

A clustering approach to anonymize locations during dataset de-identification

Jenno Verdonck
imec-DistriNet
Gent, Belgium
jenno.verdonck@kuleuven.be

Kevin De Boeck
imec-DistriNet
Gent, Belgium
kevin.deboeck@kuleuven.be

Michiel Willocx
imec-DistriNet
Gent, Belgium
michiel.willocx@kuleuven.be

Jorn Lapon
imec-DistriNet
Gent, Belgium
jorn.lapon@kuleuven.be

Vincent Naessens
imec-DistriNet
Gent, Belgium
vincent.naessens@kuleuven.be

ABSTRACT

Companies increasingly rely on massive amounts of data for strategic decision making purposes. In order to optimize business intelligence, companies often try to enrich their models with datasets acquired from third parties. Datasets containing sensitive attributes must be anonymized before release. For large datasets containing microdata, an often applied anonymization technique is data generalization with the goal of achieving privacy metrics such as k -anonymity. Location is an often recurring yet strategic attribute in many use cases. Multiple strategies can be employed to obfuscate precise coordinates. For example, the most significant digits can be dropped or their value can be replaced by a ZIP code. While these methods might be useful in some applications, these approaches often result in too much information loss, undermining strategic decision making. This paper proposes a novel approach to anonymize location by means of clustering. Its feasibility is evaluated and compared to traditional techniques.

CCS CONCEPTS

• **Security and privacy** → **Data anonymization and sanitization; Usability in security and privacy.**

KEYWORDS

Anonymization, Location, Clustering, Generalization hierarchies

ACM Reference Format:

Jenno Verdonck, Kevin De Boeck, Michiel Willocx, Jorn Lapon, and Vincent Naessens. 2021. A clustering approach to anonymize locations during dataset de-identification. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021)*, August 17–20, 2021, Vienna, Austria. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3465481.3470020>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ARES 2021, August 17–20, 2021, Vienna, Austria

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-9051-4/21/08...\$15.00

<https://doi.org/10.1145/3465481.3470020>

1 INTRODUCTION

Data is increasingly becoming an essential part of the business intelligence activities in many companies. Profits can grow by exploiting of the data gathered from regular business activities. For example, companies can rely on consumer data to determine the most feasible location to open a new plant. The performance of the prediction models can increase by inserting external data in the decision making system. Such data can be acquired from governmental institutes or bought from other commercial companies that, on their turn, can generate extra revenues by selling data. However, thoughtless release of strategic data can leak sensitive business information. Controlled release of personal data can even be mandatory due to upcoming privacy legislation. The GDPR regulation – which is enrolled in EU since 2018 – serves as a template for creating privacy regulations in other regions all over the world. It stipulates that datasets containing personal information can be exchanged, provided that they are anonymized. This implies that information that may (in)directly track individuals must be removed from the dataset before it may be released.

This paper focuses on the anonymization of large datasets containing personal microdata (e.g. customer data, patient records, ...) with the purpose of sharing them with third parties in a privacy friendly manner. To achieve an acceptable anonymity level, privacy metrics such as k -anonymity are described in literature [22] and implemented in anonymization tools such as ARX [20]. The rationale behind these metrics is that, by applying generalizations, the quasi-identifying attributes (such as age, gender, location...) can no longer uniquely identify individuals. In case of k -anonymity, groups of size k are formed that have exactly the same value for each (generalized) quasi-identifying attribute. These techniques require the user to define generalization hierarchies, which define levels up to what extend each attribute can be generalized. For many attributes, defining generalization hierarchies is straightforward. For example, ages can be generalized by age ranges. As such, the age of 23 becomes [20–25[, and can further be generalized to [20–30[or [20–40[. Similarly, genders can be suppressed. However, meaningful and effective generalization of location – which is an often recurring yet strategic attribute in many scenarios – proves to be a challenging task in many situations. Currently, straightforward approaches – such as splitting the area in equal squares based on GPS coordinates or relying on ZIP codes – offer no or just limited flexibility (as the group shape, area and sizes are fixed). Additionally,

the quality of the generalization operations often heavily depends, among others, on the shape, area and population density therein, on how the ZIP codes in a specific country are organized.

The optimal generalization strategy also heavily depends on the use case. If the optimal business intelligence strategy heavily relies on the absolute location – as is the case in many resource allocation problems –, the generalized location is preferably as close as possible to the real location. Other decisions heavily rely on the fact that neighboring records are in the same generalized group and do not suffer from a less accurate location. This is the case when the spread of diseases within a community needs to be analyzed. This invokes the need for more complex and fine-grained location generalization techniques.

Contributions. This work proposes a clustering approach as an alternative to the mainstream methods to generalize geolocations in datasets. The quality of our approach is demonstrated by applying multiple metrics, and compared to the traditional generalization methods. Moreover, generalization hierarchies are built, applying the newly proposed strategy, and tested in a k -anonymity setting. The utility of the obtained generalized datasets is evaluated. Lastly, this work presents guidelines for generalizing location data.

The remainder of the paper is structured as follows. Section 2 points to related work. Section 3 defines a set of generalization hierarchy quality criteria, after which the hierarchy creation techniques evaluated in this paper are presented. Thereafter, Section 4 details the clustering-based approach. Our approach is evaluated and compared to other methods in Section 5. Section 6 starts with a discussion and, thereafter, extracts general guidelines. Lastly, conclusions are drawn in Section 7.

2 RELATED WORK

Many nowadays applications offer location-based services to users. A considerable amount of research aims at improving the privacy properties of these services. [13] and [2] apply perturbation techniques on the user’s device. In these approaches, exact locations never leave the user’s device. Other papers [14, 23] employ a trusted third party to anonymize location data before forwarding it to the actual service provider. Both approaches apply anonymization techniques during the data collection phase. In contrast to these approaches, our work does not focus on enabling privacy-friendly location based services. Instead, we focus on the anonymization of location data in datasets containing a vast amount of records with a wide range of (quasi-identifying) attributes. This work is therefore complementary to research performed on the topic of location privacy in location-based services. For example, an organization might want to release a database containing location information to a third party after having anonymized the data.

Other services heavily rely on time series data that contain locations. Prototypical examples are fitness applications that track the user’s running or cycling sessions. Time series data can also occur in databases where the purchase history of clients is recorded. Anonymization strategies and de-anonymization attacks on these data are described in literature [10, 11, 15]. Different anonymization techniques apply to time series data and microdata in large datasets. This work focuses on microdata in large datasets.

