NTNU

Norwegian University of
Science and Technology

K-Anonymity as a Service for Mobility Analytics

Optimizing the value of *k* in case of *k*-anonymity

Balder Riiser Haugerud

Guro Karine Mangelrød

Master of Science in Computer Science

Submission date: June 2018

Supervisor: Herindrasana Ramampiaro, IDI

Co-supervisor: Juwel Rana, Telenor

Norwegian University of Science and Technology

Department of Computer Science

Abstract

The mobile network operators collect data from mobile communication networks about their users. The data collected is referred to as mobility data and contains the user's spatial positions within the network over time. A longstanding theory states that aggregation of mobility data is enough to protect the privacy of the users. However, in the recent year, studies have proved that sensitive information, including individual trajectories from aggregated mobility data, can be recovered.

To protect the users contained in mobility data additional privacy techniques must be utilized. *K*-anonymity is a widely used location privacy technique, which states that it should not be possible to distinguish one user from k - 1 other users. The value of k is highly dynamic and changes in response to the characteristics of the mobility data.

This master thesis aims to dynamically determine what value of k for k-anonymity is the optimal value for a variety of mobility datasets. To achieve this, we propose a system that first attempts to recover individual trajectories using a modified version of the Hungarian algorithm. Then the mobility data is further protected by applying different levels of k-anonymity until the percentage of recovered trajectories is close to zero.

To evaluate the method aggregated mobility data from four different locations in Norway with a duration of three weeks was used. Since the collection and storing of individual trajectories is a breach of privacy regulations, there is no way of testing the accuracy of the recovered trajectories against the real individual trajectories. To cope with this, synthetic trajectories, computationally created to behave like human trajectories, was added to the original mobility dataset before running the algorithm for trajectory recovery.

The proposed system implies that the level of k for k-anonymity is insignificant, as the percentage of recovered trajectories stay relatively consistent when different levels of k are applied. Further analysis suggests that the characteristics of the utilized mobility data alone protect the data from privacy attacks because it produces common movement patterns instead of individual trajectories.

Sammendrag

Mobiloperatører samler inn data fra mobilkommunikasjonsnettverket om sine brukere. Dataene som samles inn kalles mobilitetsdata og inneholder brukerens geografiske posisjoner i nettverket over tid. En langvarig teori sier at aggregering av mobilitetsdata er nok for å beskytte personvernet til brukerne, men gjennom det siste året har studier vist at sensitiv informasjon, inkludert individuelle bevegelsesmønstre fra aggregerte mobilitetsdata, kan reidentifiseres.

For å beskytte brukerne må ytterligere personvernsteknikker benyttes. K-anonymitet er en mye brukt personvernsteknikk for lokasjonsdata, som sier at det ikke skal være mulig å skille en bruker fra k - 1 andre brukere. Verdien av k er svært dynamisk og vil forandres i forhold til mobilitetsdataenes karakteristikker.

Denne masteroppgaven har som formål å dynamisk bestemme hvilken verdi av k for k–anonymitet som er optimal for ulike mobilitetsdatasett. For å oppnå dette foreslår vi et system som først prøver å reidentifisere individuelle bevegelsesmønstre ved hjelp av en modifisert versjon av den ungarske algoritmen. Deretter beskyttes mobilitetsdataene ytterligere ved å bruke forskjellige nivåer av k–anonymitet, til prosentandelen av reidentifisere bevegelsesmønstre nærmer seg null.

For å evaluere metoden ble aggregerte mobilitetsdata fra fire forskjellige steder i Norge med en varighet på tre uker brukt. Siden innsamling og lagring av individuelle bevegelsesmønstre er brudd på personvern, er det ikke mulig å finne likheten mellom de reidentifiserte bevegelsesmønstrene og de virkelige individuelle bevegelsesmønstrene. For å håndtere dette ble syntetiske bevegelsesmønstre, laget for å oppføre seg som menneskelige bevegelsesmønstre, lagt til i det opprinnelige mobilitetsdatasettet før algoritmen for reidentifisering ble kjørt.

Det foreslåtte systemet indikerer at nivået av k for k-anonymitet er ubetydelig, da prosentandelen av reidentifiserte bevegelsesmønstrene forble relativt likt når forskjellige nivåer av k ble anvendt. Videre analyse tyder på at karakteristikkene til mobilitetsdataene alene beskytter dataene mot personvernangrep fordi det produserer standardiserte bevegelsesmønstre i stedet for individuelle bevegelsesmønstre.

Preface

This Master Thesis is submitted to the Faculty of Information Technology, Mathematics and Electrical Engineering (IE) as a requirement of fulfilling the master studies in Computer Technology at the Norwegian University of Science and Technology.

This project is a collaboration with Telenor and is a continuation of a specialization project carried out during the fall semester of 2017. The task given by Telenor was initially an open task with the aim to explore their aggregated mobility data and research the issues and opportunities that it brings. With research, exploration of the data and discussions with Telenor, the task for this master thesis was formed. The project was supervised by Heri Ramampiaro (IDI) and carried out in collaboration with Juwel Rana (Telenor).

Trondheim, 08.06.2018 Trondheim, 08.06.2018

Balder Riiser Haugerud Guro Karine Mangelrød

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The daily use of mobile devices has drastically increased over the past years, which generates an enormous amount of data. Everything we do on our mobile devices leaves a digital footprint. From posting updates to social media to browsing the web, and lastly, positions are picked up by mobile communication networks or GPS equipped devices. This data can be used to offer a wide range of services, that can benefit both companies and individuals, for example, for marketing purposes or live traffic maps. Also, the collected data from mobile devices are often sold to third parties or open for research purposes. Even though this can lead to new beneficial services, it comes with a responsibility to protect the users from privacy breaches.

The data collected by mobile communication networks is referred to as mobility data and contains the spatial positions picked up by base stations over time. Analysis of this type of data has many use cases, and can, for example, be used to find an estimate of the number of people traveling through catchment areas, to optimize, or even predict the flow of traffic.

The mobile communication networks are continuously improving, which in addition to providing a better service for its users, also gives a more accurate estimation of the user's positions [8]. This type of data is sensitive since it may reveal how private citizens move, and thus, where they live, where they work, and more. Each of these movements is referred to as individual trajectories. To address the resulting privacy issues, this type of data is usually aggregated so that the data only contains, for example, the number of people in an area at a given time.

A recent study [6] has shown that even aggregated mobility data can be used to recover trajectories of individual users from seeing where they travel day to day, without any prior knowledge about the data. This contradicts the longstanding idea that aggregated mobility data protects the privacy of the users. Their methods use the Hungarian algorithm, an algorithm to find the optimal assignment from a cost matrix [12], to derive trajectories from the hidden data.

Another study [4], has suggested that the presence of random noise in the dataset can reduce the possibility of recovering individual trajectories. A different method is to implement k-anonymity. k-anonymity is a method that reduces the risks of privacy issues by changing fields of data in a dataset to a lower level of granularity [5]. For example, in a dataset which contains the number of people covered by a base station, the dataset might be changed to only display counts greater than ten.

By adding noise, or make use of k-anonymity in addition to aggregating the dataset, the data will lose features and context. This loss might not be substantial for all cases. however, it is preferable to keep as much as possible of the original features and context of the data intact without affecting the privacy of the users.

1.1 Objectives and Research Questions

With the ongoing debates about privacy, and the new privacy regulation, GDPR [1], businesses need to address how to protect their data to avoid privacy breaches. Aggregation and k-anonymity are both known methods that can be utilized to anonymize data. When data is correctly anonymized, it is no longer affected by the Norwegian privacy legislation. Data is anonymous if it is no longer possible, with the tools that may reasonably be used, to identify the connection between information and individuals in the dataset [13]. Thus, when distributing mobility data as a service, it is crucial to know that it is impossible to recreate users individual trajectories, or other sensitive information. However, deciding the level of noise and anonymity that is necessary to protect the users without losing valuable information in the data, is not trivial.

This master thesis aims to address this issue by discovering the effect k-anonymity as a privacy measure has on trajectory recovery from aggregated mobility data. Mobility data comes in many different shapes and sizes, and therefore the optimal value of k will be dynamic with respect to the characteristics of the dataset.

The main goal of this thesis is to produce a system for k-anonymity in mobility analytics, with the aim of discovering the optimal value of k. This will be accomplished by first attempting to recover individual trajectories from aggregated mobility data by recreating the procedure in [6] and modifying it to fit this case. Furthermore, different values of k for k-anonymity will be applied to protect the data until the percentage of recovered trajectories reaches a point where privacy is persevered. This process is displayed in 1.1 and will be further explained in Chapter 4.

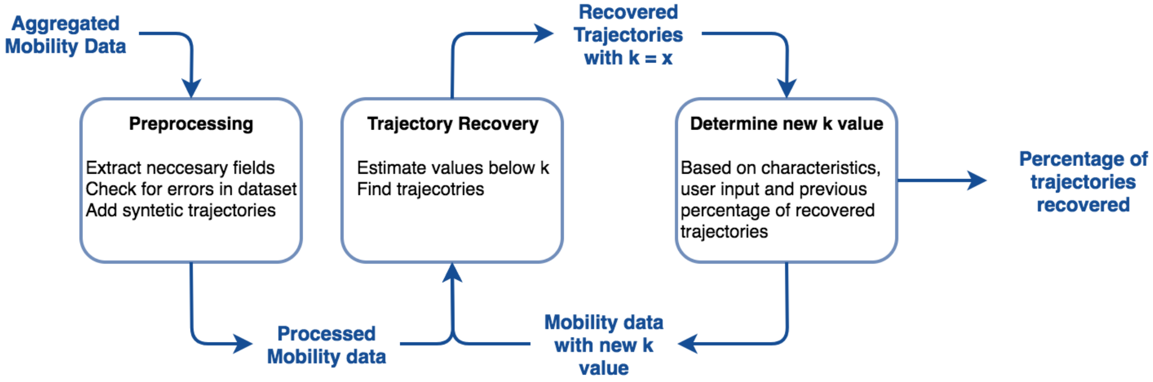

Aggregated Mobility Data
Preprocessing
Extract neccesary fields
Check for errors in dataset
Add syntetic trajectories
Processed Mobility data
Trajectory Recovery
Estimate values below k
Find trajecotries
Recovered Trajectories with k = x
Determine new k value
Based on characteristics, user input and previous percentage of recovered trajectories
Mobility data with new k value
Percentage of trajectories recovered

Figure 1.1: Mobility Analyctic Service

To achieve the main goal, this thesis addresses the following three research questions:

- **RQ 1:** How can individual trajectories from aggregated mobility data be recovered
- **RQ 2:** In case of k-anonymity in aggregated mobility data, how can the value of k be optimized in order to protect it from privacy attacks, in addition to preserving the utilization of the data
- **RQ 3:** How do the characteristics of the mobility data affect the value of k

1.2 Research Method

To achieve the goal and research questions, a review of the state-of-the-art approaches for trajectory recovery from aggregated mobility data and privacy techniques for this type of data will be conducted. As the field is relatively new, the review process will be limited to a few central researches, in addition to a review of topics which are relevant to the field.

Since the data in this project is rather large, much time will also be spent on preprocessing and preliminary analysis of the data, to understand the features and characteristics of this type of data. This analysis will also play an important role when implementing techniques discovered during the review, and adapting them to fit this case.

1.3 Evaluation Method

The evaluation of this master thesis will be based on the performance of the trajectory recovery, the quality of the recovered trajectories and the accuracy of the optimized value of k.

However, as this project is built upon aggregated, protected data and no testing set for individual trajectories is available, there is no direct and simple way to validate the performance of the system. Specifically, there is no way to know if the recovered trajectories are correct. Therefore, it is necessary to evaluate the trajectories based on other methods.

The evaluation of the trajectory recovery will, therefore, be completed by adding synthetic trajectories to the mobility data. More specifically, trajectories that are created computationally to behave like real trajectories, this will be further explained in section 4.2.3. Doing this, a testing set is available, and it will be possible to measure the performance of the trajectory recovery system by assessing the percentage of synthetic trajectories recovered by the system.

The quality of the trajectories will be evaluated by comparing the synthetic trajectories with the trajectories recovered by the system using edit distance.

Finally, the accuracy of the optimized value of k will be evaluated by looking at the percentage of the recovered trajectories to see if the data is protected while preserving the utilization.

1.4 Report Structure

The rest of this report is structured in seven chapters as presented below:

Chapter 2 Background Theory will present the background theory for the thesis. This includes Mobility data and mobile communication networks, data mining in general, and specifically data mining for mobility data. Finally, this chapter will present correlation techniques.

Chapter 3 Related Work will present the state-of-the-art approaches for trajectory recovery for aggregated mobility data, in addition to privacy research in this field.

Chapter 4 Approach will first describe the Mobility datasets used in this thesis. Next, this chapter will present the approach to recover individual trajectories from aggregated mobility data and finding the optimal value of k for k-anonymity in order to protect it sufficiently. The approach consists of data preprocessing, where the steps to make the data ready for trajectory recovery is described, the trajectory method, which presents a detailed description of the Hungarian algorithm and how it is modified to fit the data, and finally, the determination of the optimal k-value is presented.

Chapter 5 Results and Evaluation will describe the results and evaluation of the trajectory recovery and the k-anonymity optimization process.

Chapter 6 Discussion will discuss the results of this project in terms of the main goal and corresponding research questions. Also, it will discuss what aspects of the system need improvements, and which aspects perform well.

Chapter 7 Conclusion and Future work concludes this project and presents the future work.

Chapter 2

Background Theory

This chapter will present the background theory for this thesis. This includes Mobility data and mobile communication networks, data mining in general, and specifically data mining for mobility data. These theories provide fundamental knowledge for understanding the content of the data and how it behaves. Finally, this chapter will present correlation methods which are necessary for preprocessing techniques.

2.1 Mobility Data

Mobility data is data collected by location technologies like mobile communication networks, GPS, or other systems that collect data that contains spatial positions of mobile phones for a period of time. Mobility data is usually stream data and is collected in real time.

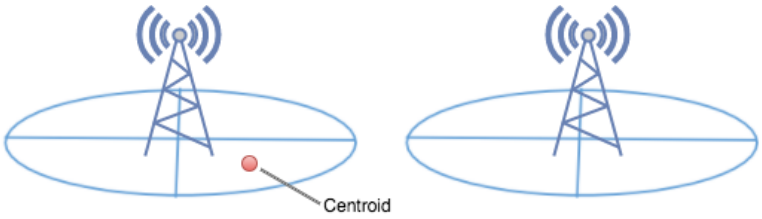


Figure 2.1: Mobile communication network

Mobile communication networks can be viewed as an infrastructure to collect mobility data, if it is used to gather the location of its users [8]. Mobile communication networks are made up of base stations and mobile cells. Each base station contains multiple cells which are scattered into smaller geographical areas in every direction of the station [10]. The center points of these areas are called centroids and are often used as the location of the cell. Figure 2.1 displays two base stations with four cells each. The size of the coverage area for the base station depends on the density of the base stations in the area.

