Generating Private Contrastive Explanations with Differentially Private Classifiers and Foil Trees

W. J. Zirkzee

Technische Universiteit Delft

TU Delft

Generating Private Contrastive Explanations with Differentially Private Classifiers and Foil Trees

by

W. J. Zirkzee

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Thursday April 29, 2021 at 4:00 PM.

Student number:	4398858	
Project duration:	May 1, 2020 – April 29, 2021	
Thesis committee:	Prof. dr. ir. M. Neerincx,	TU Delft/TNO
	Prof. dr. S. Picek,	TU Delft
	Dr. B. Kamphorst,	TNO

TU Delft

Preface

This thesis describes the research and experiments I carried out in order to obtain an MSc in Computer Science, following the track of Data Science and Technologies, at the TU Delft. Throughout both my bachelor's and master's degree in Computer Science, I have been interested in two specific subfields of computer science: cryptography and machine learning. These fields provide complex and much-needed solutions, as both of these fields continue to play an evermore important role in our lives.

I am glad that during my thesis I was able to work with both of these fields, exploring a possible overlap of privacy-preserving machine learning and explainable artificial intelligence. While the progress and breakthroughs have been harder than were initially anticipated in many aspects, requiring to reshape the research questions and direction this work has taken. With all the difficult moments that come with concluding one's study and writing a thesis, especially during the Corona pandemic of 2020, in retrospect, I am truly exited about the many things I have learned over the course of this thesis.

I would like to thank all of my supervisors and mentors. First of all, Prof. Mark Neerincx, my thesis supervisor and for bringing me aboard his project at TNO which has lead to the combination of these two fields. Bart Kamphorst and Jasper van der Waa from TNO for their extensive help directing me throughout their respective fields of security and XAI. The numerous meeting we all have had discussing and thinking of new approaches or results, helping my clear my mind, or simply telling me to move on, are deeply appreciated. Stjepan Picek, for agreeing and taking the time to be a part of my thesis committee. And last but not least, my family and friends who have been there for the good and the bad moments throughout my entire studies.

I hope you enjoy your reading.

W. J. Zirkzee
Rotterdam, April 2021

Abstract

Due to the rapid growth of diseases like diabetes, machine-learning (ML) advanced *clinical decision support systems* (CDSS) that support doctors or patients in their care might prove to be a valuable asset. In the battle to make healthcare more accessible, adopting ML techniques could be supportive in increasing the overall effectiveness and efficiency of healthcare. However, the development of a CDSS is troubled by multiple problems.

First, in many cases high performing ML models are often black boxes or highly complex, obfuscating their inner workings. In contrast to the use of ML models in ethical fields, like healthcare, it is required to explain the decision-making process, and in many cases have a verifiable outcome, to gain trust with respect to the person influenced by the decision. Increasing the transparency of ML models and the explainability of predictions belong to the field of eXplainable Artificial Intelligence (XAI). Obtaining a good explanation that fulfills humans' expectation is more complex than presenting the influential parameters. More specifically, users expect a contrastive dialogue answering the question "Why X instead of Y?". A Foil Tree is an XAI method that focuses on constructing this contrastive explanation.

Secondly, over the years (medical) data and ML models have become valuable commodities. Because of this value and privacy concerns, there has been a strong increase in rules and regulations concerning its safekeeping, especially for medical data. As data owners and ML model developers often belong to different parties involved during CDSS development. Either party is reluctant or unable to freely share their information with their counterpart, as sharing prevents a privacy leak of the data or ML model. This inability to access all aspects that are required to train a model complicates the process of training a model with multiple stakeholders, called collaborative training. Privacy-preserving machine learning (PPML) aims to provide protocols to enable collaborative training by limiting the risk of privacy loss or even theft.

The use of XAI and PPML comes at a cost. Specifically, two trade-offs can be identified. Firstly, XAI often limits the complexity of a model or generates its explanations based on simplified aspects. This simplification negatively impacts the performance. Secondly, PPML requires the use of extra protocols or processing of the data, which negatively impacts the performance. The combination of these two techniques might lead to a third trade-off: the obfuscating effects of PPML (i.e. perturbation and encryption) interfere with the goal of obtaining more transparency for explanations.

In this work, the problems and solutions of both respective fields are analyzed to answer the question of whether XAI methods can be combined with PPML in such a way the ML-output can generate private explanations with a minimal reduction of performance. A state-of-the-art PPML technique, differential privacy (DP), seems to have a bright future through defining a privacy guarantee that can be scaled to the desired level of privacy. DP makes this guarantee through the addition of noise, providing ambiguity between the result of two neighboring databases, differing by precisely one entry. Through an evaluation of DP, the decision is made to use existing DP libraries to train DP-classifiers as the basis for a Foil Tree to be fitted to. In order to quantitatively evaluate the effect of using DP classifiers with a Foil Tree, high-level performance metrics such as accuracy, sensitivity, specificity, as well as Foil Tree specific metrics such as the amount of generated rules are measured and evaluated on a tabular data set predicting diabetes. The use of DP is accompanied by an extra level of uncertainty, hindering the trustworthiness and explainability, as a result of the stochastic noise. In order to assist in the interpretation of a DP result, the use of confidence intervals (CI) could be used to disclose the range of the noise and interpret the value. Seemingly the only data-dependent parameter to compute the noise is the sensitivity. With the goal of computing a private CI, the sensitivity must be made differentially private, such that it can be publicly shared. To this end two methods are hypothesized and evaluated with the goal of providing a DP CI. The first method computes the sensitivity's sensitivity (SS), which calculates the maximum change the sensitivity can have upon removal of a second record. The second method computes a private CI through DP histograms.

The results of the experiments showed that classifiers degrade as the available privacy budget decreases from $\varepsilon = 8$ to $\varepsilon = 0.1$. This degradation was an expected result, as in DP higher levels of privacy require higher levels of perturbation. Comparing the different privacy levels, low to medium privacy budgets ($2 < \varepsilon < 4$) still performed well. For these medium settings, Opacus provided the least amount of degradation. But using

low privacy values ($\varepsilon \leq 1$), Opacus also showed significant signs of deterioration. The Foil Tree displayed degradation comparable among the classifiers, in most settings. In some metrics, lower privacy budgets even showed an average improvement despite a significant gain in the variance. This unexpected improvement is contributed to classifiers being more heavily distorted. As a result, the classifier becomes easier to fit a tree too, as the balance of the predicted labels shifts such that a significant majority belongs to one class. The generated explanations vary significantly, even for low privacy settings. From these results, it is concluded that Foil Tree's explanations are extremely sensitive to small changes of, even low, differential privacy settings ($\varepsilon = 8$) and, therefore, such explanations are not suitable in the chosen private setting.

To obtain a query sensitivity that is private, such that it can be used in the generation of a private CI, noise is added to the sensitivity. The amount of noise is based on the SS. The results show that out of three basic queries, sum, max, and avg, that are evaluated, only avg resulted in a low enough SS such that spending $\varepsilon \leq 1$ suffices to obtain usable results. However, it is discovered that these approaches are based on an oversight. In many situations the sensitivity can only be computed in a private manner, and holds no entropy with regard to the data. Computing the sensitivity in this manner spends no epsilon, and therefore makes the hypothesized methods redundant. The use of CI remains interesting and although usage in simple one-query situations a CI can easily be interpreted. For more complex applications, as is the case for machine learning models, it's usage has to be further explored and evaluated.

There are limitations to the study. For example, the use of a small (medical) data-set consisting of 768 records, and the (likely) suboptimal application of DP due to its complexity, provide room for improvement. Future work concerning DP XAI could explore the more complex approach of developing a true DP XAI system, applying the noise only to the explanations leaving as much of the computation pure. The use of DP is still in an early stage and could see major developments regarding ease of application or performance over the years.