Many research has been performed on the anonymization of large datasets by applying privacy metrics such as k -anonymity [6, 22]. These contributions propose novel privacy metrics, and propose and evaluate algorithms to achieve an acceptable privacy level. The algorithms require modelers to define generalization hierarchies. Building high-quality generalization hierarchies can be tedious and often requires domain knowledge. Several papers attempt to automate the creation of the generalization hierarchies. [9] and [1] focus on numerical quasi-identifiers while [7] and [5] on categorical attributes. While numerical methods can be applied in straightforward location generalization strategies (ZIP code translation and coordinate rounding), no existing work focuses on creating meaningful location generalizations hierarchies. This paper focuses on increasing the information retained in location data while at the same time automating the process. To this end, general-purpose clustering algorithms [3, 4, 12, 18, 21] are applied to create the generalization hierarchies.

The use of clustering algorithms for the purpose of full dataset anonymization is also explored in research [8, 17, 19]. These approaches attempt to achieve a k -anonymous dataset by directly applying clustering on the records of the dataset without the need for generalization hierarchies. However, these methods often provide sub-optimal anonymized datasets, as these methods often fail to grasp the semantic meaning of attributes. Therefore, they often fail to group categorical attributes in a meaningful way, or require manually defined generalization hierarchies to accomplish this. The generalization hierarchies created in this paper can be used for this purpose.

3 GENERALIZATION HIERARCHIES FOR LOCATIONS

This section first outlines the criteria of suitable (location) generalization hierarchies. Thereafter, a variety of strategies to construct generalization hierarchies for locations are proposed.

3.1 Location generalization quality

This subsection first defines three properties a generalization hierarchy creation strategy should incorporate in order to be considered in this work. Thereafter, we focus on quality properties specific to location hierarchies and the tests conducted in this paper.

A suitable generalization hierarchy **(1) consists of multiple generalization levels** L_i . L_0 represents the values of the original dataset. Upper levels L_i contain less elements than lower levels L_{i-1} , and hence, lead to increased information loss. Additionally, it is mandatory that **(2) each original attribute value maps to exact one element at each level** L_i . Third, **(3) the generalization hierarchy follows a strict tree structure**. This means that an element at generalization level L_i incorporates multiple elements of generalization level L_{i-1} .

In our approach, the original dataset consists of the exact GPS coordinates. Generalizations are constructed by grouping the original coordinates together. Each element in the upper levels $L_{i>0}$ is a new coordinate $c_j^{L_i}$ that is calculated by a fair weighted function of all coordinates $[c_x^{L_{i-1}}, c_{x+1}^{L_{i-1}}, \dots, c_{x+N}^{L_{i-1}}]$ within that specific group. The generalization strategy determines the specific weight function (see Section 3.2).

In order to assess alternative strategies in Section 5, a set of metrics is defined. Some metrics assess the generalization hierarchy while others target the datasets.

Metrics w.r.t. generalization hierarchies.

- The **distance to generalization** metric defines the *median of the distances between each of the original coordinates $c_i^{L_0}$ and their generalized counterparts $c_x^{L_j}$* . A low outcome is desirable, as such behavior implies that actual locations and their generalized counterpart are close to each other. The median was preferred over the average to eliminate the effects of extreme outliers. This metric is meaningful in use cases which heavily rely on the accuracy of the location in the dataset.
- The **neighbor pairing** metric returns *the percentage of the original coordinates $c_x^{L_0}$ that point to the same generalized coordinate $c_i^{L_j}$ as their closest neighbor $c_y^{L_0}$* . Grouping close neighbors together is important in scenarios where correlations need to be extracted between regions and other sensitive information.
- **Group size stability**. The variance of the size of each group should be low within one generalization level. Balanced groups in generalization hierarchies typically have a positive impact on the utility of anonymized datasets. A low standard deviation of group sizes is therefore beneficial.

Note that it is unlikely that one location generalization strategy excels in all three metrics, as the effect of one metric can negatively impact others. For example, pursuing equal group sizes will result in a reduced number of coordinates that are paired to their nearest neighbor.

Metrics w.r.t. anonymized datasets.

- **The amount of suppressed records**. When creating a k -anonymous dataset, a tradeoff is made between generalizing attributes (i.e. location) and suppressing records. To achieve dataset compliant with a predefined k , the algorithm can increase the *generalization level* or suppress all records in *equivalence classes* that contain less than k records. A low *amount of suppressed records* along with low *generalization levels* are preferable.
- **The amount of equivalence classes** contained in a k -anonymous dataset. This is a measure for the amount of information that remains in the data. A lower number of equivalence classes typically expose a lower utility level.
- **The average equivalence class size**. This is defined as the amount of records in the anonymized dataset, not counting suppressed records, divided by the amount of created equivalence classes. This metric can be used as a general utility loss metric and is an indication towards the utility of the remaining data after the suppressed records are removed. A higher average equivalence class size means a loss in utility. A low value for this measurement is therefore preferred.

3.2 Location generalization techniques

Our contribution starts from datasets that contain GPS coordinates of assets (f.i. individuals, places of interest, items...). We present

three approaches to create generalization hierarchies. These hierarchies are later used to anonymize datasets. This section gives an overview of the mainstream approaches, namely *coordinate rounding* and *translation to (and masking) zip codes*. Thereafter, we propose a novel *clustering approach*. All three strategies are depicted in Figure 1.

Coordinate rounding (CR). The most straightforward but naive method rounds off the coordinates by increasingly reducing the amount of significant bits in the coordinate. As displayed in Figure 2, this strategy splits the map in rectangular areas.

Coordinate rounding offers little to no flexibility nor configurability when creating generalization hierarchies. Moreover, the surface of an area covered by its next-level generalization increases with a factor 100 (i.e. 10^2). Hence, the utility of the data decreases significantly when a higher generalization level is applied. In higher generalization levels – where only a few very large areas remain – countries can be divided in very unequally sized areas near to their borders. This is detrimental for the anonymization process, as records in these smaller areas no longer fit in an equivalence class which can lead to a high amount of suppressed records. An example of the coordinate rounding technique for Belgium is displayed in Figure 2.a. The figure displays the fourth and fifth generalization level (i.e. represented by the small and large squares respectively). Note that the accuracy drastically decreases between the two levels.

