Privacy Guarantees in Geolocation Services

Marco da Silva Teles Coelho

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisor: Prof. Pedro Miguel dos Santos Alves Madeira Adão

Examination Committee

Chairperson: Prof. Francisco João Duarte Cordeiro Correia dos Santos
Supervisor: Prof. Pedro Miguel dos Santos Alves Madeira Adão
Members of the Committee: Prof. Rui Miguel Carrasqueiro Henriques

October 2019
Dedicated to my family
Acknowledgments

I want to thank my thesis advisor Professor Pedro Miguel Adão for this opportunity, for all the meetings and for advising me over the year, without him this thesis wouldn’t be possible.

Also, I want to thanks to my parents and friends for being present, for the motivation and support throughout this tough year and in the previous ones.

Finally, I want to thank to all the teachers who gave me knowledge in this course over the years and to the college Instituto Superior Técnico. Thank you all.
Abstract

Nowadays a lot of people use location-based services (LBS) to get information for example: the nearest restaurant, allowing to post a recommendation on the social network about that place making it useful for everyone. That type of content inside of a LBS is a gold mine for companies because with that they can learn, using pattern mining techniques, favorite places, frequent user trajectories, etc. exposing home location or any other type of private detail, which raises concerns about location privacy. It is a challenge to create a mechanism capable of having an excellent balance between privacy protection and utility of the data against attacks (e.g. probabilistic inference attack, background attack, etc.). The anonymization mechanism AdaTrace [13] seems to have the robustness to handle that challenge. So, with the help of AdaTrace components and our tool that identifies the best risk areas on the map to expose points of a trajectory that are consider as important and it also has a method that distributes privacy (noise) across the sub-paths of the trajectory. All of this allows to publish content that is more useful with a significant anonymization component to prevent them from discovering clues, through places that people have passed along the trajectory, where one can discover personal information knowing what kind of person is, for example, someone with reduced mobility chooses a route without stairs.

Keywords

location-based services, location privacy, utility, trajectories.
Resumo

Atualmente, muitas pessoas usam serviços baseados em localização (LBS) para obterem informações, por exemplo: o restaurante mais próximo, podendo publicar uma recomendação na rede social sobre esse local tornando-se útil para todos. O conteúdo de um LBS é uma mina de ouro para as empresas, porque com isso elas podem aprender, usando técnicas de pattern mining, os lugares favoritos, trajetórias frequentes dos utilizadores, etc. expondo a localização da residência ou qualquer outro tipo de informação privada, o que levanta preocupações sobre a privacidade da localização. É um desafio criar um mecanismo capaz de ter um excelente equilíbrio entre a proteção da privacidade e a utilidade dos dados contra ataque, tais como ataque de inferência probabilística, ataque com conhecimento, etc. O mecanismo de anonimização AdaTrace [13] parece ter a robustez para lidar com esse desafio. Assim e com a ajuda das componentes do AdaTrace e da nossa ferramenta que identifica as melhores áreas de risco num mapa para expor pontos de um caminho que considere importante e que também contem um método que distribui privacidade (ruído) nos segmentos da trajetória. Tudo isto, permite publicar conteúdo com mais utilidade com uma componente de anonimização significativa para evitar que se descubram pistas, através de locais que as pessoas tenham passado ao longo do caminho e no qual permita compreender informações pessoais sabendo que tipo de pessoa se trata, por exemplo: alguém com mobilidade reduzida escolhe um percurso sem escadas.

Palavras Chave

Serviços baseados em localização, privacidade na localização, utilidade, trajetórias
Contents

1 Introduction .......................................................... 1
   1.1 Thesis Proposal .................................................. 1
   1.2 Outline for the Dissertation .................................. 2

2 Related Work ....................................................... 3
   2.1 Types of anonymization ....................................... 3
       2.1.1 K-anonymity .............................................. 3
       2.1.2 L-diversity .............................................. 6
       2.1.3 T-closeness .............................................. 8
       2.1.4 ε-Differential Privacy ................................ 9
   2.2 Difficulties in geolocation ................................... 11
   2.3 Location privacy techniques .................................. 11
       2.3.1 Geo-indistinguishability ................................ 11
       2.3.2 AdaTrace .................................................. 14

3 Dataset characterization ........................................... 20
   3.1 Trajectory Records ............................................ 20

4 Proposed solution .................................................. 22
   4.1 Solution proposal overview ................................... 22
   4.2 Choosing points from trajectories ........................... 23
   4.3 Anonymization .................................................. 23

5 Evaluation Results ................................................ 30
   5.1 Machine characteristics ....................................... 30
   5.2 Evaluation Metrics ............................................ 30
       5.2.1 Query AvRE .............................................. 30
       5.2.2 Frequent pattern AvRE .................................. 31
       5.2.3 Frequent pattern Similarity ............................. 31
       5.2.4 Length error .............................................. 32
       5.2.5 Kendall-tau coefficient ................................ 33
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>Evaluating the 4 methods</td>
<td>34</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Noise replication results</td>
<td>34</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Noise division results</td>
<td>35</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Uniform noise results</td>
<td>36</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Crescent uniform noise results</td>
<td>37</td>
</tr>
<tr>
<td>5.3.5</td>
<td>Comparing methods results</td>
<td>37</td>
</tr>
<tr>
<td>5.4</td>
<td>Attack resilience</td>
<td>42</td>
</tr>
<tr>
<td>5.5</td>
<td>Our solution utility vs AdaTrace utility</td>
<td>48</td>
</tr>
<tr>
<td>6</td>
<td>Conclusion</td>
<td>51</td>
</tr>
<tr>
<td>6.1</td>
<td>Main Contributions</td>
<td>51</td>
</tr>
<tr>
<td>6.2</td>
<td>Future Work</td>
<td>52</td>
</tr>
</tbody>
</table>

**Bibliography** | 54 |

**Appendix A** | 57 |

**Appendix B** | 60 |
List of Figures

Figure 2.1: Table result with generalization of a 2-anonymity ........................................... 4
Figure 2.2: Table result of a 3-diversity .................................................................................... 7
Figure 2.3: Table result after applying Anatomy ....................................................................... 8
Figure 2.4: Trade-off between privacy and data utility ................................................................. 10
Figure 2.5: Partial Sniffing Attack ............................................................................................. 13
Figure 2.6: Architecture of AdaTrace ......................................................................................... 14
Figure 2.7: Density-Aware Grid transformation with N=2 .......................................................... 17
Figure 3.1: Example of a trajectory ............................................................................................ 20
Figure 3.2: Example of trajectories in the database ................................................................. 20
Figure 4.1: Example of anonymized trajectory from Figure 3.1 ................................................. 23
Figure 4.2: Example of points in the grid .................................................................................. 24
Figure 5.1: Evaluation mean results with different noise values from Brinkhoff database .......... 38
Figure 5.2: Evaluation results with different noise values from Taxi database ......................... 40
Figure 5.3: Time execution using different methods with different noise values from Brinkhoff database ......................................................................................................................... 41
Figure 5.4: Time execution using different methods with different noise values from Taxi database ........................................................................................................................................ 42
Figure 5.5: Partial Sniffing attack in both databases for 1 exposed point .................................. 43
Figure 5.6: Partial Sniffing attack in both databases for multiple exposed points ..................... 43
Figure 5.7: Bayesian Inference attack in both databases ............................................................. 44
Figure 5.8: Risk areas in the map for the Bayesian Inference attack in Brinkhoff database ......... 45
Figure 5.9: Risk areas in the map for the Bayesian Inference attack in Taxi database ............... 45
Figure 5.10: View of the Porto city map .................................................................................... 46
Figure 5.11: EMD value over the increase of exposing important points in both databases .......... 47
Figure 5.12: Variation of EMD value when using different solutions in (a) Brinkhoff database and (b) Taxi database ................................................................................................................. 48
Figure 5.13: Evaluation results with $\varepsilon = 1$ for the (a) Brinkhoff database and for (b) Taxi database 49
Figure 5.14: Evaluation results with $\varepsilon = 1.5$ for the (a) Brinkhoff database and for (b) Taxi database .................................................................................................................................................. 49
Figure 5.15: Evaluation results with $\varepsilon = 2$ for the (a) Brinkhoff database and for (b) Taxi database 50
List of Tables

Table 2.1: Private table (PT) ........................................................................................................4
Table 2.2: Anonymized table (AT1) ............................................................................................5
Table 2.3: Anonymized table (AT2) ............................................................................................5
Table 2.4: Anonymized table (AT3) ............................................................................................5
Table 2.5: Anonymized table (AT4) ............................................................................................5
Table 3.1: Analyze of Brinkhoff and Taxi database .................................................................21
Table 5.1: Evaluation results of noise replication with different noise values ......................35
Table 5.2: Evaluation results of noise division with different noise values ...........................36
Table 5.3: Evaluation results of uniform noise with different noise values .........................36
Table 5.4: Evaluation results of crescent uniform noise with different noise values ..........37
Acronyms

AT Anonymized table
AvRE Average Relative Error
DTW Dynamic Time Warping
EMD Earth Mover’s distance
FN False Negative
FP False positive
FrP Frequent Pattern
GPS Global Positioning System
JSD Jensen-Shannon Divergence
KLD Kullback-Leibler Divergence
LBS Location-based service
PT Private table
QI Quasi-identifier
QIT Quasi-identifier table
ST Sensitive table
TP True positive
1 Introduction

In a modern world, people use location-based services that uses geographical coordinates, longitude and latitude. This geographical information is useful because it provides recommendations to the user about points of interest based on their location.

One of the major concerns about location-based applications (such as Google and Uber) is location privacy, which can be used by companies to learn users’ movement patterns [20] by exposing their home location, frequent visiting points or some other private detail.

Privacy preservation techniques are an important topic to consider, so it modifies an individual’s information so that an attacker cannot identify an individual or learn new information from the released data. This process is called anonymization. Here there is always a tradeoff between privacy and utility which makes the analysis and publishing of sensitive data a challenge.

Many anonymization mechanisms have been proposed during the last decade to provide the best privacy-utility data, such as k-anonymity [2] which makes a record of an individual to be within a group that has at least k-1 other records equal to his, and consequently the probability of re-identifying data is 1/k, the higher the k, the lower is the risk of re-identification. L-diversity [3] makes sure that in each group there are at least L different sensitive values. t-closeness [8] aims to create groups with similarity no more than t. However, all these mechanisms are susceptible to attacks, for example background attacks, probabilistic inference attacks, unsorted matching attacks and many other attacks. For that, a stronger technique was proposed, Differential privacy [21] which adds a controlled noise to the true value but it is important to refer that too much noise makes data less useful.

Differential privacy has been accepted as a standard for privacy preservation. It was originally proposed to protect data privacy of a database but not for location privacy.

There are many mechanisms to protect location privacy, one example is Geo-indistinguishability [12] that uses the principle idea of Differential privacy, which it can offer a very good probabilistic defense, for protecting the user location within a radius but Geo-indistinguishability is susceptible to temporal correlations of multiple locations.

1.1 Thesis Proposal

An interesting mechanism has recently been published which is called AdaTrace [13] that also use the main idea of Differential privacy for his four components to have robust statistical resilience. Those four components (Density-Aware Grid, Markov Chain Mobility Model, Trip Distribution and Route Length Distribution) are the main key for their Trajectory Synthesis algorithm (that produces the synthetic trajectories) to create an anonymous database that has attack resilience against an attacker in case he has accesses to that database and tries to discover personal information through the locations a person
has passed and was left along his or her trajectory, for example: a person with reduced mobility does not make a trajectory that has stairs or rocky pavements.

