GPS trajectory Anonymization Comparing methods of Privacy Preserving Data Publication

Yorick Kooij (ICA-3919609)

Department of Information and Computing Sciences
Utrecht University

March 16, 2021

Contents

1 Introduction

The last few decades there has been a rapid increase in the collection of trajectory data belonging to objects such as cars and people, using applications that make use of various technologies like GPS, wifi, or mobile signal. These massive amounts of data can be a treasure of information that can be useful for a wide variety of purposes, including geographical planning, crowd monitoring and risk assessment. However, this data also raises some serious privacy concerns, especially when the data is to be released (publicly) for research purposes.

In this domain a trajectory usually consists of a list of coordinates ordered by their respective timestamps. Even when this kind of data is not directly accompanied by sensitive information, re-identification can still take place. This can happen through the correlation of the signal with a space restricted to a single source, or through the correlation of the signal with an external observation of a single source.

Even though location data may be sensitive, there can be good reasons to want to publish the collected data. To ensure the privacy of the owners of the collected trajectories, some kind of transformation has to be applied to the database before it can be published that can both ensure the usability of the published data while at the same time providing a guarantee of privacy for the users involved. This process is called Privacy-Preserving Data Publication (PPDP). There are various approaches to this, which will be further discussed in the related literature section.

1.1 Motivation

Currently a lot of the research done comparing the PPDP algorithms only covers methods within their own respective family, for instance, multiple variations of a k-anonymization algorithm, and these comparisons are rarely made within the context of geospatial data. Additionally, a lot of the techniques are designed and developed with relational databases containing strictly private informational fields in mind rather than trajectory data containing quasi-identifiers. Quasi-identifyers are fields containing data that does not disclose a user's identity by itself, but when combined with additional information can lead to identification.

There are a few reasons why the amount of comparative research done towards the anonymization of geospatial data is so limited. Firstly, since location data is privacy sensitive information, access to most datasets that consist of natural data is limited and they are generally managed by private corporations. Because of this most publicly available datasets are synthetic, which means a comparison made using this data does not necessarily reflect real-life performance.

Secondly, researchers often publish their research without fully disclosing an algorithm they developed, which makes it difficult to reproduce their research or compare their algorithm's performance to others.

Thirdly and finally, there is no clear-cut consensus as to how these comparisons should be made. There is a lot of variation in the use of data mining operations, quality measures and error metrics, which may only be useful for a specific family of algorithms.

These issues result in a lot of research not paying attention to the other branches of location-data anonymization, which makes it difficult to navigate the vast amount of literature that deals with the problem at hand, and leaves us only to speculate as to what technique is best fit for privacy preserving data publication. Since this anonymized data could serve a plethora of different purposes, each of which could potentially benefit from a different anonymization strategy, the current literature steers clear of making any statements regarding practical application. All in all there is a big need for an objective, straight-forward approach to analyzing and comparing algorithms used for PPDP.

This document aims to construct the start of a framework for comparing different anonymizations of the same data, using metric candidates that may be able to directly or indirectly disclose information about the utility loss that is being introduced.

1.2 Problem Statement

The definition of a **trajectory** that is commonplace in the literature and will be used in this report as well is as a set of 3-tuples containing a timestamp t, longitudinal coordinate x, and latitudinal coordinate y. This set is ordered by the timestamps:

Definition 1 (Trajectory) $(t_1, x_1, y_1), ..., (t_n, x_n, y_n)$ *satisfying* $t_1 < t_2 < .. < t_n$

Possible additional measurements that can be stored in these data points include speed and heading, although these can also be inferred from the data model as defined above.

This kind of collected trajectory data is generally accompanied by an identifier of the object that is being tracked, which can range from a mac-address or an ip-address which are both obvious cases of privacy-sensitive information by themselves, but is more likely to be a hashed value generated from these personal identifiers called a **pseudonym**. Since these devices are generally only used by one person, called a **user** from now on, a mac-address or ip-address can be regarded as personal information. However, even without these identifiers GPS-trajectories can be susceptible to re-identification by the following means:

- **Restricted space correlation**: When a certain location is only accessible to one known person a single data point of a trajectory can disclose the user identity that it belongs to.

- **External observation correlation**: If the position and identity of an individual is known through other means than the trajectory data at hand (e.g., video, transactional records, license plate registration, etc.) and there is no other possible origin of the trajectory the corresponding identity can be disclosed.

When the identity of a user is inferred by means of an attack, even when the user identifier is only stored as a hashed attribute an attacker can link the user identity to every trace that shares this id. The attacker does not need to have the hashing function at his/her disposal to do so.

It is very difficult to determine the vulnerability of a single trace, since a lot of factors are in play. For instance, the accuracy of the GPS measurements needs to be sufficient before it can be linked to an external observation or restricted space. The problem here is that it is impossible to pinpoint what would be enough accuracy for re-identification. When it involves a single measurement with a precision radius of 250 metres, it might be near impossible to link this coordinate to a user, but when a commuting pattern is suspected from one or multiple traces that suggest travel between a workplace and a residential area, one could theoretically cross-reference the payroll of the company/companies situated at the workplace with the residents of the neighbourhood the user is assumed to live in. How this data is acquired specifically is not important for this example, which shows how trajectory data does not need to be very accurate to still be considered privacy-sensitive information.

However, this research does not concern whether or not the exact privacy guarantees provided by an algorithm are sufficient. Anonymization techniques possess two important properties a comparison can be drawn from. Next to the extent of privacy that is offered there is also the extent to which the utility of the original data is preserved. The literature generally accepts that there exists a trade-off in between privacy and utility. An empty dataset provides minimal utility and maximal privacy, whereas the original dataset provides maximal utility and minimal privacy. Anonymization techniques tend to either maximize the utility under the condition of a specified level of privacy, or alternatively, maximize the privacy under the condition of a certain standard of utility. An important assumption about utility that is not always clarified in the literature is that the utility of the original dataset is 100%, whereas the utility of an anonimized dataset is always less than 100% provided the anonymization process introduced any perturbation of the original data. The focus of my research will be on the degree of utility retained by PPDP algorithms that maximize utility while upholding a minimum required level of privacy.

1.3 Research Questions

The absence of a straight-forward non-empirical way to compare the utility of different approaches to location privacy raises a lot of questions. The main concern is as follows:

Research Question 1 (Main Research Question) *How can the utility of an anonymized publication D' of trajectory database D be measured independently of the method of anonymization?*

A general assumption in the literature is that the utility of original data is 100%. The utility of a privatised version of this dataset is then determined by examining how much of the information present in the original dataset is retained. However, there might be situations imaginable in which a large amount of synthetic data generated from a small original dataset could provide more use than the original could. There could also be situations in which some property of a PPDP algorithm might be beneficial to the utility of the data instead of detrimental. An example could be when an algorithm uses some form of noise, which could theoretically help to prevent overfitting to a dataset.

Additionally, there are a lot of different measures that can be taken from a trajectory dataset that may be used to construct a utility metric, like the average amount of displacement of individual trajectories, the average length or timespan of a trajectory when compared to the original data, etc. Each of these might tell us something about the utility of a privatised dataset, but this could also be highly dependent on the situation at hand.

These matters raise the following questions:

Research Question 2 *What are possible utility measures of a privacy preserving trajectory data publication that compare it to the original data?*

Research Question 3 *Can a comparison in between two different privacy preserving trajectory data publication algorithms be made using one or more utility measure candidates?*

To put these measures into practice and to make an attempt towards answering the aforementioned research questions a comparison should be drawn in between two different PPDP algorithms. For this an optimization for a differentially private PPDP algorithm was chosen that implements a budget manager, which will be compared with a basic version of the same algorithm. Since the budget manager is an optimization, the H0 hypothesis for this comparison is that it does indeed provide a significant improvement.

Research Question 4 *Does a budget manager improve the resulting data utility while providing differential privacy during privacy preserving data publication?*

2 Related Work

2.1 Utilizing Trajectory Data

Before going into detail about publishing location data in a privacy-preserving manner, it is important to identify what kinds of tasks are to be executed using this data. An overview paper by Yu Zheng [30] identifies three important steps in trajectory data mining:

The first step towards trajectory data mining usually consists of a preprocessing stage. In this stage, a variety of techniques can be applied.

Amongst those are noise filtering, trajectory compression and trajectory segmentation. Operations like stay point detection and map matching are more concerned with the semantic nature of the visited area. Stay point detection detects whether an object stays in or around a certain location for a prolonged period of time, where this location can be a shopping area or another location of interest. Map matching tries to project points of a trajectory onto a map, which is usually a road map of some kind. Additionally, the pre-processing stage can involve trajectory indexing, which can provide more efficient retrieval. This can be especially useful in case of an application that queries the dataset continuously.

After the pre-processing stage, there are various possible mining tasks. The paper by Zheng identifies four categories of trajectory mining tasks:

- **Trajectory Uncertainty:** When a trajectory contains location data points at a specific interval, the exact location in between these measurements is uncertain. To enhance the utility of trajectory data, this uncertainty can be modeled and reduced.

- **Trajectory Pattern Mining:** Large quantities of trajectory data allow for the detection of patterns. These patterns can be either represented by a single trajectory, or a group of trajectories displaying similar behaviour. In the literature, trajectory patterns generally fall into one of the following four categories: moving together patterns, trajectory clustering, periodic patterns and frequent sequential patterns.

- **Trajectory Classification:** By means of supervised learning, trajectories or parts of trajectories can be automatically categorized. Examples of categorical distinctions that can be learned are certain activities, like hiking or shopping, or means of transport, like taking the train, cycling, etc.

- **Trajectory Outlier Detection:** The last category of tasks concerns itself with the detection of outliers within a dataset. This can be done using some kind of similarity measure. Alternatively, when a trajectory does not follow an expected pattern it can be marked as an outlier. An example of this would be a car that enters a traffic jam, after which its speed is drastically lowered in comparison to the expected speed.

The last stage of data processing identified by Zheng is that of data transformation. This means the trajectory data is represented in a different format like a graph, matrix or tensor. These new representations open up the possibility of new mining operations that take these representations as an input. The anonymization of a trajectory base can also be seen as a data transformation.

2.2 Approaches to Privacy Preserving Data Publication of Location Data

There are various approaches to protecting the privacy of users during the privacy preserving data publication (PPDP) of location data. While the two most important strains of algorithm either base themselves on the notion of k-anonymity or ϵ-differential privacy, there exist a multitude of approaches to PPDP that are not based on a strict privacy principle.

2.2.1 k-anonymity

One of the major approaches to PPDP proposes a notion of privacy that is based on the principle of indistinguishability. It is called k-anonymity and was first described by Samarati et al. in 1998 [28]. The property guarantees that in the release of a dataset, the information of every single person cannot be distinguished from the information of at least $k-1$ other individuals. In a database containing GPS-traces this means that the set of (sub)trajectories belonging to a certain user that end up in the aggregated database must be identical to the sets of $k-1$ other users. This means that a higher value of k indicates greater anonymity, since an increase in the number of indistinguishable trajectories decreases the chance of a successful re-identification to $1/k$ or smaller.

A k-anonymous version of a database can be constructed by either generalizing or repressing data points. Both of these techniques were first introduced together with the notion of k-anonymity by Samarati et al. It is also possible to use some combination of both techniques, until the database satisfies the required k-anonymity for a given value k.

In order to tackle certain vulnerabilities there have been several elaborations of the k-anonymity property that have been proposed together with algorithms that construct such an anonymization. A property that guards against homogeneity attacks is called l-diversity and was first proposed by Machanavajjhala et al. [3] Another property that is not applicable to location data is t-closeness, a property introduced by Li et al. that prevents the induction of sensitive information from the value distributions of sensitive fields.

Additionally, weaker notions of k-anonymity have been proposed as well, for example, LKC-privacy was proposed by Mohammed et al. [25]. This property guarantees that as long as an attacker does not know more than L visited locations, which are all shared separately with at least K users, the probability of a successful attack is limited by C.

2.2.2 Differential Privacy

Another approach to privacy preserving data publication of location data is through uninformativeness. This is based on the idea that in order for a database to disclose sensitive information, this sensitive information has to be present in the database. If information is not present, it can not be disclosed, which is why an empty dataset provides the best privacy guarantees. A well-defined privacy measure based on the principle of uninformativeness is that of ϵ-differential privacy [11]. The idea of differential privacy is that during a query any single user's data has a negligible effect on the data that is released, with the effect of a single entry being bounded by the privacy parameter ϵ. A privacy mechanism is ϵ-indistinguishable if for all databases x and x' that differ by a single row, the probability of obtaining any transcript t —this includes the differing row as well— when database x is queried and the probability of obtaining this transcript when database x' is queried are within a $(1 + \epsilon)$ multiplicative factor. This can be achieved by incorporating Laplacian noise into a data mining operation [11] or by returning a randomized probability distribution [24].

Because ϵ-differential privacy aims to privatise a mining operation instead of a dataset, it cannot be directly applied to the release of location data. One algorithm that does allow for a form of differentially private publication of location data is that of $(\epsilon$-$\delta)$-differential privacy, first proposed by Shao et al. [13]. By allowing a small probability of δ that ϵ-differential privacy will not be guaranteed, it achieves $(\epsilon$-$\delta)$-differential privacy for the rest of the dataset.