We propose two improvements to the coordinate rounding technique (Figure 2.b). Firstly, simple coordinate rounding can be replaced by constructing more refined intervals. A major advantage of this strategy is that the increase in size between two levels can be reduced from 100 to 4 (i.e. 2^2). Secondly, starting the intervals in the center of the target area (f.i. a country) increases the distribution of the higher-level generalizations. In this strategy, each location is mapped to the center point of the area that reflects the generalization.

Note that coordinate rounding exposes one major disadvantage. The technique does not take into account differences in population density between areas within a country. Therefore, applying coordinate rounding either results in large suppression rates or high generalization levels to meet a certain anonymity level (in order to compensate for areas that are sparsely populated).

ZIP code translation (ZIP). The ZIP code strategy first maps every coordinate in the dataset to its corresponding ZIP code at generalization level L_1 . It subsequently approximates the centre point of the area covered by that ZIP code by calculating the centroid of the coordinates in the dataset that map to that area. Higher generalization levels are created by subsequently starring out the least significant digit(s) of the postal code, and subsequently approximating the centre point (as in generalization level L_1). This technique is visualized in Figure 3. The grey lines reflect generalization level L_3 , which means that the two least significant digits are starred out. The black lines represent generalization level L_4 (starring out three digits in the ZIP code).

Note that the quality of the generalizations generated by this strategy heavily relies on how ZIP codes are assigned by the specific country. While ZIP codes often take population density into account (by covering less surface in densely populated areas), they often split dense city centers in multiple areas. In addition, many neighboring


level - 0		level - 1	level - 2		
 Original location (coordinates) (XX.XXXXX, YY.YYYYY)	Decimal rounding	(XX.XXXX*, YY.YYYYY*)	(XX.XXX**, YY.YYY**)	...	Coordinate Rounding
	Advanced fine-grained rounding	(XX.XXXXX', YY.YYYYY')	(XX.XXXXX'', YY.YYYYY'')	...	
	Translate to zip code and mask	ZZZZZ	ZZZZ*	...	ZIP Code Translation
	Apply clustering strategy: K-Means, Divisive, Agglomerative replace by cluster centers	(CC.CCCCC ₁ , DD.DDDDD ₁)	(CC.CCCCC ₂ , DD.DDDDD ₂)	...	Clustering strategy

Figure 1: Overview of the location generalization options and their representation.

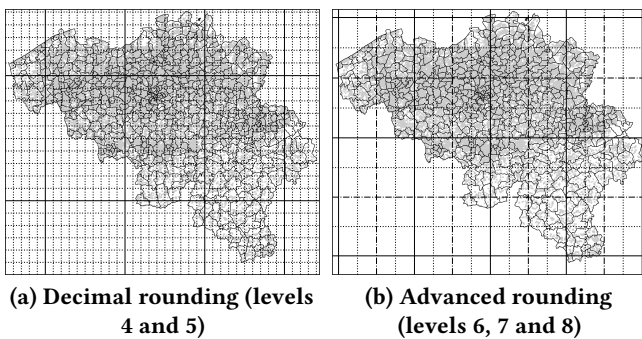


Figure 2: Coordinate rounding techniques Belgium

areas have totally different ZIP codes, artificially splitting them up in the generalization process.



Figure 3: ZIP code translation Belgium (levels 3 and 4)

Coordinate clustering. Both the coordinate rounding and the zip code translation strategy result expose major drawbacks as discussed before. This paper proposes an alternative to generalize location data, namely coordinate clustering. Clustering algorithms aim at grouping neighboring locations. A generalization hierarchy is built by replacing the original coordinate by the center point of the cluster it belongs to. The latter is calculated as the centroid

of the coordinates in that specific cluster. Figure 4 displays four generalization levels after clustering is applied. Note that higher generalization levels lead to larger clusters. Next section applies multiple clustering algorithms, discusses their applicability for this specific purpose, and outlines the generalization strategy.

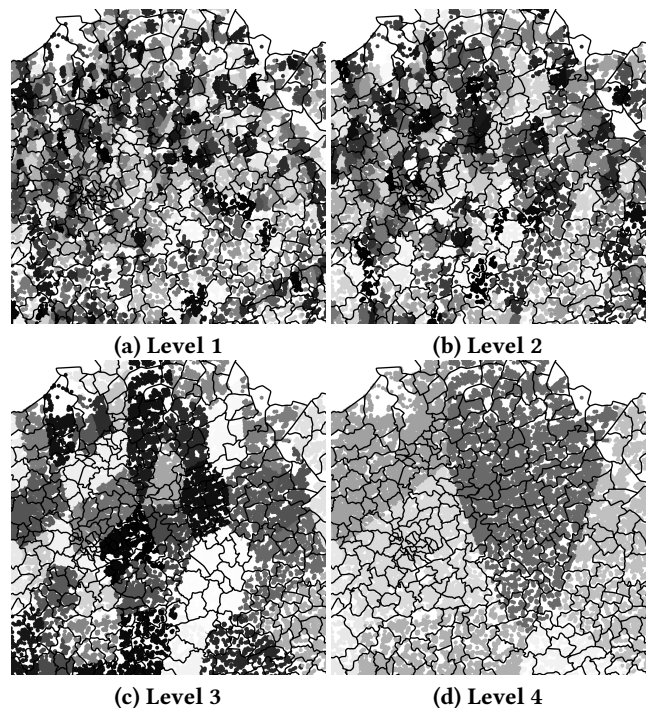


Figure 4: Coordinate clustering Belgium compared to zip (levels 1-4)

4 LOCATION CLUSTERING STRATEGIES

This section demonstrates how location generalization hierarchies are built by applying clustering algorithms. Firstly, a set of viable clustering algorithms are selected. Thereafter, the algorithms are applied for the creation of generalization hierarchies. Note that this work only considers unsupervised clustering algorithms enabling

Table 1: Overview of the selected clustering algorithms

Algorithm	Determinism	Implementation
K-Means	Randomizer	scikit-learn
Agglomerative	Deterministic	scikit-learn
Divisive	Deterministic	own DIANA implementation

to identify clusters in unlabeled data. Table 1 presents an overview of the selected clustering algorithms.