The users connect to the network through their mobile phone, and if the user leaves the area of a cell, it directly connects to another cell controlling the new area it enters. The mobile phone and the base station communicate with each other through the exchange of radio signals. If the phone is connected to the network, the location of the user is available within the size of the cell and can be recorded by the mobile operator companies.

Trajectories

Trajectories can be used to gain knowledge from mobility data for decision-making purposes. In [8], a *trajectory* is defined as the path made by a moving entity through space where it moves over time and can be viewed as pairs consisting of location and time. Moving entities is a broad term that can define different areas, for example, fish in a stream, weather phenomena, vehicles, and of course people.

Human trajectories can be divided into active and passive recordings [11]. An active recording is when people purposely track their movements, for example tracking a jog or a bike ride from a mobile application to evaluate the workout, or by tagging locations on photos or posts in social media. A passive recording is when the mobile device is connected to mobile cells and produces spatial data streams that are collected by mobile operators. However, passive recordings can also include transaction data from using for example credit cards.

Characteristics of Trajectories	
Shape of trajectory	Travelled distance
Duration of trajectory in time	Mean, median and maximum speed
Movement vector or direction	Dynamics of speed
Dynamics of directions	

Table 2.1: Characteristics of Trajectories

To be able to analyze these trajectories and group them, it is beneficial to study the characteristics. Characteristics of trajectories are displayed i figure 2.1 [8]. Extracting knowledge from mobility data and trajectories is discussed in the following section.

2.2 Data Mining

Data mining is the process of discovering useful information in large data repositories. It consists of multiple techniques that are used to find new or unusual patterns, that might otherwise stay unknown [2]. The most common data mining techniques include classification, association analysis, and clustering. Data mining is a part of knowledge discovery in databases (KDD) which is the process of converting raw data into useful knowledge. KDD consists of three steps including data preprocessing, actual data mining and post-processing. Data preprocessing transforms the raw data into a suitable format for data analysis, and post-processing involves ensuring that only valid and useful results are incorporated into the decision support system [2].

Even though data mining techniques have proved to be powerful and efficient, new types of data combined with more complex and heterogeneous data require new ways of exploring patterns in data. Mobility data and other forms of spatial data are examples of data forms that might require additional methods and algorithms to extract patterns and knowledge. This is data that have a reference to location, and where the data typically arrives into systems as streams of logs from the cells in mobile networks.

Mobility Data Mining
Mobility data mining refers to the analysis of mobility data by using appropriate patterns and models found by effective algorithms. In addition, it aims to create a new knowledge discovery process for analysis of mobility regarding geography. The Geographic Knowledge Discovery (GKD) process is similar to KDD but is the process of converting raw mobility data into useful knowledge. GKD consists of three main steps, trajectory reconstruction, knowledge extraction and knowledge delivery [8], and is explained below.

1. **Trajectory Reconstruction** aims to restore the individual movements contained in the raw mobility data. This part can be challenging, and the quality of the results is highly dependent on the quality of the mobility data.

2. **Knowledge Extraction** deals with extracting knowledge from the reconstructed trajectories by using data mining techniques. The first technique is classification, which can be used to group similar trajectories. Grouping similar trajectories can give insight in paths taken by people during the day and may give knowledge that can aid in for example traffic optimization. Another method is to find frequent patterns, to for example disclose subpaths. Lastly, classification can be used to predict new paths.

3. **Knowledge Delivery** is the process of interpreting the extracted patterns, with the help of background knowledge of the geographical area. Visualizations and other forms of presentations are used to support the interpretation.

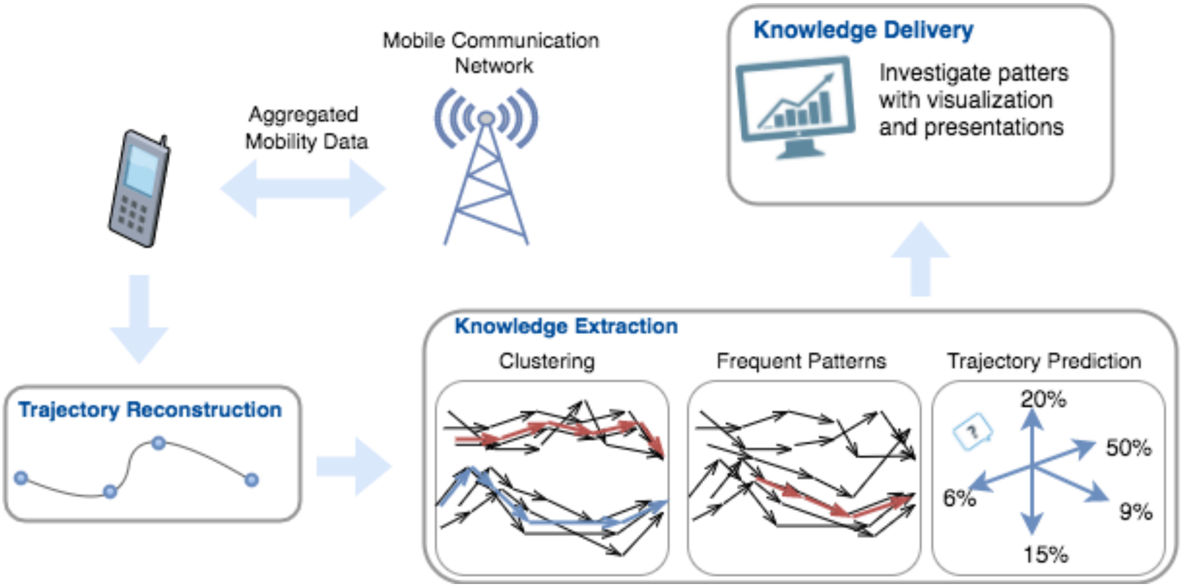


Figure 2.2: Geographic Knowledge Discovery

The GDK process, summarized in figure 2.2, will result in support and input to decision making in different areas. For example, in planning traffic systems like new roads or railways, deciding where to locate new services, and detecting change and problems due to movement behavior [8].

2.3 Correlation

Pearson correlation is a measure of linear correlation between pairs of continuous variables x and y [3]. Correlation can have values between $[-1, 1]$, where the sign indicates the direction of the relationship, and the value indicates how strong the relationship is. Equation 2.1 display the Pearson correlation, where x, and y are continuous variables, and Cov is covariance and Var is variance

$$r = \frac{Cov\left[X, Y\right]}{\sqrt{Var\left[X\right] Var\left[Y\right]}} \tag{2.1}$$

The Spearmans correlation is a measure of the monotonic relationship between pairs of continuous variables [9]. A monotonic function is one that either never decreases or increases as its variables increases.

$$p = 1 - \frac{6 \sum d_i^2}{n\left(n^2 - 1\right)} \tag{2.2}$$

Spearmans Correlation is a rank correlation, which means the first step is to find the ranks of the variables. If there are no tied ranks, equation 2.2 is used, where d is the difference in the ranks, and n is the number of variables. When there are tied ranks, Pearson correlation is used with the Spearman-produced ranks as variables.

Correlation can be used as an important tool when working with machine learning. This thesis later presents a method for estimating unknown values in a vector of counts. The correlation techniques are used to build an effective training model with vectors similar to the one being estimated.

Chapter 3

Related Work

This chapter presents the state-of-the-art approaches for trajectory recovery from aggregated mobility data, in addition to privacy research in this field. There are multiple techniques to protect the privacy, and finding the best techniques is not trivial. This chapter firstly presents privacy preserving researches, where different techniques is discussed. Secondly, the research for trajectory recovery from aggregated mobility data and the Hungarian algorithm is presented.

3.1 Aggregated Mobility Data Privacy

With the new EU regulations, GDPR [1], data privacy has never been more relevant. The mobility data is no exception. Storage of data with individual movements is a clear violation of privacy laws. To collect useful information from these types of data without breaking any laws, the data could be aggregated, so that instead of showing personal movements, the data displays public movements.

3.1.1 Differential Privacy

Differential privacy is a collective term for techniques that provide high accuracy for queries from statistical databases and a low chance of identifying the records in the database at the same time. The research in [4], presents an analysis evaluating aggregation-based location privacy, where different privacy protection techniques are tried on sets of aggregated data. To do so, a framework attempting to recover user's mobility patterns was introduced, to test the privacy against an attacking adversary.

Their evaluation of differential privacy shows that the performance of both privacy and utility are highly dependent on the characteristics of the dataset. A sparse dataset over a high number of locations gives good results with techniques like data perturbation, while a denser matrix, containing fewer locations, barely gives any privacy protection. Their analysis shows that it is difficult to achieve a good utility on continuous data and that the trade-off between privacy and utility is generally challenging to tune. Data preprocessing techniques, like clustering and sub-sampling, are proposed to gain better utility with the differential privacy mechanisms, but as this is not a generalized solution, it would not fit all applications.

3.1.2 K-anonymity

K-anonymity is one of the most used location privacy technique. To hide the users, k-anonymity shows minimum k users at a time step and location area. The definition is that it should not be possible to distinguish one user from k - 1 other users. For aggregated mobility data, this means that if an aggregated value is lower than k, the value will be set to k to achieve k-anonymity.

In the research presented in [5], k-anonymity is analyzed based on the effectiveness of the approach for protecting location privacy. In the analysis, they argue that this technique is insufficient in giving users location privacy. An example of this is situations when a high number of users are present in a small location, then even with k-anonymity, the exact positions of every user can still be recovered. However, it is important to note that their approach of k-anonymity is based on increasing the size of the location or time until k users are within the same region, rather than using fixed regions while hiding the exact values contradicting the k-anonymity.

3.2 Trajectory Recovery from Aggregated Mobility Data

As more and different data becomes available for the public, different approaches to exploit these data occurs. Xu et al. [6], conducted experiments showing that it is possible to recover 73-91% of the personal trajectories on a set of aggregated mobility data. A longstanding theory states that aggregating the data is sufficient for concealing the personal trajectories and preserve the privacy of users. However, new attack methods have proven that this is not enough. Further techniques are needed to create noise in the data and disrupting the aggressive trajectory recovery attacks. However, this has to be completed without damaging the aggregated data, which would not have broken any privacy regulations without the presented attack-methodology.

The framework in [6] is described as an “unsupervised framework that leverages the universal characteristics of human mobility to recover users trajectories from aggregated mobility data without any prior knowledge”. Universal characteristics of human mobility are divided into three parts: At nighttime, when people usually stay in the same place for several hours, at daytime, when people often move more frequently, and across the days, when connecting trajectories with other trajectories from different days.

The first step is to create a matrix where every row element is a human trajectory, and every column is a possible next position based on the aggregated data from the following time step. The values of the matrix describe the likelihood for a row to be combined with the corresponding column. To calculate this likelihood, the universal characteristics of human mobility are used. The large cost matrix represents an assignment problem, that is solved by using the Hungarian Algorithm. The main focus is to exploit the mobility characteristics to create an accurate cost matrix. As previously mentioned, they used three different approaches to create the cost matrix:

- **Nighttime Trajectories:** The main idea for recovering nighttime trajectories is that people usually stay in the same position at nighttime. The results show that between 12:00 am and 06:00 am, 62-88% is located in the same base station. Additionally, it showed that this base station is almost always their most frequently visited. With these observations, they built the cost matrix mainly on the distance between locations. In that way, most trajectories would find themselves at the same position throughout the night, while just a few would create movement.

- **Daytime Trajectories:** For recovering the daytime trajectories, when people move more frequently, a different strategy is needed. The use of velocity was incorporated into the method. The velocity is derived from finding the distance between the two previous points. This measure is then added to the last known point to create a fictional point. The likelihood of presence in a cell is then calculated by finding the distance between the cell and the fictional point. With this modification, the cost matrix would increase the likelihood of moving people to continue their direction and pace, making it easier to track users in cars, trains and other vehicles, which are frequently used in the daytime scenario.

- **Across the Days:** The final step is to recover trajectories across days, which means connecting the recovered trajectories to others at different days. To associate trajectories from different days that belong to the same person, they use information gain of two connecting trajectories to measure similarity. The information gain measures how different two trajectories are, based on the frequency distribution of the different cells from the trajectory. After calculating information gain between trajectories, they obtain an optimal match problem, which needs to be calculated. Once that was solved, the full trajectory of each user was recovered with an accuracy of 73-91%

To evaluate the system, Xu et al. [6] used mobility data collected from two different sources. The first dataset is collected using a mobile application that stores the user's positions when the device is active. The data stored includes accurate coordinates of the user's position, anonymized user identification, accessed base station and the timestamp. This dataset contains on average 496 records from 15,500 users.

The second dataset is collected from the mobile communication network and contains coordinates of the base station the user is connected to, anonymized user identification and timestamp. This dataset contains on average 261 records from 100,000 users. Both datasets are collected from a major city in China in an area with over 8000 base stations.

Hungarian Algorithm

The Hungarian Algorithm is an algorithm for solving the assignment problem. The assignment problem appears when trying to find the optimal combinations of two groups. A classic example is the task of assigning workers to jobs. Each worker does each job at a different time, and the task is to find the most efficient job distribution. For trajectory recovery, the task is to assign people at time stamp t to people at time stamp t+1, creating one step at a time of a final trajectory. The Hungarian Algorithm solves the assignment problem in polynomial time, with a complexity of $O(n^3)$. The algorithm contains the following four steps [12]:

1. The first step is to make sure that every row contains a zero-element. For each row, find the element with the lowest value, and subtract this value from every element in the row.

0.9	1.0	0.3	0
0.1	0.2	0.5	0.1
0	0.8	0.8	0.3
0.3	0	0.2	0.2

$\Rightarrow$

0.9	1.0	0.3	0
0	0.1	0.4	0
0	0.8	0.8	0.3
0.3	0	0.2	0.2

2. The second step is to make sure that every column contains a zero-element. The method is the same as in step one.

0.9	1.0	0.3	0
0	0.1	0.4	0
0	0.8	0.8	0.3
0.3	0	0.2	0.2

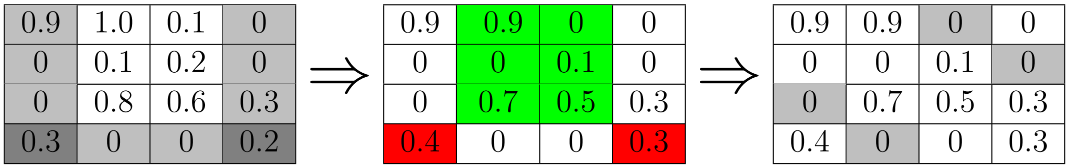
$\Rightarrow$

0.9	1.0	0.1	0
0	0.1	0.2	0
0	0.8	0.6	0.3
0.3	0	0	0.2

3. The third step is to find the lowest amount of horizontal and vertical lines needed to cover every zero-element in the matrix. There are different approaches to how this can be completed. If the number of lines is the same as the number of rows/columns in the matrix, the algorithm stops here. However, if it is possible to cover the zero-elements with fewer lines, the algorithm loops through the fourth step until it is not possible anymore.