2.2.3 Variational Autoencoder

A popular unsupervised deep machine-learing technique that can discover complex distributions in data is the Variational Autoencoder (VAE). This is a variation of an autoencoder. An autoencoder consists of two networks, an encoder and a decoder. The encoder is a neural network that aims to convert data to a more minimal representation, generally the last layer of a convolutional network. This representation has a dimensionality that is much lower than the original input, which is often an image. This compact representation can then be passed to a different neural network called a decoder to reconstruct the original data. The more this resulting image looks like the original image, the better the quality of the encoding mechanism. The information loss can be measured using the log-likelihood function. Using this, the network can be trained using a stochastic gradient descent that minimizes this error metric, after which the final, trained network, can be used to convert data of the same type as has been trained on into a minimal representation, and can also be used to remove noise.

Now one would think that this architecture can also be used to generate new, similar data. However, when you create an encoding by sampling from a normal distribution and then feeding this generated representation into the decoder, the result would not make sense, and would not look like the training data at all. A VAE is an adaptation of the regular autoencoder modified specifically to make generation of new data in this manner possible. This is achieved by making the assumption that the latent variables that make up the representation follow some sort of distribution, for instance a Gaussian one. A network trained under these assumptions will be able to come up with new, similar data by sampling from the presumed variable distribution. Because most properties that can be encountered in the real world follow some kind of distribution, this approach can be effective for a wide range of domains. Since a VAE suffers from information loss during encoding, images generated using a VAE tend to be blurry when compared to the training data, way more so than is the case with artificial data generated by, for example, a generative adversarial network. Since this loss of information happens when using images one can assume that this is also the case for other types of data. This raises the question as to how useful this type of synthetic data is in practice, when compared to other methods of artificial data generation.

Since there is a chance the distribution learned by the VAE discloses sensitive data, the synthetic data it generates can not fully guarantee the privacy of individuals present in the training set. To guarantee their privacy, the principles of differential privacy can be ingrained in the training phase, both for a VAE and the regular autoencoder. This was proposed by Chen et al. [8]. They suggest that the protective qualities of incorporating differential privacy into a generative model emerge from the perturbation of the training data, and not necessarily from the principles of differential privacy itself.

A sequential VAE has been proposed by Huang et al. [19] that can generate human trajectories in an urban setting. Another more loosely related example is that of city plan generation [20], where images of urban layouts are generated using a VAE.

2.2.4 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of artificial neural network that accepts a temporal sequence of variable length as input data. Since trajectories consist of data points ordered in a temporal manner, they fit the input model of a RNN rather well. In theory, a RNN is able to detect dependencies that can lie far apart in the input sequence. However, in practice, the influence of input that lies far apart quickly converges to either zero or infinity -called vanishing or exploding gradients respectively-, since RNNs make use of finite-precision calculations [17]. To overcome this problem a specific type of RNN was proposed called long short-term memory (LSTM) [18]. With the addition of a part called a constant error carousel, information is allowed to pass the network unchanged, thereby circumventing the repeated multiplication that causes gradient vanishing and exploding. More recently a simplified version of the LSTM has been proposed called a gated recurrent unit (GRU) that achieves similar performance while using less parameters [9].

Since RNNs are highly suitable for domains containing subsequent datapoints over time, there have been several implementations that make use of this technique that can learn the rules of a specific trajectory-data domain unsupervised and can then subsequently be used to generate synthetic data by sampling from the network. One example of this would be the generation of vehicle to vehicle encounters [12].

Unfortunately the literature currently does not provide a clear answer concerning the privacy risks associated with this family of generative models in a geospatial context, but one can assume that these risks are very similar if not the same as those seen in the other deep learning based generative techniques. It is likely that these risks can be mitigated by incorporating differential privacy into the training phase, and such a technique has been proposed for the domain of recurrent natural language models [23].

2.2.5 Generative Adversarial Networks

One of the newer techniques in machine learning that has been developed as recently as 2014 is that of the generative adversarial network (GAN) [16]. The goal is to build a network that generates data that is indistinguishable from natural data, without depending on natural data. These structures actually consist of two neural networks that engage in a perpetual game, if you will. One of the networks, called the generative network, generates new data from random noise, and the other network, called the discriminative network, determines if data it sees is natural or not, based on statistical qualities and patterns learned from a known training set. The generator is usually a deconvolutional network while the discriminator is a convolutional one.

Since backpropagation is applied to both networks, over time the discriminator will get better at differentiating between synthetic and natural data, and the generator will get better at producing synthetic data that is not deemed synthetic by the discriminator. One example of the possibilities of these types of networks is the generation of pictures of human faces, that are able to fool humans even though the depicted person does not actually exist, but these techniques can also be applied to location data [29].

Although no direct link exists in between the data fabricated by the generative network from noise and the data that was used to train the discriminative network, there is always the chance that some sensitive information is leaked, or that the presence of a person in the training set can be deducted when an attacker possesses a sufficient amount of background knowledge, since the underlying distribution of the data is learned from training data that belongs to real people. To ensure this is no longer possible, the training phase of a GAN can be done in a differentially private manner [22].

So far only a single attempt at implementation in the domain of geospatial data can be found in the literature [21], but with promising results. Analysis has shown that not only the statistical properties of the original data are preserved, but also its intrinsic semantics (which means the way certain locations are visited can be learned without requiring the manual labeling of these locations).

2.2.6 Other Approaches

Various approaches exist that are relevant to the field of PPDP even though they are not built upon a well defined privacy principle like k-anonymity and differential privacy, like swapping locations [27] or creating so called mix zones [5]. Since these methods do not provide any privacy guarantees they are currently not fit for the privacy preserving publication of datasets.

2.3 Comparative Research

While some comparative research to the privacy preserving publication of location data has been done, these studies tend to be limited to the comparison of algorithms within a single algorithmical family. A good example of such a paper is a comparison of different algorithms providing k-anonymity done by Ayala-Rivera et al. [4], which proposes three metrics. While they claim that the metrics they chose are genera$l$-purpose utility measures applicable to most anonymization algorithms, two of them appear to draw heavily on the inner workings of the k-anonimitization family of algorithms. It should be noted that this paper does not cover location data specifically. Their proposed metrics are as follows:

- **Generalized Information Loss:** This metric quantifies the degree of generalization. In the generalization step, data fields are replaced by a range in which the real value lies. The informativeness of this range can vary greatly. The Generalized Information Loss metric measures how descriptive the generalized range is, with a value of 0 for the original data, and 1 for maximal generalization. For example, when age is a field, a bracket from ages 0 to 100 is less informative (and might be assigned value 1) than a bracket from age 10 to 20 (which might be assigned value 0.1). While not directly applicable to trajectory data (since these anonymizations generally do not make use of brackets to introduce uncertainty), the amount of information loss introduced in the generalization step of a location based k-anonymity algorithm can be measured for example by taking the location displacement in some form or another. Since most anonymization techniques introduce some form of information loss, this is the best candidate for a utility metric applicable to a wider spectrum of algorithms given by this paper.

- **Discernability Metric:** This metric introduces a penalty equal to the number of identical records. If a record is suppressed, this penalty is set to be the size of the table, which is the theoretical maximum of identical records present in a dataset. This is done to discourage suppression in favor of generalization, even though no argument is made as to why this implies a higher utility of the data.

- **Average Equivalence Class Size Metric:** This metric measures how well the equivalence classes created by the algorithm approach the ideal size of k. An ideal anonymization of a database that contains x records would have x/k classes of size k. The existence of smaller classes is impossible since this would violate the principle of k-anonymity. The metric is specified to be the number of records divided by k times the number of equivalence classes. This means the optimal value for this metric is one, while having larger equivalence classes than what is considered to be ideal results in a lower value. Both this metric and the Discernability Metric are explicitly based on the notion of k-anonymity and are not applicable or useful for other families of anonymization algorithms.

Another paper by Chatzikokolakis et al. [6] starts off with a broad overview of the field and concludes with a privacy and a utility metric for anonymized datasets which are heavily based on the principles of differential privacy. Given the real location of a user, the probability distribution of the possible reported values given by an anonymization technique K, and an undefined quality metric d that measures how much the quality decreases when location z is reported while the actual location of the user is x. This framework is not only applicable to probabilistic anonymization techniques since a deterministic technique can be seen as having a probability distribution with one value having probability 1.

Taking a more practical approach to privacy and utility metrics, Cormode et al. [10] propose an empirical model that allows for comparisons between different privacy models instead. They argue that most metrics used in the literature are based on some form of information loss, while it is not clear how these metrics relate to the actual use of the data. In their model a set of representative queries is executed on both the original dataset and the anonymized version, after which the accuracy of the results of the queries executed on the anonymized dataset can be measured. This approach enables drawing a comparison between any family of anonymization algorithm and allows for an investigative plot of the privacy-utility trade off. An interesting finding is that using their measures, while the privacy-utility trade off is almost the same for k-anonymity and differential privacy, it appears that especially when working with smaller datasets, algorithms based on k-anonymity retain more utility, while also presenting a bigger privacy risk. Their conclusion confirms the suggestion found across the literature, that for releasing a dataset to a third party, the preferred method of choice is differential privacy.

Another paper that proposes a utility metric is that of ElSalamouny et al. [15]. In addition to their introduction of an adaptation to ϵ-differential privacy, (D,ϵ)-location privacy, in which parameter D determines the distance at which the sources of two points should not be distinguishable, they define a general notion of privacy called l-privacy that aims to generalize any model that reaches indistinguishability through a differential process. While this model appears to be useful when trying to compare various algorithms based on differential privacy, it does not appear to allow for the generalization of models based on a different notion of privacy.

Since no clear-cut method for comparing the utility of the different approaches to PPDP appears to exist within the context of location data, this document proposes a study to further investigate the means of comparison that do exist and attempt to formulate and apply a metric that could serve as a general-purpose utility metric for PPDP of location data.

3 Methodology

In order to compare the utility provided by two different PPDP algorithms, several components are needed, which will be discussed in their separate subsections. Two datasets will be used, one of them being natural, the other synthetic. To draw the comparison several candidate metrics have been devised, each of which may provide some insight into the way a PPDP algorithm perturbates the original dataset. After describing the metric candidates two PPDP algorithms are explained, after which the ranges of privacy parameters that are to be used in these experiments are determined. Finally, the last subsection describes how all these parts are combined and how the experimental results will be analyzed.

3.1 Datasets

To draw a comparison the literature makes use of both natural and synthetic datasets, often making use of both. Therefore this study will do so as well. However, it has proven to be very difficult to find an appropriate synthetic dataset, in spite of the fact that these datasets are not privacy-sensitive. It appears to be easier to find natural datasets publicly available, mainly from taxi drivers in an area that have been followed over an extended period of time. I will be using such a dataset recorded in Porto, Portugal. For the synthetic dataset necessary for the comparison I designed a primitive algorithm that generates a set of simple trajectories.

Research from MIT [26] argues that experiments done with synthetic data do not necessarily yield significantly different results compared to the usage of natural data. To examine this, data analysts were provided with either a natural dataset or a synthetic one, after which no significant differences in the performance of developed metrics were observed overall. Nonetheless, it will be interesting to see if the results obtained from the two datasets are indeed very similar. Since the synthetic data generated does not attempt to emulate vehicular traffic in an urban area, and the generated trajectories are relatively short when compared to the natural dataset, the results obtained could very well be significantly different, especially in the case of algorithms that do not anonymize trajectories indepentently of one another, like the k-anonymity family of algorithms.

3.1.1 Natural Dataset

The Porto Taxi Dataset [1] is a publicly available dataset containing recorded data of real taxis. All 442 taxis that operate in the city of Porto were followed for a year, from July 1st, 2013 to June 30th, 2014, by registering their GPS location every 15 seconds of a trip. The entire dataset constitutes around 1.7 million trips, which is a lot of data to process. To keep the amount of trips manageable I chose a single day from the dataset randomly. This turned out to be April 24th, a Thursday outside of the holiday season. Because the global coordinate system does not provide an ideal scale both for the precision of calculations, nor does it provide results that are easy to understand for a human reader, the trajectories were converted from a longitude/latitude system to a system that measures the longitudinal and latitudinal distance from an origin in metres. The best candidate for the

	Sum	Min	Max	Avg	St Dev
Trajectory Length (m)	26758018	1	139570.6	5154.7	5313
Trajectory Speed (m/s)	35221.3	0.0668	66.96	6.785	3.925
Trajectory Duration (s)	4022190	15	20340	774.84	671.75

Table 1: Statistics of the Porto Dataset

origin location is the central station of Porto, since it is the most common start and end point for the taxi trips. In addition to cleaning the dataset and putting it into a format that is easy to read into an environment based on R or C, incomplete entries were removed. In the entire Porto dataset there were 10 entries marked missing beforehand, and 5901 entries with a single GPS record, which I regarded as incomplete since a single GPS-location does not represent a proper trajectory.

On April 24th this leaves a dataset with 5919 unique trajectories that are to be anonymized. Some measurements regarding the average speed and the length of the trajectories can be found in Table 1.

3.1.2 Synthetic Dataset

Because a suitable synthetic dataset has proven difficult to find, I wrote a method that generated a very simple one. It is called the Tony-Hawk 900 dataset and can be found in the appendix.

The algorithm imagines 1000 clones of legendary skateboarder Tony Hawk, standing on the middle of a parking lot with a radius of 900 metres. At timestep zero they all depart at a constant speed of 5 m/s in a random direction. When they leave the parking lot after 900/5 = 180 seconds, the recording stops. To keep comparison to the Porto Dataset as simple as possible, the Tony's are registered every 15 seconds as well.