With the combination of AdaTrace and our tool, in which we have proposed and analyze 4 different methods to anonymize a trajectory were the best is the Crescent uniform noise method and develop an algorithm that identifies the risk of vulnerability in a given area of the map, it is possible to publish a dataset that have more information (by exposing points from the trajectory that the user considers important) and at the same time have a significant anonymization component.

The challenge of this work is to develop a system that can produce a better utility but compromises as little privacy as possible because when we want to increase one of them the other decreases.

1.2 Outline for the Dissertation

The organization of the content of this thesis is the following:

Section 2 begins with Related work, where is described the types of anonymization and their advantages and disadvantages, including a description of the anonymization technique AdaTrace that we will later use.

In section 3, we analyze trajectory databases.

Next in section 4 is described in detailed: our solution, the different methods that we use: **Noise replication, Noise division, Uniform noise and Crescent uniform noise**. From those four we want to discover the one that produces the most utility. Then we present maps that are identified by different colors the locations with low and high risk of an attack. All of this make us reach our goal.

In section 5, we present the metrics that evaluates the precious methods and compare our evaluation results with the ones obtain from AdaTrace. We also do attacks resilience’s tests to our solution to find how much vulnerable the anonymous trajectory content is.

Finally, section 6 summarizes the contribution of this dissertation and presents hints for future work.
2 Related Work

This section describes the main types anonymization mechanisms, such as k-anonymity [2], L-diversity [3], t-closeness [8] and Differential privacy [21]. Each one of them are an improvement of the previous, in which we explain in detail the advantages and the disadvantages of them and the originally difficulties in applying those basic techniques for location protection because they were created for database privacy.

I will also reference 2 more types of location privacy that use an adaptation of Differential privacy (which has been accepted as a standard for privacy preservation): Geo-indistinguishability [12] and AdaTrace [13].

In my opinion AdaTrace seems to be the most robust and complete mechanism that exists.

2.1 Types of anonymization

A structured data contains various elements such as: Identifier attributes are those that identify an individual from the data, for example: names; Quasi-identifiers (QI) are a set of attributes values that together may be used to re-identify a single or a group of individuals, for example: Gender, age, zip code; and Sensitive attributes are the private details of the individual that must be kept secure, for example: health condition.

2.1.1 K-anonymity

The goal of k-anonymity [2] is to protect from identity disclosure by guarantying that each sequence of quasi-identifier attributes appears at least k times in the table, which makes the probability of re-identification equals to 1/k.

There are two common methods for k-anonymity that are used to eliminate potential risks of identification of persons through the quasi-identifiers in a database table:

- **Suppression**: Replace 1 or more values of the quasi-identifiers columns with a star (*)
- **Generalization**: Generalize the values of the quasi-identifiers columns, for example: the value “07029” and “07028” of the attribute “Zip code” both can be replaced by an interval.
Figure 2.1 from [15] below shows the result with generalization of a 2-anonimity of a table. If it was with suppression the column of birth could have 19* and would still be 2-anonimity. As you can see there are at least 2 sequences of Quasi-identifiers in the final table.

Using generalization with suppression can limit the amount of generalization used on the table. K-
minimal generalization with suppression [1] tries to find the minimal amount of generalization needed with a given maximum suppression threshold allowed, this definition is based on the concept of distance vector.

Types of attacks

Even if we identify carefully all the QI (Quasi-identifiers), attacks to k-anonimity can still occur. I will mention some of them. Consider Table 2.1, K=2 and QI={(Race, Zip)} as an example for the attacks.
a. **Unsorted matching attack [2]**

<table>
<thead>
<tr>
<th>Race</th>
<th>Zip</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>02138</td>
<td>Cancer</td>
</tr>
<tr>
<td>Person</td>
<td>02139</td>
<td>Flu</td>
</tr>
<tr>
<td>Person</td>
<td>02138</td>
<td>Gastritis</td>
</tr>
<tr>
<td>Person</td>
<td>02138</td>
<td>Pneumonia</td>
</tr>
<tr>
<td>Person</td>
<td>02139</td>
<td>Bronchitis</td>
</tr>
<tr>
<td>Person</td>
<td>02138</td>
<td>Dyspepsia</td>
</tr>
</tbody>
</table>

Table 2.2: Anonymized table (AT1)

<table>
<thead>
<tr>
<th>Race</th>
<th>Zip</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asian</td>
<td>0213*</td>
<td>Cancer</td>
</tr>
<tr>
<td>White</td>
<td>0213*</td>
<td>Flu</td>
</tr>
<tr>
<td>White</td>
<td>0213*</td>
<td>Gastritis</td>
</tr>
<tr>
<td>Asian</td>
<td>0213*</td>
<td>Pneumonia</td>
</tr>
<tr>
<td>White</td>
<td>0213*</td>
<td>Bronchitis</td>
</tr>
<tr>
<td>Asian</td>
<td>0213*</td>
<td>Dyspepsia</td>
</tr>
</tbody>
</table>

Table 2.3: Anonymized table (AT2)

This attack is based on the order in which tuples appear in the released table, if the two 2-anonymity tables from PT, AT1 and AT2, are not randomly sorted then it is easy to relate each entry of the tables revealing sensitive information, for example: the entry {\{(Person, 02138)\}} in AT1 is the same as the entry {\{(Asian, 0213*)\}} in AT2 and so on. With this 2 tables, it is easy to determine the PT.

b. **Temporal attack [2]**

Since tuples are always changing because data collections are dynamic, over time the anonymity of the table can be broken, for example: supposed a 2-anonymity table from PT, AT3, is published and then it is added in PT these two entries {\{(Indian, 02138, Rash), (Indian, 02138, Headache)\}} resulting a 2-anonymity table AT4. Then even with random tuple positions it is still possible to link AT3 and AT4 on \{Condition\}. This would not have happened if those new entries were added to AT3 instead of PT.

<table>
<thead>
<tr>
<th>Race</th>
<th>Zip</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>02138</td>
<td>Cancer</td>
</tr>
<tr>
<td>White</td>
<td>0213*</td>
<td>Flu</td>
</tr>
<tr>
<td>Person</td>
<td>02138</td>
<td>Gastritis</td>
</tr>
<tr>
<td>Person</td>
<td>02138</td>
<td>Pneumonia</td>
</tr>
<tr>
<td>White</td>
<td>0213*</td>
<td>Bronchitis</td>
</tr>
<tr>
<td>Person</td>
<td>02138</td>
<td>Dyspepsia</td>
</tr>
</tbody>
</table>

Table 2.4: Anonymized table (AT3)

<table>
<thead>
<tr>
<th>Race</th>
<th>Zip</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asian</td>
<td>0213*</td>
<td>Cancer</td>
</tr>
<tr>
<td>White</td>
<td>0213*</td>
<td>Flu</td>
</tr>
<tr>
<td>White</td>
<td>0213*</td>
<td>Gastritis</td>
</tr>
<tr>
<td>Asian</td>
<td>0213*</td>
<td>Pneumonia</td>
</tr>
<tr>
<td>White</td>
<td>0213*</td>
<td>Bronchitis</td>
</tr>
<tr>
<td>Asian</td>
<td>0213*</td>
<td>Dyspepsia</td>
</tr>
<tr>
<td>Indian</td>
<td>02138</td>
<td>Rash</td>
</tr>
<tr>
<td>Indian</td>
<td>02138</td>
<td>Headache</td>
</tr>
</tbody>
</table>

Table 2.5: Anonymized table (AT4)
c. **Homogeneity Attack [3]**

This only works if all sensitive attributes values inside of a group of tuples have the same values. Even though the data has been k-anonymized, using the example of Figure 2.1 we can see that a Male with a range year of birth between 1970-1979 (record ID 9 and 10) satisfies 2-anonimity but the sensitive information (Test result) is the same, which means that if Alice knew that Bod was born in that range year and he has done the test, she would know that the test result came out negative because all the results are the same. The solution in this case is to use L-diversity.

d. **Background Attack [3]**

Is when we have extra background knowledge about some information and we use that to reduce possible combinations entries of the table, for example: suppose AT1 is released and the attacker knows that the Zip code of Bob is 02139 then he can infer that Bob either has a Flu or Bronchitis.

### 2.1.2 L-diversity

Machanavajjhala et al. [3] showed that the degree of privacy protection of k-anonymity decays with the number of distinct sensitive values in each equivalence classes, the higher the number the more private it is, for example: in Table 2.5 the 2 entries that are highlighted have 2 distinct "Condition" values (Rash, Headache) which it is a 50/50 guess but if the values were the same, it would be easy to find the “Condition” of a person using Homogeneity Attack. So, L-diversity [3] was created to prevent homogeneity attacks, however it cannot prevent attribute disclosure if the sensitive data in equivalence class are correlated.

The principle behind is that all the equivalence class should have at least L different sensitive attribute values in them.

In Figure 2.2 (Image obtain from [3]) we see that each equivalence classes have at least 3 different types of sensitive information.
Types of attacks

- If the values of attributes are not unevenly distributed, L-diversity is subject to probabilistic inference attacks [4]. For example, if we use the same columns from Figure 2.1 and if we have non-sensitive information \{(Male), (Male), (Male)} and the sensitive information associated was \{(+ve), (-ve), (+ve)}}, we conclude it is more likely that Bob had a positive test result.

- L-diversity does not work well with multiple sensitive attributes, [5] and [6] attempt to directly use it, which result in a lot of information loss. A solution to that would be from all the sensitive attributes, we treat one as the sole sensitive attribute and the rest plus the nonsensitive attributes are treated as quasi-identifier. [3]

Anatomy [7] is a new method that works better than generalization. It significantly outperforms generalization, in both effectiveness of data analysis and computation cost (tuples with identical QI always need to end up in the same QI group for the probabilities of the two levels be always equivalent). Anatomy creates 2 tables: Quasi-identifier table (QIT) and sensitive table (ST). In QIT there are only Quasi-identifiers values and in ST there are only sensitive data. It preserves privacy because the QIT does not link the sensitive value of any tuple, which is randomly guessed from the ST (Figure 2.3) – Image obtain from [7].

<table>
<thead>
<tr>
<th>ID</th>
<th>Zip Code</th>
<th>Age</th>
<th>Nationality</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13053</td>
<td>28</td>
<td>Russian</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>2</td>
<td>13068</td>
<td>29</td>
<td>American</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>3</td>
<td>13068</td>
<td>21</td>
<td>Japanese</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>4</td>
<td>13053</td>
<td>23</td>
<td>American</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>5</td>
<td>14853</td>
<td>50</td>
<td>Indian</td>
<td>Cancer</td>
</tr>
<tr>
<td>6</td>
<td>14853</td>
<td>55</td>
<td>Russian</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>7</td>
<td>14850</td>
<td>47</td>
<td>American</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>8</td>
<td>14850</td>
<td>49</td>
<td>American</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>9</td>
<td>13053</td>
<td>31</td>
<td>American</td>
<td>Cancer</td>
</tr>
<tr>
<td>10</td>
<td>13053</td>
<td>37</td>
<td>Indian</td>
<td>Cancer</td>
</tr>
<tr>
<td>11</td>
<td>13068</td>
<td>36</td>
<td>Japanese</td>
<td>Cancer</td>
</tr>
<tr>
<td>12</td>
<td>13068</td>
<td>35</td>
<td>American</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Zip Code</th>
<th>Age</th>
<th>Nationality</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1305*</td>
<td>≤ 40</td>
<td>*</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>4</td>
<td>1305*</td>
<td>≤ 40</td>
<td>*</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>9</td>
<td>1305*</td>
<td>≤ 40</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>10</td>
<td>1305*</td>
<td>≤ 40</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>5</td>
<td>1485*</td>
<td>&gt; 40</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>6</td>
<td>1485*</td>
<td>&gt; 40</td>
<td>*</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>7</td>
<td>1485*</td>
<td>&gt; 40</td>
<td>*</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>8</td>
<td>1485*</td>
<td>&gt; 40</td>
<td>*</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>2</td>
<td>1306*</td>
<td>&gt; 40</td>
<td>*</td>
<td>Heart Disease</td>
</tr>
<tr>
<td>3</td>
<td>1306*</td>
<td>≤ 40</td>
<td>*</td>
<td>Viral Infection</td>
</tr>
<tr>
<td>11</td>
<td>1306*</td>
<td>≤ 40</td>
<td>*</td>
<td>Cancer</td>
</tr>
<tr>
<td>12</td>
<td>1306*</td>
<td>≤ 40</td>
<td>*</td>
<td>Cancer</td>
</tr>
</tbody>
</table>

Figure 2.2: Table result of a 3-diversity
2.1.3 T-closeness

T-closeness [8] is an improvement of L-diversity because it reduces the correlation between the quasi-identifier attributes and the sensitive attributes. Using Figure 2.2 as an example: if one of the classes has these 3 conditions (Flu, Pneumonia, Bronchitis) it would be easy for an attacker to infer (based on QI attributes) that a person has Respiratory infection because those 3 conditions are related. T-closeness makes sure that the distance between the distribution of a sensitive attribute in a class does not differ from the overall distribution of the attribute in the whole table more than T.