0.9	1.0	0.1	0
0	0.1	0.2	0
0	0.8	0.6	0.3
0.3	0	0	0.2

4. The fourth step is to update the matrix for a new iteration of the third step. For all elements which are not covered by lines from step three, find the element with the smallest value. This value is subtracted from all the non-covered elements and added to all the elements which are covered by both a horizontal and a vertical line. The matrix is now updated and step three runs with the new matrix.



0.9	1.0	0.1	0
0	0.1	0.2	0
0	0.8	0.6	0.3
0.3	0	0	0.2

$\Rightarrow$

0.9	0.9	0	0
0	0	0.1	0
0	0.7	0.5	0.3
0.4	0	0	0.3

$\Rightarrow$

0.9	0.9	0	0
0	0	0.1	0
0	0.7	0.5	0.3
0.4	0	0	0.3

For the experiments conducted in [6], the Hungarian algorithm was used to assign trajectories to possible new positions. However, when n becomes very large, an algorithm with complexity $O(n^3)$ tends to use much time. To cope with this, they used a suboptimal solution, where pairs in the cost matrix with values below a threshold was directly linked together. This approach made the matrix significantly smaller, which gave a reduced running time as well as keeping the excellent performance.

3.3 Summary

There is a significant amount of work invested in the topic of data privacy. As there continuously appears new methods to exploit sensitive data, the solution is often to add a new layer of protection to the data. This thesis tries to utilize the attack method presented in 3.2 and transform it into a framework for determining the optimal value of k. With the framework, the protection layer of k-anonymity will be optimized to keep as much of the utilization of the aggregated mobility data as possible.

In difference to the related work, this thesis contributes to improve and optimize the trajectory recovery method, as well as applying k-anonymity to further protect the data. The attack model is also extended to challenge the applied k-anonymity. Finally, a new evaluation method, using synthetic trajectories is utilized as the original trajectory data is not available.

Chapter 4

Approach

This chapter firstly describes the Mobility datasets utilized in this thesis. Next, this chapter presents the approach to recover individual trajectories from aggregated mobility data and finding the optimal value of k for k-anonymity to sufficiently protect it. The approach consists of data preprocessing, where the steps to make the data ready for trajectory recovery is described, the trajectory recovery method, which presents a detailed description of the Hungarian algorithm and how it is modified to fit the data in this thesis, and finally, the determination of the optimal k-value is presented.

4.1 Mobility Datasets

To evaluate the methods data collected from the three largest cities in Norway in addition to the main Norwegian airport is utilized. This section presents the initial exploration of the datasets used for this project including initial data analysis results, characteristics of the data and visualizations.

4.1.1 Data Features

The mobility datasets are collected from the largest mobile network operator in Norway and contains aggregated mobility data from the region of Oslo, Bergen, Trondheim, and Gardermoen. The dataset contains hashed ID of base-stations, coordinates (cell easting and cell northing) with consistent noise of the base-stations, count and collect time of the data. The cell easting and cell northing are applied to measure latitude and longitude. The format of the data is displayed in table 4.1.

As stated in section 2.1, mobile communication networks are built up of base stations and mobile cells, where each base station contains multiple cells which represent smaller geographical areas. The data utilized in this thesis only contains base stations and no reference to the cells it contains. This means that the positions in the data are not as accurate as if the positions of the cells were included as well since a base station can cover a large area.

V1	count	cell_easting	cell_northing	lat	long	collect_time
1	452	277431	6677140	10.987	60.170	2018-01-08 12:45:00
2	349	277431	6677140	10.987	60.170	2018-01-08 12:45:00

Table 4.1: Data format

To get a better grasp of what the data contains, an initial analysis of the data was conducted concerning different features, the results are summarized in table 4.2.

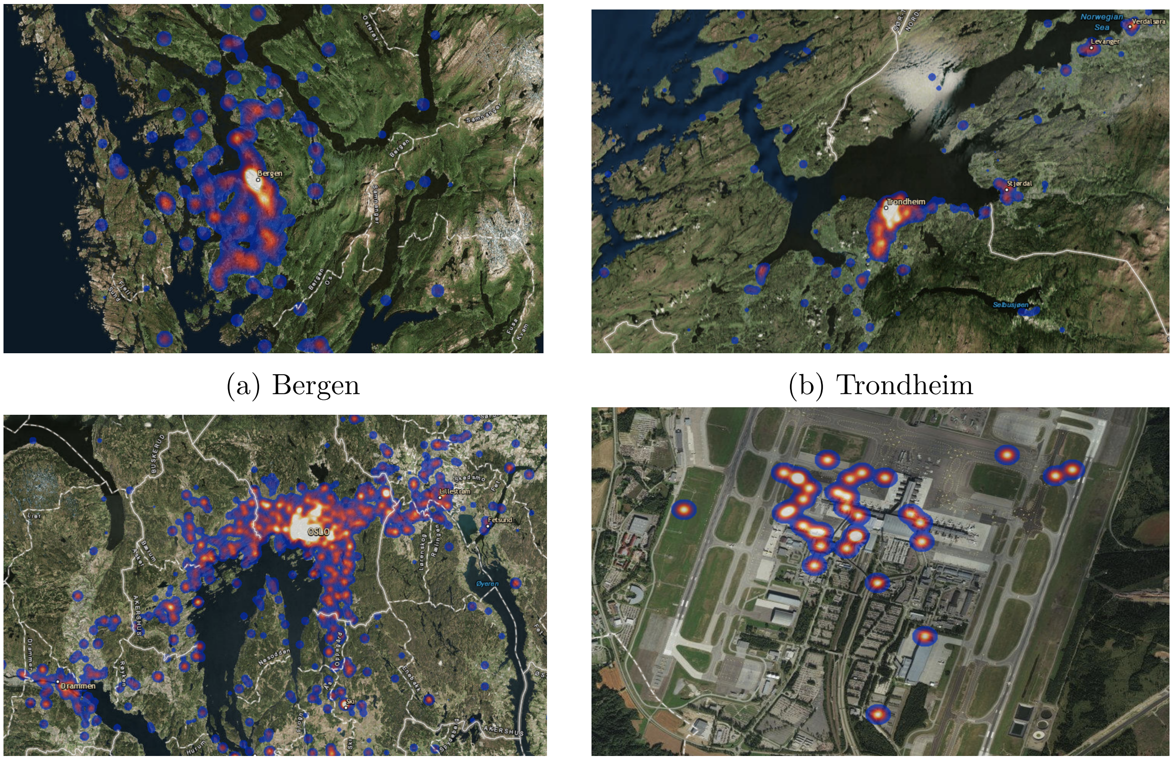
Features	Oslo	Bergen	Trondheim	Gardermoen
Size of data	856 MB	250 MB	206 MB	284 MB
Number of records	10 million	2.7 million	2.6 million	4.5 million
Duration	3 Weeks			
Time Interval	60 Minutes			5 Minutes
Radius	80 Km			2.5 Km
Number of base stations	1998	615	589	36
Aproximate population	1.3 million	356 000	246 000	20 000
Avg. dist. between base stations	41km	42km	45km	1,2km
Avg. dist. to nearest base station	38m	85m	138m	188m

Table 4.2: Data Features

The datasets vary in both size and features. However, the most dominant difference is where the data is collected. Mobility data collected from cities have different characteristics than mobility data collected from airports. At airports, the placement of base stations is denser, and the number of people is lower. Besides, the moving patterns are different. A typical movement in the city will be from a residential area to a workplace or university with different stops in between, while a typical movement on an airport will be to or from a gate to the entrance.

4.1.2 Visualization

Data visualization is a method to explore the characteristics, patterns or trends of the data, by presenting the data graphically. In addition, it may result in deciding additional techniques for preprocessing and analysis.



(a) Bergen

(b) Trondheim

(c) Oslo

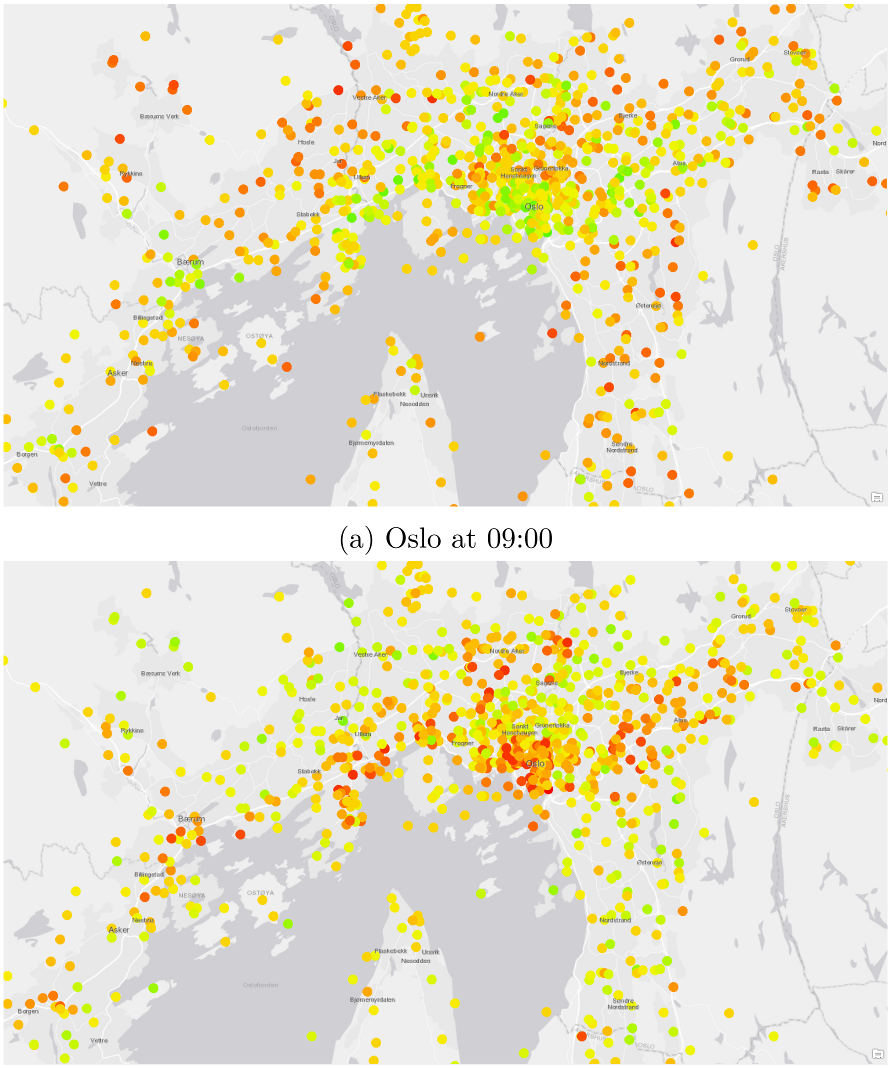
(d) Gardermoen

Figure 4.1: Heat maps representing the four locations

The suitable technique for visualizing the data highly depends on data types and context. Since the data has spatial content, it makes sense to visualize the data on a map. However, other aspects are not suitable for map visualization, for example, displaying counts over a period of time is more straightforward to interpret on a line diagram.

Figure 4.1 displays heat maps for the areas surrounding Bergen, Trondheim, and Oslo in addition to the base stations inside the Oslo airport Gardermoen. The heat maps are created by summarizing the counts for each base station. For each city, it is clear that most people are located in the city center, while people are more evenly distributed in the airport.

To show a typical movement pattern, figure 4.2 displays the change in counts for each base station at 09:00 and 18:00 in Oslo and the surrounding area. The colors are green when there is a significant positive change and gradually turns red as the change reaches a significant negative change.



(a) Oslo at 09:00

(b) Oslo at 18:00

Figure 4.2: Oslo in the morning and evening

For figure 4.2a the dark red and orange points suggest residential areas, and the green and bright yellow points suggest workplaces and industry areas. This figure displays the pattern of people leaving their houses and arriving at their workplace. Figure 4.2b shows the opposite pattern, and displays the pattern of people leaving their workplace and arriving back home.

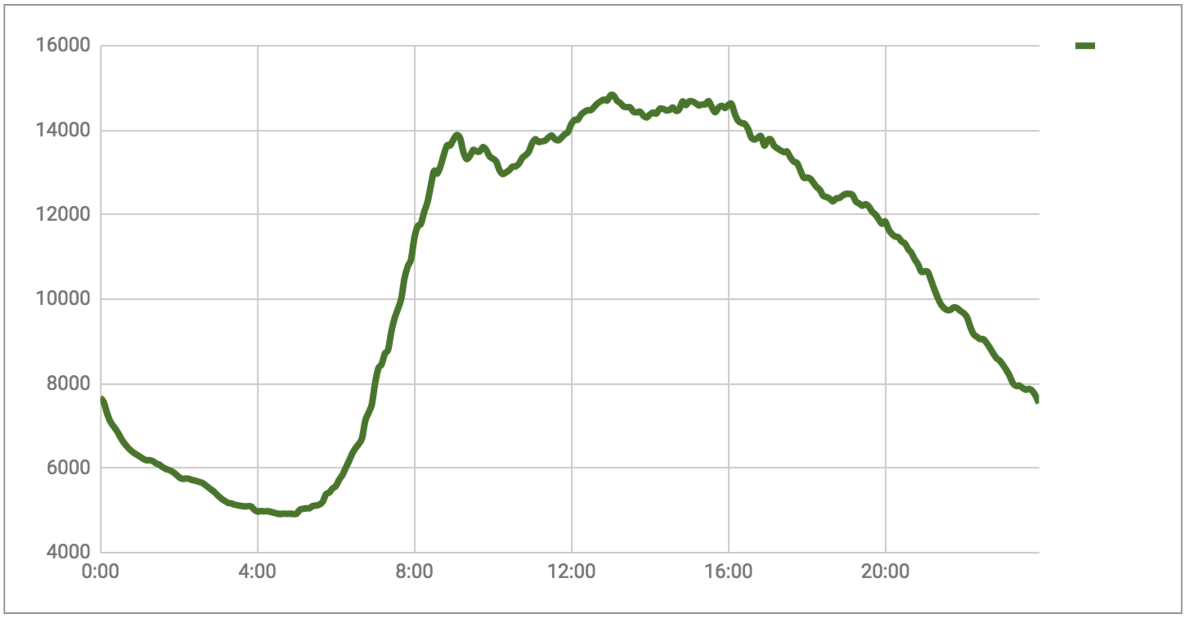


Figure 4.3: Sum of counts at Gardermoen

Figure 4.3 shows the number of people that are present at the Gardermoen airport and the surrounding area during one day. As seen, the number of people raise around 06:00, which corresponds with the earliest departures and arrivals. The count is relatively constant during the day before it decreases at night. The people that are present during the night are likely people staying at airport hotels or people that live in the area around Gardermoen.