Contents

List of Acronyms vii

1 Introduction 1

2 Type 2 Diabetes Mellitus 4
2.1 Symptoms and Diagnosis . . . 4
2.2 Treatment . . . 4

3 Explainable Artificial Intelligence 6
3.1 Cognitive . . . 6
3.2 Increasing Transparency . . . 7
3.3 Foil Trees . . . 8

4 Privacy-Preserving Machine Learning 9
4.1 Attack models . . . 9
4.1.1 Re-identification . . . 9
4.1.2 Membership Inference . . . 10
4.1.3 Property Inference . . . 10
4.1.4 Model Inversion . . . 10
4.1.5 Model Extraction . . . 10
4.2 Federated Learning . . . 11
4.3 Cryptographic approaches . . . 11
4.3.1 Homomorphic Encryption . . . 11
4.3.2 Secure Multiparty Computation . . . 11
4.4 Perturbation Approaches . . . 12
4.4.1 De-identification . . . 12
4.4.2 Dimensional Reduction . . . 12
4.4.3 Differential Privacy . . . 12
4.5 Privacy-Preserving XAI . . . 13

5 Differential Privacy: Details 14
5.1 Privacy Loss Parameter: Epsilon . . . 15
5.2 Important Properties . . . 16
5.3 Mechanisms . . . 16
5.3.1 Randomized Response . . . 16
5.3.2 Laplace . . . 17
5.4 Architectures . . . 17
5.5 Relaxations . . . 18
5.5.1 (ε, δ)-Differential Privacy . . . 18
5.5.2 (α, ε)-Rényi Differential Privacy . . . 18
5.6 Trade-offs . . . 19
5.6.1 Applications . . . 19
5.6.2 Fairness & Bias . . . 19
5.7 Differentially Private XAI . . . 20

6 Methods 21
6.1 Models . . . 21
6.1.1 Reproducibility . . . 22
6.1.2 Epsilon . . . 22
6.2 Data . . . 22
6.2.1 Processing . . . 23

6.3 Metrics . . . 23
6.3.1 Classifier . . . 23
6.3.2 Foil Tree . . . 24
6.3.3 Rules . . . 24
6.3.4 Explanation . . . 24
6.4 Interpretation of DP mechanisms . . . 24
6.4.1 Private Sensitivity . . . 25
6.4.2 Perturbed Distribution . . . 25

7 Experiments and Results 27
7.1 Opacus . . . 27
7.1.1 Parameters . . . 27
7.2 DP Classifiers . . . 28
7.3 Effect on Foil Tree . . . 29
7.3.1 Overall . . . 29
7.3.2 Rules . . . 30
7.3.3 Explanations . . . 31
7.4 Interpretation . . . 31
7.4.1 Sensitivity's Sensitivity . . . 31
7.4.2 Oversights . . . 34

8 Discussions and Future Work 35
8.1 Discussion . . . 35
8.1.1 Effect on Private Explanations . . . 35
8.1.2 Interpretation of Private Explanations . . . 35
8.2 Conclusion . . . 36
8.3 Limitations and Unknowns . . . 37
8.3.1 Data . . . 37
8.3.2 Models . . . 37
8.3.3 Epsilon Bounds . . . 37
8.4 Future Work . . . 37

Bibliography 38

List of Acronyms

CDSS	Clinical Decision Support System
T2DM	Type 2 Diabetes Mellitus
ML	Machine Learning
XAI	eXplainable Artificial Intelligence
PPML	Privacy-Preserving Machine Learning
FL	Federated Learning
HE	Homomorphic Encryption
(S)MPC	(Secure) Multiparty Computation
DP	Differential Privacy
DPL	diffprivlib
LDP	Local Differential Privacy
GDP	Global Differential Privacy
RDP	Rényi Differential Privacy
CI	Confidence Interval
SS	Sensitivity's sensitivity
GNB	Gaussian Naive Bayes
LR	Logistic Regression
DP-SGD	Differentially Private Stochastic Gradient Descent)
PATE	Private Aggregation of Teacher Ensembles

1 Introduction

Type 2 Diabetes Mellitus (T2DM) is a disease that requires a high level of care [1, 2]. This chronic disease typically develops later in life in some people with an unhealthy lifestyle. It is characterized by high levels of blood sugar as a result of insulin resistance. Prolonged high levels of blood sugar can lead to several dangerous complications, e.g. reduced blood flow and hardening of blood vessels, which contribute to risk factors of diseases including stroke, nerve damage, kidney failure, and blindness. However, in many cases, T2DM can be managed through strict diet and exercise [3]. As this is a perpetual issue where multiple decisions a day have to be made, a significant part of the treatment comes down to self-management. In order to successfully self-manage, consistent education and training of the patient are a very important factor [4]. Many patients, especially in the early stages of the disease, are able to manage T2DM through diet and exercise alone, without any form of medication or insulin. Studies have shown patients are able to prevent the development of the disease or even return to a state of remission through lifestyle management [5–7].

T2DM requires intensive disease management [1, 2], and there could be a significant benefit through the use of a clinical decision support system (CDSS). A CDSS is not a replacement for clinicians but rather a health-information tool to aid and support a healthcare provider or patient. These systems are an active research subject as the need for faster and improved healthcare grows. These systems have surpassed the stage of early rule based AI like expert systems, and the latest developments try to incorporate current state-of-the-art machine learning (ML) techniques to become evermore useful and accurate. Advancements in CDSSs are of great social importance as CDSSs can reduce the cost and increase the quality of treatment [8]. For example, increasing the efficiency of healthcare practitioners and reducing patient waiting lists.

To illustrate the need for a CDSS for diabetes patients, imagine the following example; John Doe has not been feeling well lately. He is an older man in his late 50's. He is slightly overweight, and over the last weeks, he noticed he has been feeling unwell at times, paired with headaches, and urinating more frequently than usual. To get a checkup, he would like to see a specialist, but as doctors tend to have long waiting lists, this could take a few weeks. A preliminary diagnosis, possibly accompanied with some suggestions, can be given through an accurate CDSS that has been trained using a large data-set from multiple hospitals or even countries. If the CDSS returns a negative result or its suggestions solve the problem, John could feel a reduced need to continue scheduling an appointment with the specialist, freeing up time for other patients. On the other hand, if the CDSS's prediction is positive and John is eventually clinically diagnosed with T2DM, it will likely lead to a path of lifelong treatment plans. Instead of having regular appointments to discuss and adjust the plan to his individual needs, these doctor's appointments could be shortened, or more intermittently, by the usage of a CDSS. This will, in turn, allow the doctor to more efficiently allocate their time.

The example above contains two major stumbling blocks. First, why should John trust the outcome of the model? ML models have shown to be able to obtain precise and accurate results in disease classification [9], but for both ethical and confidence issues, patients and caregivers do not like to blindly trust a model's answer. To gain the trust of the users, the CDSS must explain its decisions. The growing complexity of ML models, for example, with the rise of Deep Neural Networks, is a likely reason for the surge in research towards explainable artificial intelligence (XAI), a research field with the goal of explaining the behavior of ML models. Generally, a drawback of XAI is that, generally, a gain in the transparency of a model corresponds to a

loss in performance due to the need for less-complex classification systems [10, 11]. Besides the technical aspect of gaining transparency, there is a cognitive aspect to be considered. To keep explanations informative, understandable, trustworthy, engaging, and useful, it needs to be customized to each user [2]. In a clinical setting, these users can be divided into two groups, healthcare professionals and patients. While there is an expected level of understanding from a healthcare professional, the patients' understanding can significantly vary based on previous education and experience with the disease. In general people expect an explanation that is *contrastive*, answering the question "Why *this* answer instead of *that* answer?". An XAI method whose objective is to generate contrastive explanations is Foil Tree[12]. Through the generation of local decision trees, a set of decision rules are identified to be used as the basis for such an explanation.

With respect to the stated example, this means that John wants to obtain an explanation similar to that a real doctor would be able to provide him: a dialogue explaining why John does or doesn't have diabetes in the form of a dialogue. Through the dialogue, extra aspects can be clarified if John inquires about it.

Secondly, there is a concern regarding the confidentiality of medical records, therefore such a system must adhere to strict privacy requirements. Especially medical records are accompanied with great concerns when it comes to sharing information [13]. In order to obtain the data that is needed for any AI system to train, the system should be able to guarantee the privacy of the user data to effectively put these concerns to rest. This can be achieved using privacy-preserving machine learning (PPML), a set of techniques used to enhance privacy with regard to both the data and the model, and collaboratively train ML models while guaranteeing one's privacy.

