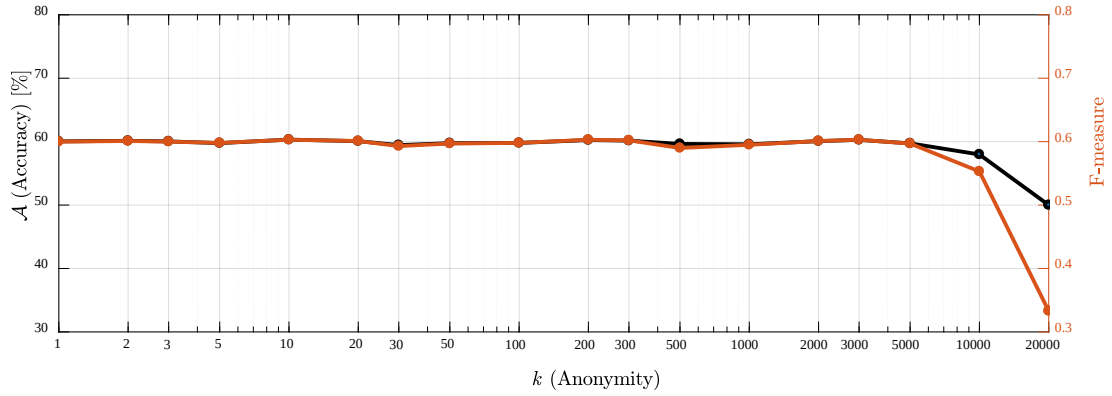


Figure 3.7: Degradation of the empirical utility (accuracy and F-measure) of our synthetic data set when microaggregated (using MDAV) for a wide range of k.

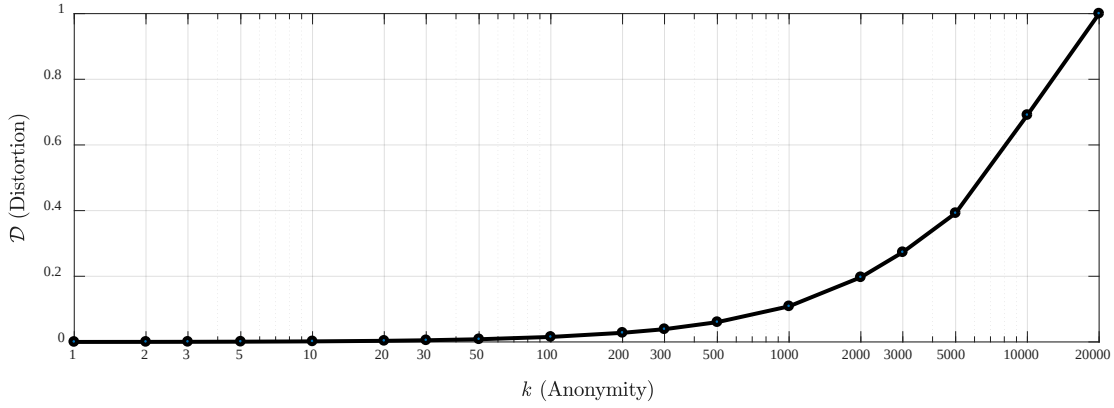


Figure 3.8: Distortion, measured as MSE, introduced by MDAV k-anonymous microaggregation to our synthetic data set, considering a wide range of k.

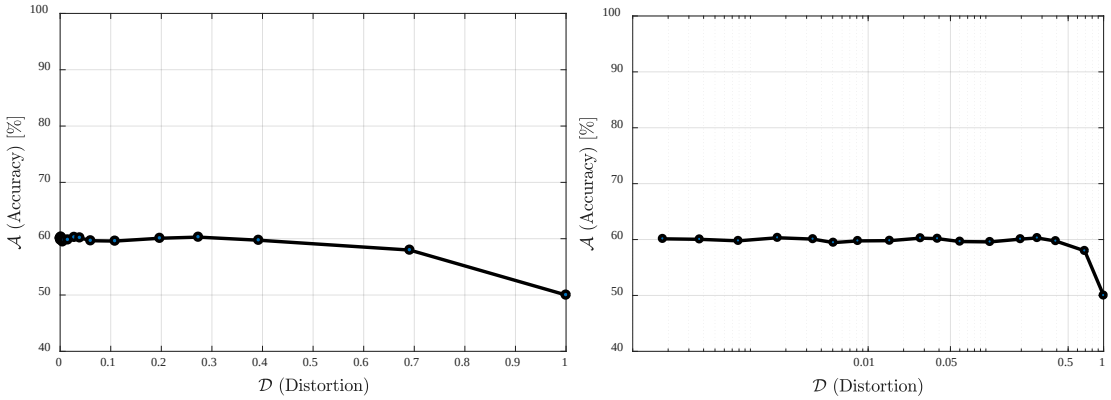


Figure 3.9: Accuracy of the *bagging* machine learning model trained on our microaggregated synthetic data set, against the distortion due to MDAV.

illustrates the variation of accuracy with the number of predictive attributes for UCI Adult.

Although intuition could suggest that even small levels of data perturbation might yield important reductions in utility, riveting results were found in our experiments when using microaggregation. First, Fig. 3.11 shows how the accuracy and F-measure of the classifier degrades as the privacy parameter k increases, when anonymizing the UCI Adult data set. As expected, accuracy attains its highest value (about 85%) when no anonymization is applied ($k = 1$). For $k = 200$, which is a relatively large value of cluster size, accuracy only decreases up to 82%. We also note that, even for a value of k of $3,000$, which implies a strong level of anonymity, accuracy is approximately 80%.

Figure 3.11 also depicts a dotted line to represent the lowest accuracy achieved by the machine learning algorithm (76.37%) when no predictor attributes are used (suppression of all quasi-identifiers); this provides the highest level of privacy protection. Note that, when all quasi-identifiers are suppressed, the machine learning model always classifies a new instance depending on the majority value of the class attribute.

From the figure, we observe that a reduction in accuracy from 85% to 82% (attained for $k = 200$) when the key attribute (important predictor) "Capital Gain"

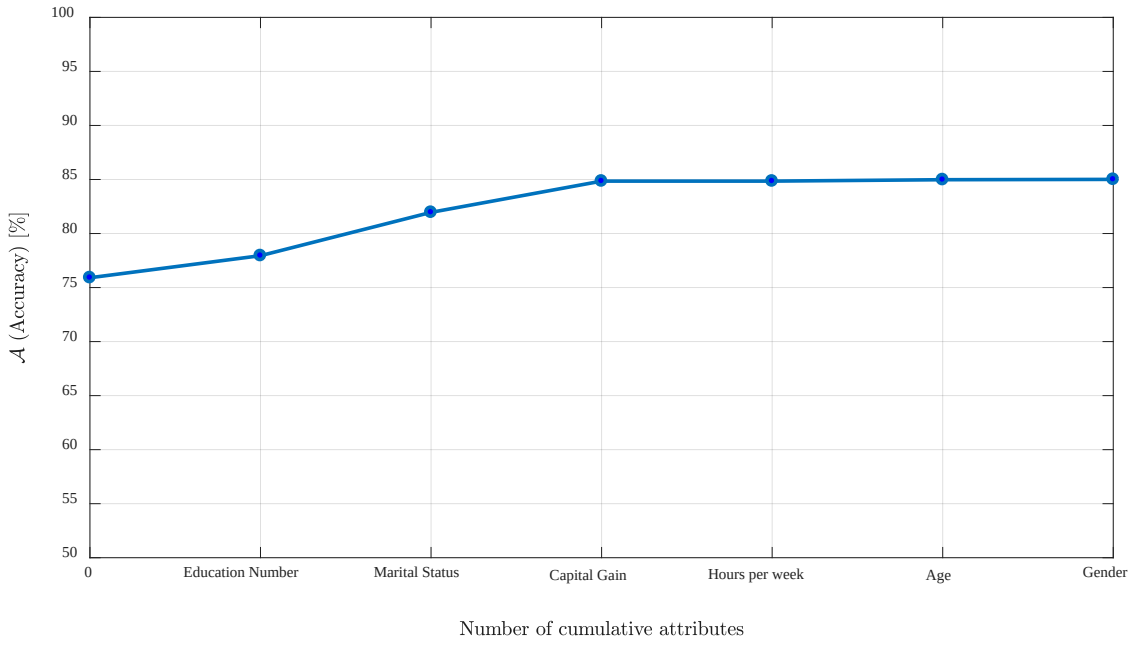


Figure 3.10: Relevance of the cumulative number of selected attributes from the UCI Adult data set as predictors of the class attribute (Annual Salary).

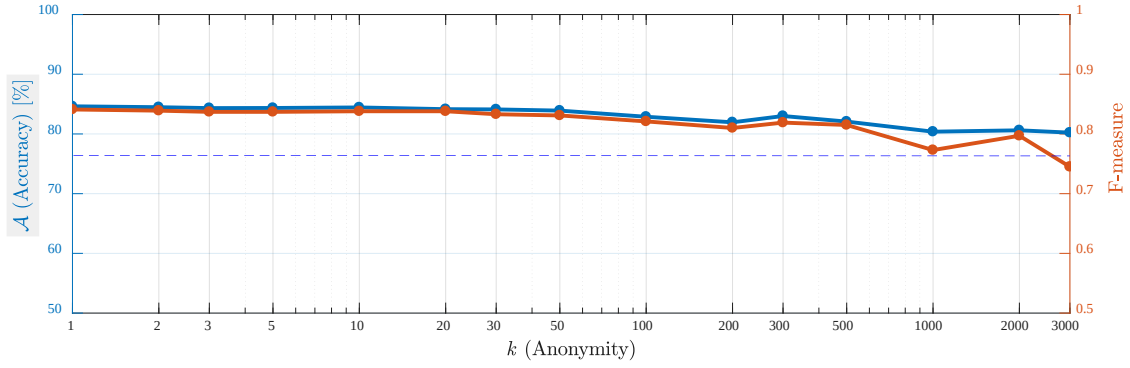


Figure 3.11: Degradation of the empirical utility (accuracy and F-measure) of the UCI Adult data set when microaggregated (using MDAV) for a wide range of k.

is eliminated. Similarly, even when $k = 3,000$, we obtain a smaller impact on utility (accuracy of 80%) than when all predictors –except "Education Number"– are suppressed. This are good news for microaggregation, since it suggests that we can still get useful microdata after applying more than reasonable levels of privacy. The reported values of accuracy and other metrics (F-measure and AuC) are shown, in more detail, in Table 3.3.

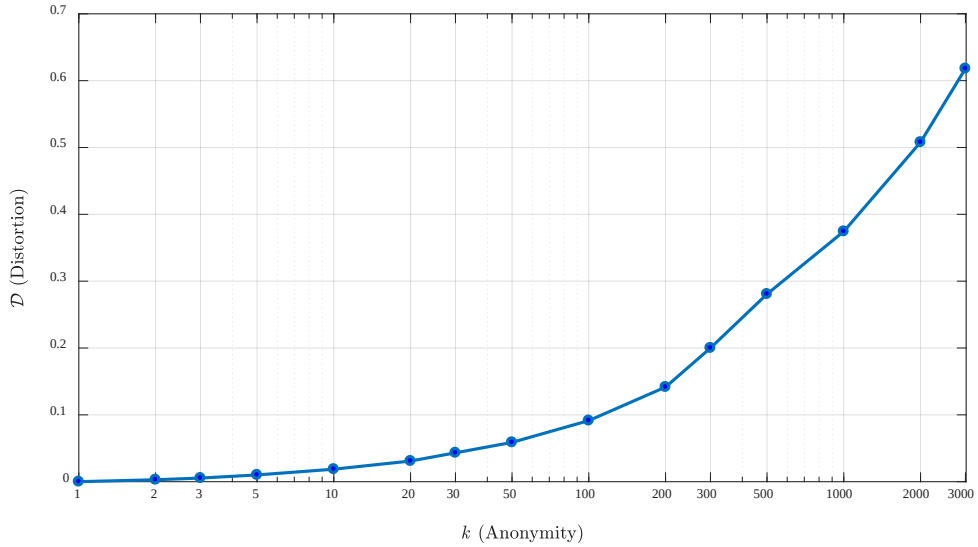


Figure 3.12: Distortion introduced by MDAV k-anonymous microaggregation to the UCI Adult data set, when microaggregated for a wide range of k.

The impact of MDAV on the UCI Adult data set was also measured in terms of the distortion introduced to quasi-identifiers. We used MSE to quantify such distortion. In Fig. 3.12, we can see how distortion increases from 0 (when $k = 1$) to 0.62 (for $k = 3,000$). Specifically, we observed a pronounced growth from $k = 100$, although for values of k smaller than 100, distortion did not seem significant.

In Fig. 3.13, we plot accuracy vs distortion. The most relevant conclusion that can be drawn from this figure is that accuracy stays relatively stable (greater than 80%) up to distortions of 0.7. Precisely, although MSE is conventionally used in SDC to compare the utility of microaggregation algorithms, we observe that this distortion metric says little about the impact on the performance of a machine-learning classifier.

Table 3.3: Different utility metrics for the UCI Adult data set when microaggregated for a wide range of k.

k	Accuracy	F-Measure	AuC
1	84.63	0.841	0.902
2	84.48	0.839	0.898
3	84.33	0.837	0.897
5	84.35	0.837	0.897
10	84.44	0.838	0.898
20	84.15	0.838	0.891
30	84.11	0.833	0.887
50	83.91	0.831	0.883
100	82.88	0.821	0.875
200	81.95	0.810	0.861
300	83.00	0.819	0.848
500	82.07	0.815	0.827
1000	80.38	0.773	0.794
2000	80.61	0.797	0.693
3000	80.22	0.745	0.585

In other words, the data yielded by this figure seems to provide convincing evidence that MSE is not a suitable measure of utility for classification tasks.

In our evaluation of the UCI Pima Indian Diabetes data set in Fig. 3.14, we noted that the degradation margin of utility goes from 74.2% (when $k = 1$, thus without perturbation) to 65.23% (from $k = 100$). Microaggregation showed a similar behavior to that observed in the UCI Adult data set but, being 50 times smaller, it evidently degrades more quickly as k increases. However, a noticeable stability is appreciated in accuracy up to k =30 and, in fact, this performance metric remains close to the upper baseline at around 74%. For values of k between 10 and 30, accuracy was even improved, which could be explained by the denoising effect of averaging through clever clustering, that may positively contribute to a more robust inference. This effect was described in Sec. 3.3.2. Interestingly, Fig. 3.15 showed a sustained increase in distortion as k becomes larger. To gain insight into this relative stability in accuracy, we also plotted accuracy vs distortion in Fig. 3.16 and confirmed that, up to distortions of 50%, utility remains close to the upper baseline. The values of accuracy and other metrics (F-measure and AuC) obtained for this data set are also shown in Table 3.4.

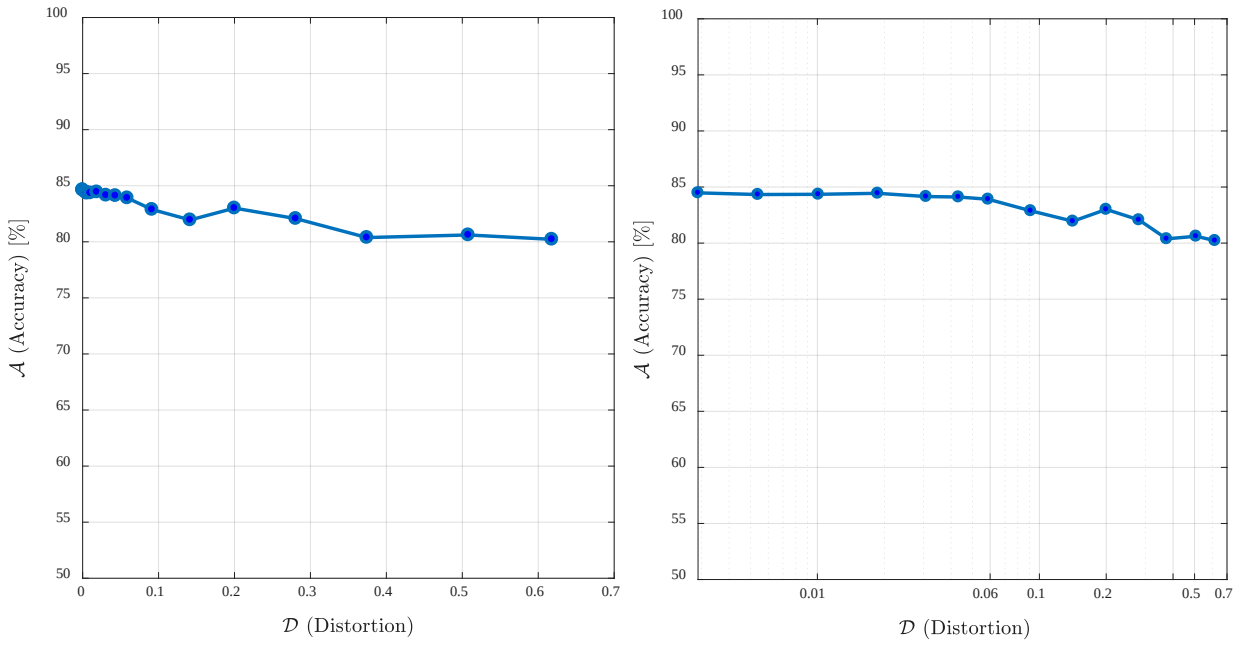


Figure 3.13: Accuracy of the *bagging* machine learning model trained on the microaggregated UCI Adult data set, against the distortion due to MDAV.

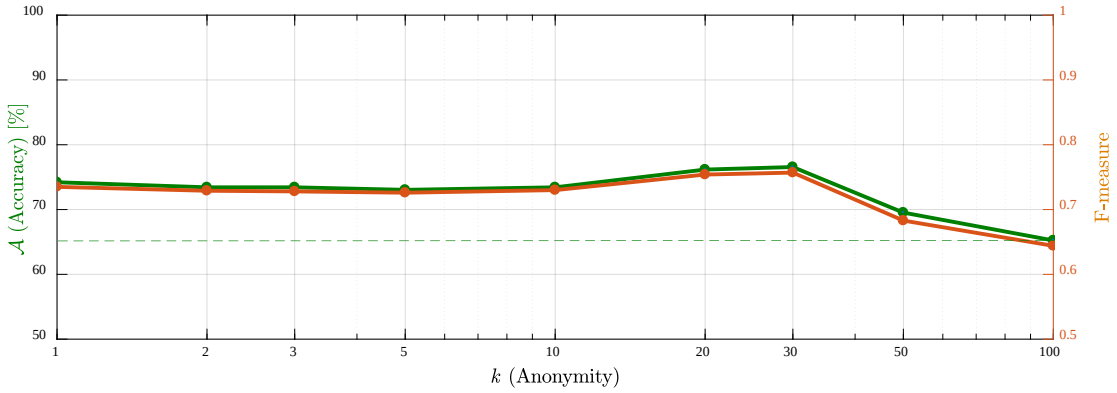


Figure 3.14: Degradation of the empirical utility of the UCI Pima Indians Diabetes data set when microaggregated (using MDAV) for a wide range of k.

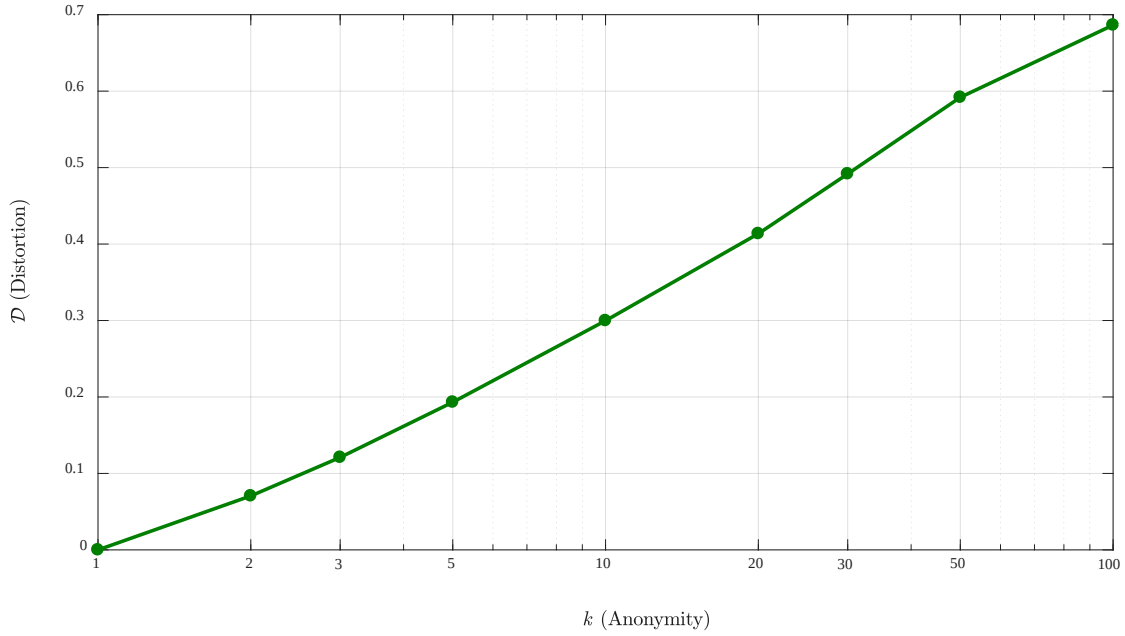


Figure 3.15: Distortion introduced by MDAV k-anonymous microaggregation to the UCI Pima Indians data set, for a wide range of k.

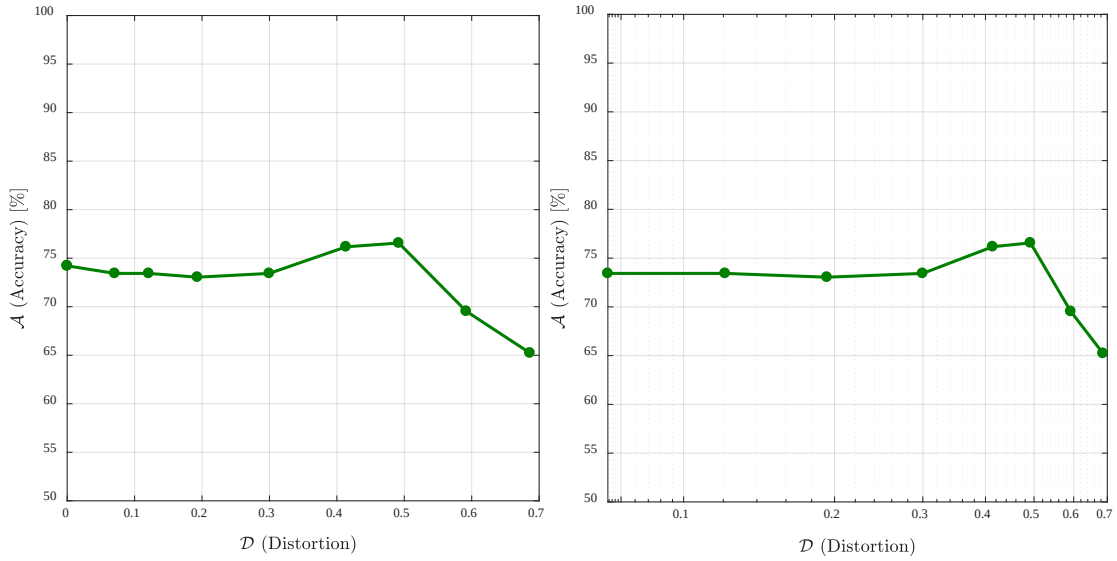


Figure 3.16: Accuracy of the *logistic regression* model trained on the microaggregated UCI Pima Indians Diabetes data set, against the distortion due to MDAV.

Table 3.4: Different utility metrics for the UCI Pima Indians data set when microaggregated for a wide range of k

k	Accuracy	F-Measure	AuC
1	74.21	0.735	0.813
2	73.43	0.729	0.810
3	73.43	0.728	0.808
5	73.04	0.726	0.804
10	73.43	0.730	0.806
20	76.17	0.754	0.807
30	76.56	0.757	0.789
50	69.53	0.683	0.758
100	65.23	0.644	0.716

Finally, we examine the Irish data set in Fig. 3.17. Here, we observe a wide degradation margin since its label attribute has balanced classes. Specifically, accuracy goes from 72.62% to about 68.04% when the privacy parameter k equals 3,000. Also, we can see, once again, that accuracy remains quite high (more than 70%) and stable up to $k = 2,000$. A similar behavior is observed for F-measure. Although the size of the data set at hand is relatively large (100K instances), the available evidence suggests that the reduction of empirical utility of the data due to microaggregation is not significant for a wide range of values of k. Such effect is also noticeable in Fig. 3.18, where we plot accuracy vs distortion. Table 3.5 shows the reported values of accuracy, as well as other metrics (F-measure and AuC), in greater detail.

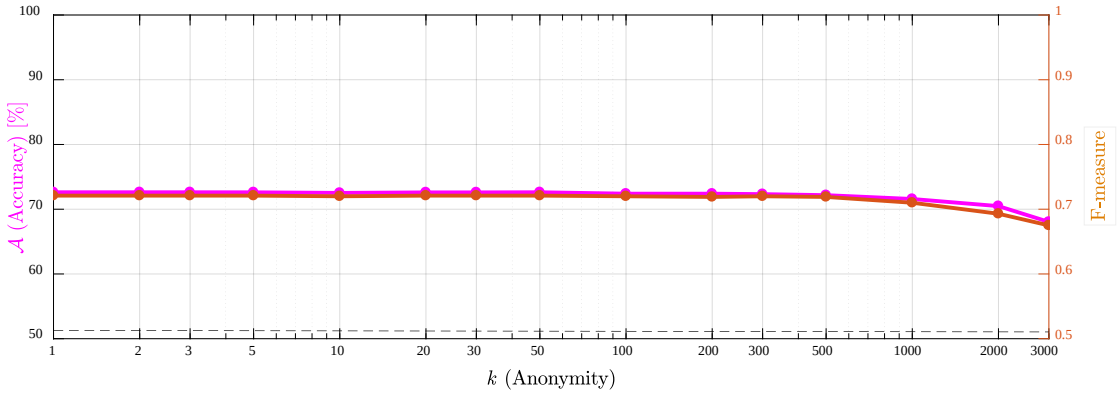


Figure 3.17: Degradation of the empirical utility (accuracy) of the Irish Census data set when microaggregated for a wide range of k.

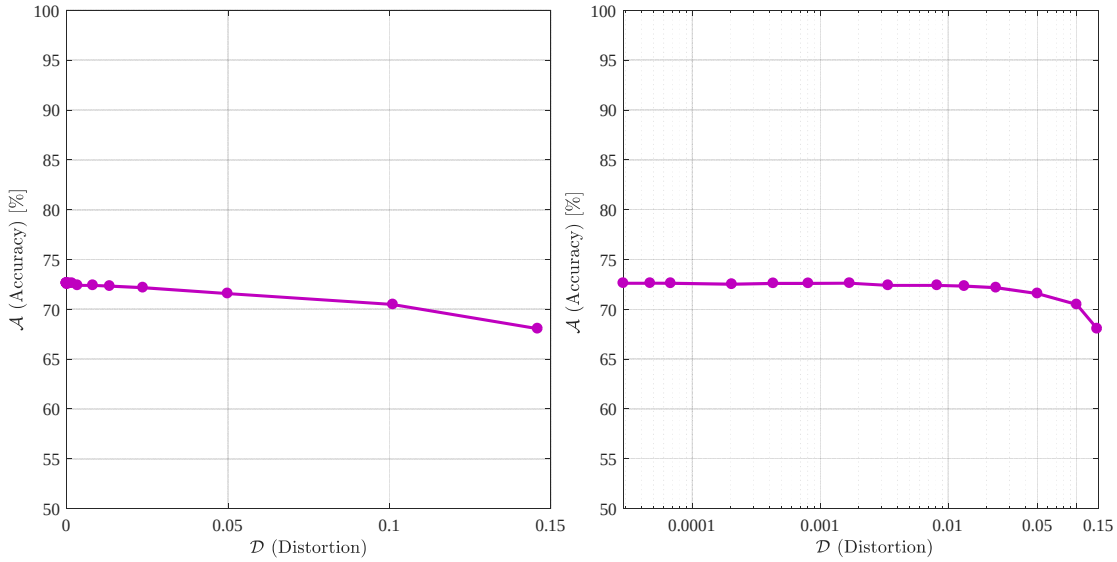


Figure 3.18: Accuracy of the *C4.5* machine learning model trained over the microaggregated Irish Census data set, against the distortion due to MDAV.

Our experimental findings confirm that MDAV introduces sufficiently small levels of perturbation in the quasi-identifiers, so that the statistical properties of the published data can be preserved to a large extent, while satisfying a given k-anonymity constraint. The upshot is that much of the empirical utility is retained within the microaggregated data. In fact, the results of our experiments suggest that such impact is often minor, since microaggregation preserves machine-learned macrotrends. We believe that the average operations performed by MDAV to find a centroid representative of k tuples are working as a noising removal filter that prevents the classifier algorithm from adjusting to the existing noise in the data.