4.2 Data Preprocessing

This section presents the preprocessing techniques performed on the datasets before the process of recovering individual trajectories can start.

4.2.1 Data Transformation

The first step of the data preprocessing process is to extract the necessary fields from the dataset, and transform it into a set format to fit the following algorithms.

The data transformation results in a dataset where each row consists of a base station ID followed by all the associated counts for that station. The counts are indexed in accordance with the date, and all missing counts are set to zero. With this alteration, the dataset is reduced by about 70%. In addition, a transformation back to the original format is still possible, without losing any valuable information. The dataset after the data transformation is presented in table 4.3

ID	0	1	2	3	4	...	499	500	501	502	503
1	24	23	23	25	24	...	26	28	27	28	25
2	132	135	134	134	133	...	144	144	142	140	139

Table 4.3: The dataset after data transformation

4.2.2 Error Handling

When collecting a significant amount of data, irregularities in the received data often occurs. Regular problems include missing time steps and inconsistencies in variable types across the columns. The system to recover trajectories uses data from two and two subsequent time steps to find the original trajectory, and a breach in data quality will possibly prohibit the method to work. To cope with these issues an error handling method checks the datasets before the trajectory recovery methods start.

The first step is to locate missing time steps and exclude the selected day from the dataset. As the system finds day trajectories before binding them together, removing a complete day should not provide the same problems as missing time steps. With the dataset on the form described in 4.2.1, the discovery of missing time steps proved to be a simple task. By adding every vector together, and a quick search for zero values through the sum-vector shows all missing time steps. Then, the method filters out these days from the dataset, and the following methods can run without problems.

The second step is to look for irregularities in the variable types in the columns. The column check will first identify the variable type for the column, and then check if the column has rows that are not consistent with that variable type. If this is the case, the system will print an error message so the user can evaluate what to do next.

4.2.3 Synthetic Trajectories

The final step of the preprocessing process is to add synthetic trajectories to the aggregated mobility data. As described in Section 1.3 there is no direct way to check if the recovered trajectories are correct because of privacy concerns. To solve this issue synthetic trajectories are added to create a set of trajectories that can be used to asses the accuracy and quality of the recovered trajectories.

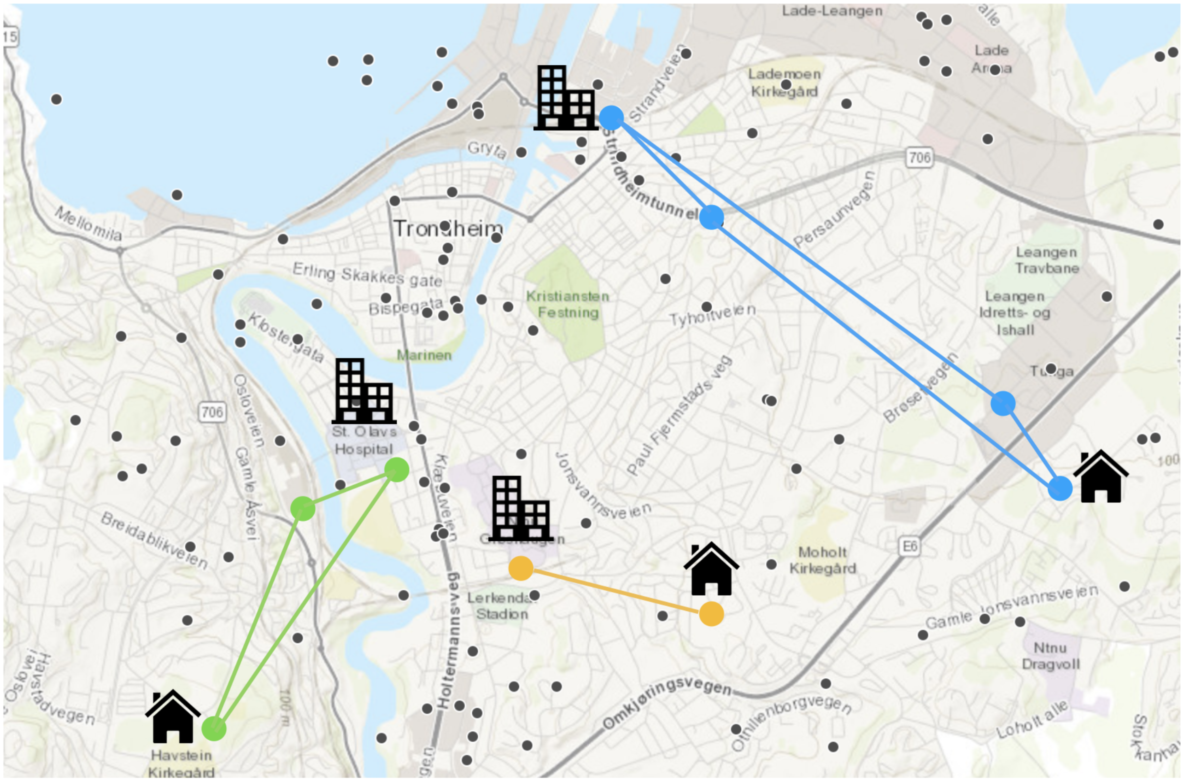


Figure 4.4: Synthetic Trajectories

Synthetic trajectories are trajectories that are created computationally to act like real trajectories. They are created to represent the same mobility characteristics, previously described in Section 3.2, that real trajectories have. A regular person will usually leave their residence in the morning, then travel through a couple of new base station areas before arriving at the place of work. Then take about the same route on the way home from work in the evening. This pattern is the base for the synthetic trajectory construction. A visual representation of these patterns is presented in figure 4.4, where common resident areas and work areas are shown.

The first step of generating synthetic trajectories is to identify resident areas and work areas. This is accomplished by looking at the number of people that are connected to base stations during nighttime and daytime. If a base station has a higher count during the night than the day, it is characterized as a residential area. Conversely, if a base station has a low count during the night and a high count during the day, it is characterized as a work area.

To create the synthetic trajectory, the method picks a random base station from the list of residential areas and a random base station from the list of workplace areas. The maximum possible distance between these areas is set to five kilometers. This is to prevent the trajectories from traveling from one end of the area the other, as this movement is unlikely. Even though there will always be commuters who travel a lot longer, this five kilometer mark is needed to prevent the dataset to have a high amount of long travels. The higher the distance mark is set, the chance of two randomly selected base stations to be very far apart rather than close to each other increases.

The velocity of travel is also chosen at random to represent both travels by foot, car or public transport. The final synthetic trajectory then includes a resident area base station, the base station(s) the person travels through to get to the workplace, the workplace and about the same route on the way home.

4.3 Trajectory Recovery

This section presents the approach for implementing the algorithm for reconstructing individual trajectories for all users in aggregated mobility data. The algorithm is highly based on the approach described in section 3.2, however, modifications were necessary to suit this scenario.

4.3.1 Cost Matrix

The algorithm loops through every time step in the dataset, while calculating which location is most likely to belong to each trajectory. This is completed by using a cost-matrix with relations between trajectories and location in the next time frame. To build an efficient cost matrix, it is important to use the mobility characteristic, meaning the known movement patterns to improve the performance of the model.

An essential aspect of the cost matrix is its size. The Hungarian algorithm is known to have a complexity of $O(n^3)$. The datasets have between 500 and 2000 base stations, which summed up, points to several hundred thousand individuals. The cost matrix from the smallest dataset, Trondheim, would be 246000 × 246000, which gives over 60 billion matrix entries and complexity in the quadrillion order. With the execution time of that matrix, it would not be a realistic approach, so as in section 3.2, this approach uses a sub-optimal solution to enhance the execution time.

Firstly, instead of creating one column for each person in the next location, this approach has one column for each position. This makes the horizontal dimension 100 times smaller but leads to modifications in the algorithm. With this dimension modification, the Trondheim cost-matrix is 246000 × 589. This makes the matrix much smaller, but it still has over 100 million matrix entries. In addition, the modifications to the algorithm also have an effect. As the cost matrix only points to each base station in the columns, the number of people each station can acquire must be stored alongside. The Hungarian algorithm must also be modified, which is addressed in the following subsection.

As the method described in 3.2, this cost matrix also automatically assigns the best matches before the Hungarian algorithm runs. This is necessary to make the Hungarian algorithm possible to run, which is accomplished by sorting every matrix entry on the cost value, and assigning the lowest values before the algorithm runs on the rest. How much the algorithm automatically assigns is added as a variable and can be selected by the users depending on whether speed or accuracy is most important.

4.3.2 Hungarian Algorithm

As modifications were performed on the cost matrix to better the execution times, modifications were also necessary for the Hungarian algorithm. The approach mainly follows the original algorithm and the first- and the second step is precisely the same as described in section 3.2. As listing 4.1 shows, this is accomplished by finding the lowest value in rows and columns and then subtract this from every element from that row or column.

```
#Step 1
for row in row_matrix
    lowest = max
    for element in row
        if element.value < lowest
            lowest = element
    if lowest > 0
        for element in row
            element = element - lowest
Return row_matrix

#Step 2
for element in row_matrix
    if element < lowest(element.j)
        lowest(element.j) = element
for element in row_matrix
    if lowest(element.j) > 0
        element = element - lowest(element.j)
return row_matrix
```

Listing 4.1: Pseudo code of the first and second step in the Hungarian algorithm

The third step is to assign entries to rows while crossing out zero valued-elements with lines. As shown in listing 4.2, this is accomplished by going through all rows, assigning one row at the time. To find the optimal column for each row, the number of elements a column can assign is divided by the sum of each value of rows left containing a zero-element in the selected column. The value of a row is given by one divided by the number of zero valued-elements in the row. The highest valued column is selected, and the row is temporarily assigned to it, and a horizontal line is drawn as well. If there are no suitable columns, meaning that all columns with a one-value are filled, the row is marked as an empty row. Next, the algorithm goes through all empty rows and for each element with a zero the column is marked as assigned. Then, all rows which contain a zero in a marked column are unassigned, leaving all zero-values crossed out with either a vertical or horizontal line. These lines are returned to be used in the fourth step of the Hungarian algorithm.

```
#Step 3
for row in row_matrix
    sum = count (elements in row equal to 0)
    for element in row equal to 0
        verticals(entry.j) += 1.0 / sum
        create element in col_matrix
for row in row_matrix
    row_entries = new matrix
    for element in row
        if element.value == 0 and verticals(entry.j) != 0
                    and column(entry.j) != 0
            vertical_value =
                column(entry.j) / verticals(entry.j)
            entry = (entry.i, entry.j, vertical_value)
            add entry to row_entries
    if row_entries is not empty
        selected = highest(elements in row_entries)
        column(selected.j) -= 1
        sum = count(elements in row equal to 0)
        for element in row equal to 0
            verticals(entry.j) -= 1.0 / sum
        assignment(selected.i) = selected
        row_assigned(selected.i) = true
    else
        add row to empty_rows
for element in empty_rows
    if element.value == 0
        col_assigned(element.j) = true
for col in col_matrix
    if col_assigned(col.number)
        for element in col
            if assignment(element.i) == col.number
                row_assigned(element.i) = false
if some element in row_assigned == false
    assignments = null
return (row_assigned, col_assigned, assignments)
```

Listing 4.2: Pseudo code of the third step in the Hungarian algorithm

If all rows are assigned, the Hungarian algorithm is finished, and the assigned values are entering the trajectories. Then it finds the lowest value over zero of all non-marked values. This value is then added to all non-marked values and subtracted from every value marked by both a vertical and a horizontal line. The matrix is now updated, and the algorithm can start over from the third step.

```
#Step 4
lowest = max
for element in row_matrix
    if element.value != 0 and !row_assigned(element.i) and
                !col_assigned(element.j)
        if value < lowest
            lowest = value
for element in row_matrix
    if element.value != 0 and !row_assigned(element.i) and
                !col_assigned(element.j)
        element.value -= lowest
    if row_assigned(element.i) and col_assigned(element.j)
        element.value += lowest
```

Listing 4.3: Pseudo code of the fourth step in the Hungarian algorithm

4.3.3 Associate Sub-Trajectories Across Days

The method for trajectory recovery is constructed in a way that only finds trajectories for one day at a time. To create full trajectories the method needs to associate trajectories from different days with each other. From 3.2 this is completed by using the Hungarian algorithm. However, this matrix cannot be scaled down the same way as the night and day trajectories, and the matrix becomes very large. The calculation of information gain is not a simple task, and even just creating the matrix uses a fair amount of time.

Instead of using the cost matrix and the Hungarian algorithm, this approach goes through each day trajectory, and picks the trajectory from the next day with the lowest information gain, as long as the value is beneath a specified threshold. The threshold is then slightly raised until all trajectories are braided together. With this approach, the optimal assignments will not be found, but in the same way as the night- and day-trajectories, the sub-optimal solutions does enhance the execution time.

4.3.4 Estimating Values for *K*-anonymity

To be able to recover individual trajectories correctly it is necessary to have as accurate data as possible. When increasing the value of k for k-anonymity, the accuracy of the data decreases which affects the recovered trajectories. To improve the accuracy and imitate a possible attack system, a method for estimating values below k is necessary.

The estimation process starts by structuring the data. As the values between two time steps from the same base station are coherent, estimation based on the neighboring counts should work to estimate the missing values. To structure the data for the estimation phase, the data transformation from section 4.2.1 is utilized.

The estimation phase starts with identifying the rows that are similar to the row that needs estimation. To find similar rows, the correlation methods described in section 2.3 is utilized. Spearmans Correlation is based on rank and gives good results for similar areas with the approximately constant difference in count. However, as there are multiple tied ranks in the data, the estimation method calculated correlation based on Pearson similarity of the ranking data. The five most accurate matches for all insufficient rows contribute to estimating the new values for the row.

The actual estimation uses this five most similar rows to train a machine learning network. The machine learning network uses the counts from the previous ten time steps and the next ten time steps, and measure the change of count. The network trains on these situations, and if the next time step after a substantial decline is unknown, the method probably estimates this with a continuously decline. The machine learning technique utilized is the Gradient boosting three regression. The technique has been implemented through the Scala Spark Machine Learning library.

