CHALMERS
UNIVERSITY OF TECHNOLOGY

Applied Differential Privacy in the Smart Grid

Master of science thesis in Computer Systems and Networks

HEDVIG JONSSON
BOEL NELSON

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2015

Master's thesis 2015:Applied Differential Privacy in the Smart Grid

Applied Differential Privacy in the Smart Grid

HEDVIG JONSSON
BOEL NELSON

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Division of Networks and Systems
Chalmers University of Technology
Gothenburg, Sweden, 2015

Applied Differential Privacy in the Smart Grid
HEDVIG JONSSON
BOEL NELSON

Supervisor: Vincenzo Massimiliano Gulisano, Department of Computer Science and Engineering
Examiner: Magnus Almgren, Department of Computer Science and Engineering

Master's Thesis 2015:Applied Differential Privacy in the Smart Grid
Department of Computer Science and Engineering
Division of Networks and Systems
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Printed by Chalmers Reproservice
Gothenburg, Sweden 2015

Applied Differential Privacy in the Smart Grid
HEDVIG JONSSON
BOEL NELSON
Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

Privacy is an important guarantee to give to users in order for them to agree to release their, possibly sensitive, data for scientific or commercial purposes. However, guaranteeing privacy is not a trivial task. Previously there have been several cases where released data was believed to have been anonymized, where it later proved not to be anonymous at all [28, 38]. One methodology to be able to release anonymized calculations is differential privacy, where controlled noise is added to the calculation before it is release. However, there exists a trade-off between the privacy and the accuracy of the results when differential privacy is used. Previous work has mostly focused on differential privacy in theory, but there also exists work that applies differential privacy to a use case [32]. However, the utility of the differentially private results have not previously been evaluated when using only counting queries. In this thesis differential privacy is applied to one use case found in the smart grid, an evolved version of the electricity grid, to show that differential privacy is applicable in practice and not only in theory. The particular use case in this thesis compares a differentially private sum to the true sum, to estimate the error introduced by applying differential privacy. The results demonstrate that differential privacy shows promise also for realistic usage, providing privacy while still producing accurate results compared to the true results without differential privacy applied. For a setup with 1,000 simulated households, the best results for the mean error is between 0.42% and 0.59%, and the spread of the error ranged from 0% to 2.07%. All of these results have a confidence interval of 95%.

Keywords: big data, differential privacy, distributed systems, smart grid.

Acknowledgements

We would like to thank Magnus Almgren and Vincenzo Massimiliano Gulisano for their expert knowledge, guidance and support through this master thesis project.

Hedvig Jonsson and Boel Nelson, Gothenburg, June 2015

Contents

Dictionary

Advanced Metering Infrastructure (AMI) A network of communicating devices that measures some attribute.

Adversary Model A model of an attacker. This includes what knowledge the attacker possesses, and can be used in order to test the security of a system.

Counting Query A query that counts the number of rows that matches some given condition.

Equivalence Class A set of records whose quasi identifiers are indistinguishable from each other. The sensitive attributes do not have to have the same value however.

***k*-anonymity** A syntactic privacy model that ensures the indistinguishably of an individual in a set of $k - 1$ others.

Language Integrated Query (LINQ) A query language which is similar to SQL.

***l*-diversity** A syntactic privacy model where each value for a sensitive attribute must be well-represented, that is, each value must appear l times in an equivalence class.

NumPy A library for Python that includes for example arrays and matrices.

Pandas A library for Python that provides data structures for data analysis.

Privacy-Preserving Data Mining (PPDM) An approach for anonymizing data. It is applied dynamically to data that is being requested from a data set.

Privacy-Preserving Data Publishing (PPDP) An approach for anonymizing data. It is applied statically to data, so that all of it may be published without further modification.

SciPy A scientific library for Python that includes for example mathematical operations.

Smart Grid The upgraded version of the electricity grid where smart devices aid in distributing energy in a more efficient manner.

Syntactic Privacy Model A set of privacy models that uses PPDP. These include k-anonymity, l-diversity and t-closeness.

***t*-closeness** A syntactic privacy model where the distance between the distribution of a sensitive attribute in every equivalence class is less than some threshold t.

True answer Unmodified answer to a query, without differential privacy applied.

Quasi Identifier A set of non-sensitive attributes that by itself is not a unique identifier.

Response Answer to a query with noise added to it, is differentially private.

List of Figures

List of Tables

1
Introduction

Data can be a powerful tool of reference when developing new software services or improving existing ones, as it can be used for statistical purposes. Today, more data than ever is gathered, which results in collections of big data that can be used for developing and improving services. While these sets of data generate new opportunities, it is also important to take into consideration that measures must be taken to protect the anonymity of each data point in the set.

One area where the volume of data is growing is the smart grid. The smart grid is an evolved electricity grid that makes energy distribution more efficient and the grid more resilient to failures. In order to do this, households are equipped with smart metering devices, that measure and communicate the energy consumption to the distributor's servers, in order to bill the consumers accordingly. Data gathered by the smart meters could, apart from being used for billing, also be used in many ways to improve the smart grid. Examples of such improvements include distributing energy using the shortest path to minimize energy loss during transition, detecting energy fraud and predicting energy usage in order to meet the consumption demand.

While releasing energy consumption data without anonymizing it first may seem harmless, there has been research studies that show that sensitive information can be deduced from observing patterns in energy consumption [19, 26, 30]. Note that in order to deduce information from energy consumption, one needs access to high frequency data. Information that can be learned include if the residents of a household eat their breakfast hot or cold, if they got a good night's sleep and if they were home during their sick leave. As Molina-Markham et al. [26] point out, these seemingly harmless facts could actually be of value to third-parties, as they could help regulate insurance rates, resolve legal conflicts, be used for targeted advertising or even be used by criminals to plan burglaries.

Because of the potentially sensitive nature of data gathered, it is important to be able to provide privacy guarantees for individuals. Previous work shows that releasing truly anonymized data, where no data can be traced back to a single individual, is not a trivial task. In the past, anonymizing data has failed several times [28, 38], as what was believed to be anonymized data at the time of release could be traced back to individuals.

In order to release anonymized data, several different approaches have been proposed. Among these are k-anonymity [38], l-diversity [22], t-closeness [21] and dif-

ferential privacy [12]. The approach used in this thesis is the concept of differential privacy, which adds controlled noise to data statistics that are to be released rather than adding it to the existing data set.

Since differential privacy guarantees anonymity of data points when they are used for statistical purposes, it could potentially work as an incentive for individuals to agree to releasing data. Differential privacy has been theoretically established [10, 12, 13, 14, 15], and there also exists some practical implementations [17, 24, 31, 32, 41]. However, none of the practical implementations evaluate the utility of the results when using counting queries in a real setting.

This thesis will apply differently privacy to smart grid data by using a simulated smart grid data set to investigate how differential privacy can be used in practice. In order to test how well applying differential privacy works, a realistic use case will be developed and implemented, to show a real application of differential privacy. The utility of the differentially private result will be investigated by calculating how much noise is added to the true result, which is the result calculated without differential privacy. The amount of noise added depends both on the query type, and on the amount of queries asked. Therefore, querying the data set is not always trivial due to the fact that the added noise can affect the utility of the query result.

The contributions of this thesis are the following.

- Formulate a use case in the setting of the smart grid where differential privacy can be applied.
- Provide a way to query the data set with queries that generate low noise, by creating partitioning strategies for how this should be done.
- Investigate the correlation between the number of queries and the noise added.
- Investigate the correlation between the number of households used and the noise added.
- Give an evaluation of the utility of the differentially private result in the setting of the smart grid.

This thesis is structured as follows. In Section 2, a background of the different approaches used to achieve anonymity is summarized as well as a brief discussion of their flaws. Differential privacy will also be introduced in Section 2. In Section 3, related work will be outlined. Then the use case will be presented in Section 4. Thereafter the method for this thesis will be described in Section 5. The implementation of the use case will be shown in Section 6. In Section 7, the simulation of the smart grid data used in the implementation will be presented. In Section 8, the results will be shown. Then, in Section 9, a discussion of the results and the method used will be held and lastly in Section 10 the thesis will be concluded.

2
Background

To understand what type of privacy model is suitable for this thesis a short introduction of different privacy models is given. After that the concept of the smart grid will be introduced, to provide an overview of the basic layout.

2.1 Privacy

Previously, work has been done to preserve privacy and to expose leaks when querying databases [6, 7, 8]. These days there exist several different privacy models that can be applied to sets of data in order to achieve certain privacy guarantees. One group of privacy models is the syntactic privacy model where k-anonymity, l-diversity and t-closeness are included [3]. Differential privacy [13], on the other hand, does not belong to this group because it handles data in a fundamentally different way. It is important to choose a privacy model that can protect against the adversary model that is assumed in order to provide adequate privacy guarantees. Furthermore, it is interesting to explore the fundamental difference in handling data, and to get to know the different privacy models' strengths and weaknesses, since it aids in making an educated choice of privacy model. The choice of privacy model for this thesis will be motivated in Section 5.2.

2.1.1 Syntactic Privacy Models

The syntactic privacy models [3] use privacy-preserving data publishing, also known as PPDP. This type of models aim to publish already anonymized data. Because the published data is anonymized, it can be analyzed in any way possible without impact on privacy. PPDP does not assume anything about the queries used, or what kind of analysis that can performed on the data [3]. This means that the anonymization performed on the data set is independent of the query types used.

All models in this section originates from k-anonymity [34]. k-anonymity was presented by Samarati and Sweeney [34] as a technique to protect data. For example, Sweeney noticed that it was possible to re-identify individuals by linking attributes from one data set to another, even when data had supposedly been anonomyized [38]. The data sets Sweeney linked were medical data gathered and released by the Group Insurance Commission (GIC) and a voter registration list for Cambridge Massachusetts. With these two data sets, Sweeney could pinpoint the governor of Massachusetts at the time when data was gathered, by linking his ZIP code, birth

date and gender between the two sets of data. Later, syntactic privacy models providing stronger privacy guarantees than k-anonymity were invented [21, 22].

2.1.1.1 *k*-Anonymity

k-anonymity is a formal protection model for releasing anonymized data introduced by Samarati and Sweeney [34]. The model states that if an individual in a set of k individuals cannot be distinguished from at least $k-1$ others in a set of data, the data set has k-anonymity.

A central concept in k-anonymity is quasi-identifiers. These are the sets of attributes that identifies an individual in a data set, like the example in 2.1.1 [38]. However, a quasi-identifier does not have to uniquely identify an individual. A quasi-identifier is formally defined as follows [38].

> **Definition 1.** *Given a population of entities U, an entity-specific table $T(A_1, ..., A_n)$, $f_c : U \to T$ and $f_g : T \to U'$, where $U \subseteq U'$. A quasi-identifier of T, written Q_T , is a set of attributes $A_i, ..., A_j \subseteq A_1, ..., A_n$ where: $\exists p_i \in U$ such that $f_g(f_c(p_i)[Q_T]) = p_i$.*

To prevent re-identification by linking, Sweeney [38] defined k-anonymity as follows.

> **Definition 2.** *Let $RT(A_1, .., A_n)$ be a table and QI_{RT} be the quasi-identifier associated with it. RT is said to satisfy k-anonymity if and only if each sequence of values in $RT[QI_{RT}]$ appears with at least k occurrences in $RT[QI_{RT}]$.*

That is, a data set satisfies k-anonymity if and only if each quasi-identifier appears at least k times in the set. Table 2.1 shows an example of a data set where 3-anonymity is achieved.

	Zip Code	Birth Date	Gender	Medical Condition
1	a	b	c	x
2	a	b	c	y
3	a	b	c	z
4	d	e	f	y
5	d	e	f	y
6	d	e	f	z

Table 2.1: A table with 3-anonymity, where zip code, birth date and gender are considered non-sensitive attributes and medical condition is a sensitive attribute. Thus at least 3 entries in the table share the same quasi-identifier. In this table, row 1-3 and 4-6 share the same quasi-identifiers.

2.1.1.2 Attacks Against *k*-Anonymity

Machanavajjhala et al. [22] used two attacks to show that k-anonymity has severe privacy problems. The first attack, the homogeneity attack, showed that sets with little diversity in sensitive attributes allowed an adversary to figure out the value of the attributes. With the second attack, the background knowledge attack, Machanavajjhala et al. showed that k-anonymity does not guarantee privacy when the adversary has background knowledge.

The homogeneity attack arises due to the fact that a data set can contain groups of individuals with the same quasi-identifiers that have the same value for their sensitive attributes. For example, consider medical condition as a sensitive attribute. If a group of k individuals has the same quasi-identifiers, but also share the same medical condition, the medical condition is leaked even when k-anonymity applies to this group. An example of a situation in which this problem would arise is displayed in Table 2.2. This scenario might seem uncommon, but according to Machanavajjhala et al. it is in fact not.

	Zip Code	Age	Nationality	Medical Condition
1	a	b	c	x
2	a	b	c	x
3	a	b	c	x
4	a	b	c	x
5	d	e	f	x
6	d	e	f	y
7	d	e	f	z
8	d	e	f	x

Table 2.2: A table with 4-anonymity, where zip code, age and nationality are considered non-sensitive attributes and medical condition is a sensitive attribute. However, if the adversary knows that the zip code, age and nationality of some individual X, is a, b and c, X must be on either row 1, 2, 3 or 4. Since all entries within the tables with this quasi-identifier have the same condition, the adversary can conclude that X have the condition x.

For the second attack to be successful the adversary must have access to some background knowledge, hence the name background knowledge attack. While an individual will share its quasi-identifies with k others individuals in the data set, by having access to other statistics the adversary can figure out the value of a sensitive attribute with near certainty. For example, if the adversary discovers that an individual, X, has either medical condition x or y, he or she can take advantage of background knowledge. Perhaps the adversary knows the nationality of X, and also knows that y is a more common medical condition in the country from which X comes from. Then the adversary is able to conclude that it is more likely that X has condition y than x. An example of this scenario is shown in Table 2.3, where X's data is either on the first or second row.

	Zip Code	Age	Nationality	Medical Condition
1	a	b	c	x
2	a	b	c	y
3	d	e	f	x
4	d	e	f	y

Table 2.3: A table with 2-anonymity, where zip code, age and nationality are considered non-sensitive attributes and medical condition is a sensitive attribute. However, if the adversary knows that the zip code, age and nationality of some individual X, is a, b and c, X must be either row 1 or 2. If the adversary also knows that y is a more common condition than x in the country from which X is from, then the adversary can conclude both that X has x or y, but also that it is more likely that X has medical condition y than x.

Another background knowledge attack was conducted in 2008 by Narayanan and Shmatikov [28]. They showed, by using real data, that k-anonymity does not provide any meaningful privacy guarantees when the adversary has only a little background knowledge. Furthermore, k-anonymity fails on high-dimensional data [1]. This is due to the fact that the number of possible quasi-identifiers increases when large data sets are anonymized, which leads to a removal of data points that result in a loss of information that is too high. The attack was conducted by linking users from a data set released by Netflix, an online movie rental service, to users from the Internet Movie Database (IMDb), which acted as their background knowledge. This made it possible to identify if a certain individual's entry was in the data set, or at least pinpoint it to being within a subset of entries.

Another interesting finding by Narayanan et al. was that the presence of non-null columns can release data. The authors point out that in most cases, as much information is released from knowing which columns are non-null as knowing the value of a column. For example, simply having rated a movie would make the corresponding column non-null, and thus reveal that the user has, with high certainty, seen the movie.

2.1.1.3 *l*-Diversity

Succeeding k-anonymity, l-diversity was introduced by Machanavajjhala et al. [22] as a stronger notion of privacy. It protects against identity disclosure, as well as attribute disclosure in its stated adversary model. In order to define l-diversity, Machanavajjhala et al. first defined a q^*-block as follows.

Definition 3. *A q^*-block is the set of tuples in a table, T^*, whose values of the nonsensitive attributes generalize to q^*.*

In other words, each q^*-block is a block of data points that have the same values for their quasi-identifiers. Machanavajjhala et al. then proceeds to define the l-diversity principle as follows [22].

> **Definition 4.** *A q^*-block is l-diverse if it contains at least l well-represented values for the sensitive attribute S. A table is l-diverse if every q^*-block is l-diverse.*

Where well-represented values are values that have roughly the same frequency. An example of a 3-diverse table is shown in Table 2.4.

	Zip Code	Age	Nationality	Medical Condition
1	a	b	c	x
2	a	b	c	y
3	a	b	c	z
4	d	e	f	x
5	d	e	f	y
6	d	e	f	z
7	g	h	i	x
8	g	h	i	y
9	g	h	i	z

Table 2.4: A 3-diverse table, where zip code, age and nationality are considered non-sensitive attributes and medical condition is a sensitive attribute. Note that each medical condition appears the same amount of times in each q^*-block, and thus is well-represented.