Interestingly, although not explicitly reported in these terms, previous work surveyed in Section 2.3.2 appears to be consistent with our findings. For example, in [55], where different algorithms based on generalization and suppression are compared, the degradation in accuracy is certainly small in many cases. Other works in the literature give some clues about a potential "constructive effect" of anonymization mechanisms. In that sense, [53] mentions that anonymization might sometimes behave as a form of feature selection or construction. Moreover, in [58], the authors conclude that a selective anonymization may not be so destructive. Finally, although using a less

Table 3.5: Different utility metrics for the Irish Census data set when microaggregated for a wide range of k

k	Accuracy	F-Measure	AuC
1	72.62	0.721	0.736
2	72.62	0.721	0.733
3	72.62	0.721	0.733
5	72.61	0.721	0.733
10	72.52	0.720	0.733
20	72.60	0.721	0.733
30	72.60	0.721	0.733
50	72.62	0.721	0.734
100	72.40	0.720	0.731
200	72.40	0.719	0.735
300	72.33	0.720	0.729
500	72.17	0.719	0.718
1000	71.58	0.710	0.729
2000	70.48	0.693	0.739
3000	68.04	0.675	0.703

conclusive argument, [56] states that, while making no changes to existing tools and systems, significant utility can be extracted from anonymized data.

Testing a wide range of values of the privacy parameter helps to make visible the overall effect of anonymization on data utility. Doing so also assists in noticing the influence of other critical criteria such as the size of the data set and the absolute upper and lower bounds of utility. As shown in our experimental results, the utility of anonymized microdata, measured as classification accuracy, may not take values strictly from 0 to 100%. The intrinsic statistical properties of released data would already limit the capabilities of machine learning algorithms and, thus, the improvements they get over baseline methods (e.g., always predicting the most frequent class in the training set). Evidently, very little utility can be maintained after anonymization if machine learning (classification) algorithms perform poorly, by default, with respect to the baseline. Unfortunately, these considerations are not always made when evaluating the performance of k-anonymous microaggregation or, in general, of anonymization mechanisms.

3.4 Conclusion

The experiments presented in this chapter have indicated, with some consistency, that the impact of the de facto microaggregation standard (Maximum Distance to Average Vector, MDAV) on the performance of machine learning algorithms is often minor to negligible for a wide range of k, for a variety of classification algorithms and data sets. Furthermore, experimental evidences have suggested that the traditional measure of distortion in the community of microdata anonymization may be inappropriate for evaluating the utility of microaggregated data.

With the advent of the Internet and the development of sophisticated data analytics, the availability of massive amounts of information has increased the demand for data sharing. In the context of structured data, microdata are an invaluable source of information for their potential to reveal patterns or macrotrends about the population there represented.

Before these data can be made public or shared with external entities, data holders must ensure individual privacy is safeguarded. Perturbing quasi-identifiers attributes is the usual approach to prevent identity disclosure in microdata. Nonetheless, while perturbation may prevent reidentification attacks, it may have a large impact on data utility, particularly on the performance of machine-learning tasks. To cope with it, several works have proposed adapting data-anonymization or machine-learning algorithms to get more utility from anonymized data. We claim in this work, however, that the default operation of some anonymization mechanisms may not affect data utility significantly.

We have investigated in this chapter the high-utility SDC spectrum, implemented by syntactic k-anonymous microaggregation, which has a direct application on the health domain where utility is critical. Our experiments have shown, with some consistency, that k-anonymous microaggregation implemented through MDAV does not have a significant impact on machine-learned macrotrends for multiple data sets and a wide range of machine-learning algorithms. Trying to consider the domain of data in our evaluation, we not only tested different data sets but also multiple learning

algorithms to extract the maximum utility from the data. Then, these algorithms were selected to get the highest accuracy from each data set.

These excellent results on learning performance from microaggregated data deserve careful attention. As the lack of substantial degradation in classification accuracy for a generous range of microcell sizes k may be somewhat counterintuitive, we conducted further verification on such remarkable finding. Specifically, we applied the k-nearest neighbor algorithm (kNN) to the original, unperturbed data, in order to verify our working hypothesis that clustering effectively acts as a form of averaging and thus denoising. In our verification, k is the usual name for the parameter governing the size of the cluster of the kNN algorithm, analogous to some extent to the anonymity parameter.

We contend that a similar denoising effect, akin to averaging through clustering, is the underlying cause of the striking utility of k-anonymous microaggregation. Conceivably, for reasonable values of the anonymity parameter k, microaggregation should not substantially devalue the process of inference of macrotrends carried out by the machine learning algorithm. Moreover, high-utility microaggregation algorithms such as MDAV may, in some cases, positively contribute to a more robust inference by denoising through clever clustering of demographically similar individuals. The benefit of preprocessing data with unsupervised techniques based on clustering, prior to supervised learning, is known in the machine-learning literature. The lack of substantial degradation in classification performance due to k-anonymous microaggregation, and the occasional slight improvement in utility, is a novel result of strategic importance in the privacy arena.

Finally, these results provide confirmatory evidence that, while keeping a monotonicity relationship with accuracy, the traditional utility metric of SDC (i.e., MSE) is not an ideal metric to determine the impact on the utility of microaggregated data, since there exists a non-specific non-linear dependence.

Chapter 4

Comparison of the impact of different microaggregation algorithms on the empirical utility of data

4.1 Introduction

As we discussed in Sec. 2.1, currently, in the big data era, there are several incentives to exploit data. In general, there is more data available, and better and cheaper technology to take advantage of it, including, e.g., a lot of algorithms for machine learning analytics. The potential benefits of these technologies are countless in several fields such as healthcare, advertising, and even industrial engineering ([76–78]). Said benefits entail important economic profits, so giant tech companies are leveraging data as core assets ([79]) that are disclosed (exploited, shared or even sold) to maximize profit.

Unfortunately, since personal information is inevitably involved in this data, such incentives and tools to exploit data may easily imply abusing user privacy. Even if direct identifier attributes such as full names are suppressed, the combination of several non-direct identifier attributes (also known as quasi-identifiers) may still be

used to re-identify an individual. If a sensitive attribute (e.g., gender, health status, income) were disclosed, re-identification would enable an attacker to associate an individual with such attribute, violating her privacy.

To mitigate such risk, the data needs to undergo first a process of anonymization, which typically implies modifying the data. In this regard, *statistical disclosure control* offers an interesting approach to protect individual privacy while preserving some of the data utility.

Since the criteria posed by privacy models are invariably met by perturbing quasi-identifiers to anonymize data, there is an impact on the data in terms of loss in utility ([24]). However, said utility loss may vary according to the strategy followed by the privacy mechanism, even when the privacy parameter is already met. If the resulting data utility does not meet the requirements of the application domain (e.g., health) a different privacy parameter or mechanism should be used. Some of these mechanisms include microaggregation, suppression, generalization and noise addition.

Being a high-utility approach that may be applicable for critical domains such as health, our work is devoted to k-anonymous microaggregation. In Sec. 2.2 we have described some of its foundations, and, in chapter 3, the de facto standard microaggregation algorithm (MDAV) was evaluated in terms of data utility degradation.

Although MDAV demonstrated to be a utility-preserving anonymization algorithm to some extent, there are other algorithms following a similar microaggregation spirit. Once agreeing, as discussed in Sec. 2.3 , on the relevance of metrics capturing the practical utility of anonymized data, it would be interesting to assess the impact of these privacy mechanisms on such utility. This would help unveil the strategies that best preserve utility, but also whether or not standard metrics faithfully predict such practical utility.

k-Anonymous microaggregation is typically implemented through different mechanisms. In this chapter, we evaluate the most relevant of such mechanisms [2, 26, 42] in terms of the practical utility of the anonymized data.

Our evaluation aims to provide insight into those implementations by assessing them in terms of the loss in classification accuracy of the machine-learned models built from modified data. We employ non standard, but empirical utility metrics

taken from machine learning, which is currently a very common application data domain.

In our evaluation of such mechanisms, we aimed to identify the anonymization parameters of each of them that may help preserve the macrotrends of the data. Our extensive experiments found out that the efforts to preserve the statistical dependence within quasi-identifiers and confidential attributes (such as in MDAV with statistical dependence) may effectively attenuate the impact of microaggregation on the utility of data.

Last but not least, for all the examined microaggregation algorithms, we also investigated the capability of a standard distortion metric to predict the empirical utility of anonymized data.

Chapter outline

The rest of this chapter is organized as follows. Section 4.2 reviews the k-anonymous microaggregation algorithms evaluated here. Next, Sec. 4.3 describes the methodology followed to evaluate such impact. Section 4.4 shows the experimental results obtained for a variety of microaggregation algorithms, data sets and machine-learning algorithms. Lastly, a brief discussion is presented in Sec. 4.5 and conclusions are drawn in Sec. 4.6.

4.2 Background on k-anonymous microaggregation algorithms

In this section we briefly describe some well-known microaggregation algorithms with the aim of introducing the strategies followed to group and reconstruct microcells. This will provide with some feedback for the evaluation performed in this chapter that focuses on unveiling the utility preserving capabilities of k-anonymous microaggregation, but particularly on showing that some efforts to preserve the statistical dependence within data would help to increase said empirical utility.

In 2.2.4, we already referred to *MDAV* as the de facto standard for microaggregation of numerical microdata [29]. By systematically finding the furthest k-anonymous cells within the data set, MDAV replaces each record with the centroid (average) of its corresponding cell. It evolved from the multivariate fixed-size microaggregation method and was proposed by [26]. MDAV provides an excellent heuristic method for multivariate microaggregation [42] in terms of utility, measured both syntactically [42] and empirically [16], and in terms of computation complexity. Note that MDAV generates cells of fixed size k and potentially a cell with size $2k - 1$.

V-MDAV ([42]), follows a similar strategy to MDAV but enables the aggregation process to generate variable-size cells. When k records are already aggregated, an extension step may include more records to the cell being formed (up to a total $2k - 1$) if they are "close enough" to this cell. The inclusion decision is defined by a gain parameter γ that must be adjusted depending on the data set. It offers less distortion for some data sets at a computational cost comparable to that of MDAV.

Unlike traditional k-anonymous microaggregation (e.g., through MDAV) where only the values of quasi-identifiers X are considered when building microcells, **microaggregation with preservation of statistical dependence** (we call it *MDAV with SD*) also includes confidential attributes ([2]) in the partition design. Thus, if a confidential attribute Y has to be predicted, this approach would lead to a more accurate prediction (e.g., classification) from perturbed quasi-identifiers $\hat{X}$. To involve both types of attributes, the authors propose designing a cell assignment function that minimizes a multiobjective Lagrangian distortion function

$$\mathcal{D} = (1 - \lambda)\mathcal{D}_X + \lambda\mathcal{D}_Y$$

where $\mathcal{D}_X$ is the traditional information loss term based on MSE, $\mathcal{D}_Y$ characterizes the degradation in statistical dependence, captured through the non linear predictability of Y from X, and λ controls the tradeoff between these two optimization objectives.

Finally, **Mondrian**, presented in [80], is a greedy algorithm that recursively partitions a microdata set in regions of at least k records, where a dimension (attribute) and a value about which to partition have to be heuristically chosen in each iteration. This is a microaggregation algorithm in the sense that it partitions a microdata set

in variable-size cells, satisfying the k-anonymity criteria. The values of the quasi-identifiers for each cell are reconstructed as non-overlapping intervals in which such values are contained. Intuitively, such partitions are defined as hyperrectangles in the multidimensional space of quasi-identifiers.

4.3 Methodology of Evaluation

4.3.1 Evaluation context

Our evaluation scenario is similar to that presented in Sec. 2.1.1, i.e., involves a microdata set whose quasi-identifiers are correlated with its corresponding confidential attribute. Moreover, this information may have to be publicly released for research purposes, so k-anonymous microaggregation is applied over quasi-identifiers to protect the privacy of data subjects. This is the standard attack model of the SDC literature ([68]).

Accordingly, anonymized quasi-identifiers (here also input samples) would be published along with untouched confidential attributes (also output labels) to feed a machine learning classifier, which is the enabler of the selected application domain of data. The resulting models would allow external data analysts to build predictive models on different testing data. Intuitively, the quality of the statistical trends embedded in the resulting anonymized data would be undermined with respect to those in the original data.

Although in chapter 3 we confirmed that MDAV offers interesting benefits in terms of distortion and classification accuracy, additional variations exist, some even incorporating utility improvements ([2]), which have not been assessed in this context.

Finally, the privacy metric we use is naturally k-anonymity since microaggregation algorithms aim at guaranteeing such criteria. In addition, we also assume binary classification as the application domain of data, so the utility metric is the accuracy of the classification model built from anonymized data, as performed in [53, 55, 58]. Basically, accuracy quantifies the rate of correctly classified samples in a test set.

Besides, we also use a complementary machine learning metric, F-measure, to confirm our results in the next sections.

Thus, higher values of k, implying larger anonymous microcells, will offer more privacy but, at some point, less utility.

4.3.2 Scenario setup

As can be grasped from the sections above, our experimental setup builds on the algorithms for privacy protection and utility exploitation, the data sets used to assess the impact of anonymization, and the steps taken to get the results.

Being MDAV the de facto microaggregation algorithm, we extend the study presented in chapter 3 by assessing not only MDAV but also V-MDAV ([42] and MDAV with SD [2]). As explained in Sect. 4.2, both of them aim at increasing the data utility preserved, measured from the distortion applied by these two variants of MDAV. While V-MDAV proposes building larger microcells, when possible, to favor forming more compacted clusters, MDAV with SD builds microcells capturing the statistical dependence between quasi-identifiers and confidential attributes. Moreover, Mondrian ([80]) is also considered in our setup to corroborate the performance of microaggregation algorithms, no matter the strategy used to build k-anonymous microcells. Some of the implementation details of these algorithms and further references are included in Sec. 4.3.1.

To measure the utility of microaggregated data, we use the machine learning algorithms that obtain the best performance, in terms of classification accuracy, from each of our data sets. Since the intrinsic nature of the data sets might vary, we experimentally determine the best performer by testing a series of algorithms such as boosted trees, logistic regression, Support Vector Machine, and k-nearest neighbor on the original data. This way we more rigorously adapt our evaluation to the specific utility context.

The data where microaggregation algorithms were assessed includes both real and synthetic data sets. As in chapter 3, we look for data sets meeting two main requirements: include demographic attributes and evidence a correlation between the

quasi-identifiers and a confidential attribute. We briefly describe their characteristics in Table 4.1. The first is the "Adult" data set ([71]), which is a standard when assessing microaggregation algorithms. Although this data set has 15 attributes, for our tests we use the six that contribute the most to the application domain; the contribution of the rest for binary classification is null. We also tested the "Breast Cancer Wisconsin" data set ([81]) and the "Heart disease" data set ([82]) that contain medical data extensively used to evaluate binary classification tasks. Finally, we created an elementary synthetic data set with three attributes mimicking two quasi-identifiers and a binary confidential attribute; to do it, two groups of two-dimensional quasi-identifiers are generated following two different, but overlapping, normal distributions.

Table 4.1: Description of the Data sets Used to Evaluate the Impact of k-Anonymous microaggregation.

Data set	**# of records**	**# of attributes used as quasi-identifiers**	**list of quasi-identifiers used (input)**	**confidential attribute (output label of the data set in ML terms)**
Adult [71]	45,222	15	Age, education-num, marital-status, sex, capital-gain, hours-per-week	Salary (>50K?)
Breast Cancer Wisconsin [81]	699	9	clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses	class (benign/malignant)
Heart Disease [82]	303	13	age, sex, chest pain type, trestbps, serum cholestoral, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, ST depression induced by exercise, the slope of the peak exercise ST segment, number of major vessels (0-3) colored by flourosopy, thal	diagnosis of heart disease
Synthetic	1000	2	x_1, x_2	y

We employed Matlab 2018B to implement the aforementioned microaggregation algorithms ([2, 27, 42]), except for Mondrian, as well as to deploy the evaluation of perturbed data sets, and to process and plot results. Said evaluation implies loading data, building machine learning models over it, and applying such models over new data to measure classification accuracy, F-measure, and distortion. The implementation of Mondrian was written in Python and was taken from [83]. Since the

reconstruction method applied by Mondrian returns intervals instead of single values for each microaggregated attribute, we adapted this reconstruction such that the multidimensional hyperrectangles (microcells) were replaced by their corresponding centroids. The exploratory analysis to define the best suitable classification algorithm for each data set is performed with the Classification Learner application included in Matlab 2018B and then the model training and evaluation were automatized using specific embedded functions for each algorithm.

4.3.3 Methodology

The experimental methodology used to assess the performance of microaggregation algorithms in terms of resulting empirical utility is basically the same followed in the previous chapter, which is particularly described in Sec. 3.2.4. Figure 4.1 synthesizes the main elements of such procedure.

First, the original data set is *preprocessed* through three steps. To start, since MDAV based algorithms only work with numerical data, any categorical values for quasi-identifiers are represented numerically (e.g., the values female and male for sex are replaced with 1 and 0). Moreover, for validation purposes explained in the next paragraphs, we split each data set in two sets: a training set and a test set such that the former's size is 3/4 of the data set. Afterwards, each column of the training set, involving only quasi-identifiers, are normalized such that each column has zero mean and unit variance. Note that normalization is useful to avoid the harmful impact on microaggregation resulting from attributes having different ranges.

Once normalized, the *microaggregation* algorithm is fed with the training set for data perturbation. Only in the case of MDAV with SD, confidential attributes are also considered since this algorithm exploits the statistical dependence between quasi-identifiers and confidential attributes. We use progressively increasing values of k to then measure the utility degradation of data due to k-anonymous microaggregation. Besides the generic privacy criteria k, other parameters are configured for some algorithms.

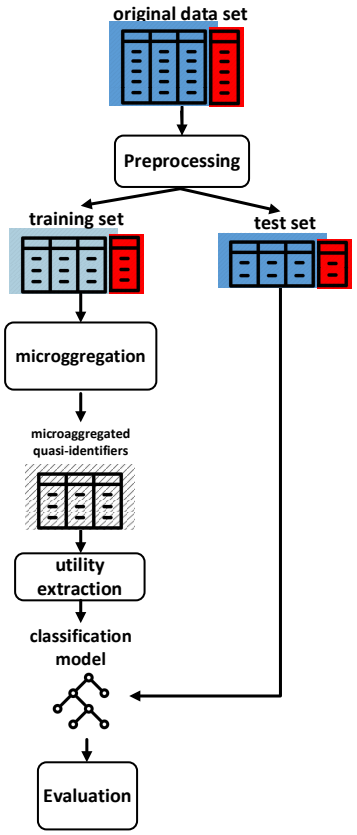


Figure 4.1: Experimental methodology followed to assess k-anonymous microaggregation algorithms in terms of the empirical utility preserved.

V-MDAV requires a gain parameter γ that we set in 0.9 as set by [42]. Additionally, MDAV with SD can be tuned by a λ parameter that regulates the tradeoff between distortion of quasi-identifiers and distortion of confidential attributes; we test different values of λ from 0 to 1 in order to get those showing the highest utility (maximum utility trace).

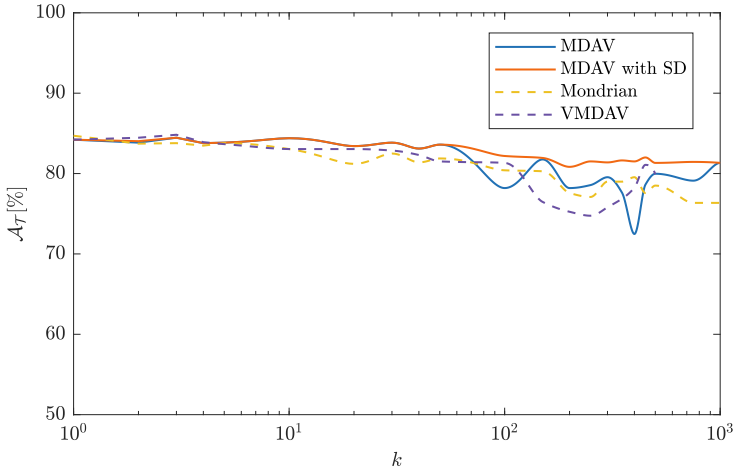
Once quasi-identifiers are perturbed, we implement the *utility extraction* phase. For this, we build a classification model using the microaggregated version of each data set as input. The algorithms showing the best performance in terms of utility are *boosting trees* and *logistic regression*, and the specific functions implemented in Matlab 2018b are used for training using 5-fold cross validation. Finally, each resulting classification model is evaluated over the test set originally extracted during the preprocessing phase; then accuracy and F-Measure are obtained. Namely, the machine-learned model built from microaggregated data is tested on a different portion of original data.

4.4 Experimental Results

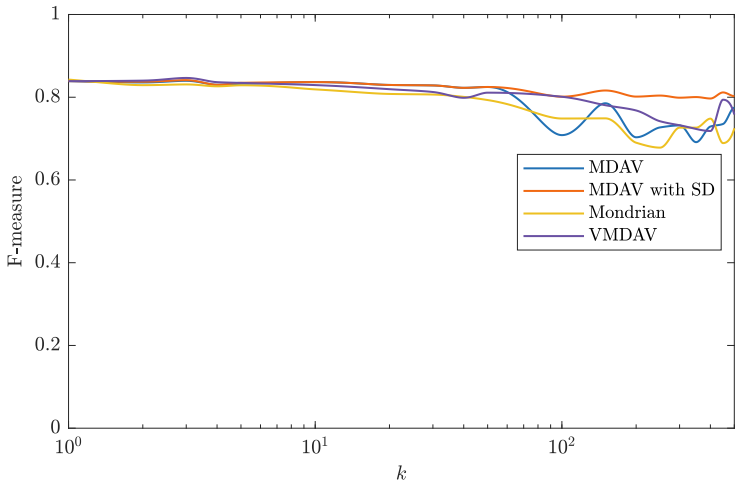
In this section, we present the results obtained from measuring the degradation of empirical utility of microdata due to k-anonymous microaggregation. This implies assessing the accuracy of machine learned models when trained over data microaggregated using an increasing value of k. Also distortion as MSE is measured in these terms to validate its capability to estimate the practical utility of data.

Said two main results are depicted in two groups of figures for each data set: one where accuracy and F-measure are shown and another where distortion is drawn against accuracy to unveil their potential correlation.

Our first experiment builds on the *UCI Adult data set*. In this particular case, we do not use the entire data set of more than 45 thousand records, but only 10% of them, i.e., a random sample that preserves the prevalence of the output (confidential) attribute. Suppressing potentially valuable data might reduce even more the data utility after microaggregation, an effect that we are interested in studying.



(a) Accuracy degradation



(b) F-measure degradation

Figure 4.2: Degradation of the empirical utility of the microaggregated "Adult" data set.

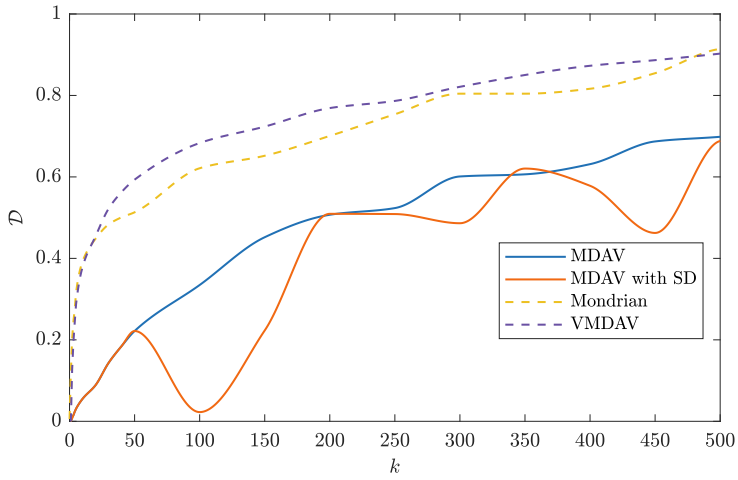
Accordingly, we illustrate in Figure 4.2 how empirical utility is affected when microaggregation is applied over the UCI Adult data set. As expected, data perturbation eventually renders data useless, as shown by the decreasing trend in accuracy as k gets higher values. Note that the lowest value in accuracy does not reach zero since, in the worst case, when the data input (quasi-identifiers) is completely perturbed, machine learned models predict based only on the prevalence of classes in the output data.

Despite this inevitable degradation in the long term, microaggregated data shows high levels of utility even up to $k = 50$. Namely, for such values of k, accuracy easily keeps greater than 80% for any of the four microaggregation algorithms evaluated. Interestingly enough, in the case of the UCI Adult data set, this means that said utility in terms of machine learning accuracy might be kept even when vast amounts of data are suppressed.

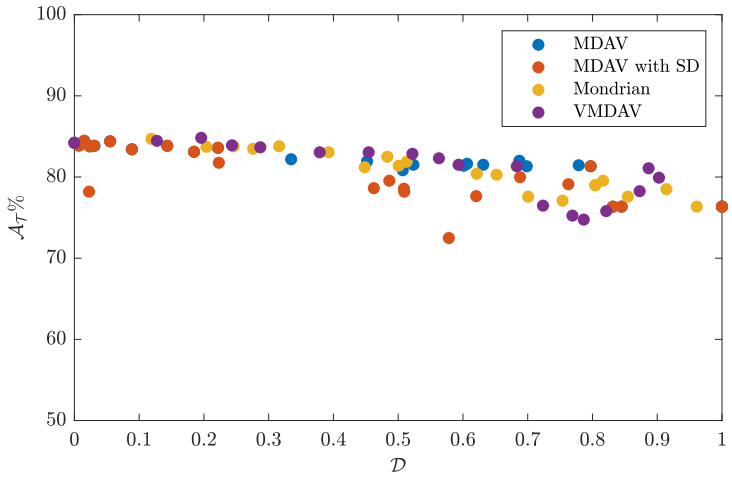
Furthermore, from Fig. 4.2a, utility is remarkably preserved by MDAV with SD. In fact, accuracy do not drop below 80%, even for $k = 1000$. Similar encouraging results are obtained when measuring F-measure as illustrated in Fig. 4.2b. Besides, we can see that the original MDAV is the second best performer in regards to practical utility, at least up to $k = 60$. On the other hand, V-MDAV and Mondrian are the worst performers, although for very few small values of k, V-MDAV gets the best results.

When plotting the evolution of distortion as k is progressively increased, while microaggregating the Adult data set, Fig. 4.3a confirms that MDAV with SD applies less distortion (as measured through the combined metric proposed by [2]) than the other algorithms. Original MDAV repeats as the second best performer, now in terms of MSE, but Mondrian and V-MDAV seem to introduce more perturbation. In any case, distortion grows exponentially so, according to this metric, data would render useless very quickly. In fact, when $k = 50$, MDAV and MDAV with SD would have injected more than 20% of distortion while Mondrian and V-MDAV more than 40%.

The utility metric obtained empirically may not go hand in hand with a more syntactical measure based on MSE. This is confirmed in Fig. 4.3b where we plot accuracy vs data distortion. The scatter plot shows that, although the distortion



(a) Distortion measured for different values of k-anonymity.



(b) Representation of accuracy vs distortion

Figure 4.3: Distortion of the microaggregated "Adult" data set. The distortion corresponding to MDAV with SD is measured according to the hybrid metric $\mathcal{D}$ proposed by [2] and presented in 2.2.6.

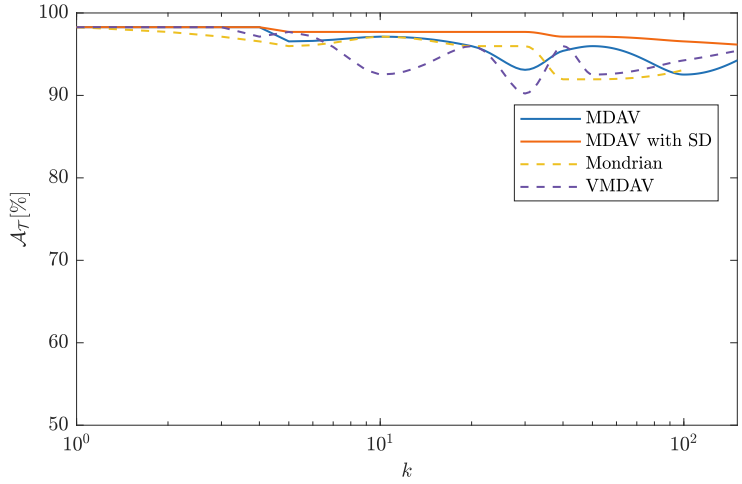
increases, e.g. up to 0.5, the corresponding accuracy keeps more or less stable in 80% for all the microaggregation algorithms. This implies that distortion, in general, is not a good predictor of the practical utility of microaggregated data, at least in the application domain here studied.

As described in Section 3.2.3, the results aforementioned are corroborated in experiments with three more data sets. When testing the *Breast Cancer Wisconsin* data set, the resilience of empirical data utility manifests again when k-anonymous microaggregation is enforced. Once again, the benefits of MDAV with SD are evident when outperforming the accuracy obtained by the rest of algorithms, as can be seen in Figure 4.4. Beyond the clear superiority of MDAV with SD, it is not clear for this data set which of the other algorithms performs the best in terms of accuracy.