A common approach to improve privacy is data anonymization through providing *k-anonymity*. Application of k-anonymity generalizes the data-set to have at least k similar records [14], for example, by categorizing ages into various brackets. However, even this anonymized data can be combined in an attempt to re-identify the data at a later stage [15].

Currently, a relatively new privacy technique named differential privacy (DP) is considered the gold standard in privacy preservation. Differential privacy addresses the paradox of learning nothing about an individual while learning useful information about a population, as stated by one of its founders, Cynthia Dwork [16, 17]. However, DP is only a definition and not an easy-to-use implementation. The key technique in DP is injecting statistical noise to limit the amount an adversary is able to learn due to the publication of some information, defined as *information leakage*. What kind, how much, and where to add this noise are a few of the variables that need to be determined. Moreover, it guarantees interesting properties such as being closed under post-processing, meaning that without adding additional knowledge, any post-processing can not make it less private. Additionally, rather than a binary secure or insecure, it quantifies the amount of privacy loss. More specifically, it quantifies the probability an adversary can differentiate between the output of a query performed on two neighboring databases, databases that are precisely one entry apart. To obtain a higher level of privacy, a higher level of noise has to be added. As the added noise does distort the real values, this leads to a decrease in predictive accuracy in machine learning models.

With the use of XAI and PPML, two trade-offs regarding model performance can be identified, as the learning performance of a model is hindered by the following aspects:

1. Increasing the interpretability to explain the model or its predictions limit the complexity.
2. Increasing the privacy of data requires more noise.

However, the combination of XAI and PPML leads to a possible third trade-off regarding the compatibility of this combination of techniques. Intuitively privacy-preserving techniques try to obfuscate values to stop or reduce information leakage, where explainability tries to obtain higher transparency through, for example, exposing model parameters. E.g. an explanation could be based on similar examples, but giving those examples gives information about other records in the data-set, and is an information leak. Anonymization of the examples can limit the value of the explanation. Based on this example, a third trade-off can be defined:

3. The effects of PPML methods, such as the added noise of DP or encryption of the model, interfere with the goal of obtaining higher transparency to generate explanations.

These trade-offs require further research to be accurately determined whether there is a suitable combination of privacy-preservation techniques and explainability. To explore these trade-offs the following objectives have been formulated:

(i) **Can PPML and XAI be combined to form private explanations?**

(ii) **If so, to what extend does the use of PPML distort a Foil Tree?**

The structure of the rest of this work is as follows: first, an overview of T2DP is given in Chapter 2. Following in Chapter 3 is a literature study into both the technical and cognitive aspects of XAI of obtaining a good explanation. Chapter 4 provides an overview of the set of techniques that make up PPML. In Chapter 5 a complete overview of the state-of-the-art techniques that belong to differential privacy is provided. The methods of the experiments are laid out in Chapter 6, followed by the execution of the experiments in Chapter 7. Finally, in Chapter 8 conclusions are drawn from the obtained results and discuss the limitations and future work.

2

Type 2 Diabetes Mellitus

Diabetes mellitus, commonly known as diabetes, is an increasingly common chronic disease where the production or effect of insulin is reduced or completely inhibited. Insulin is a hormone produced by the pancreas for regulating the level of blood sugar by promoting glucose uptake into its cells. While a variety of causes are known, most cases of diabetes can be categorized into two distinct mechanisms leading to a disruption in the pathway of insulin, called type 1 and type 2. The former can be characterized by a deficiency of insulin, as the result of an autoimmune condition where insulin, which is normally produced by the body's pancreas β-cells. This type often has a quick childhood-onset, resulting in a diagnosis between the age of 4 and 14. Whereas the latter is a result of the body's resistance to insulin. While insulin seems to correctly bind to the insulin receptor, for unknown reasons, it does not result in the proper reaction activation of the protein to allow for glucose uptake [18].

Diabetes is a major problem all around the world, especially since the number of patients with diabetes is increasing at a rapid pace. The World Health Organization estimates there are 422 million adult diabetes patients and 1.6 million diabetes-related deaths in 2016, a significant increase compared to the 0.94 million deaths in 2000. Of all these adult diabetes patients, the vast majority, more than 90%, is accounted for by type 2 Diabetes Mellitus (T2DM) [3].

2.1. Symptoms and Diagnosis

The frequent symptoms of diabetes include, but are not limited to: urinating more than normal (*polyuria*), feeling thirsty (*polydipsia*) or hungry (*polyphagia*) all the time, fatigue, blurred vision, and slow healing of wounds or cuts [19]. During the early stages of the disease, symptoms are often mild, or patients can even be completely asymptomatic, making an early diagnosis difficult. As the disease progresses, the symptoms worsen.

The development of T2DM has a relation with numerous risk factors such as: age, gender, racial heritage, being overweight, family history, high blood pressure (*hypertension*), high triglyceride levels, and low HDL cholesterol[20, 21].

Due to the progressive nature of this disease, a precursor of the disease called *prediabetes* can often be detected before the clinical onset of diabetes. The main methods of diagnosis are a *random plasma glucose* (RPG), *fasting plasma glucose* (FPG), 2-hour plasma glucose (2h PG) after 75g oral glucose tolerance test (OGTT), or an HbA1C (hemoglobin) test [22]. The advantage of an HbA1C test is there is no need for an 8 hour fast or a 2 hour long sampling period and greater day-to-day stability of measurements. But this comes with lower sensitivity (increased false negatives), a higher cost, and variations among ethnicities. Section 2.1 gives an overview of the test outcomes to be positively diagnosed with either diabetes or prediabetes. For cases diagnosed with FPG, 2h-PH, or HbA1C, a second test, ideally using the same test the following day, for confirmation is required [23].

2.2. Treatment

Managing T2DM is difficult consists of finding a lifelong treatment plan that is tailored to the needs of each individual patient. Treatment plans can consist of monitoring blood sugar levels and finding the right combi-

Test	Diabetes	Prediabetes
FPG	126 mg/dL	100 mg/dL
2h-PG 75g OGTT	200 mg/dL	140 mg/dL
HbA1C	6.5%	5.7%
RPG	200 mg/dL	

Table 2.1: An overview of the diagnostic results to clinically diagnose a patient with diabetes, or prediabetes.

nation of diet, exercise, oral medication, or supplemental insulin. As the disease progresses, treatment plans will have to be continuously adjusted. Many patients, especially in the early stages of the disease, are able to manage T2DM through diet and exercise alone, without any form of medication (or insulin). Studies have shown that some patients are even able to prevent further development of the disease [7] or even return to a state of remission [5], through lifestyle management.

As diabetes plans call for around-the-clock care, an important aspect is to educate the patients. Diabetes self-management education (DSME) teaches the patients how to manage their T2DM. This is reduces the need for patient care by a healthcare professional [2, 23], improves their condition [4], and increases the cost-effectiveness of the treatments[24, 25]. All of these reasons allow for the healthcare professionals' time and budget to be spent wisely on first-line treatment of the rapidly growing diabetes patients.

3

Explainable Artificial Intelligence

In recent years there has been a noticeable surge in the field of AI, as its increasing performance is unrivaled by traditional solutions. Early AI systems' results were easily understandable, but as the development of AI systems continued, the resulting systems' inner workings, like Deep Neural Networks (DNNs), have become increasingly opaque. These more-and-more advanced neural networks allowed the systems to be able to be more accurate by finding patterns in higher-dimensional parameter spaces. While the increasing opaqueness of the model is not an issue in all applications, in some fields where AI decisions can severely impact human lives, e.g. health care or law enforcement, there is an intrinsic need for understanding these inner workings. Understanding the decisions a model has made helps to justify or trust its outcome, and prevent erroneous behavior like discrimination. In 2018 this intrinsic need has been fortified by the European union's General Data Protection Regulation (GDPR), a regulation with regard to data protection and privacy. One of the laws declared in the GDPR is the "right to explanation", which provides the right to obtain an explanation for AI decisions that significantly impact one's life [26].