2.1.1.4 Attacks Against *l*-Diversity

Even though l-diversity is an attempt to go beyond k-anonymity it has some issues in itself. Li et al. discuss some of these in their paper *t-Closeness: Privacy Beyond k-Anonymity and l-Diversity* [21]. It has been discovered that l-diversity does not prevent attribute disclosure in a sufficient way. Li et al. demonstrate this by describing two types of attacks; the skewness attack and the similarity attack.

The authors point out that if the distribution is skewed then l-diversity can not prevent attribute disclosure. The reason for why attribute disclosure cannot be prevented, is that the distribution for the data set in that case is distinctively different from that of the distribution for the real population. Consider a scenario where the overall population has a 1% chance of some sensitive attribute being positive, while any individual in the equivalence class has a 50% chance of the sensitive attribute being positive. In this scenario a privacy risk arises due to the skewed distribution, and thus l-diversity does not prevent the disclosure of attributes.

The second attack, the similarity attack is similar to the homogeneity attack described in Section 2.1.1.2. In the similarity attack, unlike the homogeneity attack,

the adversary guesses one sensitive attribute in order to discover a second sensitive attribute. Ponder the scenario where the adversary knows that the value for a sensitive attribute, x must be in a specific range for a targeted individual. Then consider all individuals with values for x that are within this range a group. If the individuals in this group have similar values for another sensitive attribute, y, the adversary can conclude that the targeted individual has a certain y value too.

Lastly, Li et al. [21] also mention that it might be difficult to achieve l-diversity and in some cases even unnecessary. Consider the case where there is only one attribute that is sensitive, and only having a positive value for the attribute is considered sensitive information. Li et al. states that for such a case 2-diversity is completely unnecessary for an equivalence class containing only negative entries. Furthermore, if the amount of entries is large, for example 10 000 records, where only 1% of all entries are positive, there can be at most 100 equivalence classes in order to achieve 2-diversity. This would result in an information loss, which is another reason for why applying l-diversity is both difficult and sometimes unnecessary.

2.1.1.5 *t*-Closeness

t-closeness is a privacy notion proposed by Li et al. [21]. To define t-closeness, equivalence classes must first be explained. These are sets of records that are indistinguishable from each other, in other words, an equivalence class is a set of data that have the same values for their quasi-identifiers. This means that an equivalence class is the same as a q^*-block that was introduced in 2.1.1.3. Table 2.5 provides an example of an equivalence class.

	Zip Code	Birth Date	Gender	Medical Condition
1	a	b	c	x
2	a	b	c	y
3	a	b	c	z

Table 2.5: In this table zip code, birth date and gender is a quasi-identifier. All entries are within the same equivalence class, since they have the same values for their attributes that make up the quasi-identifier.

In t-closeness, the distribution of a sensitive attribute in any equivalence class must not have a greater distance than t to the distribution of the attribute in the overall table. This means that the distribution of the sensitive attribute in an equivalence class is similar to that of the entire table. The formal definition of t-closeness given

by Li et al. is given next.

> **Definition 5.** *An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness.*

2.1.1.6 Weakness of Syntactic Privacy Models

As discussed in the previous section, k-anonymity and l-diversity both have issues. t-closeness is an attempt to fix these. However, t-closeness share the same basic issues as both k-anonymity and l-diversity due to it also being a syntactic privacy model. The syntactic privacy model's greatest issue is that it assumes what kind of background knowledge the adversary has [9]. Therefore they are all defenseless against and adversary that knows more than was assumed by the privacy model. The syntactic privacy models have also shown to have utility issues for large data sets [1, 9].

2.1.2 Differential Privacy

Differential privacy differs from the syntactic privacy models because it uses privacy-preserving data mining (PPDM) instead of PPDP. In differential privacy the entire data set will not be published directly as in PPDP; data will instead be anonymized as it is requested. Answers to such requests, or queries, will not disclosure any sensitive information about individual data points, thus guaranteeing privacy. This is done by adding noise to data statistics before it is released, rather than directly to the data set as in PPDP.

More formally, differential privacy is a privacy guarantee that ensures that the result of an analysis does not change notably if one single item is added or removed from the data set [13]. In simpler terms, differential privacy is a concept that ensures that the output distribution does not alter much based on one data point. Therefore it imposes no risk for an individual to join a data set, since it will not notably affect the outcome of any calculations performed on the data set.

The formal definition of ϵ-differential privacy given by Dwork [12] is as follows.

> **Definition 6.** *A randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*
>
> $$Pr[\mathcal{K}(D_1) \in S] \leq exp() \times Pr[\mathcal{K}(D_2) \in S]$$

2.1.2.1 Privacy Mechanism

A key aspect in differential privacy is that no changes are made to the data set. Instead, controlled noise is added to queries as data is released. This release of data is referred to as the privacy mechanism. The privacy mechanism adds random noise to the result of every query, $f(X)$, where f is the query function and X is the data set. Dwork [13] writes that the privacy mechanism $\mathcal{K}$ for a function f that consists of k components, produces an answer using the following definition.

Definition 7.

$$f(X) + (Lap(\Delta f/\epsilon))^k$$

That is, each query will have added noise with Lap($\Delta f/\epsilon$) distribution.

One important thing to note is that the privacy mechanism is independent of the size of the data set, unlike all PPDP models. This is because it only depends on ϵ and the sensitivity of the function. Because of this, differential privacy can be applied to big data, seeing as the noise introduced by the privacy mechanism only causes relatively small errors according to Dwork [13]. Note however, that since the noise introduced is based on the sensitivity of the query, it is still important to choose a query with low sensitivity in order for the noise to be small.

2.1.2.2 Sensitivity Function

The amount of noise added by the privacy mechanism is chosen as a function of the largest change a data point could have on the output of the query function. This is the L_1-sensitivity of the function. It is also defined by Dwork [12] as.

Definition 8. *For $f : \mathcal{D} \rightarrow R^d$, the L_1-sensitivity of f is*

$$\Delta f = \max_{D_1, D_2} ||f(D_1) - f(D_2)||_1$$

for all D_1, D_2 differing in at most one element.

2.1.2.3 Known Challenges with Differential Privacy

There are some known challenges with differential privacy. One is to choose a suitable value for the privacy parameter ϵ, which is not a trivial task. Lee and Clifton [20] point out that ϵ has to be chosen carefully depending both on the domain differential privacy is applied to, and the type of query that should be supported. They, Lee and Clifton, write that if a low value for ϵ is chosen, the privacy risk is low. Furthermore, Lee and Clifton point out that a low value for ϵ leads to a low utility of the result, since pure random noise is added to the result. Therefore choosing a suitable value for ϵ poses a trade-off between privacy and utility [20].

Another known challenge, which was noted by Clifton and Tassa [3], is how to calculate the global sensitivity. The reason for why calculating the global sensitivity

is hard, is because it requires one to know the maximum difference of any two possible values within the domain. If these values are not chosen to be realistic they will generate excessive noise, which lowers the utility of the result.

2.2 The Smart Grid

In recent years the traditional electricity grid has evolved into an interactive and dynamic grid, known as the smart grid. Among the causes for this evolution is the need to support real time measurement of electricity consumption in order to make the grid more resilient as well as making it possible to forecast energy usage [27]. Therefore the smart grid provides a more efficient way to distribute electricity.

Part of the smart grid is the Advanced Metering Infrastructure, or AMI for short, which is a composition of networks of communicating devices. The AMI consists of several parts. The first component is a metering device, called a smart meter. It is an electronic device placed at the consumer end points in the AMI, that among other information sends consumption data, in the form of tuples, at certain intervals. These tuples normally contain at least the current electricity consumption reading, a timestamp and the id of the smart meter. Due to EU recommendations [4], each smart meter should send consumption data at least every 15 minutes. However, it is possible to send consumption data at more frequent intervals.

The second part is a communication network to send the data over. Like any network, the network in the smart grid is unreliable. This means that messages can be lost before they reach the receiver, or arrive out of order. Because of this there exists a need to validate tuples sent.

Lastly, there exists a service provider that provides the electricity. In order for the service provider to distribute the electricity a utility is used that is connected to several smart meters. The service provider also keeps track of the billing and accounts for the distribution of electricity.

3
Related Work

Several authors have conducted research concerning privacy that is relevant to this thesis. In Section 2.1 the problem of privacy in general was discussed, and differential privacy was compared to syntactic privacy models. In this chapter, the focus will be on more closely related work. Other authors have previously identified and addressed the privacy concerns in the smart grid. These studies are presented in Section 3.1. There have also been studies that apply differential privacy to specific settings, similar to this thesis, which are presented in Section 3.2. Furthermore, Section 3.3 presents work that focuses on the challenges with applying differential privacy in software, and presents some notable examples of where this has been attempted.

3.1 Privacy Concerns in the Smart Grid

The smart grid has privacy issues that are not newly discovered. However, there is still no general solution as to how these issues should be addressed properly, and what techniques should be used. Therefore several researchers have made efforts to identify the privacy issues, and in some cases also suggests solutions to these.

Much can be learnt about an individual's living patterns by just observing their energy consumption. This has been shown by, for example, Patel et al. [30]. Furthermore, Lam et al. [19] constructed a taxonomy for mapping load signatures to electrical appliances. Due to the fact that detailed information about what an individual is doing can be deduced by observing the energy consumption alone, this information would be a clear privacy violation.

Molina-Markham et al. [26] proposed a way to preserve the consumers' privacy in the smart grid, by limiting the information sent by the smart meters. However, since the producers still need to verify how much each household has consumed in order to bill the consumer accordingly, data still has to be sent. Molina-Markham et al. suggested a zero-knowledge proof in order for the producer to verify the consumption by a consumer, without revealing when and how this energy was consumed.

In a paper from 2012, Siddiqui et al. [36] discuss known privacy issues in the smart grid. Siddiqui et al. mention that smart meters may reveal information about consumer activities, and that roaming smart grid devices would possibly generate even more personal information. Another issue is that there are several different standards

and privacy policies involved in collecting information in the smart grid. Furthermore, the definitions of what is considered personal identifiable information are not consistent in the industry. Lastly, Siddiqui et al. write that the industry does not really have a clear picture of what the privacy issues are in the smart grid, which is worrisome as that makes the problems hard to address. A similar remark was made by McDaniel and McLaughlin [23] in 2009, when they wrote that the privacy issues in the smart grid needs to be further investigated.

All of the related work presented in this section point out potential privacy problems, and in some cases solutions to these. The work in this thesis is not trying to find more potential privacy problems, but rather provide a new solution to guaranteeing privacy for released data from the smart grid.

3.2 Differential Privacy in Specific Settings

In this thesis differential privacy is applied in a specific setting. However, there exists previous work that also apply differential privacy to specific settings, but they do not adapt the query type used in order to lower the error introduced by differential privacy.

Ács and Castelluccia [42] applied differential privacy to smart meters in a way that is similar to this thesis. They calculate the sum of energy consumption by letting each smart meter apply differential privacy to their consumption before it is encrypted and sent to an aggregator. However, their work differs in the sense that they do not use counting queries in order to approximate the total energy consumption of all the smart meters. Furthermore, they are concerned with the security and safety of the smart meters, and their communication in the network, which is not in the scope of this thesis.

Danezis et al. [5] propose a way to apply differential privacy to billing systems, in order to anonymize money transactions. Their work differs from this thesis since they apply differential privacy to already existing queries, rather than constructing a way to adapt counting queries to the use case, as is done in this thesis.

3.3 Other Applications of Differential Privacy

Prior to this thesis other efforts to apply differential privacy to practical applications have been made. These will be presented briefly in this section, as they provide insights to what the potential challenges when applying differential privacy might be. Previous work also give a perspective on applying differential privacy in software. The works presented differ from the work conducted in this thesis in the sense that none of them actually show how the accuracy of the results is affected by the number of queries asked and the amount of data points in the data set when their implementations are applied to a real use case, as is done in this thesis.

3.3.1 Privacy Integrated Queries (PINQ)

Privacy Integrated Queries [24], or (PINQ), is a platform for privacy-preserving data analysis. PINQ does not execute any queries itself, but merely applies differential privacy to the result of a query written in language integrated query (LINQ) which is similar to SQL. Hence, PINQ acts as a protective layer for data sources rather than a query language.

However, Lee and Clifton [20] remark that the users have to choose ϵ themselves in PINQ. This means that while PINQ provides a way to apply differential privacy to queries, it requires expert knowledge from the user, since choosing an appropriate value for ϵ is not trivial. In this thesis, however, ϵ will be set to a static value, therefore this issue does not apply in the case of this thesis.

3.3.2 Provenance for Personalised Differential Privacy (ProPer)

Provenance for Personalised Differential Privacy, or ProPer [17], is a system similar to PINQ. Just like PINQ, ProPer is based on LINQ. The main difference is that ProPer, unlike PINQ, has personalized privacy budgets instead of one global budget. Due to the personalized budgets, the system also accounts for databases that expand dynamically.

3.3.3 Streaming PINQ

Streaming PINQ is an extension to PINQ introduced by Waye [41]. Unlike PINQ this framework focuses on applying differential privacy to data streams. It has an extendable interface that makes it possible to provide access to any kind of data stream. The framework also lets the user decide on the trade-off between privacy and utility of the results.

The work done by Waye stands out is in the sense that the data set is dynamic. While Waye provides a general framework intended to work for any case, the work in this thesis focuses on applying differential privacy to one real use case.

3.3.4 Airavat

Another system that provides security guarantees by applying differential privacy is Airavat which was developed by Roy et al. [32]. Airavat is based on MapReduce and combines mandatory access control and differential privacy to achieve both strong privacy and security guarantees. The prototype developed by Roy et al. uses Hadoop in its implementation in order to support distributed computations.

As observed by Lee and Clifton [20], Airavat does not pick a suitable value for ϵ, but rather leaves the choice up to the user. Because of this, Airavat just like PINQ, requires that the user possess expert knowledge in order to use the system properly.

This thesis chooses a static value for ϵ, and thus the impact of this decision will not be the focus of this thesis.

3.3.5 Fuzz

A different approach to implementing differential privacy was taken by Reed and Pierce. They constructed a functional programming language, called Fuzz, with a type system that guarantees differential privacy [31]. Because of this, any program written in Fuzz will also be differentially private. The language manages to capture the sensitivity of functions by also using a distance-aware type system.

4

Use Case

In order to implement differential privacy it needs to be applied to a use case. Therefore, a couple of use cases is introduced in the setting of the smart grid in this section. The scenario that is described in detail in Section 4.1 is the main use case in this thesis. Furthermore, the implementation of this use case is explained in detail in Section 6. Two other use cases will also be introduced, but these will not be implemented. However, they will be discussed since it is interesting to identify several uses for differential privacy in the smart grid.

4.1 Scheduling Use Case

Imagine a block of houses. All of these consume a certain amount of energy that the producer wants to supply in an efficient manner; that is without overproducing or delivering too little energy. In order to have an efficient supply the producer can thus gather and process information on previous consumption to predict future energy consumption.

The problem with gathering data for these statistics is the privacy concerns that arises. While the consumers might agree to release this data, the consumers probably do not want to say when they need the energy, as was discussed in Section 3.1. Under these circumstances differential privacy is a solution that provides the necessary privacy guarantees, even when fine-grained consumption data is gathered.

In this scenario the producer wants to calculate what the sum of the energy consumption is for a block of houses, in order to create statistics that can be used to predict future consumption. To be able to see differences in the consumption, the consumption needs to be fine-grained. It is not enough to be able to predict how much a household consumes on a monthly basis, since that does not give any useful information about when the energy should be delivered. Therefore, the data has to be gathered at shorter intervals. These intervals must be able to capture peaks in energy consumption, to make sure enough energy is provided at all times. The challenge however, is that the predicted energy consumption cannot be too optimistic or else consumers will experience blackouts, but it cannot be too pessimistic either, since this would result in requiring higher capacity equipment than needed. Because of this the intervals have to be carefully chosen to be close to the true consumption. To choose such intervals precisely, in-depth knowledge about energy consumption and power grids is required.

4.2 Fraud Detection

Another use case is a fraud detection scenario. Imagine that a consumer tries to convince the producer that he or she has consumed less energy than in reality by sending faulty data to the producer. This would lead to the consumer getting an incorrect bill. Even worse, the consumer could send faulty data claiming to have consumed a negative amount of energy, in other words having produced energy, which would mean that the consumer should get paid rather than pay for the energy. While applying differential privacy to this use case does not remove the need to further investigate a suspect, it can potentially preserve the privacy of innocent consumers.