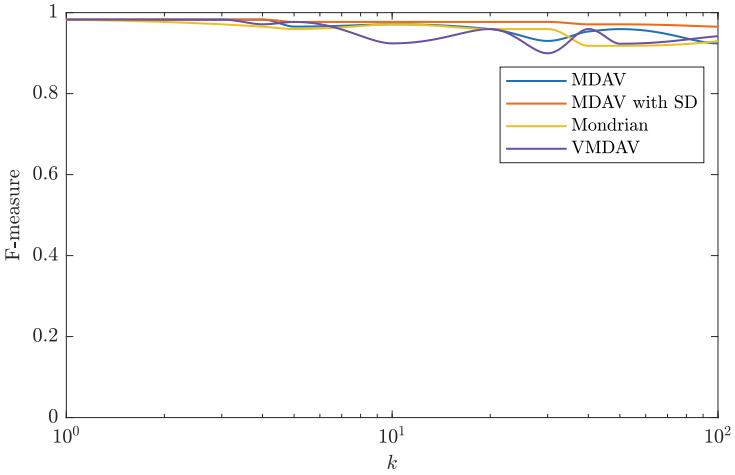
Regarding the standard metric of utility (degradation), note in Figure 4.5a that MDAV with SD also has the least distortion, that Mondrian performs the worst, and that both MDAV and V-MDAV show a similar distortion trend. As with the previous data set, the results of distortion hardly explain the practical utility of microaggregated data because it can be seen in 4.5a that accuracy does not vary as significantly as MSE when measuring the impact of microaggregation algorithms.

Figures 4.6, 4.7, 4.8, and 4.9 illustrate the results of assessing microaggregation algorithms over the Heart Disease and synthetic data sets. For both of them, microaggregation, in general, performs quite well in terms of practical utility (see Figures 4.6a and 4.8a) while distortion grows much faster (see Figs. 4.7a and 4.9a). In any case, the original MDAV exhibits anonymized data with lower distortion and stable accuracy, only improved by its statistically dependent variant, MDAV with SD.

We must note that in Fig. 4.7b, for values of distortion greater than 0.5 larger values of distortion do correspond to lower accuracies, particularly when contrasting with the results depicted in other figures. However, when distortion is lower than 0.5, accuracy is not degraded, so we feel that this behavior still fits our claim that distortion is not a great predictor of the practical utility of microaggregated data. Namely, accuracy is not degraded for this stretch of distortion increase. In any case, each of the different processing techniques may definitely have a particular effect on the intrinsic trends within a data set. Since we may not be able to model said trends,

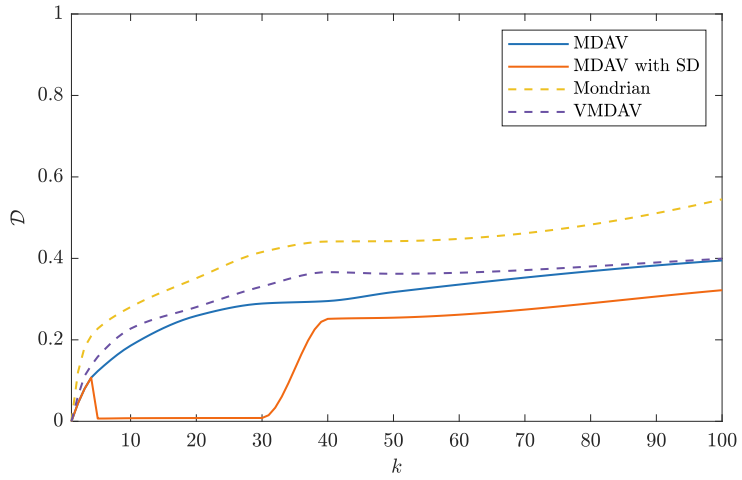


(a) Accuracy degradation

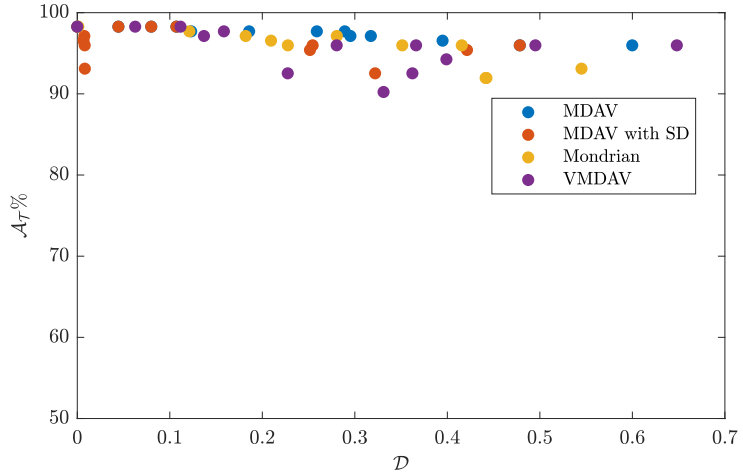


(b) F-measure degradation

Figure 4.4: Degradation of the empirical utility of the microaggregated "Breast Cancer Wisconsin" data set.

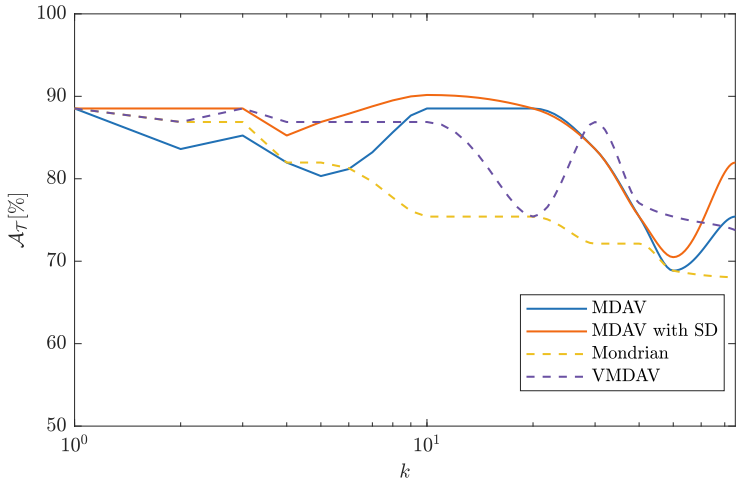


(a) Distortion measured for different values of k-anonymity.

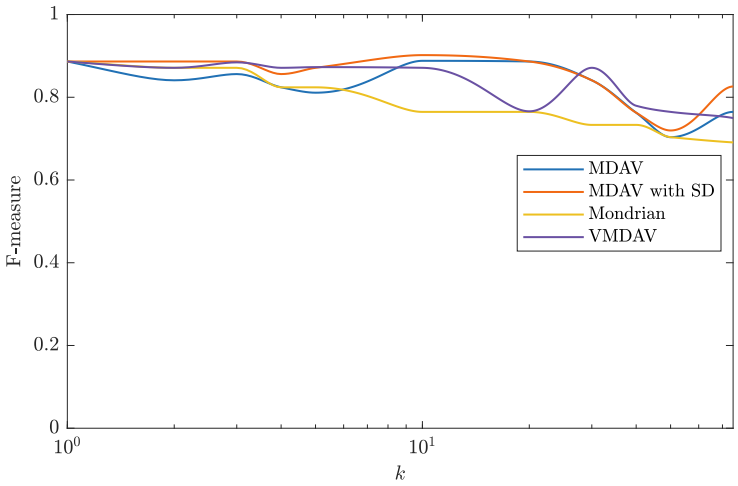


(b) Representation of accuracy vs distortion

Figure 4.5: Distortion of the microaggregated "Breast Cancer Wisconsin" data set. The distortion corresponding to MDAV with SD is measured according to the hybrid metric $\mathcal{D}$ proposed by [2] and presented in 2.2.6.

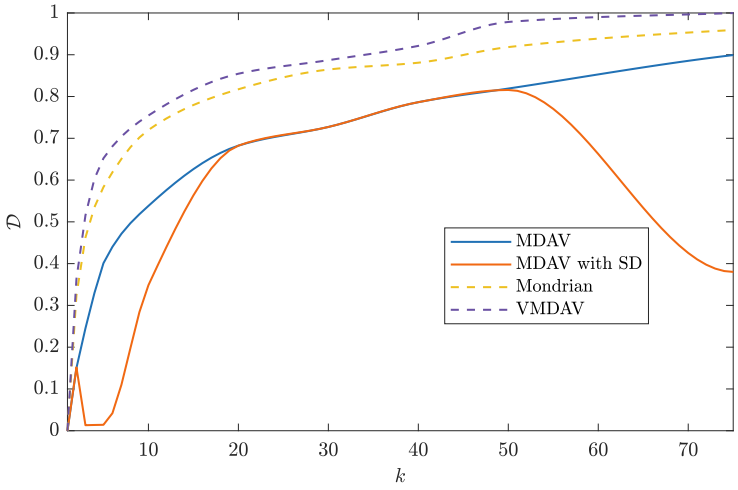


(a) Accuracy degradation

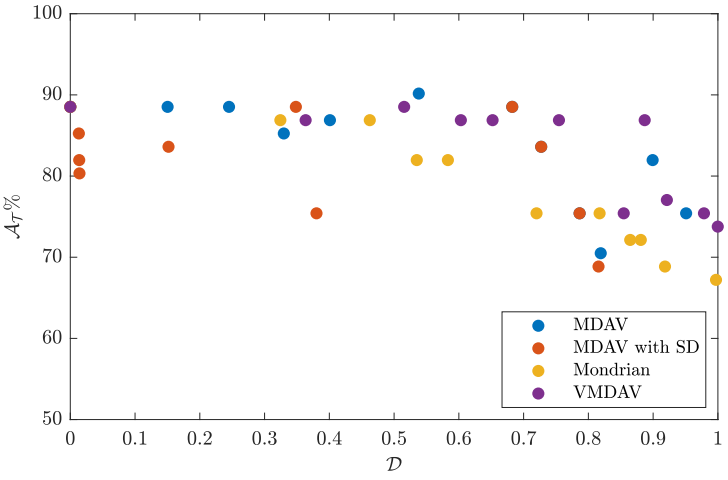


(b) F-measure degradation

Figure 4.6: Degradation of the empirical utility of the microaggregated "Heart Disease" data set.

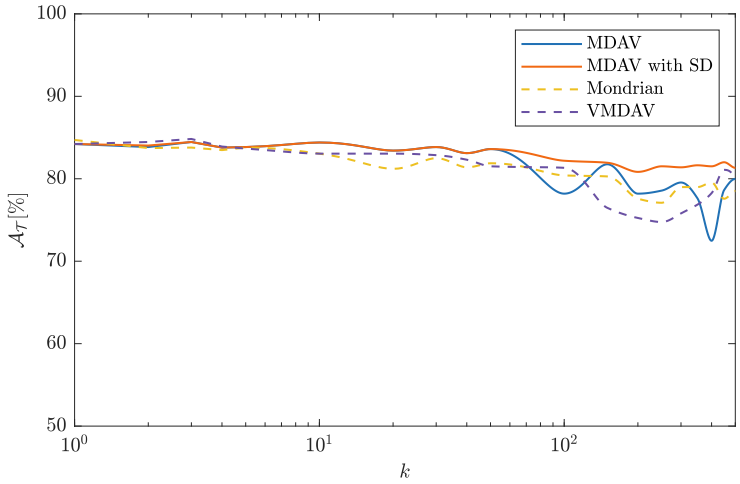


(a) Distortion measured for different values of k-anonymity.

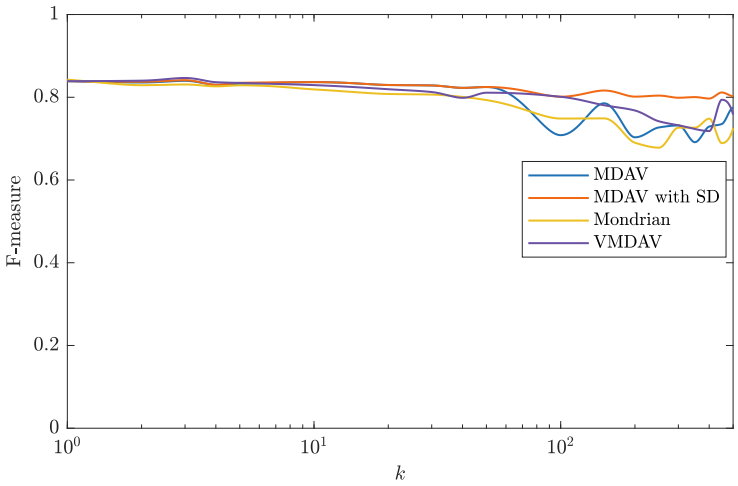


(b) Representation of accuracy vs distortion

Figure 4.7: Distortion of the microaggregated "Heart Disease" data set. The distortion corresponding to MDAV with SD is measured according to the hybrid metric $\mathcal{D}$ proposed by [2] and presented in 2.2.6.

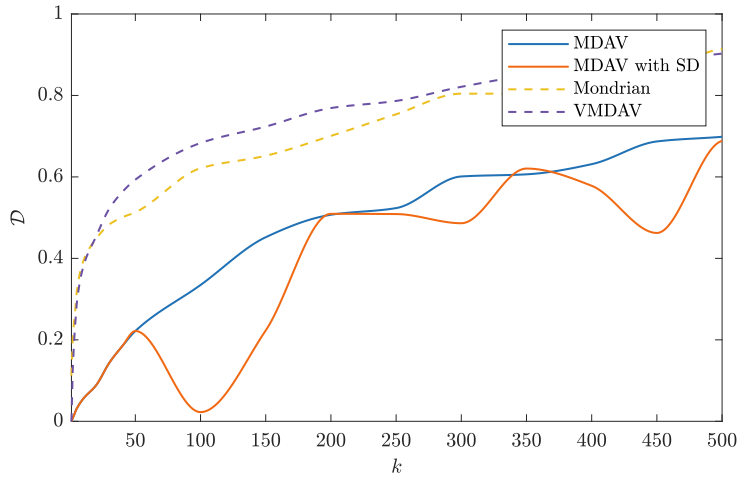


(a) Accuracy degradation

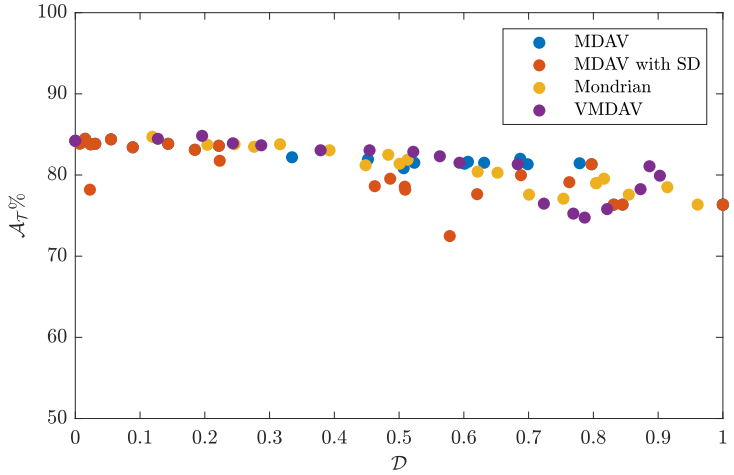


(b) F-measure degradation

Figure 4.8: Degradation of the empirical utility of the microaggregated synthetic data set.



(a) Distortion measured for different values of k-anonymity.



(b) Representation of accuracy vs distortion

Figure 4.9: Distortion of the microaggregated synthetic data set. The distortion corresponding to MDAV with SD is measured according to the hybrid metric $\mathcal{D}$ proposed by [2] and presented in 2.2.6.

we performed several tests on different data sets to estimate the *general* impact of each protection approach.

Finally, V-MDAV and Mondrian show interesting results on these data: while the former would spawn more distortion than the latter, V-MDAV apparently preserves better the data utility when measured as accuracy of the resulting machine learning model.

4.5 Discussion

Our systematic experimentation shows that k-anonymous microaggregation has a benevolent, still destructive, effect on microdata in terms of its empirical utility, which is measured as the accuracy of learning models built from such data. Namely, while meeting a k-anonymity criteria, microaggregation preserves data utility even for high values of k, as previously pointed out in chapter 3 for MDAV. This effect is attributed to the averaging operations to find a centroid that would be denoising the data, making it more resistant to perturbation.

In addition, although said averaging, inherent to microaggregation, might be even convenient, the distortion metric based on MSE would measure it as utility degradation. In this sense, MSE is a pessimistic metric that, in general, is not able to predict the practical utility of microaggregated data in this domain. As a matter of fact, not even the combined distortion metric proposed by [2] for MDAV with SD is capable of estimating such practical utility, despite its great performance in terms of accuracy.

The results obtained by MDAV with SD confirm that adapting privacy protection mechanisms to the intrinsic statistical properties of microdata and to the specific application domain might open the door to interesting improvements in utility preservation. This approach has not been addressed for microaggregation algorithms and particularly for MDAV-based approaches, so there is an appealing avenue for future work.

The "positive" impact of anonymization algorithms is indirectly reported by previous work that accounts for, e.g., the reduced degradation of obfuscated data under certain conditions [55, 56], and the beneficial contribution to utility of some anonymization techniques ([53]) that may act as feature selection mechanisms, particularly when the protection strategy is selectively tailored to the application domain [58].

Accordingly, from the results depicted when representing accuracy vs distortion along several experiments, we corroborated that distortion may not predict the practical utility of microaggregated data. Although minimum values of accuracy are measured when having maximum values of data distortion and vice versa, as the distortion increases, the accuracy metric does not vary in the same magnitude, even when distortion takes values as high as 0.5. Namely, while a syntactic utility metric indicates that microaggregated data is, e.g., 50% distorted, a more empirical utility metric suggests that such data might be almost as useful as the original.

V-MDAV and Mondrian show, in general, a lower performance than the ones of MDAV and MDAV with SD in terms of both distortion and accuracy. However, since the strategies of V-MDAV and Mondrian operate on the internal distribution of the microdata set, such results could vary according to the data set being microaggregated.

Beyond the promising results, it is worth noting that our approach has inevitably some limitations that arise, essentially, from the bounded evaluation context we have defined . For instance, the application domain, where utility is empirically measured, is binary classification. However, many other domains may exist where utility is extracted differently.

Furthermore, a statistical dependence should exist between quasi-identifiers and confidential attributes such that something can be learned and preserved when microaggregating. Evidently, if this is not the case, another utility metric should be assessed.

4.6 Conclusion

In this chapter, we have corroborated the intuition that further catching and processing the statistical properties of microdata (e.g., the statistical dependence between quasi-identifiers and confidential attributes) when building microaggregation algorithms cause an additional slowdown in the degradation of empirical utility. This is clearly evidenced by MDAV with SD through our extensive tests. Sadly, the hybrid metric created to assess its performance is not a good enough predictor of the practical utility of microaggregated data as would be expected. However, high values of distortion measured using such metric do suggest some correlation with metrics of empirical utility.

Although Mondrian and V-MDAV consistently perform worse than MDAV and MDAV with SD, the two former algorithms behave differently between each other in terms of accuracy and MSE-based distortion. This would evidence the dependence of their performance on the internal distribution of the data set, as claimed by their creators. Such dependency calls again our attention to the need of considering the application domain of data (size, exploitation mechanisms, distribution of tuples) when designing or adapting privacy protection.

Finally, we confirmed in this chapter that k-anonymous microaggregation algorithms are able to preserve much of the data utility while protecting the privacy of each subject in groups of k individuals. Their clustering and averaging operations seem to contribute to filter, normalise, or consolidate the statistical information within microdata, e.g., when exploiting data through machine learning applications.

Chapter 5

Preserving empirical utility of microaggregated data through LDA

5.1 Introduction

Modern technologies and massive access to them by billions of people have enabled the generation of vast amounts of data. Also, more powerful and sophisticated information systems are developed to exploit such data with the aim of getting unprecedented intelligence and personalization. The potential benefits of these technologies are countless in several fields such as healthcare, advertising, and even industrial engineering [76–78]. For most of such fields, more utility can be mined from data to unveil qualitatively superior insight into challenges and opportunities that may otherwise remain undiscovered [3, 4].

A compelling example of application where data utility is absolutely critical is, undoubtedly, health and, particularly, precision or personalized medicine. In this domain, a large data sample could reveal otherwise subtle patterns. To illustrate this point, we recall a well-known medical experiment conducted in 1989, in which a large number of participants in a study allowed practitioners to find out a slight but clinically relevant effect of aspirin tablets in participants who had a myocardial

infarction [4]. From a sample of 22,071 individuals, the study found that heart attacks were 0.77% less frequent when participants took an aspirin table every other day, a phenomenon that would have been much harder to observe without such a large sample.

Unfortunately, as explained in Sec. 2.1, exploitation of data encompasses serious privacy risks when information is associated with individuals. Since abundant details are usually collected about them, even after suppressing identifier attributes such as full names, other, apparently innocuous, personal attributes (quasi-identifiers), could still be used to re-identify an individual [23]. Thus, if a sensitive attribute were disclosed along with other information, re-identification would enable an attacker to associate an individual with such attribute, thus violating her privacy. But this risk is exacerbated by the fact that data has become a core asset for companies [79], so there is a great incentive to exploit, share, and even sell data to maximize profit.

We discussed in 2.2 that SDC is commonly used to tackle these privacy risks when disclosing microdata files. SDC techniques build on perturbing quasi-identifier attributes to de-identify records, a process also called *anonymization*. The privacy models enforced through user data perturbation, e.g., k-anonymity [7, 11] or ϵ-differential privacy [12], are usually conditioned by a privacy parameter that defines an upper bound on the re-identification risk.

k-anonymous microaggregation, as probed in chapters 3 and 4 is a high-utility mechanism to protect privacy in microdata by obfuscating demographic attributes. Carefully aggregating these attributes, a minimum level of distortion must be applied to original data. In fact, on the last two chapters, we have found that k-anonymous microaggregation is an excellent approach for applications requiring the preservation of data utility [16].

Obfuscating data to protect privacy naturally affects its resulting utility [24]. This was briefly discussed in Sec. 2.3. Consequently, there is a trade-off that must be addressed so that data exploitation keeps feasible and usable. In this line, the role of SDC mechanisms is guaranteeing a given level of privacy while preserving (some of) the utility of anonymized data.

Also in Sec. 2.3 we addressed some ways to measure the impact of these mechanisms on the utility of data. In general, it has been commonly measured using standard, but merely syntactical, metrics, such as mean-squared error (MSE). However, to capture the practical utility of anonymized data, other metrics related to its application domain might be more relevant. Since a very common domain of application is building machine learning models, accuracy or F-measure of these models are reasonable metrics of empirical utility.

Aiming to find a balance among privacy and empirical utility, some research is devoted, not only to design new less-"destructive" protection algorithms, but also to "adapt" already existing algorithms that increase the resulting utility of anonymized data. In this line, recent work is increasingly oriented to propose semantic (more empirical) approaches to the preservation of data utility when protecting privacy [84–86]. Part of this work was described in the two previous chapters.

Although utility is certainly the *raison d'etre* of our effort, another parameter key to privacy protection usability is computational complexity. If protection mechanisms cannot cope with the (sometimes real-time) requirements of modern applications, they render unusable no matter how much utility is preserved. A few works have been proposed recently in this direction [16, 17] and are presented in the next chapter.

In this chapter, we present and assess a strategy to preserve (empirical) utility of data after a k-anonymous microaggregation algorithm is applied. By representing original data in a new rotated and scaled domain, we adjust the implementation of the microaggregation algorithm to the specific application domain of data, which in this case is also binary classification. As a result, the error of the machine learning model, when evaluated over new testing data, was reduced, at no cost, even for high anonymity levels.

The anonymization method addressed in this work is computationally and functionally efficient since the utility of data is preserved while the privacy level offered by an underlying microaggregation algorithm is left intact, at no additional cost in terms of running time.

Interestingly, data utility preservation at no (computational) cost could be a great incentive to adopt privacy protection technologies. In fact, some big tech companies

are turning their privacy stance into a huge competitive advantage. Thus, the companies that best adapt their operation to privacy requirements (preserving data utility and algorithm usability) will be in better position to exploit such advantage. In this context, these parameters could become a powerful value generator.

The work presented in this chapter is summarized in the next items.

- We develop a method to preserve empirical data utility when microaggregating data. Namely, it is built on a practical metric derived from the application domain of data which is binary classification in this case.
- This is done by leveraging on Linear Discriminant Analysis to find the direction of maximum discrimination within data, which enables the microaggregation mechanism to adapt its anonymization strategy to binary classification.
- This approach also involves weighting (by scaling) said discriminating direction in such a way that distances in this direction are penalized when building k-anonymous groups. The upshot is that k-anonymous microcells are grouped to not overlap with the classification threshold.
- To give some intuition regarding our approach, we included in this work a running example to illustrate the transformation applied to data for preserving utility.
- We systematically evaluate this method on several data sets, both real and synthetic, using different machine learning algorithms, and increasing anonymity levels and scaling factors.

Chapter outline

The rest of this chapter is organized as follows. Sec. 5.2 formally presents the proposed formulation of our privacy preserving approach, while Sec. 5.3 presents the experimental analysis and outcomes of this strategy. Finally, conclusions are drawn in Sec. 5.4.

The work presented in this chapter was accepted to be published in Elsevier Engineering Applications of Artificial Intelligence [14].

5.2 Application of LDA to k-anonymous microaggregation

To explain the concept of LDA and then illustrate its application to preserving data utility while implementing k-anonymous microaggregation, we next introduce some principles and notation that are explained later through a running example. This example builds on a synthetic data set, generated according to the scenario and parameters described below.

5.2.1 Introduction to the preservation of the utility of microaggregated data through LDA

Following the scenario stated in chapters 3 and 4, we use binary classification as the application domain since machine learning is increasingly used to exploit data. Namely, we assume that data requiring anonymization through microaggregation will be further processed to extract a binary classification model.

However, k-anonymous microaggregation groups records (building cells) without considering any application domain, so both privacy protection and data exploitation might be naturally incompatible in terms of utility preservation. Thus, our aim is to modify this aggregation process such that it adjusts to the binary classification algorithm while privacy is still protected.

Binary classification, in general, obtains a threshold that enables classifying the elements of a given set that, in our scenario, consists of multidimensional numeric points. Since k-anonymous microaggregation groups such points in cells without any particular shape or direction, it is likely that said threshold will split some of the cells, implying that their corresponding centroids misrepresent their aggregated points when obtaining a classification model. In order to address this issue that would affect the resulting utility of data, we resort to LDA.

LDA [87, 88] is a method commonly used as a preprocessing step before implementing machine learning classification. It aims at modeling the difference between classes of data by projecting a data set onto a lower-dimensional space. To do this, loosely

speaking, LDA looks for maximizing the distance (separability) among the data of different classes (their means) while minimizing the variation within each class. Such projection enables good class separability and even a reduction of computational costs on classification tasks ([89]) .

LDA and Fisher's linear discriminant technique ([88]) are often used interchangeably, but there is a subtle difference. On the one hand, with Fisher's linear discriminant, we seek to maximize the ratio between the determinants of the between-class covariance and the within-class covariance. On the other hand, LDA fits a Gaussian homoscedastic mixture to the generative model via maximum likelihood estimation. The original linear discriminant was described for a 2-class problem, and it was generalized later for multiple classes. Both methods result in the same direction of best discrimination for the corresponding class from the multivariate observation.

Interestingly, such direction of best discrimination can be used to tailor the microaggregation process such that microcells are built aligned to such direction by, basically, a *rotation*. In addition, we propose a *weighing* step of the records. Both of this building blocks (rotation and scaling/weighting) aim at increasing the separability of the two classes embedded in data to facilitate the construction of utility-preserving microcells. Namely, our approach would be implemented before applying the original microaggregation process, as depicted in the scheme of Fig. 5.1.

In the next subsections we try to depict by example how this direction of best discrimination is obtained.

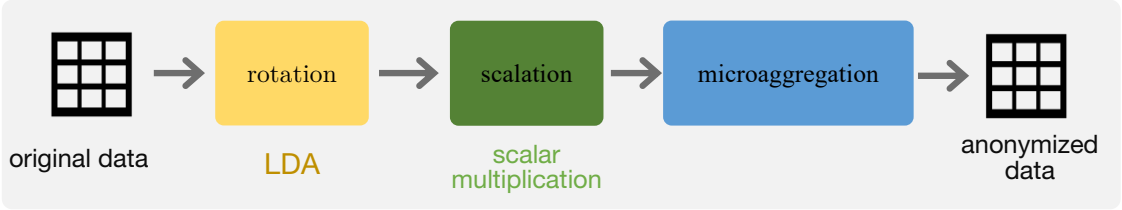


Figure 5.1: Main building blocks of our proposal to preserve utility from microaggregated data.

5.2.2 Integration of LDA into k-anonymous microaggregation

In this section we explain our proposed method in detail. We include a description on the scope considered –in particular for data utility exploitation– and a step-by-step illustration of the integration of LDA into k-anonymous microaggregation.