4.1 Basic clustering algorithms

K-Means (KM). The most well-known clustering algorithm is K-Means [4]. K-Means clusters data in K clusters by randomly assigning K points as centroids. It then assigns each point to its closest centroid before recalculating the centroid as the average of all the data points in the cluster. This is repeated until a predefined amount of cycles has passed or the centroids stopped changing. To decrease the calculation time of the algorithm, the minibatch K-Means algorithm proposed by D. Sculley [21] can be used. This executes the K-Means algorithm in smaller batches resulting in faster computation without significant accuracy losses. The scalability of the minibatch K-Means algorithm in combination with the ability to select the required cluster amounts makes it a feasible algorithm for the creation of anonymization hierarchies.

Hierarchical. Hierarchical clustering algorithms create full hierarchical trees of the data. These trees can be cut at any height resulting in the requested amount of clusters. There are two hierarchical clustering strategies, namely agglomerative and divisive clustering.

The *hierarchical agglomerative (HA)* algorithm [18] works by first putting every datapoint in its own cluster. It then selects two clusters to merge according to a linkage function. For this purpose, the *complete* (also sometimes referred to as *max*) linkage function is applied. It aims to minimize the distance between the furthest points of the two clusters. Note that *complete* linkage function was selected over *single (min)* and *average* linkage because early tests demonstrated better results for this linkage function. The favorable results can be attributed to the fact that minimizing the maximal distance between the points in two clusters makes the resulting clusters more compact.

The *hierarchical divisive (HD)* algorithm, in contrast to the agglomerative method, builds the hierarchy in a top-down manner. This is done by first putting every datapoint in the same cluster. Afterwards, the algorithm selects a cluster to be split based on the diameter. A splinter element, the furthest outlier, is then selected to be removed from the cluster. All datapoints closer to this element compared to their current cluster are merged in a new cluster together with the splinter element. This algorithm, described by L. Kaufman et al. [16], is called DIANA.

Density-based clustering algorithms. In initial tests, multiple alternative clustering algorithms were considered. *Optics* and *DBScan* are both density based algorithms. Both techniques merge datapoints in a cluster where the density of the datapoints is similar.

While often cited as a favorable strategy for location based clustering, these techniques are not suitable for this particular use case. First of all, neither allows the user to select the desired amount of clusters. Hence, the strategy does not allow to generate multiple generalization levels. Next, the density based clustering exposes two disadvantages for our purpose. Firstly, the algorithms do not assign data points in sparsely populated areas to clusters. This implies that outliers are effectively lost. Secondly, the cluster sizes – both with respect to surface and amount of records – can be very unbalanced. Note that the anonymization process further suppresses clusters with less than k records if k -anonymity must be achieved. Suppressing clusters – as well as very large clusters – have a negative impact on the utility of the anonymized dataset.

4.2 Building clustering-based generalizations hierarchies

Our work creates generalization hierarchies that contain a predefined amount of clusters (groups) for each level. The generalization of a point is represented by the centroid of the corresponding cluster. The centroid is calculated by averaging the coordinates in the cluster. Note that this strategy was preferred over the geographical center of the area contained by a cluster in order to embrace the location density distribution in a cluster.

Hierarchical trees can be constructed *bottom-up* and *top-down*. The agglomerative clustering algorithm is an example of a bottom-up approach while the divisive algorithm is an example of a top-down approach. Since these two algorithms construct a hierarchical tree, a generalization hierarchy can easily be extracted by cutting the tree at several levels.

Constructing a hierarchical tree with the K-Means clustering algorithm is less trivial. A top-down approach is applied by repeatedly re-clustering the subclusters from a previous level. The amount of groups a subcluster is divided in, is based on the ratio of points in that cluster compared to the total dataset size.

5 LOCATION GENERALIZATION QUALITY ANALYSIS

This section applies the generalization strategies discussed in section 4 to various location based datasets. Moreover, their outcomes are evaluated. This section first outlines the scope of the experiments that are performed in this work. Thereafter, meaningful results are presented with a major focus on the comparison of the generalization strategies.

5.1 Test strategy

The hierarchy creation techniques discussed in this paper are applied to address-centric datasets of three countries, namely Australia, Spain and Belgium. This approach ensures a variety in areal properties. The Australian dataset, retrieved from G-NAF¹, was selected for its diverse population density across different regions. The majority of the people in Australia live around the South and East Coast. On the contrary, the population density in the center of the country is low. Belgium and Spain have more equally distributed

¹<https://data.gov.au/data/dataset/19432f89-dc3a-4ef3-b943-5326ef1dbecca>

populations but differ in shape. Spain’s shape approximates a rectangle, while Belgium is triangular with a more capricious border. The Belgian dataset is provided by the federal government² and the Spanish dataset can be found on openAddresses³. The experiments in this paper are executed in four steps, as outlined in Figure 5. The acquired location datasets are preprocessed in a first step after which generalization hierarchies are created. Thereafter, the hierarchies are applied to anonymize datasets. Finally, the quality of the created location generalization hierarchies is assessed.

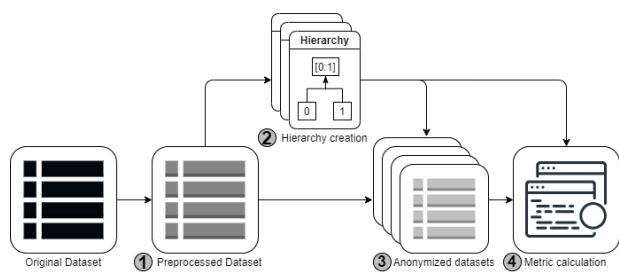


Figure 5: Overview of the different steps in the experimental setup.

STEP 1: Preprocessing. The original datasets were preprocessed by removing incomplete and faulty data. Amongst others, addresses without zip codes are removed as the latter are needed in the ZIP code translation technique described in section 4. Moreover, addresses situated outside the country borders are stripped. The resulting location coordinates were rounded to five decimals after which samples were taken. For the anonymization experiments, representative datasets containing multiple quasi-identifiers are required. For this purpose, synthetic datasets are generated by including data from aggregated datasets for Belgium⁴, Spain⁵ and Australia⁶. The resulting synthetic datasets contain location, gender and age as quasi-identifiers.