4.3 Detecting Illegal Activity

The last use case suitable for differential privacy and energy consumption readings is detecting illegal activity. This is use case arises when a customer consumes a suspicious amount of energy. Suspicious amounts can be detected by noticing specific patterns, such as an abnormally high energy consumption over a certain time span. This might indicate that the consumer is conducting illegal activities, such as growing marijuana or hosting illegal file servers. Such information could be of use for authorities, for example the police.

As for the previous use case, concerning fraud detection, suspicious activity would have to be investigated further without differential privacy applied, but normal activity would be anonymized. This means that consumers that does not have suspicious patterns in their energy consumption would still get their privacy preserved.

5
Method

In order to apply differential privacy to smart grid data, decisions about how this should be done must first be motivated. This section will present those decisions, as well as argue for why these decisions were made.

5.1 Adversary Model

In this thesis, the adversary is assumed to be strong. This means that the adversary may have access to arbitrary background knowledge. Still, the privacy model must be able to protect against such an adversary. The model must even be able to protect when the adversary has a complete overview of all except one individual, X, in the smart grid. It must then be improbable for the adversary to learn anything new about X from just having access to a result, otherwise it is considered a privacy breach.

An example of how the adversary could use this knowledge is if the adversary receives a sum of everyone's energy consumption. By knowing everyone elses consumption, it would then be possible for the adversary to deduce X's energy consumption. This is the most extreme case, but it should still be prevented.

5.2 Choice of Privacy Model

Since the adversary model is assumed to be strong the syntactic privacy models do not provide sufficient privacy due to the potential auxiliary knowledge the adversary possesses. Attacks against the syntactic privacy models have previously been explained in Section 2. Because these models only protect against an adversary with limited knowledge, they will not be used in this thesis.

Differential privacy, however, does not have the same limitations, but is challenging to implement correctly. It also requires the implementer to maintain an adequate trade-off between utility and privacy of the result, but it can handle a very strong adversary model. Therefore differential privacy is chosen as the privacy model in this thesis.

5.3 Local versus Global Sensitivity

Since ϵ-differential privacy depends on the L_1-sensitivity of the function used, it is important to decide whether to use local or global sensitivity before applying ϵ-differential privacy. Local sensitivity is a derivative-based approach, and therefore has a very efficient computation time. This is due to the fact that the local sensitivity is the maximum difference between any two data points in a data set, which means that it is always possible to find the answer. However, the local sensitivity is only useful for linear systems, since it fails to explore the rest of the space except for the base point [33]. This is due to the fact that the approach is derivative-based, and thus it does not work well when the input and the linearity of the system is unknown [33].

Because the energy consumption of a household is not considered to be a linear system, this thesis will use global sensitivity. However, global sensitivity can suffer from large values. Large sensitivity will cause problems since the noise added from the privacy mechanism in ϵ-differential privacy depends on the L_1-sensitivity. If the sensitivity is too large, it will ultimately give a result that cannot be used since it will have introduced too much noise.

The sensitivity of, for example, the energy consumption in kWh per day for a household is too large if the data should be used to predict the future energy consumption. This is because the sensitivity of such a query would be the maximum amount of energy that a household could consume within a day, which is only constrained by the actual physical limit of the fuse.

In order to prevent the sensitivity of the function from being too large the authors propose the opposite approach. Rather than controlling the sensitivity of a function, a function with low sensitivity will be adapted to fit the use case. A function that is a natural choice, since it has a low sensitivity, is the counting query as explained in Section 5.4.

5.4 Constructing Queries

An important aspect of ϵ-differential privacy is the L_1-sensitivity of a function. To avoid adding too much noise to a query using the privacy mechanism, the query should have low L_1-sensitivity. If the sensitivity is too large, the added noise will make the utility of the result lower, since it will no longer be accurate enough given the scenario where it is to be used. Therefore it is important to choose a query that does not add too much noise, in order to have adequate accuracy.

The second aspect to be aware of is the amount of queries asked. If the same query is repeated several times, the same amount of noise can be added to each query, thus always yielding the same result [15]. However, if new, overlapping, queries are used, the amount of noise added has to increase in order to preserve privacy.

This means that the more different queries that are asked, the more noise is added to the true answer. Also note that asking several non-overlapping queries does not add more noise than asking these queries on completely disjoint data sets would [16].

As more noise is added to the true answer, the utility of the response decreases. Eventually the response will be indistinguishable from pure random noise, which means the result will have low utility. Therefore it is important to query the data set in an intelligent way, to make sure as little noise as possible is added to the response, thus resulting in high accuracy. Depending on the scenario differential privacy is applied to, different approaches for constructing queries are possible.

5.4.1 Counting Queries

A counting query asks how many times a specific data point occurs in a certain data set; for example "How many occurrences of data type y is there in data set x". Counting queries have low sensitivity, which means they produce small noise when differential privacy is added. This is because the sensitivity of a counting query is always 1. Dwork [11] states that such queries are very powerful when it comes to privacy-preservation. Therefore this is a suitable query to use when applying differential privacy, seeing as the privacy mechanism depends on the sensitivity of the query.

5.4.2 Translating the Use Case into Queries

In order to implement the scheduling use case presented in Section 4 the query type that should be used has to be chosen first. A naive solution would be to use the query "How many kWh has been consumed?", but such a query would have very high sensitivity. Therefore the answer to the query, "How many kWh has been consumed?", has to be found by using other queries with lower sensitivity.

To translate the query "How many kWh has been consumed?", several counting queries will be used, since they have low sensitivity. Each counting query will ask how many households consumed energy within a certain interval, rather than how much they consumed. Such a counting query will be on the form "How many data points have a consumption between y and z kWh?". Since this approach requires several counting queries to be asked once per interval, the entire range of possible consumptions must first be divided into those intervals. These intervals can be constructed in different ways, in this thesis those ways are referred to as partitioning strategies, and examples of them are presented in Section 6.2.2. Since there are several ways to query the distribution, a number of partitioning strategies are constructed to see how the result differs.

The whole distribution range is partitioned into different amount of intervals, where each interval is queried on the form "How many data points have a consumption between y and z kWh?". These queries will be referred to as bins for simplicity from

now on. An example of such a bin is illustrated in Figure 5.1, where a bin that covers the interval from y to z is shown with its corresponding counting query.

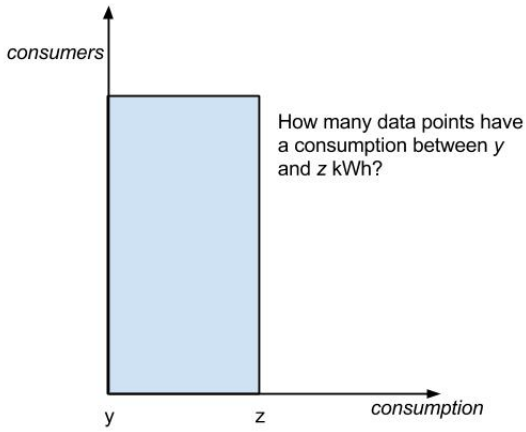


Figure 5.1: An illustration of a bin, which represents one counting query. This particular bin covers the interval y to z and has its corresponding counting query written next to it.

Each query will return how many data points is within a given interval. The answer is then multiplied with the middle value of the bin. This is illustrated in Figure 5.2.

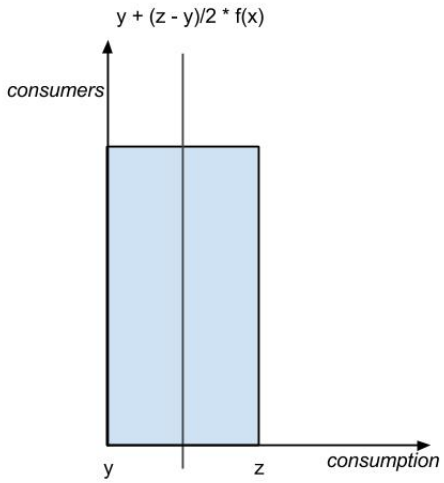


Figure 5.2: The line in the middle of the bin represents the middle value for the interval y to z. This middle value is then multiplied with the answer to the query, f(x), which represents the query "How many data points have a consumption between y and z kWh?". The value that is achieved by doing this is the amount of energy consumed by all households in the interval y to z.

Lastly all the answers, from each counting query, are summed up to give the answer to the original query "How many kWh has been consumed?". An illustration of this

is shown in Figure 5.3.

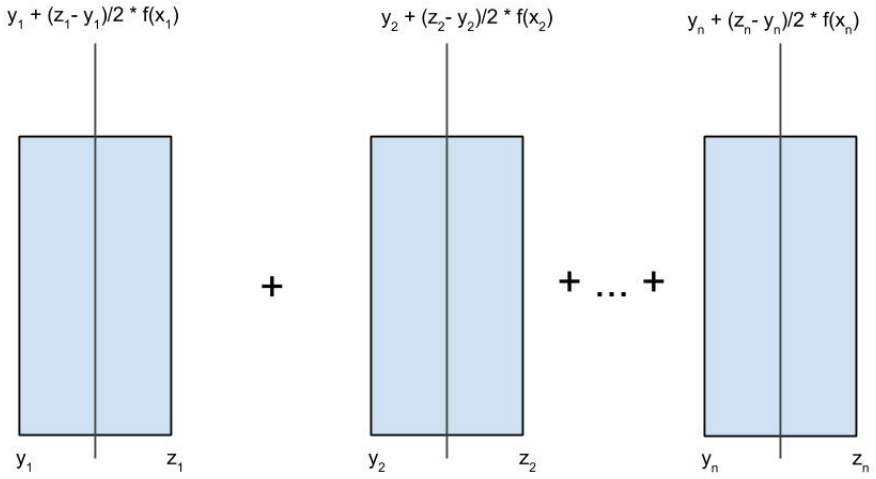


Figure 5.3: Several counting queries, each represented by a bin, are multiplied with the middle value of their interval before they are summed up. Note that this sum corresponds to the query "How many kWh has been consumed?".

5.5 Method Evaluation

The general method for processing data used in this thesis is the following; first raw data, that is simulated, is used as input for the use case explained in Section 4. From this use case, a simulation will be run to produce both a differentially private and a true result, this is the implementation phase which is further presented in Section 6. The differentially private result produced by the use case will then be compared to the true result to determine how much they differ. This means that the final result produced by the method corresponds to the error between using the true sum and the differentially private sum. An illustration of the general process can be seen in Figure 5.4.

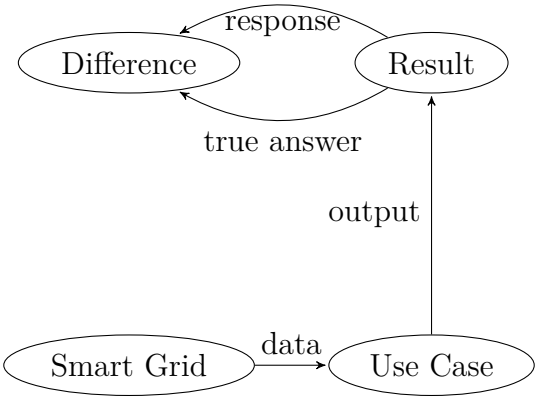


Figure 5.4: How data flows from the smart grid until it is compiled into a result

5.6 Programming Language

The language needed to implement the method presented in this chapter has to have support for mathematical operations and it also has to be able to simulate a normal distribution. Since there will be a lot of data produced, the language has to be able to handle large volumes of data. Furthermore, since the range of possible energy consumptions should be divided into bins, it would be helpful if the language provided enough programmable strength to implement this.

Because of these requirements, the partitioning strategies will be implemented using IPython Notebook [18] which is an interactive environment for Python. To simulate a normal distribution as well as perform mathematical computations the libraries NumPy [29] and SciPy [35] will be used. Furthermore, to handle the volumes of data, data structures from the Pandas [39] library will be used. The reason for why Python was chosen for the realization is because it supports mathematical computations, is able to simulate different distributions and it also provides the programmable strength needed to implement the different partitioning strategies.

6
Implementation

Four main topics will be explored in this section. The first area, Section 6.1, concerns setting up a correct environment for development. This will be followed by a detailed explanation of the general design, which includes presenting three different querying strategies in Section 6.2.2. Lastly, the implementation of the use case from Section 4 is given in Section 6.2.3.

6.1 Assumptions

To implement differential privacy in this specific setting some assumptions has to be made. First, the range of possible values must be known. In this thesis, a trimmed normal distribution, shown in Section 7.1, will be used for this purpose. The reason for trimming the distribution is to only allow positive values for the energy consumptions. Since choosing intervals at when energy consumptions should be sent is not the main topic of this thesis, for simplicity, energy consumptions will also be assumed to be sent once per hour.

Another assumption in this thesis is that the network used is both secure and reliable. Furthermore, for the scheduling scenario, the energy consumption sent on the network is assumed to have been verified in some way to assure that the data is correct, and does not contain negative values. However, in real life, one should never assume that the network is secure and reliable by default, this must be guaranteed by some other measures. Examples for achieving security are message authentication codes (MAC) and encryption, while examples for reliability are message acknowledgements and re-transmissions of lost messages. Additionally, the verification process for each consumption sent must be added in real life, but since this is not in the scope of this thesis it will be overlooked.

6.2 Design

One of the main contributions of this thesis is to create different partitioning strategies. First the information flow from smart meters to when the data processing happens is explained. Then a detailed introduction of four different partitioning strategies is given.

6.2.1 Model

The implementation in this thesis consists of two components; a smart meter and a server. This design is illustrated in Figure 6.1. Note, however, that there are several smart meters connected to each server.

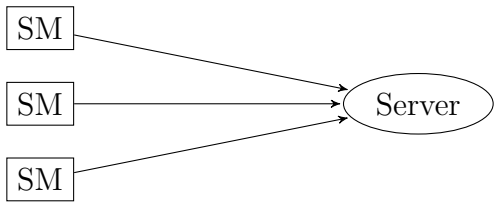


Figure 6.1: Each smart meter, denoted SM, is connected to a server which processes the data sent by the SM

Each smart meter is connected to a trusted server that reads its data. Every hour the smart meters sends data containing its energy consumption for the last hour which is then read by the server. An illustration of this implementation can be seen in Figure 6.2.

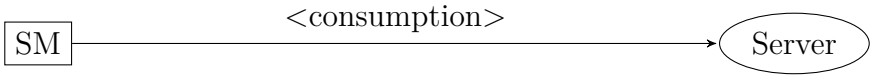


Figure 6.2: Each smart meter, denoted SM, sends data to the server once per hour

The server gathers data for the past hour, one reading per connected smart meter and performs some calculation. For the case without differential privacy applied the server just sums all data to get the true answer, but further actions needs to be taken when differential privacy should be applied.

Since differential privacy must be possible to apply to the use case, the server has to construct suitable queries, before applying the privacy mechanism. For the case of differential privacy, counting queries are suitable because they assure that the privacy mechanism adds only a small amount of noise to the true answer, since such queries have low sensitivity.

In order to find out how much energy each group of smart meters consumed several counting queries have to be constructed, as was explained in Section 5.4. These counting queries therefore count how many smart meters consumed energy within a certain interval. The queries are on the form "How many times was a kWh per hour consumed within the last hour?", where a is a range. An illustration of this is provided in Table 6.1.

Question	Answer
How many times was 0 kWh per hour consumed within the last hour?	Number
How many times was $0 + z$ kWh per hour consumed within the last hour?	Number
...	...
How many times was $y - z$ kWh per hour consumed within the last hour?	Number
How many times was y kWh per hour consumed within the last hour?	Number

Table 6.1: Queries constructed by the server, representing the energy consumption for the last hour. This represents the partitioning strategies translated to queries.

6.2.2 Partitioning Strategies

In this thesis, four different types of partitioning strategies are used to construct queries. All of the strategies were designed with the probability density function of the normal distribution, displayed in Figure 6.3 in mind. The normal distribution is used since real data is not used, but the normal distribution is a good approximation until the work can be further extended to include real data.

The strategies are tested with simulated smart grid data, presented in Section 7, and the results are then shown in Section 8, where the different strategies are compared.