The estimated values are then used instead of the real values and precision is calculated. However, there are situations where every count from a base station is between zero and k. This makes the task of estimating values impossible, and it would not make sense to try. In these cases, the counts are set to $k - anonymity/2$.

4.4 Determine the New Value of *k*

After the trajectory reconstruction is completed, the next step is to determine the new value of k or decide if the current k-value is optimal. The output of the trajectory recovery method is a percentage of how many synthetic trajectories is recovered, and this is also the base for finding the new k-value.

Recovery percentage	***k*-value increment**
< 30	0
30 - 35	1
35 - 40	2
40 - 45	3
45 - 50	4
> 50	5

Table 4.4: k-value increment from recovery percentage

At what percentage of recovered trajectories should be to protect the data is not an easy question. From the article presented in section 3.2, the privacy is breached when recovery percentages are $73\% - 91\%$. The highest percentages acceptable should be way lower, but precisely what depends on many different factors, for instance, the uniqueness and the precision of the trajectories. However, since the system gradually increases the k-value, which then gradually decreases the recovery percentage, there is better to set the value rather low than high.

The system works with a maximum acceptable percentage of successfully recovered trajectories of 30%, which then implies that the dataset is safe against hostile attacks. The closer the recovery percentage is up to 30%, the lower gets the step length of the k-values increment. Table 4.4 shows this distribution.

Chapter 5

Results and Evaluation

This chapter describes the results and evaluation of the trajectory recovery and the k-anonymity optimization process. As stated in section 1.3 this master thesis is evaluated by the performance of the trajectory recovery method, the quality of recovered trajectories and the accuracy of the optimized value of k.

The chapter starts by introducing the mobility analytics service for k-anonymity and its features. Furthermore, it presents the results and evaluation of the trajectory recovery process including synthetic trajectories and the estimation of counts below k, and then, it presents the results and evaluation of the optimization of k for k-anonymity. Finally, the quality of the recovered synthetic trajectories is evaluated.

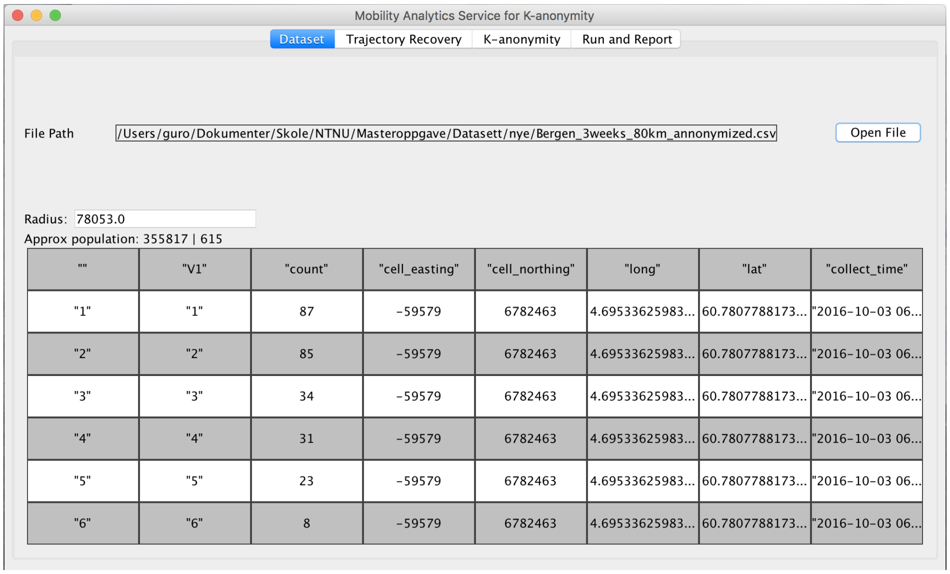
5.1 Mobility Analytics Service

This master thesis has resulted in a mobility analytics service for k-anonymity. This section shows each part of the service and explains the different components. In addition, it will discuss the performance of the system and issues regarding this.

5.1.1 User Interface

The Mobility analytic service is an application created in Scala swing with a simple and intuitive user interface, which aims to make it easier to input variables to the algorithm and see the results.

The first tab is dedicated to the dataset and can be viewed in figure 5.1. After opening the mobility dataset, a preview is shown in addition to the radius in meters around the location. This radius is editable, which makes it possible to reduce the size of the dataset and thus, make the runs quicker. The population and number of base stations contained within that radius are displayed below to make it easier to find the preferable size.



File Path /Users/guro/Dokumenter/Skole/NTNU/Masteroppgave/Datasett/nye/Bergen_3weeks_80km_annonymized.csv

Radius: 78053.0

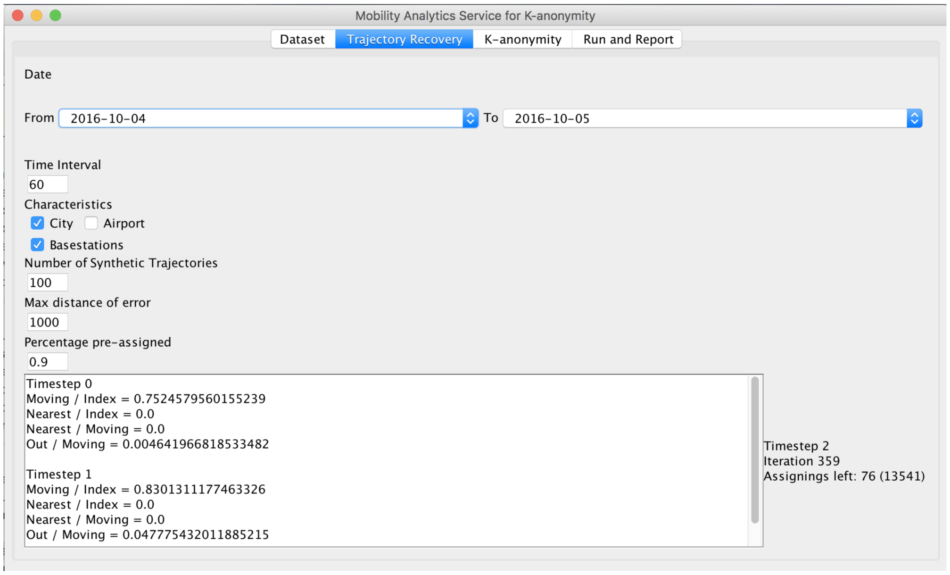
Approx population: 355817 | 615

""	"V1"	"count"	"cell_easting"	"cell_northing"	"long"	"lat"	"collect_time"
"1"	"1"	87	-59579	6782463	4.69533625983...	60.7807788173...	"2016-10-03 06...
"2"	"2"	85	-59579	6782463	4.69533625983...	60.7807788173...	"2016-10-03 06...
"3"	"3"	34	-59579	6782463	4.69533625983...	60.7807788173...	"2016-10-03 06...
"4"	"4"	31	-59579	6782463	4.69533625983...	60.7807788173...	"2016-10-03 06...
"5"	"5"	23	-59579	6782463	4.69533625983...	60.7807788173...	"2016-10-03 06...
"6"	"6"	8	-59579	6782463	4.69533625983...	60.7807788173...	"2016-10-03 06...

Figure 5.1: Dataset tab

The following tab is for trajectory recovery, displayed in figure 5.2. In this tab, the user enters all the necessary variables needed to run the trajectory recovery algorithm. The user can edit the dates, time interval and the number of synthetic trajectories to be added to the dataset. It is also possible to change the dataset between using base stations or cells if cells are present in the dataset. The max distance of error decides how close every point in the recovered trajectory needs to be the original trajectory to become successfully recovered. The last variable is the percentage pre-assigned. As described in section 4.3.1 the cost matrix needs to be decreased and the percentage of pre-assigned variable states how much of the matrix which will be processed before the algorithm runs.

During the run, the text box output statistics for each time step. This includes the percentage of moving people and the people that move out of the area covered by the dataset, as well as percentages for movement between cells if cells are applied. The progress of the trajectory recovery is also displayed here, which makes it simple to follow the process.



Date

From 2016-10-04 To 2016-10-05

Time Interval
60

Characteristics
☑ City □ Airport
☑ Basestations

Number of Synthetic Trajectories
100

Max distance of error
1000

Percentage pre-assigned
0.9

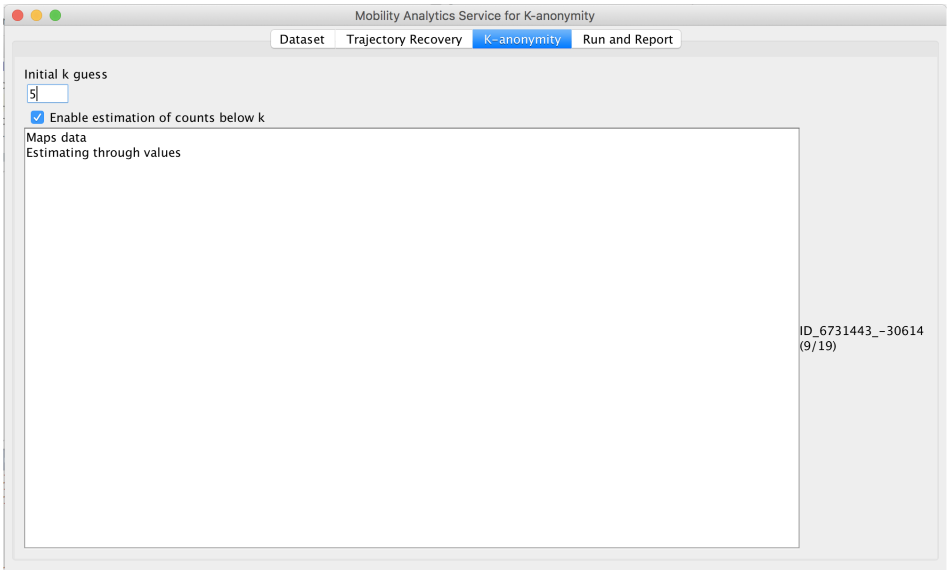
```
Timestep 0
Moving / Index = 0.7524579560155239
Nearest / Index = 0.0
Nearest / Moving = 0.0
Out / Moving = 0.004641966818533482

Timestep 1
Moving / Index = 0.8301311177463326
Nearest / Index = 0.0
Nearest / Moving = 0.0
Out / Moving = 0.047775432011885215
```

```
Timestep 2
Iteration 359
Assignings left: 76 (13541)
```

Figure 5.2: Trajectory Recovery tab

The following tab, displayed in figure 5.3, is for k-anonymity. This is where the user enters the initial k-value guess, which gives the algorithm a starting point when attempting to find the optimal k-value. The text box gives updates regarding the estimation process of counts below k and similar to the trajectory recovery tab, the estimation progress can be followed on this tab.



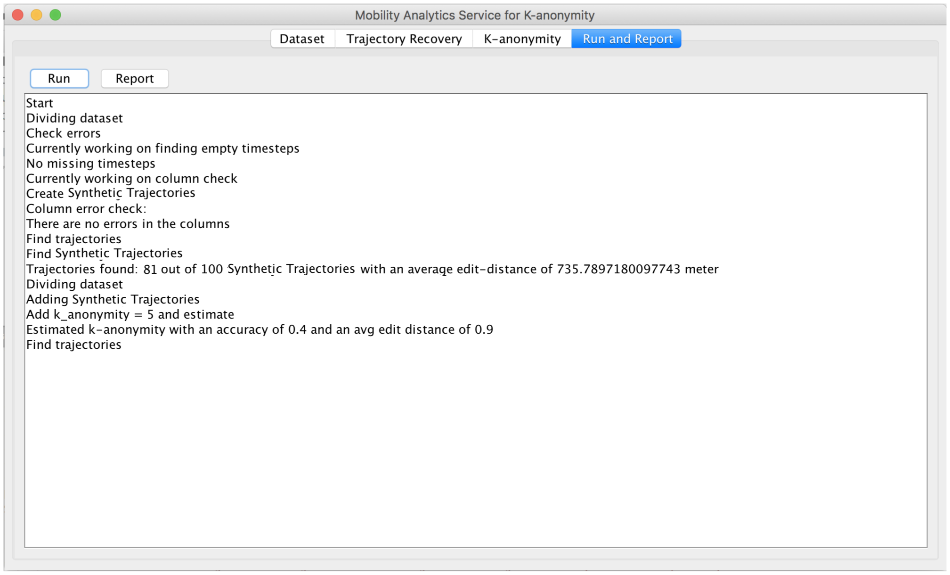
Initial k guess
5

☑ Enable estimation of counts below k

```
Maps data
Estimating through values
```

```
ID_6731443_-30614
(9/19)
```

Figure 5.3: k-anonymity tab



```
Start
Dividing dataset
Check errors
Currently working on finding empty timesteps
No missing timesteps
Currently working on column check
Create Synthetic Trajectories
Column error check:
There are no errors in the columns
Find trajectories
Find Synthetic Trajectories
Trajectories found: 81 out of 100 Synthetic Trajectories with an average edit-distance of 735.7897180097743 meter
Dividing dataset
Adding Synthetic Trajectories
Add k_anonymity = 5 and estimate
Estimated k-anonymity with an accuracy of 0.4 and an avg edit distance of 0.9
Find trajectories
```

Figure 5.4: Run tab

The final tab is run and report tab displayed in figure 5.4, which is where the user starts the algorithm and follows the progress. Information about what the system is currently doing and how well the method is performing appears in the text box, which makes it simple for the user to get control over the run. The report button prints the variable inputs that the user has set through the service.

5.1.2 Performance

Due to the high number of people in the areas surrounding Trondheim, Oslo, and Bergen, the trajectory recovery system struggles when using the complete datasets. Section 4.3.1 showed that with the whole Trondheim data set, the algorithm ends up with over 100 million matrix entries. As the resources available for this thesis could not handle such amounts of data, the datasets have been reduced quite a bit. Reducing the radius from 80km to 6km in the Trondheim dataset resulted in 163 base stations and approximately 90,000 people. This reduction creates a matrix with around 15 million entries, which the resources can handle. However, even with massive reductions the system still has a long completion time.

Because of the performance of the system, the results from the runs are concentrated to the Trondheim dataset, while the other datasets have been tested to ensure that the results correlate.

5.2 Synthetic Trajectories

This section describes the characteristics and visualize the synthetic trajectories used in the trajectory recovery system. As stated in section 4.2.3, the synthetic trajectories are created by picking random base stations from a residential area and a workplace area with a maximum distance of five kilometers and connecting them with possible stops in between.