The best way to calculate that distance is to use Earth Mover’s distance (EMD) [16] which calculates the minimum amount of work needed to transform one distribution into another one.

Let P and Q be two distributions of sensitive attributes where P contains all table groups of that attribute and Q contains all table values of the same attribute.

The distance calculation changes according to whether it is for numeric or categorical attributes.

For numerical attributes - The values in the domain are ordered and the size of each groups is always equal [8]. Let m be the size of Q and y be the size of one group, so the distance between two values is based on

\[
D[P, Q] = \frac{1}{m} \frac{1}{m-1} \sum_{i=1}^{y} \sum_{j=1+y(i-1)}^{y} \left| p_i - q_j \right| .
\]  

(2.1)

After calculating each distance, we choose the one with minimal cost.

<table>
<thead>
<tr>
<th>ID</th>
<th>Age</th>
<th>Sex</th>
<th>Zip Code</th>
<th>Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>23</td>
<td>M</td>
<td>11000</td>
<td>Pneumonia</td>
</tr>
<tr>
<td>2</td>
<td>27</td>
<td>M</td>
<td>13000</td>
<td>Dyspepsia</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td>M</td>
<td>59000</td>
<td>Dyspepsia</td>
</tr>
<tr>
<td>4</td>
<td>59</td>
<td>M</td>
<td>12000</td>
<td>Pneumonia</td>
</tr>
<tr>
<td>5</td>
<td>61</td>
<td>F</td>
<td>54000</td>
<td>Flu</td>
</tr>
<tr>
<td>6</td>
<td>65</td>
<td>F</td>
<td>25000</td>
<td>Gastritis</td>
</tr>
<tr>
<td>7</td>
<td>65</td>
<td>F</td>
<td>25000</td>
<td>Flu</td>
</tr>
<tr>
<td>8</td>
<td>70</td>
<td>F</td>
<td>30000</td>
<td>Bronchitis</td>
</tr>
</tbody>
</table>

(a) The quasi-identifier table (QIT)

<table>
<thead>
<tr>
<th>Group-ID</th>
<th>Disease</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dyspepsia</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>Pneumonia</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Bronchitis</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Flu</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Gastritis</td>
<td>1</td>
</tr>
</tbody>
</table>

(b) The sensitive table (ST)

Figure 2.3: Table result after applying Anatomy
For the Categorical Attributes - According to [8] there is 2 ways to calculate the distance:

- Equal Distance: meaning that the distance of any two values is 1.
- Hierarchical Distance: meaning that the distance between two leaf values is defined on the lowest level ancestor of those two values, divided by the height of the tree.

k-anonymity prevents identity disclosure but not attribute disclosure, L-diversity solves that problem but it has some limitations and T-closeness tries to fill those gaps but does not protect against identity disclosure.

2.1.4 ε-Differential Privacy

Differential privacy was introduced in [21] and differs from the above techniques because it belongs in the family of randomization techniques, which means that this model uses a different approach: before the dataset is released for the query, differential privacy does not change the original data but controls how much noise is needed to insert into the true result to get the necessary privacy guarantees. ε is a parameter that represents the measure of privacy lost, the higher the ε the less private the dataset becomes.

Let A be a randomized algorithm that accepts as an input a database D and outputs an answer to some query on the database. A is said to provide ε-differential privacy if for any two databases, D1 and D2, that differ in only one entry, and for any subset S of values in the range of outputs of A,

\[ P[A(D_1) \in S] \leq e^\varepsilon P[A(D_2) \in S] \] .

(2.2)

i.e., the probability of a mechanism produces any given output changes by at most a multiplicative factor when add or remove one record from the input.

A(D) = Result of the query + noise mechanism

The type of mechanism that is added to the true result depends if it is for numeric or is it for non-numeric queries. For numeric queries - Dwork et al [9] propose the Laplace mechanism because when adding noise samples to the true value with \( \mu = 0 \) and \( \beta = \frac{\Delta F}{\varepsilon} \) (where \( \Delta F \) is the maximum distance between the query results of the two datasets, called global sensitivity). It is used Laplace mechanism because it preserves differential privacy, although different distributions could also be used but it requires to be calibrated differently. For non-numeric queries - McSherry and Talwar [10] propose the Exponential mechanism. Which has an arbitrary range \( R \) of outcomes, a quality score function \( q(D,r) \) that represents how good output \( r \) is for the database \( D \), and have a global sensitivity \( \Delta q \) where is the maximum difference between the score \( q(D_1,r) \) and \( q(D_2,r) \) for any outcome \( r \). He chooses an outcome \( r \) from the range \( R \) with probability
\[ P[r] \sim \exp\left(\frac{\varepsilon \cdot q(D,r)}{2\Delta q}\right). \]  

The idea is to make high quality outputs exponentially more likely at a rate that depends on the sensitivity of the quality score and the privacy parameter.

**Example:** Suppose the query is "how many people have green eyes?" and suppose that we choose \( \varepsilon = 0.1 \). Since the query result is numeric, we can use Laplace mechanism. The sensitivity of the query is \( \Delta f = 1 \), because the maximum change that the query can suffer is 1. So, the noise taken from the samples of \( \text{Lap}(\beta = 1/0.1) \) is added to the true value, for example: if the true value is 1 and a random sample from Laplace is 10 then the result would be 11.

There are two composition properties that are used: Sequential Composition [10] for a sequence of differentially private executions, that means that if a set of \( n \) mechanisms are sequentially executed and each one of them provides \( \varepsilon_i \) privacy, then the all set provide \( (\sum_{i=1}^{n} \varepsilon_i)\)-differential privacy, and Parallel Composition [14] for disjointed subsets, that means that if each mechanisms of that set are performed on a disjoint subset of the database, then the all set provide \( (\max\{\varepsilon_1, \ldots, \varepsilon_n\})\)-differential privacy.

Differential privacy prevents information disclosure from strong background attacks when the result of the query for a specific information, in different datasets, has a high probability to both results be the same. With that, the attacker cannot distinguish whether that specific information was or was not in the data.

Independent of which technique is used there is always a trade-off between privacy and data utility (see Figure 2.4) – Image obtain from [11]. The ideal solution would be maximizing privacy and data utility but it is impossible to achieve because due to the limitations of the models and the need to connect databases from different systems.

![Figure 2.4: Trade-off between privacy and data utility](image)
2.2 Difficulties in geolocation

The above anonymization techniques cannot be used directly to geolocation for 2 reasons:

1. They were created for data privacy and not for location-based queries.
2. They only make privacy for a snapshot of the database.

But Differential privacy seems to be accepted as a standard for privacy preservation.

For the geolocation privacy mechanisms, it needs to take into consideration temporal and spatial correlation and not only static timestamp because adversaries can use that information to infer a user’s trajectory. **For example:** Suppose the user location is obfuscated within a radius, since it does not consider temporal correlation, it is easy to understand that the radius is moving with a given direction. The real challenge is to create a synthetic trace that resembles a real trace while giving privacy protection at the same time.

Another way to anonymize, for example a medical record, is using a hybrid system, which is the combination of model-based (Machine Learning models) and rule-based (pattern matching) [29], like Best-of-Breed System [30] “a hybrid design system, uses rules and dictionaries to score a higher recall, and it also uses model-based classifiers in order to score a higher precision” [31]. This system is applied to unstructured data and in our case, we have structured data.

2.3 Location privacy techniques

Concerns about protecting the individual’s location, has led to the creation of new anonymization methods that preserve the privacy of geolocation. Here are some examples:

2.3.1 Geo-indistinguishability

This method [12] is a generalization of Differential privacy. The user has a L-privacy, where L represents the user privacy level which is proportionally to a radius parameter R, meaning that when R is smaller, L is also smaller so the privacy is higher, because L = εR.

Let X be the user’s possible locations (points of interest), Y the set of possible reported values and K is a probabilistic function assigning each location a probability distribution on Y.

The idea of ε-geo-indistinguishability for ε=L/R, is to make two points in radius R indistinguishable, which is defined by a probabilistic model [12].

Based on that probabilistic model, for any point x, x’ ∈ X (where the Euclidean distance metric between x and x’ is less or equal to R) the distance between two distributions K(x) and K(x’) should be less than L.
In the case of multiple points of interest, his location should also be kept private. According to [12] they define the distance between two sets of tuples, \( x = [x_1, ..., x_n] \) and \( x' = [x'_1, ..., x'_n] \), has the maximum distance, \( D \), in each \( i \)-th points of both tuples. When that distance is less or equal to \( R \) it means that all dots from the two tuples are within distance \( R \) from each other. So, we can use the \( D \) as the distance metric.

For the case of multiple points of interest, to have a set of reported values \( y=[y_1, ..., y_n] \), it is important to independently apply \( K_i \) to \( x_i \), \( 1 \leq i \leq n \) and the level of privacy that the combined mechanism \( K \) satisfies \( \varepsilon \)-geo-indistinguishability is for \( \varepsilon = \sum_{i=1}^{n} \varepsilon_i \).

The user’s location points, let it be \( x_0 \in IR^2 \), which are within radius \( R \) gets an addition noise that comes from the Laplace distribution in the continuous plane, but that distribution is linear so we need to use the planar version of it [12],

\[
K_{\varepsilon}(x_0)(x) = \frac{\varepsilon^2}{2\pi} e^{-\varepsilon d(x_0,x)} \quad (2.4)
\]

Where \( \varepsilon^2/2\pi \) is normalization factor and \( K(x)(y) \) is the probability of reported point of \( x \) in \( y \subseteq Y \) i.e. represents the probability of reporting the fake point \( y \) when the real point is \( x \).

Since the planar Laplace version only depends on the distance from \( x_0 \), it would be more useful if we draw dots efficiently using polar coordinates \( (r, \theta) \), where \( r \) is the distance of any other point within \( R \) [12]. After that, it remaps \( (r, \theta) \) to the closest point \( x \) on the grid.

The final step is to choose the dots that are only within the desired area, this step is important because the Laplace mechanism can generate points anywhere in the grid and we must choose the ones that belongs in the region with radius \( R \).

**Vulnerability**

As good as Geo-indistinguishability seems to be, it is vulnerable to inference attacks [24] – for example: Suppose the user location is obfuscated within a radius \( R \), since it does not consider temporal and spatial correlation, it is easy to understand that the radius is moving with a given direction, exposing the user’s trajectory.