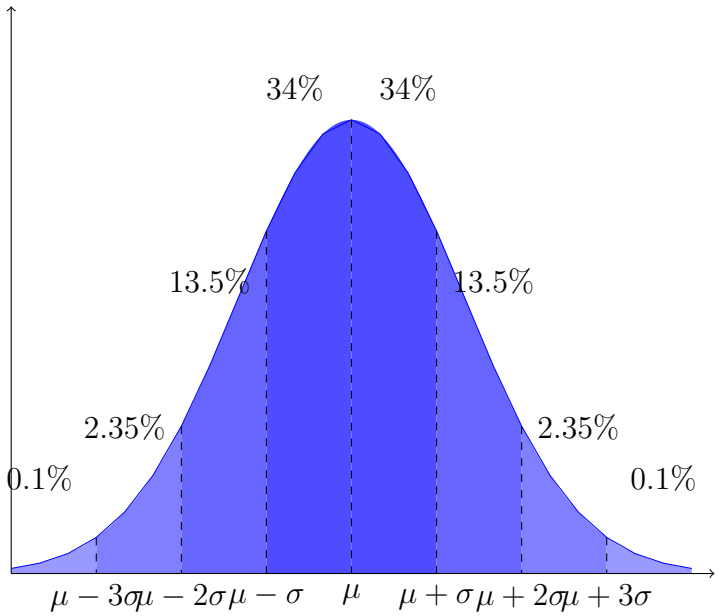


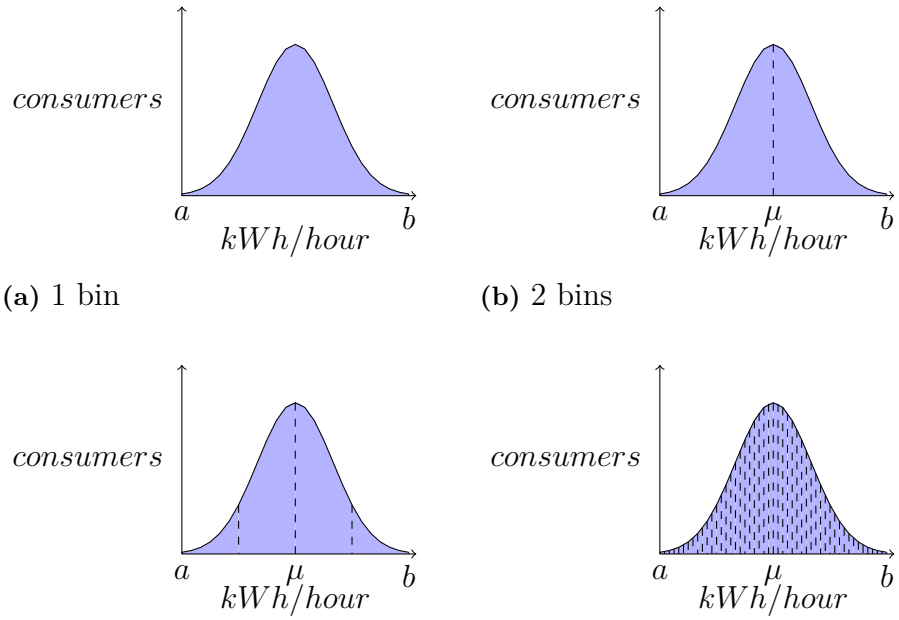
Figure 6.3: The probability densitiy function of the normal distribution with labels representing how many percentages of all values fall into each interval. μ is the mean and σ is the standard deviation of the distribution. Note that 95% of all values fall within the range $\mu - 2\sigma$ to $\mu + 2\sigma$.

6.2.2.1 Fine-Grained Partitioning

The first partitioning strategy that was constructed was the fine-grained partitioning strategy. This type of partitioning divides the whole range into smaller bins, where

each bin is equally big. The smallest case in this implementation is represented in Figure 6.4 (d). Different amount of bins using the fine-grained partitioning strategy are shown in Figure 6.4 (a), (b), (c) and lastly (d). Since the normal distribution is used when generating values, this is also shown as reference in each of the figures.

The first case of this partitioning strategy is where only one bin covering the entire range is used. This is illustrated in Figure 6.4 (a). The second case in fine-grained partitioning contains two evenly divided intervals, as shown in Figure 6.4 (b). After dividing the range into two bins, each bin is divided again to create four evenly divided bins. This is illustrated in Figure 6.4 (c). This division of each bin continues until a given value for how small the step between each bin has been reached.



(a) 1 bin

(b) 2 bins

(c) 4 bins

(d) Maximum amount of bins used

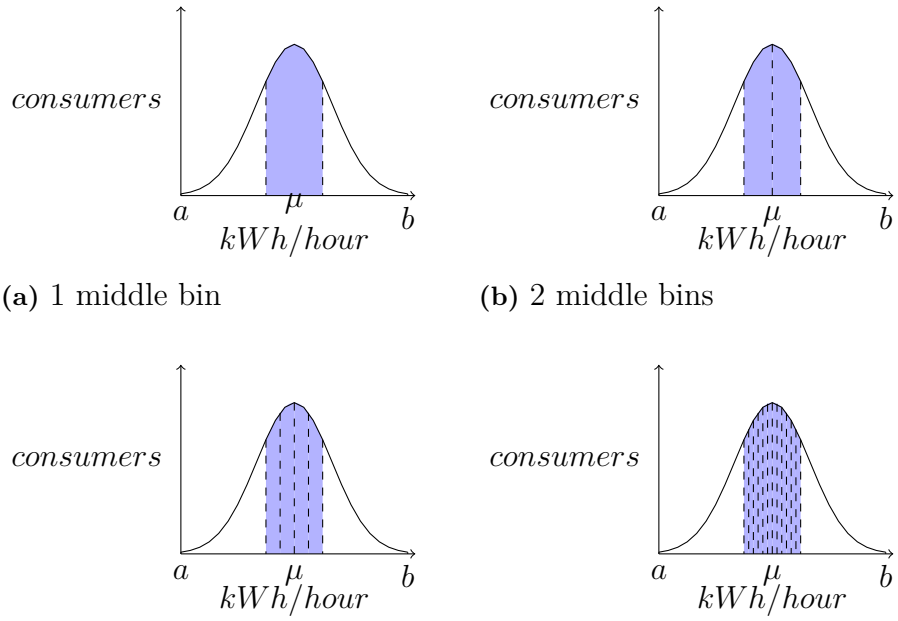
Figure 6.4: Different bin sizes for the fine-grained partitioning strategy. Note that every bin has the same size for each of the different partitionings.

If the maximum value is unknown, it becomes hard to properly apply fine-grained partitioning. Therefore some tweaks have to be made to make it more flexible. To make it more dynamic, bins that act as sinks, containing every value greater than the assumed maximum or smaller than the minimum, can be added. Note, however, that in order to benefit from the fine-grained partitioning it is important to choose a maximum value that is realistic, so most values end up in a small bin rather than in the sink bins.

6.2.2.2 Fine-Grained Mean Partitioning

The fine-grained mean partitioning strategy is similar to the fine-grained partitioning strategy. However, the fine-grained mean partitioning differs in the way that it only does the fine-grained partitioning around the mean, and then at each edge sink bins are added. The middle area is calculated based on the mean plus 1 standard deviation in both sides of the mean. As seen in Figure 6.3, 68 % of all values lies in this middle area.

The different steps for the fine-grained mean partitioning strategy can be seen in Figure 6.5. In Figure 6.5 (a) one bin is used for the middle, then in (b) the middle bin has been divided into two equal bins. This process of dividing the middle into equally large bins continues in (c) and finishes in (d). Note that there always exists two additional bins for each edge of the distribution.



(a) 1 middle bin

(b) 2 middle bins

(c) 4 middle bins

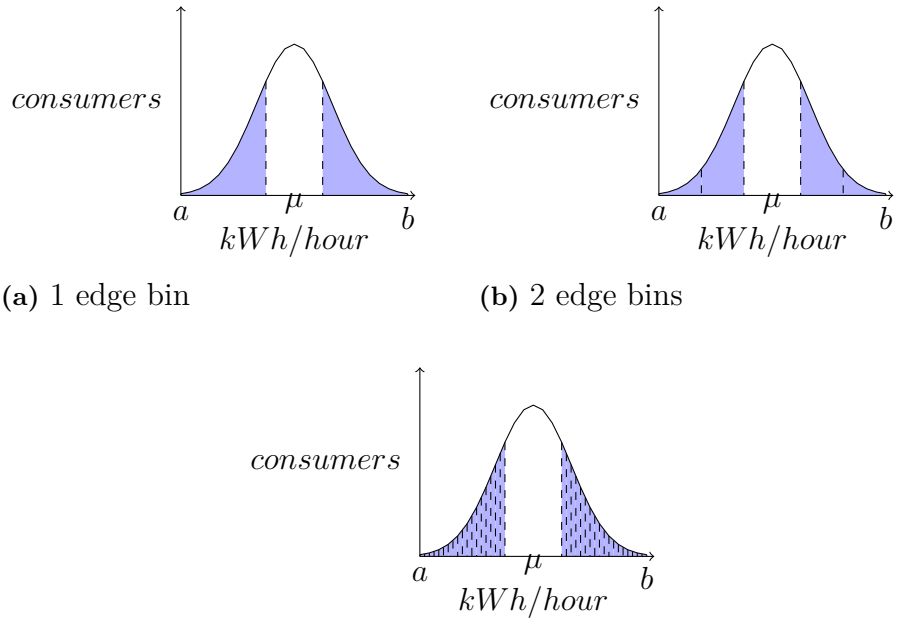
(d) Maximum amount of bins used

Figure 6.5: Different bin sizes for the fine-grained partitioning strategy. Note that only the middle bin continues to get divided, and that each of those bins are equally large.

6.2.2.3 Fine-Grained Edges Partitioning

The third partitioning strategy is called fine-grained edges partitioning strategy. This partitioning strategy divides the edges of the distribution into smaller bins, while the middle area is considered one bin. Each edge represents the area outside the mean which is 16% of the population, for a total of 32% of the population according to the probability density function of the normal distribution.

Figure 6.6 shows the different bin sizes used. Figure 6.6 (a) shows how each edge of the distribution has one bin each, then in (b) each edge has been divided into two bins for a total of five bins. In (c) the edges have been divided into the maximum amount of bins.



(a) 1 edge bin

(b) 2 edge bins

(c) Maximum amount of bins used

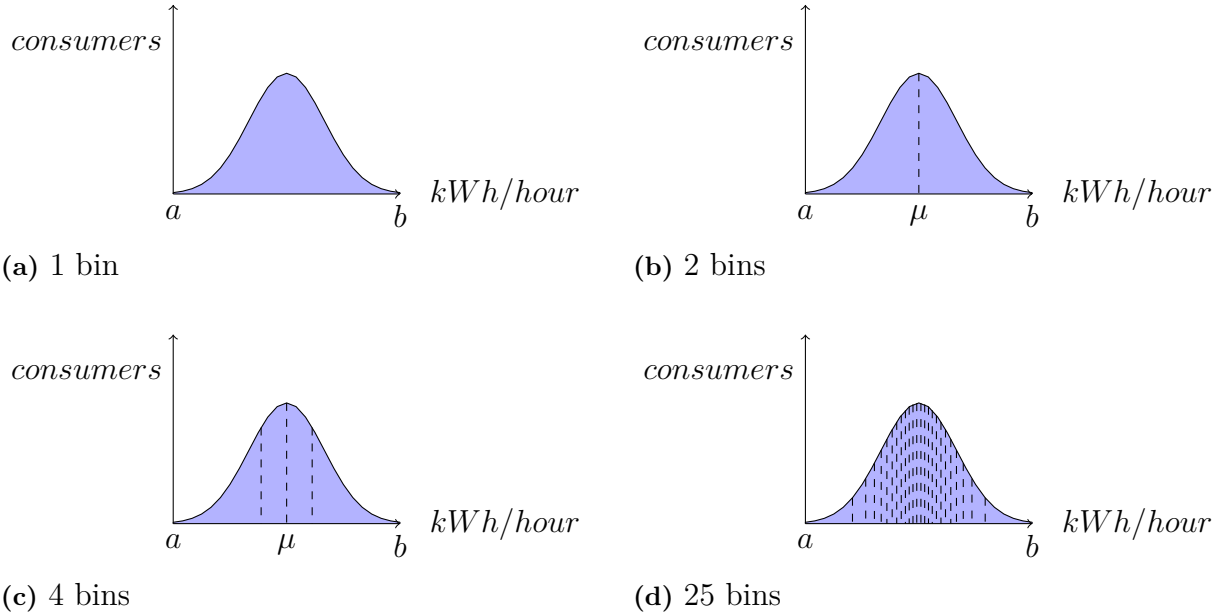
Figure 6.6: Different bin sizes for fine-grained edges partitioning strategy. Note that both edges are divided into more bins, while the middle is always one bin. Also note that the bin sizes for the edges are equally large for each partitioning.

6.2.2.4 Percentage Partitioning

The previously presented partitioning strategies focus on creating bins that all cover an equally large interval of the range. However, due to the values being normally distributed this can lead to bins that are very unevenly filled since most of the generated values will be close to the mean. This could even lead to some bins being completely empty, which means that once they are queried in a differentially private manner they will just return noise.

In order to deal with this potential problem another partitioning strategy which focuses on creating bins that covers a certain percentage of the entire range. This percentage partitioning strategy divides the entire range into n bins with x percent in each bin. A visualization of this partitioning scheme is provided in Figure 6.9, where each bin covers 1% of the entire interval.

Figure 6.7 show some of the bin sizes used for the percentage partitioning strategy. The first step has one bin (a) which is 100% of the range, the second (b) two bins with 50% in each, and the third (c) has four bins with 25% of the population in each bin. Picture (d) shows 25 bins, where each bin holds 4% of the population. Note that the bins cover an equal percentage range of the population each.



(a) 1 bin

(b) 2 bins

(c) 4 bins

(d) 25 bins

Figure 6.7: The different bin sizes for the percentage partitioning strategy. Note that all bin sizes cover an equal number of percentages of the range.

Figure 6.8 has 50 bins. Note that the two edge bins are not equally large, since every bin covers 2% of the entire range.

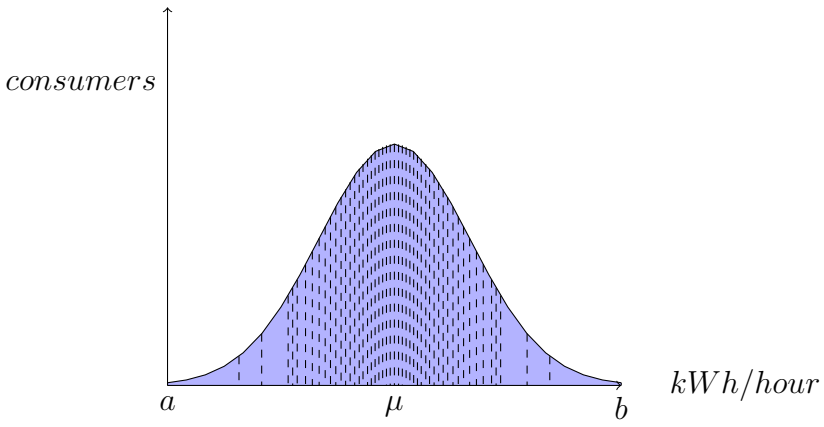


Figure 6.8: Percentage partitioning, where each bin covers 2% of the entire range. Note that 0.03% of the values will be below point a and 0.02% will be above point b.

The last step for the percentage partitioning strategy is shown in Figure 6.9. Here each bin covers 1% of the entire range.

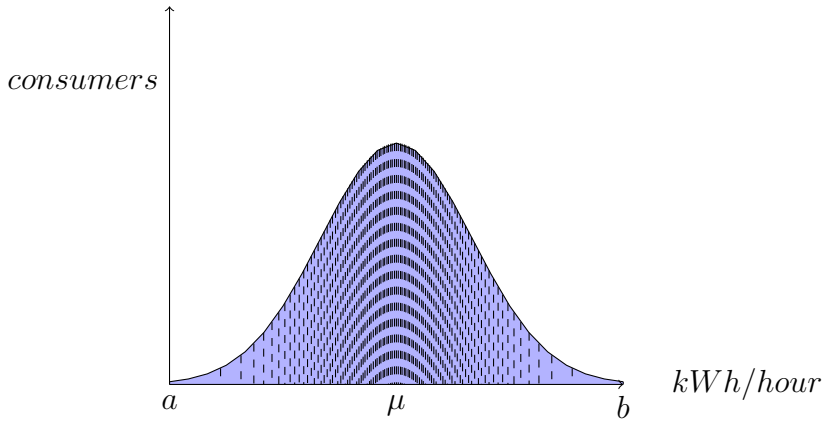


Figure 6.9: Percentage partitioning, where each bin covers only 1% of the entire range. Note that 0.03% of the values will be below point a and 0.02% will be above point b.