These needs and regulations have likely contributed to the rise in research and publications in the field of eXplainable Artificial Intelligence (XAI), a research field focused on improving the explainability of AI models. This field typically consists of developing techniques that help to increase the transparency of a models underlying abstractions [10]. However, how can the results of XAI be presented such that they compose an understandable explanation? This question is concerned with the social cognitive theory of how to represent these aspects in an approach humans can interpret well [27, 28].

3.1. Cognitive

The cognitive aspect of an explanation is the field of representing the obtained decision factors in an understandable manner. This is a challenging question as it is difficult to define what constitutes a good explanation and when an explanation is "good enough". Explanations can be given in many forms and can consist of a combination of graphs, images, animations, and natural language generation [29, 30]. Additionally, explanations are usually different depending on who is asking the question and what the goal of the explanation is. Much of XAI focuses on explicitly explaining decisions of how it got to this result. Commonly the researcher simply decides how good an explanation is, based on his intuition. Miller [27] evaluated what makes an explanation good, and shows that humans expect a *contextual* explanations. The most important factors of a contextual explanation are that an explanation should be *contrastive*, *biased*, *social*, and *not excessively probabilistic*. A contrastive explanation means the answers should be formulated such that it answers the question "Why X instead of Y?". In the process of answering this question, there can be many factors. But the explanation should be constructed selectively or *biased* as not every minute aspect must be explained, but rather a select few causes that have the most impact. The probabilities for these causes, while important for many reasons besides a selective nature, only matter when accompanied by an underlying explanation or cause for the generalization. Lastly, an explanation is social and should be seen as a human-agent interaction (HAI), a conversation that accounts for the knowledge and beliefs of the target. Rather than a simple answer, it should try to close the gap between the questioner's knowledge base and the one required to understand the outcome. Through HAI a natural dialogue should be formed where a user model can be formed to infer the questioner's knowledge or answer a user's subsequent questions to clarify further.

3.2. Increasing Transparency

The technical aspect of increasing transparency is still a very broad field. An initial separation can be made on the type of model. Models can have an inherent degree of transparency, these kinds of models are named *white-box* models. On the other hand, a model can be completely opaque such that only the input and output of a system can be considered, a *black-box* model.

White-box models (e.g. decision trees, K-nearest neighbors) follow or are able to provide a set of reasons as to how it obtained the final result. Figure 3.1 illustrates an method of explanation for six common white-box models. One of the most straightforward methods is a decision tree. Decision trees are a tree-like structure, where each node represents a choice and its outcome and eventually leads to a single leaf determining the outcome. K-nearest neighbors is an example-based approach as it looks for the k examples closest (read: most similar) to the data point that are to be predicted. Using these k examples, a voting strategy defines the most likely prediction.

Even though white-box models are transparent by design, this transparency is achieved at a variety of levels. For example, training a decision tree on a highly complex data-set might result in more opaque transparency as the depth of the decision tree can grow to an inexplicable large tree. Often such factors are limited by setting *hyperparameters* to limit the number of abstractions. In the case of a decision tree, this includes parameters such as maximum depth or maximum children per node in order to reduce the size of the final tree to a more understandable size. Limiting machine learning models does usually mean they are unable to obtain the best fit, as certain abstractions can no longer be accurately captured in the simplified version, therefore reducing the performance.

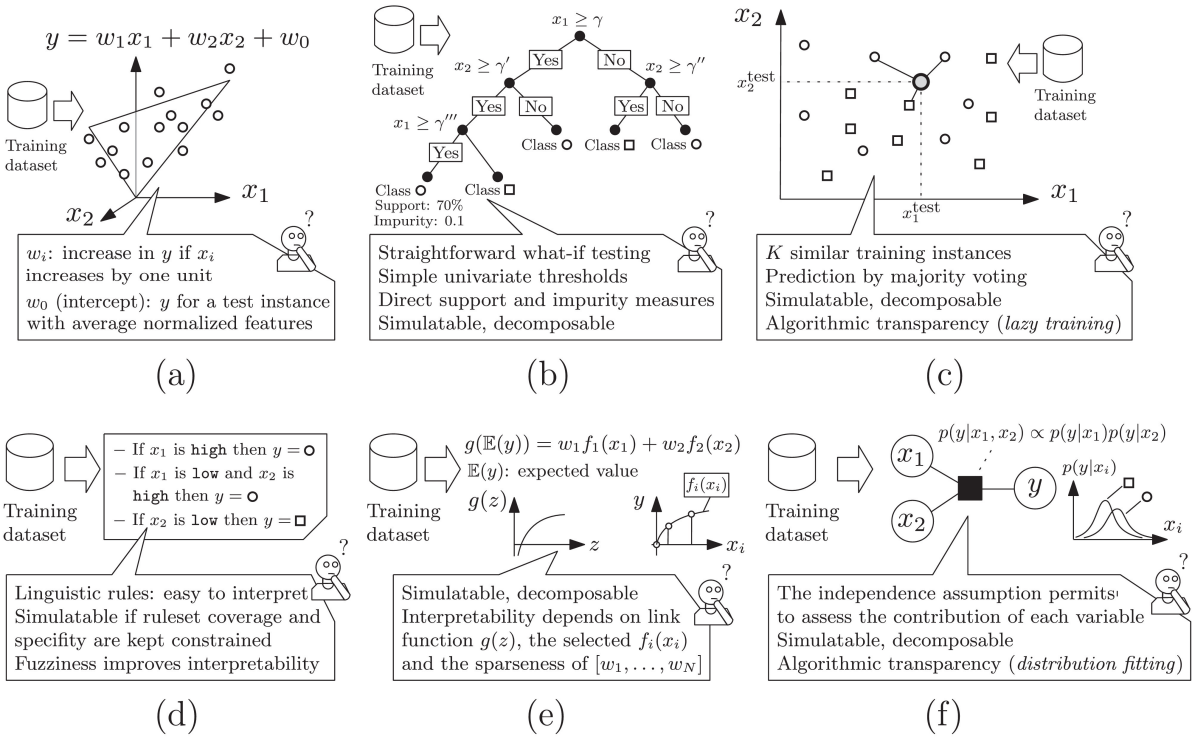


Figure 3.1: Graphical illustration of the levels of transparency of different ML models considered in this overview: (a) Linear regression; (b) Decision trees; (c) K-Nearest Neighbors; (d) Rule-based Learners; (e) Generalized Additive Models; (f) Bayesian Models. From [10]

Black-box models require separate post-hoc techniques to be applied to the system to derive the decisions it has made [31]. These post-hoc techniques can be split into two categories, model agnostic and model specific. The model agnostic techniques can be used with any model to extract information, including white-box models, but in general, there is less of a need for such models. The model agnostic techniques are based on *interpretable explanators*. These are methods to explain either the entire model or a single prediction. In order to explain entire models, it might be reduced to a more understandable model such as a decision tree or a set of decision rules. Another common approach for both model and prediction is to use *feature importance*, a number estimating the weight and magnitude of the used features. Image classifications commonly use

a *saliency mask*, a sort of visual feature importance, by highlighting the aspects which were important in determining its classification through a deep neural net.

Two popular model-agnostic XAI methods are LIME (Local Interpretable Model-Agnostic Explanation) [32] and SHAP (SHapley Additive exPlanations) [33]. SHAP evaluates the Shapley value for the decision, which can be thought of as local feature importance around the decision. LIME's full name shows they value interpretable results rather than just values. While both offer explanations for the current outcome, they do not support a method of obtaining a *contrastive* explanation. One method specifically designed to deduce explanations in a contrastive form is a Foil Tree [12].

3.3. Foil Trees

Foil Tree is a model-agnostic XAI method, increasing transparency of predictions and also addresses the cognitive aspect of generating a suitable, contextual explanation. A Foil Tree creates a local evaluation of a model around the prediction that is to be explained. More precisely, training a Foil Tree consists of 8 steps (Figure 3.2).

First, the normal classifier that requires additional explanation is used to obtain a prediction, the fact, of a data point. In step two, another class, the foil, is selected as the target of the explanation. The training data for the Foil Tree can be obtained by random sampling from a data-set or be generated. Subsequently, the Foil Tree is trained on these sampled or generated data points. Based on the data-set, many distinct decision trees are trained (a *random forests*) as it tries to find decision boundaries that distinguish the foil class. This approach is called an *one-versus-all* decision tree. In the tree resulting from training, a fact and foil leaf are located. Using the path in the tree between the two nodes, a variety of strategies (e.g. shortest path) can be used to find a path between the two different decision leaves (fact and foil). This path is the basis of constructing a textual contrastive explanation.