There are several other techniques that make location perturbation or obfuscation [22, 23, 25, 26, 27, 28] but just like Geo-indistinguishability they do not consider the temporal sequence and spatial correlation, making the trajectory of a user more predictable.

For the ones that rely only on Differential Privacy, they are vulnerable to 3 types of attacks [13] (Bayesian Inference Attack, Partial Sniffing Attack and Outlier Leakage Attack) because it is out of protection reach of Differential Privacy.
a. **Bayesian Inference Attack**

Let $A(S)$ be the *informed prior* knowledge, which is the strongest belief that an attacker can have, over a sensitive area $S$. Let $T_{syn}$ be the synthetic trace produced by AdaTrace. If the difference between $A(S)$ and $A(S|T_{syn})$ is significant then there is privacy disclosure.

**Example:** Imagine that an attacker gets a public information from a website that, on a normal season, 5% of the people that goes to the hospital lives in a certain area. So, our $A(S)$ is the 5%. If after seeing $T_{syn}$ the 5% can become 60%, $A(S|T_{syn})$, then he can infer that there is a flu in that area, because the difference from 60% and 5% is high, which means that are more sick people coming from that region to the hospital.

Differential privacy cannot protect from this because even with adding Laplace noise the 60% could be 55%, which is still high.

b. **Partial Sniffing Attack**

Suppose that an attacker can sniff real trajectories, which consists of sniffing a sub-trajectory of the full trajectory, within a certain area. Even if $T_{syn}$ is publish, the attacker can try to see, in that area, the trace that is more like the real trajectory, linking the real one to the user with some high probability and after that he can break the privacy of the trajectory by finding the rest of it outside of the sniffing area.

Using Figure 2.5 as an example - Image obtained from [13], inside the sniff region the synthetic that is more like to the real one is Synthetic number 1. With that is easy to understand that $T_s$ is the synthetic trajectory of $T_u$.

![Figure 2.5: Partial Sniffing Attack](image)

In the image $T_s$ is the synthetic trace and $T_u$ is the real trace.
c. **Outlier Leakage Attack**

An outlier is an observation point with characteristics that are considerably different than most of the other observations. So, an attacker can use that unusual characteristic to break privacy. **Example:** Suppose that we observe a lot of trajectories of taxis in the city then there is not much we can learn about it, but sometimes trajectories from isolated locations to the city appears, which leaks information about travel plans, identity of the individual, etc.

Outliers do not work well on Differential privacy because it adds too much noise to the real value, since the value of sensitivity, $\Delta f$, is high.

A recently robust mechanism called AdaTrace [13] was published, capable of defending against the above attacks and solving the previous challenges among others.

### 2.3.2 AdaTrace

AdaTrace [13] generates a location trace synthesizer from the true location traces by doing feature extraction, noise insertion, and feature synthesis. The generation of synthetic traces satisfy differential privacy so it guarantees that those traces are unlikable to any real trace, i.e., when adding or removing real traces it only changes by at most a multiplicative factor on the outcome. In Figure 2.6 we can see the architecture of AdaTrace – Image obtain from [13].

![Architecture of AdaTrace](image)

Figure 2.6: Architecture of AdaTrace

Besides using the 2 mechanisms Laplace and Exponential mechanisms, and the 2 composition properties that Differential privacy has: Sequential Composition [10] and Parallel Composition [14], which were described on Differential privacy section. AdaTrace adds a third
composition property [13], that makes sure that after processing the output of the differential privacy does not compromise the privacy or the value of $\varepsilon$, by making the synthesis algorithm doing the calculation on the features without using the real data.

They identify 3 types of attacks, it is explained on the previous section, (Bayesian Inference Attack, Partial Sniffing Attack and Outlier Leakage Attack) which are important to consider when sending synthesized traces because they are out of Differential privacy protection. We will explain the defense to those attacks.

a. Bayesian Inference Attack

A defense for this attack proposed by AdaTrace [13] is to make sure that the EMD [16] of $A(S)$ and $A(S|T_{syn})$ is less or equal to some controlled parameter $L$, if we want to limit the distance between those two probabilities, we have to specified the maximum distance allowed using the parameter $L$, with that those distances becomes smaller.

$$EMD\left(A(S), A(S|T_{syn})\right) \leq L.$$  \hspace{1cm} (2.5)

b. Partial Sniffing Attack

AdaTrace prevents this attack by finding the best similar synthetic trajectory in a sniffed region with a real trajectory sniffed, that also has a certain amount of common points between them and certain number of times the synthetic trace visits an area.

More formally, it needs values for the parameters $\varphi$ and $\varphi$ (where $\varphi$ represents the maximum number of intersection allowed between two trajectories and $\varphi$ represents the maximum number of times a given trajectory visited a certain area) and with that two parameters he does the following [13]:

1. From all synthetic trajectories in that sniff region, he tries to find $T_s$ that has the minimal distance of similarity (by doing a distance matrix in which each entry has the result of the difference module for each point of the trajectories) from the real trajectory that was sniffed in that region by using the Dynamic Time Warping (DTW) [17] as the distance metric.
2. If the number of intersection between $T_s$ and $T_u$ is greater than a parameter $\varphi$ then $T_s$ cannot be used.
3. If the number of times that $T_s$ visits a certain area $S$ is greater than a parameter $\varphi$ then $T_s$ cannot be used.

If $T_s$ passes all those validations then $T_s$ is safe against this attack.
c. Outlier Leakage Attack

AdaTrace [13] identifies 3 types of different outliers (Trip outlier is when occurs unusual start and end locations, Length/duration outlier is when the length of the trip is too different from the unusual ones and Mobility outlier is when occurs unusual mobility patterns) by using 3 different distance measures to each of them:

1. **Trip outlier:** It uses geographical distance between the start and end locations, to find non-usual trips.
2. **Length/duration outlier:** It uses the length distance between the 2 trajectories, to find bizarre long or short trajectories lengths.
3. **Mobility outlier:** It uses Jensen-Shannon Divergence (JSD) distance [19] to quantify the similarity between the probabilities distributions Markov Chain mobility model of both trajectories, to find unusual mobility.

After identifying those outliers, AdaTrace uses the distance measures, D, that was used to find those outliers to defend. The idea of the defense is to check whether an outlier trajectory is blend into a group of trajectories.

More formally, it needs values for the parameters $\kappa$ and $\beta$ (where $\kappa$ represents the number of elements needed around an outlier, $\beta$ represents the maximum difference allowed between distances from outliers and real trajectories, and from outliers and a selected trajectory and $T_{real}$ contains all real traces) and with that parameters he can do the following conditions [13]:

1. Tries to find the trajectory $T_{similar}$, that has the minimal distance D, between the outlier trajectory $T_{outlier}$, and all real trajectories.
2. Let $Y$ be the number of trajectories, where the difference between the distance of $T_{outlier}$ and all real trajectories and the distance of $T_{outlier}$ and $T_{similar}$ is less or equal to a parameter $\beta$. So, the condition is: $D(T_{outlier}, T \in T_{real}) - D(T_{outlier}, T_{similar}) \leq \beta$
3. If $Y < \kappa$ then $T_{outlier}$ does not have enough $\kappa$ elements around him to satisfy that condition. So, a non-outlier trace must replace $T_{outlier}$.

To be able to create synthetic trajectories, AdaTrace created 4 different features, shown in Figure 2.6, and each one of them has its own objectives and different $\varepsilon_i$ privacy ($\varepsilon_1$ is for the Density-Aware Grid, $\varepsilon_2$ is for the Markov Chain Mobility Model, $\varepsilon_3$ is for the Trip Distribution and $\varepsilon_4$ is for the Route Length Distribution), which means that the total privacy is $\varepsilon = \sum_{i=1}^{4}\varepsilon_i$.

Those features are:
1. **Density-Aware Grid**

To limit the number of combinations of simulated movements, it uses a discretization of location space with a grid structure that adapts according to the density of the cell [18], the higher the density region the more number of subdivisions to smaller regions it will need to take. An example is shown in Figure 2.7.

First, the grid is initialized with NxN cells with identical size, where N is a parameter and each cell is denoted by $C_1, C_2, \ldots, C_{N^2}$.

Then it counts the number of times a given cell $C_i$ appears in a trajectory $T$, normalized by the number of cells that $T$ has, denoted by $|T|$. The formula is the following:

$$g(T_{real}, C_i) = \sum_{T \in T_{real}} \frac{\text{# of occurrences of } C_i \text{ in } T}{|T|} \quad . \quad (2.6.a)$$

$$W = \{ g(T_{real}, C_1), g(T_{real}, C_2), \ldots, g(T_{real}, C_{N^2}) \} \quad . \quad (2.6.b)$$

The sensitivity ($\Delta W$) of the set of queries is 1, because it is normalized by $|T|$ otherwise the $\Delta W = |T|$, as proved in [13].

Finally, each $C_i$ are divided into $M_i \times M_i$ cells in which $M_i$ is proportional to $g(T_{real}, C_i) + Lap\left(\frac{1}{\epsilon_1}\right)$ and a grid constant to restrict the value of M. For example, see Figure 2.7 – Image obtain from [13].

![Figure 2.7: Density-Aware Grid transformation with N=2](image)

After doing the calculus for each cell $C_3 < C_4 \approx C_1 < C_2$ the values $M_i \times M_i$ for $C_i$ will be 1x1, 2x2, 2x2 and 3x3, respectively.
2. **Markov Chain Mobility Model**

Markov Chain has the property that the probability of the next state depends only on the previous one but in AdaTrace it uses a Markov Chain of order \( r \) which means that it only depends on the previous \( r \) locations instead of all of them. Each trajectory \( T \) is a vector with a sequence of cells that it passes through. So, the probability of \( T \) to be in a next cell \( C_{\text{next}} \) is:

\[
P(T[n+1] = C_{\text{next}} | T[1] \ldots T[n]) = P(T[n+1] = C_{\text{next}} | T[(n - r) + 1] \ldots T[n]) \quad (2.7)
\]

To satisfy Differential Privacy, Laplace noise, \( \text{Lap}(\frac{1}{\epsilon_2}) \), is added to the Markov probabilities and the sensitivity (\( \Delta \)) is 1 because they are calculated based on the sequence counts.

3. **Trip Distribution**

This feature prevents possible association between a start and a destination. The trip distribution is calculated differently. If it is a one-layer it just calculates how frequently the pair \((C_{\text{start}}, C_{\text{end}})\) is from all the others. If it is a two-layer it follows Ordinary Least Square approach to make consistent and optimized trip count. Using for simplicity the left grid of Figure 2.7 as an example, it is a one-layer grid so, it calculates for every possible pair \((C_{\text{start}}, C_{\text{end}})\) how frequent every pair is in the grid and it adds to his result a Laplace noise: \( \text{Lap}(\frac{1}{\epsilon_3}) \).

4. **Route Length Distribution**

It tries to find the best possible distribution that fits from all traces with that specific trip \( C_{\text{start}} \sim C_{\text{end}} \), by using the chi square test. They change their parameters by adding Laplace noise: \( \text{Lap}(\frac{1}{\epsilon_4}) \) for the mean and an adaptation of Exponential Mechanism for the median using score function as \( q = -|\text{rank}(x) - \text{rank}(\text{med})| \).

The combinations of these four features creates synthetic trajectories, by doing the following:

1. It gets the pair \((C_{\text{start}}, C_{\text{end}})\) from the trip distribution.
2. It finds the best probability distribution for that trip and takes a sample from it to be the length of the synthetic trajectory \( T_s \).
3. It fills the first entry and the last of \( T_s \), respectively, with \( C_{\text{start}} \) and \( C_{\text{end}} \). The rest it is fill in the next step.
4. Each cell of the grid is a candidate $C_{\text{cand}}$ and each one of them will contain a value from the multiplication of 2 weights. The first one comes from the probability that the next entry of $T_s$ is the candidate given the previous entries. The second weight comes from the probability that the final entry is $C_{\text{end}}$ given the previous entries and the current entry is the candidate.