Since the trajectories generated by this algorithm all move in a straight line at a constant speed, they are not a realistic representations of human urban traffic. But one could argue that the behaviour of taxi traffic and pedestrian traffic is very different from one another as well. A comparison with the most minimalistic form of synthetic trajectory data imaginable could be advantageous, because it eliminates the influence of complex human behavioural patterns from the results, which might lead to additional insights regarding the influence of the algorithm itself. When findings between the natural and the synthetic dataset are very similar, they are more likely to be the consequence of the algorithm instead of a mere expression of the way it deals with the underlying human behaviour present in a specific dataset.

3.2 Utility Metric Candidates

Although it does not directly supply the reader with a specified quality metric, the method given by Chatzikokolakis et al. [6] as described in Section 2.3 lends itself well to the comparison of two different families of algorithms.

Important to note is that the metric candidates described in this section do not always result in a percentage that allows for easy comparison. In cases where a metric results in a real value this value might only be significant within the context of a dataset. Alternatively, its value could entirely depend on the privacy parameters chosen for the applied algorithm.

While the results of the metric candidates might not always directly relate to utility, they could provide some insight into the extent to which the data is manipulated during a PPDP process. Since any manipulation of the dataset introduces information loss, these metrics could be indirectly related to the introduced loss of utility.

A good candidate should measure a specific change introduced to the trajectories, either by comparing trajectories pairwise to their anonymized counterparts—something that is not possible for every PPDP algorithm—or alternatively, comparing a statistic taken over the entire dataset.

The measurements applied by a metric candidate should be simple, both in terms of calculations and in terms of understanding. In terms of calculations this is important because this type of data is often collected at a massive scale, and expensive calculations may result in the analysis of a bigger anonymization taking up hours or even days. In terms of the understandability the candidates should be easy to grasp, because the raw values might not mean a lot by themselves, and they should be examined with understanding of the underlying principles of the algorithm.

When a metric is calculated for the dataset by comparing trajectories with their anonymous versions in a pairwise manner, the average of these measurements is recorded in addition to their sum, minimum, maximum and standard deviation.

Over the course of this section various metric candidates will be proposed and their expected merits will be briefly discussed.

3.2.1 Average Change of Trajectory Length and Duration

The first family of metrics I chose when trying to measure the information loss introduced by a PPDP algorithm is the change of trajectories' length and duration. These measurements can be taken as a percentage measuring the relative reduction in length or duration introduced for each trajectory. In cases where an algorithm increases the length and/or duration of trajectories, a negative change value will be the result. If some trajectories are lengthened and others are shortened these measurements will cancel each other out. A simple solution for such a case that penalizes the shortening of a trajectory equally to the lengthening of a trajectory is simply to take the absolute value of a measure. Additionally, in order to penalize bigger changes of the original data more over smaller ones these relative measurements can be squared.

Alternatively, the change of the trajectory length can be measured in metres, and the change of the duration in seconds. While these measurements might tell an inquisitor less about the amount of information loss introduced -since they do not represent the amount of a trajectory that is left after an anonymization- they could provide some additional insight.

When we imagine a PPDP algorithm that prunes the start and/or end of the original trajectories, a percentage that represents the reduction of a trajectory, be it in length or duration, can be a direct indicator of the information loss introduced. However, when an algorithm introduces a lot of noise to the individual trajectory points, the length of the trajectory can increase drastically. In these cases, the metric measurements will represent a growth in length that is a direct indicator of the amount of noise introduced, instead of the amount of information of the original trajectories that is retained. This means the length measurements become an indirect indicator of the utility that might be preserved under this noise.

In conclusion, the change-related metric candidates used during further investigation are as follows:

- Change of length (%)
- Change reduction of length (m)
- Change of duration (%)
- Absolute change of duration (s)

In addition to the pairwise comparison metrics, measurements taken over the entire dataset could be useful. The average trajectory length and duration should be compared between the original and anonymized datasets, since they are a possible utility metric as well. This gives us two additional metric candidates that can be represented by either a metric or a relative value:

- Change in average trajectory length (m)
- Change in average trajectory duration (s)

3.2.2 Average Trajectory Displacement

From a user-perspective, sending a signal to a location-based service provider is more useful as it is closer to the true location. One can imagine that the utility for a researcher develops in a similar manner, where an anonymized trajectory is more useful as it stays closer to its original counterpart.

There are various approaches to the construction of a metric based on the displacement of individual anonymized trajectories. In this context the displacement of a single trajectory point would always be the distance of the anonymized location to its original location. One option is to measure the displacement of every point in a trajectory, after which the minimum, maximum or average displacement can be taken as a measure. An important prerequisite here is that individual points of an anonymized trajectory can still be matched to their source, something which is not always the case.

When we imagine a situation where these metrics are applicable, the average maximum displacement would indicate the expected maximum error for the user, were they to request a service providing the anonymized location when in reality they are at the original location. Simultaneously this maximum error can also indicate the amount of data perturbation to a researcher.

A final option that does not require the matching of individual trajectory points is to take the Hausdorff distance between the original and the anonymized trajectories as a metric candidate. Similarly to the average maximum displacement of the individual points, this measure indicates the maximal error that can be expected to be introduced. Since a point can be moved a long distance while staying close to the entire trajectory, this measurement is a bit more forgiving and is expected to be lower than the maximum displacement, although the averaged displacement found within a directory is likely to be more forgiving in nature.

An alternative way of using the Hausdorff distance as a metric can be especially useful in cases where trajectories in an original dataset do not correspond pairwise to trajectories in the anonymized dataset. This is the case in some forms of differential privacy where data points of different trajectories are recombined, like the GANs mentioned in the literature discussion. Using the Hausdorff distance as a measure in these situations would be as follows: One could calculate the Hausdorff distance of every anonymized trajectory towards all original trajectories, take the minimum Hausdorff distance found this way, and finally average these statistics over the entire set.

Since our chosen algorithms result in anonymous datasets in which every trajectory corresponds to one trajectory in the original set, we do not have to use this method. However, we will use the Hausdorff distance as our displacement measure since it seems to be more robust against changes to the number of trajectory points that can be omitted or introduced. This gives us one candidate metric to investigate:

- Average Hausdorff distance between anonymized trajectories and their original trajectory (m)

3.2.3 Centroid displacement

The last candidate in this section considers the displacement of trajectory centroids. Similar to the previous approaches that measure trajectory displacement, the distance that a trajectories' centroid is moved through a privatisation process may present an alternative insight in the information loss introduced. For these measures a way to further penalize a big displacement over a small one would also be simply to square the change in centroid location.

While there are different ways to take the centroid of a trajectory, I have chosen the simplest way: by averaging all points of a trajectory. This is the most obvious way because the interval between the points is a constant 15 seconds.

The following candidate can be added to our list:

- Average centroid displacement (m)

3.3 Privacy Preserving Data Publication Algorithms

Even though a comparison between an algorithm providing differential privacy and an algorithm providing k-anonymity was originally proposed, due to issues anonymizing the dataset with a k-anonymity providing algorithm, instead two versions of a differential privacy algorithm were used in the comparison using the candidate metrics.

3.3.1 Differential Privacy

A straightforward implementation of differential privacy in a geospatial context is that of geo-indistinguishability which was first proposed by Chatzikokolakis et al. [2]. It takes the basic idea of ϵ-differential privacy from Dwork et al [14], namely that of achieving privacy through incorporating Laplacian noise in the publication of privacy-sensitive data while keeping track of the privacy budget ϵ. By adding noise from a 2-dimensional planar Laplace distribution to the location of an individual, the location of this individual within a certain privacy radius will be indistinguishable from other anonymized locations that lie within this radius, since the distributions of possible reported locations are indistinguishable from one another. This is achieved by drawing from a gamma distribution with shape parameter 2 and a varying scale parameter that is equal to the protection radius divided by the available noise budget. This means that a smaller noise budget will lead to larger amounts of noise being added to the data.

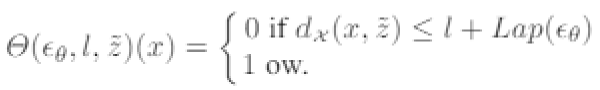
Important to note is that the anonymized locations that are generated by the addition of noise do not necessarily lie within the chosen privacy radius, as large amounts of noise have to be added for a relatively small radius of guaranteed privacy, especially when the available privacy budget is limited.

Another unfortunate property of the mechanism proposed by Chatzikokolakis et al. is that the mechanism in itself is not suitable for trajectories that consist of multiple locations, since it does not take the correlation between these into account. One can imagine a case where an object is standing still. An attacker with that prior knowledge will be able to interpolate the various points that are generated and given enough of those, will be able to reconstruct the original trajectory. Such a naïve implementation that applies planar Laplacian noise to a trace of n locations independently will satisfy $n\epsilon$-geo-indistinguishability instead of ϵ-geo-indistinguishability.

3.3.2 The Differential Privacy Algorithm

A solution to this problem has also been proposed by Chatzikokolakis et al [7]. They propose a mechanism that aims to spend a predetermined privacy budget ϵ more efficiently. Their mechanism consists of three modular parts: a prediction function, a noise mechanism and a test mechanism.

When a trace is being anonymized, for every point in the trajectory, the previously reported locations are given to the prediction function. The prediction function then predicts the next location that is to be reported, noted as $\tilde{z}$. This prediction is then passed to the test mechanism Θ the pseudocode of which can be found in Figure 1, as presented in the original paper. This is a differentially private mechanism that tests if the predicted location is within a certain distance to the real location, using Laplacian noise determined by the test budget ϵ_θ and a threshold value l. Then, if the predicted location passes the test mechanism, it is chosen as the location to be published, otherwise, the noise mechanism can be used to generate a new location. This means when the test mechanism is passed, no budget will have to be spent on the noise mechanism. The noise function used in this thesis randomly draws from a gamma distribution with shape 2.0 and scale set to the protection radius divided by noise budget ϵ_N. By determining the next reported location from the previously reported locations, no new information is disclosed which is why the prediction mechanism does not have to spend any budget or implement any noise.



$$\Theta(\epsilon_\theta, l, \tilde{z})(x) = \begin{cases} 0 \text{ if } d_{\mathcal{X}}(x, \tilde{z}) \leq l + Lap(\epsilon_\theta) \\ 1 \text{ ow.} \end{cases}$$

Figure 1: Test mechanism using ϵ_θ and l to test prediction z̃ against the actual location x

Both during a noise step and during a test step some information is disclosed, the amount of which is managed by the available budget. A budget manager decides what amount of the budget can be spent on each step. Chatzikokolakis et al. propose two approaches to the budget manager. One attempts to produce data with a constant utility by allowing the budget consumption rate to vary, another aims to keep the budget consumption rate constant by allowing for a varying utility of the data produced.

Geo-indistinguishability is in itself the most straight-forward implementation of differential privacy for trajectory data. However, the improved algorithm that uses a predictive mechanism and a budget manager appears to be the superior choice for real applications. This thesis aims to draw a comparison in between a simple geo-indistinguishability algorithm that equally divides the available budget for noise generation all datapoints of a trajectory and the predictive mechanism.

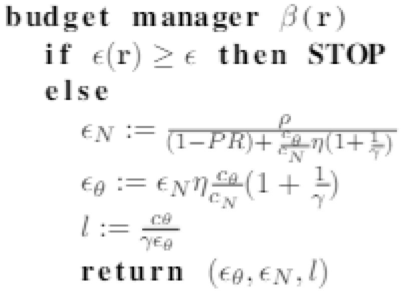
3.3.3 Implementing the Differential Privacy Algorithm

While the algorithm in the original paper [7] was fully implemented, during implementation there were some unclarities and some adjustments worth mentioning.

The paper proposes two different budget managers. One fixes the utility under a varying budget consumption, the other fixes the budget consumption rate with varying utility. Since the length of a trace is known beforehand the available budget can be chosen accordingly, the latter option was chosen. To prevent massive differences in noise levels between short and long trajectories, the budget parameter was taken as a budget per trajectory point. In reality a good privacy budget might lie around $\epsilon = 1$, spread over the duration it takes to realistically travel the distance of the protection radius, after which the budget can either be reset or the tracking can be terminated. However, determining this for my research would only over-complicate any comparison that can be drawn from the resulting data. A realistic implementation might choose longer intervals in between location samples in order to improve privacy guarantees.

The budget manager determines two budget values at every step —the locations that make up the trajectory are anonymized sequentially with each location taking one step to anonymize—, the noise budget ϵ_N and the test budget ϵ_Θ. After doing so the threshold value l that is used in the test is determined by the budget manager as well. To calculate the new values the ideal budget consumption rate ρ and the ratio of successful predictions PR are used in addition to the previous budget, of which the previous ϵ_N and ϵ_Θ are used to calculate the new budget values. The pseudocode that shows this calculation, taken from the original paper, can be found in Figure 2.