5.2.2.1 Scope and preliminary notation

As stated in Sec. 5.1, the scope of this work, in terms of data utility extraction (and application domain of data), is binary classification. Thus, we next make a brief description of the main elements of this scenario, the math connecting them, and the notation that will be used along the rest of this section.

First, consider a population of patients whose attributes (e.g., height/weight) and diabetes status are studied to build a model capable of detecting diabetes in new individuals, based on said attributes, i.e., a binary classification problem.

Then, let x be a numeric random variable (r. v.) in $\mathbb{R}^n$, i.e., an n-dimensional vector representing these attributes for an individual. Also, let Y be a binary random variable representing whether a patient has a diabetes condition ($Y = 1$) or not ($Y = 0$), i.e., a label. Let μ_1 and μ_0 be the mean vectors of the diabetic and non-diabetic subpopulations, respectively, considering only their attributes. Accordingly, let Σ_1 and Σ_2 be the corresponding covariance matrices, and p the prevalence of diabetics in this example. Finally, let

$$\Sigma_W = (1-p)\Sigma_0 + p\Sigma_1$$

be the within-class covariance matrix associated to the two-class data mentioned above. For single-class Fisher's discriminant, there is no need to compute the between-class matrix

$$\Sigma_B = (1-p)\, p(\mu_1 - \mu_0)\, (\mu_1 - \mu_0)^T.$$

Based on the previous definitions of x and Y, suppose a data set with n numerical attributes, resembling n quasi-identifiers, and a binary label as the confidential attribute. Besides, assume Y is to some extent predictable from the quasi-identifiers

represented by x so the data set is useful in the realm of machine learning classification. Accordingly, consider a *generative model* defined by

$$\begin{aligned} &x|Y \sim \mathcal{N}(\mu_1,\, \Sigma) \\ &x|\bar{Y} \sim \mathcal{N}(\mu_0,\, \Sigma), \text{and} \\ &p \end{aligned}$$

that builds a Gaussian homoscedastic mixture fit via machine learning estimation. After characterizing a generic representation of the data on which our approach would be applicable, below we describe the method for preserving its utility when microaggregated.

5.2.2.2 Data rotation and scaling

Our strategy for preserving data utility when microaggregating consists of building microcells shaped in parallel to a discriminative direction and scaling data; all this with the aim to increase the separability of numeric records when a learning model is built. Accordingly, the following paragraphs describe the steps for finding such direction and implementing scaling of data.

To discern between Y and $\bar{Y}$, we use a *discriminative model* defined by $\mathrm{P}(Y|x)$, i.e., the probability *a posteriori* of the event Y. Recall that the corresponding Bayes factor (BF)

$$\frac{\mathrm{P}(x|Y)}{\mathrm{P}(x|\bar{Y})}$$

can be perfectly used as the discrimination function since it is a minimal sufficient statistic for Y from x under this homoscedastic and multivariate Gaussian model.

If we obtain the natural logarithm of the BF (which can be seen as a unit change), it can be finally expressed as a simple scalar product, i.e.,

$$\ln \mathrm{BF} = \left\langle \mu_1 - \mu_0,\, x - \frac{\mu_0 + \mu_1}{2} \right\rangle_{\Sigma_W^{-1}} = (\mu_1 - \mu_0)^T \, \Sigma_W^{-1} \left(x - \frac{\mu_0 + \mu_1}{2} \right).$$

We obtain a linear discriminant function whose *direction of maximum discrimination* (given that Σ_W is symmetric and applying some properties of the matrix multiplication) can be expressed as

$$U = \Sigma_W^{-1}(\mu_1 - \mu_0).$$

In general, for the multi-class Fisher's discriminant, the compression matrix U contains the orthonormal eigenvectors associated with the $L-1$ largest eigenvalues of $\Sigma_W^{-1}\,\Sigma_B$ (regarded as the solution to a generalized eigenvalue problem), where L denotes the number of classes. The optimization criterion is

$$\max_U \frac{\det(U^T \Sigma_B\, U)}{\det(U^T \Sigma_W U)}.$$

Rotation. LDA projects the data set (the part defined by x) on U, which defines the direction on which the distance among the different classes of the data is maximized while their variance is minimized. As a note, this direction can be more efficiently calculated, e.g., in MATLAB, without resorting to the calculation of an inverse matrix but by solving a system of linear equations.

Then, with full QR decomposition, we find an orthonormal base extension of U, V (an orthonormal base where one of the axes is U). This contains the normalized Fisher's discriminant direction. Next, the original attributes of the data set, which are points in the Euclidean space, are represented in terms of the new axes defined by V. Thus, we get the projection

$$x' = V^T\, x,$$

where x' is a transformed version of the original attributes represented by x. The first component of x' is the linear combination of the original attributes that best discriminate between the classes, while the rest can be considered less relevant.

Scaling. In line with the spirit of increasing the separability of two-class data, we complement the application of LDA with another strategy. We propose weighting the *first* transformed component, that is, first component of the LDA projection, by a factor $\alpha \geq 1$. In this manner, distance and distortion calculations will penalize the discrimination direction. Namely, we increase the distance among points in this direction so that they can be more easily grouped into microcells that do not overlap with the classification threshold. This scaling operation turns the new representation of data into the product $S\,V^T\,x$, for $S = \mathrm{diag}(\alpha, 1, \ldots, 1)$. Note that the scaling affects the first rotated component only, and this scaling can be regarded as a multiplication by a diagonal matrix. This product can be equally computed as $(S\,V^T)\,x$ or $S\,(V^T\,x)$,

but if the data set to be transformed is very large, the former is much faster. Namely, this scaling by S can be regarded as matrix multiplication and the rotation by V can be associatively lumped into a transformation by a linear operator incorporating both scaling and rotation, for efficiency.

In Fig. 5.2 we summarize the main building blocks of the theoretical analysis of this proposal.

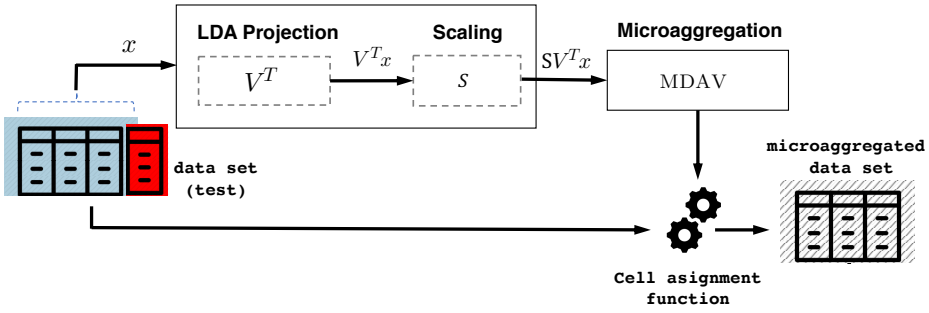


Figure 5.2: Main building blocks and theoretical operations involved in our proposal for preserving data utility. This can also be read as the particular experimentally methodology followed for its implementation.

To graphically illustrate the wellness of our utility-preserving methods, we next depict their application in a simple scenario. In Sec. 5.3 we assess them experimentally using real data sets.

From the scenario and generative model proposed in Sec. 5.2.2.1, assume a toy synthetic dataset of 1000 records, with two numerical quasi-identifiers (say, e.g., weight and height) x_1, x_2, and a corresponding binary confidential attribute y for each individual (e.g., diabetes status). For the sake of clarity, let us illustrate the distribution of these quasi-identifiers in Fig. 5.3, where x_1 and x_2 are plotted as points in two dimensions in the Cartesian plane. Evidently, the confidential attribute y is somewhat dependent on the contribution of the quasi-identifiers x_1, x_2, so a model can be learned to predict the former one from the latter ones.

If k-anonymous microaggregation through MDAV is employed to protect the identity of data owners, these points are grouped in cells of size k as graphically depicted in Fig. 5.4. As can be seen in this figure, microcells are built considering only relative closeness among points, so they tend to be grouped more or less equidistantly from

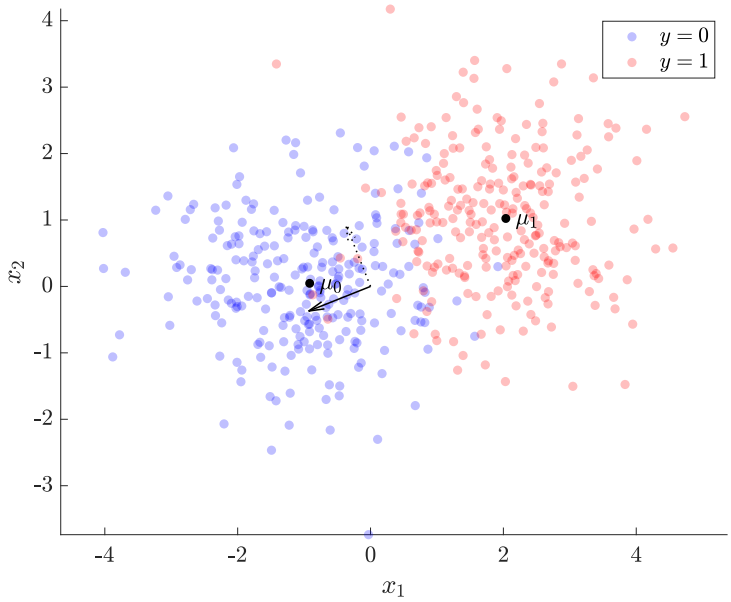


Figure 5.3: Depiction of the quasi-identifiers (x_2 vs x_1) of our toy synthetic data set. Samples are colored according to their class, y; blue for $y = 0$ and red for $y = 1$. The direction defined by mean points of both classes is the direction of maximum discrimination on which data will be projected to maximize its separability.

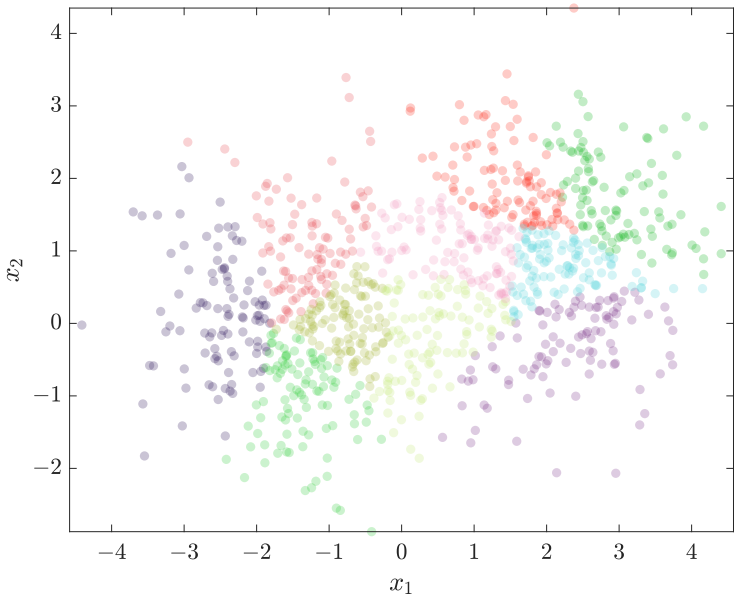


Figure 5.4: Microcells of samples obtained by applying k-anonymous microaggregation with MDAV on our toy synthetic data set ($k = 100$). Note how the single criteria to group points in clusters is their relative closeness.

a centroid. This produces "thick" groups with no particular orientation in any direction. Such thickness, and the omnidirectional distribution of cells, however, makes them more prone to fall over the classification threshold; thus, their corresponding centroids will likely misrepresent such points when a classification model is built. This evidently may contribute to reducing data utility.

Finding a maximally discriminative direction over which this data can be represented, LDA seems to be a convenient technique for k-anonymous microaggregation in terms of resulting empirical utility of anonymized data. In practice, LDA will maximize separation of data of the two classes and the inherent distortion would be weighted by an empirical parameter α. While in Fig. 5.3 we draw such direction, defined by the mean points of both classes of data, in Fig. 5.5 we can see the LDA projection of the data set on this direction. Said otherwise, data is rotated and scaled in this direction.

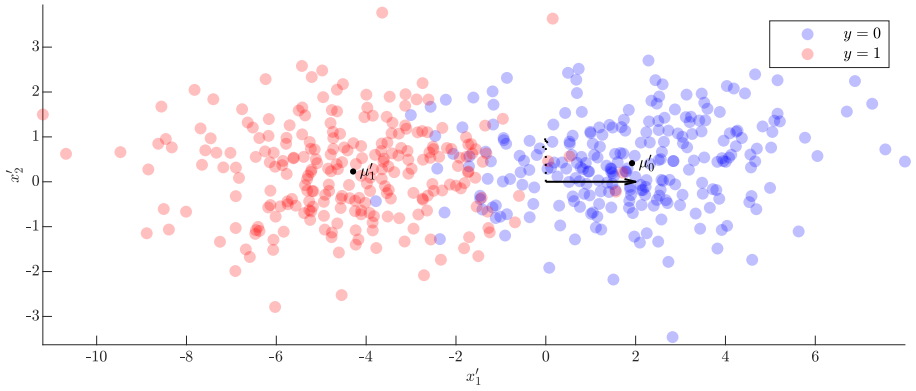


Figure 5.5: LDA projection of our toy synthetic data set on the direction of maximum discrimination x_1'. Scaling is also applied with $\alpha = 2$.

5.2.2.3 Brief discussion

Within this new representation of data, MDAV builds "thinner" microcells in the direction of maximum discrimination. Namely, increasing the separability between classes will enable MDAV to tailor the shape of resulting microcells to the intrinsic classification threshold estimated by LDA. This new distribution of cells is illustrated in Fig. 5.6 for our toy example. There we plot the microcells built from the original

data set, following the microcell assignment obtained from microaggregating the LDA projection of the data set.

Since the resulting cells are clearly distributed in parallel to the intrinsic classification threshold gotten by LDA (Fig. 5.6), it is much less likely that such threshold falls over multiple cells. Thus, very few centroids would misrepresent data when a machine learning model is built from microaggregated data, preserving, in this way, its utility.

Besides preserving data utility, our method does not involve any additional computational complexity since the microaggregation process is not essentially changed but the representation of data before being anonymized. Fortunately, rotating and scaling data to change its representation are tasks performed once and does not entail significant complexity with respect to that of the iterative and complex process of microaggregation.

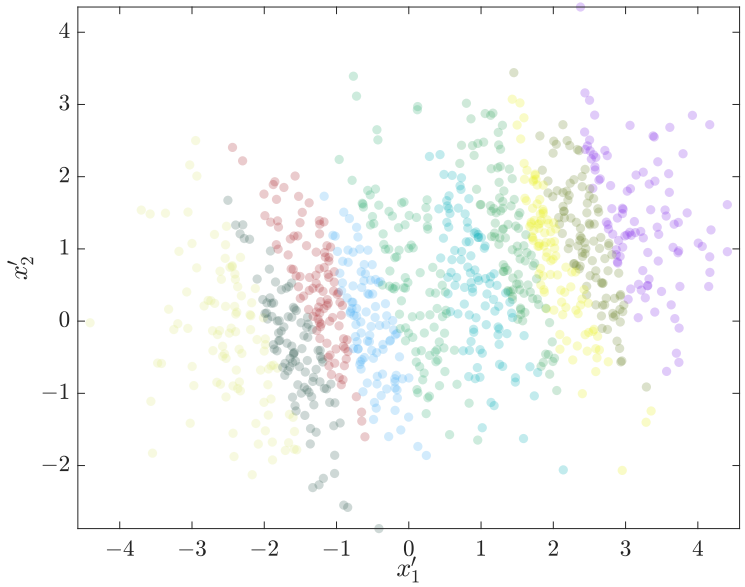


Figure 5.6: Microcells built in our original toy example by using the microcell assignment obtained from microaggregating the LDA projection of the data set ($k = 100$). Note how microcells are thinner in the direction of maximum discrimination, favoring the separation of the two classes by a classification task.

5.3 Experimental evaluation

In this section we aim at describing the general context of the evaluation of our proposal on preserving the utility of anonymized data. For this, we describe the scenario assumed, the evaluation criteria (privacy and utility metrics), the tools used, and the phases implemented.

5.3.1 Evaluation scenario

The evaluation revolves around the standard attack model in the SDC literature ([68]). To start, we assume a microdata set that needs to be released for research purposes. This microdata set has quasi-identifiers and a single confidential attribute. In this case, the utility of data lies in the statistical dependence among quasi-identifiers and a confidential attribute (such a diagnosis). In particular, such dependence would derive in a learning model to classify other individuals, e.g., as sick or healthy. In this data mining context, quasi-identifier records used to build the model are *input samples*, while the confidential records are *output labels*.

Besides, due to evident privacy concerns in this context, k-anonymous microaggregation is applied over quasi-identifiers to protect the privacy of data subjects. Thus, instead of original data, anonymized quasi-identifiers along with untouched confidential attributes are released. However, the utility of anonymized data would be undermined since obfuscating quasi-identifier records will most likely affect the quality of statistical trends embedded.

As mentioned in previous sections, to preserve such utility, we propose using LDA and scaling on the data as part of the microaggregation process. To assess this approach, we test it on several data sets and compare the resulting utility with that of data anonymized only with MDAV.

5.3.2 Data sets

With respect to the data to assess our mechanism, we used essentially the same data sets tested in chapter 4 including real and synthetic data sets. Namely, given the

scenario proposed in this work, two main conditions were met when selecting data, in particular for real data sets. First, we looked for microdata sets, i.e., data containing demographic information about actual individuals, such that a privacy concern might be involved. Second, we required data whose confidential attribute evidenced a clear statistical dependence on its quasi-identifiers, since data utility is measured in terms of the capability of a machine learning algorithm to exploit such dependence. Given the last condition, standardized data sets that do now show such statistical characteristic were excluded.

As in chapter 4, we used four data sets: three real and one synthetic. The first one is "UCI Adult" data set [71], standardized in the evaluation of microaggregation algorithms but, conveniently, also employed to assess machine learning algorithms. The other two real data sets are "Breast Cancer Wisconsin" data set [81] and "Heart disease" data set [82], both containing medical data extensively used to evaluate binary classification tasks. Finally, we created an elementary synthetic data set with three attributes mimicking two quasi-identifiers and a binary confidential attribute, in the same way as the toy example illustrated in Sec. 5.2.2.2. Table 4.1 includes greater details of these data sets.

5.3.3 Evaluation criteria

The privacy metric we use is k-anonymity since microaggregation algorithms aim at guaranteeing such criteria. Higher values of k imply larger anonymous microcells, so will offer more privacy to the subjects involved. Naturally, less utility is expected from data anonymized with higher values of k.

As described in Sec. 5.1, our evaluation scenario assumes that binary classification is the application domain of data. Thus, the corresponding utility metric here employed is classification accuracy, i.e., the accuracy of the classification model built from data, whether anonymized or not. Basically, accuracy quantifies the rate of correctly classified samples in a test set.

5.3.4 Algorithms and tools

In order to assess the effectiveness of our approach, we used some tools that we put together and describe next. We refer to the algorithms used for privacy protection and utility exploitation.

As expected, the privacy protection mechanism we use is MDAV, the de facto microaggregation algorithm. Besides its benefits in terms of time complexity, it has demonstrated to offer interesting results in terms of distortion and classification accuracy.

As in chapter 3, to measure the utility of microaggregated data, we use the machine learning algorithms that obtain the best performance, in terms of classification accuracy, from each of our data sets.These algorithms are boosting trees (Adult) and logistic linear regression (for the rest).

Finally, all the tests whose results are here presented were implemented with MATLAB 2018B. This includes loading and preprocessing data, the implementation of MDAV [27], as well as the evaluation of the resulting utility of perturbed data sets. This evaluation implies building machine learning models over data and applying such models over new data to measure classification accuracy and F-measure; all of this automatized using specific embedded functions for each algorithm. Greater detail is given in the next subsection.

5.3.5 Methodology

Next we describe the experimental methodology we used to assess the effectiveness of our (empirical) utility-preserving approach for k-anonymous microaggregation. Figure 5.7 synthesizes the flow of the evaluation procedure, while Fig. 5.2 illustrates the specific methodology implemented for our utility-preserving strategy.

In general, our evaluation builds on determining whether the utility of microaggregated data is preserved better when LDA is considered as part of the anonymization process. In this scenario, two main steps are carried out: anonymization through k-anonymous microaggregation, and utility extraction through the application of a machine learning algorithm over anonymized data. Figure 5.7 illustrates the flow of

these steps. To assess the benefits of our LDA-based approach, then, we measure the performance of such algorithm when LDA is used and when not.

Note that some preprocessing on the data set was necessary: numerization of some categorical values, split of data sets to get training and test sets, and zero-mean, unit-variance normalization of quasi-identifiers.

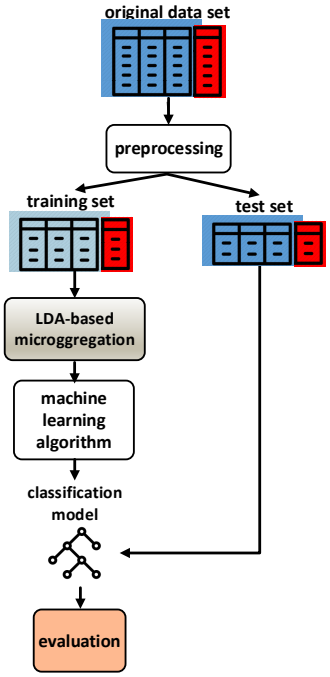


Figure 5.7: Main experimental methodology followed to implement our utility-preserving privacy protection approach on top of MDAV-based k-anonymous microaggregation.

Once normalized, the *microaggregation* algorithm is fed with the training set for data perturbation. We test progressively increasing values of k to then measure the utility degradation of data due to k-anonymous microaggregation. Figure 5.2 shows the specific process followed to obtain the anonymized data set from our approach proposed here. To start, the quasi-identifier values of the training set are transformed by projecting them through LDA and scaling them by a factor α. Then, the resulting transformed data is microaggregated using MDAV. Finally, the microcell assignment (a vector indicating the cell to which each record belongs) from the last step is applied on the original data to obtain the microaggregated data set, as depicted in Fig. 5.2.

With respect to the scaling, we made several tests varying the factor α from 1 (no scaling) to 64. Then, when presenting the results, we drew the corresponding maximum trace, i.e., the highest accuracy and F-measure values reached for each value of k.

After the anonymization phase, we implement the *utility extraction* phase. This is implemented following the same methodology described in Sec. 4.3.3. Namely, we build a classification model from microaggregated data. Specific functions implemented in MATLAB 2018b are used for training using 5-fold cross validation. Finally, each resulting model is evaluated over the test set; then accuracy and F-Measure are obtained.

5.3.6 Experimental results

In this section, we describe the results of assessing the performance of our LDA-based k-anonymous microaggregation in terms of utility preservation. To this end, we present here a series of figures where such performance was compared with that of MDAV. As previously explained, since we addressed the empirical utility of data, the metrics used were accuracy and F-measure of machine learned models when trained over data microaggregated, using an increasing value of k.

To start, we assessed our approach on *UCI Adult data set*. In this case, we do not use all the records but a sample of 10% of them, looking for reducing even more the data utility after microaggregation. To keep the structure of the original data set, we took a random sample that preserves the prevalence of the output (confidential) attribute. By reducing the baseline utility, we thought we could better visualize the effects of data utility preservation.

In Fig. 5.8 we depict the results of empirical utility extracted from the *UCI Adult* data set after applying k-anonymous microaggregation. Note that, as expected, the values of both metrics show a decreasing trend as the value of k increases: the impact of anonymization eventually renders data useless.

However, as depicted in Fig. 5.8, despite the inevitable degradation, the improvement, both in terms of accuracy and F-measure, is not only clear but significant in

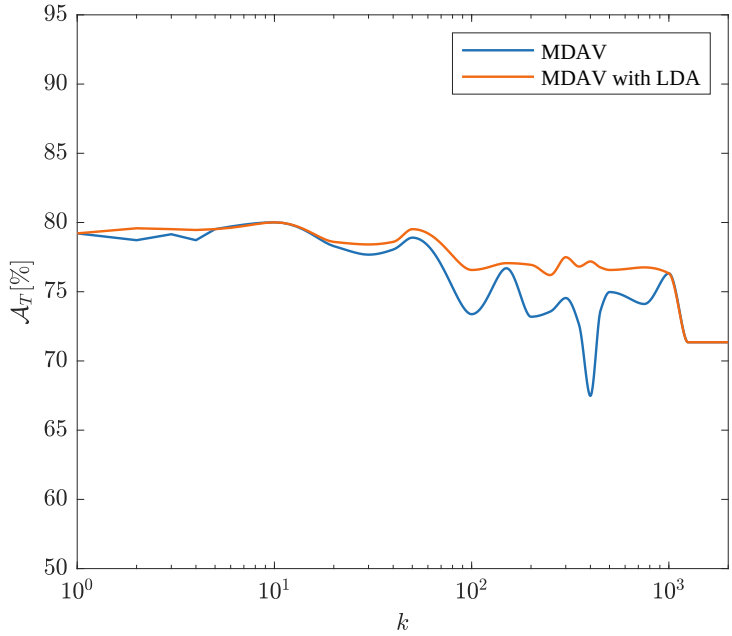
some cases when using MDAV with LDA. For example, when $k = 50$, the accuracy of the machine learning model goes from 81.8% to 83.9%, i.e., the error is reduced from 16.1% to 13.2%, which is a relative reduction of 18%. In the general, curves of utility look more stable when LDA and scaling are introduced, which implies that utility gets preserved even with relatively high values of k.

As described in Sec. 5.2, the results aforementioned are corroborated in experiments with three more data sets. When testing *Breast Cancer Wisconsin* data set, the benefits of MDAV with LDA are again evident. Also in this case, for some values of k, the reduction is significant. Figure 5.9 illustrates this in terms of accuracy and F-measure. Although the results of our method are better than those of "plain" MDAV, they do not seem as good as those obtained with the UCI Adult data set. There are several reasons that justify this behavior. Different data sets might naturally involve different macrotrends whose quality, in terms of utility, could also vary depending even on the amount of data. In addition, learning models built from the Breast Cancer Wisconsin data set show a maximum reachable accuracy of about 97% (i.e., very high), while it is about 80% for UCI Adult. Thus, we suspect that, when the room for improvement is greater, it is more likely that higher increases in accuracy can be reached.

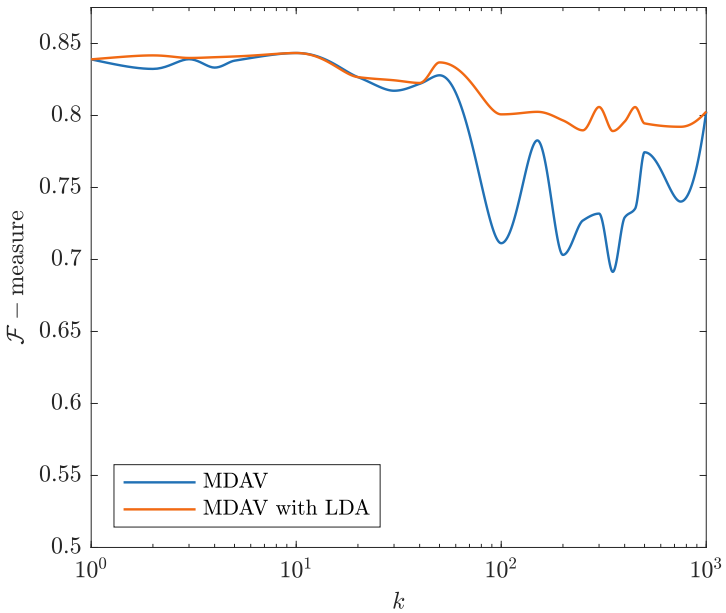
Figures 5.10 and 5.11 illustrate the results of assessing microaggregation algorithms over Heart Disease and synthetic data sets, respectively. For this two data sets, we confirm that MDAV with LDA achieves its goal of preserving utility of microaggregated data sets better than with MDAV. Once more we verify the benefits of our proposed mechanism but also the difficulty to do so given that MDAV already offer a privacy preserving approach.

Even though experimenting over real data sets might be enough for validation purposes, we use a synthetic data set with the aim to validate the results obtained over real data.