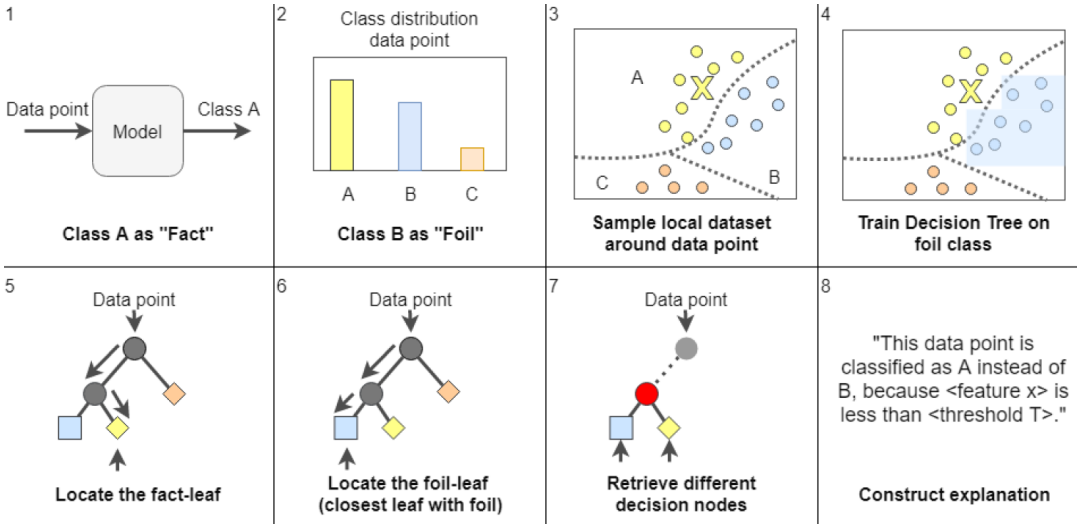


Figure 3.2: Graphical depiction of the steps required to train a Foil Tree and generate a contrastive explanation. From [12]

4

Privacy-Preserving Machine Learning

To train a machine learning model and effectively use it two main requirements can be identified. First, there is a need for data to train the model. Data is usually obtained through institutions that specialize in aggregating information. In medical settings, the data often comes from hospitals or healthcare institutions, where patients need to give explicit consent in order for their data to be used for research purposes. A rich history of medical data-leaks, which make up as much as 70% of all reported breaches [34], are likely to make many people hesitant to agree to such a request. This factor needs to be taken into consideration. To increase the willingness of people to share their data, there needs to be assured that data is handled with care and remains private.

Second, the trained model or its architecture also hold value as its training requires investments in the form of time, computational power, and research to determine the right parameters and architecture for a problem. But in order to use a model it needs to be released or made accessible. A common method to do this is through a client-server approach, like an API, giving a user *black-box* access. However, either of these methods leaves the model vulnerable to attacks in which both the model or data can be stolen through various attacks.

As both the data and the model are valuable, parties are reluctant to share their respective part, complicating the training process of a CDSS. In order to protect privacy and ownership during a collaborative effort to train a machine learning model, federated learning (FL) and privacy-preserving machine learning (PPML) aim to provide a solution. FL [35] is a remote execution technique, which migrates some of the risks by transferring the model to the data owner, rather than moving the data to the model owner. An important notion to be made is that FL mitigates or transfers the risk, however, it is not capable of providing any security or privacy guarantees [36]. PPML [37, 38] focuses on the ability to release data or to collaboratively train a model while preserving privacy. These techniques can roughly be divided into two categories, *cryptographic* or *perturbation* based. Cryptographic approaches rely on cryptographic protocols to encrypt the data that retain specific properties that are required to perform training or testing on encrypted data, thus preserving the value and anonymity. Perturbation approaches are based on techniques based on altering the data or model to restrict or hide sensitive information. While each individual group of techniques only provides a partial privacy setting, a combination of the approaches given above is able to provide a complete privacy-preserving approach to machine learning [39].

4.1. Attack models

There is a wide collection of known attacks on machine learning models with the goal to extract information that was not intended to be released. While there are many specific attacks, five of the most common attacks are *re-identification*, *membership inference*, *property inference*, *model inversion*, and *model extraction*. A brief introduction is provided for each of these attacks to illustrate their approaches and goals.

4.1.1. Re-identification

Data re-identification is the practice of re-identifying data records that have previously been anonymized through redacting or removing identifiable information in order to maintain privacy. In many cases the data is combined with other (anonymized) data-sets to rediscover (or reduce the uncertainty of) identifiable fields.

Through the recovered values the anonymization process is reversed to a point the data-set becomes re-identifiable to its human source [40, 41]. Identifying the source based on a few data points is easier than most would expect. Once only a few simple demographics are obtained, a combination (e.g. ZIP code, gender, date of birth) quickly becomes a unique identifier that can be used to re-identify the data. The combination of ZIP code, gender, and date of birth already satisfies to uniquely identify 87% of the US population [42].

Two well-known re-identification attacks stem from a challenge started by the streaming service Netflix [43] and a data leak from an online service provider AOL [44]. The companies released an anonymized data-set of movie ratings and search histories, respectively, replacing or removing usernames, IP-addresses, etc. By matching Netflix rankings to those made on IMDb[1], an online movie rating and review database, it has been shown how remarkably simple it is to re-identify data. Obtaining only two movie ratings on a 1-5 scale and the approximate date of rating is sufficient to uniquely identify over 40% (+- 14 days), up to 68% (+- 3 days). Increasing the number of movie rankings to 8, of which 2 are completely false, to 99%, even with a 14-day error [43].

4.1.2. Membership Inference

A membership inference attack is when the adversaries' goal is to seek and infer whether a data instance or a specific person was part of the used training data. The risk of being identified as part of the training data lies within the specification of the data. E.g. if a persons participation is inferred from a study among a group of HIV-positive subjects, the inferred person must be one of the HIV-positive subjects, and their privacy has clearly been compromised. This attack is possible in both a white- or black-box setting [45, 46]. In a white-box setting, the adversary knows all model parameters, but even in a black-box scenario where all parameters are unknown and only the classifier's decision or class losses are available through remote access (e.g. API) membership inference is possible.

4.1.3. Property Inference

An adversary can perform a property inference attack to infer properties about the entire data-set. If the specified data does not specifically imply or specify a desirable property that an adversary wants to learn (e.g. medical condition), one can inference properties about the data-set [47]. Once a property has been inferred, using the previously explained membership inference attack can implicate a specific person to be present in the training data. Through this the target can be exposed through the combination of a property inference and membership inference attack.

4.1.4. Model Inversion

In a model inversion attack, an adversary tries to reconstruct (a part of) the training data [48]. Similar to membership inference, model inversion can be defined under both white- and black-box settings.

An example applying a model inversion attack followed by a re-identification attack stems from an internal experiment based on the 2010 U.S. census, a nation-wide population count held every 10 years to determine the population distribution, among other statistics such as age, sex, ethnicity, etc. to determine the allocation of federal funds and apportion the chairs of the U.S. House of Representatives. An experiment to reconstruct and re-identify (Section 4.1) data from the 2010 census' statistics, 71% was partly reconstructed, 39% fully reconstructed, of which 17% of the US population or 52 million individuals were accurately re-identified. These results are significantly worse than previous estimates of being able to reconstruct 0.017% of which only 22% accurately [49, 50].

4.1.5. Model Extraction

While the previous attacks focused on data, this attack focuses on the model. In a model extraction attack an adversary intents to reverse-engineer a *victim* or *target* model. Through black-box access to a victim model, the adversary tries to learn a substitute model that replicates the outcome of the victim model [51, 52]. Using such a substitute model the imposed query limits set by the original owners can be avoided to continue with other attacks through the substitute model or to avoid the associated costs of an offered ML-as-a-service.