All of the points for the x-coordinates in the previously shown Figures have been taken from *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences* [25]. Since the exact values for the x-coordinates in most cases do not exist, the closes rounded up value is used instead.

6.2.3 Scheduling Scenario Design

In the implementation of the use case the server will calculate both a true answer and a response. This corresponds to the mean and the differentially private mean for the energy consumption for a group of smart meters. After calculating both means, the difference between these will be calculated, which is the final result. This process is illustrated in Figure 6.10.

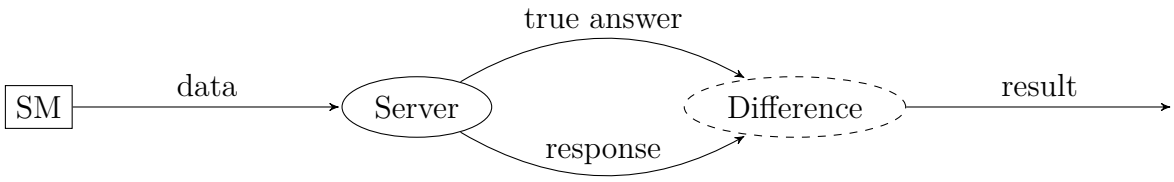


Figure 6.10: Each smart meter, denoted SM, sends data to the server once per hour. The server then calculates the true mean and the differentially private mean before the result is calculated. Note that the comparison between the true answer and the response is a way to evaluate the result; in a real implementation the server would only release the response.

In order to calculate the true answer, the server gathers data, holding one reading per connected smart meter, and then calculates the mean. This is illustrated in

Figure 6.11.

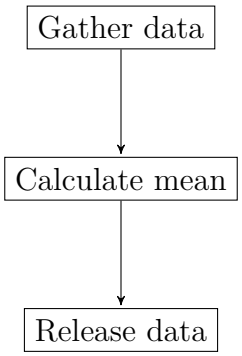


Figure 6.11: The processing of data done by the server before it can be released. Notice that the server holds one consumption reading per smart meter.

In order to apply differential privacy, some changes have to be made to the chain of events. Instead of directly releasing the true answer, the server now has to apply first the counting queries to the data set, then apply the privacy mechanism to each query. An illustration of how data flows can be seen in Figure 6.12.

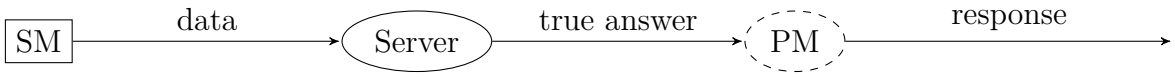


Figure 6.12: Each smart meter, denoted SM, sends data to the server once per hour. For the case with differential privacy applied, noise has to be added to the true answer for every query. This is done by the privacy mechanism, PM, which resides in the server.

The server will gather data for the last hour, and partition the entire range using the strategies from Section 6.2.2. For each bin, one query is run. After this has been done the queries are applied in the third step. In the fourth step the privacy mechanism will be applied to each query answer, before it is released to the analyst. This process is illustrated in Figure 6.13.

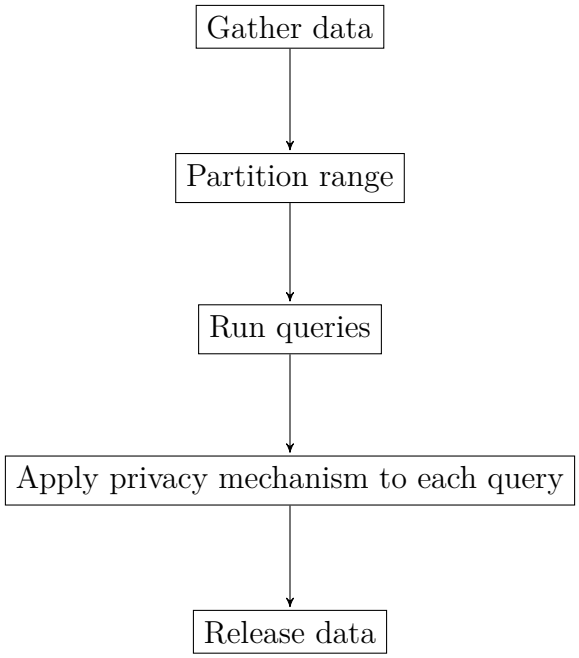


Figure 6.13: The processing of data done by the server before it can be released. Note that more steps are added to the process when differential privacy should be applied.

7
Data Simulation

In order to apply differential privacy to the use case, data must be generated. This data is generated according to a normal distribution explained in Section 7.1. Then the percentage difference between the true sum and the differential private sum can be compared. To make it possible to draw an arithmetic mean, each calculation should be run at least n times to get a certain confidence interval. The calculation of how many samples will be run is made in Section 7.2

7.1 Statistics

To choose a plausible range for the electricity consumption for the experiments, statistics from the Swedish government agency Statistics Sweden [37] have been used. Statistics concerning electricity consumption for household purposes, without including heating, in Sweden is hard to come by. Therefore statistics where heating using solely electricity is included as well as electricity for household purposes is used in this thesis, as it best fits the requirements. The statistics retrieved from Statistics Sweden, shown in Table 7.1, concern electricity consumption per square meter for one and two dwelling buildings.

Type \ m^2	–85	86–100	101–120	121–140	141–160	161–200	201–
Electricity (d)	188	165	147	131	129	136	125
Electricity (w)	198	182	177	136	129	132	133
Heat pump	172	165	127	110	112	104	112

Table 7.1: The table shows the electricity consumption, in kWh per year, per square meter for one and two dwelling buildings. Note that the statistics are for 2008 which was a leap year. The abbreviation d corresponds to direct electricity and w is water-borne electricity.

Since the proportion of houses that are heated with the different methods is not known, this thesis will assume the worst case heating type, which in the case of Table 7.1 is water-borne electricity. This will not significantly change the method or validation of the thesis. In order to use these statistics, however, it is also necessary to know more about the size of a house. To estimate the average house size statistics from Brosenius [2] is used. Borsenius states that the average size of a one and two dwelling building in 1967 was 110 square meters.

By combining the statistics from Table 7.1 with the average size of a one and two dwelling building it is possible to calculate the mean electricity consumption per hour.

$$\mu = \frac{177 \times 110}{24 \times 366} \approx 2.217$$

Compared to the average consumption for an American household in 2013, which is 1.263 kWh per hour according to the U.S. Energy Information Administration (EIA) [40], this number is a bit higher. However, due to the fact that Sweden has different climate from the US and that the statistics in this thesis uses electricity as the main source of heating, this is to be expected. Another reason for why this comparison is made is to make sure the assumed consumption is not too low, since the worst case is to be assumed.

When the mean has been found, it is then possible to generate values from this mean by applying a normal distribution. However, since all the individual data points used to calculate the mean size of a one and two dwelling building are not given by Borsenius, it is not possible to calculate the standard deviation, σ. Because of this a value for σ has to be assumed.

Since σ is unknown, the worst possible value for it will be assumed. In this scenario the worst case is when the standard deviation allows energy consumptions to be zero, due to the fact that consumptions in this model cannot be negative. To find a value for σ where the probability density function ranges from zero, one can solve the following equation for zero.

$$z \times \sigma + \mu = 0 \Leftrightarrow$$

$$\sigma = \frac{\mu}{z}$$

By inserting the lowest value for z, representing the point at 0.03% of the population, from Milton and Arnold's book [25] it is possible to solve for σ.

$$\sigma = \frac{2.217}{-3.4} \approx 0.625$$

An important thing to note is that the normal distribution used has been trimmed. This is because it is not realistic to have values ranging from minus to plus infinity when dealing with energy consumptions. Values are therefore generated according to a normal distribution, but all values below the 0.03% mark and above the point for 99.98% of the population are converted to the minimum or the maximum value respectively. This is illustrated in Figure 7.1. As can be seen, only the white area of the graph contains allowed values, that is; all values between a and b will be included.

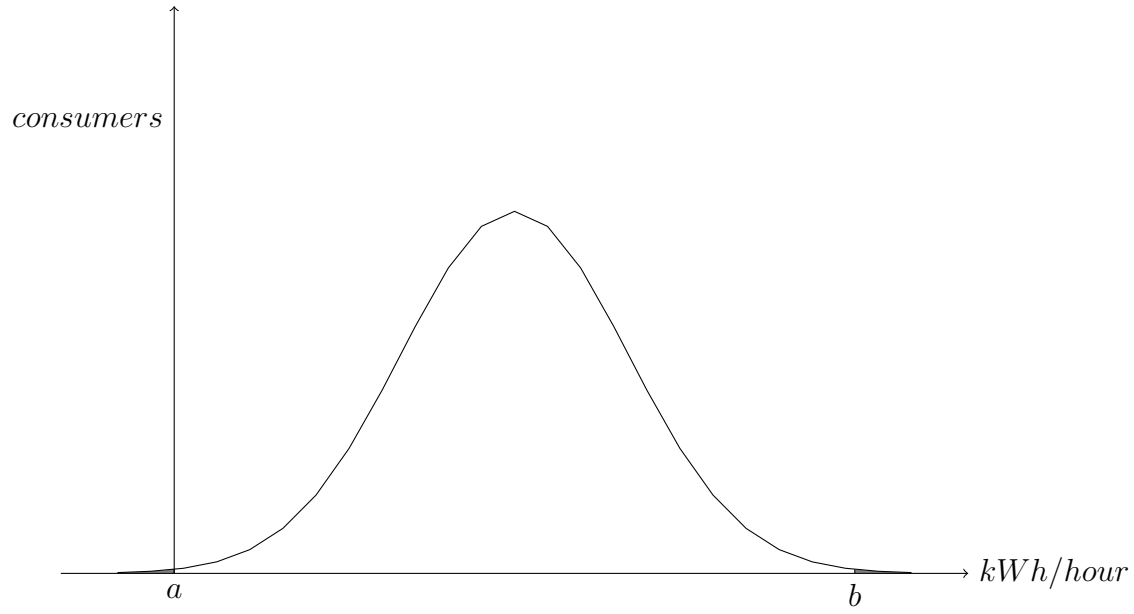


Figure 7.1: The trimmed normal distribution, where the shaded areas represent values that will be converted to the minimum value, a, or the maximum value, b, depending on on which side of the curve they end up. Note that the white area represents 99.95% of the entire population, which means 0.05% will be trimmed away.

7.2 Samples

In order to get the confidence interval 95% the number of calculation samples needed is according to Milton and Arnold [25] the following.

$$\frac{{z_{\alpha/2}}^2 \times \sigma \times (1-\sigma)}{\text{margin of error}^2}$$

To get a result within 0.05 with 95% confidence interval this becomes the following.

$$\frac{1.96^2 \times \sigma \times (1-\sigma)}{0.05^2} \approx 350$$

Because of this, each calculation has to be run 350 times in order to achieve a confidence interval of 95%. All of the settings used for the calculations are shown in Table 7.2. Each calculation will be run for a different amount of households, ranging from 100 to 1,000 with a step size of 100. For each calculation, different amount of bins will also be used, to see the correlation between noise and amount of queries.

Settings	
Parameter	Value
Number of Smart Meters	[100,1000] with step 100
Input Range (kWh)	Trimmed normal distribution: μ=2.217 σ=0.652
Number of Bins	[1,2,4,5,10,20,25,50,100]
ϵ	1
Sample Size	350

Table 7.2: Setup for the modifiable variables for the different partitioning strategies. Note, however, that not all partitioning strategies can handle all number of bins due to their nature.

Note that the bin sizes used are numbers that 100 are divisible by. This is to make sure that all bin sizes work for the percentage partitioning strategy, since all bins should contain an even number of percentages of the population.

8

Result

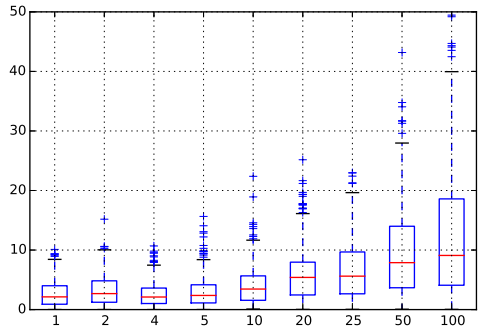
In the implementation presented in Section 6 the difference between the true sum and the differentially private sum is calculated in percentages. To show a correlation between the number of queries asked and the error rate, each computation of the sum has been run with different amount of queries. Furthermore, to show a correlation between the number of households and the error added the implementation has also been run with different amount of households.

8.1 Comparison of the Partitioning Strategies

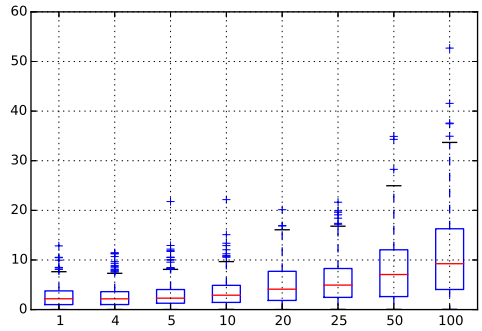
To compare the different partitioning strategies, each was tested by changing the number of households from 100 to 1,000 using 100 as the step size. First the box plots of the confidence interval will be compared in Section 8.1.1, which show the spread of possible values for the error. Then, the arithmetic mean of the error will be compared in Section 8.1.2.

8.1.1 Box Plots

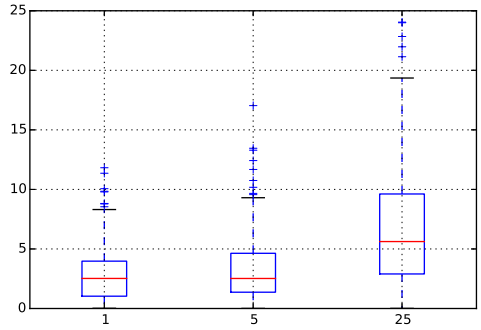
To visualize the spread of the 350 samples, box plots are used. Since the worst result and the best result is when 100 and 1,000 households respectively are used, the box plots show these two scenarios. The other results when 200 to 900 households are used are shown in Appendix A. In Figure 8.1 the spread of the calculations when 100 households are used is shown and in Figure 8.2 the spread of the calculations when 1,000 households are shown.



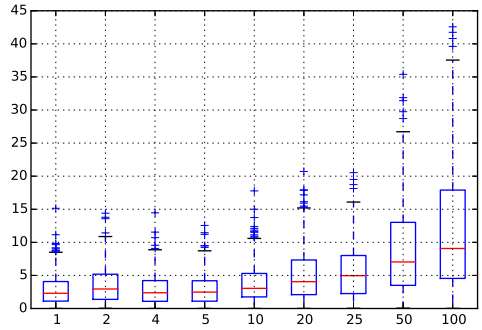
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning



(c) Fine-grained edges partitioning

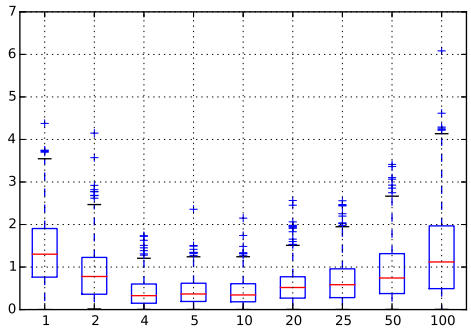


(d) Percentage partitioning

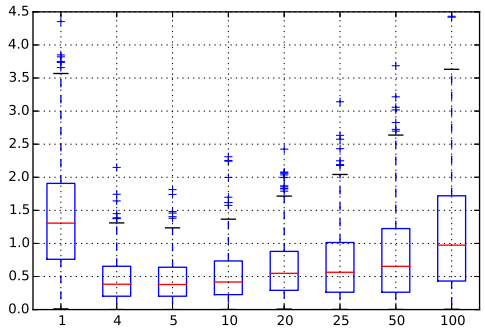
Figure 8.1: Readings from 100 simulated households. Note that the spread increases as the number of queries increases. Also note that the scale differs for the graphs. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.

The worst spread is received when 100 households are used. As can be seen in Figure 8.1 the spread of the data is tighter when the bin sizes are bigger. That is when fewer queries are asked. When the number of bins increases the spread gets less tight. This is because more noise is added when more queries are asked. Figure 8.1 (a) shows the fine-grained partitioning strategy and as can be seen here the worst case is when 100 bins are used. This will yield an error of 0.01% to 52.71%. However, all the partitioning strategies yield a lower error when 1 to 10 bins are used. Then the percentage difference is between 0% to 22.38%.