As a last note, classical distortion metrics based on MSE does not make sense in this study since the transformation based on LDA does not modify distances among points. In the case of scaling points are indeed separated in the direction of maximum

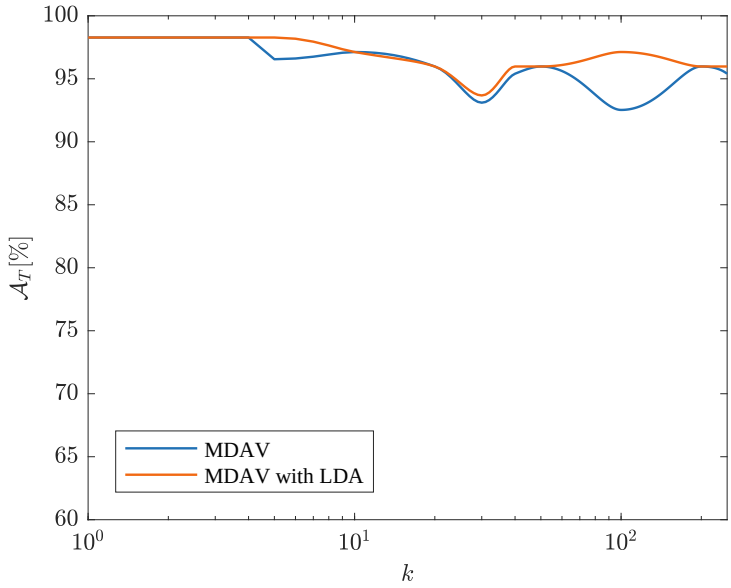


(a) Accuracy

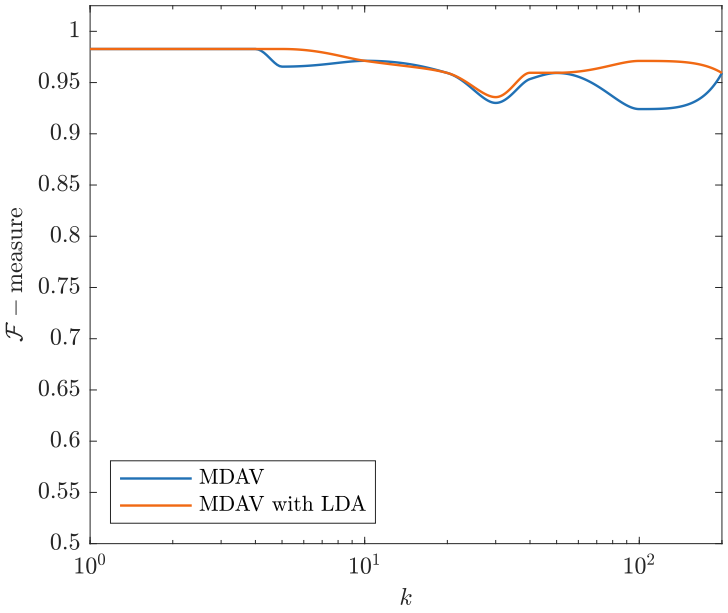


(b) F-measure

Figure 5.8: Empirical utility extracted from the UCI Adult dataset, microaggregated with original MDAV (blue) and with LDA-based MDAV (orange). Both in terms of accuracy and F-measure, LDA-based MDAV preserves better the utility of anonymized data.

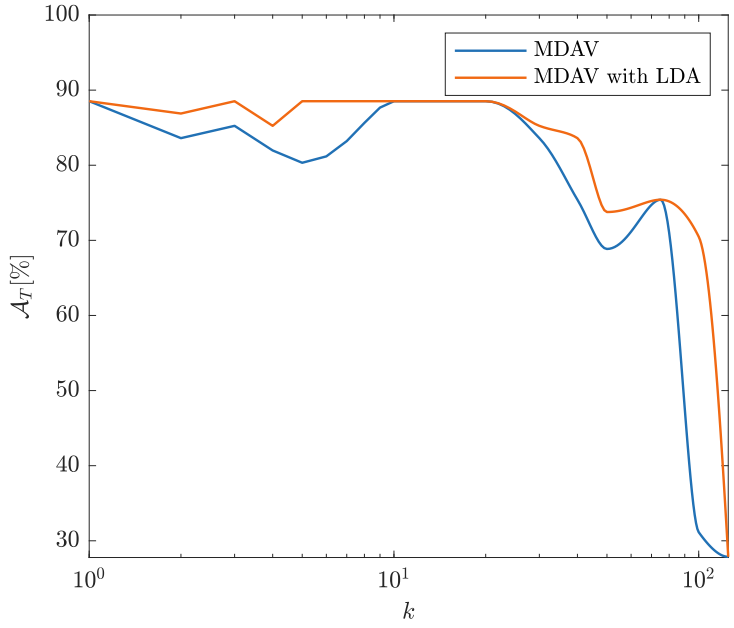


(a) Accuracy

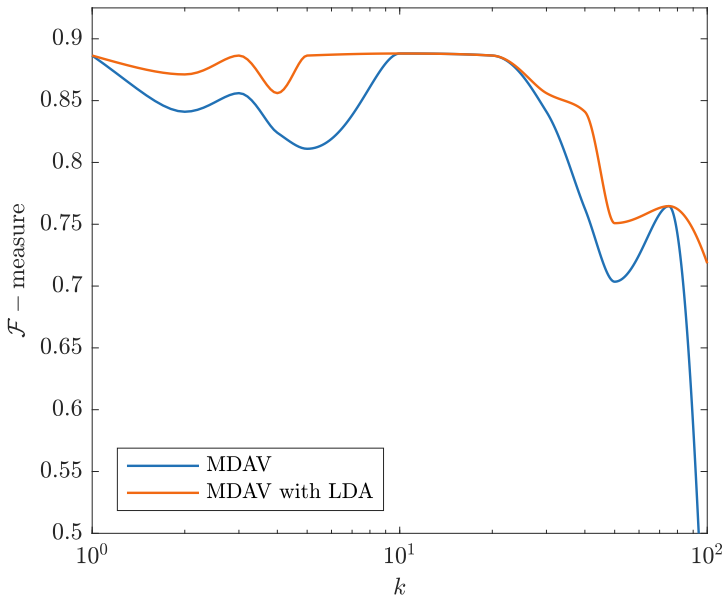


(b) F-measure

Figure 5.9: Empirical utility extracted from the Breast Cancer Wisconsin dataset, microaggregated with original MDAV (blue) and with LDA-based MDAV (orange). Both in terms of accuracy and F-measure, LDA-based MDAV seems to preserve better the utility of anonymized data.

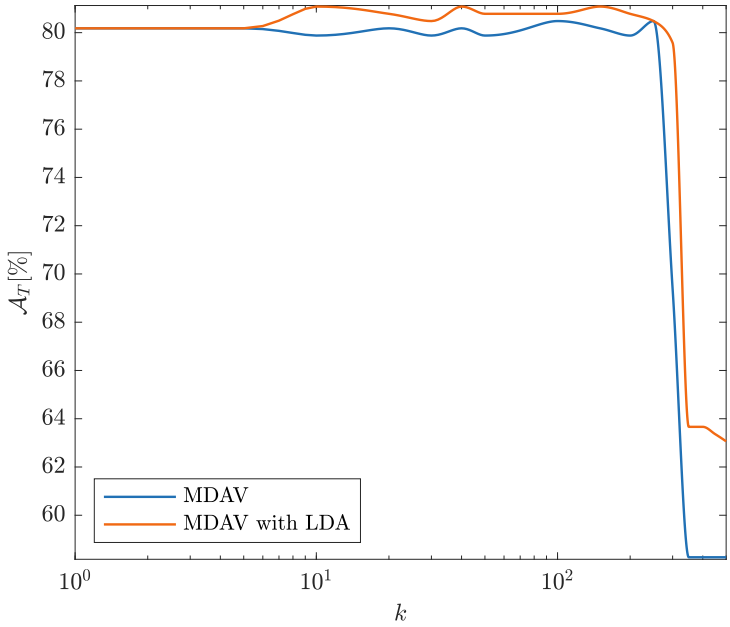


(a) Accuracy

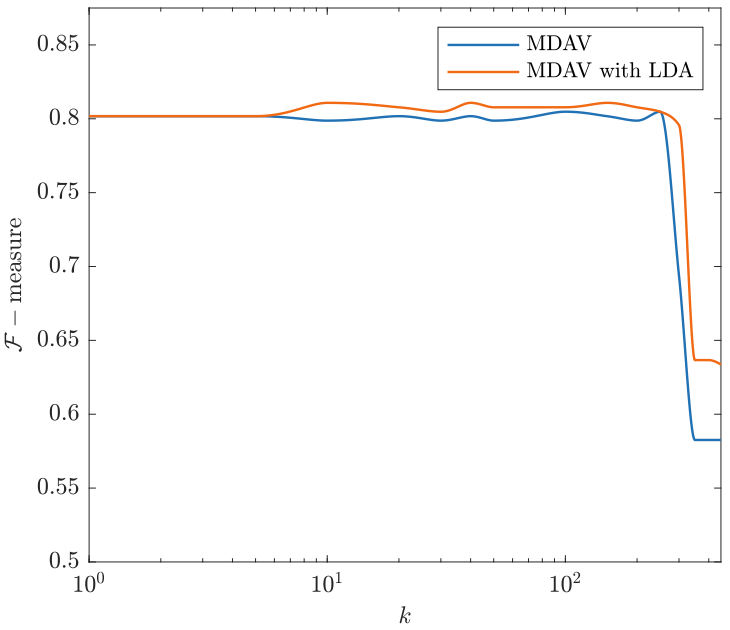


(b) F-measure

Figure 5.10: Empirical utility extracted from the Heart disease dataset, microaggregated with original MDAV (blue) and with LDA-based MDAV (orange). Both in terms of accuracy and F-measure, LDA-based MDAV preserves better the utility of anonymized data.



(a) Accuracy



(b) F-measure

Figure 5.11: Degradation of the empirical utility for the synthetic data set.

discrimination, so it is even possible that the resulting distortion in this context is even greater than 1 although the empirical results are improved.

The results obtained by our method are encouraging in that they show a consistent and, in some cases, significant preservation of data utility for microaggregated data. We would like to make some points below about this matter.

First, although MDAV with LDA behaves consistently better, in terms of data utility, than classical MDAV, the increase in utility may depend on the data set at hand, particularly on the information it can contribute to a learning model to improve its performance. Little could be done if machine learning algorithms cannot obtain practical accurate models from data even before applying privacy protection methods.

Second, in practice, our proposal does not imply any modification of the iterative process performed by MDAV. Given that our method modifies the representation of data before being microaggregated, the resulting computation complexity remains invariable. This detail is important because, in times when the world revolves around big data, processing time quickly becomes a bottleneck with respect to the potential applications of large-scale databases. Moreover, domains as critical as health, vehicular traffic, or network intrusion detection are currently using tons of data to help computational systems make real-time, and even life-or-death decisions. Due to such demanding requirements, privacy issues related to data processing are commonly overshadowed. Thus, from the perspective of privacy, we feel that any improvement in preserving data utility *without a price in (computing) efficiency* is not negligible and some works are currently being purposed in this direction [15, 17]. In fact, the next chapter presents a proposal addressing this particular issue.

Finally, we would like to point out that, since this strategy resorts to changing the representation of data –although not necessarily its semantics–, conventional, syntactic, utility metrics such as distortion (measured as MSE) would be hardly applicable in this context. This fact gradually characterized syntactic metrics as less meaningful in practical, real-word applications.

5.4 Conclusion

Our method successfully preserves the empirical utility of data when microaggregated through MDAV. This is done by transforming quasi-identifier values in such a way that, after microaggregated, the resulting k-anonymous cells enable the construction of a more effective machine learning classifier.

Graphically illustrated, our proposal gets "thinner" microcells in the direction of maximum discrimination, obtaining a distribution of cells and reconstruction that better preserve the statistical properties on microaggregated data. Linear Discriminant Analysis and scaling were applied to find this direction and to weight the inherent distortion by an empirical parameter α.

In terms of accuracy and F-measure of resulting machine learning models, LDA applied to MDAV outperforms the classical implementation of MDAV. Although MDAV is by default benign when affecting the statistics within data, our approach successfully preserves the utility of data after microaggregation. This is confirmed trough systematic experimentation over synthetic and real data sets.

Conveniently, this benefit comes at no cost, e.g., in terms of running time, as other utility preserving proposals do ([90]). Thus, our approach is both functionally and computationally effective. Furthermore, ours is the first application of LDA to the domain of statistical disclosure control, applying a substantial and non trivial modification of any microaggregation algorithm, although here is assessed with MDAV.

Chapter 6

Computational improvements for microaggregating large-scale data sets

6.1 Introduction

As we discussed in Sec. 2.1, big data is bringing new, unprecedented business opportunities to companies around the world. Currently, it is possible to collect and process vast amounts of information from which more, new, better and varied customer knowledge is mined. As a result, better decisions can be made in sectors like health care, banking, marketing and transportation [91–93].

Despite these benefits, within such an abundance of data, it is common to find personal sensitive information, which poses serious privacy risks. First, in the name of this data revolution, information is more prone to be openly published or shared with untrusted third parties. Also, although identifiers are typically suppressed, other demographic attributes, when combined, can be used to reidentify individuals [8, 9, 23]. Thus, sensitive attributes might be easily linked to the subjects to whom the disclosed information corresponds, which might lead to privacy risks [10]. This scenario was described in Sec. 2.1.1.

Statistical disclosure control aims at addressing these privacy issues in the special case of microdata files. As stated in Sec. 2.2.1, the goal of SDC is to reduce the risk of sensitive data disclosure while preserving the internal macro trends of data, i.e., its utility. Along this work, we have concentrated on k-anonymous microaggregation and particularly on MDAV as a high-utility privacy protection algorithm.

By carefully aggregating microdata attributes, a minimum level of distortion must be applied to data. Unfortunately, current microaggregation algorithms entail a very high computational cost when anonymizing big data [90]. Thus, since utility extraction from big data is a priority, and already time consuming, privacy protection might be easily neglected. That is why some works are starting to propose strategies to reduce the running time of privacy enhancing mechanisms while preserving the utility of data.

In this chapter, we propose an avenues for improving the performance of MDAV, in terms of computational time. The fundamental aim of such improvement is to facilitate the implementation of privacy protection in big data.

The proposal allows obtaining remarkable reductions in running time by diminishing the number of operations necessary to aggregate data with MDAV, all of this without yielding any additional loss in data utility.

This effort is interesting since the reduction of the computational cost of privacy protection algorithms may encourage its implementation, especially when computation usually entails important economic costs for companies exploiting big data.

Furthermore, due to the massification of Internet access, the vast amounts of data containing personal information may grow and change very dynamically, commonly feeding online services. Then, microaggregation, in this context, is likely to be implemented as an ongoing process, running as fast as possible, rather than as a static one-time job. In fact, if data is sufficiently vast, microaggregating it once could be unfeasible in practice for some, e.g., real-time, applications due to the quadratic complexity of MDAV, so optimizing its running time in the big data era seems mandatory.

Interestingly, in this context, increasing the efficiency of privacy protection mechanisms (e.g., reducing their runtime) could become a powerful value generator for companies implementing privacy.

The work presented in this chapter was published in [15].

Chapter outline

Section 6.2 describes the adaptations applied to MDAV in order to reduce its running time while leaving untouched the resulting utility of data. Section 6.3 presents the results of experimental evaluation. Finally, conclusions are drawn in Sec. 6.4.

6.2 Strategies for speeding up MDAV

Our first effort towards speeding up k-anonymous microaggregation lies in analyzing the microaggregation algorithm, i.e., in finding the components subject to be accelerated, and devise the mechanisms and algebraic properties that could implement such improvements.

As described in Sec 2.2.4, MDAV creates partitions or microcells from a data set by aggregating neighboring records. Since MDAV operates with numerical attributes, each record is seen as an m-dimensional point ($x_s \in \mathbb{R}^m$) in the Euclidean space, being m the number of attributes of the data set. Note that the microaggregation process iteratively extracts pairs of cells while $2k$ points or more in the data set remain to be assigned. First, a centroid C is calculated as the average of the remaining points. Then, from C, two points P and Q are found from the data set, which serve as references to obtain the neighboring points of each of the two new microcells: one formed by P with its $k-1$ nearest points and another by Q with its $k-1$ nearest points. P is obtained as the furthest point from C (the maximum distance to average vector) and Q as the furthest point from P.

The previous description reveals a set of mathematical operations over the records of the data set. These operations mainly involve centroids calculation, distances calculation, and sorting. Since these operations are used repetitively and executed over a vast number of tuples, there is an interesting chance for improvement in the overall performance of MDAV. Next, we describe in detail the improvements we propose on these operations; Table 6.1 summarizes the MDAV tasks improved in this work, the

respective strategy followed, and gives a brief description of each one. We call the new version of MDAV as Fast MDAV or F-MDAV. In Sec. 6.3, we show the benefits of these strategies through extensive experimentation.

Table 6.1: Summary of computational improvements for MDAV

MDAV task	**Improvement strategy**	**Description**
Distance calculation	Algebraic modification, precomputing	We propose using a property of the inner product to calculate and compare distances so that less operations are needed for microaggregation. Being these algebraic operations, their implementation is usually even optimized in multiple computing libraries.
Sorting	Use partial sorting (or selection)	Since MDAV does not strictly require total order when finding the $k-1$ shortest points to a reference, a more relaxed version called partial sorting can help MDAV save computing resources.
Assignment of microcell	Reuse of distance calculations	Given that much of the running time of MDAV is devoted to calculating distances among the tuples of a data set and a reference point (to obtain its shortest and furthest points), such distances could be precomputed and then reused to prevent redundant operations.
Centroid calculation	Precomputation and reuse of calculations	We propose modifying the calculation of centroids in MDAV so that redundant addition operations are eliminated.
All calculations in general	Use less precision in the calculations	Since MDAV operations may not require much precision to build microcells, we propose changing the numerical representation to single precision so that less bits be processed, thus implying a reduction in MDAV's running time.

6.2.1 Algebraic improvement

From line 3 of Algorithm 1, we can devise that much of the MDAV runtime is intended to calculate distances in three moments: when finding the furthest point P from centroid C, when finding the furthest point Q from P, and when obtaining the $k-1$ nearest points from P and from Q to build two microcells.

The fact that distances between the (in some cases the same) points of a data set are continuously calculated turns each iteration very redundant. The computation

complexity of MDAV, then, derives from such redundancy, which we tackle through this improvement.

Since MDAV considers each record of the data set as an m-dimensional point in the Euclidean space, a distance D_j between a reference point x_0 (which is C, P and Q, depending on the moment) and a collection of points x_j for $j \in \{1, \ldots, n\}$ is calculated as a quadratic Euclidean distance, i.e.,

$$D_j = \|x_j - x_0\|^2 .$$

Then, to get D_j, for each m-dimensional point x_j, an element by element subtraction (m operations) and a square norm ($2m - 1$ operations) must be calculated. That is, a total of $3m - 1$ operations for each point of the data set.

To reduce the resulting runtime, we consider finding an analogous expression to calculate D_j such that less operations are performed. In this case, we harness the polarization identity of the inner product to put the expression of D_j in terms of the inner product of x_j and x_0. So we expand the last expression such that

$$\|x_j - x_0\|^2 = \|x_j\|^2 + \|x_0\|^2 - 2\langle x_j, x_0\rangle .$$

If both sides of the last equation are subtracted $\|x_0\|^2$ and multiplied by $1/2$, we obtain

$$\frac{1}{2}\left(\|x_j - x_0\|^2 - \|x_0\|^2\right) = \frac{1}{2}\|x_j\|^2 - \langle x_j, x_0\rangle .$$

Although the expressions on both sides of the equation no longer represent the real value of D_j, they are a metric still useful to compare distances since they were summed and multiplied by a constant. Thus, when the calculation of distances in MDAV is used to determine the furthest point from a fixed point x_0, we can safely use the right part of the last expression for comparison issues.

Conveniently, for each compared point x_s, the value of $\frac{1}{2}\|x_j\|^2$ can be precomputed once before MDAV is initiated, out of the redundant process, and avoiding significant recalculation in every iteration of MDAV. Thus, in this case, the distance comparison is reduced to the calculation of the inner product $\langle x_j, x_0\rangle$, which consists in an element by element multiplication, i.e., m operations, and a sum of the resulting m terms,

i.e., $m-1$ operations, for each point x_j. For this representation of distances, we have a grand total of $2m-1$ operations.

By analytically operating on an expression, we get less operations than the original expression of distance D_s. More precisely, the number of operations is reduced from $3m-1$ to $2m-1$ for each distance calculation where m is the number of quasi-identifiers of each record in the data set.

Not only the number of operations is reduced, but they are algebraic in nature, and that is something for which much of the current code is optimized (e.g., in Matlab or the C standard library). Consequently, it is reasonable that Matlab uses vectorized code or more efficient CPU instructions, that is, advanced vector instructions (AVX) through the Intel math kernel library (MKL). In fact, there are instructions that compute an accumulated sum and a product, designed for the efficient computation of vector and matrix products, called multiply-accumulate operations and fused multiply-add (FMA), which are included in certain Intel processors (e.g., Haskell). Interestingly, if FMA were implemented, our proposal would lead to reduce the number of operations here analyzed to m.

In Fig. 6.1, we summarize the analysis carried out in the last paragraphs. Finally, this improvement lends itself to the implementation of MDAV in graphics processing units (GPU).

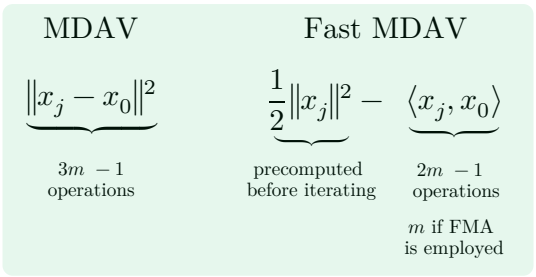


Figure 6.1: Representation of the distance calculation performed for each m-dimensional point x_j when k-anonymous microcells are built. We can see that our approach Fast MDAV is able to reduce the number of operations from $3m-1$ to $2m-1$ for each of these n records. Furthermore, since the inner product $\langle x_j, x_0\rangle$ is subject to optimization if FMA is used, the number of operations could be even reduced to m.

6.2.2 Distance reuse

The high utility offered by MDAV comes from smartly grouping the closest points into microcells. As already pointed out, this process certainly involves several steps where distances need to be calculated. In this section we concentrate on steps 3 and 4 of Algorithm 1 with the aim of reducing the runtime of MDAV.

We can see that, when aggregating a microcell, given a reference point P, two distance-calculation operations are performed. First, to find the $k-1$ nearest points to P necessary to form a microcell, the distances from all points to P have to be calculated previously. Afterwards, the furthest point Q from P is needed to serve as a reference for building a new microcell; this also entails calculating distances to P.

Evidently, for both steps, the calculations of distances to P can be calculated once and reused. Here, our proposal is using the distances from every point x_s to P both to find the points nearest to P and to find the furthest point from P.

6.2.3 Partial sorting

Sorting is another time-consuming operation within the original version of MDAV. It is recurrently implemented, e.g., to find the points closest to a given centroid in order to establish the most appropriate members for each microaggregated cell, as posed in line 4 of Algorithm 1.

Finding the points closest to a centroid C implies sorting all the distances from those points to C upwards (total sorting) and then getting the first ones, i.e., the shortest ones. Interestingly, in the case of MDAV, such total sorting is not necessary because only the $k-1$ nearest points are required. In fact, their corresponding distances to the centroid do not even need to be ordered. Finding the k smallest elements implies a more relaxed sorting approach called *partial sorting* or partial selection.

In computer science, total sorting is extensively implemented through the quicksort algorithm [94] while partial sorting through the quickselect algorithm [95]. The computational complexity of the sorting task when using quickselect may be significantly reduced, since it finds the kth smallest number in an unordered list, which

does not require total order. This relaxation makes quickselect's problem a much easier one.

As expected, our proposal is to resort to the use of an implementation of partial sorting (e.g., quickselect) in MDAV when the phase of microcell assignment is performed.

Quickselect is a selection algorithm by which a single element, the kth smallest, is found from a list. Its approach starts by randomly selecting a pivot element that will partition the elements in two; the elements smaller than the pivot on the left and the larger ones on the right. Then, this same approach is recursively implemented only into the side where the element being searched lies up until a single element is obtained. On the other hand, total sorting implemented through quicksort applies the aforementioned approach on both branches, which significantly increases the computational cost. Figure 6.2 offers a brief scheme of the extensive reduction of computation complexity when using quickselect instead of quicksort.

By using quickselect, the average complexity of the operations in question is reduced from $\mathcal{O}(n \log n)$ to $\mathcal{O}(n)$ on the average case. This is very convenient for a context such as big data where millions of records may have to be processed.

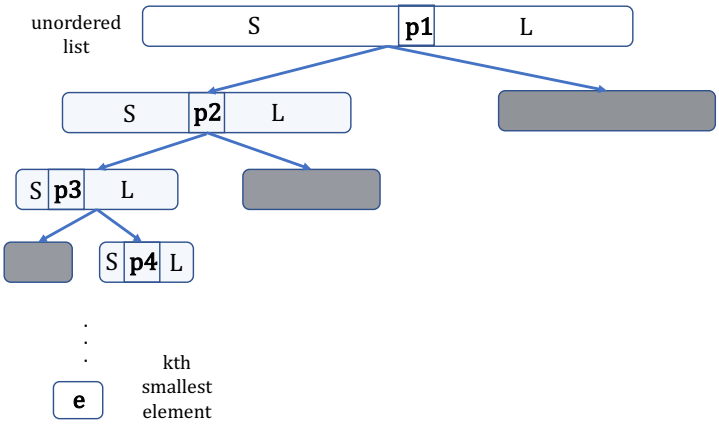


Figure 6.2: Brief depiction of the recursive steps carried out for the quickselect algorithm. To find the kth element from an unordered list, quickselect starts by randomly choosing a pivot that partitions the list into two parts: the left one with the elements smaller than the pivot and the right one with the elements larger than the pivot. This process is is applied again only on the part where the searched element lies. Finally, all this operation is recursively executed up until the kth smallest element is found. The gray blocks represent the part of the data where the algorithm is not executed (unlike quicksort), thus significantly reducing redundancy.

6.2.4 Centroid by subtraction

MDAV builds on another critical operation, centroid calculation (line 2 of Algorithm 1). At each iteration of MDAV, a centroid C is obtained to then serve as a reference in the construction of two microcells (line 3 of Algorithm 1). Every time a couple of microcells are created, their corresponding microaggregated tuples are removed from a stack (line 5 of Algorithm 1) and a new centroid is calculated using the remaining tuples, to build other two microcells. A centroid is calculated, at the beginning of every iteration, by averaging the remaining tuples, which simply consists in adding up all these tuples and dividing by the number of tuples. However, in this iterative process, several tuples of the data set get added again and again multiple times, which definitely results in redundant work and thus running time that can be saved.

In order to accelerate the execution of MDAV, we can modify the calculation of centroids so that the redundant operations of sum are eliminated. To this end, we propose to calculate centroids "by subtraction". Accordingly, we first calculate the sum $S = \sum_{j=1}^{n} x_j$ of all the tuples of the data set $(x_1, \ldots, x_n)$. Moreover, we keep track of the tuples being aggregated by adding them up in S' as soon as they are assigned a microcell. Conveniently, subtracting S' from S has the same effect as obtaining the sum of all the tuples not already aggregated for each iteration so said subtraction can be used to calculate centroids as $C_s = \frac{S-S'}{n_s}$. The benefit evidently lays in that unnecessary adding operations are not done. Finally, note that initially precomputing the sum of all the tuples of the data set does not represent significant complexity since it is only done once.

6.2.5 Single precision

In computer hardware, numerical data is represented with a number of bits that define the precision of calculations. These options include single precision, where 32 bits are used, and double precision, which uses 64 bits.

Due to the higher computing capabilities of modern hardware, most of the algorithms are implemented using double precision as a standard. However, if single precision could be implemented, we could speedup the execution of such algorithms

since less bits would have to be processed. Consequently, given that the standard version of MDAV performs a series of mathematical operations over numerical values, we propose to use single precision for them in order to accelerate the microaggregation process.

This is the only modification that might imply a change in the results of the calculations performed by MDAV. Notwithstanding, since this algorithm might not require extremely precise operations (in terms of the number of decimal points considered), we expect no significant changes in the structure of microcells obtained with respect to the original version of MDAV, but a faster k-anonymous microaggregation.

6.2.6 Prepartitioning

Prepartitioning, or dividing data into multiple chunks, is a known mechanism to enable, e.g., the distribution, among various instances, of the computing load necessary to process such data. Since the execution time t of k-anonymous microaggregation is super linear ($t = \frac{n^2}{k}$), thus super additive, this "divide and conquer" strategy is appropriate for reducing such execution time.

As explained in [17], the strategy consists in two steps: first, dividing the data set in big macrocells of size K (macroaggregation) through MDAV; and, second, applying MDAV to each of the resulting macrocells to obtain microcells of size k that satisfy k-anonymity. The execution time of microaggregation, after applying this strategy, is subject to be optimized based on the size K of macrocells.

The speedup reached by prepartitioning can be improved if the strategy is applied recursively. The resulting execution time may have a quasilinear form, but at a price to pay in terms of data distortion.

Although this approach is out of the scope of this chapter, we described it here for the sake of illustration of the potential avenues of improvement for the acceleration of microaggegation.

6.3 Experimental evaluation

In this section, we evaluate the efficiency of the proposed computational enhancements to MDAV. The objective of evaluation through experimentation of our approach is mainly to determine its impact on microdata. As mentioned in previous sections, such impact can be measured in terms of the algorithm's speedup and the resulting data distortion (although in this particular case it is unlikely to occur) spawned by F-MDAV. Another objective is finding out whether such effect is independent of the data set employed.

We conducted this evaluation across two dimensions: *speedup* and *performance*. Speedup was measured as time gain, while performance was measured as the additional distortion incurred by the adapted versions of MDAV. However, since most of the proposed modifications do not change the internal operations of the algorithm, there was no additional distortion in the data and thus we mainly focused on measuring speedup. Below, we describe the experimental setup and our results from systematic tests over a variety of data sets. Such results are depicted for each enhancement and data set in Figs. 6.3, 6.4, and 6.5, while the overall speedup is illustrated in 6.8.