[1] https://www.imdb.com/

4.2. Federated Learning

An opposing view of the centralized approach of training models is based on remote execution. One technique that emerged in 2016 is federated learning (FL) [35, 53]. Using FL the data remains on a remote location and trains a global model in a decentralized. The data is kept by its owner and never leaves the device. Device is a broad term which contains the scope from a data center to a single mobile phone. Instead of transferring the data to a central location to train a model, each device trains a separate model and shares the learned parameters. A central authority aggregates all received parameters to compute a global model.

But FL is not strong enough to make a guarantee on privacy which holds up the standards set by the GDPR [36], as even the release of learned model parameters of a sub-model is sensitive enough for personal information to be deduced [39].

4.3. Cryptographic approaches

Cryptographic approaches address the case of ownership. Through encoding, data loses its value until decoded. *Homomorphic encryption* allows performing calculations on encrypted data to result in a correct and encrypted result. This ensures the privacy is kept throughout the computation. Another promising technique is *Secure Multiparty Computation* which allows for a function to be computed or value to be shared, without every revealing the input. With these cryptographic techniques, it is possible to train complete ML models without ever seeing the (unencrypted) data. Training on encrypted data results in encrypted model parameters and predictions, but to interpret the predictions the value needs to be decrypted and can therefore not guarantee output privacy.

4.3.1. Homomorphic Encryption

Homomorphic encryption (HE) exploits the homomorphic property that is found in some encryption schemes. This property allows functions to be performed on ciphertexts and results in a valid ciphertext [54].

$$f(a) + f(b) = f(a + b)$$

Two popular examples are ElGamal [55] and Pallier [56] cryptosystem. Since only certain mathematical operations conserve the homomorphic property, it is required the application only uses these allowed operations. There are certain degrees of HE allowing different sets of operations: *partially*, *somewhat*, *fully* homomorphic encryption support an increasing amount of functions, but are in a similar fashion increasingly difficult to realize. Several ML implementations, including deep networks, have already been realized [57, 58].

But homomorphic encryption also has shortcomings. Specific disadvantages can differ between implementations, but two important ones are the inherent lack of verifiable computation due to the homomorphic property and its speed. While improvements are being made, current implementations are incredibly slow and unusable in real-world applications.

4.3.2. Secure Multiparty Computation

Secure multiparty computation ((S)MPC, also SMC) is a field that safeguards the privacy of two or more parties' inputs to compute a joint result. A sub-problem of MPC is the field with exactly two parties, secure two-party computation (2PC) [59, 60]. Over the years many different protocols have been developed for 2PC and MPC, adhering to different privacy guarantees.

2PC allows two parties to compute the result of their input through *garbled circuits* [59, 60], or *homomorphic encryption* (Section 4.3.1). Through garbled circuits, one party, the *garbler*, provides the circuit to the other party, the *evaluator*. Through cryptographic protocols, the evaluator evaluates this circuit to learn the output without learning the other person's input. Optionally, the evaluator can share the resulting output with the garbler.

MPC defines multiple privacy guarantees based on the type of adversaries that are part of the computation. There are two active adversaries willing to deviate from the protocol and cheat, and one passive adversary that always follows protocol. In order from strongest to weakest they are named *malicious*, *covert*, and *semi-honest* [61, 62]. Malicious adversaries knowingly deviate from the agreed protocol to manipulate the output or find out what the other parties' shares are. A covert adversary is similar to a malicious, except for being limited in their options as they are not willing to be caught cheating. In a *semi-honest* setting, the adversary follows protocol but might also try to reveal other parties' shares without cheating.

An example of an MPC protocol is secret sharing. Distributing a value in parts, no party knows the real

value until combined. One such implementation is based on *Shamir secret sharing*, a distributed protocol for shared ownership of a secret. This is an algorithm where a secret is divided into n separate *shares*. Each share holds no value and can be distributed to independent owners. To reconstruct the original secret, a previously defined *threshold* t of the *shares*, $t \leq n$, are required to reconstruct the original secret [62].

The biggest challenges of MPC are, similarly to HE, concerning its speed. MPC requires connectivity and communication between all parties, causing a significant overhead in cost [61]. The last few years have found significant speedups for 2PC [63], but progress regarding MPC remains limited.

4.4. Perturbation Approaches

To obtain output privacy perturbation approaches are required. Perturbation approaches modify the data to mitigate the risk of both an intentional or an accidental release of data. Over the years, different approaches have been constructed to try and achieve such privacy. The most common approach is de-identification of the data through *k-anonymity*, but over the years this approach can no longer be thought of as private. A state-of-the-art approach is *differential privacy*, providing a quantifiable measure of privacy.

4.4.1. De-identification

One of the most common approaches is de-identification accomplished through *k-anonymity*. K-anonymity is a concept that generalizes the personally identifiable columns until there are k identical records through suppressing and generalizing the values [14]. For example, two records containing an age (21, 26) can be generalized to a bucket (e.g. 20 to 30) to obtain 2-anonymity.

In some cases, the data-set has little diversity in the sensitive attributes and can lead to generalized records with the same target category. This weakness can be addressed through the requirement of *l-diversity*, each set of k identifiable records must have at least l distinct sensitive outcomes [64].

Finally, if data is l-diverse, but all l categories have negative implications (e.g. various distinct diseases), δ-presence is a metric that evaluates the chance the risk is identifiable based on public data [65]. More precisely, say there is a data-set with x anonymized records with the identifiable columns age and ZIP code, that are linked to (different) sensitive outcome. Then if the real world contains y people that fall within the same anonymized categories based on age and ZIP code as x, then their identifiable risk is $\frac{x}{y}$.

4.4.2. Dimensional Reduction

A dimensional reduction is the transformation of projecting $(n \times l)$-dimensional data on a $(m \times l)$-dimensional subspace, where $n < m$, or even $n \ll m$. This technique is most common in fields that have a large number of variables, such as speech or video recognition and other fields of signal processing, picking the essential (combination of) features that define most of the variance within the data-set [66]. Another application is that data can easier be visualized in two or three dimensions so that a better interpretation of the data can be gained. Various well-known dimensional reduction techniques (e.g. t-SNE, PCA) have also been suggested to be used as the basis as a privacy-preserving method, preventing a strong adversary from reconstructing the original data [67, 68].

4.4.3. Differential Privacy

The original definition by Dwork [17], introduces ε-differential privacy. Differential privacy (DP) is a definition that is able to guarantee the level of privacy loss on algorithms. It is defined based on the distance between two data-sets that differ on a single row of data. DP offers a measurable, mathematical guarantee on the likelihood an adversary can detect the removal of a single entry. From not being able to detect this missing entry, it follows that the output is not dependent on this entry and therefore does not leak privacy. This is achieved through the addition of stochastic noise, based on the privacy parameter ε, and the maximum effect a single entry has on the algorithm's output, called *sensitivity*. Differential privacy is a state-of-the-art technique and still has many challenges and unknown aspects, and is still actively researched. One of the challenges with DP is that level of privacy defines the required amount of noise, influencing the quality and capability of the trained model. Another challenge is that DP is just a definition and optimal implementations depend on the use case and other (privacy) techniques it is combined with. At the current state of development, this requires each implementation to be thoroughly researched before being able to develop an implementation.

4.5. Privacy-Preserving XAI

There are many different techniques in PPML that can be applied depending on the attacks one needs to protect against. An ML system that needs to be protected from attacks on both the data and the model, it is needed to use a combination of both cryptography and perturbation approaches.

The cryptographic approaches encrypt the model in a way that without authorization the becomes unusable. As encryption has already become an important aspect the most people's everyday life, guaranteeing its effectiveness and its effect regarding privacy are relatively easy to explain. With the proper authorization, the model works as intended without any difference from a non-private model. This also means the output of such a model remains non-private.

To obtain output privacy perturbation methods are required. From all the described perturbation methods differential privacy currently gives the strongest guarantee of privacy. Like all other perturbation methods, using differential privacy imposes changes to the model or its output through the perturbations from which the privacy stems. As the output of a model will change, so will the explanations of an XAI system. Explaining why and how differential privacy exactly provides privacy can be non-trivial and requires additional work to understand and explain.

