Research Article

Sarah Zouinina*, Younès Bennani, Nicoleta Rogovschi, and Abdelouahid Lyhyaoui

Data Anonymization through Collaborative Multi-view Microaggregation

<https://doi.org/10.1515/jisys-2020-0026>

Received Jun 30, 2020; accepted Feb 25, 2020

Abstract: The interest in data anonymization is exponentially growing, motivated by the will of the governments to open their data. The main challenge of data anonymization is to find a balance between data utility and the amount of disclosure risk. One of the most known frameworks of data anonymization is k -anonymity, this method assumes that a dataset is anonymous if and only if for each element of the dataset, there exist at least $k - 1$ elements identical to it. In this paper, we propose two techniques to achieve k -anonymity through microaggregation: k -CMVM and Constrained-CMVM. Both, use topological collaborative clustering to obtain k -anonymous data. The first one determines the k levels automatically and the second defines it by exploration. We also improved the results of these two approaches by using pLVQ2 as a weighted vector quantization method. The four methods proposed were proven to be efficient using two data utility measures, the separability utility and the structural utility. The experimental results have shown a very promising performance.

Keywords: Microaggregation, k -Anonymity, Collaborative Topological Clustering

2020 Mathematics Subject Classification: 68T05, 68T30

1 Introduction

Nowadays, data is used in every aspect of the human life. Data is collected by sensors, social networks, mobile applications and connected objects to treat it, explore it, transform it and learn from it. To mine collected data without security breaching, some rules related especially to the privacy of the people on the dataset have to be respected. The process of preserving data privacy is called data anonymization and was used for quite a while to statistical purposes.

Conscious of the costly analysis provided by good quality data, researchers, studied data anonymization methods with the purpose of proposing a good trade-off between identity disclosure and information loss. Data anonymization is *the process of de-identifying sensitive data while preserving its format and data type* [40] [33], generally this procedure is achieved by masking one or multiple values in order to hide some aspects of the data. The growing interest in data anonymization was mainly motivated by the desire of governments and institutions to *open their data* as a proof of democracy and good practices. *Open data* is a very promising study field and it is very challenging because the data released must be anonymized forever with very low re-identification rate and should ensure sufficient quality for the analytics [7, 31]. Aware of the importance of the balance between privacy and utility, many approaches were introduced to tackle this problem, the first

***Corresponding Author: Sarah Zouinina:** Université Sorbonne Paris Nord, LIPN UMR 7030 CNRS, France; Université Abdelmalek Essaadi, ENSA of Tangier, LTI, Morocco; Email: sarahzouinina1@gmail.com

Younès Bennani: Université Sorbonne Paris Nord, LIPN UMR 7030 CNRS, France; LaMSN, La Maison des Sciences Numériques, USPN, France

Nicoleta Rogovschi: Université de Paris, LIPADE, France; LaMSN, La Maison des Sciences Numériques, USPN, France

Abdelouahid Lyhyaoui: Université Abdelmalek Essaadi, ENSA of Tangier, LTI, Morocco

approaches were mainly based on the randomization method which consists of adding noise to data [1]. This method was proven to be inefficient since data reconstruction was feasible [20].

The risk of data privacy breach using randomization was overtaken by the emergence of the k -anonymization method [38]. This group based anonymization method outputs a dataset containing at least k identical records and the anonymization is achieved by firstly removing the key-identifiers like the name and the address and secondly by *generalizing* and/or *suppressing* the pseudo-identifiers which are for example: the date of birth, the ZIP code, the gender and the age. The k value should be chosen in a way to preserve the information provided by the database. The method itself is interesting and was widely studied [3, 26, 27, 30], what gave a strong basis to further works on anonymization. Since the k -anonymity is a group based method, clustering was considered as one of its strongest assets [21, 32]. Microaggregating k elements and replacing the data by the group representatives gives a good trade-off between the information loss and the potential data identification risk [5]. However, the clustering methods presented are based on the k -means algorithm which is prone to local optima and may give biased results.

In this paper, we propose two techniques to achieve k -anonymity through microaggregation: k -CMVM and Constrained-CMVM. Both, use topological collaborative clustering to obtain k -anonymous data. The first one determines the k levels automatically and the second defines it by exploration. To do that we take advantage from the topological structure of the Self Organizing Maps (SOM) [22] and its ability to prone less to local optimas [2]. We will use SOM as a clustering model since it was proven to give good results on practical applications when the aim is to visualize and perform dimensionality reduction. The results of the clustering are enhanced using the collaborative learning process [17]. At the end of the topological learning, the "similar" data will be collected in clusters, corresponding to the sets of similar patterns. These clusters can be represented by a more concise information, such as their gravity center or different statistical moments since we believe that this information is easier to manipulate than the original one.

In the second part of the paper, we are going to introduce the discriminative information to tackle the cases where the labels are given and how they may affect the anonymization process. The exploration of the supervision is given by the Learning Vector Quantization Method (LVQ) [22]. We will use a particular version that gives weights to each of the features what results in better preservation of the utility of the anonymized dataset; the approach is called pLVQ2 and was detailed in [4].

Ultimately, in this paper, we will tackle the following points:

- Multi-view collaborative Self Organizing Maps to achieve data anonymization.
- Constrained collaborative Self Organizing Maps to attain a predetermined k anonymity level.
- The introduction of the discriminative information and the use of the pLVQ2 to achieve highest anonymity levels with a good utility trade-off.

The remainder of this paper is organized as follows: Section 2 discusses the Theoretical Background, Section 3 presents the different algorithms proposed for anonymization, in Section 4 we illustrate the different experimental results and the conclusions and future directions are given in Section 5.

2 Fundamental background of the proposed approaches

In this section we will dress the theoretical foundations of the methods proposed in the remainder of the paper. In the subsection 2.1, we present foundations of k -anonymity through Microaggregation, in the subsection 2.2, we dress an overview of the multi-view collaborative learning and last in subsection 3 we list the notations and the definitions needed in the rest of the paper.

2.1 k -anonymity through Microaggregation

The privacy-preserving method that was widely studied is k -anonymity [38]. The model assumes that person-specific data is stored in a table of attributes and records. To anonymize a dataset, Sweeney [38] proposed

a method that consists of suppressing or/and generalizing the *quasi-identifiers* in a way that any record is indistinguishable from at least $k-1$ records. *Quasi-identifiers* are the variables that, alone, don't disclose much of information about the individuals but if combined, quasi-identifiers might leak the identity of their holder. This approach promoted the idea of grouping similar elements to anonymize them.

The objective of classical k -anonymity is to reduce information loss since data can be hidden in multiple ways depending on the used method. Minimal generalizations and fewer suppressions are preferred. In fact, heuristics to tackle k anonymity are motivated by some preference criteria or user policies [8]. In data mining, the k anonymous data should hold enough information about the respondents to be useful for subsequent operations related to pattern detection.