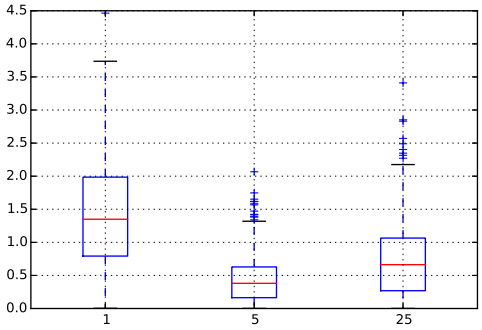
The tightest spread is received when 1,000 households are used. Figure 8.2 illustrates the spread from all the partitioning strategies and as can be seen the error is remarkably lower than when using 100 households. For all the partitioning strategies the error is at most 6.08%. The worst result is from the fine-grained partitioning strategy when using 100 bins. The best results are between 0% to 2.07%, where the fine-grained partitioning strategy yields the lowest result, between 0% to 1.74%, for 4 bins.



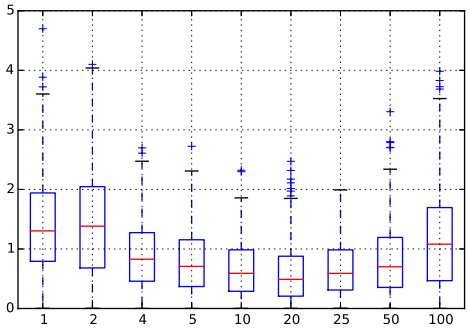
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning



(c) Fine-grained edges partitioning



(d) Percentage partitioning

Figure 8.2: Readings from 1,000 simulated households. Note that the spread increases as the number of queries increases, but also that the spread is large when using 1 query. Also note that the scale differs for the graphs. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.

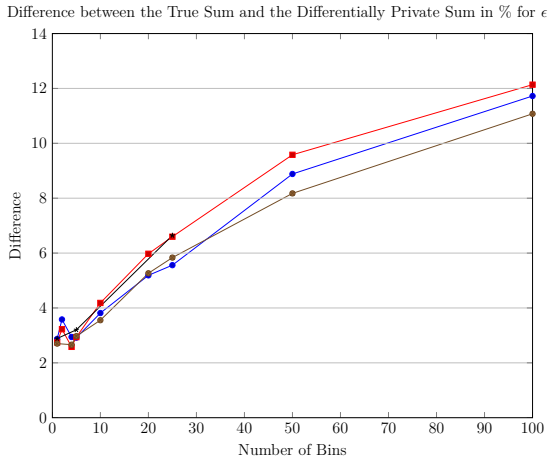
8.1.2 Arithmetic Mean

The results shown in this section are the arithmetic mean difference between the true sum and the differentially private sum, in percentages and the median difference in percentages. Using 1,000 households, again, gave the best results. These results are shown in Figure 8.6 and Figure 8.7.

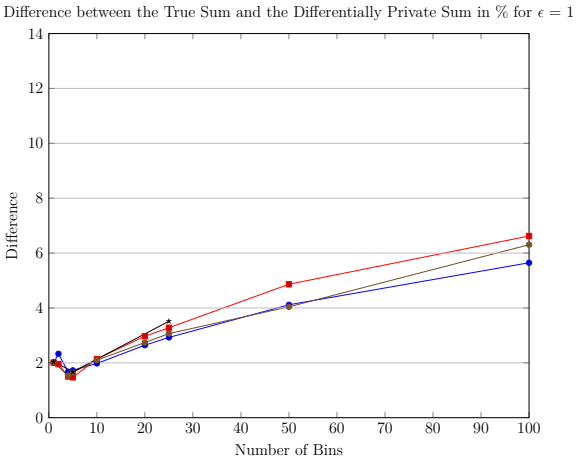
For relatively few values, 100 simulated households, all strategies have several cases where the results are worse than using 1 bin. As can be seen in Figure 8.3 (a), all of the results get increasingly worse as more bins are added. When 200 simulated households are used, the results are better than when using only 100. However, none of the partitioning strategies improves as more bins are added. These results can be seen in Figure 8.3 (b).

In the case of using 300 simulated households, the curve for all partitioning strategies are smoother. Still, none of them manage to achieve much difference in their results as more bins are added. This can be seen in Figure 8.3 (c). For 400 simulated

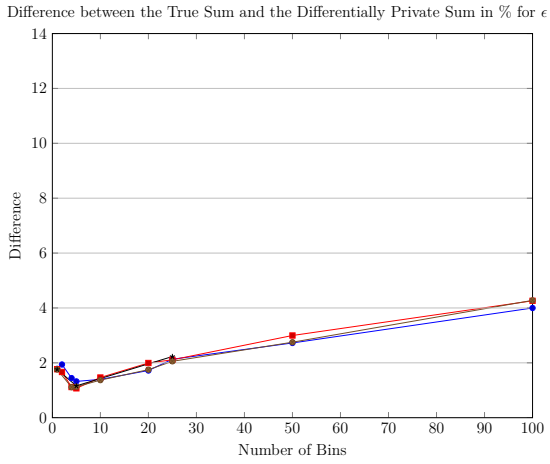
households, the results continue to improve for all partitioning strategies, compared to using fewer simulated households. Nonetheless, there are still bin sizes, such as 50 and 100, that still yield worse result than using only 1 bin. Some of the results, for example when using between 4 and 20 bins are better than using 1 bin. This is illustrated in Figure 8.3 (d).



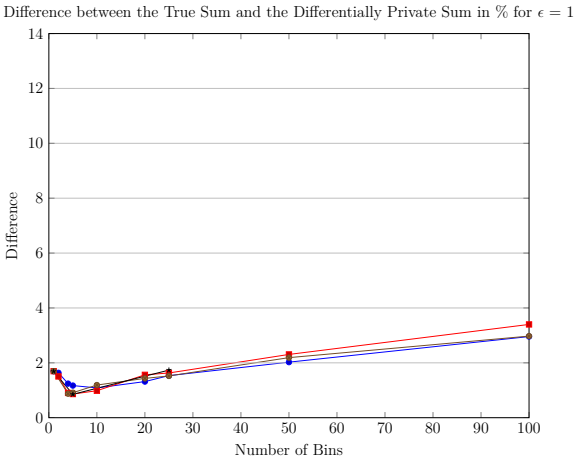
(a) Readings from 100 simulated households



(b) Readings from 200 simulated households



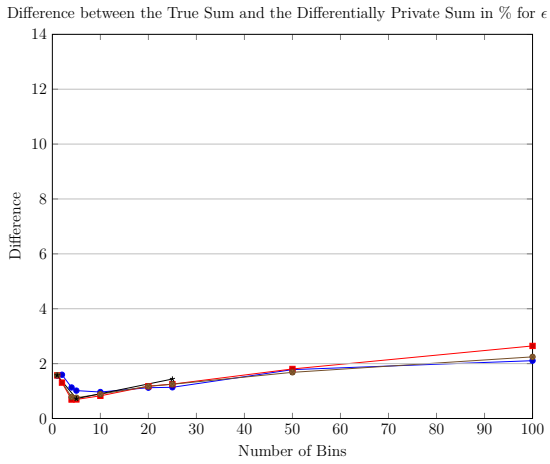
(c) Readings from 300 simulated households



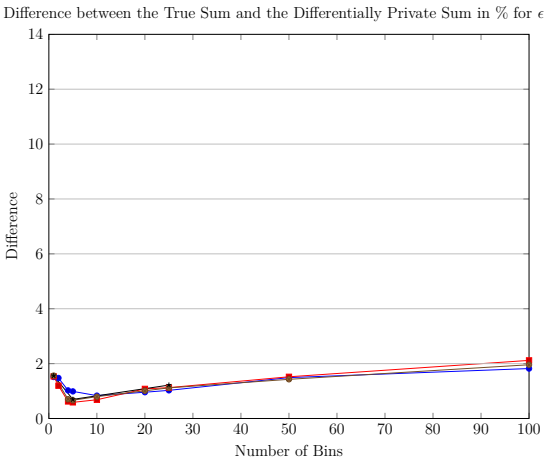
(d) Readings from 400 simulated households

Figure 8.3: The results for the fine-grained partitioning strategy is represented by the red line, the fine-grained mean by the brown and fine-grained edges by the black line. Lastly, the blue line shows the results for the percentage partitioning strategy.

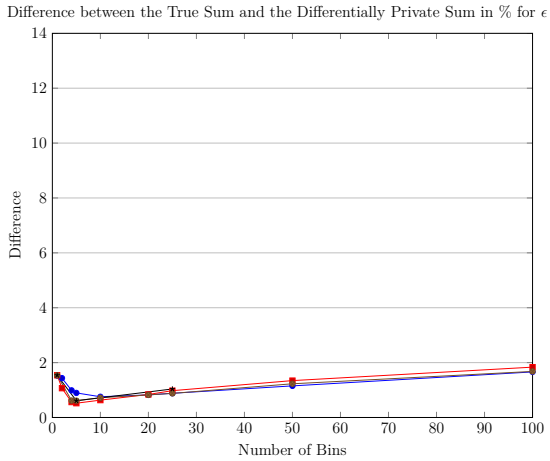
When using 500 simulated households the worst result for all partitioning strategies is almost the same as when using only 1 bin, as can be seen in Figure 8.4 (a). Some of the bin sizes, between 4 and 20, still continue to get lower error. The same goes for Figure 8.4 (b), where 600 simulated households are used. Moving on to 700 and 800 simulated households, Figure 8.4 (c) and (d) respectively, the curve continues to get smoother. An interesting note is that almost all bin sizes now are better than when only 1 bin is used.



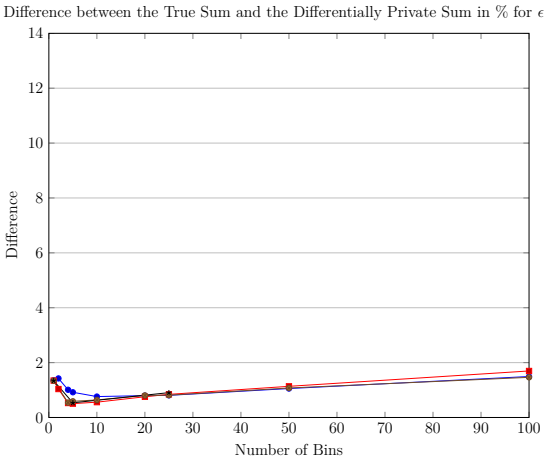
(a) Readings from 500 simulated households



(b) Readings from 600 simulated households



(c) Readings from 700 simulated households



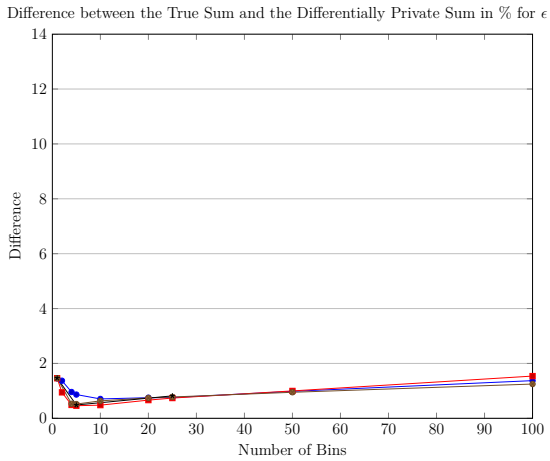
(d) Readings from 800 simulated households

Figure 8.4: The results for the fine-grained partitioning strategy is represented by the red line, the fine-grained mean by the brown and fine-grained edges by the black line. Lastly, the blue line shows the results for the percentage partitioning strategy.

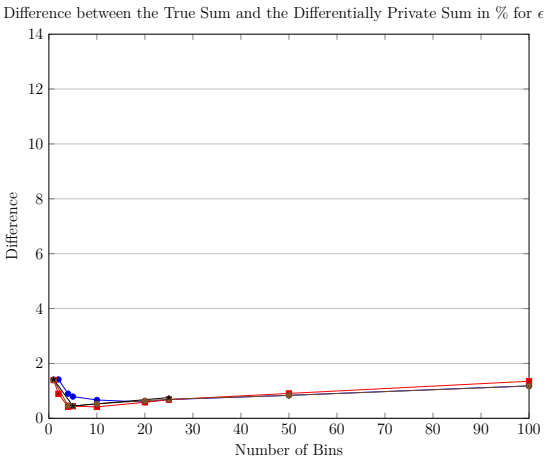
As even more simulated households are used, all strategies continue to improve. This can be further seen in Figure 8.5 (a) and (b).

Figure 8.6 shows an evident trend where all the partitioning strategies yield good result, especially when 4 to 20 bins are used. However, as the number of bins increases the results get worse. It is interesting to note that all partitioning strategies shows the same trend and none of the partitioning strategies difference is above 1%, except for the case when using 1 bin.

The best mean result when comparing the different partitioning strategies is given when 10 bins are used for the fine-grained partitioning strategy; then the error is 0.42%. All partitioning strategies yield a result in the range of 0.42% - 0.67% error



(a) Readings from 900 simulated households



(b) Readings from 1,000 simulated households

Figure 8.5: The results for the fine-grained partitioning strategy is represented by the red line, the fine-grained mean by the brown and fine-grained edges by the black line. Lastly, the blue line shows the results for the percentage partitioning strategy.

when using 10 bins. Note that not all strategies had their best result when 10 bins were used. The percentage partitioning strategy got its lowest result at 0.59% using 20 bins, the fine-grained mean partitioning strategy got 0.45% error using 5 bins and the fine-grained edges also got 0.45% error using 5 bins. Therefore, the lowest error is between 0.42% and 0.59% if all partitioning strategies are compared, however, these results were not all produced using the same amount of bins.

Furthermore, from observing the median results for 1,000 simulated households, shown in Figure 8.7, it can be seen that the results are very similar to those from the mean difference. However, almost all values are a bit lower when the medians are compared. For example, the lowest value for the error for the fine-grained partitioning strategy is 0.34% for 10 bins, compared to 0.42% when the mean values were compared.

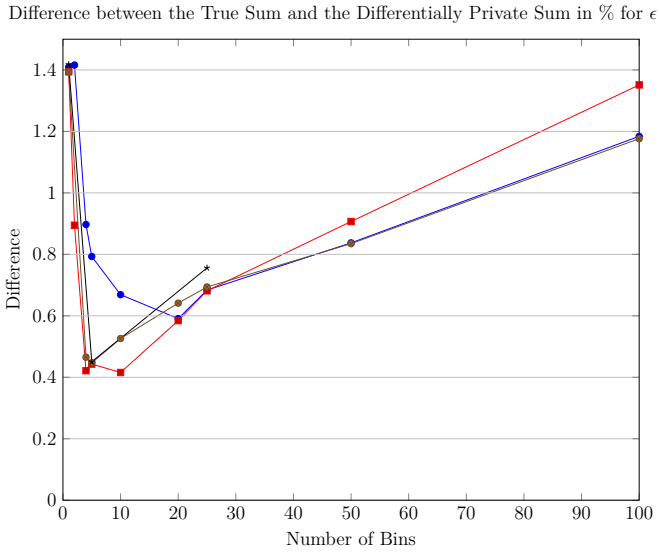


Figure 8.6: The results for the fine-grained partitioning strategy is represented by the red line, the fine-grained mean by the brown and fine-grained edges by the black line. Lastly, the blue line shows the results for the percentage partitioning strategy. This experiment used readings from 1,000 simulated households.

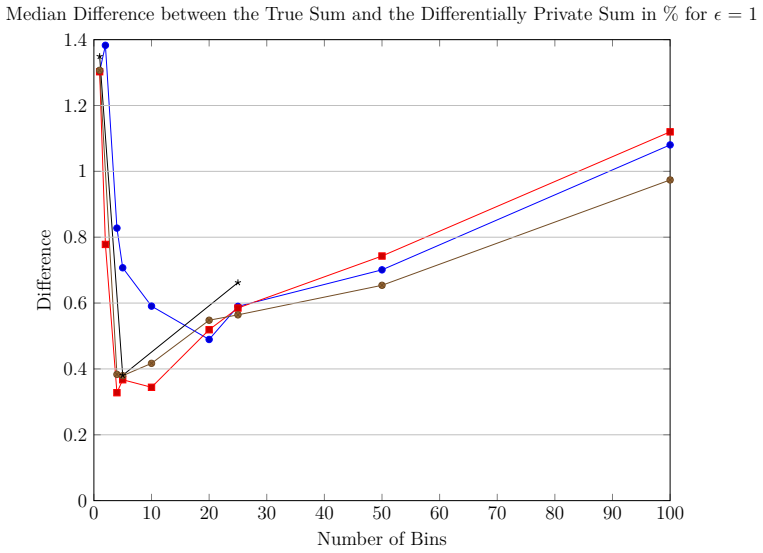


Figure 8.7: The results for the fine-grained partitioning strategy is represented by the red line, the fine-grained mean by the brown and fine-grained edges by the black line. Lastly, the blue line shows the results for the percentage partitioning strategy. This experiment used readings from 1,000 simulated households.