Characteristics	Created	Recovered
Number of trajectories	100	54
Average length of trajectories	2.20	2.27
Average distance of trajectories	10.64 km	5.95 km

Table 5.1: Characteristics of synthetic trajectories from 4km around Trondheim over a two day period

Table 5.1 displays the characteristics of the synthetic trajectories that was initially created compared to the ones the system managed to recover. The data used for the table is from four kilometers around the city center of Trondheim over a two day period.

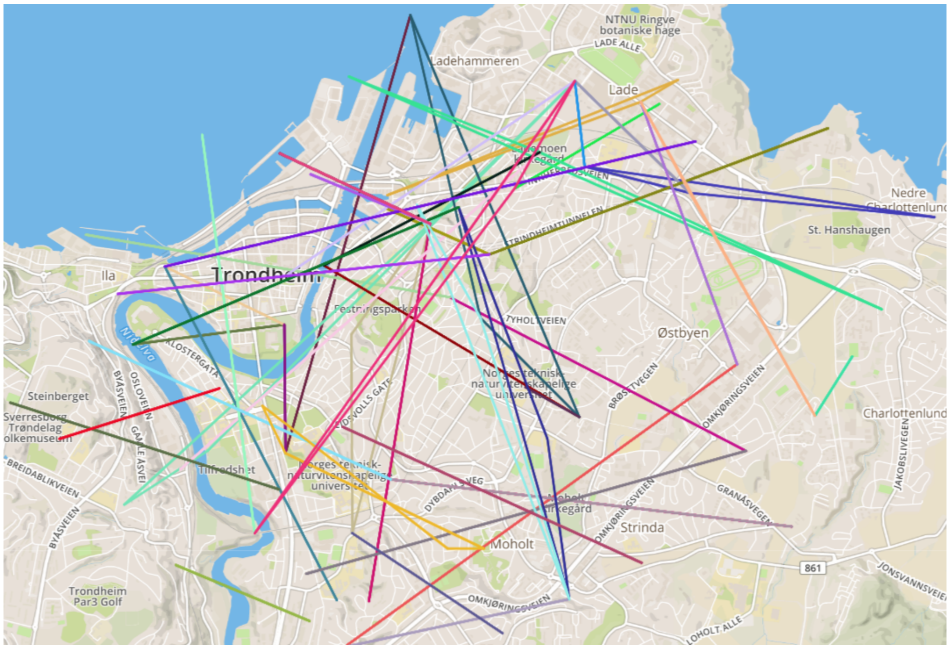


Figure 5.5: Synthetic Trajectories in Trondheim

While the length of the trajectories, meaning the number of distinct base stations it contains, are about the same, the average distance of the recovered trajectories is about half of the ones initially created. This suggests that the distance of the synthetic trajectories may be too long for the trajectory recovery system.

Figure 5.5 shows 50 synthetic trajectories created over four km radius around Trondheim. The trajectories which form a triangle, have a recording on the way to or from work. Only some of the trajectories have this because the precise time a person leave is randomly generated, and that can be right before a recording, which creates a point in the trajectory in between the home and workplace.

5.3 Estimation Model

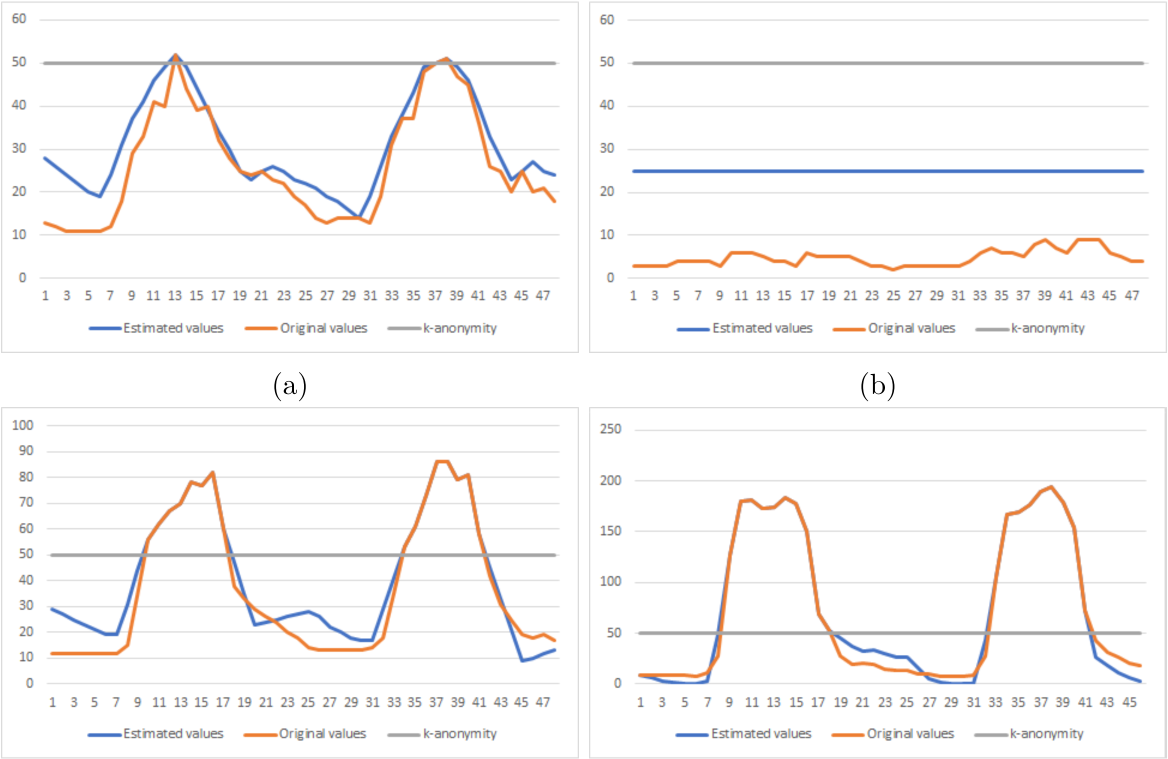
When different levels of k are applied to the dataset, the values below k are estimated by the estimation model, previously described in section 4.3.4. Table 5.2 shows how good the estimation model performs for the complete Trondheim dataset.

k	Estimation		Guessing		Results	
	Base stations	Counts	Base stations	Counts	Accuracy	Edit distance
5	26	453	0	0	0.35	1.04
10	37	688	3	144	0.22	2.06
15	47	1029	5	240	0.16	2.83
20	52	1249	8	384	0.12	3.69
25	62	1317	14	672	0.09	4.52
30	65	1497	18	864	0.09	5.15
35	82	1801	20	960	0.08	5.99
40	93	2155	23	1104	0.08	6.75
45	92	2217	33	1584	0.06	7.46
50	95	2468	39	1872	0.06	8.13

Table 5.2: Estimation results for the Trondheim dataset

The estimation column shows the number of base stations that contains counts that needs estimation and the total number of counts that are affected, for different levels of k-anonymity. The guessing column shows the same measures, but here the values represent counts where the estimation model did not have enough data to estimate the counts. This is when all counts for a base station are between 0 and k.

The last column shows how good the model is performing. The first measure is accuracy, which is the number of counts estimated correctly, and the second column is the edit distance, which measures the distance from the estimated counts to the actual counts. The results show that the estimation is not very accurate. However, the edit distance shows that the estimations are somewhat close to the actual value, which keeps the dynamic of the movements consistent. It is important to note that all the counts in the guessing column are set to $k - anonymity/2$, which affects the accuracy and edit distance of the estimation model.



(a)

(b)

(c)

(d)

Figure 5.6: Estimation results for the Trondheim dataset

Figure 5.6 displays the resulting estimation of counts when k-anonymity is equal to 50. The graphs show that even though the estimation is not accurate, the estimated counts follow the same movement as the original counts. Figure 5.6b displays an example of when a base stations does not have any counts within the k anonymity range, and therefore sets all counts to $k - anonymity/2$. Consequently, the counts are far off the original values. However, for the counts the model manages to estimate, the estimation model reduces the effect of k-anonymity.

5.4 Trajectory Recovery

This section presents the results of the trajectory recovery method. It starts by presenting different runs of the algorithm and evaluate the optimization of the k-anonymity process. Then, this section explains the characteristics of the recovered trajectories and show visualizations of the movements.

5.4.1 Optimization of k-anonymity

When recovering trajectories, the system first attempts to find the complete set of trajectories from the dataset. This set of trajectories are then compared to the added synthetic trajectories. The trajectories that are within the edit distance threshold set by the user is then successfully recovered. For the rest of this thesis, the successfully recovered trajectories are referred to as the recovered synthetic trajectories.

This section presents different runs of the trajectory recovery and optimization of k-anonymity algorithm. The multiple runs are displayed as graphs, showing the percentage of recovered synthetic trajectories, and the average edit distance in kilometers between the trajectories and the synthetic trajectories for different values of k.

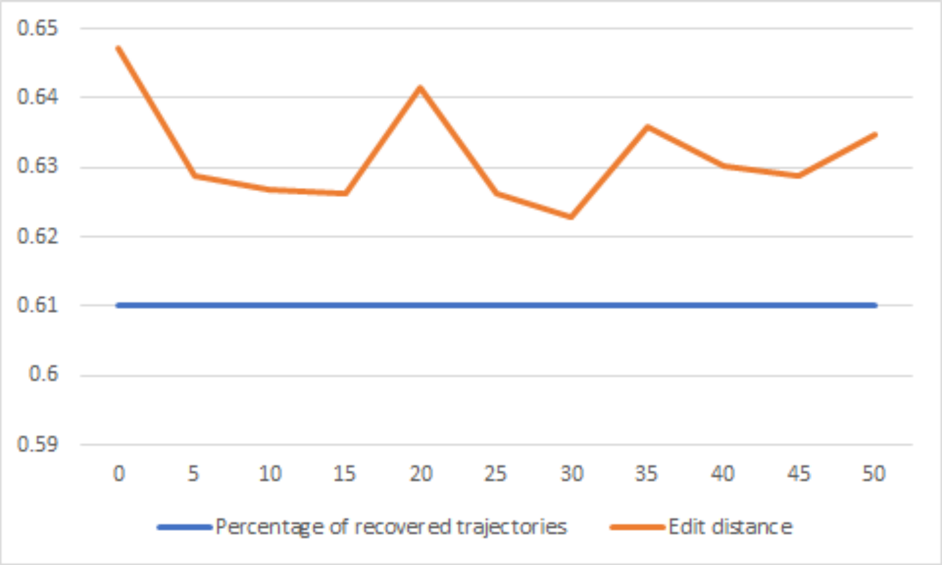


Figure 5.7: Trajectory recovery 4 km around Trondheim

The first run, presented in figure 5.7, displays the trajectory recovery for an area in Trondheim with four kilometers radius. The area contains 128 base stations and over 70,000 people. As the figure displays, the method successfully recovers 61% of the synthetic trajectories without k-anonymity. As the value of k gradually increases, the percentage of recovered trajectories stays at 61%, while the edit distance slightly varies. The value of k has no impact on the trajectory recovery in this run.

The next run, presented in figure 5.8, covers a slightly bigger area. With a radius of six km, 163 base stations, and over 90,000 people, the results of the trajectory recovery show approximately the same result as the previous run. The percentage of recovered trajectories stays at 50%, while the edit distance also stays consistent for the different values of k.

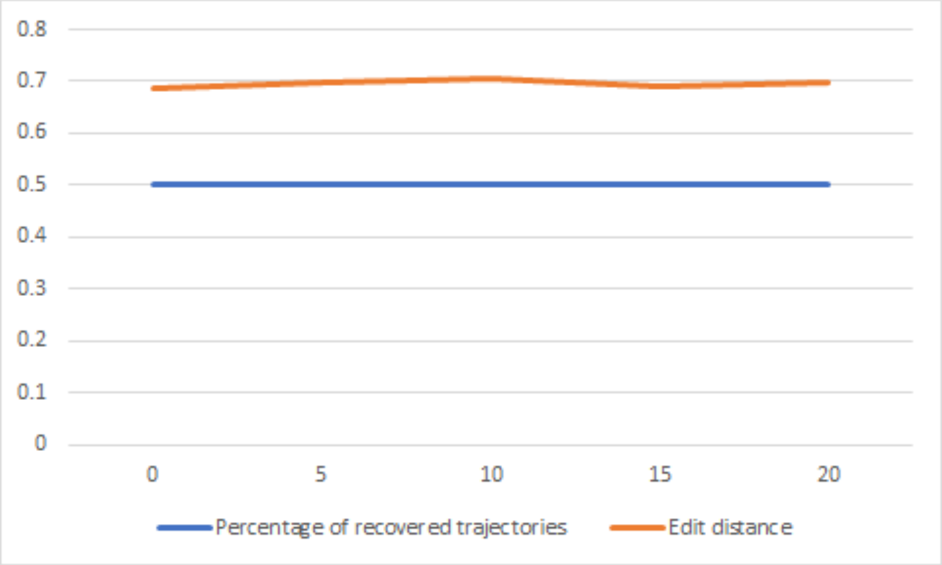


Figure 5.8: Trajectory recovery 6 km around Trondheim

For the next run, presented in figure 5.9, the same dataset as in the first run is used. However, this run is operating without the 70,000 people from the first run, by using only the synthetic trajectories. Utilizing only the synthetic trajectories results in a lot fewer people per base station, which would imply an easier task. The results, however, show that it recovers fewer trajectories with an approximately equal edit distance. The percentage of recovered trajectories varies more with a sparsely populated dataset. However, there is no apparent decline with a higher value of k, even though it is at its highest percentage with no k-anonymity.

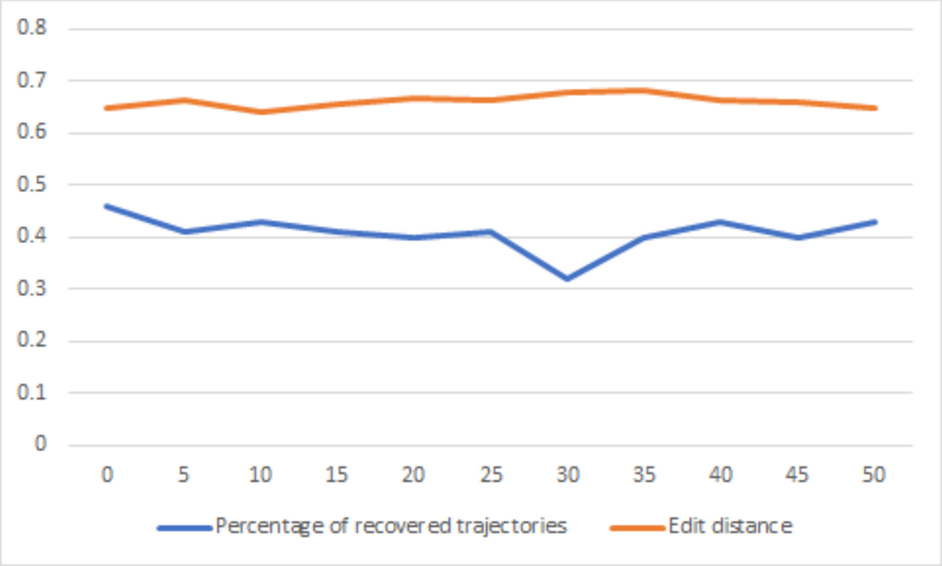


Figure 5.9: Trajectory recovery 4 km around Trondheim with synthetic trajectories only

As the results show, the percentages of successfully recovered trajectories stay almost constant while the value of k increases, which means the algorithm is not able to find an optimal value of k. Therefore there are no results of an optimal value of k for the different dataset in this thesis. This finding is discussed further in the next chapter.