STEP 2: Hierarchy creation. For each of the synthetic datasets in the previous step, location generalization hierarchies are created. The techniques outlined in the previous sections are applied, namely coordinate rounding, ZIP code translation, K-Means clustering, agglomerative clustering and divisive clustering. All strategies – except the ZIP code translation method – replace the original coordinate by the centroid of the elements in the group at each level. For the ZIP code generalization method, the center point of each (masked) ZIP code is preprocessed and stored in a separate file. We rely on the mainstream methods to generalize age and gender attributes. Age attributes are generalized by creating age interval which increase in size in upper generalization levels. Gender is generalized by suppression.

STEP 3: Dataset anonymization. The hierarchies are applied to the preprocessed datasets and result in anonymized ones. For this task, we rely on the ARX anonymization API [20] and set the k -anonymity privacy metric as a parameter.

STEP 4: Quality analysis. This step assesses the quality of hierarchies after traditional and clustering hierarchy creation methods were applied. The quality parameters described in Section 3.1 are used during the analysis. Three experiments were performed in this work. Experiment 1 and 2 assess the quality/utility of the hierarchies. Experiment 3 focuses on the anonymity level of the resulting dataset.

Experiment 1: Comparing traditional methods to clustering strategies. A fair comparison between the different hierarchy creation methods is only possible if each hierarchy consists of the same amount of groups. As coordinate rounding and ZIP code translation offer no flexibility at all with respect to the number of groups, this experiment is first executed for ZIP code translation and coordinate rounding. The clustering methods are subsequently initialized with the amount of groups generated by the two aforementioned approaches. The result of this experiment allows for a fair comparison between both of the traditional methods and our proposed clustering strategies.

Experiment 2: Comparing different clustering strategies. The previous experiment compares traditional methods to clustering based methods. As traditional methods offer no flexibility with respect to group sizes, the latter are not optimal. Clustering methods do not impose group size constraints. Hence, the number of groups and levels can be chosen freely. In this experiment, an in-depth comparison is made between clustering strategies. Hierarchies are created with a number of groups ranging from 18000 down to 125, decreasing by 125 at subsequent level. Afterwards, the various clustering methods are compared.

Experiment 3: Anonymization using ARX. In this experiment, the synthetic datasets are anonymized by the ARX data anonymization tool into k -anonymous datasets. This process is repeated for all five proposed hierarchy creation techniques. The cluster-based techniques are applied to create five generalization levels containing 100, 50, 25, 10 and 5 groups respectively. The *gender* hierarchy only contains one suppression level. The *age* attribute is generalized by creating ranges with sizes 5, 10, 20 and 40 after which a complete suppression level is added. The experiment is executed for different k values (i.e. 5, 10, 20, 50, 100). Finally, the quality of the anonymized datasets is analyzed.

5.2 Test results

A complete overview of all executed experiments is available online⁷. As it is infeasible to include all measurements in the paper, this section compiles the most significant results. All experiments were executed on different sample sizes of the dataset. Various tests exposed similar results. The tables and graphs presented in this section reflect the results from the experiments on a 100K sample dataset. The lv1 column in the tables represent the generalization hierarchy level.

²<https://opendata.bosa.be/index.nl.html>

³<https://batch.openaddresses.io/job/68218>

⁴https://appsso.eurostat.ec.europa.eu/nui/show.do?dataset=demo_pjan

⁵https://www.ine.es/dyngs/INEbase/en/operacion.htm?c=Estadistica_C&cid=1254736176951&idp=1254735572981

⁶<https://www.abs.gov.au/AUSSTATS/abs@.nsf/DetailsPage/3101.0Jun%202019?OpenDocument>

⁷<https://kuleuven.box.com/s/p5g098vlfgn0l5rfg06rmbpg7oy51a7d>

Table 2: Coordinate rounding comparison in Australia.

lvl	Distance to gen. [median in meters]			Paired neighbors [in %]			Group Size [Std. Dev.]		
	CR	KM	HA	CR	KM	HA	CR	KM	HA
	1	3161	627	1342	92,84	90,23	96,92	55	6
2	6270	935	2597	95,95	92,90	98,59	133	11	80
3	12861	1501	5325	97,75	95,27	99,41	321	24	223
4	26488	2617	10837	98,80	97,15	99,77	693	59	660
5	56308	4726	16418	99,37	98,33	99,90	1283	139	1282
6	97399	8767	26764	99,66	99,16	99,96	2946	457	2712
7	182204	23117	43147	99,84	99,67	99,98	5899	2126	5411
8	430964	45748	56409	99,95	99,91	100,00	12898	6790	10578
9	710610	182971	522391	99,98	99,99	100,00	33329	10485	30154

1. *Coordinate rounding is always an inferior option.* Table 2 reflects the generalization hierarchy properties extracted from the first experiment for Australia with coordinate rounding. We aim at minimizing *distance to generalization* and maximizing *group size stability*. The coordinate rounding strategy results in inferior values with respect to both metrics. The median distance from a point to its generalizations is in many cases five times larger in comparison to other strategies. The same conclusions apply for Belgium and Spain. The *neighbor pairing* metric aims at maximizing the highest possible percentage. Experiments with the Australian dataset demonstrate that coordinate rounding scores marginally better than the K-Means strategy with respect to this metric, especially related to lower generalization levels. This discrepancy mainly occurs in the Australian dataset and can be attributed to the fact that the majority of the Australian population is packed together in smaller coastal areas.

2. *The quality of ZIP based hierarchies strongly depends on areal properties.* Table 3 and 4 return the results obtained from the first experiment for Spain and Belgium, and compare various clustering methods to the ZIP code translation technique. Both the *distance to generalization* and the *group size stability* of ZIP codes are similar to the results gathered from the clustering techniques. The results may assume that ZIP code translation is a simple yet effective generalization hierarchy. However, strong differences can be noticed when comparing various countries. Level 5 of the Spain ZIP codes scores significantly worse on both metrics while the Belgium metrics are more consistent throughout the levels. This shows that the effectiveness of ZIP code based hierarchies strongly depends on the assignment of the ZIP codes in a particular country.