Venkatasubramanian [41] classifies the methods of anonymization into three classes. First, are the *statistical methods* that proposed measures of privacy in term of variance, the larger the variance, the greater is the privacy of the perturbed data. Second, are the *probabilistic methods* that attempt to quantify the idea of *background information* that a third party might possess. Researchers deployed tools from information theory and Bayesian analysis and more precisely notions of information transfer. The third class of methods is *secure multi-party computations*, these methods were inspired of the cryptography field and the amount of information leakage is measured by in terms of the amount of information accessible by the adversary. One of the most illustrative example of these methods is Yao's the Millionaire Problem [43] where two millionaires wish to know who is richer without revealing any information about each others wealth.

Grouping, as in the *probabilistic approaches* recalls *classification* in case of supervised learning, and *clustering* in the case of unsupervised learning. Li et al. [28] introduced the first algorithm that combines clustering and anonymization. The algorithm forms equivalence classes from the database by finding an equivalence class with records' number less than k . It measures the distance between the found equivalence class and the other equivalence classes and merges it with the nearest equivalence class in order to form a cluster of at least k elements with minimum information distortion. This method gives good computational results but it is very time consuming.

The k -member clustering algorithm was detailed in [5] and it forms clusters of at least k records in a way that the clusters are intersimilar. This approach fixes the value of k , looks for the record and the cluster with the minimal information loss, adds the record to the cluster and iterates until getting clusters with at least k members. Another approach is the *Clustering based greedy algorithm*. First, introduced by Loukides et al. [29], it focuses on capturing the usefulness of the data and protecting its privacy by presenting quality measures, taking into account the attribute, the tuples' diversity and a clustering algorithm. This algorithm is similar to the previous k -member clustering algorithms [5] but with the constraint of maximizing the dissimilarity of sensitive data values (privacy) and minimizing the similarity of the quasi-identifiers (usefulness). Those algorithms gave an opening to further studies on anonymization using clustering [21].

k -anonymity is a global framework to evaluate the amount of privacy in some dataset, as the elimination of key identifiers was proven to be inefficient, microdata was disclosed using the microaggregation technique [14]. Microaggregation is a technique for disclosure limitation aimed at protecting the privacy of data subjects in microdata releases. It is used as an alternative to generalization and suppression to generate k -anonymous data sets, where the identity of each subject is hidden within a group of k elements. Unlike generalization, microaggregation perturbs the data in a way to improve data utility in several ways, such as increasing data granularity, reducing the impact of outliers and avoiding discretization of numerical data.

In microaggregation, records are clustered into small aggregates or groups of size at least k . Rather than publishing an original variable V_i for a given record, the average of the values of the group over which the record belongs is published. In order to minimize information loss, the groups should be as homogeneous as possible.

The approach we are presenting in the following, consists of anonymizing microdata using multi-view topological collaborative microaggregation [16]. To anonymize the dataset, we start by determining the number of views to explore and then we randomly split the data vertically and we build a SOM for each view to get the corresponding prototypes.

2.2 Multi-view Collaborative Learning

Learning and detecting patterns in data is the ultimate aim of machine learning. Suppose we had a collection of datasets explained by different ensembles of attributes, extract information about these elements comes to extracting information about each family of descriptors alone. This is what we call multi-view decomposition [16], each view of the dataset allows to extract specific patterns of the studied data. The collaborative learning, on the other hand, aims to develop methods grounded on statistics to recover the topological invariants from the observed data points [9]. The models that interest us in this paper are those that both, reduce dimension and achieve clustering. Since SOM models [22] allow projection in small spaces that are generally two dimensional and they are often used for visualization and unsupervised topological clustering. In order to improve the SOM's clustering quality, the collaboration approach is used and the outputs of several self-organizing maps are compared. Each dataset is clustered through the SOM approach. The main idea of the used collaboration between different SOM maps is that if an observation from the ii -th dataset is projected on the j -th neuron in the ii - SOM map, then that same observation in the jj -th dataset will be projected on the same j neuron of the jj -th map or one of its neighboring neurons. In other words, neurons that correspond to different maps should capture the similar observations.

Algorithm 1: The Topological Collaborative Multi-view Algorithm

Input: P views dataset $V[ii]$
Output: P SOM optimized $\{w[ii]\}_{ii=1}^P$

Step 1 : Local Step:

- 1: **for** $ii = 1$ to P **do**
- 2: Learn a SOM for view $V[ii]$
- 3: $w^{[ii]} \leftarrow \arg \min_w [R_{SOM}^{[ii]}(\chi, w)]$
- 4: Compute DB index for $SOM[ii]$
 where $DB^{[ii]}$ is the Davies Bouldin index computed using $w^{[ii]}$
- 5: $DB_{Beforecollab}^{[ii]} \leftarrow DB^{[ii]}$
- 6: **end for**

Step 2 : Collaborative learning:

- 7: **for** $ii = 1$ to P **do**
- 8: **for** $jj = 1, jj \neq ii$ to P **do**
- 9: $\lambda_{[ii]}^{[jj]}(t+1) \leftarrow \lambda_{[ii]}^{[jj]}(t) + \frac{\sum_{i=1}^N \sum_{j=1}^{|w|} K_{\sigma(j, \chi(x_i))}^{[ii]} K_{\sigma(j, \chi(x_i))}^{[jj]}}{2 \sum_{i=1}^N \sum_{j=1}^{|w|} (K_{\sigma(j, \chi(x_i))}^{[ii]} - K_{\sigma(j, \chi(x_i))}^{[jj]})^2}$
- 10: $w_{jk}^{[ii]}(t+1) \leftarrow w_{jk}^{[ii]}(t) + \frac{\sum_{i=1}^N K_{\sigma(j, \chi(x_i))}^{[ii]} x_{ik}^{[ii]} + \sum_{jj=1, jj \neq ii}^P \sum_{i=1}^N \lambda_{[ii]}^{[jj]} L_{ij} x_{ik}^{[jj]}}{\sum_{i=1}^N K_{\sigma(j, \chi(x_i))}^{[ii]} + \sum_{jj=1, jj \neq ii}^P \sum_{i=1}^N \lambda_{[ii]}^{[jj]} L_{ij}}$
- 11: $DB_{AfterCollab}^{[ii]} \leftarrow DB^{[ii]}$
- 12: **if** $DB_{AfterCollab}^{[ii]} \geq DB_{BeforeCollab}^{[ii]}$ **then**
- 13: $w_{jk}^{[ii]}(t+1) \leftarrow w_{jk}^{[ii]}(t)$
- 14: **end if**
- 15: **end for**
- 16: **end for**

Therefore, the classical SOM objective function was modified by adding a term of collaboration. Based on the works of [17, 18], we add a new collaboration step to estimate the importance of the collaboration, during

the collaborative learning process. Formally, the objective function is composed of two terms:

$$R^{[ii]}(\chi, w) = R_{SOM}^{[ii]}(\chi, w) + (\lambda_{[ii]}^{[jj]})^2 R_{Col}^{[ii]}(\chi, w) \quad (1)$$

with

$$R_{SOM}^{[ii]}(\chi, w) = \sum_{i=1}^N \sum_{j=1}^{|w|} K_{\sigma(j, \chi(x_i))}^{[ii]} \|x_i^{[ii]} - w_j^{[ii]}\|^2 \quad (2)$$

$$R_{Col}^{[ii]}(\chi, w) = \sum_{jj=1, jj \neq ii}^P \sum_{i=1}^N \sum_{j=1}^{|w|} \left(K_{\sigma(j, \chi(x_i))}^{[ii]} - K_{\sigma(j, \chi(x_i))}^{[jj]} \right)^2 * D_{ij} \quad (3)$$

$$\text{with } D_{ij} = \|x_i^{[ii]} - w_j^{[ii]}\|^2 \quad (4)$$

where P represents the number of views, N - the number of observations, $|w|$ is the number of prototype vectors from the ii SOM (the number of neurons). $\chi(x_i)$ is the assignment function which allows to find the Best Matching Unit (BMU), it selects the neuron with the closest prototype from the data x_i using the Euclidean distance.