As the cryptographic approaches do not alter the resulting outcome, perturbation approaches have the most interesting effect on an explanation. Differential privacy is the strongest candidate to provide a quantifiable level of output privacy. In order to decide what the best approach is in order to use differential privacy with the goal of obtaining privacy-preserving XAI and private explanations, differential privacy is researched in-depth in the following chapter.

5

Differential Privacy: Details

Differential privacy (DP), introduced by Dwork [16, 17], is a strong and formal definition of privacy loss. The formal definition provides an upper bound on the probability anyone can differentiate between two *neighboring databases*, databases that differ by exactly one record.

A clear example can be given with the use of histograms. A study with diabetic patients using a histogram binned by, e.g., age, after removing one record from the database will clearly indicate to which bin the record belonged (Figure 5.1). This information leak can be used by an adversary in an attempt to re-identify a person in the data-set. Differential privacy hides this difference through the addition of noise. If each bin of the histogram is perturbed, each computation of this histogram is different and does not show the exact value. This difference increases uncertainty and makes it harder or impossible to observe the statistics of the removed records.

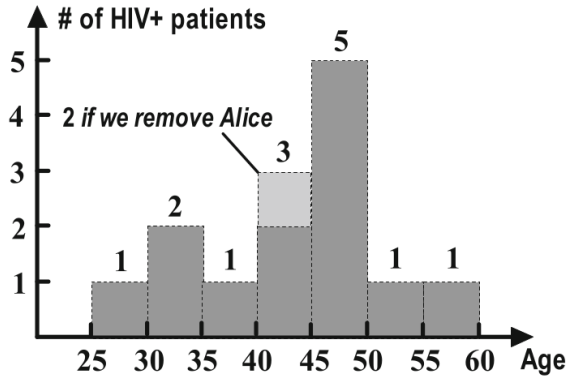


Figure 5.1: Example of the change in histogram due to removing a single record. Removing a single record gives an indication on the age of the removed record. From [69].

The definition of differential privacy (Definition 1) states that for two databases, x and y, which differ by exactly one record, the probability anyone can differentiate between the result of the same query on either database is at most the exponential factor e^{ε}. In this equation, the privacy loss is captured by the parameter ε. Increasing the value of ε allows for a greater difference in probability making the result more identifiable, guaranteeing a lower level of privacy.

Definition 1 (ε-Differential Privacy)**.** A randomized algorithm M is ε-differentially private if for all $x, y \in S$, such that $||x - y||_1 = 1$

$$Pr[M(x) \in S] \leq e^{\varepsilon} Pr[M(y) \in S]$$

5.1. Privacy Loss Parameter: Epsilon

The parameter ε captures the maximum loss of privacy. Therefore it is trivial setting the right value for ε is important, but choosing the right value for ε is a difficult "social question" according to Dwork. Two studies [70, 71] explore determining the privacy-parameter ε through the construction of elaborate threat models to obtain a good real-life bound of the risks. DP is still a new technique and not often used, but a survey of Dwork has shown that the current practical applications usually choose $0.01 < \varepsilon < 10$, without further justification but often motivated by performance targets rather than social aspects [72].

One of the companies that have already adopted DP is Apple [73]. User data is collected using various privacy budgets, depending on how delicate the associated data is. I.e. health data is transmitted a maximum of once a day with an ε=2, while emoji prediction is less private and is allowed a daily budget of ε=4. Additionally, the 2020 U.S. census was held on April 1st, 2020, using DP to improve a discovered vulnerability that exposed the 2010 census to be more re-identifiable than initially expected, by multiple orders of magnitude (Section 4.1). After evaluation of various approaches, the census has been made private using various privacy budgets depending on the scope (e.g. national, state, county), that are set by the Data Stewardship Executive Policy Committee based on accuracy-privacy-loss graphs [74]. However, in conjunction with differential privacy, the same methods of the 2010 census were applied to prevent any less security due to any unexpected flaws in the DP approach [50].

In a further attempt to substantialize the meaning of ε, the adversaries' perspective is taken. Suppose a database D is targeted by an adversary and wants to learn if his target X is in the database, denoted by $X \in D$. Additionally, the adversary obtains the output Out of an ε-DP query made on the database, $M(D) = Out$. Without any information, the adversary had an initial suspicion, but after learning the output he uses this new information to make a better guess and obtain an updated suspicion. Using Bayes' rules the updated suspicion can be defined as follows:

$$Pr[X \in D | M(D) = Out] = \frac{Pr[X \in D] \cdot P[M(D) = Out | X \in D]}{Pr[M(D) = Out]}$$

Rewriting this equation using Bayes' rules and the definition of DP (Definition 1), the updated suspicion can be bounded in terms of epsilon. Plotting these bounds for various levels of epsilon results in a contour graph Figure 5.2, providing a more visual overview of the significance of epsilon [75]. The white diagonal line represents an infinite epsilon, meaning perfect privacy. On this diagonal line, the updated suspicion is equal to the initial suspicion, as nothing is learned from the output. The higher the privacy budget ε, the more information is leaked and can be used to increase suspicion.

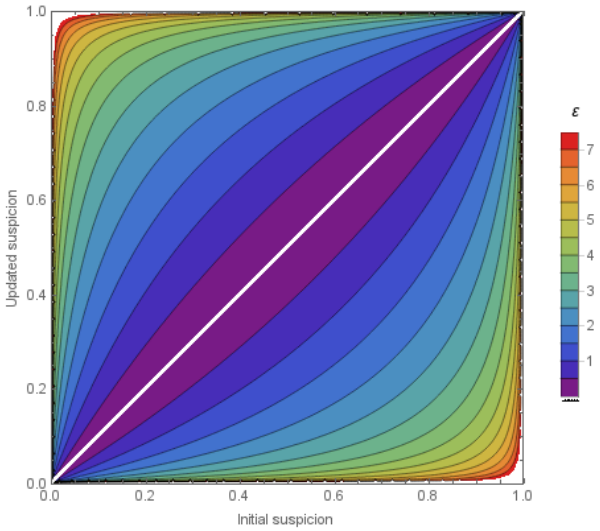


Figure 5.2: Contour graph of the probability of an adversary trying to distinguish whether his target is in a database. Without any additional information the adversary has an initial suspicion. Based on the output of an ε-DP query on the database, a new guess is make, his updates suspicion. From [75].

5.2. Important Properties

DP has multiple interesting and valuable properties that give it's strong and adaptive privacy guarantee that sets it apart from other privacy approaches.

- Quantification of privacy loss
- Composition
- Group Privacy
- Closure under post-processing

First of all, DP is quantifiable. Other methods of anonymization (e.g. using k-anonymity) provide a more binary form of security, where it is deemed secure or insecure. In differential privacy, the parameter ε defines the maximum privacy loss and can be set to accommodate any notion of privacy.

Composition allows an arbitrary combination of DP mechanisms to be combined or repeated and remain DP, such that the combination of any two mechanisms that are ε_1-DP and ε_2-DP, guarantees the combined mechanism to provide at least $(\varepsilon_1 + \varepsilon_2)$-DP. Therefore, this property allows for the modular construction of more sophisticated DP algorithms that remain quantifiable, using simple mechanisms as building blocks.

So far DP has been defined on neighboring databases, exactly one record apart. *Group privacy* allows for the protection of up to c records, offering protection to correlated groups of users or multiple records from a single user. Protecting c rows up to a loss of ε requires an increase of the privacy budget to $c\varepsilon$. In other words, ε provides groups of records up to size c with $\frac{\varepsilon}{c}$-DP.

Finally, DP provides *closure under post-processing*. Post-processing refers to the processing of differentially private results. All differentially private results can not become any less private due to any arbitrary transformation of the data. This includes combinations with any future data releases.

5.3. Mechanisms

Differential privacy can be obtained through many different techniques, often referred to as *mechanisms*. Two fundamental mechanisms of obtaining DP, *randomized response* and *Laplacian noise*, are introduced to gain a deeper understanding of how these mechanisms provide DP.