3. *K-Means scores best with respect to the distance to generalization and cluster size stability metrics.* The graphs in figures 6 and 7 reflect the conclusions for Australia with respect to the second experiment (which compares the three clustering methods under study). For both the *median distance to generalization* and the *group size stability* metric, K-Means exposes significantly better values compared to the two other clustering methods. This can be attributed to the fact that the top-down approach applied in K-Means clustering heavily focuses on equal distribution of records between the different groups. This also means that large, densely populated

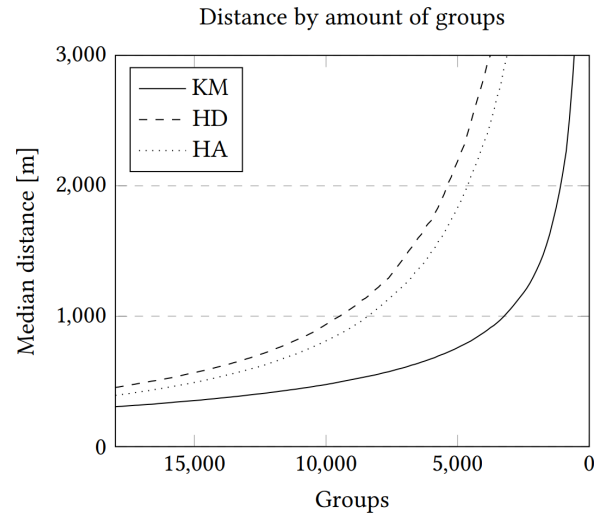
Table 3: ZIP comparison in Spain.

lvl	Distance to generalization [median in meters]				Group Size [Standard Deviation]			
	ZIP	KM	HD	HA	ZIP	KM	HD	HA
	1	835	611	1070	834	13	5	16
2	3628	2024	3714	2994	36	17	56	45
3	14454	9068	13655	11841	201	109	379	294
4	35368	32309	42849	38366	1270	864	2226	1735
5	269333	97793	126718	12245	6937	3196	8367	5581

Table 4: ZIP comparison in Belgium.

lvl	Distance to generalization [median in meters]				Group Size [Standard Deviation]			
	ZIP	KM	HD	HA	ZIP	KM	HD	HA
	1	1445	1144	1816	1474	100	26	146
2	2006	1744	2612	2301	119	65	275	204
3	6698	5589	7400	6750	643	487	1490	1045
4	20186	18560	24058	23023	3723	3463	13013	6911

areas are split in multiple subclusters, which, on its turn, is favorable for the *median distance to generalization* metric. Note that the K-Means approach exposes inferior values for the *paired neighbor* metric as displayed in figure 8.

**Figure 6: Median distance in Australia.**

4. *Agglomerative clustering exposes the best results with respect to the paired neighbor metric.* The graph in figure 8 shows the results of the *paired neighbor* metric for Australia. The hierarchical agglomerative method scores up to 8% better than the other methods for this metric. This is caused by the functioning of the agglomerative clustering method. In generalization level L_0 , each cluster maps to a separate location. The upper levels are created by merging clusters with neighboring clusters. The *complete* linkage function supports

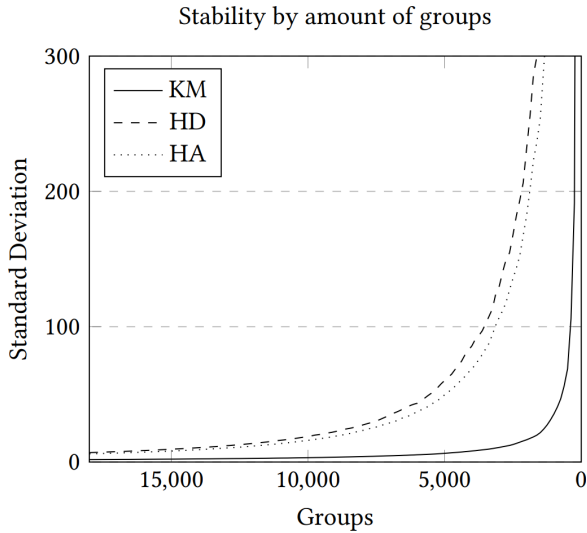


Figure 7: Stability in Australia.

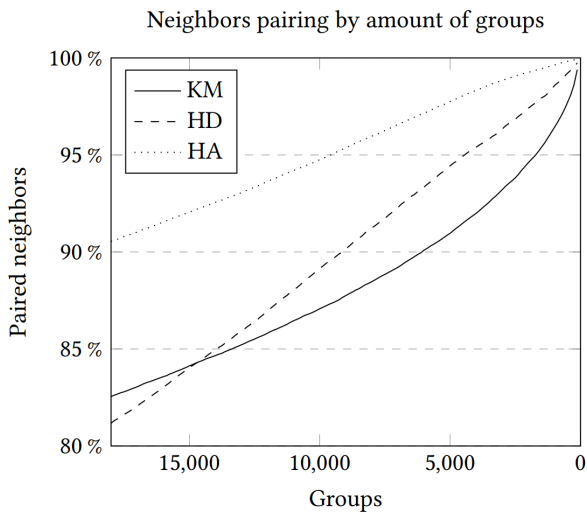


Figure 8: Neighbor pairing in Australia.

the creation of compact clusters. This metric shows that hierarchical agglomerative clustering is an ideal solution for neighbor driven use cases.

5. *Divisive clustering exposes similar behavior as agglomerative clustering, but scores worse.* The graphs in figures 6 and 7 show that the agglomerative and divisive methods are similar for the *distance* and the *stability* metrics. For both metrics, agglomerative clustering outperforms the divisive method. The graph in figure 8 shows that the agglomerative method also scores better with respect to the *neighbor pairing* metric, where the divisive method scores similar to K-means.

6. *Even when the original dataset only contains the ZIP codes and not the exact GPS location, applying clustering techniques is feasible.* Table 5 presents the results of additional tests conducted in this

Table 5: ZIP based clustering in Spain.

lvl	Distance to gen. [median in meters]				Group Size [Std. Dev.]			
	ZIP	KM	HD	HA	ZIP	KM	HD	HA
1	3292	1457	3155	2554	36	19	51	44
2	14400	8923	13098	12043	199	111	353	296
3	35220	32421	44108	37750	1247	880	2334	1539
4	269810	96483	127150	132084	6925	3002	8189	4902

research. In these tests, the original dataset contained only the ZIP code of the location and not the exact coordinates. Four different generalization hierarchies were created. One applied the ZIP masking method; the other three performed the clustering methods. The table only contains values for the *distance to generalization* and the *group size standard deviation*. The *neighbor pairing* parameter was omitted as this parameter is irrelevant if all locations are packed together in a small amount of coordinates in the original dataset.