The value of the collaboration link λ is determined. This parameter determines the importance of the collaboration between each two SOM, i.e. to learn the collaboration link between all datasets and maps. Its value is in the interval $[1-10]$, 1 - for the neutral link, when no importance to collaboration is given, and 10 for the maximal collaboration within a map. Its value changes for each iteration during the collaboration step. In the case of the collaborative learning, as it is shown in the Algorithm 1, this value depends on topological similarity between both collaboration maps.

This function depends on the distance between two neurons and is defined as follows:

$$K_{\sigma(i,j)}^{[cc]} = \exp \left(-\frac{\sigma^2(i,j)}{T^2} \right) \quad (5)$$

$\sigma(i, j)$ represents the distance between two neurons i and j from the map, and it is defined as the length of the shortest path linking cells i and j on the SOM. $K_{\sigma(i,j)}^{[cc]}$ is the neighborhood function on the $SOM[cc]$ between two cells i and j . T is the temperature which allows to control the size of the neighborhood influence of a cell on the map, it decreases with the T parameter. The value of T can be decreased between two values T_{max} and T_{min} .

The nature of the neighborhood function $K_{\sigma(i,j)}^{[cc]}$ is identical for all the maps, but its value changes from one map to another: it depends on the closest prototype to the observation that is not necessarily the same for all the SOM maps. Indeed, during the collaboration with a SOM map, the algorithm takes into account the prototypes of the map and its topology (the neighborhood function).

3 Proposed Anonymization Approaches

Notations

We use the k -anonymity notation: data is organized as a table of rows (Records) and columns (Attributes) where each row is defined as a tuple, the tuples are not unique but attributes are. Each row is an ordered m -tuple of values $\langle a_1, a_2, \dots, a_j, \dots, a_m \rangle$.

Notation 1. Let $T\{A_1, A_2, \dots, A_m\}$ be a table with a finite number of tuples corresponding to attributes $\{A_1, A_2, \dots, A_m\}$. Given $T = \{A_1, A_2, \dots, A_m\}$, $\{A_l, \dots, A_k\} \subseteq \{A_1, A_2, \dots, A_m\}$
For $t \in T$, $t[A_l, \dots, A_k]$ refers to the tuple of elements $x_l, \dots, x_k$ of $A_l, \dots, A_k$ in T .

Let us consider a table T of size $n \times m$, m is the number of attributes and n is the number of elements. The table is denoted $T = \{A_1, A_2, \dots, A_m\}$.

Definition 3.1. *k-anonymity* $AT\{A_1, A_2, \dots, A_m\}$, is a table, OT is said to be *k-anonymous* if and only if each tuple in AT has at least k occurrences.

Definition 3.2. The Davies Bouldin Index The DB index [11] is based on a similarity measure of clusters R_{ij} that is a fraction of the dispersion measure s_i and the cluster dissimilarity d_{ij} [25]. R_{ij} should satisfy the following:

$$\begin{aligned}
 R_{ij} &\geq 0 \\
 R_{ij} &= R_{ji} \\
 R_{ij} &= 0 \quad \text{if} \quad s_i = s_j = 0, \quad s_i = \frac{1}{\|c_i\|} \sum_{x \in c_i} d(x, c_i) \\
 R_{ij} &> R_{ik} \quad \text{if} \quad s_j > s_k, \quad d_{ij} = d_{ik} \\
 R_{ij} &> R_{ik} \quad \text{if} \quad s_j = s_k, \quad d_{ij} < d_{ik} \\
 R_{ij} &= \frac{s_i + s_j}{d_{ij}} \quad \text{where} \quad d_{ij} = \text{dist}(w_i, w_j) \\
 DB &= \frac{1}{n_c} \sum_{i=1}^{n_c} R_i \quad \text{where} \quad R_i = \max_{j=1, \dots, n_c} R_{ij}, \quad i = 1..n_c
 \end{aligned}$$

Where w_i are the prototypes of the neuron, n_c is the number of cells, c_i is the i^{th} cell.

Davies-Bouldin is a cluster validity index used to measure the "goodness" of a clustering result [11]. It takes into account the compactness and the separability of clusters and works best and foremost with hard clustering (when the clusters have no overlapping partitions).

Since the objective is to obtain clusters with minimum intra-cluster distances, small values for DB are interesting, the usage of this validity index is justified by our wiliness to evaluate how the elements of the same cluster are similar. Therefore, this index is minimized when looking for the best number of clusters [37].

3.1 Pre-Anonymization Step

3.1.1 k-CMVM

In this work, we propose to use a pre-anonymization step in the approach which can give the choice to have two different levels of anonymization. The first using the prototypes of the BMUs(k-CMVM) and the second uses the linear mixture of models(Constrained CMVM).

The Self Organizing Maps, when introduced by Kohonen [39], seemed like a simple yet powerful algorithm to produce "order out of disorder" [19] by building a one or two dimensional lattice of neurons for capturing the important features contained in an input space. SOM are based on *competitive learning*, in other words the output neurons of the map compete among themselves to be activated or fired. In the course of this competition, the neurons are selectively tuned to the various input patterns. Their locations become ordered with respect to each other in a meaningful way. The best tuned neuron is called the *winner neuron* or the *Best Matching Unit*, in our case, we chose to encode the input vector by its corresponding prototypes i.e. Best Matching Unit. The idea joins the Group Anonymization methods since the SOM creates a map of neurons i.e. clusters and each cluster is defined by its prototype so the closest representative of an element is the prototype of the cluster it belongs to.

3.1.2 Constrained CMVM

In [23], Kohonen extended the use of the SOM by proving that instead of representing inputs by the "Best Matching Unit" i.e.. the "Winning neuron", they are described using the linear mixture of the reference vectors [24]. This novel method analyzes input data and approximates it by a set of models that defines the item more

accurately. Compared to the classical SOM learning process where only the BMUs are used. The linear mixture of models preserves better the information.

Let us consider each input as a Euclidean vector x of dimensionality n . The SOM matrix of prototypes is denoted as M of size $(p \times n)$ where p is the number of nodes in the SOM. To get the coefficients of the models we minimize the following equation:

$$\min \|M'\alpha - x\|, \quad (6)$$

where α is a vector of non negative scalars α_i . The constraint of non negativeness is important when dealing with inputs consisting of statistical indicators because their negatives have no meaning.