Additionally, the budget calculation uses two constants η and γ which aren't deeply discussed in the original paper. Gamma often represents a component of decay, and it appears to do so within the proposed budget managers as well. The authors report 0.8 to be a good gamma value during



```
budget manager $\beta(\mathbf{r})$
  if $\epsilon(\mathbf{r}) \geq \epsilon$ then STOP
  else
    $\epsilon_N := \frac{\rho}{(1-PR)+\frac{c_\theta}{c_N}\eta(1+\frac{1}{\gamma})}$
    $\epsilon_\theta := \epsilon_N \eta \frac{c_\theta}{c_N}(1+\frac{1}{\gamma})$
    $l := \frac{c\theta}{\gamma\epsilon_\theta}$
    return $(\epsilon_\theta, \epsilon_N, l)$
```

Figure 2: Fixed budget comsumption rate using ϵ, ρ and PR

their experiments, and the η value to be able to "go as low as 0.5 without issues". When you look at the mathematics, both in the calculation of the noise budget as well as the test budget, the proportion between the test and noise budget is multiplied with $\eta(1 + 1/\gamma)$, while the calculation of the l threshold only uses γ in its multiplication. This indicates that η sets a preference for the division of budget over the test function and generating noise, while γ implements some form of natural decay. This natural decay can be useful in situations where it is uncertain how long a trajectory will continue, or where the total available budget is not known during the privacy-preserving registration of a trajectory. While experimenting with the γ value, it appeared that using a value below 1 led to the available privacy budget not being fully consumed. Additionally, choosing an η of 0.5 and a γ of 1 result in all three of the aforementioned multiplications being crossed off against each other, since they simply come down to multiplications with 1. An observation here is that these settings open up a way for the computational optimization of the algorithm, something that could potentially be beneficial when confronted with large amounts of data having to be privatized at once.

For the first point of a trajectory an initial budget has to be supplied, but recommendations for an appropriate budget were not specified by the authors. The initial noise-budget was set to be ρ, which represents the target consumption of budget per step, and also the consumption of the first step. The original paper keeps ρ at a constant, because the amount of location data yet to come is not known beforehand. However, to spend the full budget over the course of a single directory, and because the amount of points that is yet to come is always known beforehand, I chose to re-estimate ρ during every step, by dividing the available budget equally over the points that had yet to be processed. Not doing so resulted in a budget consumption rate of 50-60%, while recalculating the target budget consumption during every step resulted in the consumption of the full budget over the course of every trajectory. Since there can be no testing when anonymizing the first datapoint, ρ is spent directly on the noise generation, which means ρ is always unchanged after the first iteration.

While the initial noise budget is easy to determine, an initial test-budget has proven to be difficult to set. The authors propose that it should be small relative to the noise budget, in order to minimize the budget loss during a miss, and maximize budget gains during a hit of the testing mechanism. Even though it is not used and not spent in the first step of the algorithm, it must be set to a positive value that is a small fraction of the initial noise-budget, which is equivalent to ρ. Intuitively one could think that the initial budget parameters should accumulate to 1, in order to encourage spending all of it, and thus the initial test-budget should be 0. However, since a multiplication with the previous test-budget is used to divide the rest of the budget between noise and testing during every budget calculation step, an initial test-budget of 0 results in a noise budget that will always be equal to the initial ρ, and the test-budget always being equal to 0. Essentially, under these circumstances the algorithm reverts to a form without budget-management. Through some empirical experimentation, 0.1 appeared to be a reasonable setting for the initial test-budget, which has been used for the generation of all anonymized data.

The prediction mechanism used by Chatzikokolakis et al. is the simplest one imaginable. Appropriately named "Parrot", given the previously disclosed points, the mechanism parrots back the last reported location. Intuitively one would presume that the optimal mechanism would attempt to follow a user on their path. However, in practice it is generally very hard to distinguish the actual movement pattern of a user from the noise added by the differential mechanism, whose influence on the observed anonymized trajectory points will exceed the influence of the underlying human behaviour under most parameter settings. The theoretical optimal mechanism would always report the current real location, but aside from the fact that this is impossible, this would be quite detrimental for the privacy guarantees offered.

3.4 Privacy Parameters

In order to draw a better comparison, a wide range of privacy parameters should be taken into consideration, so that the relationship between these parameters and their influence on the proposed utility measures can be explored. This may subsequently help further draw a comparison in between different PPDP algorithms. The best approach to choosing these parameters is to follow existing literature. In algorithms based on differential privacy, common values for ϵ appear to lie in the range $[0.1; 2.0]$. The parameters chosen for the experiments are as follows: $[0.1, 0.2, 0.5, 1, 1.5, 2]$

Geo-indistinguishability also takes a privacy radius as a parameter. In an urban environment a limited indistinguishability-radius can be sufficient since places of interest are very close to one another. The range of parameters chosen for the protection radius are as follows:
$[25m, 50m, 100m, 150m, 200m, 300m, 400m, 500m]$.

3.5 Analysis Strategy

For each combination of privacy parameters —adding up to 48 different privacy settings in total—, 100 anonymizations are made. The for every metric, the statistics are calculated on every anonymization, and the statistics of these 100 anonymizations are then averaged. This leaves 48 sets of sum, minimum, maximum, average and standard deviation for each metric. This process is repeated for every PPDP algorithm, on every dataset, so in the case of this thesis, four times.

In the next section, the method of analysis used on the raw data resulting from the experiment will be discussed. Through extensive experimentation and analysis of the data a common method of analysis has been constructed that has been used to analyze every candidate measure that was applicable to the chosen algorithms. This subsection will explain step by step how this methodology was constructed and how the choices made in this process were motivated.

3.5.1 Initial Exploration

The first exploratory plot of the results was made grouping the obtained average measurements by the protection radius. After attempting various types of regression, the models best fit to explain the underlying data were polynomial, taking the form ax^b. Coefficients of determination (R^2) of 1 or closely approaching 1 were observed. This means these models were able to explain the variance of the data very well when compared to the dependent variable —in these cases the average value observed—, namely 99 to 100 percent better, and since the percentage of standard deviation explained can be found by taking 1 minus the square root of the R^2 value, it can be deducted that the standard deviation of the observed data is explained 90 percent better or more by these models, when compared to the dependent variable.

On the other hand, grouping by privacy budget, and then by protection radius resulted in plots that were more linear in appearance. After exploring various regression strategies, linear regression was determined to be the best strategy. Despite the fact that it is clear that polynomial relationships are able to explain the underlying data better —which is always the case when comparing a linear and a quadratic function— the resulting comparison was very straightforward, and since the resulting linear models had to be subjected to another round of regression, this turned out to be the only suitable method of analysis. Additionally, looking at the way the noise algorithm uses the protection radius, a linear dependency is what one would expect here, too. Since R^2 values in the range of $0.96 - 1$ were observed, mostly in the high end of this range, it is indicated that these models are able to explain the underlying data variation better than the independent variable by this percentage, and the standard deviation by 80 percent or more.

3.5.2 Linear Regression

Models obtained through linear regression take the form $ax + b$. However, these models attempt to explain the influence of the chosen protection radius on the observed measurements of the metric at hand. Since a protection radius of 0 results in an anonymization that is identical to the original, most metrics will result in a value of 0 when the protection radius is set to 0. The exceptions here are speed, duration and length metrics, which will yield results identical to the average measurements of the original dataset. This observation led to the choice of setting the intercept during linear regression to the metric average result obtained from comparing the original dataset to itself in all cases. Even though allowing the b-component to vary in this way results in linear models that provide a marginally better explanation of the underlying data, this is likely to be the result of overfitting, and the varying b-components obtained this way are difficult to include in any further analysis steps.

The linear regression models obtained this way all result in a single variable, which is the a-component obtained from the resulting formula. This component represents the observed metric measurements divided by the chosen protection radius. Differently put, these values represent the influence of the chosen protection radius on the observed metric measurements under a certain privacy budget. Since the data was grouped by the privacy budget ϵ, after the previous step, for each privacy algorithm, and for each candidate metric, there are six linear models whose a-component represents this influence under a specific privacy budget. The next step is to investigate whether the influence of the chosen privacy budget on this a-component can be explained by another regression model.

3.5.3 Second Layer of Linear Regression

When the a-component values are subjected to another round of regression analysis, another exponential relationship can be observed, that is able to explain the underlying data very well, often approaching an R^2-value of 1. However, the same difficulties arise when an attempt to compare these models is made, since there is no straightforward way to compare the components that result from such an analysis. After some experimentation, a different approach was found. By putting the privacy budget ϵ on the horizontal axis, and the inverse of the a-component on the vertical axis instead of the component itself, another linear relationship can be observed. This relationship expresses the influence of the chosen privacy budget on the relationship between the chosen protection radius and the observed measurements. This results in a new and final a-component that expresses the influence of both the privacy budget and the protection radius on the metric measurements. To avoid confusion this component will be called the c-component from now on. Unfortunately no applicable literature was found for this form of 2-staged linear regression. However, since the underlying linear models already incorporate some uncertainty, this needs to be taken into account as well when interpreting the final results. Since there was no literature found on a proper approach to this problem, I propose an addition to my methodology that could possibly deal with some of these possible critiques in the discussion.

3.5.4 Resulting Values

These final c-component values allow for a straightforward comparison between the different algorithms that are being investigated. This comparison results in a single value that attempts to express the relative performance of the two algorithms as a percentage. However, when comparing the performance of the two algorithms a more realistic plot of the relative difference in performance can be made using the previously obtained a-components instead, which shows a more nonlinear relationship of the relative performance under varying ϵ. This graph is more useful for choosing the privacy parameters in a real-world scenario, since especially at the lower end of the possible ϵ-values, the observed differences in performance do not always reflect the generalized performance comparison obtained from the c-components.

3.5.5 Significance and Comparison

The linear regression results obtained from the experiments come with a 95% uncertainty range. Whether or not a linear model is statistically significant within 95% certainty can be determined by checking whether 0 lies within the uncertainty range, since if it does, the existence of a relationship can not be certain within the 95% certainty range. Additionally, in the residual plots, almost in every case some form of non-linear function can be seen, which indicates that the linear models are unable to fully explain the data by themselves. However, since these residual relationships offer very limited improvement towards explaining the data, are very similar in most cases, and are difficult to compare, the results will focus on the linear relationships found only.

Not only can the 95% ranges be used to check if a linear relationship is statistically significant, they can also be used to determine if two models differ with 95%. If there is no overlap, there is a significant difference found for a certain metric. The results will both compare the models' a-components, and their c-components.

4 Results

Out of the candidate metrics, the duration metrics always returned a value of 0, since both algorithms did not change the duration. These metrics were omitted. Additionally, the Absolute Length Change metric has not been fully explored, because the results obtained from it were identical to those obtained from the Average Length metric. This means the metrics that are fully explored in this section are as follows:

- Average Length
- Average Speed
- Relative Length Change
- Hausdorff Distance
- Centroid Displacement

Recall from subsection 3.5 that for each algorithm, for each dataset, there are 48 anonymizations. That means for every metric a total of $2 * 2 * 48 = 192$ sets of a sum, minimum, maximum, average and standard deviation were obtained and analyzed using the methodology described in subsection 3.5.

This section will go through the results obtained for each of the metrics, firstly discussing the results for the anonymizations of natural data and subsequently comparing these to the results found with the synthetic dataset.

The tables containing the complete metric results for each anonymization can be found in the appendix, together with a full overview of the regression statistics. This section will limit itself to the display of the most interesting and relevant values and statistics found, but considering the sheer size of the output it is impossible to go into every detail in this report. Since no statistically insignificant linear models were found, all values presented are significant.

4.1 Length and Speed Statistics

The first thing that becomes obvious from the results of the length and speed measurements —as well as the length change metrics discussed in the next subsection— is that they both increase massively. The noise introduced by the geo-indistinguishability mechanism is theoretically able to move two subsequent points a multitude of protection radii apart, while a decrease in total trajectory length is much more unlikely. A PPDP algorithm that uses pruning instead of manipulation of all individual points would lead to very different measurements that may have to be interpreted differently.

In the case of these algorithms, an increase in length is the direct result of the amount of noise added to the trajectories, which is why a lower increase of length indicates better performance. However, in different cases where anonymized trajectories tend to be shorter than their original counterparts, a lower decrease of length can be an indication of better performance as well.

4.1.1 Average Length

While the average trajectory length in the Porto dataset was 5.1 kilometres, averages of hundreds of kilometres have been observed for various settings, which can be seen in Figure 3. While the increase in length is an indication to the amount of noise included in an anonymization, it is possible that reconstructing the original trajectories from the anonymized database could yield lengths that are much closer to the original trajectories' length, which in turn could potentially be a better indication of utility loss.