The evaluation of the computational performance of our methods was conducted with three *standardized* data sets. These real data sets included "Large Census", "Quant Forest" and "USA House", which were previously used in [17, 96]. The "Large Census" data set has 149,642 records and includes 13 numerical attributes; "Quant Forest" has 581,012 records, from which we use a random sample of 150,000 records, and 10 numerical attributes. The "USA House" data set has 5,967,303 records and 13 numerical attributes. We used the "Large Census" data set since it is extensively employed in SDC, whereas "Quant Forest" and "USA House" data sets were used to validate the results obtained in "Large Census". For our study, all attributes were considered to be quasi-identifiers.

The experiments described in this section were run in an Intel Core i7 CPU 3.4 GHz with 32 GB RAM. The microaggregation algorithm MDAV, its adapted versions,

and the measurement tests were implemented entirely in Matlab R2017b, where, for the sake of fairness, we disabled any form of parallelization.

In general, all versions of MDAV were parameterized with $k = 10$, which implies a reasonable level of privacy without incurring a significant distortion of data. Moreover, before microaggregation was applied, we followed the common practice of normalizing each column of the data set to have zero mean and unit variance.

To find the speedup obtained by our improvements, we measured the running time of MDAV before [27] and after implementing our refinements. Our reference MDAV is once again the algorithm specified in Algorithm 1. We refer to it as *traditional MDAV*. Furthermore, the modifications proposed in this work were applied to MDAV *individually*, with the aim of measuring their separate contribution to the speed of the microaggregation algorithm. Also, to show the combined effect of the five improvements, we implemented them in a version of the algorithm we called Fast MDAV.

Our experiments relied on a speedup to show how faster MDAV may become due to the proposed computational improvements. Let t_0 be the running time of traditional MDAV and t the running time of any improved version of MDAV (including fast MDAV). Essentially, the speedup factor $s = \frac{t_0}{t}$ tells us how fast this version is with respect to traditional MDAV. For instance, consider $t_0 = 15$; if, after adapting the algorithm, its running time were reduced to $t = 5$, we would have gotten a speedup factor of $s = 3\times$, i.e., an MDAV that is 3 times faster.

Regarding our experimental *methodology*, we have a few final remarks. First, we assessed MDAV over a varying number of records n with the aim of verifying the impact of our methods when the size of the data is increased. Thus, from each data set, we extracted portions of data of varied sizes (different values of n). For each value of n, the running time we measured was the averaged time that it took MDAV to microaggregate n records. To that end, for every data set, we systematically obtained 3 random samples of n records each and then averaged the corresponding running times of MDAV. The running times for every improvement were registered and then compared with the time of traditional MDAV through the aforementioned speedup factor s.

The results of our computational strategies are presented in 4 bar charts; the first 3 illustrate, for each tested data set, the speedup factor obtained by every method. Although we experimented with several lengths, for the sake of visibility, the results are shown only for 3 representative values of n (10,000, 70,0000 and 150,000). In the same line, the last bar chart exhibited the speedup factor reached by the fast MDAV, i.e., when all the improvements were consolidated within the same MDAV implementation. Although in essence the modifications we propose to MDAV do not imply a change in the numerical results of its internal operations, we verified whether or not each improvement leads to a variation of the built microcells or an increased distortion with respect to traditional MDAV.

In the following subsections, we depict and explain the results of our experiments.

Algebraic improvement

As already explained in Sec. 6.2.1, this method reduces the number of operations needed to calculate distances by taking advantage of a property of the inner product. Remarkably, numerical libraries are usually optimized for these algebraic operations.

Consequently, the results of our experiments show a significant speedup of MDAV that could reach a factor of 1.54×. This is depicted in Figs. 6.3, 6.4, and 6.5, for the three data sets we use, over which a homogeneous computational improvement is revealed.

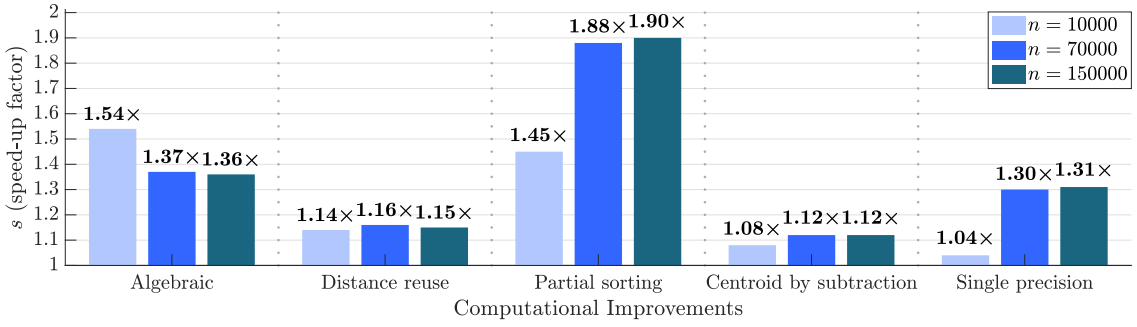


Figure 6.3: Speedup factor (s) of each of the five proposed improvements, i.e., when applied individually on the Large Census data set.

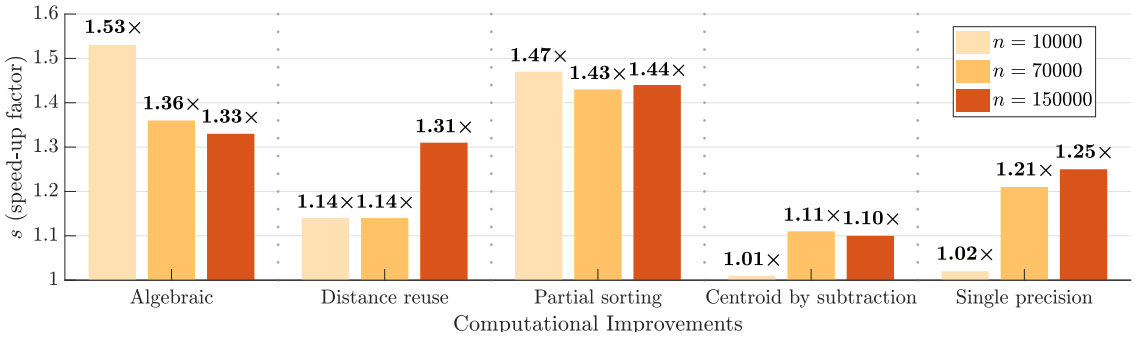


Figure 6.4: Speedup factor (s) of each of the five proposed improvements, i.e., when applied individually on the Quant Forest data set.

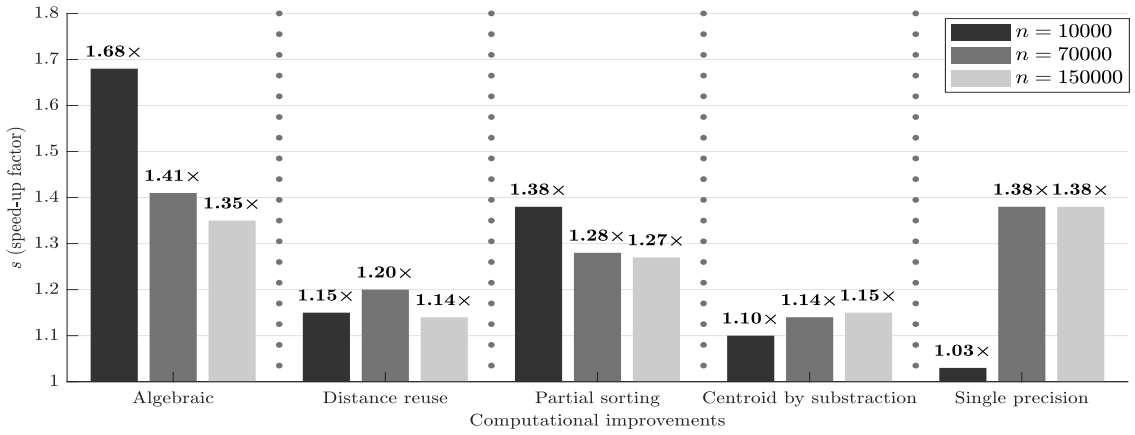


Figure 6.5: Speedup factor (s) of each of the five proposed improvements, i.e., when applied individually on the USA House data set.

Along with the partial sorting improvement, this algebraic strategy presents the best performance in terms of running time. In addition, the results of the cell assignment function $c(j)$ and the resulting distortion of this new version of MDAV remain unchanged with respect to traditional MDAV.

Distance reuse

Given the recurrence of distance calculations in MDAV, its running time can be reduced by precomputing some of such distances as theoretically explained in Sec. 6.2.2. Accordingly, after testing this improvement, when using the experimental setup described in this section, we observe a speedup factor between 1.14 and 1.31×. This execution performance is illustrated in Figs. 6.3, 6.4, 6.5 for the three data sets previously mentioned.

Once again, implementing this distance reuse does not reflect any variation neither in the structure of the microcells obtained nor in the distortion imposed to the data sets.

Partial sorting

As described in Sec. 6.2.3, the impact of sorting operations on the running time of MDAV could be reduced by using partial sorting, given its lower complexity with respect to total sorting. To illustrate the potential improvements due to partial sorting, we first performed an experiment in Matlab comparing two applications of both problems. Although not explicitly stated in the documentation, it is reasonable to assume that the functions `sort` and `mink` of Matlab R2017b implement variants of quicksort (total sorting) and quickselect (partial sorting) algorithms, respectively. In fact, this experiment confirms that these functions follow the behavior of quicksort and quickselect in terms of computational complexity.

For this initial test, we did not only measure how long `sort` and `mink` take to execute over a list of random generated numerical values, but we tried to mimic the sorting operations performed by MDAV through a few simple steps: sorting real-valued numbers and allocating and returning the indices of sorted values. It turned

that returning indices noticeably slowed down the execution of sorting functions for short lengths. Moreover, when necessary, we preallocated outputs to exclude the time for memory allocation from our measurements. Particularly for the function sort, we also considered the time of trimming off the shortest values from the list.

The test involved more than 300 experiments, each of which consisted in measuring the time it takes to obtain the k shortest values from a randomly generated list of length n. This process, along with the considerations of the last paragraph, emulated the role of sorting within MDAV. To evaluate the benefits of partial sorting over total sorting, we obtained the running times when using `mink` and `sort` functions to find the shortest values; we tested their performance for several values of n, which ranged from 10 to 10 million, and for $k = \{5, 10, 20, 50\}$.

For the sake of reliability, we measured the running times for several repetitions in each experiment. Then, we computed the mean, as an estimate of average performance. Also, while the length of the data used was the same for every experiment, the values of the list were randomly sampled for every repetition. After a one-hour experiment, we found that our measurements were extremely reliable: the worst coefficient of variation, calculated as the standard deviation divided by the mean, was observed to be 1.63%.

Figure 6.6 shows how our experiments took longer (the running time t gets higher) as the length n of the list increases. We used double logarithmic scales since we had very wide ranges of values for n and t, and thus extremely low and high values may appear. We can see that the running time for `mink` grows lineally with n, regardless of the value of k used, in line with the complexity $\mathcal{O}(n)$ of the quickselect algorithm; this is important evidence that `mink` would be implementing a variant of this algorithm. For `sort`, the corresponding running times are certainly higher. However, the magnitude of the difference with respect to `mink` is not very clear. For that reason, we depicted in Fig. 6.7 the running time per element $\frac{t}{n}$ for every experiment. Using a semilogarithmic scale, this figure does show that `mink` (partial sorting) is much more efficient than sort (total sorting) since while for `mink` the running time per element was constant, said time grows logarithmically with n for

sort. As a reference, we also plotted the running time of the function *min* that retrieved only the lowest value from each list.

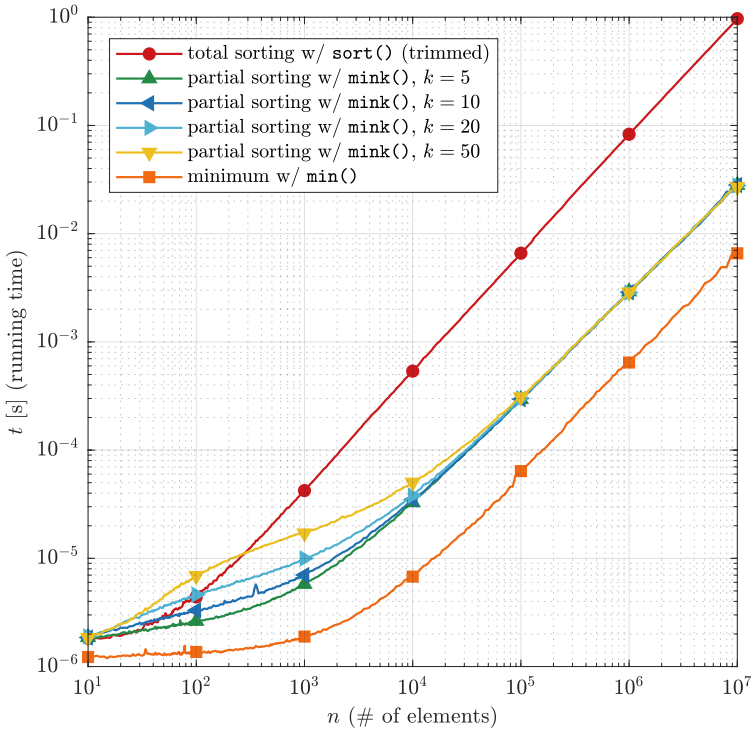


Figure 6.6: Running time of different variants of sorting implemented in Matlab R2017b. Extensive testing was performed for several values of n (number of elements in the sorted list) and k (here representing the number of elements to be selected and sorted from the list, when partial sorting was tested). For the sake of clarity, double logarithmic scales were used.

Finally, to estimate the speedup of microaggregation due partial sorting, we ran the experiments according to the setup proposed at the beginning of Sec. 6.3, but using a version of MDAV that relied on the function `mink` for microcell assignment. We then compared the resulting running times with those of traditional MDAV that used `sort` by default.

As expected, partial sorting introduced interesting computational improvements. In fact, a speedup factor of almost 2× was reached for the Large Census data set when $n = 150,000$, as depicted in Fig. 6.2. However, the degree of improvement was not uniform for the three data sets, as it is shown in Fig. 6.4 and Fig. 6.5 for the partial sorting method where the maximum speedup factor did not attain 1.5×. This

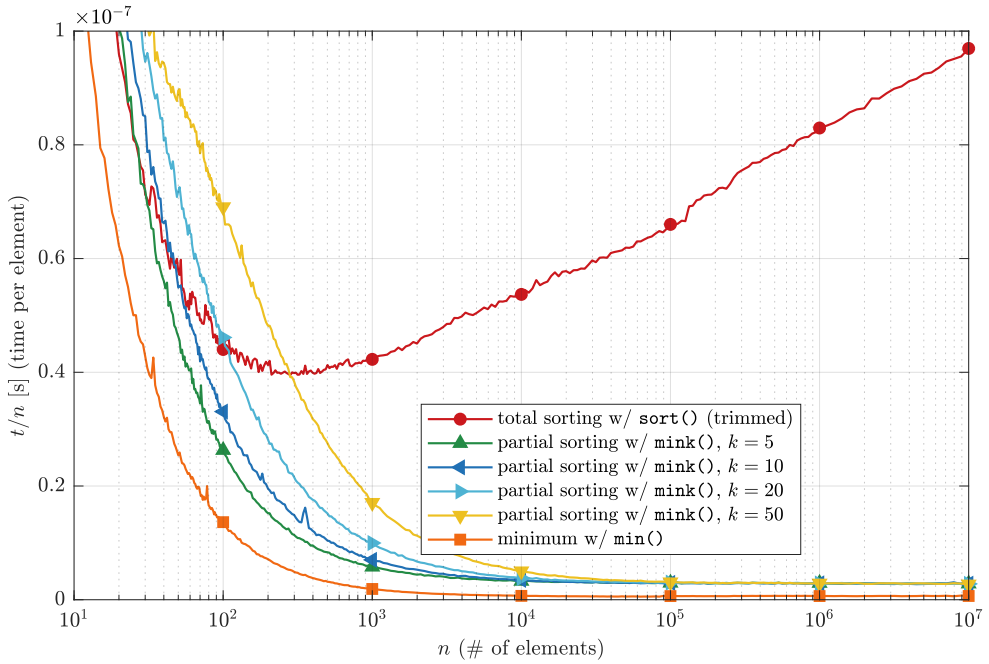


Figure 6.7: Running time of different variants of sorting implemented in Matlab R2017b. Here, we depict the time taken per element $\frac{t}{n}$ (t is the time taken to sort a list of n elements) to have a clearer illustration of the remarkable performance of partial sorting implementations compared to those of total sorting. Again k represents the number of elements to be selected and sorted from the list in the case of partial sorting. Briefly, the running time of partial sorting remained constant for large values of n, while for total sorting time grew logarithmically. Also, for clarity, a semilogarithmic scale was used.

seems reasonable since the complexity of the variants of sorting may depend on the intrinsic structure of the data.

Centroid by subtraction

It is clear from Sec. 6.2.4 that the operations for the calculation of centroids in MDAV are subject to redundancy since the tuples of a microdata set have to be added recurrently to find a representative mean. Our centroid by subtraction strategy, which uses precomputing and subtraction, obtained a speedup factor of up to 1.15× as shown in Figs. 6.3, 6.4 and 6.5. Although performance gain was more moderate for MDAV, it was still important, considering that its implementation does not represent any additional cost in terms of distortion.

Single precision

Being MDAV an algorithm whose calculations may not require extreme precision, our last method is based on using single precision for the corresponding mathematical operations. Strikingly, this modification allowed a computational improvement that was even better than that of centroid by subtraction, i.e., a speedup factor of up to 1.35×, as depicted in Figs. 6.3, 6.4, 6.5, for our three data sets.

As anticipated in Sec. 6.2.4, the results from using this strategy showed a slight variation in the structure of the microcells built by MDAV. However, the resulting distortion remained unchanged.

Fast MDAV

Our last series of experiments analyzed the case when all proposed modifications were combined in a single version of the microaggregation algorithm, Fast MDAV. The tests showed remarkable results, as reflected in Fig. 6.8. We confirmed that fast MDAV was up to 4 times faster than the original version and that, as expected, no additional distortion was introduced in the three microaggregated data sets.

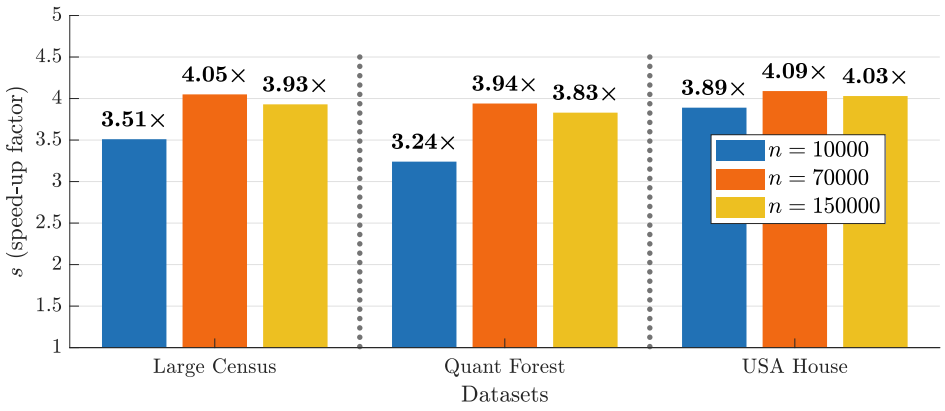


Figure 6.8: Overall speedup factor s of our fast MDAV obtained over the three data sets. The five strategies are consolidated in a single version and it is tested for several values of n. Due to space considerations, only the results of tests for $n = 10\,000, 70\,000, 150\,000$ are depicted in this figure.

Information loss with F-MDAV

As mentioned when describing each computational improvement, the rationale behind the work presented in this subsection was simplifying redundant operations when implementing MDAV. Such simplification was based on finding alternate algebraic expressions, reusing and precomputing (repetitive) calculations, adapting a more relaxed sorting strategy, and even using less precise calculations.

When single precision is used for calculations, there was a risk of obtaining a structure of microcells different from that of MDAV. However, microaggregation did not require extremely precise calculations since those were used only for comparing distances between points. Namely, in practice, our proposal did not imply any modification of the resulting k-anonymous groups built by MDAV, so there might not be a price in distortion. As a matter of fact, we verified that microcell allocation remains invariable after F-MDAV is implemented. Consequently, our strategies did not incur in additional distortion or information loss with respect to that provoked by original MDAV.

6.4 Conclusions

In this chapter we have addressed the problem of computational complexity of k-anonymous microaggregation for large data sets with a substantial amount of numerical records. This effort striven to obtain a more usable privacy mechanism in contexts brought by the current big data era.

We proposed an approach with five algorithmic and algebraic strategies that reduced the running time of MDAV by a factor of 4, without affecting the resulting utility of data.

This approach mainly spanned the reuse of calculations, precomputing, algebraic modifications, and a relaxed approach of sorting; all of them were implemented in the main tasks performed by the maximum distance to average vector algorithm, e.g., distance calculation, sorting, assignment of microcell, and centroid calculation. These strategies focused on the most repetitive operations, e.g., distance calculation and sorting, lead to the highest performance.

A not negligible detail about these strategies is that can be combined with that of other proposals without additional distortion. Naturally, said effect enables an interesting opportunity for capitalizing on several of the efficiency approaches proposed for microaggregation and, particularly for MDAV. Interestingly, these computational improvements can also be combined with the functional (data utility) improvements introduced by other works where microaggregation is involved [63, 97, 98].

Chapter 7

Anonymizing cybersecurity data in critical infrastructures

7.1 Introduction

Although all the work presented in previous chapters is devoted to k-anonymous microaggregation (in particular MDAV) and its implications on data utility and usability, here we address a different anonymization approach that we used as part of our participation in the CIPSEC European Project [13].

A different scenario was tackled where unstructured and non-numeric data needs to be protected in terms of privacy. In particular, we refer to the security logs generated by critical infrastructures. This context poses specific challenges to the adoption of privacy technologies, including, e.g., practical issues related to their implementation.

Critical infrastructures (CIs) are either physical or virtual systems whose operation directly supports the functioning of a society. In fact, given the wide reach of critical infrastructures, even small problems on their operation could have a massive impact on a vast population [99]. Besides their reach, CIs are tightly coupled with sensitive areas such as health, telecommunications or economy, which are strategic for a country, so their interruption might imply severe affectation for citizens [99].

We are talking about the infrastructure of hospitals, transportation, oil and energy distribution, banking, environment monitoring, etc. Since these services are essential to the security, prosperity, and social welfare of the population, their corresponding CIs must not stop working and are usually managed by governments.

Given the importance of CIs, their information systems are usually strongly protected against intrusions, mainly against those coming from the Internet. Currently, the resources available for such protection involve "intelligent" cybersecurity solutions that "learn" how attackers behave and ultimately detect and stop future incidents. To do so, these solutions are fueled with so called logs, i.e., detailed information about past events, which are stored as records describing every security incident. Furthermore, logs from multiple sources are commonly shared among several devices and then aggregated so that more input information can improve the efficiency of protection . Aiming to ensure the continuity of their services, CIs have widely adopted such protection mechanisms that generate very detailed and vast information about the entities and interactions involved in security incidents.

Although more granular logs provide more intelligent security protection in CIs, inappropriate sharing of sensitive data may rise serious privacy concerns. Cybersecurity logs could include identifying attributes (IP addresses, user names, fingerprints, etc.), strategic information of companies, e.g., about vulnerabilities, software versions, and several other indicators (path names, user data) that, when disclosed, could easily be used to violate the privacy of the individuals or companies involved. The risk for privacy in this context is not only exacerbated by the increasing need of security services to aggregate shared cybersecurity data to get improved protection mechanisms, but also by the large number of data items enclosed in cybersecurity logs.

Beyond the security they require to protect their information systems, CIs are more exposed to external attacks than conventional infrastructures due to a number of factors. First, since CIs commonly serve a large population, they are desired targets of attackers who aim at magnifying the impact of their offensive [100, 101]. Also, dealing with strategic processes and information, CIs are usually the target of high-level adversaries supported by powerful organizations and even governments [102, 103]. These factors aggravate even more the effects of information leakage to the point

that, e.g., the mere revealing of internal IP addresses or user names might imply severe risks for the integrity of such infrastructures. Interestingly then, the privacy of companies and individuals whose information is revealed in logs may have a direct impact on the security of CIs.

In this chapter, we present an effort to preserve the privacy of individuals and organizations in the context of the CIPSEC framework, and particularly in what involves the sharing of cybersecurity information. The EU project CIPSEC proposes a unified security framework to orchestrate state-of-the-art heterogeneous, diverse, security products aiming to offer high levels of cybersecurity protection. To do so, this framework is able to collect and process security-related data (logs, reports, events) so as to generate security anomaly alerts that can affect a CI health and that can have cascading effects on other CI systems. Our proposal includes a methodology and a tool (data privacy tool, DPT) for obfuscating sensitive data from cybersecurity logs to protect the privacy of the involved entities and individuals.

Namely, our DTP will modify sensitive data with the aim of *sanitizing* or cleaning it from too distinctive attributes. This involves applying several anonymization mechanisms to cybersecurity logs (suppression, generalization, pseudonymization) whose implementation will depend on the specific anonymized attributes.

The work presented in this chapter was published in [104].

7.2 The CIPSEC framework

7.2.1 CIPSEC objectives

The main objective of the EU project CIPSEC is to create a unified security framework that orchestrates state-of-the-art heterogeneous, diverse security tools and offers high levels of cybersecurity protection in IT & OT Critical infrastructure environments. The framework is currently built to collect and process security-related data (logs, reports, events) so as to generate alerts for security incidents that can affect the integrity of a CI together with the potential cascading effect affecting other parts of the CI or event other CIs. The framework aims to be very flexible, adaptable and

causing minimum interference to the normal operations of the CI, allowing for its easy updating when needed in a secure and easy manner.

The CIPSEC framework is capable of collecting events supported by different tools that monitor different aspects of the CI, such as network traffic, malware threats or wireless spectrum among others. Along with the operations for collecting events there is also a reasoning capability based on correlation algorithms that generate alerts for the anomalies detected in the events collected. Additionally, the CIPSEC framework provides with additional services, transverse to the CI monitoring activity, which complements the activities carried out:

- vulnerability tests and recommendations, including cascading effect attacks; which allows to have a snapshot of the level of protection against cyber threats exploiting current vulnerabilities of the assets within a CI ;

- security information sharing, leveraging the report of security incidents either across the infrastructure or to the rest of the world, in order to, for instance, prevent incidents propagation;

- training services, assisting on the usage of the framework and on different characteristics of security management aspects, allowing for an easy training of security staff in the context of the CIPSEC framework;

- updating and patching mechanisms, with the purpose of having a unified view of the status of all the monitoring tools deployed in the infrastructure and giving the possibility to automatically update them, guaranteeing the timely protection against the latest security threats.

The CIPSEC framework was being validated in real environments using the infrastructure of three pilots that covers different domains: rail transportation, environmental monitoring and health sector.

7.2.2 CIPSEC architecture

For the sake of flexibility, the CIPSEC framework was designed to be independent from the underlying critical infrastructure (i.e., independent from the resources managed or the security requirements). The reference architecture of CIPSEC was conceived based on the flow of the data managed within a CI, or, said otherwise, was designed to be infrastructure-agnostic by design. With this aim, the architecture is defined according to the life cycle of the security data (logs, events, reports) acquired, disseminated and consumed in CIs.

Data Acquisition refers to the process of collecting or storing the information (logs, events) generated by end devices devoted to secure the integrity of CIs. Thus, there are multiple sources of this data, e.g., intrusion detection systems.

Data Dissemination covers the transmission of the acquired security information to the components that will further process it. The dissemination of this data is usually performed in real-time describing the multiple processes carrying out in a CI, so that they can be monitored and controlled. In the context of CIPSEC, the information disseminated encompasses security data related to events, alarms, updates, etc.

Data Consumption concerns the processing of the acquired security information after being disseminated to the relevant consumers (e.g., incident correlators). Such information is processed and interpreted to fuel several assessment tools that enable users to make informed decisions.

Figure 7.1 depicts the architecture of the CIPSEC framework based on a group of layers that follow the flow of security data described above. This illustration also shows how the security data travels from the CI to a user interface so that the system admin can take appropriate decisions based in the processing activities carried out by the framework, such as enforcing mitigations or applying contingency plans.

The acquisition layer obtains a lot of information directly from the CI components dedicated such as vulnerability assessment, identity access management, integrity management, endpoint detection and response, and cryptography. The information collected is aggregated and processed by a component called anomaly detection reasoner that triggers security alerts depending on the patterns devised in security data.