For the solution of the above objective function, there exist several ways. The most used and straightforward is the gradient-descent optimization. It's an iterative algorithm that can take into account the non-negativity constraint.

The present fitting problem belongs to the quadratic optimization, for which numerous methods have been developed over the years. A one-pass solution is based on the Kuhn Tucker theorem [15].

3.2 Fine tuning

One of the challenges in data mining is to mine multi-view data distributed on different sites. Here, we propose to use collaborative clustering in an attempt to answer this problem. This method can deal with multi-source data i.e. several sets that are presented with the same individuals in different attributes' spaces where even the data type can be different. In other words, each database is a view of a global dataset about the same individuals. This way, the curse of dimensionality is implicitly dealt with, as the algorithm treats each part alone and the results are proved to be more accurate.

The k -CMVM & Constrained-CMVM (algorithm 2 & 3) build classical SOM for each view of the dataset and uses the collaborative paradigm to exchange topological informations between collaborators as described in the algorithm 1. It takes the Davies Bouldin index [11] which is a clustering evaluation indicator that reflects the quality of the clustering, as a stopping criterion. If DB decreases, the collaboration is positive and if it increases, we stop the collaboration and use the initial map. We mean by a positive collaboration the fact that the collaborators improve the clustering quality of one another; a negative collaboration, on the contrary, is used to describe when a collaborator affects another negatively by deteriorating the quality of its clustering. By using the DB index as a stopping criterion, we control how the views collaborate and we only collaborate if the collaboration improves the clustering. Therefore, the collaboration allows us to obtain more homogeneous clusters by using the topological information from all the views.

After the clustering and collaboration step, the pre-anonymization step where the elements of each of the collaborating maps are coded using the BMUs for the k -CMVM, and using the linear mixture of the map's prototypes in Constrained-CMVM. We found that the use of the linear mixture of models gave better results than anonymizing the data with BMUs because it preserves most of the information contained in each element. The pre-anonymized parts are then reorganized in the same way as the original dataset.

The second part of each of the algorithms makes a huge difference between the two algorithms. On the one hand, k -CMVM, outputs a pre-anonymized dataset that will be fine-tuned using a SOM model where the map size is determined by the Kohonen heuristic [22]. The resulting dataset is recoded using the prototypes of the closest object to the BMU and we examine the anonymity level of the dataset. The k levels is not a predefined value but it is given automatically by the model.

On the other hand, for the Constrained-CMVM algorithm, the fine tuning step works as follow: we use a constrained SOM on the pre-anonymized dataset. To have a constrained map, we initially create a SOM that is learned on the outputs of the pre-anonymization step as stated before. A k levels of anonymity is predefined and the elements from the neurons that don't respect the constraint of k cardinality are redistributed on the closest neurons. This process modifies the topology of the map, but helps designing groups of at least k elements in each neuron. We code the objects of each neuron using the best matching unit, to get a k -anonymized dataset. We then explore the different k values to determine the one that satisfies our requests.

Algorithm 2: k -CMVM.

Input : $OT(A_1, A_2, \dots, A_m)$ a table to anonymize
 P number of views $V^{[ii]}$

Output: $AT(A_1, A_2, \dots, A_m)$ Anonymized table
 k anonymity level

Collaboration step:

- Generate P views $V^{[ii]}$

$$V^{[ii]} \leftarrow OT(A_j, \dots, A_l); \text{ Where } (j, l) \in 1, \dots, m \quad (7)$$

- Compute $w^{[ii]}$ using the collaboration algorithm 1 with all $V^{[ii]}$

Pre-Anonymization:

For each $V^{[ii]}$, $ii = 1$ to P :

- Find the BMU of each object j in $V^{[ii]}$ using corresponding $w_{jc}^{[ii]}$ where c is the matching neuron:

$$\arg \min (X_i^{[ii]} - w_{jc}^{[ii]})$$

- Code each element j of OT with its corresponding vector: $X'_j \leftarrow [w_{jc(1)}^{[1]}, w_{jc(2)}^{[2]}, \dots, w_{jc(q)}^{[P]}]$, where $c(q)$ is the index of the cell associated with element j .

Fine-tuning and anonymization:

- Build a global SOM using the pre-anonymized dataset OT'
- For each c in cells 1 to n_c
 - Replace the j^{th} element of OT' by w'_{jc} : $X''_j \leftarrow [w'_{jc(1)}, w'_{jc(2)}, \dots, w'_{jc(q)}]$
 - Output results in $AT(A_1, A_2, \dots, A_m)$
- Output level of anonymity:

$$k \leftarrow \text{count}(\text{occurrences}AT(j)) \quad (8)$$

To sum up, the proposed anonymization methods use the multi-view approach with the purpose of treating complex data and multisources data. This method is also used to preserve the quality of the dataset to recode and prevent the dimensionality curse. The number of subsets to be used for collaboration is fixed by the user and it depends on the size of the data. The algorithm 1 uses classical SOM and collaborative paradigm to form the maps by exchanging the topological information between the collaborated maps. In the pre-anonymization step shown in the k -CMVM, & the Constrained CMVM, the dataset is coded using the prototypes of the best matching units for each data point or by the linear mixture of the SOM models. The pre-anonymized data is then fine tuned using BMUs or clustered under the constraint of k elements by neuron.

3.3 Incorporating Discriminative Power

After evaluating the different results of data anonymization using the k -CMVM (algorithm 2) and the Constrained-CMVM (algorithm 3), we wanted to explore the case where the data is labelled and *to what extent the supervision might influence on the quality of the anonymized results?* To tackle this topic we experimented with the Learning Vector Quantization approach (LVQ). This choice was motivated by its ability to improve the clustering results by taking into account the class of each object. The algorithm learns from a subset of patterns that best represent the training set.

Algorithm 3: Constrained-CMVM

Input : $OT(A_1, A_2, \dots, A_m)$ a table to anonymize
 P number of views $V^{[ii]}$
 k anonymity level

Output : $AT(A_1, A_2, \dots, A_m)$ Anonymized table

Multi-view Clustering step :

- Generate P views $V^{[ii]}$

$$V^{[ii]} \leftarrow OT(A_j, \dots, A_l); \text{ Where } (j, l) \in 1, \dots, m \quad (9)$$

- Compute $w^{[ii]}$ using the collaboration algorithm 1 with all $V^{[ii]}$.

Pre-Anonymization :

For each $V^{[ii]}$, $ii = 1$ to P :

- Find the linear mixture of SOM models for each object j in $V^{[ii]}$.

$$c(j)^{[ii]} \leftarrow \sum_{l=1}^{l=q} \delta_l w_{jk}^{[ii]} \quad (10)$$

where $c(j)^{[ii]}$ is the coding of the j^{th} element of the $[ii]^{th}$ view.

δ_l are the coefficients of linear mixture of models.

- Code each element j of OT its corresponding vector. $X'_j \leftarrow [c^{[1]}(j), c^{[2]}(j), \dots, c^{[P]}(j)]$.

Constrained Clustering and Anonymization :

- Build a global SOM using the pre-anonymized dataset OT' .
- Let $|\cdot|$ denote the number of elements in a cluster: find the clusters with $|E_c| \leq k$.
- Redistribute these elements on the other cells in a way to have at least k elements in each remaining cell.
- Recode the table OT' using the matching BMUs, output results in AT .