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	1506.1	745.66	289.96	138.67	88.733	64.038	5154.7	0.0076
Predictive Model	705.23	341.77	124.76	53.833	31.069	20.213	5154.7	0.0226
95% range overlap	-	-	-	-	-	-		-

Table 2: Results: Length Metric, Porto Dataset

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	362.29	179.77	70.289	33.888	21.75	15.754	929.8	0.031
Predictive Model	526.58	168.7	40.479	14.581	8.0586	5.2622	929.8	0.0859
95% range overlap	-	-	-	-	-	-		-

Table 3: Results: Length Metric, Tony-900 Dataset

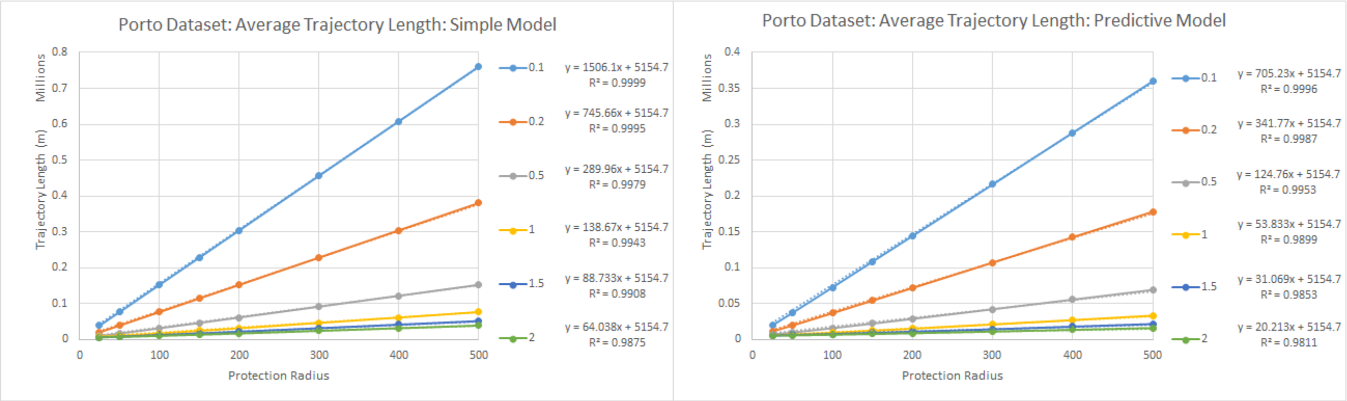


Figure 3: First Layer Regression Results: Average Length Metric: Porto Dataset

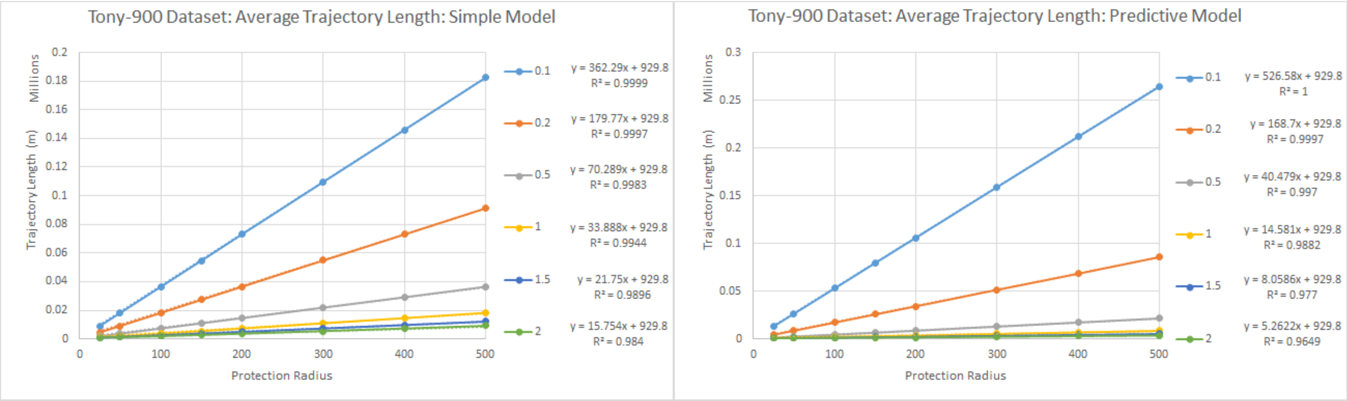


Figure 4: First Layer Regression Results: Average Length Metric: Tony-900 Dataset

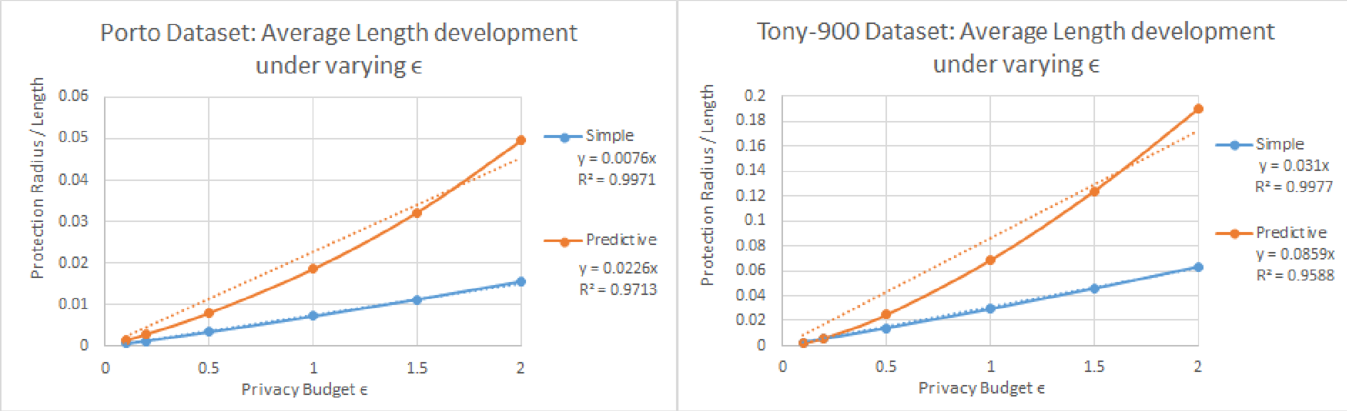


Figure 5: Final Regression Results: Average Length Metric

As can be seen in Figure 3, with real data the predictive model yields a lower increase in length than the simple model. However, in the case of the synthetic dataset, the predictive model trained with a privacy budget of 0.1 performs significantly worse, while in the case of a larger privacy budget it performs better. This can be seen in Figure 4.

This difference could partially be due to the synthetic trajectories being much shorter, in which case the predictive mechanism is not able to recover from failing the test-step a few times at the start of anonymizing a trajectory.

Additionally, it could be because these trajectories are straight, which means every displacement of a point, regardless of the direction, can only lead to an increase of trajectory length—except for the first and last point of the trajectory— while in the case of the natural data, there is the theoretical possibility that a modification of a trajectory point results in a shorter trajectory length, when a trajectory does not lie on a straight line.

4.1.2 Average Speed

As with the Average Length Metric, the speed increases drastically under most settings, which can be observed in Figure 6 and 7. While the original natural dataset had an average speed of 6.8 km/h, the anonymized versions go from a 0.2 km/h increase, with the largest available budget and the smallest protection radius, to an increase of around 1000 km/h, when the smallest privacy budget that was experimented with is combined with the largest protection radius. As with the length statistics, the speed statistics are mostly an indication of the amount of noise that was added to the trajectories, which makes it more of an indirect indicator towards utility preservation than a direct one. The average speed of reconstructed trajectories, that attempt to cancel out some of the noise, could provide some additional insights into the real utility loss with regard to the measurement of object speed.

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	1.9434	0.962	0.3739	0.1787	0.1142	0.0823	6.785	5.9111
Predictive Model	0.924	0.4478	0.1635	0.0706	0.0407	0.0264	6.785	17.27
95% range overlap	-	-	-	-	-	-		-

Table 4: Results: Speed Metric, Porto Dataset

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	1.9483	0.9667	0.378	0.1822	0.117	0.0847	5	5.7619
Predictive Model	2.8315	0.9072	0.2177	0.0784	0.0433	0.0283	5	15.976
95% range overlap	-	-	-	-	-	-		-

Table 5: Results: Speed Metric, Tony-900 Dataset

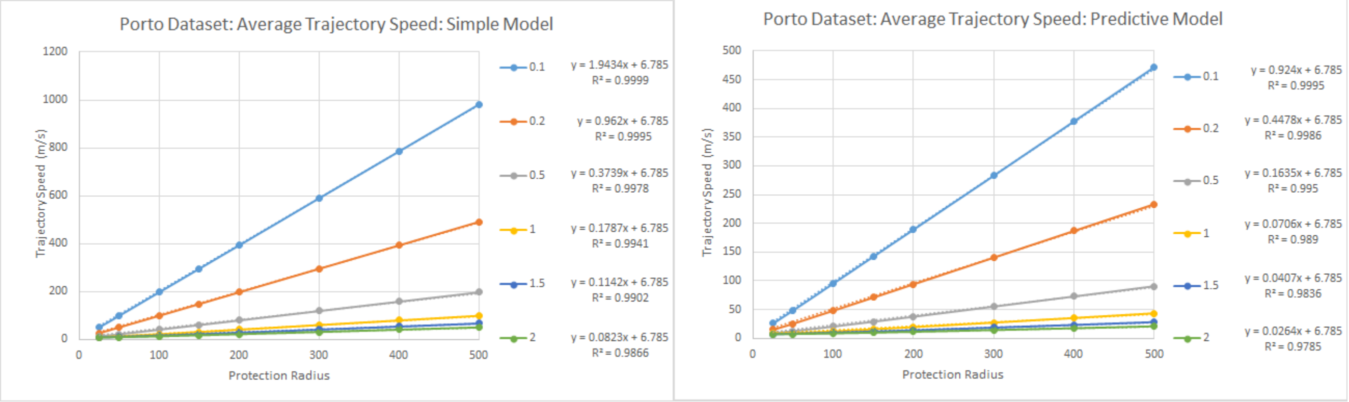


Figure 6: First Layer Regression Results: Average Speed Metric: Porto Dataset

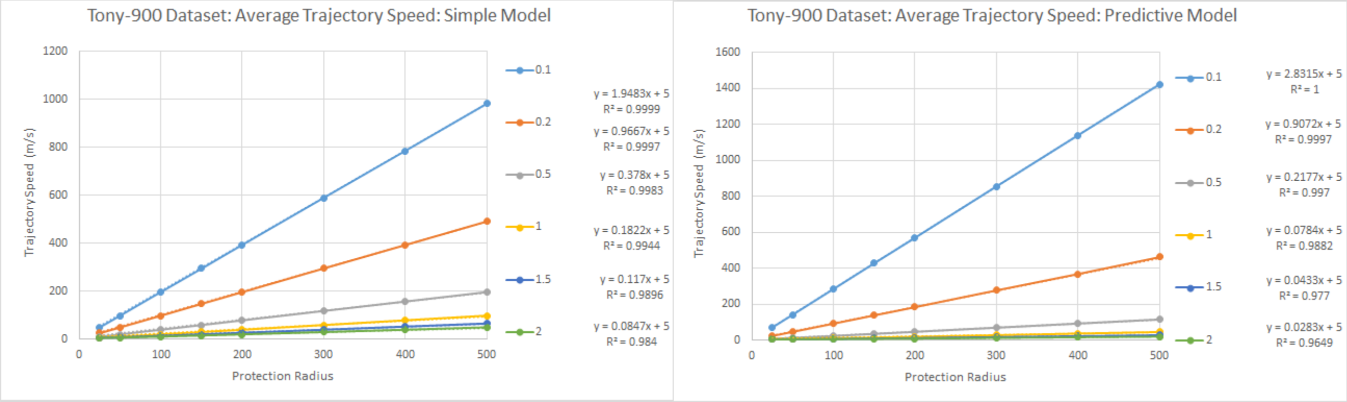


Figure 7: First Layer Regression Results: Average Speed Metric: Tony-900 Dataset

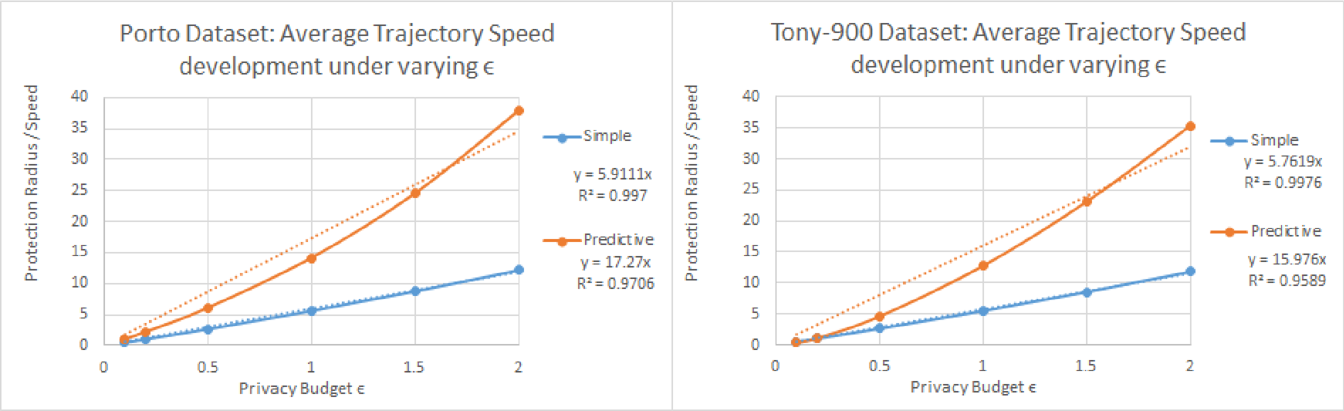


Figure 8: Final Regression Results: Average Speed Metric

Like with the Average Length Metric, the predictive model outperforms the simple model in every case on the natural dataset. However, for the synthetic dataset the same exception can be seen. At a privacy budget of 0.1 the simple model performs better than the predictive model, which only performs better when given a higher privacy budget.

Another interesting observation is that the simple model produces nearly identical results for both the natural and synthetic dataset, something that was not the case with the Average Length Metric. This is likely due to the fact that trajectory speed is essentially a local property that is being locally influenced by the algorithm, while the amount of increase of trajectory length depends on the length of the original.

4.2 Length Change Metrics

In this section the metrics that measure the change in trajectory length will be discussed.

4.2.1 Absolute Length Change

The Absolute Length Change Metric is nearly identical to the Average Length Metric. The slight differences in measurements and model outcomes are likely due to some very short trajectories that often acquire a length of zero through the predictive mechanism. While the absolute length change metric was designed especially for these cases, for these two algorithms this does not provide any more insight into the performance of the two algorithms in addition to what is already provided by the Average Length Metric.