5.4.2 Trajectory Characteristics and Visualizations

This subsection describes the characteristics of the recovered trajectories and shows visualizations of the trajectories.

Table 5.3 displays the characteristics of the synthetic trajectories recovered from a Trondheim dataset with a radius of four km. The results are for two days of trajectories where 54% of the synthetic trajectories are correctly recovered. As the table states, 39,441 of the recovered trajectories have a length of base stations equal to one, which signifies that 45% of the trajectories stay connected to the same base station throughout the two days. About 25,000 trajectories, or 29%, are moving out of the area covered by the dataset, which means it has one or more absences from the dataset within the two days. The table also shows that the average length of base stations is 1.72 and that the average distance traveled is 4.35 km, which means that most of the users travel relatively short distances throughout the two days. However, with the 45% of the dataset not moving in mind, these numbers are increased when removing the still standing trajectories.

Characteristics	
Number of recovered trajectories	87052
Number of trajectories with length equal to 1	39441
Number of people moving out of the dataset	25217
Average length of trajectories	1.72
Average distance traveled	4.35 km
Average length of moving trajectories	2.33
Average distance of moving trajectories	7.95 km

Table 5.3: Characteristics of the recovered trajectories

Figure 5.10 shows 100 trajectories recovered from a run in Oslo. The illustration shows that there are a lot of different movements, with most of them moving relatively short distances, while a few are moving quite large distances. Another observation is that most trajectories move between two base stations and not anywhere else during the two days span. There are some trajectories moving further, but the majority of trajectories functions like this.

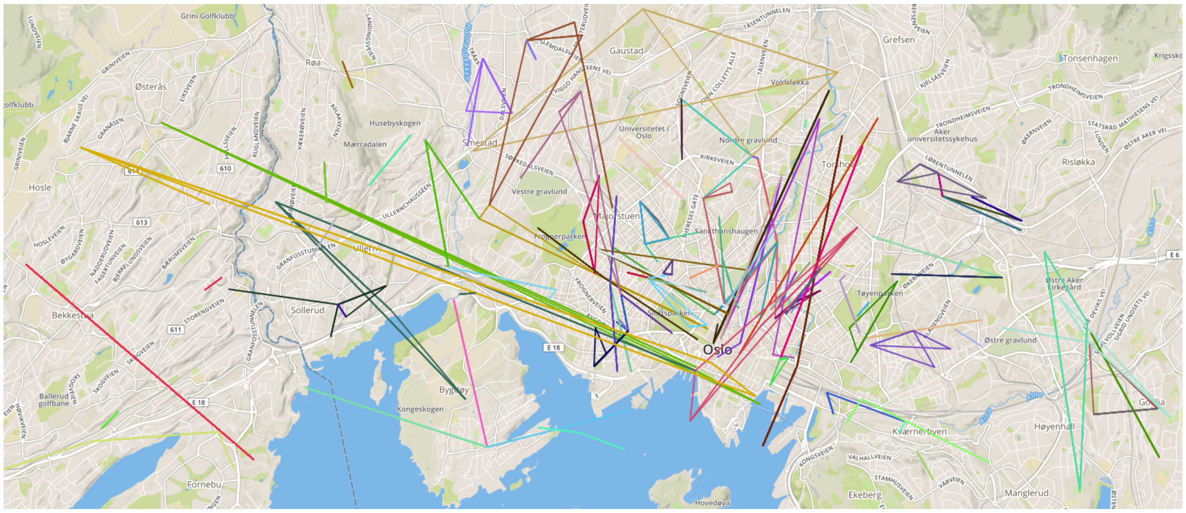
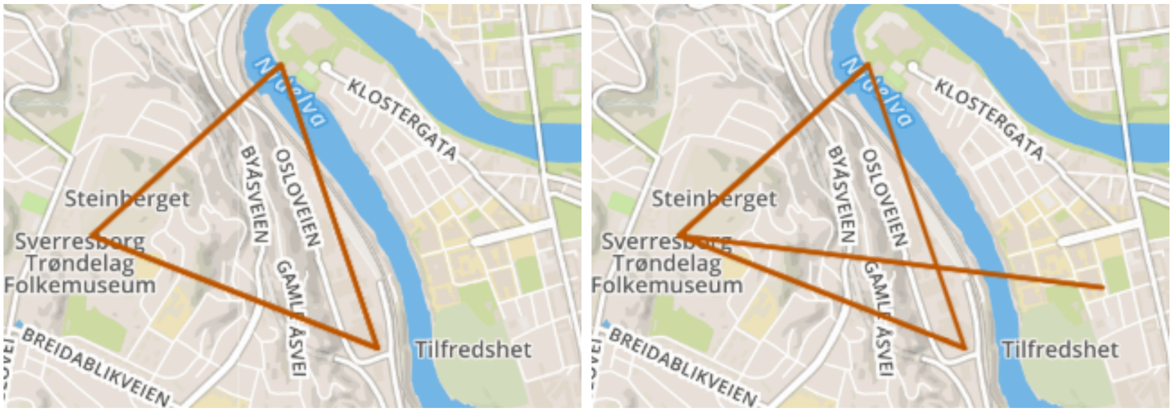


Figure 5.10: Trajectories recovered from Oslo

Figure 5.11 shows one recovered synthetic trajectory from a run in Trondheim, for two different days. The illustrations show that the three base stations create a triangle which stays the same for both days, but the second day a fourth base station has also been included in the trajectory. Most trajectories proved to be the same for all days. However, the ones that were not equal often showed a single detour like displayed in the figure.



(a) Day 1

(b) Day 2

Figure 5.11: One recovered trajectory in Trondheim

5.5 Quality of the Recovered Synthetic Trajectories

An optimal system for trajectory recovery would need the actual individual trajectories to evaluate the accuracy as they have in the framework discussed in section 3.2. In this thesis synthetic trajectories were used instead, however, how good they are in comparison to real individual trajectories is not certain. To evaluate the quality of the recovered synthetic trajectories, edit distance, and the characteristics are utilized.

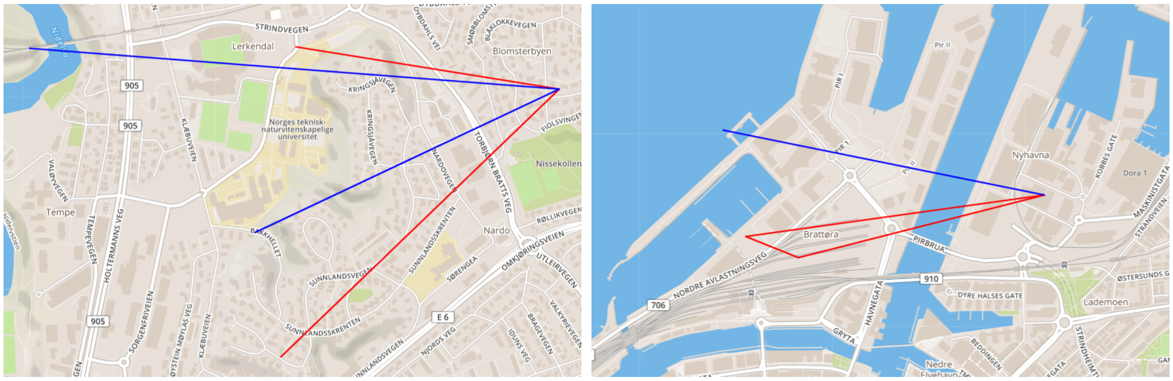


Figure 5.12: Comparison of created synthetic trajectories and recovered synthetic trajectories

Figure 5.12 displays different recovered synthetic trajectories compared with the synthetic trajectories they matched with. The red trajectories are the recovered one, while the blue illustrates the synthetic. It is clear that the recovered synthetic trajectory does not need to follow the synthetic trajectory entirely as long as the time step of the movement is the same.

From the figures 5.7, 5.8, and 5.9 in section 5.4.1, the average edit distance between the synthetic trajectories and their closest recovered synthetic trajectory is about 650 meters. However, this includes the edit distance of the trajectories which the algorithm could not find. When removing these, the average edit distance is around 500m, which means that most of the recovered synthetic trajectories are a bit off the original.

The characteristics from section 5.2, and 5.4.2 are compared and presented in table 5.4. The table shows that an average moving recovered trajectory goes through 2.33 base stations over a period of two days, with an average traveling distance of 7.95 km. When comparing this to the characteristics of the created synthetic trajectories, it is clear that these travel longer. In the third column of the table, the characteristic of the recovered synthetic trajectories is listed. Even though the number of visited base stations is about the same, these trajectories travel shorter distances compared to both the created synthetic trajectories and the average recovered trajectory. This characteristic is also reflected in the figure 5.5, which shows that the synthetic trajectories contain much movement over relatively large distances.

Characteristics	R.T.	Created S.T.	Recovered S.T
Number of recovered trajectories	87052	100	54
Average length of moving trajectories	2.33	2.20	2.27
Average distance of moving trajectories	7.95 km	10.64 km	5.95 km

Table 5.4: Comparison of features of recovered trajectories, created synthetic trajectories and recovered synthetic trajectories

Even though the comparison of the different trajectories shows differences, it does not create a basis for an evaluation of the created synthetic trajectories. It is essential that these trajectories are created in a way which simulates trajectories in the real world, rather than the trajectories recovered by the method. Even though the same attack model, presented in section 3.2, recovered 73-91% of the real world trajectories, the synthetic trajectories cannot be built upon the result of this model. The reason for this is that it would make trajectories to fit the model, which would give a better return of correctly recovered trajectory, but no verification for it to be a good measure.

Chapter 6

Discussion

The main goal of this master thesis was to provide a service to dynamically find the optimal value of k for k-anonymity on different aggregated mobility datasets from large cities in Norway. This was attempted by creating a service that takes a mobility dataset as input, then try to recover individual trajectories before hiding the data using different levels of k-anonymity until the percentage of recovered trajectories are close to zero.

The results and evaluation presented in the previous chapter suggest that the level of k-anonymity in the aggregated mobility data does not make much difference in the number of recovered trajectories. This chapter discusses the results of this project regarding the main goal and corresponding research questions. Also, it discusses what aspects of the system needs improvements, and which aspects perform well.

6.1 Characteristics of Mobility Data

Mobility data comes in many different shapes and sizes, and its characteristics are essential to how well trajectory recovery performs or if it is possible at all. This master thesis is heavily based on the methods discussed in section 3.2 where the authors evaluate their methods using mobility data with different characteristics than the mobility data used for this thesis. This section discusses how these differences and other characteristics of mobility data has affected the results of this thesis.

Table 6.1 compares three different mobility datasets. The China datasets are the datasets used in [6], and explained in section 3.2. The main difference between the three mobility datasets is the ratio between base stations and users. The China datasets are gathered from a major city in China, and even without knowing the exact city it is probable that the datasets cover an area that is much larger both in terms of geographical area and population than any of the datasets used for this thesis [1]. With this in mind, the difference in this ratio becomes even more substantial.

Mobility data characteristics	China Dataset	China Dataset	Trondheim Dataset
Source	Operator	Application	Operator
Duration	1 week	2 weeks	3 weeks
Base stations	8000	8000	589
Number of users	100,000	15,500	246,000

Table 6.1: Comparison of the datasets used in [6] and the Trondheim dataset

When the amount of base stations is high, and the number of people is relatively low the process of recovering trajectories becomes easier because people are more scattered and the changes in counts are more drastic. The algorithm used to recover trajectories, presented in section 4.3.2, is built upon the fact that the movement between two time steps must be minimized. Because, if two consecutive counts for a base station contains the same value, the algorithm would rather keep all trajectories than moving some out and receiving some new.

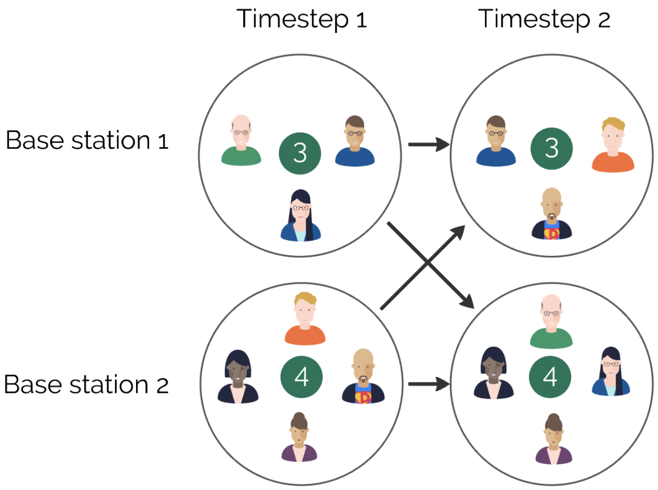


Figure 6.1: Movement between base stations

[1] We have tried to contact the authors to get more information about their mobility data without any luck

An example of a movement that most likely is undiscovered by the trajectory recovery method is visualized in figure 6.1. Here, pairs of users switch base stations while the counts for the base stations stay the same. When the number of people connected to one base station is very high, more people stay at the same base station and the chance of two and two people switching base station without the method detecting it is higher. As a result, there is less movement, and thus, less moving trajectories. As stated in section 5.4.2 as much as 45% of the trajectories recovered from the Trondheim dataset stays connected to the same base station throughout two days, which emphasized the issues with a mobility dataset with a low ratio of base stations per user.