LVQ method is best known for the simplicity and the rapidity of its convergence, since it is based on the hebbian learning. This is a prototype-based method that prepares a set of codebook vectors in the domain of the observed input data samples and uses them to classify unseen examples.

LVQ was designed for classification problems that have existing data sets that can be used to supervise the learning by the system. LVQ is non-parametric, meaning that it does not rely on assumptions about that structure of the function that it is approximating. Euclidean distance is commonly used to measure the distance between real-valued vectors, although other distance measures may be used (such as Mahalanobis distance), and data specific distance measures may be required for non-scalar attributes. There should be sufficient training iterations to expose all the training data to the model multiple times. The learning rate is typically linearly decayed over the training period from an initial value until it is close to zero. The more complex the class distribution, the more codebook vectors that will be required, some problems may need thousands. Multiple passes of the LVQ training algorithm are suggested for more robust usage, where the first pass has a large learning rate to prepare the codebook vectors and the second pass has a low learning rate and runs for a long time (perhaps 10-times more iterations).

In the LVQ model, each class contains a set of fixed prototypes with the same dimension of the data to be classified. LVQ adaptively modifies the prototypes. In the learning algorithm, data is first clustered using a clustering method and the clusters' prototypes are moved using LVQ to perform classification. We choose to supervise the results of the clustering by moving the center clusters' using the pLVQ2 proposed in algorithm 4 for each of the approaches. We use the pLVQ2 [4] since this upgraded version of the LVQ respects the characteristics of each features and adapts the weighting of each feature according to its participation to the

Algorithm 4: Adaptive Weighting of Pattern Features During Learning**Initialization :**

Initialize the matrix of weights P according to :

$$p_j^i = \begin{cases} 0, & \text{when } i \neq j \\ 1, & \text{when } i = j \end{cases}$$

The codewords $\mathbf{m}$ are chosen for each class using the k-means algorithm.

Learning Phase:

1. Present a learning example x .
2. Let $w_i \in C_i$ be the nearest codeword vector to x .
 - **if** $x \in C_i$, then go to 1
 - **else then**
 - let $w_j \in C_j$ be the second nearest codeword vector
 - **if** $x \in C_j$ then
 - * a symmetrical window win is set around the mid-point of w_i and w_j .
 - * **if** x falls within win , then

Codewords Adaptation:

- * w_i is moved away from x according to the formula

$$w_i(t+1) = w_i(t) + \alpha(t)[Px(t) - w_j(t)] \quad (11)$$

- * w_j is moved closer x according to the formula

$$w_j(t+1) = w_j(t) - \alpha(t)[Px(t) - w_j(t)] \quad (12)$$

- * for the rest of the codewords

$$w_k(t+1) = w_k(t) \quad (13)$$

Weighting Patterns features:

- * adapt p_k^k according to the formula:

$$p_k^k(t+1) = p_k^k(t) - \beta(t)x^k(t)(w_i^k(t) - w_j^k(t)) \quad (14)$$

- * go to 1.

Where $\alpha(t)$ and $\beta(t)$ are the learning rates

discrimination. The system learns using two layers: the first layer calculates the weights of the features and then it is presented to the LVQ2 algorithm.

The cost function of this approach can be written as follows:

$$R_{pLVQ2}(x, w, P) = \begin{cases} \|Px - w_j\|^2 - \|Px - w_i\|^2, & \text{If } C_k = C_j \\ 0, & \text{otherwise} \end{cases} \quad (15)$$

Where C_k is the class k , $x \in C_k$ is a training example, and P is the weighting coefficient matrix,; w_i is the nearest codeword vector to Px and w_j is the second nearest codeword vector to Px . The pLVQ2 with the Collaborative Paradigm enhances the utility of the anonymized data by the k -CMVM and the Constrained-CMVM models. The use of pLVQ2 is done after the collaboration between cluster centers' to improve the results of the Collaboration at the pre-anonymization and the anonymization steps.

4 Experimental Results

4.1 Datasets

The four methods presented earlier, *k*-CMVM, *k*-CMVM++, Constrained-CMVM and Constrained-CMVM++, were tested on several datasets provided by the UCI Machine Learning Repository [13]:

- The DrivFace database contains images sequences of subjects while driving in real scenarios. It is composed of 606 samples of 6400×480 pixels each, acquired over different days from 4 drivers (2 women and 2 men) with several facial features like glasses and beard.
- Ecoli & Yeast datasets contain protein localization sites. Each of the attributes used to classify the localization site of a protein is a score (between 0 and 1) corresponding to a certain feature of the protein sequence. The higher the score is, the more possible the protein sequence has such feature.
- Glass dataset represents oxide content of the glass to determine its type. The study of classification of types of glass was motivated by criminological investigation. Since the glass left at the scene of the crime can be used as evidence...if it is correctly identified!
- The Spam base dataset consists of 57 attributes giving information about the frequency of usage of some words, the frequency of capital letters and other insights to detect if the e-mail is a spam or not.
- Waveform describes 3 types of waves with an added noise. Each class is generated from a combination of 2 of 3 "base" waves and each instance is generated of added noise (mean 0, variance 1) in each attribute.
- Wine data is the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

Table 1: Some Characteristics of Datasets

Datasets	#Instances	#Attributes	#Class
DrivFace	606	6400	3
Ecoli	336	8	8
Glass	214	10	7
Spam base	4601	57	2
Waveform	5000	21	3
Wine	178	13	3
Yeast	1484	8	10

4.2 Utility Measures and Statistical Analysis

The impact of microaggregation on the utility of anonymized data is quantified as the resulting accuracy of a machine learning model [34]. To measure the utility of the provided anonymized datasets we designed a decision tree model and used it to see how the anonymized data was classified by this model. We then compared the separability utility of the results of both approaches before and after introducing the discriminant information to get more insights on how much data *quality* have we traded for the sake of anonymization. The pre-anonymization step was crucial to create anonymized elements by views i.e. we didn't code the whole example by one model, instead, we coded each part of the example, depending on the view it belongs to, by the BMU in the case of *k*-CMVM, and by the linear mixture of the neighboring models in case of the Constrained-CMVM, we then used fine tuning to add another layer of anonymization. In table 2 we illustrate the results of the four algorithms after the anonymization. We would like to call the accuracy of a dataset, the separability

utility since a good utility refers to good separability between the clusters. In table 2, the titles refer to the following:

- **Original:** The initial separability utility of the raw data using the decision tree model with 10 folds cross-validation.
- **k-CMVM:** The separability utility of the dataset using the multi-view clustering with collaboration between the views and using the Kohonen Heuristic to determine the size of the maps to use. The examples during pre-anonymization were coded using the BMUs.
- **Constrained-CMVM:** The separability utility of the dataset using the multi-view clustering with collaboration between the views and using the Kohonen Heuristic to determine the size of the maps to use. The examples during pre-anonymization were coded using the Linear Mixture of Models.
- The ⁺⁺ in the name of the methods refers to discriminant version.