9
Discussion

In this section we critically discuss the results, and reflect on why they are obtained. The different partitioning strategies are compared, since this provides an important reflection on how the different strategies compare to one another. Then we discuss the utility of our results. We also discuss the use of statistics, and the assumptions that are made in this thesis since this is an important aspect that affects the results. Thereafter we discuss the use of different use cases, before we discuss the sustainability and ethics aspects of our work. Lastly, we give a recommendation for future work.

9.1 General Discussion of the Results

The partitioning strategies in this thesis show a clear trend when investigating the results. All strategies show potential when around 4 to 20 bins are used. However, the results get worse when an increasing number of bins are used. This is due to the fact that each bin represent one added query, which in turn adds more noise. Also when 1 bin is used the result will be affected negatively, since this is the same as calculating the average of the entire range and adding noise to it. This speaks in favour for our partitioning strategies, as they can improve the accuracy of the result.

We also see a correlation between using more households and the error added to the result. When more households are used, we get a better result. This can be explained partly by the fact that there are less empty bins. The other part of the explanation is that the relative size of the noise decreases in comparison to the true answer when the size of the true answer increases. This is because the noise stays the same no matter how big the query answer gets, since the noise only depends on the sensitivity of the query and not on the size of the data set.

9.2 Comparison of the Partitioning Strategies

We get the lowest error from using the fine-grained partitioning strategy, however, it does not always provide the best results as the amount of bins continue to grow. The reason for why it is not always the best strategy is that the bins on the edges will capture a too small part of the entire population. If a bin is empty, or only holds a small part of the population, it will be more affected by the noise, since the

answer will not conceal it.

If we want to make sure that no bins are empty, the percentage partitioning strategy is a good choice. This is because the percentage partitioning strategy, statistically, will have no empty bins. As the results for the fine-grained partitioning strategy and the percentage partitioning strategy are similar they can be used interchangeably. Therefore it should not differ much which one is used. However, the percentage partitioning strategy is easier do adapt to different distributions, so it is more flexible.

9.3 Utility of Results

Privacy has a price. In this case a certain error is the price that has to be paid. We have shown in this thesis how error rates correspond to the number of households used and the amount of queries asked. If a company wants to apply our results, they would first have to decide upon the error rate they are willing to introduce in order to achieve privacy. From our results they can then find out how many households they would have to include, and how many queries they need to ask in order to fall within the specified error rate.

9.4 Data and Statistics

A normal distribution has been assumed in order to verify the usefulness of our strategies. However, this does not mean that our strategies only work for a normal distribution. Due to the fact that our strategies place their bins on points that represent a certain percentage of the entire population, this could be applied to any other distribution. This is especially true for the percentage partitioning strategy, since all bins are chosen to represent n percentages of the entire population.

In this thesis we have used statistics to assume the average consumption by calculating a mean value and then applying a normal distribution. However, if one has access to real data this mean could be more accurately calculated. In this case, though, we only use this data to verify that our partitioning strategies work.

The normal distribution used for generating values in this thesis has been trimmed. This is because we cannot simulate values that are in the range minus to plus infinity. Besides, values below zero would not be realistic values for electricity consumption in our model, since we do not consider smart meters that introduce new energy into the system. Therefore we have chosen to not include values below the point where 0.03% of the population would fall, and not the values above where 99.98% of the population belongs to. This means that we do not account for 0.05% of the entire population.

In this thesis we had to estimate σ for the statistics concerning energy consumption, since the statistics lacked this information. Due to the fact that we used the absolute worst case for our model, knowing σ should only improve the results.

If companies were to apply our strategies they would probably be able to choose better minimum and maximum values, as it is likely that they would already have detailed knowledge about the data distribution. Note, however, that in order for them not to leak information about the distribution, they cannot analyze the data they will apply differential privacy to before they apply it. This means that they would have to base their choice of distribution either on old or already public data in order to prevent leaks.

Also, if an electricity company wanted to apply our strategies to their data, they would probably validate it by removing erroneous energy consumptions. Note that our trimming of the normal distribution is similar to this validation, as it also removes values that are not realistic for their setting. The use of validation also means that there is a potential to get better results than the ones we got, since the trimming done by a company might be tighter than ours, and thus no extreme values would be included in the calculation. Because results could improve if companies apply our partitioning strategies, it would be interesting to carry out more research supported by more data.

9.5 Use Cases

In this thesis we have put our main focus on a scheduling scenario. However, we have also introduced a couple of other use cases that can be of interest in the setting of the smart grid. We acknowledge that there exist other uses cases that we have not implemented, but can be just as valid. Even though these use cases have not been implemented in this thesis they bring an interesting discussion.

The first other use case is the detection of fraud. In order to preserve the privacy of other consumers, who are not conducting fraud, differential privacy could be applied. It would then be possible to query per block, for example one could ask "How many consumed a negative amount of kWh in block x?", and if the answer is positive this could help pinpoint where there is possible fraud. This would mean that all innocent consumers do not have to be investigated further, while blocks where potential fraud is discovered require further inspection.

A similar approach could be used for the detection of suspicious activity. In this case it would be possible to ask "How many consumed more than y kWh in block z?". A positive answer in this case would mean that the block have to be further investigated, but a small enough answer would mean that everyone in the block is probably innocent.

It is important to note that even if an answer in the two last use cases come back

positive, it is not certain that there is illegal activity. This is because the noise added by differential privacy, which means that even when the true answer is zero, the differentially private answer will not be zero. Because of this a threshold needs to be established, which represents how much noise added still corresponds to the value zero.

9.6 Ethics and Sustainability

The work carried out in this thesis aims to enhance the integrity of data points in a data set. In this case those data points are energy consumptions, which can be used to deduce information about individuals living patterns, as we brought up in Section 3.1. Therefore our application of differential privacy can be used to enhance the privacy of individuals, which we think makes our work highly ethical.

When it comes to sustainability our work can be used to make the smart grid more efficient. This is because our scheduling use case, previously explained in Section 4, provides a way to predict how much energy should be distributed to a block of houses in advance. Because it would be possible to predict the amount of energy needed, it would no longer be necessary to produce excessive energy, which means there would be no need to have equipment for a larger capacity than what is actually required.

The smart grid is the way forward to a more sustainable energy distribution and sustainability will become an even more pressing issue in the future. Since our work provides much needed privacy in the smart grid, it could potentially make consumers more compliant to let service providers use their data to improve their services. Also, we predict that users will get more involved in the way their data is used and privacy concerns will be a much larger issue in the future. Our work will help service providers tackle some of the privacy issues and work to evolve the smart grid even further.

9.7 Future Work

There are several interesting directions for future work in this thesis. One interesting direction for future work would be to investigate more partitioning strategies. We believe there are even more ways to query the data set, which might even be better to study. Another interesting direction is to use our partitioning strategies when using a real time stream processing engine. By using a stream processing engine, an even more realistic environment could be simulated where data can be collected hour by hour. It would be interesting to see the results when using the partitioning strategies in this environment. Lastly, we would like to test our partitioning strategies on real smart meter data in the future to see if this will yield as good a result as with the simulated data.

10
Conclusion

In this thesis differential privacy has been applied to a real use case in the smart grid, showing that differential privacy can be used in this setting. Furthermore, four different partitioning strategies have been designed to query a generated data set by using counting queries only. These partitioning strategies have then been investigated by comparing them to each other, to show how using a different number of queries affects the accuracy of the results. The fine-grained partitioning strategy yielded the best mean results, with a 0.42% error when using 10 bins. This thesis also shows that the best case, when comparing the entire error spread, is when using 4 bins with the fine-grained partitioning strategy. The error is then between 0% to 1.74%. However, the error for all strategies range from 0% to 2.07%, which means the difference between the best and the worst is 0.33 percentage units.

The comparisons have also been made for different number of households. In this case using 1,000 simulated households results in the highest accuracy for all partitioning strategies. Due to the low error induced by differential privacy, a company could decide how much they are willing to pay for privacy, and investigate what setup they should use to achieve this.

Bibliography

[1] C. C. Aggarwal. On k-anonymity and the curse of dimensionality. In *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB '05, pages 901–909. VLDB Endowment.

[2] H. Brosenius. Energival vid småhusuppvärmning: Direkt elvärme eller flexibel vattenburen värme? *Meddelande. Inst. för byggnadsteknik*, 91, 1970.

[3] C. Clifton and T. Tassa. On syntactic anonymity and differential privacy. In *ICDE Workshops*, pages 88–93, 2013.

[4] E. Commission. A joint contribution of DG ENER and DG INFSO towards the digital agenda, agenda 73: set of common functional requirements of the smart meter full report. `http://ec.europa.eu/energy/sites/ener/files/documents/2011_10_smart_meter_funtionalities_report_full.pdf`. [Accessed 2015-05-27].

[5] G. Danezis, M. Kohlweiss, and A. Rial. Differentially private billing with rebates. In T. Filler, T. Pevný, S. Craver, and A. Ker, editors, *Information Hiding*, number 6958 in Lecture Notes in Computer Science, pages 148–162. Springer Berlin Heidelberg.

[6] D. Denning and J. Schlörer. Inference controls for statistical databases. *Computer*, 16(7):69–82, July 1983.

[7] D. E. Denning. Secure statistical databases with random sample queries. *ACM Trans. Database Syst.*, 5(3):291–315, Sept. 1980.

[8] D. E. Denning and P. J. Denning. The tracker: A threat to statistical database security. *ACM Trans. Database Syst.*, 4(1):76–96, Mar. 1979.

[9] J. Domingo-Ferrer and J. Soria-Comas. From t-closeness to differential privacy and vice versa in data anonymization. *Knowledge-Based Systems*, 74:151–158, 2015.

[10] C. Dwork. The promise of differential privacy: A tutorial on algorithmic techniques. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 1–2. IEEE.

[11] C. Dwork. Ask a better question, get a better answer a new approach to private data analysis. In T. Schwentick and D. Suciu, editors, *Database Theory*

ICDT 2007, number 4353 in Lecture Notes in Computer Science, pages 18–27. Springer Berlin Heidelberg, 2006.

[12] C. Dwork. Differential privacy. In *Automata, languages and programming*, pages 1–12. Springer, 2006.

[13] C. Dwork. Differential privacy: A survey of results. In M. Agrawal, D. Du, Z. Duan, and A. Li, editors, *Theory and Applications of Models of Computation*, number 4978 in Lecture Notes in Computer Science, pages 1–19. Springer Berlin Heidelberg, Jan. 2008.

[14] C. Dwork. The differential privacy frontier. In *Theory of cryptography*, pages 496–502. Springer, 2009.

[15] C. Dwork. Privacy against many arbitrary low-sensitivity queries. In *Proceeding of the International Congress of Mathematicians*, volume 901, pages 2634–2647, 2010.

[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, editors, *Theory of Cryptography*, number 3876 in Lecture Notes in Computer Science, pages 265–284. Springer Berlin Heidelberg, 2006.

[17] H. Ebadi, D. Sands, and G. Schneider. Differential privacy: Now it's getting personal. pages 69–81. ACM Press.

[18] IPython development team. The IPython Notebook. `http://ipython.org/notebook.html`. [Accessed 2015-05-25].

[19] H. Lam, G. Fung, and W. Lee. A Novel Method to Construct Taxonomy of Electrical Appliances Based on Load Signatures. 53(2):653–660.

[20] J. Lee and C. Clifton. How much is enough? Choosing ϵ for differential privacy. In X. Lai, J. Zhou, and H. Li, editors, *Information Security*, number 7001 in Lecture Notes in Computer Science, pages 325–340. Springer Berlin Heidelberg, 2011.

[21] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, volume 7, pages 106–115, 2007.

[22] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1), Mar. 2007.

[23] P. McDaniel and S. McLaughlin. Security and privacy challenges in the smart grid. *IEEE Security Privacy*, 7(3):75–77, May 2009.

[24] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 19–30. ACM, 2009.

[25] J. S. Milton and J. C. Arnold. *Introduction to Probability and Statistics: Principles and Applications for Engineering and the Computing Sciences*. McGraw-Hill, Inc., 4th edition.

[26] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin. Private memoirs of a smart meter. In *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*, pages 61–66. ACM, 2010.

[27] J. Momoh. *Smart grid: fundamentals of design and analysis*. John Wiley & Sons, Hoboken, 2012.

[28] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy, 2008. SP 2008*, pages 111–125, May 2008.

[29] Numpy developers. Numpy. `http://www.numpy.org/`. [Accessed 2015-05-25].

[30] S. N. Patel, T. Robertson, J. A. Kientz, M. S. Reynolds, and G. D. Abowd. *At the flick of a switch: Detecting and classifying unique electrical events on the residential power line*. Springer Berlin Heidelberg, Berlin, Germany, 2007.

[31] J. Reed and B. C. Pierce. Distance makes the types grow stronger: A calculus for differential privacy. In *Proceedings of the 15th ACM SIGPLAN International Conference on Functional Programming*, ICFP '10, pages 157–168, New York, NY, USA, 2010. ACM.

[32] I. Roy, S. T. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: Security and privacy for MapReduce. In *NSDI*, volume 10, pages 297–312, 2010.

[33] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, and S. Tarantola. *Global sensitivity analysis: the primer*. John Wiley & Sons, 2008.

[34] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report, SRI International, 1998.

[35] SciPy developers. Scipy.org. `http://www.scipy.org/`. [Accessed 2015-05-25].

[36] F. Siddiqui, S. Zeadally, C. Alcaraz, and S. Galvao. Smart grid privacy: Issues and solutions. In *2012 21st International Conference on Computer Communications and Networks (ICCCN)*, pages 1–5, July 2012.

[37] Statistics Sweden. Electricity consumption in one- and two dwelling buildings 2008. `http://www.scb.se/en_/Finding-statistics/Statistics-by-subject-area/Energy/Energy-supply-and-use/Energy-statistics-for-one--and-two-dwelling-buildings/Aktuell-Pong/2008A01/Electricity-consumption-in-one--and-two-dwelling-buildings-2008/`. [Accessed 2015-03-19].

[38] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[39] the pandas development team. Python data analysis library. `http://pandas.pydata.org/`. [Accessed 2015-05-25].

[40] U.S. Energy Information Administration (EIA). How much electricity does an american home use? - FAQ - U.S. Energy Information Administration (EIA). `http://www.eia.gov/tools/faqs/faq.cfm?id=97&t=3`. [Accessed 2015-05-07].

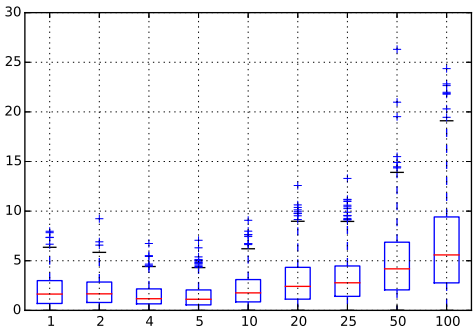
[41] L. Waye. Privacy integrated data stream queries. In *Proceedings of the 5th annual conference on Systems, programming, and applications: software for humanity. ACM*, 2014.

[42] G. Ács and C. Castelluccia. I have a DREAM! (DiffeRentially privatE smArt Metering). In T. Filler, T. Pevný, S. Craver, and A. Ker, editors, *Information Hiding*, number 6958 in Lecture Notes in Computer Science, pages 118–132. Springer Berlin Heidelberg.

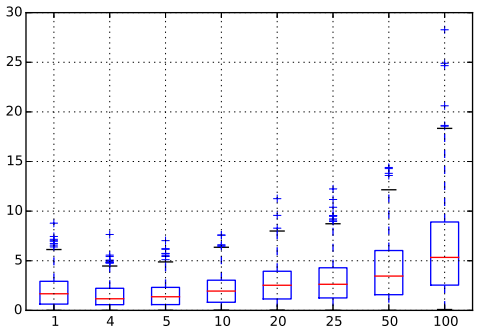
A

Box Plots

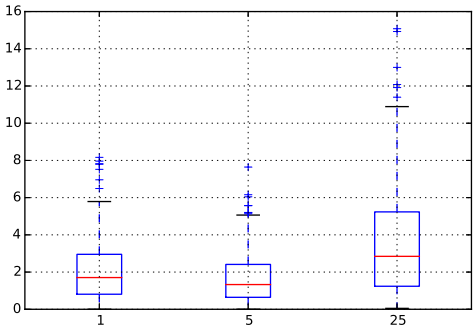
In this Appendix the detailed results that were left out of Section 8 are presented. The following figures show the results when 200 to 900 simulated households are used. Note for every figure that the x-axis represents the number of bins used, and the y-axis is the error in percentages introduced by applying differential privacy.