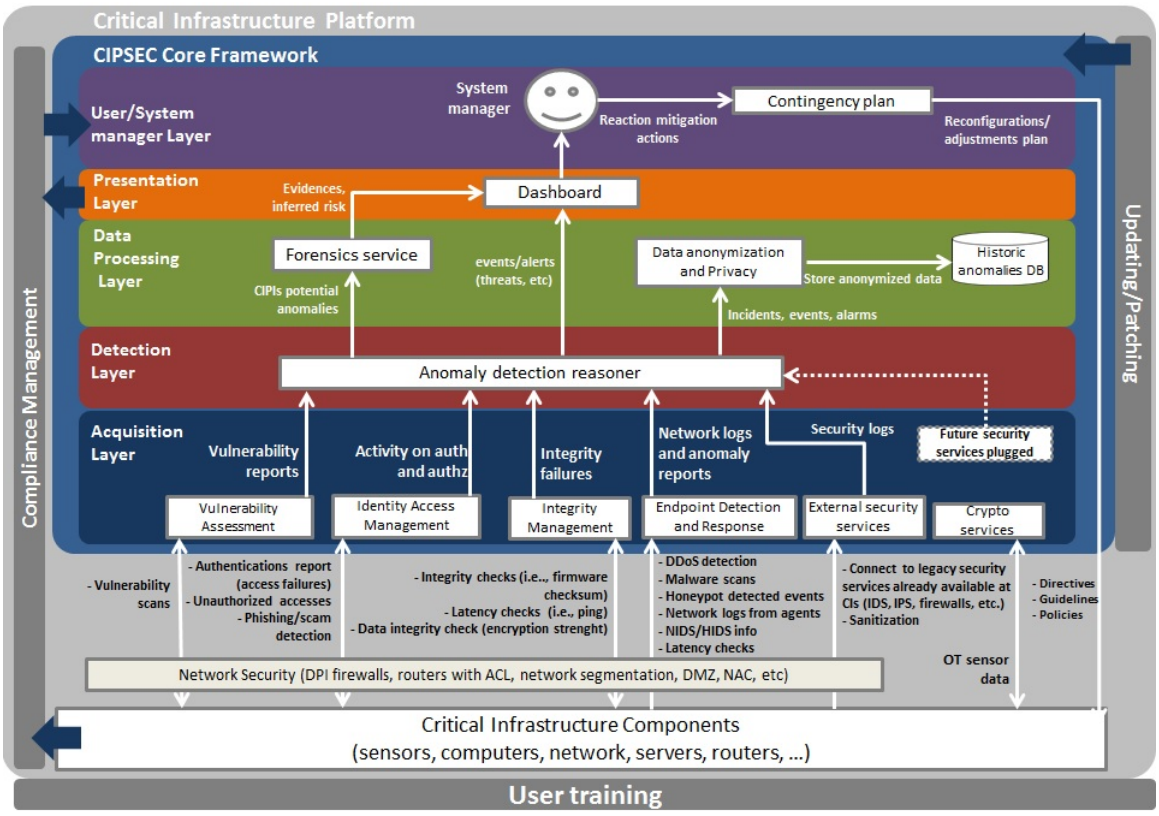


Figure 7.1: CIPSEC Reference Architecture for protecting of critical infrastructures.

The *data processing* layer is on top of the acquisition layer and involves two main components: the DPT and the forensics service. While the forensic service filters and analyzes indicators potentially useful for forensic analysis, the DPT aims at preserving privacy in the security data coming from the CI and at storing such sanitized data in a different database for sharing purposes. This component is the one whose implementation we present in this chapter.

The *presentation layer* aggregates the information produced by the underlying layers through a dashboard that offers a user interface where statistics and evolution indicators are presented to illustrate the security status of the whole CI. Such interface provides with an aggregated and uniform view of this status to the user in order to facilitate decision-making processes.

Finally, other complementary services are also provided by the architecture in order to guarantee the support to end users, the compliance with the CIPSEC framework, and the continuity of the services.

7.3 Data privacy tool

7.3.1 Background on cybersecurity logs and privacy protection mechanisms

Logs are pieces of information that sequentially register the events affecting a system; therefore, when seen aggregated, they constitute evidence of the system behavior. Said diagnosis is fundamental to scrutinize and then fix a given issue, even more in the cybersecurity realm where thousands of attackers are permanently generating incidents that threaten the integrity of critical systems connected to the Internet.

Usually, logs contain a lot of granular information on a related event, starting with a time stamp. Cybersecurity logs may include, e.g., IP addresses, process IDs, hardware information and event descriptions.This information is stored in text files formatted to guarantee its agile reading and processing. For instance, two formats extensively used to store logs are XML and JSON. Both have interesting features based on labels to present information as name-value pairs, e.g., “ID: 123456” or

"Alarm type: critical". Namely, the attributes or information elements of a log are organized using a name (or label) of the attribute and its corresponding value (the raw data). This way of representing information in logs significantly facilitates further selection and replacement (transform) of sensitive attributes.

Roughly speaking, protecting said information against privacy threats builds on these two operations: *selection* of sensitive attributes in the logs and *transformation* of corresponding values to a more private version. This is more deeply described in this document, where our proposal is presented. Although the definition of said operations depend on the context (sharing policies, organization concerns, etc.), the modern ways to structure logs are decreasing the complexity of performing these operations of selection (search) and transform (replace).

First, to protect privacy, sensitive attributes (the target) must be defined and then detected in logs. In practice, this task consists in searching for specific information in plain text. Given the vast log data that cybersecurity systems could generate, such searching for specific items might be daunting if it is expected to be done manually by a human operator. Fortunately, technology can now be used to automatize the detection of this type of attributes. Moreover, the most common logging formats are based on labeling every single piece of information contained in the log. Thus, once the sensitive attribute (or its label) is defined, it is not difficult to retrieve it from the logs along with its value. If the data within logs were not appropriately formatted, sensitive information should be located by looking for specific syntax patterns that such information present in logs. For instance, if IP addresses were considered as sensitive information, the privacy protecting approach could start detecting IP addresses in logs by resorting to its unambiguous syntax. Then, searching for a pattern of four numbers separated by dots would eventually lead the system to find said IP addresses. Regular expressions are powerful constructions that can be used to represent and search such patterns.

After finding the attribute in cybersecurity logs, it has to be protected to preserve the privacy of involved companies or individuals. This implies modifying or transforming the value of the attribute to obfuscate any sensitive information there contained. This task is also referred to as *sanitization* in the sense that it "cleans"

data from too distinctive attributes. To do so, some anonymization or sanitization mechanisms are commonly implemented. These mechanisms are described in the following lines.

Suppression is the simplest strategy to protect privacy in this context. It consists in completely eliminating sensitive information, which can be interpreted as replacing it for a blank or any meaningless string. This implies that no trace of said sensitive data is left which may directly affect the utility of the logs.

Generalization is rather a less destructive anonymization approach. It builds on replacing sensitive information with more general but still meaningful data. For instance, if the sensitive piece of data is the IP address 192.168.1.1, a generalized version would be 192.168.0.0. In contrast with suppression, generalization could keep some utility from the data in log records, depending on the deep of generalization attained.

Pseudonymization is a mechanism that consists in replacing identifying information by artificial identifiers, also called pseudonymous. Since said pseudonymous would be used instead of the original identifier, each time the latter appears, it is possible in practice to recover the original information from its pseudonymized version. Also, if such identifiers are only used for identifying purposes, pseudonymizing them would not affect the utility of information.

As briefly described, the resulting utility of cybersecurity logs may be more or less affected depending on the anonymization mechanism used to protect privacy.

7.3.2 Privacy risks from disclosing cybersecurity logs

In general, logs contain a lot of information since they are aimed at describing the state of a system at a given point in time. Further, an aggregated set of logs should enable an administrator to have a general view of the performance of said system. In particular, the specific amount of data items (we call them attributes) present in a log record will depend on the level of granularity set in the logging service. Interestingly, some equipment, e.g., networking devices, allow to be configured with such high levels of granularity that manufacturers explicitly warn about the risk of saturating storage

or processing resources. Thus, logs could become extremely detailed pieces of data describing a system where companies and individuals are involved.

Cybersecurity logs might include very sensitive data since they are commonly associated with vulnerabilities and security threats. If that information fell into the wrong hands, it could cause severe damage to the data owners. Besides, the level of granularity of security logs is usually higher to afterwards enable the detection of security breaches (which use to be provoked by undercover interactions), so more and more attributes are included in logs to improve protection to the same extent. As a consequence, the potential leakage of this information implies serious privacy risks for the entities involved, not only due to the weaknesses that such logs could reveal to attackers, but also due to the increased detail of the information.

Ironically, the risk of leaking this information does not necessarily come from deliberate attacker intrusions to steal it, but from the voluntary release of such logs when sharing them to other partners. In fact, sharing cybersecurity logs has become a common practice among organizations as a collaboration mechanism to enhance the effectiveness in detecting and preventing security threats. The attributes characterizing a security incident in a system, e.g., IP addresses, file names, sizes, can be shared with system administrators with the aim to help other systems detect or prevent related threats. More specifically, information sharing enables sharing partners to enhance their defensive capabilities, i.e., detecting, responding, and recovering from cybersecurity incidents. As a matter of fact, the collective aggregation of shared security logs is currently the main input fueling powerful antivirus and network security devices.

Despite the great benefits that sharing cybersecurity logs may bring, some challenges still remain. One of said challenges is safeguarding sensitive information that might be included in these logs, i.e., protecting the privacy of the entities whose information is shared. The violation of privacy in this context (e.g., due to the disclosure of personally identifiable information) may have serious consequences, particularly for companies, such as financial loss, legal action, loss of reputation, and exposure to protection capabilities.

As explained above, the nature of cybersecurity information contained in logs is inherently sensitive since it includes several attributes and also some very punctual data items that may reveal strategic operations of systems regarding their security. Virtually every computing device and application are enabled to generate this type information, especially if they engage networking or web interactions. Some examples of sensitive information that could be included in cybersecurity logs are described next.

Timestamp. A time stamp is fundamental to determine the moment when a security incident occurred. The exact date and time of the incident allow to correlate other events that could contribute on the investigation of the threat. However, if coupled, e.g., with individuals, temporal data could also help attackers perform the same correlation to unveil patterns (a person's sleep time, a company's patching calendar) to violate privacy.

IP address. IP addresses individuate devices so that security issues can be associated with the entity where the incident has been generated. Nevertheless, in the same line, IP addresses are key information for privacy attackers to identify the individuals and companies involved. In fact, an IP address could unequivocally represent an individual or a family, so the security logs related to their interactions would reveal such tight association. The mere availability of this information enables further security attacks (denial of service, fingerprinting) to companies, which could reveal even more indicators about potential victims (privacy violations).

IP addresses are not the only data items with this individuating capability. Other attributes that may appear in security logs such as user names, host names and MAC addresses have similar identification capabilities, although their presence is not as common as IP addresses. There are also apparently innocuous indicators that are contained in cybersecurity logs that can serve as identifying parameters when combined, e.g., software version and patch level information, hardware information, system event, file access, etc. Interestingly, the resulting combination of said attributes can be seen as a fingerprint of the associated entity and could be used as an identifier by itself.

Table 7.1: Some attributes whose disclosure in cybersecurity logs might jeopardize privacy.

Attribute	**Privacy risk**
IP address	May enable identification of users and organizations.
e-mail address	May enable identification of users and organizations.
path names	Could disclose user names, directory hierarchies.
patching information	Could reveal software updating calendars, thus when
software versions	Along with other attributes, could enable fingerprinting and identification of users.
incident description	When associated with an origanization, could unveil its vulnerabilities.
organization name	May allow attackers to identify an organization.

Any indicator or attribute included in logs could reveal further sensitive information. The specific privacy risk, however, depends on the context, i.e., on the background information available for the attacker, and his objective, but also on the particular status of the potential victim. For example, path names could disclose information about the work a user might be performing, or operating system and patches names may reveal the preferences of a company regarding their network or software implementation (which it had been keeping secret). Disclosing such information in logs that will be shared may represent a privacy violation for users or the company whether or not the parameters included are critical for each entity.

Besides identifying attributes or other complementary indicators, the information included in cybersecurity logs may be very specific when generated by specialized devices such as routers, antivirus servers, intrusion protection systems, forensic toolkits, SIEMs (security information and event management systems), etc. Moreover, these logs contain very critical information since it is commonly derived from assessment routines, i.e., contemplates "refined data" (which in practical terms implies more and more valuable data). This information might span vulnerability alerts, system artifacts, attack alerts, or summary reports, whose disclosure is a direct threat for the privacy of the entities involved.

The risk to privacy when sharing cybersecurity logs is seriously exacerbated when CIs are involved since the corresponding entities and their workers are more exposed given the strategic role they are playing. In Table 7.1, we describe some attributes whose disclosure in cybersecurity logs may imply serious privacy risks.

7.3.3 Architecture of the data privacy tool

The objective of our DPT is offering privacy for individuals and institutions in a context where cybersecurity logs have to be shared among different partners. While disclosing and aggregating such data may improve the capabilities of security solutions in CIs, the high granularity of logs and the sensitive attributes there contained may jeopardize privacy. Thus, we devise a tool to protect this sensitive data by anonymizing it. This tool encompasses the components described below.

7.3.3.1 Target description

The first step in protecting privacy in cybersecurity logs is determining the set of sensitive attributes that will have to be sanitized. Said otherwise, the specific target of the anonymization mechanisms has to be defined since logs use to hold a lot of information.

The level of sensitivity, however, depends on the specific context in which users and companies perform (their interests, needs, worries, adversaries, etc.). Moreover, although some attributes might be defined as sensitive by default (e.g., identity numbers), or automatic mechanisms could be created to "recognize" them, the operators of the DPT should always have the last word when deciding what attributes to protect by defining a privacy policy.

Evidently then, to locate sensitive attributes and their values within the data provided by logs, some language might be necessary for the user to describe the corresponding targets. If logs were generated by CIs without any visible structure, patterns should be found to detect the sensitive attribute, e.g., looking for quartets of decimal numbers separated by dots to find IP addresses (which the operator would have defined as sensitive). Fortunately, to facilitate its exploitation and analysis, logs are commonly generated in structured formats, sometimes even in hierarchical trees, such that the information be organized according to certain logic and that every attribute value is labeled.

As logs are presented through standard approaches and attribute values are indexed through labels, it is straightforward to refer to such attributes to then retrieve

their values. For instance, let us suppose that the operator is interested in preventing individuation of his company in the logs generated by their devices. Thus, he might have to detect identifying data in logs, such as IP addresses, to anonymize them and protect privacy. There may be different approaches to search for IP addresses in a log, as described below.

- **By keyword.** Within cybersecurity logs, IP addresses are usually labeled with a keyword such as SRC_IP (source IP) or DST_IP (destination IP) or any other. Knowing such keyword, it is pretty easy to obtain the sensitive value associated in the corresponding log.
- **By pattern.** If the sensitive data to be anonymized is not systematically associated with an index or label, a pattern could be used to look for such data. In the IP address example, e.g., we could look for any group of four numbers from 0 to 255 separated by dots, which could be symbolically represented as X.X.X.X.
- **By value.** Still in the case when no specific keyword is available, the value of the sensitive attribute could be directly searched in logs. The drawback of this approach is evidently that this search spans a single value while the first two may encompass a wider spectrum of values.

Since the first step to protect privacy in cybersecurity logs involves searching for a string (keyword, pattern or value) in a piece of text, it is worth noting that, at an implementation level, the use of regular expressions is highly recommended for such tasks. See Figure 7.2 where this component of target description of our DPT is depicted as it would work with the other two components described in the next subsections.

7.3.3.2 Context definition

As stated in the previous section, the sensitivity of some data item is subject to the context where its owner performs. In the same line, the privacy protection mechanism required will vary according to the specific needs and characteristics of the subjects

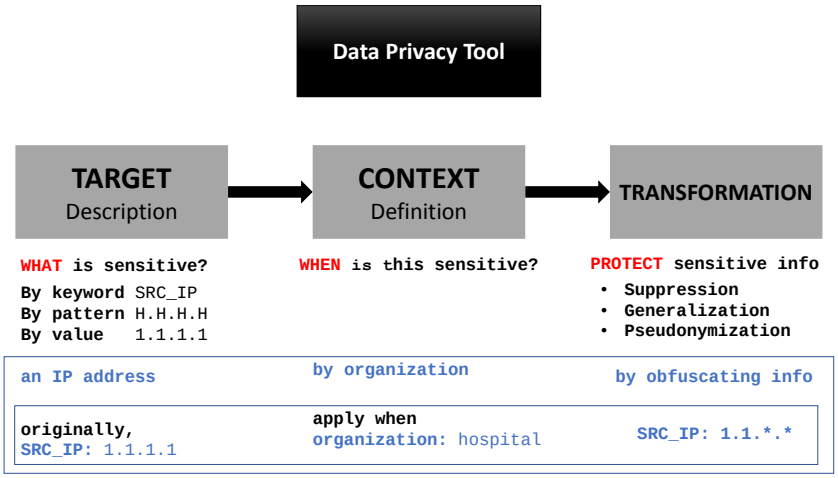


Figure 7.2: Architecture of the data privacy tool.

involved. The context-definition component then enables the user of our DPT to set any restriction or condition on the application of the privacy protection strategy.

While a lot of restrictions could be integrated, there is one in which we are interested for the CIPSEC framework. Since three different pilots or organizations are sharing their cybersecurity logs, the user of the DPT could opt for anonymization or not depending on the organization he belongs to. For instance, air quality monitoring might not involve sensitive attributes for the organization generating such data so could decide not anonymizing their data. Other more complex scenarios may be characterized, e.g., by a company having very specific needs on anonymizing attributes that in other contexts might not be critical to protect. In brief, the scope in which our DTP is used may also define the operation of the DPT.

In Figure 7.2, we illustrate this component within the whole architecture of this tool.

7.3.3.3 Transformation

Once sensitive information and context are defined, sensitive data has to be transformed in order to protect privacy. Said transformation implies perturbing attribute values so that, e.g., identifiers no longer serve to identify individuals, or that sensitive values provide less specific information regarding individuals or companies.

The transformation component in the architecture of our DPT is implemented through the anonymization mechanisms described in Sec. 7.3.1. As explained above, such mechanisms will replace the original sensitive value with another (at least less specific) string. Figure 7.2 shows how this component is integrated in the architecture of our DPT.

It is worth noting that when transformation has to be done dynamically (i.e., when replacing according to a a predefined pattern), regular expressions are also very useful as with target description.

7.3.3.4 Privacy policies

In order to enable users to set the context of their privacy protection, a privacy policy has to be defined. A privacy policy is essentially a list of named rules that include the parameters that characterize the anonymization of sensitive information, i.e., the description of the specific attribute to be anonymized, the transformation mechanism to use, and any other criterion (e.g., the organization whose logs will be anonymized).

7.4 Implementation and Integration in the CIPSEC framework

As explained throughout this chapter, cybersecurity logs enable the intelligent protection provided by the CIPSEC framework. Meanwhile, our DPT aims at preserving the privacy of individuals and organizations involved in such logs when shared among different partners. All the logs generated by several security devices in the CIPSEC infrastructure are aggregated and formatted in standard JSON format in real time to then be stored in a security information and event management server (XL-SIEM). Figure 7.3 depicts a sample of these logs that are further available for sharing in a Malware Information Sharing Platform (MISP).

As mentioned in Section 3, our DPT has three main inputs that guide the anonymization process: the cybersecurity logs that are fueled by the XL-SIEM; a privacy policy, also as a file formatted in JSON (an example is depicted in Figure 7.4); and a scope

```
{'AlarmEvent': {
'USERNAME': '', 'SRC_IP': '188.112.63.117', 'BACKLOG_ID':
'839301cfd5b5417984753 5ffa3e29adc', 'DATE': '2018-07-17
09:00:16',
'DST_IP': '84.88.67.117', 'USERDATA7': '', 'USERDATA6': '',
'FILENAME': '', 'PRIORITY': 4, 'RELIABILITY': 10,
'ORGANIZATION': 'hospital', 'SENSOR':
'AD14C6F3975ED9860E32190EA3DF2535', 'SID_NAME':
'directive_event: Detected access to SAMBA in Honeypot',
'USERDATA2': '', 'USERDATA3': '', 'USERDATA1': 'tcp',
'PROTOCOL': 6, 'RISK': 4, 'USERDATA4': '', 'USERDATA5': '',
'EVENT_ID': '04447d36c0614e3fbe70b5b4612adf2e', 'USERDATA8':
'', 'USERDATA9': '', 'PLUGIN_NAME': 'cyber-monitor',
'DST_IP_HOSTNAME': '00000000', 'RELATED_EVENTS':
'[899f11e885a4080027ea052cd289c2dc,899f11e885a4080027ea052cd2
b27c90]', 'PASSWORD': '', 'PLUGIN_SID': '2', 'CATEGORY':
'Recon', 'SRC_IP_HOSTNAME': '00000000', 'SUBCATEGORY':
'Scanner'}
```


Figure 7.3: Sample of logs generated by the CIPSEC framework.


```
"SRCIP":{
      "ORGANIZATION":"all",
      "ACTION":"generalization",
      "TYPE":"ip_address",
      "KRE":"^SRC_IP$",
      "VRE":"",
      "SRE":""
},
"USERDATA":{
         "ORGANIZATION":"all",
         "ACTION":"suppression",
         "KRE":"^USERDATA*",
          "VRE":"",
         "SRE":""
},
```

```
"ORG":{
      "ORGANIZATION":"hospital",
      "ACTION":"pseudonymization",
      "TYPE":"organization",
       "KRE":"^ORGANIZATION$",
      "VRE":"",
      "SRE":""
}
```


Figure 7.4: Sample of policies defined in JSON format.

that indicates the organization that is executing the anonymization process. The latter argument enables the user to anonymize only the logs that belong to his organization. After logs are anonymized, they are sent to the MISP for sharing purposes.

Finally, for the sake of usability, the control of the execution of the DPT and the selection of the privacy policy is delegated to a graphical user interface integrated in the dashboard of the CIPSEC framework. Figure 7.5 illustrates the components mentioned in this section and their corresponding interactions, while Figure 7.6 shows how a single anonymized log record would look. As a side note, our DPT is implemented using Python.

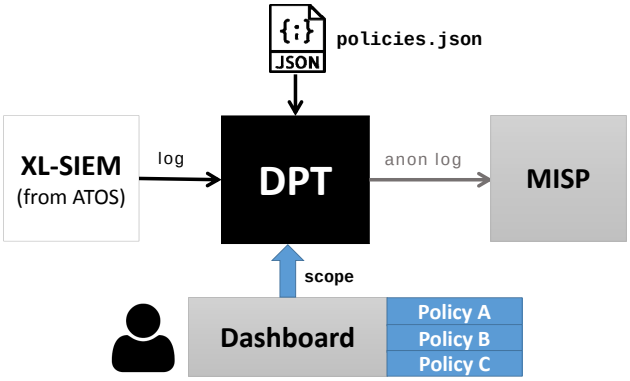


Figure 7.5: Interactions of the DPT with different components of the CIPSEC framework.

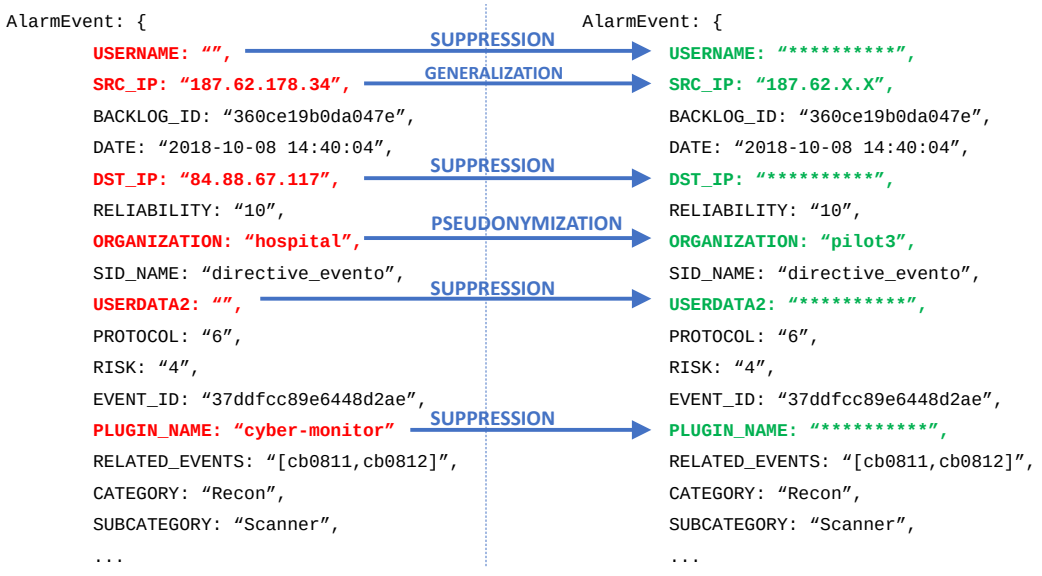


Figure 7.6: A view on how the privacy in cybersecurity logs could be protected through different anonymization mechanisms.

7.4.1 Related work

The concerns on the privacy risks from cybersecurity data are not new. The Government Accountability Office (GAO) of the USA already reports in [105] how the advances in technology have given rise to important challenges to ensure the privacy of personally identifiable information. The GAO recommends implementing privacy practices to protect personal identifying information, especially when managed in critical infrastructures. But in a wider scope, severe regulation is currently been applied in the USA and Europe [106] to protect privacy at every context, essentially by given users great control over their data. Though these documents acknowledge the increasing need to protect privacy, they are regulatory approaches that require implementation according to the specific domain.

Several approaches can be found in the literature that describe privacy preserving mechanisms on unstructured data (e.g., any type of log data). Those mechanisms are based on sanitization (through suppression, generalization, or any kind of perturbation) of such data. Some works address related mechanisms [107], not only focused on protecting privacy but also on preserving the utility of sanitized data [108]. Interestingly, some of such approaches even consider the semantics of the text to be sanitized to get more efficient mechanisms [109, 110]. Unlike those works, our approach focuses on privacy preserving within cybersecurity data in the particular context of the CIPSEC framework.

With regards to the exchange of threat information several initiatives are exploiting their possibilities. For example, the DiSIEM project [(a)] uses it to empower Security Information and Event Management systems by exporting and importing data about incidents detected, allowing for the update of detection rules according to the information imported. DiSIEM also uses MISP as platform for the exchange of information although the privacy considerations are something not considered so far.

[(a)] DiSIEM project web page: http://disiem-project.eu/

7.5 Conclusions

Cybersecurity data generated in the form of logs is very prone to including sensitive information about individuals and organizations, even more so when such logs belong to CIs. The strategic importance of such data, then, makes those individuals and organizations common targets of privacy adversaries. The EU project CIPSEC integrates the cybersecurity information systems of three CIs to improve their threat detection and reaction capabilities. However, since this integration involves sharing such cybersecurity data, there are privacy risks that must be tackled.

The solution we propose addresses this issue by pre-processing logs to anonymize sensitive attributes according to a privacy policy that defines a particular context. Enabling users to set privacy policies is definitely the most important and complicated task since many factors have to be considered to define not only what data to anonymize, but also when and how. Fortunately, the logs generated by information systems are currently represented using more structured and flexible formats (e.g., JSON), which, along with the power of regular expressions to define context, significantly facilitate matching and dynamically perturbing string-based attributes.

As a work in progress, there are several avenues to enhance our privacy tool. Perhaps the most important pending work has to do with assessing the impact of the anonymization mechanisms on the practical utility of cybersecurity logs. Undoubtedly, data perturbing strategies reduce the quality of the information involved so a balance must be reached to protect privacy while minimizing utility loss. Moreover, standardization of the definition of privacy policies is necessary to simplify the configuration of the anonymization mechanisms. This could be a daunting task so automating it according to the requirements of users and organizations might certainly help. Furthermore, a challenge in data sanitization is the reidentification risk posed by inferences based on several attributes exploited simultaneously. Finally, in the same line, more usable (probably graphical) interfaces could be developed to enable end users to provide the parameters of a personalized context in order to better guide the anonymization process of cybersecurity logs.

Chapter 8

Conclusions and future work

8.1 Conclusions

Privacy protection implemented as data perturbation inevitably degrades data utility, more if privacy requirements are stricter. This has been shown through several experiments along this work. However, this evaluation could be relative since there is not a single way to evaluate data utility. Thus, selecting the most appropriate metric is crucial to have a truthful evaluation.

To address this issue, we have presented a methodology to systematically evaluate the impact of privacy protection, particularly of k-anonymous microaggregation, on the empirical utility of data. Assuming machine learning as a popular application domain of data, we have used the accuracy of resulting learning models as a metric of the utility of microaggregated data.

We have found that the default operation of some k-anonymous microaggregation algorithms (MDAV) may not affect empirical data utility significantly. We have argued that the clustering implemented by microaggregation may be acting as a form of averaging and thus denoising. This denoising effect, akin to averaging through clustering, may be the underlying cause of the striking utility of k-anonymous microaggregation.

Although the empirical utility metric we have employed (accuracy) shows a monotonous relationship with MSE, the traditional utility metric of SDC, the latter is not an

ideal metric to determine the impact on the utility of microaggregated data. These results have been corroborated with various algorithms.