The table demonstrates that for both metrics, the clustering methods generally perform better in this case than the ZIP strategy. These results are most clear for Spain, which can be attributed to the fact that many neighboring areas in Spain do not necessarily have a similar ZIP code. In Belgium and Australia, the K-Means strategy also outperforms the ZIP masking strategy, but the other clustering methods score worse.

7. *With respect to the k-anonymity metric, K-means achieves the most favorable results.* Table 6 summarizes the results for Belgium with respect to experiment 3, which was executed for five different *k* values. For each location hierarchy creation method, the table shows the *generalization level*, the amount of suppressed records, the amount of equivalence classes and the average equivalence class size. The *generalization level* is represented by a tuple of three values reflecting the location generalization level, gender and age respectively. For each *k* value the best result is shown in **bold**. This table shows a lot of variation between the different values of *k* for which location generalization strategy scores outstanding. However, in contrast to strategies such as ZIP code translation and coordinate rounding which demonstrate heavy fluctuations – sometimes the best, sometimes the absolute worst solution –, the results for K-Means are very stable. While K-Means does not always expose a superior score for each parameter, its parameters are always relatively close to the best score. The results for the Spanish and Australian datasets are even more favorable for the K-means strategy.

8. *The selected generalization levels and created equivalence classes reflect only partial results.* It is also relevant to consider the actual information in the anonymized datasets. For this purpose, the *distance to generalization* column was added to Table 6. This column returns the median distance between the records after anonymization and the original records, not counting the suppressed records. Also in this metric, the K-means method expose superior scores. It results in the lowest median distance in three out of the five cases, and approximates the best score with respect to the other two *k* values. Similar results are valid for Spain and Australia. In these

cases, the ZIP code method scores worse for the Spanish dataset, and the coordinate rounding method scores significantly lower in Australia.

6 EVALUATION AND DISCUSSION

The conducted experiments show that the clustering approach for creating location generalization hierarchies provides a feasible – and often preferable – alternative compared to the traditional approaches. This section extrapolates the results and provides guidelines for developers in charge of the anonymization of location data.

The most suitable location generalization strategy depends on areal properties. A lot of parameters have an impact on the outcome of a certain location generalization strategy. Firstly, the shape of an area can negatively influence the performance of a location generalization strategy. This is above all the case for the coordinate rounding method because records located in remote corners near the country’s border can disregard records in groups that are too small for further use. Moreover, the demographical spread of the population in an area can have an impact. This paper has demonstrated that different techniques lead to different results for the Australian map on the one hand, and the ones where the population is more evenly spread on the other hand (like Belgium and Spain). In areas with an unevenly spread population, it is recommended to apply a clustering technique instead of *coordinate rounding*. Lastly, the scale also has a major impact on the optimal generalization strategy. The experiments conducted in this paper were all scoped to one country. Upscaling to multiple countries, continents or even the world has consequences. For instance, masking ZIP codes is no longer feasible. Overlap can occur between ZIP codes of different countries, and formats can differ. In order to work with ZIP codes, the technique described in experiment result 6 can be applied. The coordinate rounding method and the K-means clustering method also become less feasible as they introduce an artificial border on the 180th meridian. This would make it impossible to pair for example Alaskan and eastern Russian records, even if they are relatively close to each other. The divisive and agglomerative clustering methods described in this paper apply goniometric formulas to calculate the distance between two points and are therefore not impacted by this phenomenon. Downsizing the scale to one or multiple cities instead of a whole country also impacts the selection procedure. First of all, ZIP codes become unfeasible as one city only contains one or a few ZIP codes. It should also be noted that smaller areas probably also result in smaller datasets. When applying the clustering methods, the group sizes for each generalization level should scale along with the dataset size.

The purpose of the data can influence the optimal hierarchy creation strategy. First of all, while our test results demonstrate that it is advisable to apply one of the proposed clustering strategies, use cases exist that rely on zip codes (e.g. post delivery). In these instances, applying the ZIP code strategy is still preferred.

When the decision is made to apply a clustering strategy, the selection is also steered by the purpose of the dataset. If the data will be employed in use cases where the precise location is of major importance, it is advised to apply the K-Means strategy, as the

latter has proven to result in the lowest average distance between a location and its generalization. If, however, the purpose of the dataset requires strong similarity between closely related records, it is advisable to apply the agglomerative strategy, as this strategy performs outstanding in grouping the nearest neighbors together.

Guidelines for using the clustering approach. Caution has to be taken when applying clustering algorithms to location generalization. The quality of the formed clusters can be impacted by the requested group sizes. Small decreases in size between two generalization levels cause unevenly formed clusters, especially when using the K-Means algorithm. Therefore, halving the number of requested clusters each level is recommended.

The quality of the generalization hierarchies can be increased by working in two phases. In a first phase, a large range of groups can be requested and fed through an anonymization tool. Afterwards, a smaller range can be defined based on the generalization level selected by the anonymization tool in the first phase.

The clustering approach results in favorable privacy properties. An interesting side-effect of applying the clustering techniques instead of the traditional location generalization methods is that the anonymity level increases. When applying a coordinate rounding method, a rectangle can be drawn on a map in which all records contained by that generalization are located. This also applies to ZIP code translation (and masking of ZIP codes), where it is also possible to draw the exact area on a map that contains all records of that specific equivalence class. This is not the case when applying one of the clustering techniques proposed in this work. Once the location data is generalized, it is impossible to exactly reconstruct the borders of each cluster. This makes execution of background knowledge attacks significantly harder as more uncertainty is introduced, and no trivial mapping between locations in two different datasets can be made.

A note on performance. The different hierarchy generation methods have various performance properties. Utilizing the traditional ZIP and coordinate rounding techniques requires almost no extra memory or calculation time. Most computing time is spent on the calculation of all the ZIP centers. However, this is a preprocessing step. While these methods are the most easy ones to perform, our research shows that they often do not result in the most favorable results. The hierarchical algorithms require a pre-calculated geodesic distance matrix between all the points. This matrix scales with the size of the dataset squared (N^2), causing a big inflation in memory usage. However, its size can be halved by removing all duplicate values in the symmetric matrix resulting in a memory usage of 37GB for 100 000 records. The agglomerative clustering method uses more memory (up to 280GB for 100 000 records) improving the calculation time, this can be avoided by creating a proper implementation of the algorithm. The K-Means algorithm is the fastest and most memory efficient clustering algorithm. The algorithm uses approximately 60% of the calculation time needed for the agglomerative method and even less when compared to the divisive method. This computational efficiency is partially caused by utilizing the minibatch K-Means implementation.