5. Finally, the entry $i$'th of $T_s$ is fill with a cell that has a probability proportional to its weight.
3 Dataset characterization

In this chapter, we will explain how the trajectory records are organized in both databases. After this notion, we describe our method to preserve the utility of the data but still be private enough to not get too much information exposed.

3.1 Trajectory Records

Trajectories are a set of points that together create a path – see Figure 3.1.

![Figure 3.1: Example of a trajectory](image)

Each path could have a different size, two paths could have been made on two different occasions, the weather could influence a trajectory of a person, etc. Since there are many variables that affect the outcome of a trajectory, we choose the simplest one – Brinkhoff [32] and Taxi [20] that has only the coordinates in Cartesian space of each points of the trajectories, see Figure 3.2.

![Figure 3.2: Example of trajectories in the database](image)

Those trajectories are written line-by-line and each one as an identifier that begins with “#”, for example: the \texttt{#0} is the identifier for the first trajectory. On each line, trajectory’s points are separated by semicolons with a format of (x-coordinate, y-coordinate), for example: the first point of the first trajectory is \texttt{11026.0, 4693.0}. The databases have a total of 20,000 trajectories represented by $|D|$. 
To get a better understanding of the characteristics of each trajectory that exist in both databases, we analyze the number of points of the trajectory using mean, standard deviation, minimum and maximum – see Table 3.1.

| $|D|$ | $\mu$ | $\pm \sigma$ | Min | Max |
|---|---|---|---|---|
| Brinkhoff | 20,000 | 64,275 | $\pm 35,853$ | 2 | 229 |
| Taxi | 20,000 | 49,6802 | $\pm 45,373$ | 2 | 2516 |

Table 3.1: Analyze of Brinkhoff and Taxi database

With the results presented on Table 3.1 we can see that there is a very diverse content of number of points of the trajectories, for example: in the Brinkhoff database the overall number of points of the trajectories is approximate 64 points but there also trajectories with 2 points and others with a maximum of 229 points. For the Taxi database, the values are little different there are cases of trajectories that contains 2516 points, it may seem large but since it is GPS trips of taxis then it is justifiable.
4 Proposed solution

The purpose of this chapter is to explain in detail the steps needed to reach our final goal, which is, to trade as little as possible from privacy to achieve a higher utility, where an attacker cannot discover sensitive information based on the points exposed.

4.1 Solution proposal overview

With the advancement of technology and the growing use of applications that allow users to obtain geographic information, namely geolocation services, has also increased the issue of privacy in those databases.

So, in the last decade, several anonymization techniques (K-anonymity, L-diversity, T-closeness, Differential Privacy, Geo-indistinguishability, AdaTrace) have emerged trying to solve that issue, preserving the usefulness of the data. Which is a difficult objective to achieve because there is always a balance between privacy and the usefulness of the data – as shown in Figure 2.4.

Each anonymization technique (K-anonymity, L-diversity, T-closeness, Differential Privacy, Geo-indistinguishability, AdaTrace) was created to solve the privacy vulnerabilities on sensitive data, for example: in patients’ medical records, favorite places, frequent trajectories, etc. where each new technique solves the previous techniques problems, for example: k-anonymity is vulnerable to homogeneity attacks, L-diversity solves that issue but it is vulnerable to probabilistic inference attacks, t-closeness defends against that attack but does not protect from identity disclosure, etc.

Taking that into consideration, AdaTrace seems to be the most robust/secure in terms of the user’s data privacy but sometimes the result produced may be too much private and may not contain the usefulness that we need, for example: suppose we want to know what people use the most in a park if it is the elevator or the ladders, the result produced may tell us a very different reality of what people really use.

Therefore, our tool allows to publish a more useful database but at the same time has a significant anonymization component that prevents an attacker to discover sensitive content based on the exposed points using the best method (Crescent uniform noise) to camouflage the non-anonymized result by adding a noise value, which is calculated based on Route Length distribution, and discovering the best areas in the map to exposed points.

By doing this we are trading part of the privacy to get a higher utility.
4.2 Choosing points from trajectories

The best solution that we found to provide a better utility of the published data was to previous ask which of the points of the trajectory does the person considers as important so that it appear in the anonymized trajectory while the rest of the points will go through the normal anonymization process, using Figure 3.1 as an example, where P1 is the source and P4 is the destination. Suppose the user choose P3 as an important point, then P3 is added to a list of important points appearing in the anonymized trajectory which subdivides the trajectory into two sub-paths P1 to P3 and P3 to P4. The point P2 does not appear in the anonymized trajectory because it was anonymized using AdaTrace components resulting a new point P’2 – see Figure 4.1.

![Figure 4.1: Example of anonymized trajectory from Figure 3.1](image)

4.3 Anonymization

The anonymization process of those trajectories is an important part that will be performed by AdaTrace components. We consider a sub-path length as the distance travelled between cells of the grid and a trajectory length is the sum of each sub-path length, for example if there are three points P1, P2 and P3 from a trajectory and P1 is in cell C1, P2 is in cell C2 and P3 is in cell C3 and all cells are adjacent cells then the sub-path length of P1 and P2 is 1 and the trajectory length is 2.

AdaTrace, as described in the Related Work section 2, is composed of 5 components (see the Related Work section 2 for more detail). The Density-aware grid tries to subdivide cells from the grid according to the density of each cell. The Markov Chain Mobility Model tries to create fake mobilities of a trajectory by making the next location dependent on a limited number of previous locations denoted as r. The Trip Distribution tries to prevent possible associations between a start and a destination. The Route Length Distribution tries to change the parameters of a distribution that can fit the best for all length trajectories with that start and end destination. The Trajectory Synthesis algorithm uses the above components to create a synthetic trajectory.

For our tool to work we need to adjust the Trajectory Synthesis algorithm to receive as an input a list of important points of the trajectory and to anonymize the sub-paths that exist, were an anonymization is the result of adding a real value with a fake value (noise). The noise produced by Route Length Distribution is added in the same way as AdaTrace but since we will have points
exposed and others not, that noise will not be added entirely to the length of the trajectory, so it will have to be distributed to any sub-paths that may exist.

So, we consider 4 different methods (Noise replication, Noise division, Uniform noise and Crescent uniform noise) to distribute the noise sample, some of them could add more noise in each sub path of the trajectory than in the other methods. We will explain each one of them, from the noisiest to the least noisy.

Using the example from Figure 4.2 we define \((P_x, P_y)\) as the sub-path from the point \(P_x\) to \(P_y\) where \(x\) and \(y\) are numbers that identifies the points of the trajectory, the real length of a sub-path is defined as \(|P_x, P_y|\) and the result of adding noise to the real length is defined as \(|P_x, P_y|'\) where it means that the Markov Chain Mobility Model needs to create new points based on that result minus one, for example if the result is 3 than it needs to create \(3-1 = 2\) new points.

In case two cells are not adjacent then we need to calculate the shortest length between them using the Euclidean distance.

As we can see in Figure 4.2 the length of sub-paths is \(|P_1, P_2| = 1\), \(|P_2, P_3| = 1\) and \(|P_3, P_4| = 1\) and suppose the noise sample produced by Route Length Distribution is 4:

1. **Noise replication:** This method adds a noise sample in each sub-path of the trajectory, for example suppose P2 and P3 are important points so there are three sub-paths \((P_1, P_2)\), \((P_2, P_3)\) and \((P_3, P_4)\) that will need noise which makes the length of each sub-path become \(|P_1, P_2|' = |P_1, P_2| + 4 = 5\), \(|P_2, P_3|' = |P_2, P_3| + 4 = 5\) and \(|P_3, P_4|' = |P_3, P_4| + 4 = 5\). In case P2 is the only important point then the two sub-paths are \((P_1, P_2)\) and \((P_2, P_4)\) because P3 is not consider as important point so the length of each sub-path become \(|P_1, P_2|' = |P_1, P_2| + 4 = 5\) and \(|P_2, P_4|' = |P_2, P_4| + 4 = 5\). The \(|P_2, P_4| \) is equal to 1 because they are adjacent. For this method,
exposing 2 important points made us create at total 12 new points for the trajectory while exposing only 1 it needed a total of 8 new points.

2. **Noise division:** This solution has a more control addition of noise in each sub-path of the trajectory, in comparison with the previous one, because from that Route Length Distribution it tries to divide the noise sample in each sub-path of the trajectory by using the following Equation 4.1:

\[
\frac{\text{noise sample}}{n^0 \text{ of important points} + 1}
\]  \hspace{1cm} (4.1)

dividing by (the number of important points + 1) or dividing by the number of sub-path that will be anonymized will produce the same value, for example in Figure 3.1 if we only consider the point P2 as an important point then using Equation 4.1 the result would be \[
\left\lfloor \frac{4}{1+1} \right\rfloor = \left\lfloor \frac{4}{2} \right\rfloor = 2
\] which is the same result as counting the number of sub-path to be anonymized, which is \((P_1, P_2)\) and \((P_2, P_4)\), becoming \[
\left\lfloor \frac{4}{2} \right\rfloor = \left\lfloor \frac{4}{2} \right\rfloor = 2.
\]

Then each sub-path length of the trajectory becomes \(|P_1, P_2|' = |P_1, P_2| + 2 = 3\) and \(|P_2, P_4|' = |P_2, P_4| + 2 = 3\). In case P2 and P3 were consider as important point we would have three sub-paths \((P_1, P_2)\), \((P_2, P_3)\) and \((P_3, P_4)\) and the result of Equation 4.1 becomes \[
\left\lfloor \frac{4}{3} \right\rfloor = \left\lfloor 1,333(3) \right\rfloor = 2
\] then the length of each sub-path turns into \(|P_1, P_2|' = |P_1, P_2| + 2 = 3\), \(|P_2, P_3|' = |P_2, P_3| + 2 = 3\) and \(|P_3, P_4|' = |P_3, P_4| + 2 = 3\). Using this method, exposing 2 important points made us create a total of 6 new points for the trajectory while exposing only one it needed a total of 4 new points.

3. **Uniform noise:** This one divides the noise sample in the same way as the previous one and distributes the result in a uniform way over the number of sub-paths that will be anonymized, by using the following equation 4.2:

\[
\frac{\text{noise sample}}{n^0 \text{ of sub paths to anonymize}} = x + y
\] \hspace{1cm} (4.2.a)

\[
y \cdot n^0 \text{ of sub paths to anonymize} \approx z
\] \hspace{1cm} (4.2.b)

Where \(x \in \mathbb{N}\) and \(y \in \mathbb{Q}\).
The value of x from Equation 4.2.a is helpful for knowing that to distribute uniformly, the result that is needed to add in the sub paths of the trajectory is x and x+1 and the value of z from Equation 4.2.b tells us how many times does x appears and x+1 appears.

If y is higher than 0.5 it means that x+1 appears more times than x but if it is less than 0.5 it means x appears more times than x+1, for example: suppose the point P2 and P3 of the trajectory are important points then the number of sub-paths to anonymize is 3, which is \((P_1, P_2)\), \((P_2, P_3)\) and \((P_3, P_4)\). So, in Equation 4.2.a will be \(\frac{4}{3} = 1 + \frac{1}{3} \approx 1 + 0.33\) with this we know that we will distribute the noise value of 1 more times than the 2 because 0.33 is less than 0.5. Using the Equation 4.2.b for calculating the number of times the noise appears, \(0.33 \times 3 \approx 1\) which means that the noise value of 1 appears two times while the noise value of 2 appears only one time. We can conclude that the best way to distribute uniformly the noise sample of 4 in 3 sub-paths with this example is: 1, 1, 2.