Table 2: Comparison of the separability utility of the base method MDAV, k-CMVM, Constrained CMVM before and after introducing the discriminant information

	DrivFace	Ecoli	Glass	Spam base	Waveform	Wine	Yeast
Original	92.2	82.4	69.6	91.9	76.9	88.8	83.6
MDAV	89.1	75.6	61.2	70.1	69.8	68.4	83.4
k-CMVM	90.3	84.5	82.4	86.4	83.0	69.7	86.3
k-CMVM++	92.4	98.8	94.4	87.1	88.4	70.5	100
Constrained-CMVM	93.2	85.1	75.2	90.6	81.5	74.2	87.4
Constrained-CMVM++	94.1	86.3	85.9	91.5	88.4	77.8	88.7

In table 2, the separability utility of the four methods of data anonymization is compared to the original separability utility of the datasets. It is shown that the separability utility of the anonymized datasets is better than the initial separability utility using raw data. This can be explained by the process by which we anonymized the initial dataset, the process relies on clustering what implies that the different pattern of the datasets were discovered and all the noise was omitted. In other words, this can be explained by the tendency of microaggregation to remove non decisive attributes from the dataset in order to gather together elements that are similar.

For each dataset we proceed by splitting the original data to 3 views, clustering each view using the SOM clustering model, the size of each map is determined automatically by the Kohonen heuristic. The collaboration between the different views is done two by two using the Davies Bouldin index as a stopping criterion, if the index increases, the collaboration goes further and if it decreases the collaboration stops. The views are anonymized by representing each element of the cluster by its representative we then add a fine tuning microaggregation step to get a higher level of data anonymity.

To incorporate the discriminant information we use multi-view clustering with SOM by using the SOM toolbox [42]. For each class we use 10 prototypes to achieve the wLVQ2.

Let's take the Waveform data as an illustrative example. The used Waveform dataset is noisy, what explains that, at the start of the experiments, the separability utility was equal to 76.88%, after using the k-CMVM increased by 6.1% after applying the Constrained-CMVM it increased by 4.6%, for the discriminant versions we obtained an increase of 11.5% with the CMVM++ and an increase of 11.5% (table 2). Same goes for the other datasets (DrivFace, Glass, Spam base, Waveform, Wine, Yeast) where the separability utility obtained after incorporating the discriminant information increased significantly compared to the separability utility at the start of the experiments.

A well known method of the data anonymization using microaggregation literature is the Maximum Distance to Average algorithm (MDAV) introduced by [14]. MDAV represents the key attributes in a data set as

points in the Euclidean space where k -anonymous microaggregation is the partitioning of points in cells of size k . The perturbed attributes are then characterized with a representative point at maximum distance of the average. In table 2, we illustrate the results of the MDAV compared to the k -CMVM and Constrained CMVM algorithms. Both algorithms that we proposed outperform the MDAV method as shown on the figure 2

In table 2, the graphics show a comparison between the different separability utility levels of the methods. In all the cases, k -CMVM and Constrained CMVM outperform the MDAV case. This can be explained by the fact that the MDAV microaggregates the whole data and then it represents each cluster with the farthest element to the cluster center; unlike the methods that we propose, where the multiview clustering and the two levels microaggregation helps preserving the characteristics inherent to each element and the coding occurs on a local dimension.

To evaluate the performance of our proposed approaches, we use the Friedman test and Nemenyi test recommended in [12]. The Friedman test is conducted to verify the null-hypothesis that all approaches are equivalent in the respect of accuracies. If the null hypothesis is rejected, then the Nemenyi test will proceed. In addition, if the average ranks of two approaches differ by at least the critical difference (CD), then it can be concluded that their performances are significantly different. In the Friedman test, we set the significant level $\alpha = 0.05$. The figure 1 shows a critical diagram represents a projection of average ranks classifiers on enumerated axis. The classifiers are ordered from left (the best) to right (the worst) and a thick line which connects the classifiers were the average ranks not significantly different (for the level of 5% significance). As shown in figure 1, **Constrained-CMVM++** achieves significant improvement over the other proposed techniques since it incorporates discriminating information from labels to better position the prototypes in the data space. As a result, the coding of the data is of better quality because it takes into account intra- and inter-class variability.

Figure 1: Friedman and Nemenyi test for comparing multiple approaches over multiple data sets: Approaches are ordered from left (the best) to right (the worst)

The figure 2 shows the projection representations using the Principal Component Analysis (PCA) on the Ecoli, Waveform and Yeast datasets. In the figures we illustrate how the data behaves after anonymization, we can see that the shape of the data does not change after the anonymization but the number of data points represented is fewer since they are over each other. The elements of each cluster are represented in the same manner what implies that the number of points is reduced but the methods respect the initial data structure.

Figure 2: PCA on the anonymized datasets compared to the original data

4.3 Cluster's validity indices

4.3.1 Davies Bouldin Index

The score is defined as the average similarity measure of each cluster with its most similar cluster, where similarity is the ratio of within-cluster distances to between-cluster distances. Thus, clusters which are farther apart and less dispersed will result in a better score. *DB*-Davis Bouldin index [10]:

$$DB = \frac{1}{K} \sum_{k=1}^K \max_{k \neq k'} \frac{\Delta_n(c_k) + \Delta_n(c_{k'})}{\Delta(c_k, c_{k'})} \quad (16)$$

where K is the number of clusters, $\Delta(c_k, c_{k'})$ is the similarity between clusters centres c_k and $c_{k'}$ and Δ_n is the average similarity of all elements from the cluster C_k to their cluster centre c_k . This index evaluates the quality of unsupervised clustering based on the compactness of clusters and a separation measure between clusters. It is based on the ratio of the sum of within-clusters scatter to between-clusters separation. The lower the value of *DB* index, the better the quality of the cluster. In table 3, the index decreased after adding the discriminant information for almost 70% of the tests.

Table 3: Davies Bouldin Index

	DrivFace	Ecoli	Glass	Spam base	Waveform	Wine	Yeast
k-CMVM	3.39	2.68	0.40	2.07	1.51	1.55	2.31
k-CMVM++	4.15	0.59	0.40	2.06	1.37	1.72	0.24
Constrained-CMVM	3.59	1.61	0.55	2.04	1.92	1.51	2.95
Constrained-CMVM++	4.58	0.14	0.51	2.07	1.35	1.86	0.26

4.3.2 Silhouette Index

The silhouette score is calculated using the mean intra-cluster distance and the mean nearest-cluster distance for each sample [35]. This index is based on the measurement of the difference between the average of the distance between the instance x_i and the instances belonging to the same cluster a_i and the average distance between the instance x_i and the instances belonging to other clusters b_i , the closer the silhouette value is to 1 means that the instances are assigned to the right cluster.

$$S = \frac{1}{K} \sum \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (17)$$

It is generally used to find the number of clusters that produce a subdivision of the dataset into dense blocks that are well separated from each other. The score is closer to one when clusters are dense and well separated, which relates to a standard concept of a cluster. In the table 4, the only dataset that shows a different behavior after incorporating the discriminant information is the DrivFace data, this is explained by the nature of the dataset which is unbalanced.