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	1506.1	745.66	289.96	138.67	88.733	64.038	0	0.0076
Predictive Model	705.23	341.78	124.77	53.84	31.08	20.231	0	0.0226
95% range overlap	-	-	-	-	-	-		-

Table 6: Results: Absolute Length Change Metric, Porto Dataset

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	362.29	179.77	70.289	33.888	21.75	15.754	0	0.031
Predictive Model	526.58	168.7	40.48	14.586	8.0712	5.2846	0	0.0856
95% range overlap	-	-	-	-	-	-		-

Table 7: Results: Absolute Length Change Metric, Tony-900 Dataset

4.2.2 Relative Length Change

Because of the way the change of length is measured, the values observed for the Relative Length Change Metric are negative when an increase in length is observed. In some cases increases of 100 times or more were observed.

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	-0.4525	-0.2247	-0.0883	-0.0428	-0.0277	-0.0203	0	-24.213
Predictive Model	-0.2221	-0.1086	-0.0407	-0.0181	-0.0108	-0.0072	0	-64.436
95% range overlap	-	-	-	-	-	-		-

Table 8: Results: Relative Length Change Metric, Porto Dataset

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	-0.5663	-0.1814	-0.0435	-0.0157	-0.0087	-0.0057	0	-79.438
Predictive Model	-0.3897	-0.1933	-0.0756	-0.0364	-0.0234	-0.0169	0	-28.851
95% range overlap	-	-	-	-	-	-		-

Table 9: Results: Relative Length Change Metric, Tony-900 Dataset

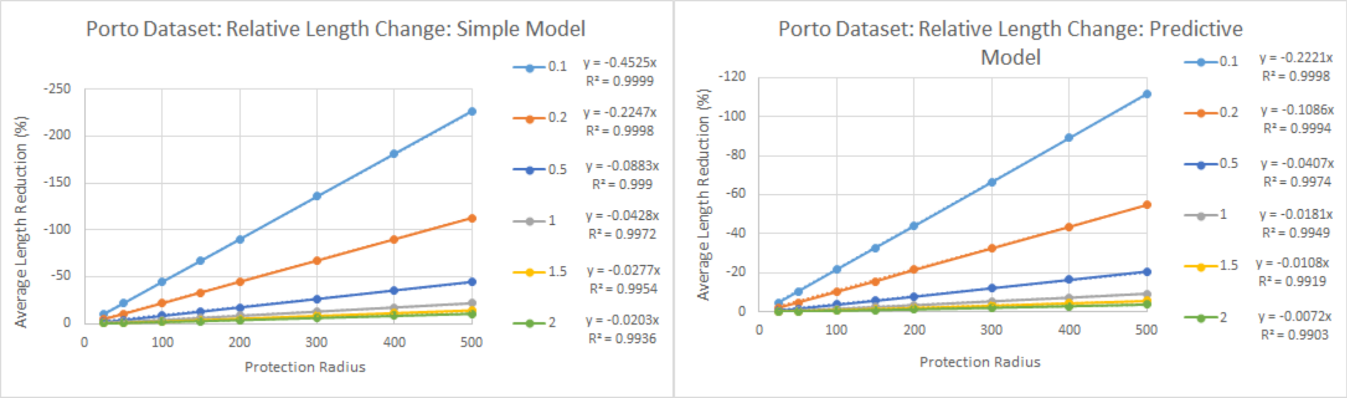


Figure 9: First Layer Regression Results: Relative Length Metric: Porto Dataset

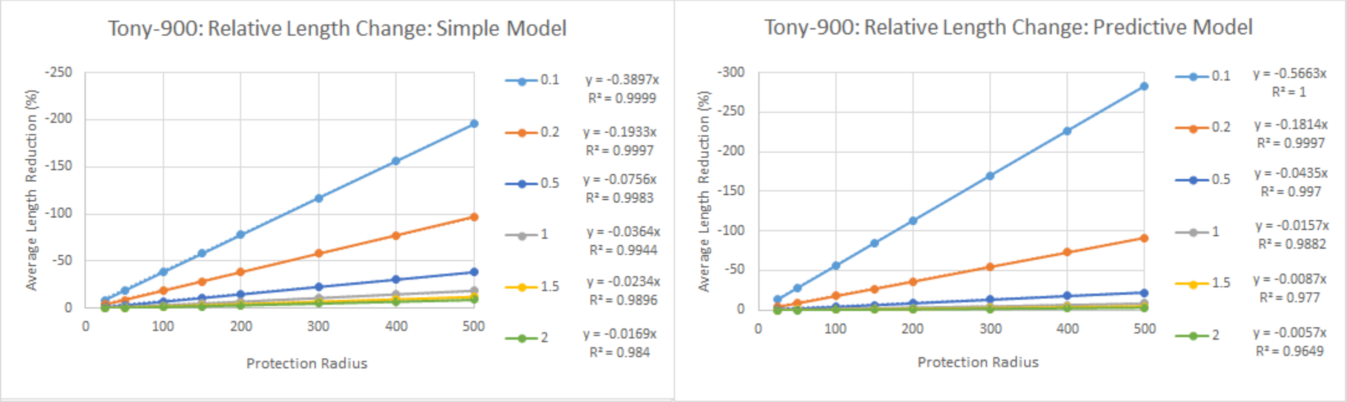


Figure 10: First Layer Regression Results: Relative Length Metric: Tony-900 Dataset

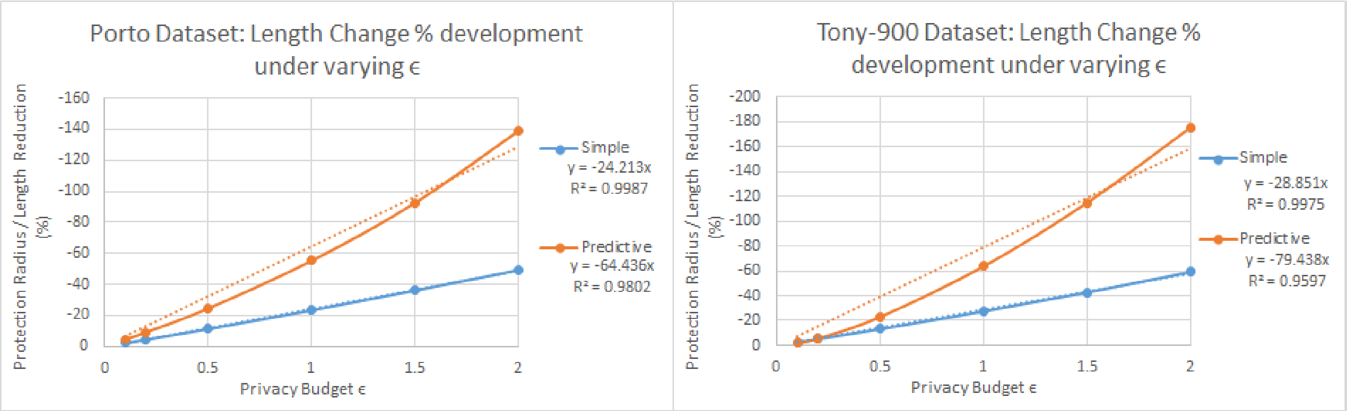


Figure 11: Final Regression Results: Relative Length Metric

Similarly to the Average Length Metric and Average Speed Metric, the predictive model performs better on the natural dataset than the simple model in every case. However, again, the simple model performs better for the synthetic dataset at a privacy budget of 0.1.

4.3 Hausdorff Distance Metric

While the Hausdorff distances taken could be very similar to a measure of the maximal displacement that occurs within the points of a trajectory, it is more forgiving towards points that have been moved close towards either the traveled or the upcoming parts of the full trajectory. When we imagine a scenario using natural data, one could argue that a point where a user is headed or even one they just visited could indeed be a more useful location to receive a location based service from.

The average Hausdorff distance of trajectories to their original counterparts can be seen as a form of a maximal expected error, measured in metres, which makes it a more direct indicator of the utility loss introduced than the previous metrics, in particular regarding the utility from a user perspective.

It is clear that regarding the data utility a lower value indicates better utility of the data produced as well. However, reconstructing the anonymized trajectories could yield much more favourable results, as is the case with the previous metrics.

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	60.403	29.308	11.104	5.3139	3.4666	2.5672	0	0.1925
Predictive Model	47.804	23.01	8.5584	4.0217	2.5873	1.8966	0	0.2586
95% range overlap	-	-	-	-	-	-		-

Table 10: Results: Hausdorff Distance Metric, Porto Dataset

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	48.799	24.049	9.2517	4.4345	2.8701	2.1132	0	0.233
Predictive Model	79.848	27.853	7.809	3.2502	2.014	1.4625	0	0.3301
95% range overlap	-	-	-	-	-	-		-

Table 11: Results: Hausdorff Distance Metric, Tony-900 Dataset

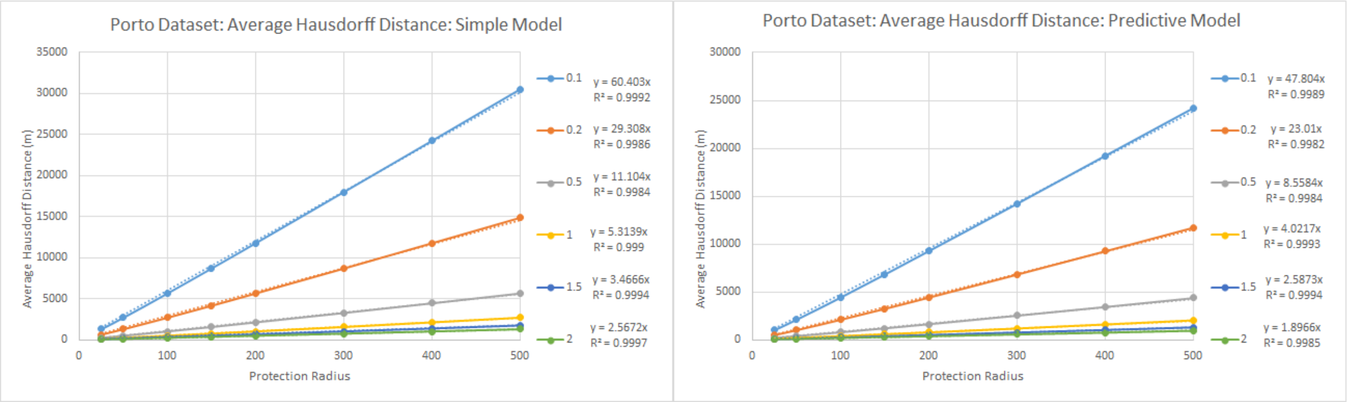


Figure 12: First Layer Regression Results: Hausdorff Distance Metric: Porto Dataset

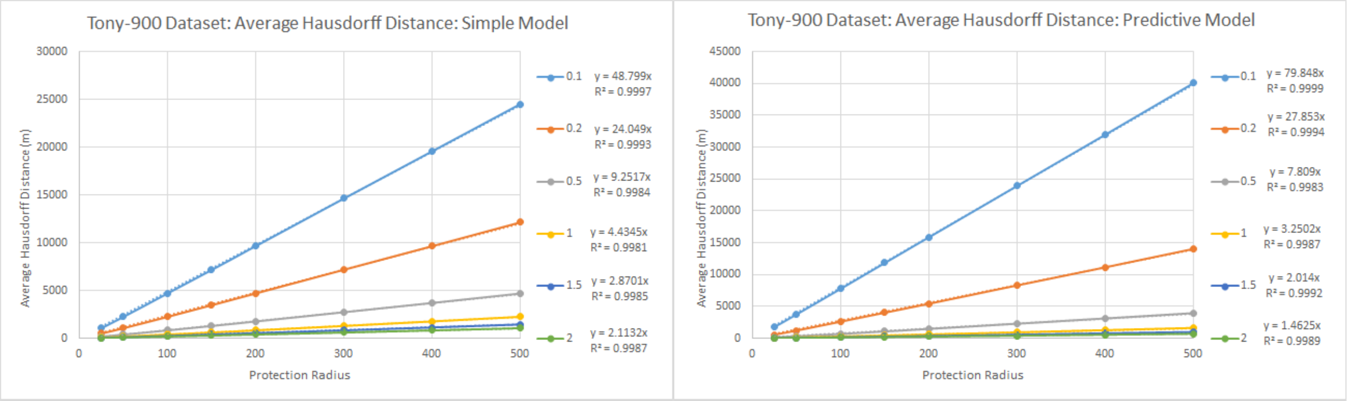


Figure 13: First Layer Regression Results: Hausdorff Distance Metric: Tony-900 Dataset

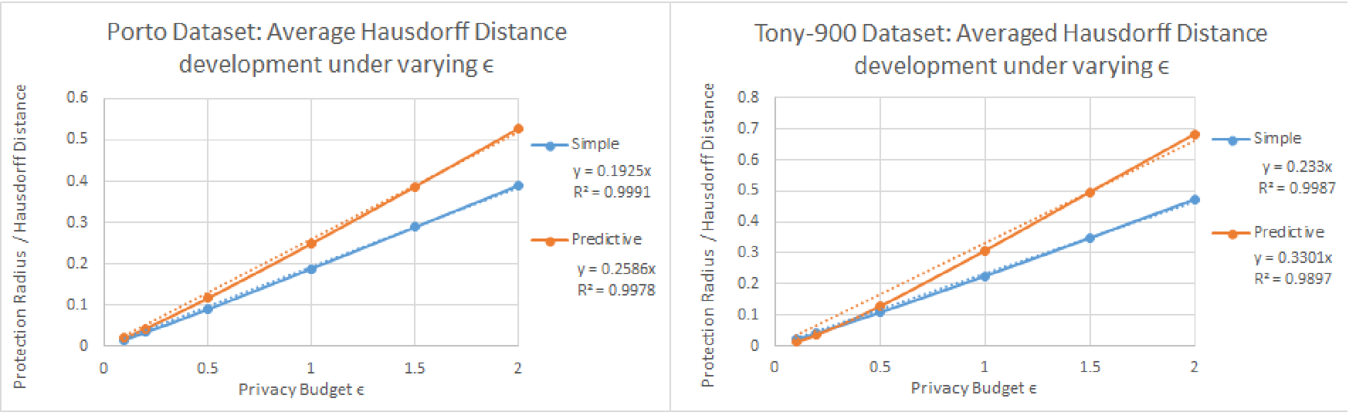


Figure 14: Final Regression Results: Hausdorff Distance Metric

Similarly to previously discussed metric results, the predictive mechanism outperforms the simple mechanism on the real dataset in every setting that was experimented with. With the synthetic data the predictive model performs worse when given a low privacy budget. In this case, not only a privacy budget of 0.1 leads to worse performance when choosing the predictive model, a budget of 0.2 leads to a slightly worse performance as well.