So, each sub-path length of the trajectory becomes \(|P_1, P_2|' = |P_1, P_2| + 1 = 2\), \(|P_2, P_3|' = |P_2, P_3| + 1 = 2\) and \(|P_3, P_4|' = |P_3, P_4| + 2 = 3\).

Now suppose that there is only one important point which is P3 then the number of sub-paths to anonymize is 2, which is \((P_1, P_3)\) and \((P_3, P_4)\). So, the result of Equation 4.2.a is \(\frac{4}{2} = 2\) where each sub-path length of the trajectory has the same as if we used the Noise division for this example. In this method exposing 2 important points made us create a total of 4 new points for the trajectory while exposing only one it needed a total of 4 new points.

4. **Crescent uniform noise:** This last one it uses the same method as the previous one but when the list of noise is uniformly calculated it does not add the noise from the first sub path to the last sub path of the trajectory but it adds in a crescent order of the list of lengths of each sub path of the trajectory.

Adding the lowest noise with the lowest length of sub-path and it adds the highest noise with the highest length of sub-path, for example: using the same example in the previous method where the best way to distribute uniformly the noise sample of 4 in 3 sub-paths is \([1, 1, 2]\) and the crescent order of length of each sub-paths of the trajectory is \([|P_1, P_2|, |P_2, P_3|, |P_3, P_4|]\). Then the first number in the list of the noise sample, which is 1, is added to the sub path \((P_1, P_2)\), the second number, which is 1 again, is added to the sub-path \((P_2, P_3)\), finally the last number, which is 2, is added to the remaining sub-path \((P_3, P_4)\).

So, each sub-path length of the trajectory becomes \(|P_1, P_2|' = |P_1, P_2| + 1 = 2\), \(|P_2, P_3|' = |P_2, P_3| + 1 = 2\) and \(|P_3, P_4|' = |P_3, P_4| + 2 = 3\).
In case P3 was an important point there were 2 sub-paths each sub-path length of the trajectory becomes the same as if we used the Noise division. For this method exposing 2 important points made us create a total of 4 new points for the trajectory while exposing only one it needed a total of 4 new points.

We have created each method and they have their advantages and disadvantages:

1. **Noise replication:**
   
   - **Advantages**
     - More privacy in the trajectory because it replicates the noise sample as many times as the number of sub paths.
     - Less time in computing a trajectory anonymized.
   
   - **Disadvantages**
     - It could produce too long sub paths lengths because we could be adding a huge noise sample.
     - If we sum the noise sample that was distributed it is must higher than the noise sample from route length distribution, in the previous example of noise replication, the noise sample from Route Length Distribution was 4 and in total the noise added was $4 + 4 + 4 = 12$ which could be an issue because the probability of getting a total noise sample of 12 could be almost nonexistent from the distribution that was taken the noise sample of 4.

2. **Noise division:**
   
   - **Advantages**
     - The noise sample to add in the sub paths of the trajectory depends in the number of existence sub paths, which helps to disperse the total noise sample from all important sub paths.
     - Less time in computing a trajectory anonymized.
• **Disadvantages**
  
  - It could be adding a high noise in comparing to a short length sub path or the opposite, be adding a very small noise to a huge length sub path.
  
  - If the result of the division produces a natural number, the noise distributed and the noise sample are equal. In cases where it does not happen, it becomes the same disadvantage of the “Noise replication” where the sum of noise sample that was distributed over the sub paths it is higher than the noise sample from route length distribution, for example: if the noise sample was 4 and the number of sub paths where 2, the list of noise distributed is [2, 2] because \( \frac{4}{2} = 2 \) and summing the noise distributed 2 + 2 = 4, which is equal to the noise sample. If we change the number of sub paths to 3, the list of noise distributed is [2, 2, 2] because \( \frac{4}{3} = 1.33(3) \approx 2 \) and summing the noise distributed 2 + 2 + 2 = 6, which is higher than the noise sample.

3. **Uniform noise:**

• **Advantages**
  
  - The noise sample to add in the sub paths of the trajectory tends to be smaller.
  
  - The sum of noise sample that is going to be distributed over the sub paths it is always equal to the noise sample from route length distribution.

• **Disadvantages**
  
  - More time needed in computing the uniform noise to add in each sub path of the trajectory.
  
  - Sometimes could add high noise to small sub path lengths and add small noise to high sub path lengths, where after the anonymization the small sub paths lengths could become the higher and the big sub paths lengths could become the smaller, using the example from Figure 4.2 suppose \( P_2 \) is in \( C_7 \) and \( P_3 \) is in \( C_3 \) and the real length of each sub path of the trajectory are |\( P_1, P_2 \)| = 3 , |\( P_2, P_3 \)| = 2 and |\( P_3, P_4 \)| = 1 and suppose the noise sample is 4, then the list of noise uniformly distributed is [1, 1, 2]. Before the anonymization, the highest sub path is \( (P_1, P_2) \) and the smaller is \( (P_3, P_4) \), after the anonymization the length of each sub path of the trajectory are |\( P_1, P_2 \)'| = |\( P_1, P_2 \)| + 1 = 4 , |\( P_2, P_3 \)'| = |\( P_2, P_3 \)| + 1 = 3
and $|P_3, P_4|' = |P_3, P_4| + 2 = 3$, where the highest sub path is $(P_1, P_2)$ and the smaller are $(P_3, P_4)$ and $(P_2, P_3)$.

4. **Crescent uniform noise:**

   - **Advantages**
     - It has the same advantages as the “Uniform noise”.
     - Allows to grow the length of each sub path with the same order as before of anonymization.

   - **Disadvantages**
     - This is the one that spends more time to finish the anonymization than with the other methods because it must order the list of sub path length in ascending order plus the time that the method “Uniform noise” spends to find a list of uniform noise to distribute.

Each method was evaluated using some metrics that are described and shown in section 5.3 to understand which one is better in terms of privacy and utility.
# 5 Evaluation Results

In this chapter, we describe the characteristics of the machine and evaluate the 4 methods, that were defined in the previous chapter, using metrics to see which one is better in terms of utility. We also test our tool with the AdaTrace results, in terms of privacy, to see how far are we from them, varying the private parameter $\varepsilon$.

## 5.1 Machine characteristics

All tests in the subsections below were performed using Java 8 in the same machine with the following hardware specifications:

- OS: Windows 10 x64
- CPU: Intel Core i5-6200U @ 2.30GHz – 2.40GHz
- RAM: 8GB 2133MHz

## 5.2 Evaluation Metrics

The list of metrics above (Query AvRE, Frequent Pattern AvRE, Frequent Pattern Similarity, Length error and Kendall-tau) are useful because it helps us to see how well does the trajectories from the Brinkhoff database [32] and Taxi database [20] becomes after the anonymization process in terms of frequented travel pattern, spatial and location density.

### 5.2.1 Query AvRE

Query AvRE is a metric that is very useful and used for evaluating data publishing algorithms based on anonymization [33, 34] in which evaluates the accuracy in answering counting queries, which is that type of query form that is used, for example: “What is the number of trajectories in a given area?”. The lower the value of Query AvRE the more useful it is for answering that query form.

In order to calculate the relative error a query $Q$ we use the Formula 5.1, let $D_{real}$ and $D_{synt}$ be the databases of real trajectories and anonymized trajectories, respectively and $L$ the lower bound that prevents the denominator to be zero, we putted a value of 0.01:

$$\text{Query AvRE} = \frac{L}{|D_{synt}|}$$
We generate randomly 200 queries where each one has an area with a random radius and after that we do the average of the relative errors of all queries.

5.2.2 Frequent pattern AvRE

A pattern \( P \) is an ordered list of cells that a trajectory passes through, using Figure 4.2 as an example: the points \( P_1, P_2, P_3, P_4 \) are in the cells \( C_1, C_2, C_3, C_6 \), respectively, then the pattern of that trajectory will be \( C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_6 \) but now suppose the point \( P_4 \) is in cell \( C_3 \), then the pattern of that trajectory will be \( C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_3 \). We define as \( occu(P, D) \), where \( D \) is a database, as the number of occurrences of the pattern \( P \) in \( D_{\text{real}} \) and \( D_{\text{synth}} \). This metric helps to measure how well does the anonymous database preserves the frequent patterns in the real database, measuring the difference in patterns’ occurrences. The top \( K \) patterns, which has the highest occurrences, in \( D \) is defined as \( T_K(D) \). The lower the value of FrP AvRE the better because it means that the top \( K \) frequent patterns did not suffer a huge different after the anonymization process. The formula to calculate the average of the top frequent patterns is:

\[
FrP\ AvRE = \frac{\Sigma_{P \in T_K(D_{\text{real}})} |occu(P, D_{\text{real}}) - occu(P, D_{\text{synth}})|}{occu(P, D_{\text{real}})} \]

We consider the top 100 frequent patterns to calculate it.

5.2.3 Frequent pattern Similarity

As already mentioned in the previous metric, the considered top \( K \) of frequent patterns also used to calculate how similar are the top \( K \) of the database \( D_{\text{real}} \) and the top \( K \) of the database \( D_{\text{synth}} \). The higher the value of F-measure the higher is the similarity of the pattern in both databases, this is a value between 0 and 1. The similarity is calculated using F-measure which is a harmonic average of the precision and recall.
The Precision tells us the percentage of the content that was labeled as positive and that is in fact positive [35]. Precision is defined as:

\[
\text{Precision} = \frac{TP}{TP + FP} .
\]  

(5.4)

TP means that it was classified by the system as positive and it was indeed positive. FP means that it was classified by the system as positive but was negative. In our case, we consider the TP as the number of patterns that are equal between \(D_{\text{real}}\) and \(D_{\text{syn}}\) and the \((TP+FP)\) as the size of \(D_{\text{syn}}\).

The Recall tells us the percentage of the content that was correctly identified by the system [35]. Recall is defined as:

\[
\text{Recall} = \frac{TP}{TP + FN} .
\]  

(5.5)

FN means that it was classified by the system as negative but that was in fact positive. In our case, we consider the \((TP+FN)\) as the size of \(D_{\text{real}}\).

5.2.4 Length error

This is a good metric because it measures the trajectory lengths and sees how different they are in both databases: \(D_{\text{real}}\) and \(D_{\text{syn}}\).

We create 20 groups, where each one of them has an interval with the same size, from the shortest to the longest trajectory length and count how many trajectory lengths fits in each group in the database \(D_{\text{real}}\) and in \(D_{\text{syn}}\) separately. So basically, we end up creating 2 distributions of lengths, let denote the distribution as \(\mathcal{L}\), from the database \(D_{\text{real}}\) and the database \(D_{\text{syn}}\). The length error is calculated using Jensen-Shannon divergence (JSD) which is used to measure the

\[
F - \text{measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} .
\]  

(5.3)
similarity between two probability distributions, the lower the value of Length error is, the more similar are the lengths from the database $D_{real}$ to the database $D_{synt}$:

$$\text{Length error} = JSD \left( \mathcal{L}(D_{real}), \mathcal{L}(D_{synt}) \right) . \quad (5.6)$$

The JSD is a symmetrized and smoothed version of the Kullback-Leibler divergence (KLD) and it is defined as:

$$JSD(P||Q) = \frac{1}{2} (KLD(P||M) + KLD(Q||M)) . \quad (5.7.a)$$

$$KLD(P||Q) = \sum_{x \in \chi} P(x) \ln \left( \frac{P(x)}{Q(x)} \right) . \quad (5.7.b)$$

Where $P$ and $Q$ are probability distributions and $M$ is the mid-point of the probability vectors $P$ and $Q$ calculated as $M = \frac{1}{2} (P + Q)$.