Table 4: Silhouette Index

	DrivFace	Ecoli	Glass	Spam base	Waveform	Wine	Yeast
k-CMVM	0.04	0.26	0.42	0.20	0.18	0.22	0.13
k-CMVM++	-0.09	0.89	0.59	0.21	0.24	0.25	0.84
Constrained-CMVM	0.08	0.24	0.43	0.20	0.13	0.18	0.07
Constrained-CMVM++	-0.07	0.84	0.45	0.21	0.25	0.25	0.81

4.3.3 Calinski Harabasz Index

Also known as the Variance Ratio Criterion, Calinski-Harabasz score [6] is the ratio of the sum of between-clusters dispersion and of within-cluster dispersion for all clusters (where dispersion is defined as the sum of distances squared).

Calinski-Harabasz index is defined as:

$$CH = \frac{\text{trace}(S_B)}{\text{trace}(S_W)} \frac{N - K}{K - 1}$$

where S_B is the between-clusters dispersion matrix, S_W is the within-cluster dispersion matrix, N is the number of examples, K is the clusters number. The Calinski-Harabasz index ranges from 0 (worst classification)

to $+\infty$ (best classification). It's highly dependent on N . All other things being equal, it grows linearly with N . Therefore, its order of magnitude can vary considerably from one dataset to another. As we look for a low intracluster dispersion (dense agglomerates) and a high intercluster dispersion (well-separated agglomerates), the grater is the index the better is the clustering. In table 5, we can deduce that the only dataset where the index didn't increase is the DrivFace dataset, what is due to its nature of unbalanced data.

Table 5: Calinski Harabasz Index

	<i>DrivFace</i>	<i>Ecoli</i>	<i>Glass</i>	<i>Spam base</i>	<i>Waveform</i>	<i>Wine</i>	<i>Yeast</i>
k-CMVM	14.44	135.31	496.45	375.39	496.45	228.54	130.61
k-CMVM++	7.55	7277.62	488.86	529.79	488.86	256.45	8862.88
Constrained-CMVM	16.22	89.77	455.73	296.32	455.73	204.49	70.50
Constrained-CMVM ++	8.62	4720.40	465.46	506.83	465.46	256.09	6081.23

4.4 Structural Utility using the Earth Mover's Distance

We believe that measuring the distance between two distributions is the way to evaluate the difference between the datasets. The amount of utility lost in the process of anonymization can be seen as the distance between the anonymized dataset and the original one.

The Earth Mover's distance (EMD) also known as the Wasserstein distance [36], extends the notion of distance between two single elements to that of a distance between sets or distributions of elements. It compares the probability distributions P and Q on a measurable space (Ω, Ψ) and is defined as follows (We are using the distance of order 1):

$$W_1(P, Q) = \inf_{\mu} \left\{ \int_{\Omega \times \Omega} |x - y| d\mu(x, y) \right\} \quad (18)$$

Where $\mu : \text{prob.measure on } (\Omega \times \Omega, \Psi \otimes \Psi)$ with marginals : $P, Q, \Omega \times \Omega$ is the product probability space. Notice that we may extend the definition so that P is a measure on a space (Ω, Ψ) and Q is a measure on a space (Ω', Ψ') .

Let us examine how the above is applied in the case of discrete sample spaces. For generality, we assume that P is a measure on (Ω, Ψ) where $\Omega = \{x_i\}_{i=1}^n$ and Q is a measure on (Ω', Ψ') where $\Omega' = \{y_j\}_{j=1}^{n'}$ - the two spaces are not required to have the same cardinality.

Then, the distance between P and Q becomes:

$$W_1(P, Q) = \inf_{\{\lambda_{i,j}\}, i,j} \left\{ \sum_{i=1}^n \sum_{j=1}^{n'} \lambda_{i,j} |x_i - y_j| : \sum_{i=1}^n \lambda_{i,j} = q_j, \sum_{j=1}^{n'} \lambda_{i,j} = p_i, \lambda_{i,j} \geq 0 \right\} \quad (19)$$

EMD is the minimum amount of work needed to transform a distribution to another. In our case we measure the EMD between the anonymized and the original datasets, attribute by attribute, to get an idea about the distortion of the anonymized datasets. We then normalize all distances between 0 and 1, then we define the utility by $1 - W_1(P, Q)$. The smaller the distance W_1 is, the more the data utility is preserved.

Table 6: Structural utility

	DrivFace	Ecoli	Glass	Spam base	Waveform	Wine	Yeast
<i>k</i> -CMVM	0.48	0.42	0.47	0.50	0.50	0.50	0.16
<i>k</i> -CMVM++	0.52	0.58	0.53	0.50	0.50	0.50	0.84
Constrained CMVM	0.48	0.63	0.34	0.50	0.49	0.50	0.15
Constrained CMVM++	0.52	0.37	0.66	0.50	0.51	0.50	0.85

4.5 Preserving combined utility

To choose the anonymization method which best addresses the separability-Structural utility Trade-off, we propose to combine the two types of utility structural and separability in a combined form while $\alpha = \frac{1}{2}$:

$$Comb_Utility = \alpha \cdot Separability + (1 - \alpha) \cdot Structural \quad (20)$$

Table 7 summarizes the clustering results of the proposed approaches in terms of combined utility (*Comb_Utility*). As it can be seen, our approach Attribute-oriented generally performs best on all the datasets. To further evaluate the performance, we compute a measurement score by following [?]:

$$Score(A_i) = \sum_j \frac{Comb_Utility(A_i, D_j)}{\max_i Comb_Utility(A_i, D_j)} \quad (21)$$

where $Comb_Utility(A_i, D_j)$ refers to the combined Utility value of A_i method on the D_j dataset. This score gives an overall evaluation on all the datasets, which shows our approach Attribute-oriented outperforms the other methods substantially in most cases.

Table 7: Combined separability and structural utility *Comb_Utility*

	DrivFace	Ecoli	Glass	Spam base	Waveform	Wine	Yeast	Score
<i>k</i> -CMVM	0.61	0.63	0.71	0.60	0.66	0.60	0.51	4.96
<i>k</i> -CMVM++	0.72	0.78	0.74	0.82	0.69	0.60	0.92	5.18
Constrained CMVM	0.71	0.74	0.54	0.70	0.65	0.62	0.51	4.92
Constrained CMVM++	0.73	0.62	0.76	0.71	0.70	0.64	0.87	5.20

As shown in the table 7, the introduction of the discriminant information improves the utility of the anonymized datasets for all of the methods proposed.

5 Conclusion

In this paper we covered in details four data anonymization using microaggregation approaches, the *k*-CMVM & Constrained CMVM that use Collaborative Multi-View Paradigm, and the *k*-CMVM++ & Constrained CMVM++ that we proposed to improve the quality of the anonymized dataset using the ground truth labels. The results shown above prove the efficiency of the methods and illustrate their importance.