4.4 Centroid Displacement Metric

While at first glance the Centroid Displacement Metric appears to indicate some form of introduced error, for algorithms providing geo-indistinguishability through addition of noise, this is likely not to be the case.

Imagine a trajectory with two points, both lying on the origin. In this scenario the centroid of the trajectory lies on the origin too. Now imagine a PPDP algorithm that would move one point one kilometre to the left, and the other a kilometre to the right. This would constitute a massive loss of information, while there is no change in the centroid position at all that indicates this. Even though this metric only measures the change in global location of the trajectories, these measurements can still tell us something about the behaviour of the investigated algorithms, since they still measure a change in the statistical qualities of the data.

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	3.5212	1.7609	0.7038	0.3523	0.2348	0.176	0	2.8401
Predictive Model	4.0718	2.0443	0.8337	0.4389	0.3134	0.255	0	2.0702
95% range overlap	-	-	-	-	-	-		-

Table 12: Results: Centroid Displacement Metric, Porto Dataset

Component	a						b	c
Privacy Budget ϵ	0.1	0.2	0.5	1	1.5	2		
Simple Model	5.8989	2.9501	1.1789	0.5895	0.3929	0.2956	0	1.6939
Predictive Model	9.6926	3.8132	1.3857	0.7199	0.5089	0.4066	0	1.282
95% range overlap	-	-	-	-	-	-		-

Table 13: Results: Centroid Displacement Metric, Tony-900 Dataset

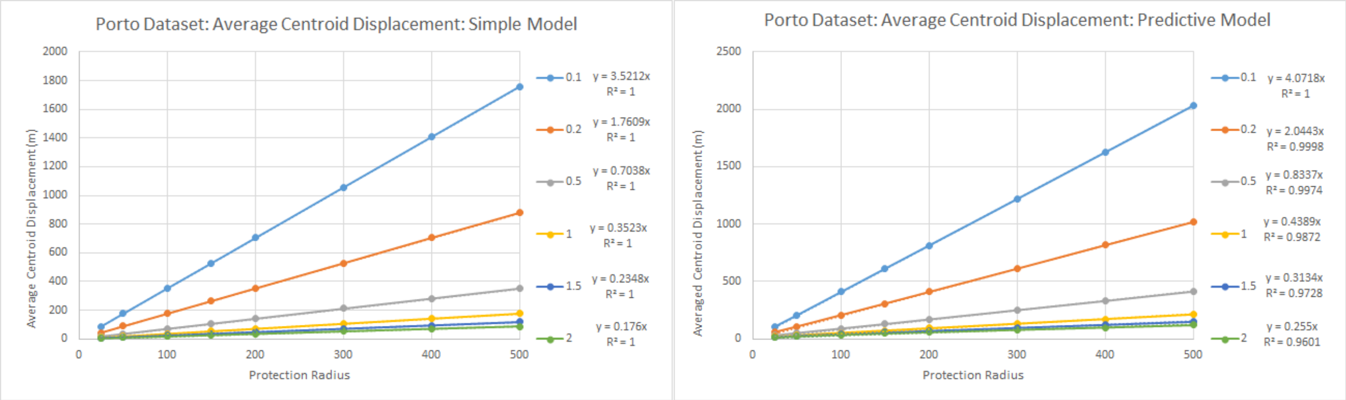


Figure 15: First Layer Regression Results: Centroid Displacement Metric: Porto Dataset

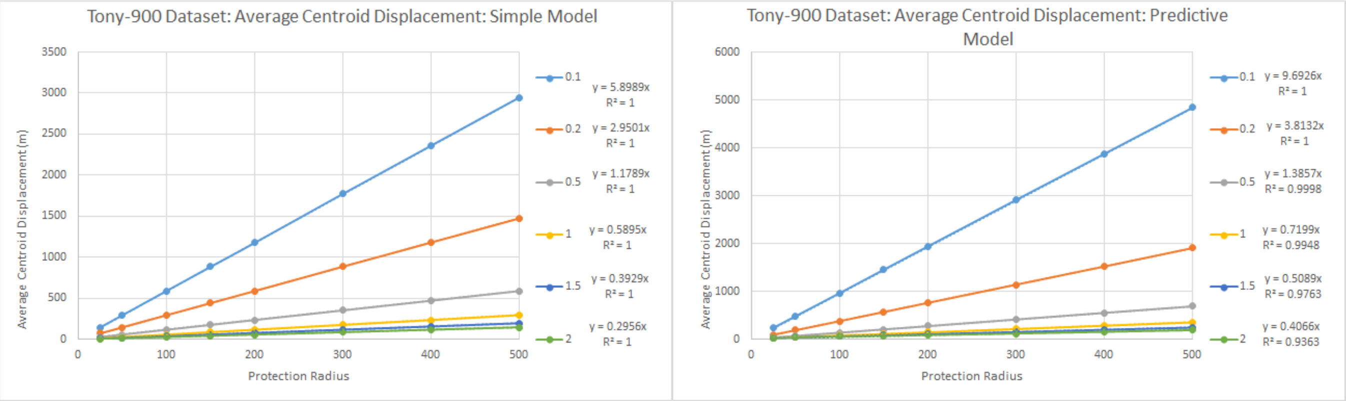


Figure 16: First Layer Regression Results: Centroid Displacement Metric: Tony-900 Dataset

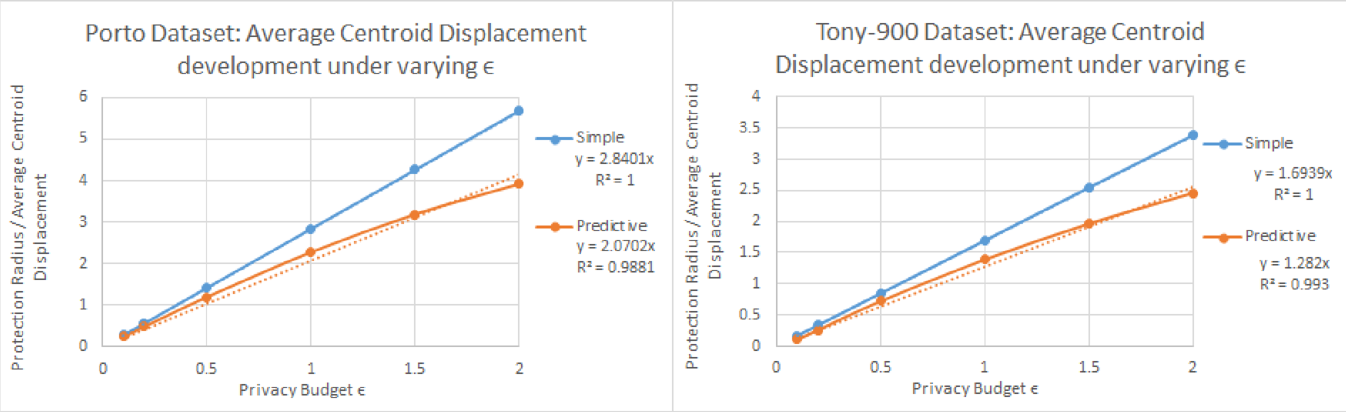


Figure 17: Final Regression Results: Centroid Displacement Metric

In the case of the Centroid Displacement Metric, the predictive model performs worse than the simple model in every situation. An interesting observation is that the regressions done using the simple model were the only ones that did not show clear patterns in the residual plots, which means that the models trained for the simple version of the algorithm are likely to be optimal. An example of this can be seen in Figure 18

A possible explanation for the performance of the predictive model is as follows. Because every time the noise mechanism is used, the centroid is being moved a bit, when the test mechanism allows these values to be

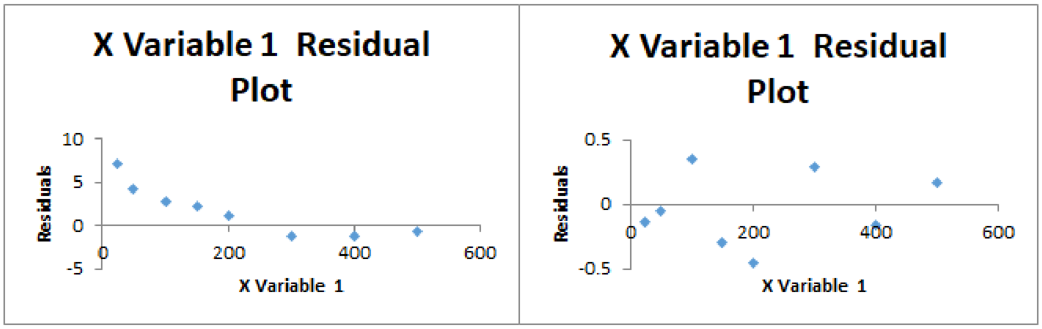


Figure 18: Centroid distance, $\epsilon = 0.1$. Left: predictive mechanism. Right: simple mechanism.

repeated, an extra bias towards a certain direction is introduced. A way to limit this influence on the centroid measurements would be to eliminate these subsequent duplicate trajectory points from the centroid calculation, which in turn might lead to a different conclusion.

It is not unthinkable that when subsequent identical points are not considered during metric calculation, the improved budget allocation of the predictive model could actually provide better results for this metric than the simple model.

4.5 Performance

While the c-components obtained through two-layered regression analysis provide some insight into the general difference in performance that can be expected when choosing between the simple and predictive differential privacy mechanisms, the actual differences in performance appear to be dependent on the privacy budget, especially under the circumstances of relatively short trajectories and/or a relatively low privacy budget. Using the a-components obtained for the different privacy budget settings, performance graphs can be plotted that provide more realistic expectations for each possible privacy budget. These plots can be seen in Figure 19.

Although the models trained in order to obtain these a-components appear to be more accurate than the ones trained on their results —the effect of an increasing protection radius on the metric results is expected to be linear in nature after all—, there may very well still be a discrepancy between the calculated percentages in these plots and the actual performance for each individual protection radius, similarly to the difference between the single performance percentage that can be calculated from the c-components. Nevertheless, these plots will be more useful for determining privacy parameters than the single percentage results obtained from the c-components.

This section will go through these plots and compare the single-value results that can be obtained by only taking the c-components into account with the actual observed performance.

4.5.1 Average Trajectory Length

While the c-components suggest that the predictive mechanism performs 197% better on the Porto dataset and 177% better on the Tony-900 dataset, the performance plots show that for the Porto dataset, the actual difference in performance lies between 113% and 216%, while the difference in performance for the Tony-900 dataset lie within -33% and 199%.

4.5.2 Average Trajectory Speed

Even though the plot of the performance of this metric appears to be identical to that of the average trajectory length metric, the underlying data is very different. However, the nearly identical performance plots do not come as a great surprise, as the metrics are both a measurement of the introduced noise.

The c-components suggest that the predictive mechanism performs 192% better on the Porto dataset and 177% better on the Tony-900 dataset. The performance plots show that in reality, for the Porto dataset this ranges from 110% to 212%, and for the Tony-900 dataset this ranges from -31% to 199%.

4.5.3 Relative Length Change

The c-components suggest that the predictive mechanism performs 166% better on the Porto dataset and 175% better on the Tony-900 dataset. The performance plots show that in reality, for the Porto dataset this ranges from 104% to 212%, and for the Tony-900 dataset this ranges from 104% to 182%.

The graphs of these first three metrics appear to be quite similar to one another. The predictive mechanism performs worse at a low privacy budget, and on the shorter trajectories in the Tony-900 dataset, it is even outperformed by the simple mechanism. On the Porto dataset, the predictive mechanism always performs better than on the Tony-900 dataset for the average trajectory length and speed metrics. The relative length change graph shows that on the Tony-900 dataset, the predictive mechanism does slightly better than it does on the Porto dataset at a privacy budget of 1.5 and above.

4.5.4 Average Hausdorff Distance

The c-components suggest that the predictive mechanism performs 34% better on the Porto dataset and 41% better on the Tony-900 dataset. The performance plots show that in reality, for the Porto dataset this ranges from 26% to 35%, and for the Tony-900 dataset this ranges from -29% to 44%.

Out of all the performance plots, the performance of the Hausdorff metric on the Porto dataset is the only one that appears to approach a constant performance gain regardless of the privacy budget, while in all the other plots the differences in performance seem to vary quite a lot, especially around the lower end of the privacy budget values.