When evaluating different microaggregation approaches, we have confirmed the intuition that processing the statistical properties of microdata when building microaggregation algorithms cause an additional slowdown in the degradation of empirical utility. This suggested that, although microaggregation was a high-utility approach, there were some space for improvement. In any case the dependence of the performance of microaggregation algorithms on the internal distribution of the data set was also evidenced.

Beyond the evaluation of k-anonymous microaggregation in terms of the application domain of data, we have proposed a mechanism to preserve its inherent empirical utility. Applied to MDAV, this mechanism has successfully preserved data utility by transforming quasi-identifier values so that the resulting k-anonymous cells enable the construction of a more effective machine learning classifier. Linear Discriminant Analysis and scaling were used as a preprocessing step to transform data such that microaggregation builds a distribution of cells that produces a more accurate learning model.

In terms of accuracy, LDA applied to MDAV outperformed the classical implementation of MDAV. Interestingly, this comes at no cost in terms of running time. Thus, this solution results both functionally and computationally effective. As the efforts presented in next chapters, this proposal was implemented on MDAV but could be applied on other k-anonymous microaggregation approaches.

Besides the natural interest in preserving data utility, the run-time overhead of privacy mechanisms should be low to encourage its adoption in practice, particular when large data sets are involved. If a privacy technology hinders the operation of a service, it will most likely be discarded. Unfortunately, utility preservation usually comes at a price in computational cost, which implies an additional trade off between utility and running time. Then, beyond the primary objective of privacy, there is an important challenge that has to do with the usability of related mechanisms, i.e., with its feasibility to be implemented and used in practice.

We have addressed this issue by proposing a method for speeding up the execution of MDAV. This included algorithmic and algebraic strategies that reduced the running time of MDAV by a factor of 4. This was done by simplifying the internal operations of MDAV, without affecting the resulting utility of data. Conveniently, these strategies can be combined with those of other proposals, provoking a multiplicative effect, without additional distortion.

In times when the world revolves around big data, processing time quickly becomes a bottleneck with respect to the potential applications of large-scale databases. Moreover, domains as critical as health, vehicular traffic, or network intrusion detection are currently using tons of data to help computational systems make real-time, and even life-or-death decisions. Due to such demanding requirements, privacy issues related to data processing are commonly overshadowed. Thus, from the perspective of privacy, we feel that any improvement in (computing) efficiency is not negligible, particularly when the strategy does not entail additional data distortion, and even more when its multiplicative effect may turn a privacy mechanism feasible for a critical application.

Finally, we have explored the anonymization of unstructured data in the form of logs through a practical implementation. In particular, we have presented a tool that preprocesses cybersecurity data to protect the privacy of the entities involved in such logs. Since sensitive information could be released in security logs, this tool anonymizes sensitive attributes, that are further shared, according to a privacy policy that defines a particular context.

The design of the tool enables users to built this policy by considering many factors that describe a particular context. This includes defining what data to anonymize, when and how. This was an interesting practical exercise that evidenced some of the issues of implementing a privacy tool beyond the general assumptions, e.g., with respect to the structure of data.

8.2 Future work

In chapters 3 and 4 we performed a systematic analysis of the impact of k-anonymous microaggregation on the empirical utility of data. Assuming machine learning as the

application domain of data, we used the accuracy of learning models as a utility metric.

These considerations paved the way for future work on improving the performance of microaggregation algorithms. For instance, other anonymization algorithms could be assessed under these conditions to test their behavior when empirical utility is measured. However, some of their reconstruction techniques, e.g., using other than numerical representations for microaggregated data, could complicate the measurement of utility when the application domain is machine learning, so further assumptions or preprocessing should be done.

Additionally, it is worth exploring adaptations or novel contributions for privacy protection that exploit to the maximum the statistical properties of all the information available within microdata. Intuitively, it seems that some of the strategies available for machine learning could be used to preserve the utility of microaggregated data.

In Chapter 5, we presented our proposal on preserving data utility when microaggregating data using MDAV. We leveraged on a machine learning technique called LDA which was applied on two-class data, i.e., for binary classification. Further research in this direction could involve the generalization of this method to address multi-class classification and not only binary classification scenarios.

More generally, it might be interesting to study other machine-learning-based models as mechanisms to represent and microaggregate data to reduce the distortion introduced to variables, combination of variables, or directions that contribute to a more accurate classification. This mainly implies exploring adaptations or novel contributions for privacy protection that exploit to the maximum the statistical properties of all the information available within microdata. This work confirmed the intuition that some of the strategies already available for machine learning could be used to preserve the utility of microaggregated data.

In Chapter 6, we explored a promising mechanism to reduce the running time of k-anonymous microaggregation, particularly on large data sets. The assessment of our proposal is limited in the sense that it is implemented only on top of MDAV, as well as the aforementioned approaches.

Finally, in Chapter 7 we presented the design of a privacy tool for anonymizing sensitive attributes in unstructured data. As a work in progress, there are several avenues to enhance this privacy tool. Perhaps the most important pending work has to do with assessing the impact of the anonymization mechanisms on the practical utility of cybersecurity logs. Undoubtedly, data perturbing strategies reduce the quality of the information involved so a balance must be reached to protect privacy while minimizing utility loss. Moreover, standardization of the definition of privacy policies is necessary to simplify the configuration of the anonymization mechanisms. This could be a daunting task so automating it according to the requirements of users and organizations might certainly help. Finally, more usable (probably graphical) interfaces could be developed to enable end users to provide the parameters of a personalized context in order to better guide the anonymization process of cybersecurity logs.

Bibliography

[1] D. Rebollo-Monedero, J. Forné, and M. Soriano, "An algorithm for k-anonymous microaggregation and clustering inspired by the design of distortion-optimized quantizers," *Data and Knowledge Engineering*, vol. 70, no. 10, pp. 892–921, Oct. 2011. [Online]. Available: http://doi.org/10.1016/j.datak.2011.06.005

[2] D. Rebollo-Monedero, J. Forné, M. Soriano, and J. P. Allepuz, "k-Anonymous microaggregation with preservation of statistical dependence," *Information Sciences*, vol. 342, pp. 1–23, May 2016. [Online]. Available: http://doi.org/10.1016/j.ins.2016.01.012

[3] A. Halevy, P. Norvig, and F. Pereira, "The unreasonable effectiveness of data," *IEEE Journal on Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.

[4] R. L. Rosnow and R. Rosenthal, "Statistical procedures and the justification of knowledge in psychological science," *American Psychologist*, vol. 44, no. 10, pp. 1276–1284, Oct. 1989.

[5] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, "Deep learning for identifying metastatic breast cancer," *arXiv Preprint*, no. 1606.05718, Jun. 2016. [Online]. Available: http://arxiv.org/abs/1606.05718

[6] K. Cukier, "Big data is better data," Technology, Entertainment, Design (TED) Talk, Berlin, Germany, Jun. 2014. [Online]. Available: www.ted.com/talks/kenneth_cukier_big_data_is_better_data

[7] L. Sweeney, "Simple demographics often identify people uniquely," Carnegie Mellon University, Working Paper 3, 2000.

[8] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2008, pp. 111–125.

[9] "AOL search data scandal," Aug. 2006. [Online]. Available: http://en.wikipedia.org/wiki/AOL_search_data_scandal

[10] Assoc. Press, "Divided by citizen wealth tables," *New York Times*, Oct. 2017. [Online]. Available: www.nytimes.com/2009/10/24/business/global/24tax.html?_r=2&ref=global

[11] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering*, vol. 13, no. 6, pp. 1010–1027, 2001.

[12] C. Dwork, "Differential privacy," in *Proceedings of the International Colloquium on Automata, Languages and Programming (ICALP)*, ser. Lecture Notes in Computer Science (LNCS), vol. 4052, Venice, Italy, Jul. 2006, pp. 1–12.

[13] "Project CIPSEC: Enhancing critical infrastructure protection with an innovative security framework," EU, Horizon 2020 Program, ref. H2020-700378, 2016–2019. [Online]. Available: www.cipsec.eu

[14] A. Rodríguez-Hoyos, D. Rebollo-Monedero, J. Estrada-Jiménez, J. Forné, and L. Urquiza-Aguiar, "Preserving empirical data utility in k-anonymous microaggregation via Linear Discriminant Analysis," *Engineering Applications of Artificial Intelligence*, 2020.

[15] A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, A. M. Mezher, J. Parra-Arnau, and J. Forné, "The fast MDAV (F-MDAV) algorithm: An algorithm for k-anonymous microaggregation in big data," *Engineering Applications of Artificial Intelligence*, 2020.

[16] A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, J. Parra-Arnau, and J. Forné, "Does k-anonymous microaggregation affect machine-learned macrotrends?" *IEEE Access*, vol. 6, pp. 28 258–28 277, May 2018. [Online]. Available: http://doi.org/10.1109/ACCESS.2018.2834858

[17] E. Pallarès, D. Rebollo-Monedero, A. Rodríguez-Hoyos, J. Estrada-Jiménez, A. Mohamad Mezher, and J. Forné, "Mathematically optimized, recursive prepartitioning strategies for k-anonymous microaggregation of large-scale datasets," *Expert Systems with Applications*, vol. 144.

[18] J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "Online advertising: Analysis of privacy threats and protection approaches," *Computer Communications*, vol. 100, pp. 32–51, 2017.

[19] J. Estrada-Jiménez, J. Parra-Arnau, A. Rodríguez-Hoyos, and J. Forné, "On the regulation of personal data distribution in online advertising platforms," *Engineering Applications of Artificial Intelligence*, vol. 82, pp. 13–29, 2019.

[20] A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, J. Forné, R. Trapero, A. Álvarez, and R. Rodríguez, "Anonymizing cybersecurity data in critical infrastructures: The cipsec approach," in *2019 International Conference on Information Systems for Crisis Response and Management (ISCRAM)*. ISCRAM, 2019.

[21] A. Rodríguez-Hoyos, J. Estrada-Jiménez, L. Urquiza-Aguiar, J. Parra-Arnau, and J. Forné, "Digital hyper-transparency: leading e-government against privacy," in *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*. IEEE, 2018, pp. 263–268.

[22] J. Estrada-Jiménez, A. Rodríguez-Hoyos, J. Parra-Arnau, and J. Forné, "Measuring online tracking and privacy risks on ecuadorian websites," in *2019 IEEE Fourth Ecuador Technical Chapters Meeting (ETCM)*. IEEE, 2019, pp. 1–6.

[23] L. Sweeney, "Uniqueness of simple demographics in the U.S. population," Carnegie Mellon University, School of Computer Science, Data Privacy Lab, Pittsburgh, PA, Technical Report LIDAP-WP4, 2000.

[24] L. Sankar, S. R. Rajagopalan, and H. V. Poor, "Utility-privacy tradeoffs in databases: An information-theoretic approach," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 6, pp. 838–852, Jun. 2013.

[25] D. Defays and P. Nanopoulos, "Panels of enterprises and confidentiality: The small aggregates method," in *Proceedings of the Symposium on Design and Analysis of Longitudinal Surveys, Statistics Canada*, Ottawa, Canada, Nov. 1993, pp. 195–204.

[26] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 1, pp. 189–201, 2002.

[27] J. Domingo-Ferrer and V. Torra, "Ordinal, continuous and heterogeneous k-anonymity through microaggregation," *Data Mining and Knowledge Discovery*, vol. 11, no. 2, pp. 195–212, 2005.

[28] J. Domingo-Ferrer, F. Sebé, and A. Solanas, "A polynomial-time approximation to optimal multivariate microaggregation," *Computers and Mathematics with Applications*, vol. 55, no. 4, pp. 714–732, Feb. 2008.

[29] A. Hundepool, A. V. de Wetering, R. Ramaswamy, L. Franconi, A. Capobianchi, P.-P. de Wolf, J. Domingo-Ferrer, V. Torra, R. Brand, and S. Giessing, *μ-ARGUS version 3.2 software and user's manual*, Statistics Netherlands, Voorburg, Netherlands, 2003. [Online]. Available: http://neon.vb.cbs.nl/casc

[30] J. Domingo-Ferrer and V. Torra, "A critique of k-anonymity and some of its enhancements," in *Proceedings of the Workshop on Privacy and Security by means of Artificial Intelligence (PSAI)*, Barcelona, Spain, Mar. 2008, pp. 990–993.

[31] D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer, "From t-closeness-like privacy to postrandomization via information theory," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 11, pp. 1623–1636, Nov. 2010. [Online]. Available: http://doi.org/10.1109/TKDE.2009.190

[32] D. Rebollo-Monedero, J. Parra-Arnau, C. Díaz, and J. Forné, "On the measurement of privacy as an attacker's estimation error," *International Journal of Information Security*, vol. 12, no. 2, pp. 129–149, Apr. 2013. [Online]. Available: http://doi.org/10.1007/s10207-012-0182-5

[33] S. Zhong, Z. Yang, and T. Chen, "k-Anonymous data collection," *Information Sciences*, vol. 179, no. 172, pp. 2948–2963, Aug. 2009.

[34] J. Domingo-Ferrer, A. Solanas, and J. Castellà-Roca, "$h(k)$-private information retrieval from privacy-uncooperative queryable databases," *Online Information Review*, vol. 33, no. 4, pp. 720–744, 2009.

[35] J. Cao, B. Carminati, E. Ferrari, and K. L. Tan, "CASTLE: Continuously anonymizing data streams," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, 2009.

[36] T. M. Truta and B. Vinay, "Privacy protection: p-Sensitive k-anonymity property," in *Proceedings of the International Workshop on Privacy Data Management (PDM)*, Atlanta, GA, Apr. 2006, p. 94.

[37] X. Sun, H. Wang, J. Li, and T. M. Truta, "Enhanced p-sensitive k-anonymity models for privacy preserving data publishing," *Transactions on Data Privacy*, vol. 1, no. 2, pp. 53–66, 2008.

[38] N. Li, T. Li, and S. Venkatasubramanian, "t-Closeness: Privacy beyond k-anonymity and l-diversity," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, Istanbul, Turkey, Apr. 2007, pp. 106–115.

[39] J. Brickell and V. Shmatikov, "The cost of privacy: Destruction of data-mining utility in anonymized data publishing," in *Proceedings of the ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining (KDD), Las Vegas, NV, Aug. 2008.

[40] D. Rebollo-Monedero, J. Forné, and J. Domingo-Ferrer, "From t-closeness to PRAM and noise addition via information theory," in *Proceedings of the International Conference on Privacy in Statistical Databases (PSD)*, ser. Lecture Notes in Computer Science (LNCS), Istanbul, Turkey, Sep. 2008, pp. 100–112.

[41] M. Laszlo and S. Mukherjee, "Minimum spanning tree partitioning algorithm for microaggregation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 7, pp. 902–911, Jul. 2005.

[42] A. Solanas and A. Martínez-Ballesté, "V-MDAV: Multivariate microaggregation with variable group size," in *Proceedings of the International Conference on Computational Statistics (CompStat)*, Rome, Italy, Aug. 2006, pp. 917–925.

[43] J. L. Lin, T. H. Wen, J. C. Hsieh, and P. C. Chang, "Density-based microaggregation for statistical disclosure control," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3256–3263, Apr. 2010.

[44] N. Matatov, L. Rokach, and O. Maimon, "Privacy-preserving data mining: A feature set partitioning approach," *Information Sciences*, vol. 180, no. 14, pp. 2696–2720, 2010.

[45] J. Domingo-Ferrer and Ú. González-Nicolás, "Hybrid microdata using microaggregation," *Information Sciences*, vol. 180, no. 15, pp. 2834–2844, 2010.

[46] M. Schmid and H. Schneeweiss, "The effect of microaggregation procedures on the estimation of linear models: A simulation study," *Journal of Economics and Statistics*, vol. 225, no. 5, pp. 529–543, Sep. 2005.

[47] A. Inan, M. Kantarcioglu, and E. Bertino, "Using anonymized data for classification," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, Shanghai, China, Apr. 2009, pp. 429–440.

[48] K. Chaudhuri and C. Monteleoni, "Privacy-preserving logistic regression," in *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada, Dec. 2008, pp. 289–296.

[49] K. Mancuhan and C. Clifton, "Decision tree classification on outsourced data," *arXiv Preprint*, no. 1610.05796, Oct. 2016. [Online]. Available: http://arxiv.org/abs/1610.05796

[50] A. N. K. Zaman, C. Obimbo, and R. A. Dara, "A novel differential privacy approach that enhances classification accuracy," in *Proceedings of the International C* Conference on Computer Science and Software Engineering (C3S2E)*, Porto, Portugal, Jul. 2016, pp. 79–84.

[51] K.-P. Lin and M.-S. Chen, "Privacy-preserving outsourcing support vector machines with random transformation," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Washington, DC, Jul. 2010, pp. 363–372.

[52] K.-P. Lin and M.-S. Chen, "On the design and analysis of the privacy-preserving SVM classifier," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 11, pp. 1704–1717, Oct. 2010.

[53] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Workload-aware anonymization," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Philadelphia, PA, Aug. 2006, pp. 277–286.

[54] Y. Jafer, S. Matwin, and M. Sokolova, "Task oriented privacy preserving data publishing using feature selection," in *Proceedings of the Canadian Conference on Artificial Intelligence*, Montréal, Canada, May 2014, pp. 143–154.

[55] S. Kisilevich, L. Rokach, Y. Elovici, and B. Shapira, "Efficient multidimensional suppression for k-anonymity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 3, pp. 334–347, Apr. 2010.

[56] G. Cormode, E. Shen, X. Gong, T. Yu, C. M. Procopiuc, and D. Srivastava, "UMicS: From anonymized data to usable microdata test," in *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, San Francisco, CA, Oct. 2013, pp. 2255–2260.

[57] M. E. Gursoy, A. Inan, M. E. Nergiz, and Y. Saygin, "Privacy-preserving learning analytics: Challenges and techniques," *IEEE Transactions on Learning Technologies*, vol. 10, no. 1, pp. 68–81, Sep. 2017.

[58] B. Malle, P. Kieseberg, E. Weippl, and A. Holzinger, "The right to be forgotten: Towards machine learning on perturbed knowledge bases," in *Proceedings of the International Conference on Availability, Reliability, and Security (ARES)*, ser. Lecture Notes in Computer Science (LNCS), vol. 9817, Salzburg, Austria, Aug. 2016, pp. 251–266.

[59] T. Li and N. Li, "On the tradeoff between privacy and utility in data publishing," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, Paris, France, Jun. 2009, pp. 517–526.

[60] S. Matwin, J. Nin, M. Sehatkar, and T. Szapiro, "A review of attribute disclosure control," in *Advanced research in data privacy*, ser. Studies in Computational Intelligence, G. Navarro-Arribas and V. Torra, Eds. Switzerland: Springer International Publishing, 2015, vol. 567, pp. 41–61.

[61] G. D'Acquisto, J. Domingo-Ferrer, P. Kikiras, V. Torra, Y. de Montjoye, and A. Bourka, "Privacy by design in big data," EU Agency for Network and Information Security (ENISA), Technical Report TP-04-15-941-EN-N, Dec. 2015. [Online]. Available: http://doi.org/10.2824/641480

[62] "Anonymisation techniques," Article 29 Data Protection Working Party (Independent EU Advisory Committee, Directive 95/46/EC, Article 29), Opinion 05/2014, Apr. 2014.

[63] D. Rebollo-Monedero, A. Mohamad Mezher, X. Casanova Colomé, J. Forné, and M. Soriano, "Efficient k-anonymous microaggregation of multivariate

numerical data via principal component analysis," *Information Sciences*, vol. 503, pp. 417–443, Nov. 2019, in press. [Online]. Available: http://doi.org/10.1016/j.ins.2019.07.042

[64] A. Calviño, "A simple method for limiting disclosure in continuous microdata based on principal component analysis," *Journal of Official Statistics*, vol. 33, no. 1, pp. 15–41, Feb. 2017.

[65] R. Mortazavi and S. Jalili, "Fine granular proximity breach prevention during numerical data anonymization," *Transactions on Data Privacy*, vol. 10, no. 2, pp. 117–144, Aug. 2017.

[66] X. Sun, H. Wang, J. Li, and Y. Zhang, "An approximate microaggregation approach for microdata protection," *Expert Systems with Applications*, vol. 39, no. 2, pp. 2211–2219, Feb. 2012.

[67] R. Mortazavi, S. Jalili, and H. Gohargazi, "Fast data-oriented microaggregation algorithm for large numerical datasets," *Knowledge-Based Systems*, vol. 67, no. 3, p. 195–.205, Sep. 2014.

[68] A. Machanavajjhala, J. Gehrke, D. Kiefer, and M. Venkitasubramanian, "*l*-Diversity: Privacy beyond *k*-anonymity," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, Atlanta, GA, Apr. 2006, p. 24.

[69] J. Domingo-Ferrer, A. Martínez-Ballesté, J. M. Mateo-Sanz, and F. Sebé, "Efficient multivariate data-oriented microaggregation," *Very Large Database (VLDB) Journal*, vol. 15, no. 4, pp. 355–369, 2006.

[70] I. Witten, E. Frank, M. Hall, and C. Pal, *Data mining: Practical machine learning tools and techniques*, 4th ed. Burlington, VT: Morgan Kaufmann Publishers, 2016.

[71] "UCI machine learning repository: Adult dataset." [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Adult

[72] "UCI machine learning repository: Pima Indians diabetes dataset." [Online]. Available: http://archive.ics.uci.edu/ml/datasets/pima+indians+diabetes

[73] V. Ayala-Rivera, A. O. Portillo-Domínguez, L. Murphy, and C. Thorpe, "COCOA: A synthetic data generator for testing anonymization techniques," in *Proceedings of the International Conference on Privacy in Statistical Databases (PSD)*, Dubrovnik, Croatia, Sep. 2016, pp. 163–177.

[74] V. Ayala-Rivera, P. McDonagh, T. Cerqueus, and L. Murphy, "A systematic comparison and evaluation of k-anonymization algorithms for practitioners," *Transactions on Data Privacy*, vol. 7, no. 3, pp. 337–370, Dec. 2014.

[75] V. Ayala-Rivera, P. McDonagh, T. Cerqueus, and L. Murphy, "Synthetic data generation using Benerator tool," *arXiv Preprint*, no. 1311.3312, Nov. 2013. [Online]. Available: http://arxiv.org/abs/1311.3312

[76] N. Mehta and A. Pandit, "Concurrence of big data analytics and healthcare: A systematic review," *International Journal of Medical Informatics*, vol. 114, pp. 57–65, Jun. 2018.

[77] S. Tiwari, H. M. Wee, and Y. Daryanto, "Big data analytics in supply chain management between 2010 and 2016: Insights to industries," *Computers and Industrial Engineering*, vol. 115, pp. 319–330, Jan. 2018.

[78] S. Mamonov and T. M. Triantoro, "The strategic value of data resources in emergent industries," *International Journal of Information Management*, vol. 39, pp. 146–155, Apr. 2018.

[79] R. Bean, "Every company is a data company," *Forbes*, Sep. 2018. [Online]. Available: https://www.forbes.com/sites/ciocentral/2018/09/26/every-company-is-a-data-company/%2523d9840d45cfc5

[80] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional k-anonymity," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, Atlanta, GA, Apr. 2006, pp. 25–35.

[81] "UCI machine learning repository: Breast cancer Wisconsin (original) dataset," 1992. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)

[82] "UCI machine learning repository: Heart disease dataset," 1988. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Heart+Disease

[83] Q. Gong and L. Kun, "Python implementation for Mondrian multidimensional k-anonymity," GitHub, 2017. [Online]. Available: http://github.com/qiyuangong/Mondrian

[84] D. Sánchez, S. Martínez, J. Domingo-Ferrer, J. Soria-Comas, and M. Batet, "μ-ANT: semantic microaggregation-based anonymization tool," *Bioinformatics*, vol. 36, no. 5, pp. 1652–1653, Mar. 2020.

[85] M. Rodriguez-Garcia, M. Batet, and D. Sánchez, "Utility-preserving privacy protection of nominal data sets via semantic rank swapping," *Information Fusion*, vol. 45, pp. 282–295, Jan. 2019.

[86] M. Batet and D. Sánchez, "Semantic disclosure control: semantics meets data privacy," *Semantic Disclosure Control: semantics meets data privacy*, vol. 42, pp. 290–303, Jan. 2018.

[87] G. McLachlan, *Discriminant analysis and statistical pattern recognition*. New York, NY: John Wiley & Sons, 2004, vol. 544.

[88] R. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, p. 179–.188, Sep. 1936.

[89] O. Siohan, "On the robustness of linear discriminant analysis as a preprocessing step for noisy speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Detroit, MI, May 1995, pp. 125–128.

[90] D. Rebollo-Monedero, J. Forné, E. Pallarès, and J. Parra-Arnau, "A modification of the Lloyd algorithm for k-anonymous quantization,"

Information Sciences, vol. 222, pp. 185–202, Feb. 2013. [Online]. Available: http://doi.org/10.1016/j.ins.2012.08.022

[91] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: Promise and potential," *Health Information Science and Systems*, vol. 2, Feb. 2014.

[92] D. Li, X. Shen, and L. Wang, "Connected geomatics in the big data era," *International Journal of Digital Earth*, pp. 1–15, Apr. 2017.

[93] A. Monaco, "Big data: The measure of humankind," *Times Higher Education (THE)*, Oct. 2016. [Online]. Available: www.timeshighereducation.com/comment/big-data-measure-humankind#survey-answer

[94] C. A. R. Hoare, "Algorithm 64: Quicksort," *Communications of the ACM*, vol. 4, no. 7, p. 321, Jul. 1961.

[95] C. A. R. Hoare, "Algorithm 65: Find," *Communications of the ACM*, vol. 4, no. 7, pp. 321–322, Jul. 1961.

[96] M. Solé, V. Muntés-Mulero, and J. Nin, "Efficient microaggregation techniques for large numerical data volumes," *International Journal of Information Security*, vol. 11, no. 4, pp. 253–267, Aug. 2012.

[97] D. Rebollo-Monedero, J. Forné, M. Soriano, and J. Puiggalí Allepuz, "p-Probabilistic k-anonymous microaggregation for the anonymization of surveys with uncertain participation," *Information Sciences*, vol. 382–383, pp. 388–414, Mar. 2017. [Online]. Available: http://doi.org/10.1016/j.ins.2016.12.002

[98] J. Parra-Arnau, J. Domingo-Ferrer, and J. Soria-Comas, "Differentially private data publishing via cross-moment microaggregation," *Information Fusion*, 2019, in press.

[99] T. W. House, "Presidential policy directive – critical infrastructure security and resilience," Feb. 2013. [Online]. Available: https://obamawhitehouse.archives.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil

[100] J. Edwards, "Someone is trying to take entire countries offline and cybersecurity experts say ítś a matter of time because 'it's really easy'," 2018. [Online]. Available: https://www.cnet.com/news/stuxnet-worm-hits-iranian-nuclear-plant/

[101] E. Moyer, "Stuxnet worm hits iranian nuclear plant," *https://www.cnet.com/news/stuxnet-worm-hits-iranian-nuclear-plant/*, Sep. 2010.

[102] S. Swinford, "Russia ing to mount cyber-attack on britain's 'critical infrastructure', gchq and fbi warn," *https://www.telegraph.co.uk/politics/2018/04/16/russia-preparing-mount-cyber-attack-britains-critical-infrastructure/*, Apr. 2018.

[103] V. Woollaston, "Wanna decryptor ransomware appears to be spawning and this time it may not have a kill switch," *https://www.wired.co.uk/article/wanna-decryptor-ransomware*, May 2017.

[104] A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, J. Forné, R. Trapero, A. Álvarez, and R. Rodríguez, "Anonymizing cybersecurity data in critical infrastructures: The CIPSEC approach," in *Proceedings of the International Conference on Information Systems for Crisis Response and Management (ISCRAM)*, València, Spain, May 2019.

[105] "High risk series: An update. Ensuring the security of federal information systems and cyber critical infrastructure and protecting the privacy of personally identifiable information," U.S. Government Accountability Office (GAO), Report to Congressional Committees GAO-15-290, Feb. 2015. [Online]. Available: www.gao.gov/assets/670/668415.pdf

[106] "General Data Protection Regulation (GDPR)," Regul. (EU) 2016/679, Eur. Parliam., Apr. 2016. [Online]. Available: www.eugdpr.org

[107] D. Sánchez and M. Batet, "C-Sanitized: A privacy model for document redaction and sanitization," *Journal of the Association for Information Science and Technology*, vol. 67, no. 1, pp. 148–163, Apr. 2015.

[108] D. Sánchez, M. Batet, and A. Viejo, "Utility-preserving sanitization of semantically correlated terms in textual documents," *Information Sciences*, vol. 279, pp. 77–93, Sep. 2014.

[109] D. Sánchez, M. Batet, and A. Viejo, "Minimizing the disclosure risk of semantic correlations in document sanitization," *Information Sciences*, vol. 249, pp. 110–123, Nov. 2013.

[110] V. T. Chakaravarthy, H. Gupta, P. Roy, and M. K. Mohania, "Efficient techniques for document sanitization," in *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, Napa Valley, CA, Oct. 2008, pp. 843–852.