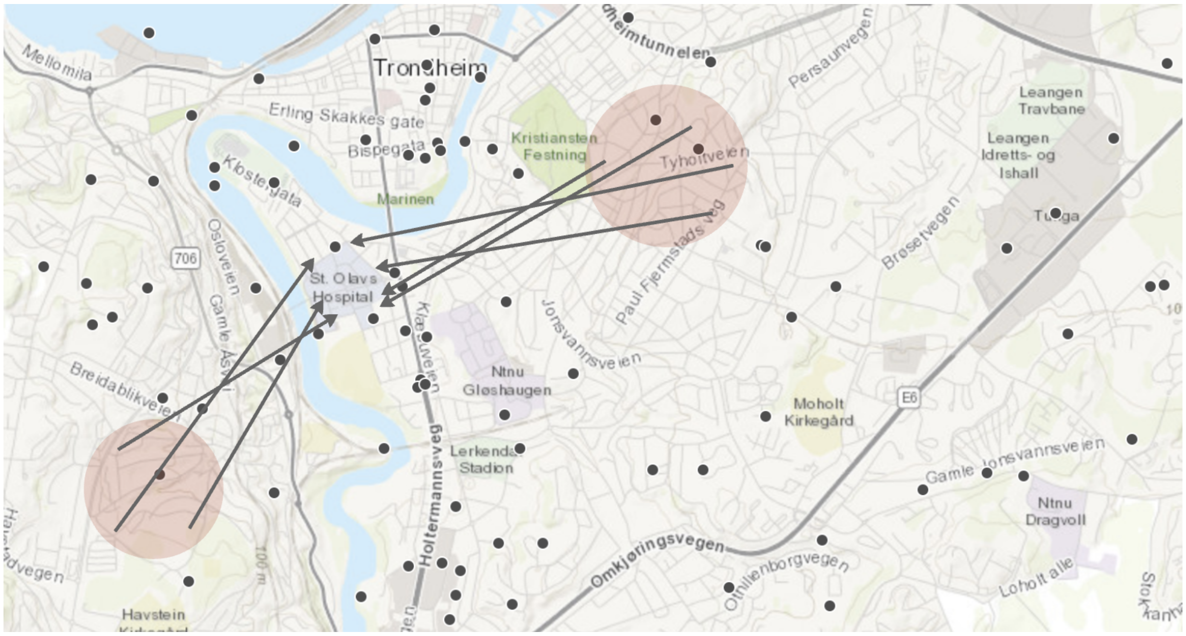


Figure 6.2: Common movement pattern

Another characteristic of the mobility datasets used in this thesis is the lack of cells. Section 4.1.1 explained that a base station is made up of cells which represents smaller geographical areas within the area covered by the base station. This characteristic also affects the trajectory recovery results. The recovered trajectories then represent paths between base stations, and without the cells present the exact positions of the users can be anywhere in the area covered by the base station. Figure 6.2 displays areas covered by base stations [2] and different fictive user's trajectories from their home to their place of work. They all live in different neighborhoods. However, all of the trajectories are about the same when recovered because the base station covers the area where they all live. With this, it can be suggested that trajectories recovered with this characteristic

[2] This is an estimated area as the actual areas are unknown

represent common movements instead of individual movements.

Another characteristic that emphasizes that theory, is the time interval for each recording in the datasets. The time interval in the Trondheim dataset is set to 60 minutes. Looking at the trajectories in figure 6.2 the paths from home to work may have included more stops if the time interval was lower. This generalizes the trajectories, and thus make the trajectories more of a common movement pattern, than individual movements.

Research question three asks how the characteristics of mobility data affect the value of k for k-anonymity. The results of this thesis suggest that k-anonymity optimization is not possible as the percentage of recovered trajectories stayed relatively consistent even when different levels of k are applied to the data. Thus, the value of k is insignificant. This section discussed why the characteristics of mobility data might be the reason for this, as they contribute to the recovery of common movements rather than individual movements. The common movement pattern present in the dataset remains intact as the level of k increases, and therefore the trajectory recovery method is still able to find trajectories. A reason for this is the low edit distance of the estimation model, which result in that the behavior in the counts are preserved. Therefore the characteristics itself aids in protecting the mobility data by producing common movements rather than individual movements and thus, makes the value of k less significant.

6.2 Estimation Model

The results presented in section 5.3 shows that when k-anonymity increases, the amount base stations without any known counts also increases. These counts would be impossible for an attack system to estimate correctly, and therefore the solution of setting these counts to $k - anonymity/2$ seems appropriate. Additionally, there are base stations with very few counts to base the estimation on. Together these problematic estimations cover a large part of the estimation which profoundly affects the accuracy of the model.

The most important aspect of the estimation model is that the estimated values follow the original path reasonably well. As the results showed, the edit distance did not increase much for each step of k-anonymity. The relatively low edit distance does not result in any significant changes in the behavior of the data, meaning, even if the estimated counts slightly varies from the original counts they still follow the same path. This is also shown in figure 5.6, where it becomes clear how the estimated values follow the same behavior as the original values even though they are not entirely correct. This means that the low accuracy does not have much negative effect on the trajectory recovery method.

6.3 Synthetic Trajectories

The synthetic trajectories are created to replicate the common movement of leaving for work in the morning and return home later in the evening. Over a few days, this could very well be the trajectories of an actual human being. The problem with these trajectories appears when the time interval covers a more extensive time perspective. A person can keep the same day-trajectory over a few days, but when the weekend or a special occasion occurs, the trajectory does change. This is an aspect which the algorithm that associates day-trajectories, described in 4.3.3 probably would recognize and handle. The synthetic trajectories, on the other hand, does not have these diversions which consequently makes them too simple for full utilization of the algorithm.

To obtain an accurate creation of synthetic trajectories, a much more detailed analysis of trajectory data is needed. This thesis does not contain the required analysis for the creation of synthetic trajectories, and the trajectories are instead a simplification of human movement. However, they are created in such a way that simulates a movement pattern which fits the majority of people. This makes the synthetic trajectories common and should be simple to discover from the aggregated data.

The synthetic trajectories are far from perfect. However, in the way they are created and added to the aggregated mobility data, it is a good measure of how suitable the method is to find individual trajectories.

6.4 Optimization of k-anonymity

Research question two ask how the value of k can be optimized to protect the privacy while preserving data utilization. This thesis has presented a framework for finding an optimized value of k, mainly focusing on privacy protection. However, with the mobility datasets utilized in this thesis, the results showed that determining the optimal value for k-anonymity was not possible. All the runs gave the same result, with the percentage of recovered trajectories staying approximately constant for all values of k. The thesis is built upon the idea that the k-anonymity is needed to protect the aggregated mobility data from attack systems to prevent them from recovering each individuals trajectory. The results leave reasons to believe that the k-anonymity has no effect at all. However, it is possible that there are other reasons to why the results keep consistent on different values of k.

As presented in section 6.1, the recovered trajectories shows signs to be more of a common movement rather than actual trajectories. When there is a decline in an area of residences and an increase in a nearby industrial area, there will be trajectories moving between these two areas. The recovered trajectories, as figure 5.10 shows, often has these kinds of simple straight back and forth structures. When k-anonymity becomes high, the method still manages to recover the trajectories with the same accuracy and approximately the same edit distance. The estimation model does not manage to estimate the unknown values, but as discussed in section 6.2 it manages to recover the same movement dynamics. This keeps the accuracy of the trajectory recovery equally good as long as the trajectories keep the common movement pattern.

6.5 Trajectory Recovery

To judge the accuracy of the trajectory recovery method with only synthetic trajectories is somewhat complicated. It is highly dependent on the synthetic trajectories to replicate actual mobility trajectories. This section discusses different aspects of the results of the trajectory recovery and answers research question 1, which asks how individual trajectories from aggregated mobility data can be recovered.

When looking at the key figures presented in 5.4.2, it is evident that a significant portion of the trajectories is motionless. However, a base station can cover a rather large area, and even though a trajectory does not move between base stations, it does not imply that they are entirely motionless as they could move between cells, which does not appear in the dataset. The results also showed that a lot of the trajectories move outside the area covered by the dataset. However, as the radius of the dataset was reduced to cover the city center where the activity is high, it seems natural that the proportion of trajectories moving outside this area is that high. Even with runs on the whole dataset, it should be quite substantial considering that there is always be movement between regions.

Figure 5.10 displayed 100 trajectories recovered from Oslo. It is not possible to verify these trajectories by looking at them, but as the figure shows the movements are highly directed towards the city center. It shows that most recovered trajectories keep very close to the home location throughout the day, while some do larger leaps into the high-density areas. Figure 5.11 is a good example of a trajectory over two days where the standard movement is the same, but one or more time steps contains a movement to a new base station. Most of the recovered trajectories are equal over two days, but quite a few have such small diversions changing up the trajectory.

Research question one asks how individual trajectories can be recovered from aggregated mobility data. The methods from [6] described in section 3.2 proves that the idea of linking trajectories based on human mobility characteristics does work if the environment is right. The displayed recovered trajectories do show sign of a realistic movement pattern. However, the key figures indicate that there is just a small portion of the recovered trajectories with this pattern. When the dataset becomes denser, more of the recovered trajectories stays still throughout the recorded interval, which indicates that large parts of the trajectory recovery do not work. However, the method does still recover quite a significant portion of the synthetic trajectories created in this thesis.

As discussed in section 6.4 the trajectory recovery method manages to recover the synthetic trajectories even though over half of the trajectories stands still or leaves the dataset. This indicates that the moving trajectories create a common movement pattern, making the different synthetic trajectories find a matching trajectory within the maximum error distance. It is arguable that the method does not find exact trajectories, but yet manages to match them up by creating standardized paths.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

To avoid breaching privacy regulations, companies need to make sure that it is impossible to recreate sensitive information, like individual trajectories when deciding to distribute their mobility data to research or as a service. To be certain of this, it is essential to test and validate different privacy techniques.

This master thesis aimed to provide a system to dynamically find the optimal value of k for k-anonymity in different aggregated mobility dataset from areas in Norway. The proposed system first attempted to recover individual trajectories using the attack system against aggregated mobility data, which was modified to fit the case. Then, different levels of k-anonymity were applied to the data. The value of k was declared optimal when the percentage of recovered trajectories was below 30% and did not produce any privacy concerns. To simulate a reasonable attack system a machine learning estimation model was incorporated to estimate the values affected by k-anonymity. The evaluation of the system was completed by incorporating synthetic trajectories to the aggregated mobility data as the original individual trajectories were not available.

The results revealed that the percentage of recovered trajectories stay consistent when different levels of k-anonymity were applied to the data, and therefore no optimal value of k was discovered. Further analysis suggested that the reason behind this discovery was that the characteristics of the aggregated mobility data were in itself enough to protect the data from privacy attacks. These characteristics included a low number of base stations per user, the lack of cells present in the data and the high time interval for each recording. Based on this discovery, the trajectories recovered using the methods in this thesis simulated a common movement pattern rather than individual movements. This discovery needs more analysis and testing to reach a reliable conclusion.

Even though more research is needed to verify the results of this thesis, our findings indicate that aggregated mobility data with specific characteristics can be distributed freely and used in research without the need to protect it beyond the aggregation. Which will, in addition to reduce the workload, decrease the potential loss of utilization that comes with protecting the data.

7.2 Future Work

This thesis has presented steps on the way to creating a method to estimate an optimal value of k for aggregated mobility data. The results are promising. However, further work is necessary to reach a reliable conclusion. This section presents three aspects to create a better environment to continue the work on the framework.

The first and possibly the most straightforward step would be to get better resources to run the algorithm, which makes it possible to run the trajectory recovery system on complete datasets. It is not believed that this would change the results drastically. However, it would give more data to analyze which may make it possible to provide a more specific conclusion about the effect of k-anonymity on aggregated mobility data of this type.

Secondly, the mobility data utilized by the framework needs to be sparser. To achieve a sparser dataset, the cell coordinates for the base stations could be included in the data, which creates more positions for the dataset to locate users, or to generate a dataset with fewer people. It seems evident that the amount of people makes it difficult to find individual trajectories and to evaluate the power of k-anonymity.

The third and possibly the most challenging aspect is to create realistic synthetic trajectories which simulate real-life movement. Here it is essential that the synthetic trajectories simulate a realistic movement without breaking privacy. It is also crucial that the created trajectories are not built upon the theories which the trajectory recovery method is built upon, but preferably a more diverse set of possible movement. Meaning, trajectories of commuters, travelers, and other types of peoples with more variation travel patterns. In addition to trajectories with a more regular movement.

These steps help the trajectory recovery method to speed up, give the method a more relevant environment and a much more precise and efficient evaluation basis to set the k-anonymity correctly.

References

[1]”EU GDPR”, EU GDPR Portal, 2017. [Online]. Available: http://www.eugdpr.org/. [Accessed: 20- Nov- 2017].

[2]J. Han, M. Kamber and J. Pei, Data mining, 3rd ed. Amsterdam: Elsevier/Morgan Kaufmann, 2012, pp. 8-91.

[3]”SPSS Tutorials: Pearson Correlation”, Libguides.library.kent.edu, 2017. [Online]. Available: https://libguides.library.kent.edu/SPSS/PearsonCorr. [Accessed: 20- Nov- 2017].

[4]A. Pyrgelis, C. Troncoso and E. De Cristofaro, ”What Does The Crowd Say About You? Evaluating Aggregation-based Location Privacy”, Proceedings on Privacy Enhancing Technologies, vol. 2017, no. 4, 2017.

[5]R. Shokri, C. Troncoso, C. Diaz, J. Freudiger, and J.-P. Hubaux, “Unraveling an old cloak: k-anonymity for location privacy,” in Proceedings of the 9th annual ACM workshop on Privacy in the electronic society, ser. WPES ’10. New York, NY, USA: ACM, 2010

[6]F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, “Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data,” in Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017, pp. 1241–1250.

[8]F. Giannotti and D. Pedreschi, Mobility, data mining, and privacy. Berlin: Springer, 2008, pp. 1-147.

[9]”Spearman’s Rank-Order Correlation - A guide to when to use it, what it does and what the assumptions are.”, Statistics.laerd.com, 2017. [Online]. Available: http://bit.ly/1NIncKc. [Accessed: 22- Nov- 2017].

[10]"Mobile Phone Base Stations", Ehtrust.org, 2005. [Online]. Available: https://ehtrust.org/wp-content/uploads/MMF_Mobile_Phone_Base_Stations.pdf. [Accessed: 28- Nov- 2017].

[11]Y. Zheng, "Trajectory Data Mining", ACM Transactions on Intelligent Systems and Technology, vol. 6, no. 3, pp. 1-41, 2015. [Accessed: 28- Nov- 2017]

[12]D. Bruff, "The Assignment Problem and the Hungarian Method", Harvard University, 2005. Available: http://www.math.harvard.edu/archive/20_spring_05/handouts/assignment_overheads.pdf. [Accessed: 28- Nov- 2017]

[13] Datatilsynet, "veileder: ANONYMISERING AV PERSONOPPLYSNINGER", 2015. Available: https://bit.ly/2sM5Sl5. [Accessed 4 May 2018].