Since these last three metrics all mostly measure the amount of noise introduced, albeit in their own ways, it appears that the predictive mechanism can greatly reduce this noise addition, with the performance relative to the simple mechanism increasing as the available budget increases. For the Porto dataset the performance of the predictive mechanism is always better, while for the Tony-900 dataset the detrimental effects of short trajectories and a low privacy budget can be seen at the lower end of the privacy budget range.

4.5.5 Centroid Displacement

The c-components suggest that the predictive mechanism performs 27% worse on the Porto dataset and 24% worse on the Tony-900 dataset. The performance plots show that in reality, for the Porto dataset this ranges from 14% to 31% worse, and for the Tony-900 dataset this ranges from 39% to 15%.

The performance of the predictive mechanism relative to the simple one seems to worsen as the privacy budget increases, although for the Tony-900 dataset in particular the detrimental effects of a low privacy budget become very clear as well.

4.5.6 c-Component and Performance

It appears that the percentages that can be obtained by comparing the c-components lie towards the high end of the spectrum compared to the ranges found by comparing the a-components. This is likely the result of linear regression being performed on inversed values. Since the inversed a-components from the lower end of the privacy budget spectrum are often a multitude of times smaller than the inversed a-components from the higher end of the privacy spectrum, the error found during regression will be smaller at that end of the budget spectrum as well, which introduces a bias towards the larger values.

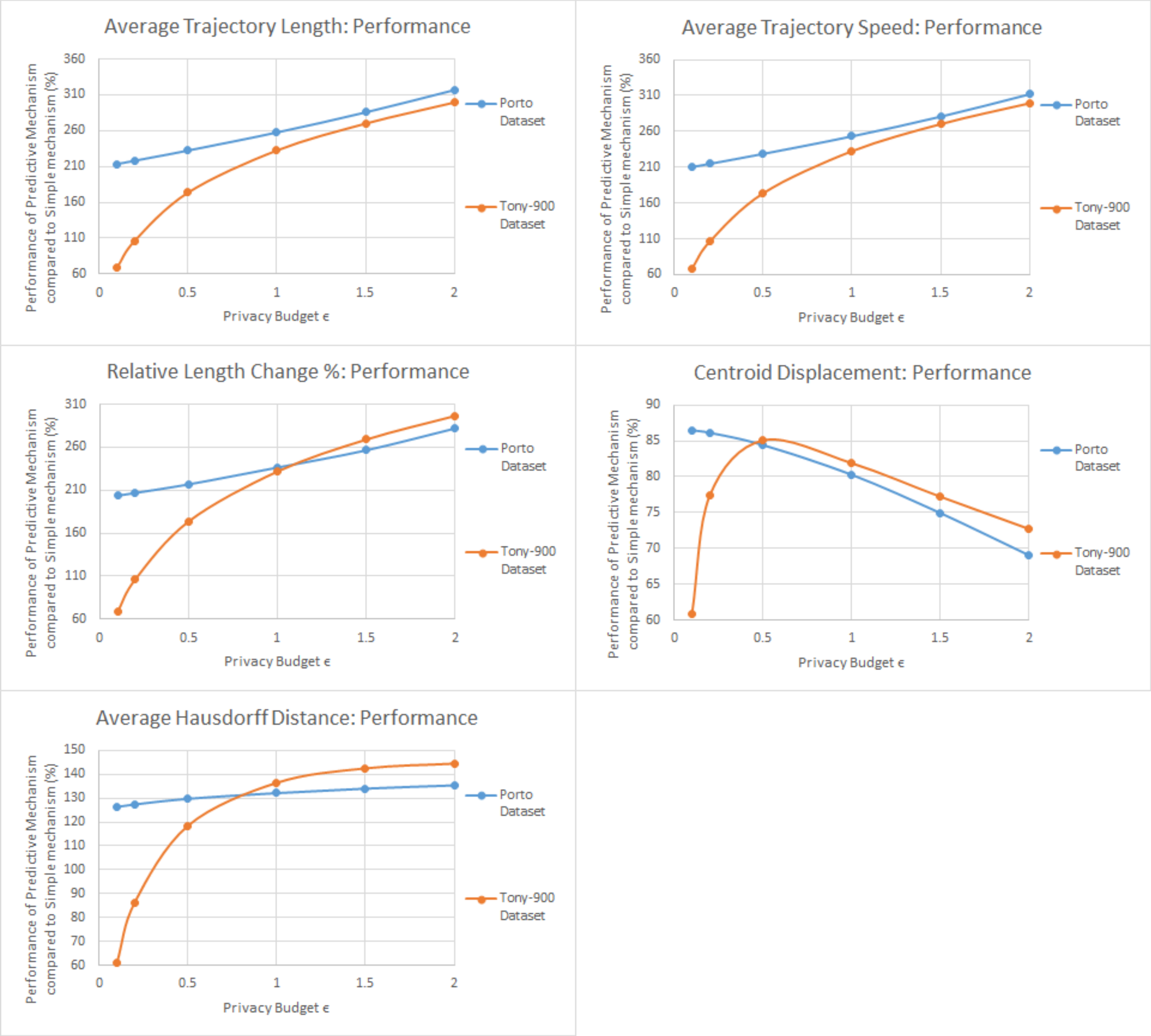


Figure 19: Performance of the predictive differential privacy algorithm compared to the simple one, for all investigated candidate metrics.

5 Conclusion

The aim of this research was to see if a framework of metrics could be devised that allowed for a comparison in between different algorithms. While these algorithms would ideally be picked from different families, for this research a comparison was drawn in between a regular implementation of geo-indistinguishability and a version that uses a budget manager.

After considering a variety of different metrics, the Hausdorff metric has proven itself to be a good indicator of utility loss, in particular from the perspective of a user of a location based service, because it can be taken as an estimate towards the maximum expected error introduced by a PPDP mechanism, where the minimum value of 0 would suggest the full preservation of utility, although if this were to be observed this would mean that there is no privacy preservation at all. While this does not allow for the direct quantification of utility loss, it does allow for a comparison between different algorithms' expected utility preserving performance, where a lower metric measurement indicates better performance.

While the Hausdorff metric could be theoretically used to compare the expected utility preservation of any two anonymizations of the same dataset with no regard towards the algorithms chosen or the privacy parameters selected, the fact that both algorithms used identical privacy parameters allowed for a more thorough comparison through multiple layers of linear regression.

Even though the results indicate that the linear models do not provide a perfectly sufficient explanation for the underlying data, it led to the prediction that using identical privacy parameters, the predictive geo-indistinguishability algorithm performs 34% better on the Porto dataset, and 41% better on the Tony-900 dataset. For the Porto dataset, this estimation does not appear to be too far off from the observed increases in performance when looking at the a-components, that lie between 26% and 35%. For most of the different privacy budget values that were experimented with the performance increase is worse than the predictions obtained from the c-components, and it appears a bias is introduced towards the performances found at a higher privacy budget.

Experiments done with synthetic data reveal something that was observed by the creators of the predictive algorithm, namely that short trajectories lead to a worse performance when using a budget manager, compared to not using a budget manager. Since this effect has only been observed at the lower end of the privacy budget parameters that were investigated, it is likely a combination of the length of the trajectories and the available budget that causes this effect.

All things considered, this research is far from conclusive, but it does present a definitive indication that it is indeed possible to design utility metrics for the comparison of PPDP algorithms' performance through the comparison of their anonymizations.

Despite the criticism provided by Cormode et al. that states most utility metrics do not clearly relate to the actual usage of the data, an argument can be made for the Hausdorff distance as a good indicator of the maximum expected error introduced during the continuous request of a location based service. While this is a good argument from a user-perspective, from a research perspective their point holds, since in this case the actual utility might depend on the nature of the research that is being conducted. Still, in these cases the Hausdorff distance can also be seen as a direct indication towards the maximum amount of perturbation that is on average expected to be present in the trajectories.

6 Discussion and Future Work

Something that was unfortunately not successful was a proper attempt at comparing PPDP algorithms from different families. Instead this thesis only showed a comparison between two types of an algorithm that provides geo-indistinguishability. Because of its relevance as well as the similarities in the mechanism —both can produce a database filled with anonymized trajectories that can be linked to a single original source trajectory— the best candidate for a proper comparison across algorithmic families would be one from the family of algorithms that provide a form of k-anonymity.

Since the length of the trajectories appears to have a big impact on the algorithmic performance, the data cleaning step should have been conducted in a more critical manner. Because every trajectory with multiple GPS locations was treated as a valid trajectory, there are trajectories in the dataset that are too short to represent a realistic taxi ride. The minimum length of a valid trajectory might have been better had it been set around 2 to 3 minutes, instead of a mere 15 seconds.

During the analysis, there has been a strong focus on the calculated averages, with little attention to the rest of the calculated statistics. While the minimum and maximum of the observed measurements could have been explored as a candidate metric as well, an important reason not to consider them was that these values potentially present themselves because of single outliers in a dataset. Additionally, the standard deviation was not explored within the analysis of the results. Incorporating these statistics into a comparison is something left for future exploration.

Even though the use of the c-component is a good attempt towards isolating the performance of the different algorithms from their privacy parameters, the bias introduced in the second layer of regression raises questions towards the validity of this method. Perhaps a different methodology could lead to a more realistic single-value estimation of the performance differences that could be found.

6.1 Layering Linear Regression Analysis

Taking the uncertainty included in the output of a linear regression analyis into consideration has proven difficult. A possible addition that would solidify the so called "c-component" would be to take both the low end and high end values of the 95% certainty range generated by the first linear regression step, and train additional "low" and "high" estimation models for comparison accordingly.

References

[1] *Porto Taxi Dataset*, 2015. `http://www.geolink.pt/ecmlpkdd2015-challenge/dataset.html`.

[2] Miguel E. Andrés, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer Communications Security*, CCS '13, page 901–914, New York, NY, USA, 2013. Association for Computing Machinery.

[3] Daniel Kifer Ashwin Machanavajjhala, Johannes Gehrke and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. volume 1, page 24, 2006.

[4] Vanessa Ayala-Rivera, Patrick Mcdonagh, Thomas Cerqueus, and Liam Murphy. A systematic comparison and evaluation of k-anonymization algorithms for practitioners. *Transactions on Data Privacy*, 7:337–370, 12 2014.

[5] Alastair R Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. *Pervasive Computing and Communications Workshops. Proceedings of the Second IEEE Annual Conference*, page 127–131, 2004.

[6] Kostantinos Chatzikokolakis, Ehab ElSalamouny, Catuscia Palamidessi, and Pazii Anna. Methods for location privacy: A comparative overview. *Found. Trends Priv. Secur.*, 1(4):199–257, December 2017.

[7] Kostas Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. A predictive differentially-private mechanism for mobility traces. 07 2014.

[8] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. Differentially private data generative models. 12 2018.

[9] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. 06 2014.

[10] G. Cormode, C. M. Procopiuc, E. Shen, D. Srivastava, and T. Yu. Empirical privacy and empirical utility of anonymized data. In *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, pages 77–82, 2013.

[11] Kobbi Nissim Cynthia Dwork, Frank McSherry and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography, TCC'06*, page 265–284, 2006.

[12] Wenhao Ding, Wenshuo Wang, and Ding Zhao. A multi-vehicle trajectories generator to simulate vehicle-to-vehicle encountering scenarios. pages 4255–4261, 05 2019.

[13] Thomas Kister Stéphane Bressan Dongxu Shao, Kaifeng Jiang and Kian-Lee Tan. Publishing trajectory with differential privacy: A priori vs. a posteriori sampling mechanisms. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, page 357–365, 2014.

[14] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 486–503, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[15] Ehab ElSalamouny and Sebastien Gambs. Differential privacy models for location-based services. *Trans. Data Privacy*, 9(1):15–48, April 2016.

[16] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[17] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In S. C. Kremer and J. F. Kolen, editors, *A Field Guide to Dynamical Recurrent Neural Networks*. IEEE Press, 2001.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[19] D. Huang, X. Song, Z. Fan, R. Jiang, R. Shibasaki, Y. Zhang, H. Wang, and Y. Kato. A variational autoencoder based generative model of urban human mobility. pages 425–430, 2019.

[20] Kira Kempinska and Roberto Murcio. Modelling urban networks using variational autoencoders. *Applied Network Science*, 4, 12 2019.

[21] David S. Rosenblum Kun Ouyang, Reza Shokri and Wenzhuo Yang. A non-parametric generative model for human trajectories. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 3812–3817, 2018.

[22] Yi Liu, Jialiang Peng, James Yu, and Yi Wu. Ppgan: Privacy-preserving generative adversarial network. 10 2019.

[23] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

[24] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS '07*, page 94–103, 2006.

[25] Benjamin C.M. Fung Noman Mohammed and Mourad Debabbi. Walking in the crowd: anonymizing trajectory data for pattern analysis. In *CIKM '09 Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1441–1444, 2009.

[26] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. The synthetic data vault. pages 399–410, 10 2016.

[27] Julián Salas, David Megias, and Vicenç Torra. *SwapMob: Swapping Trajectories for Mobility Anonymization*, pages 331–346. 2018.

[28] Pierangela Samarati and Latanya Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report.

[29] H Chen X Liu and C Andris. Using generative adversarial networks for geo-privacy protection of trajectory data (vision paper). In *Location Privacy and Security Workshop 2018 in conjunction with GIScience '18*, 2018.

[30] Yu Zheng. Trajectory data mining: An overview. *ACM Transactions on Intelligent Systems and Technology*, 6:1–41, 05 2015.