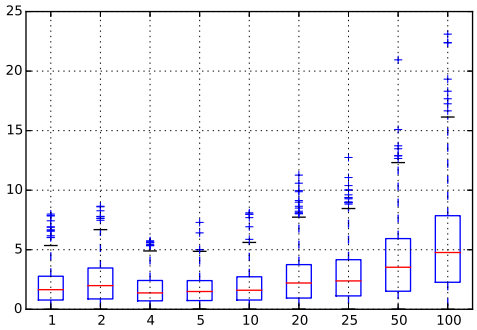
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning

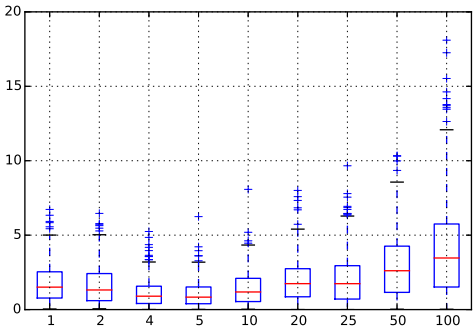


(c) Fine-grained edges partitioning

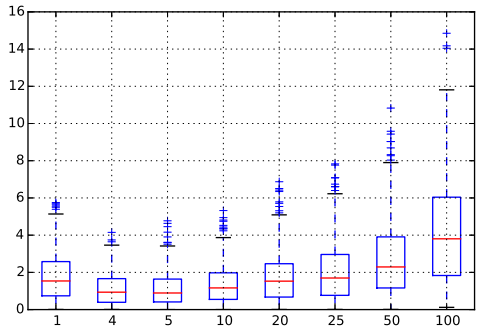


(d) Percentage partitioning

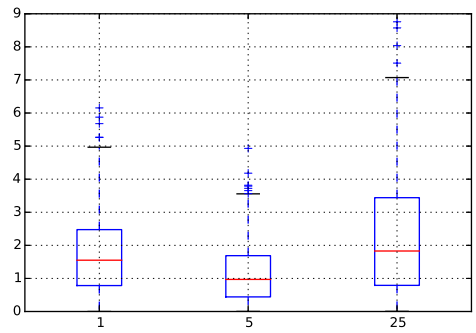
Figure A.1: Readings from 200 simulated households. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.



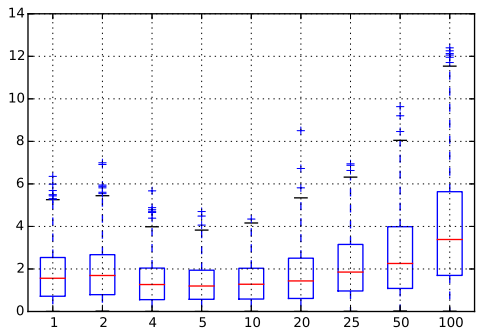
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning

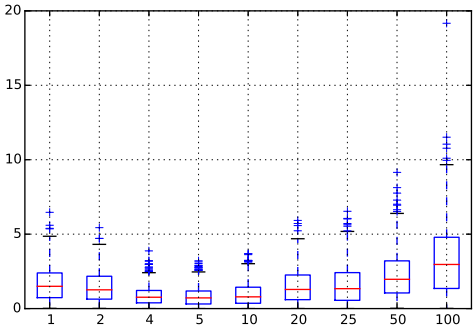


(c) Fine-grained edges partitioning

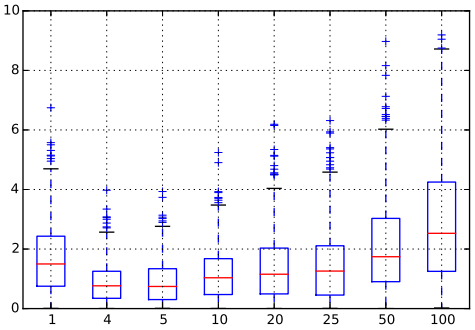


(d) Percentage partitioning

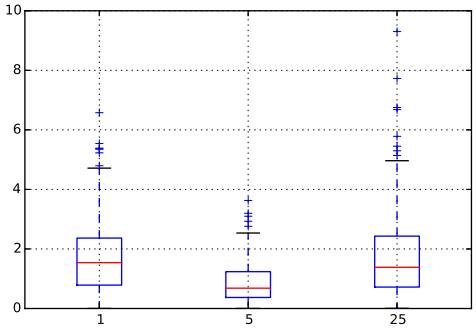
Figure A.2: Readings from 300 simulated households. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.



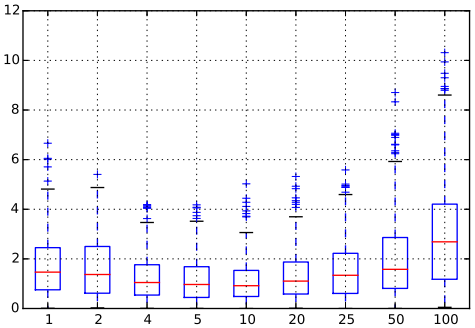
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning

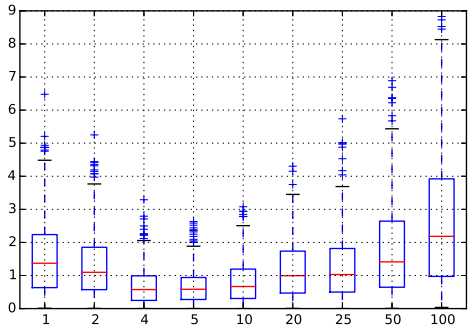


(c) Fine-grained edges partitioning

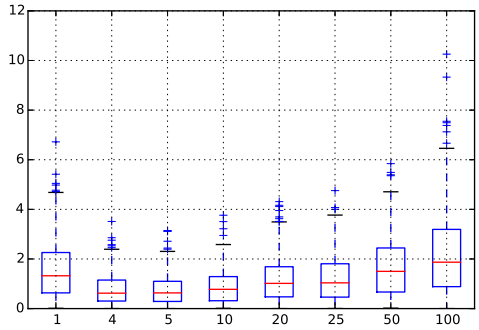


(d) Percentage partitioning

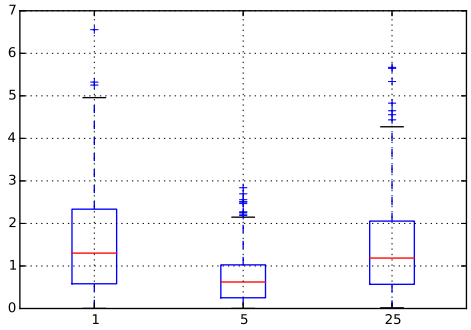
Figure A.3: Readings from 400 simulated households. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.



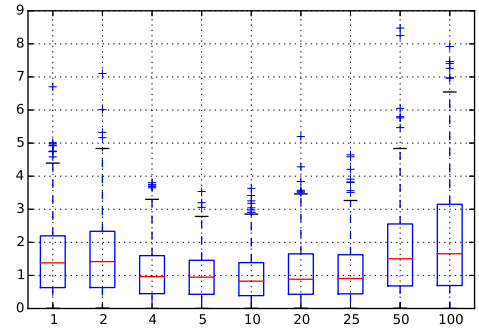
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning

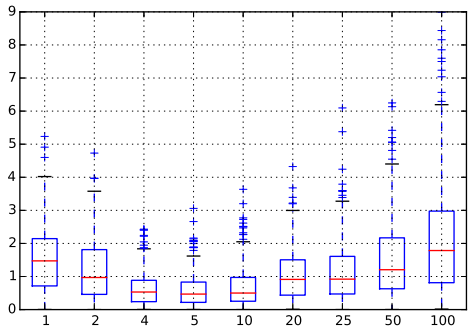


(c) Fine-grained edges partitioning

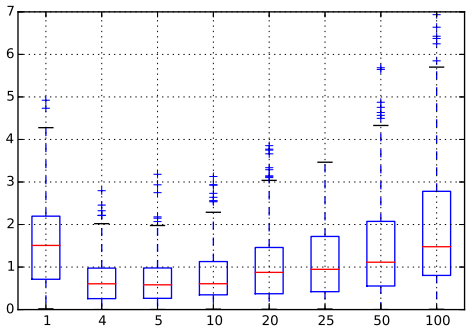


(d) Percentage partitioning

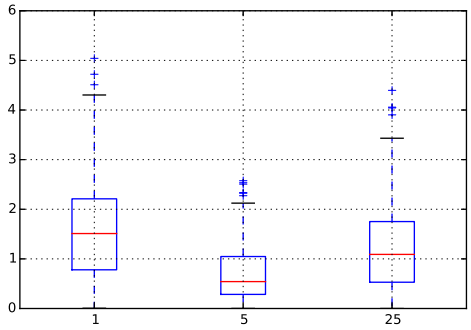
Figure A.4: Readings from 500 simulated households. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.



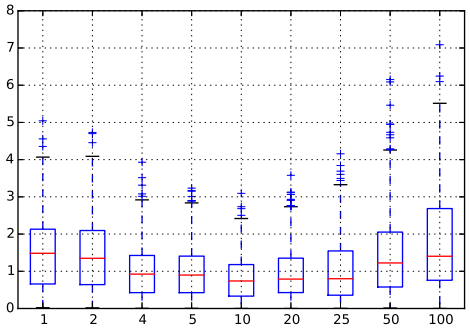
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning

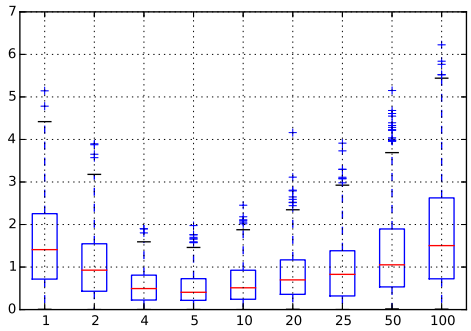


(c) Fine-grained edges partitioning

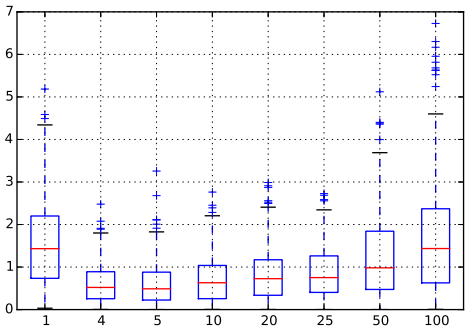


(d) Percentage partitioning

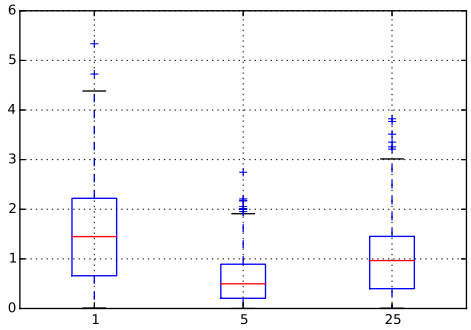
Figure A.5: Readings from 600 simulated households. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.



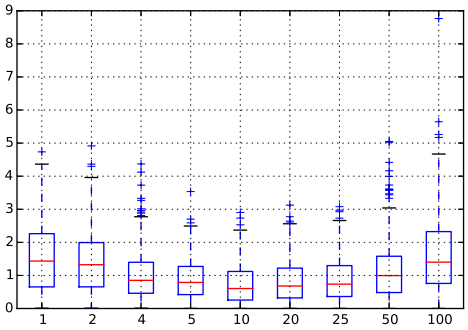
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning

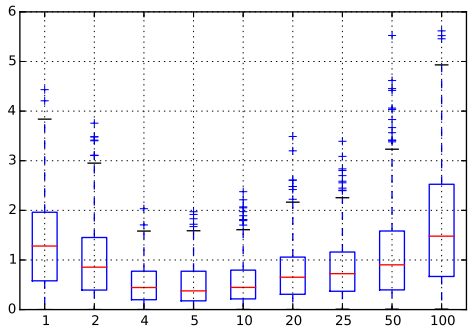


(c) Fine-grained edges partitioning

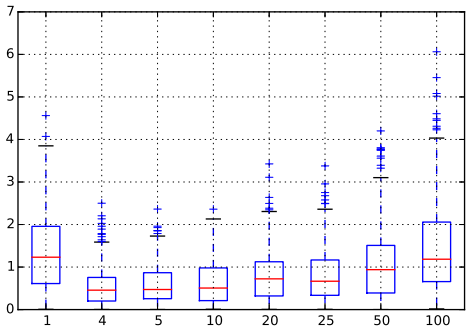


(d) Percentage partitioning

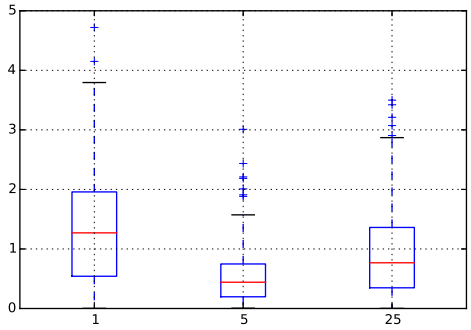
Figure A.6: Readings from 700 simulated households. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.



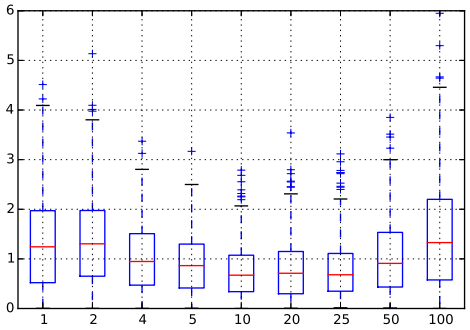
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning

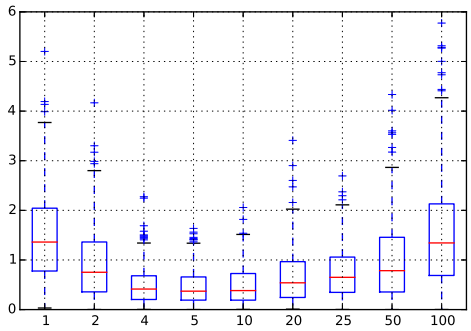


(c) Fine-grained edges partitioning

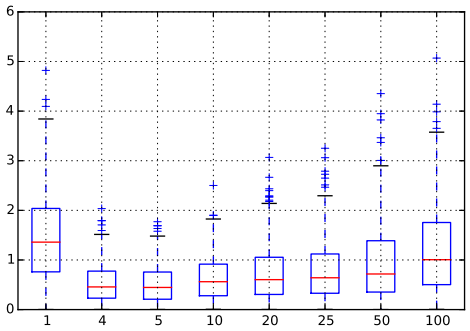


(d) Percentage partitioning

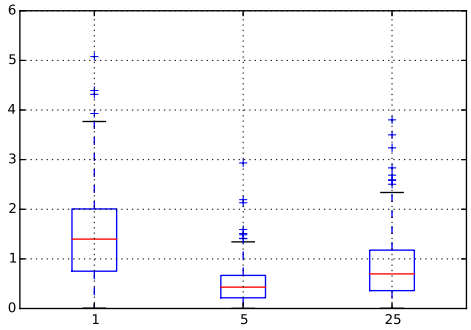
Figure A.7: Readings from 800 simulated households. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.



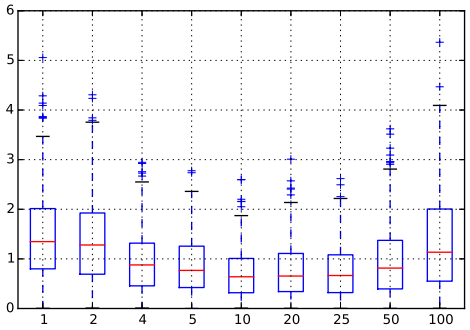
(a) Fine-grained partitioning



(b) Fine-grained mean partitioning



(c) Fine-grained edges partitioning



(d) Percentage partitioning

Figure A.8: Readings from 900 simulated households. The x-axis represents the number of bins used, and the y-axis is the error induced on the result.