The process we used started first by experimenting with the Multi-view clustering since we believe is an efficient way to deal with multisources data and high dimensional elements. Second, we have shown that

the collaborative topological clustering improves the quality of the clustering what makes the model more accurate. Third, the pre-anonymization using the Linear Mixture of SOM gives better results, in terms of the separability utility than using BMUs. Fourth, we found a good trade-off between the separability utility and anonymity levels. Finally, we evaluated the limits and possibilities of incorporating the discriminant information if the ground truth labels were known and compared its performance to the literature and the k -CMVM & Constrained CMVM.

We are looking for other ways to anonymize data and we are experiencing 1D clustering as a way to anonymize data without losing the information it is containing and we want to explore new methods to anonymize unbalanced datasets.

Acknowledgement: This research was partially supported by the ANR Pro-Text, N° ANR-18-CE23-0024-01.

References

- [1] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *ACM Sigmod Record*, volume 29, pages 439–450. ACM, 2000.
- [2] Fernando Bação, Victor Lobo, and Marco Painho. Self-organizing maps as substitutes for k-means clustering. In Vaidy S. Sunderam, Geert Dick van Albada, Peter M. A. Sloot, and Jack Dongarra, editors, *Computational Science – ICCS 2005*, pages 476–483, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
- [3] Roberto J Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 217–228. IEEE, 2005.
- [4] Y. Bennani. Adaptive weighting of pattern features during learning. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings*, volume 5, pages 3008–13, Piscataway, NJ, 1999. IEEE Service Center.
- [5] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. Efficient k-anonymization using clustering techniques. In *International Conference on Database Systems for Advanced Applications*, pages 188–200. Springer, 2007.
- [6] T. Caliński and J Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3(1):1–27, 1974.
- [7] Priyanka Chaudhary, Krupal Suthar, and Kalpesh Patel. An target-based privacy-preserving approach using collaborative filtering and anonymization technique. In Harish Sharma, Kannan Govindan, Ramesh C. Poonia, Sandeep Kumar, and Wael M. El-Medany, editors, *Advances in Computing and Intelligent Systems*, pages 591–596, Singapore, 2020. Springer Singapore.
- [8] Valentina Ciriani, S De Capitani Di Vimercati, Sara Foresti, and Pierangela Samarati. k-anonymous data mining: A survey. In *Privacy-preserving data mining*, pages 105–136. Springer, 2008.
- [9] Antoine Cornuéjols, Cédric Wemmert, Pierre Gançarski, and Younès Bennani. Collaborative clustering: Why, when, what and how. *Information Fusion*, 39:81–95, 2018.
- [10] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.
- [11] D.L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, April 1979.
- [12] Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.
- [13] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.
- [14] Josep Domingo-Ferrer and Vicenc Torra. Disclosure control methods and information loss for microdata. *Confidentiality, disclosure, and data access: theory and practical applications for statistical agencies*, pages 91–110, 2001.
- [15] W. Gentleman. Solving least squares problems (charles l. lawson and richard j. hanson). *SIAM Review*, 18(3):518–520, 1976.
- [16] Mohamad Ghassany, Nistor Grozavu, and Younès Bennani. Collaborative multi-view clustering. In *The 2013 International Joint Conference on Neural Networks, IJCNN 2013, Dallas, TX, USA, August 4-9, 2013*, pages 1–8. IEEE, 2013.
- [17] Nistor Grozavu and Younès Bennani. Topological Collaborative Clustering. in *LNCS Springer of ICONIP'10 : 17th International Conference on Neural Information Processing*, 2010.
- [18] Nistor Grozavu, Mohamad Ghassany, and Younès Bennani. Learning confidence exchange in collaborative clustering. In *The 2011 International Joint Conference on Neural Networks, IJCNN 2011, San Jose, California, USA, July 31 – August 5, 2011*, pages 872–879, 2011.
- [19] Simon S. Haykin. *Neural networks and learning machines*. Pearson Education, Upper Saddle River, NJ, third edition, 2009.
- [20] Zhengli Huang, Wenliang Du, and Biao Chen. Deriving private information from randomized data. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 37–48. ACM, 2005.
- [21] Saira Khan, Khalid Iqbal, Safi Faizullah, Muhammad Fahad, Jawad Ali, and W. Ahmed. Clustering based privacy preserving of big data using fuzzification and anonymization operation. *ArXiv*, abs/2001.01491, 2019.
- [22] T. Kohonen. *Self-organizing Maps*. Springer-Verlag Berlin, Berlin, 1995.

- [23] Teuvo Kohonen. Description of input patterns by linear mixtures of som models. In *Proceedings of WSOM*, volume 7, 2007.
- [24] Teuvo Kohonen. Essentials of the self-organizing map. *Neural Networks*, 37:52–65, 2013.
- [25] Ferenc Kovács, Csaba Legány, and Attila Babos. Cluster validity measurement techniques. In *6th International symposium of hungarian researchers on computational intelligence*. Citeseer, 2005.
- [26] Kristen LeFevre, David J DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM, 2005.
- [27] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 25–25. IEEE, 2006.
- [28] Jiuyong Li, Raymond Chi-Wing Wong, Ada Wai-Chee Fu, and Jian Pei. Achieving k-anonymity by clustering in attribute hierarchical structures. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 405–416. Springer, 2006.
- [29] Grigorios Loukides and Jianhua Shao. Capturing data usefulness and privacy protection in k-anonymisation. In *Proceedings of the 2007 ACM symposium on Applied computing*, pages 370–374. ACM, 2007.
- [30] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.
- [31] Abdul Majeed. Attribute-centric anonymization scheme for improving user privacy and utility of publishing e-health data. *Journal of King Saud University – Computer and Information Sciences*, 31(4):426–435, 2019.
- [32] Brijesh B. Mehta and Udai Pratap Rao. Improved l-diversity: Scalable anonymization approach for privacy preserving big data publishing. *Journal of King Saud University – Computer and Information Sciences*, 2019.
- [33] Balaji Raghunathan. *The Complete Book of Data Anonymization: From Planning to Implementation*. CRC Press, 2013.
- [34] A. Rodríguez-Hoyos, J. Estrada-Jiménez, D. Rebollo-Monedero, J. Parra-Arnau, and J. Forné. Does k-anonymous microaggregation affect machine-learned macro trends? *IEEE Access*, 6:28258–28277, 2018.
- [35] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [36] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.
- [37] Sandro Saitta, Benny Raphael, and Ian FC Smith. A bounded index for cluster validity. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 174–187. Springer, 2007.
- [38] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.
- [39] Kohonen T. *Self-organizing Maps*. Springer Berlin, 2001.
- [40] Nataraj Venkataramanan and Ashwin Shriram. *Data Privacy: Principles and Practice*. Chapman & Hall/CRC, 2016.
- [41] Suresh Venkatasubramanian. Measures of anonymity. In *Privacy-Preserving Data Mining*, pages 81–103. Springer, 2008.
- [42] Juha Vesanto, Johan Himberg, Esa Alhoniemi, and Juha Parhankangas. Self-organizing map in matlab: the som toolbox. In *Proceedings of the Matlab DSP Conference*, pages 35–40, 1999.
- [43] Andrew C Yao. Protocols for secure computations. In *Foundations of Computer Science, 1982. SFCS'08. 23rd Annual Symposium on*, pages 160–164. IEEE, 1982.