Table 6: ARX results for Belgium.

K	Generalization level				#suppressed				Avg. class size				#Eqv. Classes				Distance to generalization			
	ZIP	CR	KM	HA	ZIP	CR	KM	HA	ZIP	CR	KM	HA	ZIP	CR	KM	HA	ZIP	CR	KM	HA
5	3,0,1	6,0,1	1,0,1	2,0,1	496	1472	582	290	34	37	27	54	2904	2647	3636	1835	6716	8223	5107	8547
10	3,0,1	7,0,1	2,0,1	2,0,1	2213	603	627	1418	37	118	55	59	2656	845	1819	1679	6718	16564	7531	8536
20	4,0,1	7,0,1	3,0,1	3,0,1	156	1732	732	1908	292	128	110	124	342	770	905	791	20144	16569	11302	13260
50	4,0,1	7,0,2	4,0,1	4,0,1	641	3346	674	1020	304	274	273	281	327	353	364	352	20136	16580	17463	20213
100	4,0,1	8,0,2	4,0,2	4,0,2	2024	739	905	1223	317	782	547	564	309	127	181	175	20111	31278	17461	20211

7 CONCLUSIONS

This paper presented an alternative approach for creating generalization hierarchies. Three clustering strategies were proposed, namely K-means, agglomerative clustering and divisive clustering. Our experiments expose favorable results for the presented clustering methods compared to the traditional strategies, both with respect to the remaining utility as well as the anonymity level. This effect is achieved due to multiple factors. Firstly, our approach applies a more intelligent strategy to group locations. This results in more equally distributed generalization hierarchies and at the same time decreases the amount of records suppressed during the anonymization process. Secondly, in contrast to the traditional methods, the clustering approaches enable users to fix hierarchy level sizes. This increases the flexibility and also allows for intelligent tuning after an initial anonymization step. Lastly, we argue that, depending on the purpose of the data, alternative methods can be applied in order to maximize the utility of the anonymized data. Future research will investigate the effect of including additional location metadata in the clustering process. For example, an attribute that distinguishes rural and urban regions could potentially further increase the utility of the resulting anonymized dataset.

REFERENCES

- [1] Sri Krishna Adusumalli and V Valli Kumari. 2013. An efficient and dynamic concept hierarchy generation for data anonymization. In *International Conference on Distributed Computing and Internet Technology*. Springer, 488–499.
- [2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 901–914.
- [3] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.
- [4] David Arthur and Sergei Vassilvitskii. 2006. *k-means++: The advantages of careful seeding*. Technical Report. Stanford.
- [5] Vanessa Ayala-Rivera, Patrick McDonagh, Thomas Cerqueus, and Liam Murphy. 2015. Ontology-based quality evaluation of value generalization hierarchies for data anonymization. *arXiv preprint arXiv:1503.01812* (2015).
- [6] Vanessa Ayala-Rivera, Patrick McDonagh, Thomas Cerqueus, Liam Murphy, et al. 2014. A systematic comparison and evaluation of k-anonymization algorithms for practitioners. *Transactions on data privacy* 7, 3 (2014), 337–370.
- [7] Vanessa Ayala-Rivera, Patrick McDonagh, Thomas Cerqueus, Liam Murphy, Christina Thorpe, et al. 2017. Enhancing the utility of anonymized data by improving the quality of generalization hierarchies. *Transactions on Data Privacy* 10, 1 (2017), 27–59.
- [8] Ji-Won Byun, Ashish Kamra, Elisa Bertino, and Ninghui Li. 2007. Efficient k-anonymization using clustering techniques. In *International Conference on Database Systems for Advanced Applications*. Springer, 188–200.
- [9] Alina Campan, Nicholas Cooper, and Traian Marius Truta. 2011. On-the-fly generalization hierarchies for numerical attributes revisited. In *Workshop on Secure Data Management*. Springer, 18–32.
- [10] Yves-Alexandre de Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex “Sandy” Pentland. 2015. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science* 347, 6221 (2015), 536–539. <https://doi.org/10.1126/science.1256297> arXiv:<https://science.sciencemag.org/content/347/6221/536.full.pdf>
- [11] M. Douriez, H. Doraiswamy, J. Freire, and C. T. Silva. 2016. Anonymizing NYC Taxi Data: Does It Matter?. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 140–148. <https://doi.org/10.1109/DSAA.2016.21>
- [12] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise.. In *Kdd*, Vol. 96. 226–231.
- [13] Bugra Gedik and Ling Liu. 2005. Location privacy in mobile systems: A personalized anonymization model. In *25th IEEE International Conference on Distributed Computing Systems (ICDCS’05)*. IEEE, 620–629.
- [14] Aris Gkoulalas-Divanis, Panos Kalnis, and Vassilios S Verykios. 2010. Providing k-anonymity in location based services. *ACM SIGKDD explorations newsletter* 12, 1 (2010), 3–10.
- [15] Wajih Ul Hassan, Saad Hussain, and Adam Bates. 2018. Analysis of Privacy Protections in Fitness Tracking Social Networks-or-You can run, but can you hide?. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 497–512.
- [16] Leonard Kaufman and Peter J Rousseeuw. 2009. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons.
- [17] Jun-Lin Lin and Meng-Cheng Wei. 2008. An efficient clustering method for k-anonymization. In *Proceedings of the 2008 international workshop on Privacy and anonymity in information society*. 46–50.
- [18] Daniel Müllner. 2011. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378* (2011).
- [19] Sang Ni, Mengbo Xie, and Quan Qian. 2017. Clustering Based K-anonymity Algorithm for Privacy Preservation. *IJ Network Security* 19, 6 (2017), 1062–1071.
- [20] Fabian Prasser and Florian Kohlmayer. 2015. Putting statistical disclosure control into practice: The ARX data anonymization tool. In *Medical Data Privacy Handbook*. Springer, 111–148.
- [21] David Sculley. 2010. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*. 1177–1178.
- [22] Latanya Sweeney. 2002. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10, 05 (2002), 571–588.
- [23] Ge Zhong and Urs Hengartner. 2008. Toward a distributed k-anonymity protocol for location privacy. In *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society*. 33–38.