### 5.2.5 Kendall-tau coefficient

The Kendall-tau coefficient is useful for measuring relationships between columns of ranked data, in our case we will use to measure how discrepant are the number of times a cell is visited by a trajectory in which we define as the fame of the cells.

The value return is between -1 and 1, where -1 means that there is no relationship and 1 is a perfect relationship. The Kendall-tau coefficient is defined as:

$$Kendall = \frac{n^0 \text{ of concordant pairs} - n^0 \text{ of discordant pairs}}{n(n - 1)/2} . \quad (5.8)$$

We consider a list of cells as $Cell_1, ..., Cell_n$, the $n$ is the size of the list of cells and $Fame(D, Cell_i)$ the fame of the cell in the database $D$. We say that a pair of cells $(Cell_i, Cell_j)$ are concordant if condition (a) or (b) are fulfilled:

$$Fame(D_{real}, Cell_i) < Fame(D_{real}, Cell_j) \land Fame(D_{synt}, Cell_i) < Fame(D_{synt}, Cell_j) \quad (a)$$
\[ \text{Fame}(D_{\text{real}}, Cell_i) > \text{Fame}(D_{\text{real}}, Cell_j) \land \text{Fame}(D_{\text{syn}}, Cell_i) > \text{Fame}(D_{\text{syn}}, Cell_j) \] (b)

If the values are all equals the pair is neither concordant nor discordant, is consider tied.

5.3 Evaluating the 4 methods

Has described in section 4.2 we developed 4 different methods (Noise replication, Noise division, Uniform noise and Crescent uniform noise) to distribute the noise sample in a uniform way over the trajectory.

In this section, we will show the results of the previous metrics applied over the 4 methods, including execution time. Each result of the evaluation of the 4 methods was executed 10 times where we perform the mean and the standard deviation. We only publish 1 important point of a trajectory, meaning that it will only have two sub paths to be anonymized, which is the simplest case to compare the evaluation results when using different methods.

5.3.1 Noise replication results

The Table 5.1 shows us the evaluation results of this method using Brinkhoff database. After the noise value become 2, the FP Similarity decays, because it begins to have a bigger length of the sub-trajectory to generate which makes the anonymized trajectory less similar with the trajectory that is in the original database.

There is an interesting correlation in terms of frequent patterns, between FP AvRE and FP Similarity, when the mean value of FP AvRE declines then FP Similarity will increase and vice versa, for example: changing the noise from 1 to 2, the FP AvRE went down from 0.365 to 0.360 and FP Similarity went up from 0.708 to 0.710.
<table>
<thead>
<tr>
<th>Noise replication</th>
<th>Query AvRE</th>
<th>FP AvRE</th>
<th>FP Similarity</th>
<th>Length Error</th>
<th>Kendall</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise = 1</td>
<td>μ</td>
<td>0.145</td>
<td>0.365</td>
<td>0.708</td>
<td>0.067</td>
<td>0.726</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.003</td>
<td>0.010</td>
<td>0.013</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td>Noise = 2</td>
<td>μ</td>
<td>0.141</td>
<td>0.360</td>
<td>0.710</td>
<td>0.067</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.004</td>
<td>0.012</td>
<td>0.012</td>
<td>0.003</td>
<td>0.002</td>
</tr>
<tr>
<td>Noise = 3</td>
<td>μ</td>
<td>0.126</td>
<td>0.361</td>
<td>0.702</td>
<td>0.067</td>
<td>0.726</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.003</td>
<td>0.009</td>
<td>0.006</td>
<td>0.003</td>
<td>0.002</td>
</tr>
<tr>
<td>Noise = 4</td>
<td>μ</td>
<td>0.127</td>
<td>0.367</td>
<td>0.698</td>
<td>0.068</td>
<td>0.726</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.003</td>
<td>0.009</td>
<td>0.009</td>
<td>0.002</td>
<td>0.002</td>
</tr>
</tbody>
</table>

Table 5.1: Evaluation results of noise replication with different noise values

When increasing the noise value while using this method it is normal that the execution time will also increase because there will be more replicated noise to add in each sub part of the trajectory and the last column of Table 5.1 proves that.

### 5.3.2 Noise division results

The evaluation results of this method are different from the previous, for example: the correlation between FP AvRE and FP Similarity that appeared in the Table 5.1, have some discrepancies when using this method, see Table 5.2, for example: from the noise value 1 to 2 it has the correlation but when it changes the value to 3, both of FP AvRE and FP Similarity mean values goes up instead of one of them going up and the other going down.

When increasing the noise value, the FP Similarity only starts to increase after the noise value becomes 2 and looking at the last column of Table 5.2 we can see that the execution time also starts to increase after the noise value becomes 2, the same happen with Uniform noise method, see Table 5.3.
<table>
<thead>
<tr>
<th>Noise division</th>
<th>Query AvRE</th>
<th>FP AvRE</th>
<th>FP Similarity</th>
<th>Length Error</th>
<th>Kendall</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise = 1</td>
<td>μ</td>
<td>0.136</td>
<td>0.360</td>
<td>0.706</td>
<td>0.068</td>
<td>0.726</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.006</td>
<td>0.009</td>
<td>0.012</td>
<td>0.004</td>
<td>0.003</td>
</tr>
<tr>
<td>Noise = 2</td>
<td>μ</td>
<td>0.140</td>
<td>0.364</td>
<td>0.701</td>
<td>0.067</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.005</td>
<td>0.009</td>
<td>0.013</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>Noise = 3</td>
<td>μ</td>
<td>0.126</td>
<td>0.366</td>
<td>0.704</td>
<td>0.066</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.004</td>
<td>0.011</td>
<td>0.009</td>
<td>0.003</td>
<td>0.003</td>
</tr>
<tr>
<td>Noise = 4</td>
<td>μ</td>
<td>0.132</td>
<td>0.364</td>
<td>0.710</td>
<td>0.067</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.004</td>
<td>0.010</td>
<td>0.009</td>
<td>0.003</td>
<td>0.001</td>
</tr>
</tbody>
</table>

Table 5.2: Evaluation results of noise division with different noise values

5.3.3 Uniform noise results

<table>
<thead>
<tr>
<th>Uniform noise</th>
<th>Query AvRE</th>
<th>FP AvRE</th>
<th>FP Similarity</th>
<th>Length Error</th>
<th>Kendall</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise = 1</td>
<td>μ</td>
<td>0.130</td>
<td>0.362</td>
<td>0.708</td>
<td>0.068</td>
<td>0.725</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.003</td>
<td>0.008</td>
<td>0.005</td>
<td>0.003</td>
<td>0.002</td>
</tr>
<tr>
<td>Noise = 2</td>
<td>μ</td>
<td>0.147</td>
<td>0.359</td>
<td>0.703</td>
<td>0.067</td>
<td>0.728</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.005</td>
<td>0.006</td>
<td>0.004</td>
<td>0.002</td>
<td>0.003</td>
</tr>
<tr>
<td>Noise = 3</td>
<td>μ</td>
<td>0.132</td>
<td>0.361</td>
<td>0.710</td>
<td>0.067</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.004</td>
<td>0.006</td>
<td>0.004</td>
<td>0.002</td>
<td>0.002</td>
</tr>
<tr>
<td>Noise = 4</td>
<td>μ</td>
<td>0.130</td>
<td>0.363</td>
<td>0.712</td>
<td>0.067</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.004</td>
<td>0.008</td>
<td>0.004</td>
<td>0.003</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Table 5.3: Evaluation results of uniform noise with different noise values
5.3.4 Crescent uniform noise results

This method has interesting evaluation results because on Table 5.4 we can see that from the noise value 1 to 2 the mean values of FP AvRE and FP Similarity stay the same and the big change occurs when the noise value is 3. This happen because the Crescent uniform noise method was able to properly distribute the noise value over the 2 sub trajectories, i.e. he broke the value 3 in two noise values 1 and 2, and add the higher value to the highest length sub-trajectory and the lower value to the lowest sub-path length.

<table>
<thead>
<tr>
<th>Crescent uniform noise</th>
<th>Query AvRE</th>
<th>FP AvRE</th>
<th>FP Similarity</th>
<th>Length Error</th>
<th>Kendall</th>
<th>Time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise = 1</td>
<td>μ</td>
<td>0.122</td>
<td>0.362</td>
<td>0.707</td>
<td>0.066</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.004</td>
<td>0.006</td>
<td>0.008</td>
<td>0.003</td>
<td>0.002</td>
</tr>
<tr>
<td>Noise = 2</td>
<td>μ</td>
<td>0.119</td>
<td>0.362</td>
<td>0.707</td>
<td>0.067</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.004</td>
<td>0.004</td>
<td>0.010</td>
<td>0.004</td>
<td>0.002</td>
</tr>
<tr>
<td>Noise = 3</td>
<td>μ</td>
<td>0.122</td>
<td>0.353</td>
<td>0.720</td>
<td>0.066</td>
<td>0.726</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.003</td>
<td>0.006</td>
<td>0.011</td>
<td>0.002</td>
<td>0.003</td>
</tr>
<tr>
<td>Noise = 4</td>
<td>μ</td>
<td>0.120</td>
<td>0.361</td>
<td>0.712</td>
<td>0.067</td>
<td>0.727</td>
</tr>
<tr>
<td></td>
<td>σ</td>
<td>0.005</td>
<td>0.008</td>
<td>0.012</td>
<td>0.005</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Table 5.4: Evaluation results of crescent uniform noise with different noise values

5.3.5 Comparing methods results

To have a better understanding when comparing the evaluation mean results of the four methods, we use the values of the previous section and transform it into graphs using the Brinkhoff database – see Figure 5.1.
Figure 5.1: Evaluation mean results with different noise values from Brinkhoff database
Surprisingly, from Figure 5.1 the values of Length Error and Kendall-tau variate almost nothing which means that the methods to distribute the noise value does not have directly influence in those results, for example the values from Kendall-tau in Figure 5.1 always stays between 0,725 and 0,728 and those values are close to 1, so the fame of each cell have almost a perfect relationship but when we look at FP Similarity, FP AvRE and Query AvRE mean values they variate a lot, for example in Figure 5.1 when we use Crescent uniform noise for Brinkhoff database we were able to preserve about 70,7% to 72% similarity of the top frequent patterns after anonymizing and for the Taxi database is between 71,9% and 78,2% which is in Figure 5.2 but we also have in Brinkhoff database about 35,3% to 36,2% error of patterns occurrences in the anonymous database and for the Taxi database is between 33,3% and 35,2%, see Figure 5.2.

The values obtain in Query AvRE, FP AvRE, FP Similarity from Figure 5.1 and the values of Query AvRE and FP AvRE from Figure 5.2 are a great example to use because different types of methods produce a different impact in the evaluation results, in some cases the variation is small for example in Length Error, and in the others the opposite happens for example in FP Similarity, especially when the noise hits the value of 3. That difference occurs because since there are only 2 sub-trajectories to be anonymized, i.e. there is 1 important point not to be anonymized, the result of dividing the noise value of 3 is $\frac{3}{2} = 1,5$ which gives us a value that does not belong to the integer group.
Figure 5.2: Evaluation results with different noise values from Taxi database
So, the type of method to use in order to distribute the noise value over the sub-paths is crucial. Using the previous example where there are 2 sub-paths, the list of noise to distribute over each sub-path as the following representation: [noise for the first sub-path, noise for the second sub-path], each method would produce different value in that list, in which Section 4.2 explains how each method calculates their way to distribute the noise over the sub-paths, for example: in Noise replication, the noise is added in each sub-paths in which the list of noise to distribute becomes [3, 3], in Noise division it divides the noise over the number of sub-paths becoming \( \left\lfloor \frac{3}{2} \right\rfloor = \left\lfloor 1.5 \right\rfloor = 2 \) and the list of noise to distribute is [2, 2], for the Uniform noise the best way to uniformly divide 3 into 2 sub-paths is [2, 1] and for the Crescent uniform noise the list could be [1, 2]. Which affects the mean values of FP Similarity, FP AvRE and Query AvRE as shown in Figures 5.1 and 5.2.

With that in mind, we can understand that the best method that is able to minimize AvREs and maximize the similarity is the Crescent uniform noise method – in Appendix A there is the same evaluation graphic values but with a different perspective.

Unfortunately, in Figure 5.3 and in Figure 5.4 we can see that there is an increase of time execution when the noise value increase.

![TIME EXECUTION](image)

Figure 5.3: Time execution using different methods with different noise values from Brinkhoff database
5.4 Attack resilience

Now that we discover that best method to provide more data utility, where more utility is to minimize AvREs and maximize similarities, is the Crescent uniform noise see Figure 5.1 and Figure 5.2, we need to pay attention at Figure 2.4 which tells us that there is a trade-off between data utility and privacy, meaning that when we increase data utility the privacy typically decreases.

We did some attacks, Partial Sniffing attack and Bayesian Inference attack, at our solution with Brinkhoff database and Taxi database to find which type of attack he is vulnerable and how much, by comparing with the result produced by the original system AdaTrace.

For the first attack, Partial Sniffing attack, it has only two types of values that can consider a trajectory as susceptible to the attack, which is 1) the number of times the most similar synthetic trajectory in a sniffed region intersects with a real trajectory sniffed and 2) the number of times the synthetic trajectory visits an area.

For the first type, the value changes a lot because it tries to find the most similar synthetic trajectory and those trajectories are always different each time the program is executed but more importantly the value is always equal to the one obtained from AdaTrace.
For the second type, the value is always the same and is equal to the value of AdaTrace – see Figure 5.5 and Figure 5.6.

With this in mind, our solution has the same privacy as AdaTrace for the Partial Sniffing attack.

![Partial Sniffing attack](image)

**Figure 5.5:** Partial Sniffing attack in both databases for 1 exposed point

![Exposed points in an area](image)

**Figure 5.6:** Partial Sniffing attack in both databases for multiple exposed points
For the second attack, Bayesian Inference attack, it only has one type of value that can consider a trajectory as vulnerable to the attack, which is the EMD, the lower the value the better.

Unfortunately, the overall EMD value that we obtain is significantly higher than AdaTrace from both databases, i.e. our solution is vulnerable to Bayesian Inference when exposing an important point – see Figure 5.7.

So, our objective now is to minimize the EMD value to be less vulnerable to Bayesian Inference attack or in other words to find the best places in the map to expose important points in order to have a minimal EMD value.

We first divided the map into 6x6 different areas with the same size and then analyzed each areas risk of vulnerability of the attack when exposing one important point, based on the EMD value obtained. We have also executed the analysis 30 times to have a EMD mean value and a standard deviation of the area – see Figures 5.8 and 5.9. We did the same analysis for a bigger number of areas up to 100 different areas in the map – see Appendix B.

Figure 5.8 presents the vulnerability map of the Brinkhoff database. There are 3 different colors, green, red and grey. The green color means that it has the smallest EMD value of them all and the red color is the highest, for example: the lowest value in the map is 59,9351 with a standard deviation of 2,8202 and the highest in the map is 364,6268 with a standard deviation of 2,1798, the grey color means that there is no trajectory that passes through that area so we cannot expose an important point in it.
Figure 5.9 shows the risk map of the Taxi database and it also has the same color code as the previous one. The lowest EMD mean value of all areas is 36,1096 with a standard deviation of 1,992 and the highest in the map is 682,3327 with a standard deviation of 4,636. This time there is a higher number of “no data” areas than in the previous map because there is a river called Douro that separates the city of Porto and the city of Vila Nova de Gaia – see Figure 5.10.
This way of finding the lowest and the highest risk areas to publish important points is a great idea to start with because from Figure 5.8 we can understand that from all 33 areas with data, only 2 are considered as low risk areas to expose one important point in it and with Figure 5.9 there are also 2 low risk areas within 25 areas with data, but does that risk increase with the number of important points exposed? The answer is no – see Figure 5.11 and Figure 5.12.

Apparently when exposing one important point the EMD mean value remains the same as if we exposed 14 important points in the same area, this is because all points are within the same region and the Bayesian Inference attack is based on informed prior knowledge. Which means that when we publish more than one important point, the new exposed information does not change much of what the attacker already knew or inferred about it because the points are in the same area.
The only case where the EMD mean value changes is when the important point exposed belongs to a different area, for example in Figure 5.12 when a jump occurs it means that point belongs to a different area.

If we ignore the risk areas and start to publish the same total number of points without bothering where does that point belongs in the space of the map then there is a huge privacy loss.

In Figure 5.12 the stagnations in the EMD value for the light color means that the important points belong to the same area and there is a jump because it starts to expose a point in a different area even if there were more points to be exposed in previous one but for the dark color he only jumps when all points in the same area are exposed.

So, this solution of acknowledging the risk areas can minimize the privacy loss for the Bayesian inference attack.

Figure 5.11: EMD value over the increase of exposing important points in both databases
5.5 Our solution utility vs AdaTrace utility

Now that we have found a way to minimize the privacy loss, we need to pay attention to how far are we from the original AdaTrace in terms of utility.

We used the same evaluation metrics, see Section 5.2, but change the privacy parameter $\varepsilon$ with values of 1.0, 1.5 and 2.0 for the Brinkhoff and Taxi database – see Figures 5.13, 5.14 and 5.15.

Where we want to minimize Query AvRE, FP AvRE and Length Error and maximize FP Similarity and Kendall-tau.

Figure 5.12: Variation of EMD value when using different solutions in (a) Brinkhoff database and (b) Taxi database
Figure 5.13: Evaluation results with $\varepsilon = 1$ for the (a) Brinkhoff database and for (b) Taxi database.

Figure 5.14: Evaluation results with $\varepsilon = 1.5$ for the (a) Brinkhoff database and for (b) Taxi database.
Even changing the values of the privacy parameter $\varepsilon$ our evaluation results is always better in every aspect, in Query AvRE, in FP AvRE, in FP Similarity, in Length Error and in Kendall-tau.

So, our solution can provide a higher utility of the data and still be private enough for both databases.
6 Conclusion

Nowadays users are concerned about data privacy because it is difficult to figure out if a company is collecting your information and how they are using it, for example: favorites places or frequent travels. That type of information can be used to disclose home location for a user.

In this thesis, we have talked about anonymization mechanisms, for example: k-anonymity, L-diversity, t-closeness, Differential privacy, Geo-indistinguishability and AdaTrace, to solve the problem for achieving private data and still be useful but all of the anonymization mechanisms have advantages and disadvantages, for example: k-anonymity is vulnerable to homogeneity attacks, L-diversity is vulnerable to probabilistic inference attacks, t-closeness does not protect from identity disclosure, Differential privacy adds a lot of noise if the value of sensitivity is higher, Geo-indistinguishability is vulnerable to inference attacks, and AdaTrace does not allow the user to specify which of his components must be more private and the anonymous database produced may not contain the utility that is needed.

That is why we develop 4 different methods to increase the utility of the data when we expose important points of the trajectory by introducing as an input a list of important points for the Trajectory Synthesis algorithm component to know which points must not be anonymized, those methods are: Noise replication, Noise division, Uniform noise and Crescent uniform noise, where the method that hits the highest utility value is the Crescent uniform noise. With the insertion of that method that distributes an uniform way the noise sample over the sub-paths of the trajectory based on the crescent order of the real sub-path length (which makes increase his length), an algorithm that receives as an input a map of trajectories of the database and divides into areas of equal lengths, in which it uses the EMD to discover the vulnerability risk of that area from a Bayesian Inference attack (the higher the value the more risky the area is), it was possible to reach our goal which is to trade the least possible of privacy to obtain a higher utility where, for example: people with some special characteristics can be safe and the result produced by the system be useful for them when using it.

6.1 Main Contributions

The main contribution of this thesis was to present a solution to increase the utility of the anonymous database by exposing points that are consider as important from a trajectory and developing 4 different methods to distribute a noise sample, which is produced by the Route Length Distribution component, in a uniform way over the trajectory. Those methods are Noise replication which adds the same noise value in each sub-path of the trajectory, Noise division takes
the noise value and divides it by the number of sub-paths of the trajectory and adds the final result division on each sub-path, Uniform noise finds the best way to uniformly divide the noise by the number of sub-paths and adds each result on each sub-path without any order and finally Crescent uniform noise uses the same idea as Uniform noise but adds the uniform noise result in a crescent order of the sub-path lengths.

We also have evaluated each one of the 4 methods to see which can produce a better utility, is the Crescent uniform noise, using 5 evaluation metrics: 1) Query AvRE is used for evaluating the accuracy in answering counting queries. The lower the value of Query AvRE the more useful it is for answering that query form; 2) Frequent Pattern AvRE helps to measure resemblance between the real database and the anonymous database with respect to frequent patterns where a pattern is an ordered list of cells that a trajectory passes through. The lower the value of FrP AvRE the better; 3) Frequent Pattern Similarity calculates how similar are the top K frequent pattern of the database $D_{real}$ and the top K frequent pattern of the database $D_{syn}$: The higher the value of F-measure the higher is the similarity of the pattern in both databases, this is a value between 0 and 1; 4) Length error measures the difference between trajectories lengths from both databases $D_{real}$ and $D_{syn}$ and 5) Kendall-tau measures how discrepant are the number of times a cell is visited by a trajectory.

In the end of the work, we detect the best places to publish information in the map of both databases, using the EMD in each area, to minimize the risk of suffering an Bayesian Inference attack from a malicious attacker, were he can try to discover clues based on the places that people have passed, knowing what kind of person it is, for example someone with reduced mobility chooses the route without stairs.

6.2 Future Work

A great difficulty of this work was to discover a way to include those persons in the privacy protection when exposing important points of the trajectory because we know that we were dealing with sensitive and private data and we want them to be safe but at the same time useful which also contributes to the difficulty of this work.

About future work, we considered:

- To identify whether or not there are better solutions that can reduce the risk of a Bayesian Inference attack.
Bibliography


Appendix A

This section has the same evaluation mean values as presented in section 5 but now the data it is presented from a point of view of the noise values.
NOISE = 4

Query AvRE
FP AvRE
Length Error
FP Similarity
Kendall-tau

Noise replication
Noise division
Uniform noise
Crescent uniform noise

0.1276 0.3613 0.6920 0.9963 0.7267
0.1324 0.3647 0.7141 0.7270 0.7277
0.1319 0.3638 0.7158 0.7270 0.7277
0.1266 0.3677 0.7120 0.7270 0.7277
0.1217 0.3646 0.7141 0.7270 0.7277
Appendix B

We have increase the number of cells up to 10x10 to see what would happen to the privacy and utility value. As we can see below there are no areas of low risk, where in a map of 6x6 it exists 2 low risk areas, and now the majority areas have middle risk. The lowest EMD mean value of all areas is 442,6314 with a standard deviation of 6,7052 and the highest in the map is 2874,1587 with a standard deviation of 30,8464. So, with a 10x10 map we are more vulnerable to an Bayesian inference attack.

When it comes to utility the percentage of error has increased and the similarity decreases. So, to publish important points the 6x6 map is better than using a 10x10 map.