5.3.1. Randomized Response

One of the base mechanisms is randomized response [17, 76]. This mechanism grants privacy by providing a level of plausible deniability. A clear use case would be surveying taboo or illegal behavior. Due to the possible (legal) repercussions of answering truthfully, the respondents need to be assured their answers can not be used against themselves. In the randomized response mechanism, the respondent's real answer is probabilistically replaced with a random answer, providing the respondent plausible deniability.

For example, evaluating the total amount of criminals through a survey is a problematic task, as for legal reasons there could be repercussions connected to admitting to being a criminal. To preserve their privacy, respondents are asked to flip a coin in private before answering, if the coin lands heads up they answer truthfully. If the coin comes up tails, the respondent answers yes or no based on a second coin-flip. Through this procedure, the respondent can not be held accountable for their given answer. As in reality, not many people are criminals, the results in the survey's answers to be skewed. But as the overall randomness is known and many people are asked, the results can be unskewed to obtain the real number while maintaining the individual privacy of each respondent.

More precisely, if questions are answered truthfully X% of the time and $(100-X)$% answers with an equally-likely random answer, the actual statistics of truthful answers can be computed by adjusting for the skewed results, without compromising anyone their privacy.

A second more detailed example is given to illustrate the relation this mechanism has with the privacy parameter ε.

$$e^{\varepsilon} = \frac{Pr[Answer = yes|Truth = yes]}{Pr[Answer = yes|Truth = no]} = \frac{X + (100 - X)/2}{(100 - X)/2}$$

In relation to the above example, this means a respondent answers a binary question truthfully based on a fair coin-flip (X = 50%). Thus, there is a chance of 0.5 the criminal answers yes truthfully. Additionally, there is a chance of 0.25 (0.5/2) chance he randomly answers yes. The denominator, $Pr[Answer = yes|Truth =$

yes], can then be computed to be $0.5 + 0.5 \cdot 0.5 = 0.75$. As there are no other options than yes and no, the numerator can easily be computed as the complement of the denominator. This gives $\varepsilon = \ln(\frac{0.75}{0.25}) = \ln(3) \approx 1.1$. This means in the survey each respondent loses privacy equal to an ε of approximately 1.1 by answering this question.

The amount of privacy loss can be adjusted by reducing X, the chance of answering truthfully. Less truthful answers mean a higher level of plausible deniability for the respondents, but the more skewed the results become. Using a weighted coin (30/70) respondents only answer truthfully $X = 30\%$ of the time. Accordingly so, the lost epsilon decreases to $(0.3 + 0.7 \cdot 0.5 = 0.65 \rightarrow \ln(\frac{0.65}{0.35}) \approx 0.6)$

5.3.2. Laplace

When queries are of a numerical nature the most important mechanism is the Laplace mechanism [16], which lends its name from the Laplace, or double exponential, distribution from which the noise is drawn. The name double exponential distribution stems from the fact that it is a symmetrical exponential distribution, mirror-reflected along the y-axis. The exponential distribution, and therefore the Laplace distribution, has properties such that it scales multiplicatively.

Definition 2 (Laplace Distribution)**.** The Laplace distribution is a continuous probability distribution, with two parameters: *location* ($\mu \in \mathbb{R}$), defining the center of the distribution, and the *scale* ($b > 0$) determining its density.

$$Laplace(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$$

In the Laplace mechanism, the location parameter (μ) is always set to 0 so that the average of many samples converges to 0 and prevents the need for additional transposing of the noise. The scale parameter (b) of the distribution is chosen based on the level of privacy (ε) and the *sensitivity* (Δf, Definition 4) of the function. As DP hides the removal of *any* record in the data-set, the sensitivity is equal to the maximum ℓ_1-sensitivity, denoted by $||x - y||_1$, between any possible neighboring database. This way, the noise is based on this maximum p-norm (Definition 3) and is scaled to mask the removal of even the most influential record.

Definition 3 (p-norm)**.** The p-norm, for $p \geq 1$, defines the norm of a vector space in the pth dimension, used to compute magnitude, or length, of a vector.

$$||x||_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$$

Definition 4 (ℓ_p-sensitivity)**.** The sensitivity of a function f is defined as the maximum p-norm between two neighboring databases x, y, which differ at most one element.

$$\Delta f = \max_{x,y,||x-y||_1=1} ||f(x) - f(y)||_p$$

Differential privacy is achieved through the addition of noise drawn from this distribution. The Laplace mechanism can then be defined as a function that adds noise to the original value (Definition 5). For each value that needs to be made private, a new amount of noise is drawn from the distribution.

Definition 5 (Laplace Mechanism)**.** For any function f the Laplace Mechanism M_L can be defined as follows:

$$M_L(x, f, \varepsilon) = f(x) + (X_1, ..., X_k), X_i \sim Laplace(0, \Delta f/\varepsilon)$$

5.4. Architectures

Within differential privacy there are two main differential privacy models, *local differential privacy* (LDP) and *global differential privacy* (GDP) [17, 77, 78]. What sets these two models apart is where the noise is added,

at each individual output or to the computed result. Each method has its own advantages and disadvantages.

LDP perturbs the data on a user level before sending it to an aggregator, rather than GDP, which perturbs the result of a query. Because of this, GDP tends to be a more desirable model as computations can be performed using noise-less data. However, GDP uses the real data and therefore requires a trusted central authority, which has full access to the data, to aggregate the data and compute query before perturbing the result.

In 2017 Bittau [79] described a new architecture *Encode, Shuffle, Analyze* (ESA), in which the user provides two layers of encryption of which two separate services, the shuffler and the analyzer, can decrypt only one. The shuffler decrypts the outer layer that contains metadata describing the further encrypted data. The shuffler removes this metadata and groups the data together with others such that each individual record loses its significance within the batch. The analyzer then receives the batch and further decodes it to access and analyze the valuable data. This architecture is still actively being developed upon with improvements or propositions of similar algorithms [80–82].

5.5. Relaxations

Besides the original definition of ε-DP, several relaxations have been released of which the two most important ones are (ε, δ)-Differential Privacy and (α, δ)-Rényi Differential Privacy.

5.5.1. (ε, δ)-Differential Privacy

In 2013 Dwork introduced the notion of (ε,δ)-differential privacy [17]. The term δ is introduced by noise mechanisms other than Laplacian noise and represents the chance of having a privacy loss higher than ε. Typically, the value of δ is set to the inverse of the order of the data-set's cardinality ($1/|\text{data-set}|$), where if $\delta = 0$ a (ε,δ)-DP algorithm M, is ε-differentially private.

Definition 6 ((ε,δ)-Differential Privacy)**.** A randomized algorithm M is (ε, δ)-differentially private if for all $x, y \in S$, such that $||x - y||_1 = 1$

$$Pr[M(x) \in S] \leq e^{\varepsilon} Pr[M(y) \in S] + \delta$$

The so-called *catastrophe parameter* δ, is in many cases accepted and is being compared to the risk of encryption being brute-forced. Not having to provide a complete guarantee allows bounds to be looser, reducing the amount of noise to obtain the same level of ε, obtaining better results.

(ε,δ)-DP shares many of the essential properties of ε-DP. Likewise, composition sums both parameters rather than just epsilon, that is composition of any two (ε,δ)-DP mechanisms, (ε_1,δ_1) and (ε_2,δ_2), guarantees to provide at least an $(\varepsilon_1+\varepsilon_2,\delta_1+\delta_2)$-DP mechanism.

Gaussian Mechanism As the name might suggest, the Gaussian Mechanism relies on noise drawn from the Gaussian distribution [17]. The Gaussian distribution, or *normal* distribution, is one of the most commonly used distributions and has some favorable properties that, in certain cases, make for a better fit than the Laplace noise.

The Gaussian mechanism is especially strong when multiple statistics are released. This is because of the tails of the Gaussian distribution, which decrease faster than those of a Laplacian distribution. Additionally, Gaussian noise scales with the ℓ_2-distance rather than the ℓ_1-distance used to compute the distance for the Laplacian distribution. To put this into context, if the ℓ_1-distance is d then the ℓ_2-distance is equal to $\sqrt{d}$ (Definition 3). These properties especially reduce the total amount of noise added when perturbing multiple sensitive statistics. In the case of one sensitive statistic, the Laplacian mechanism is usually a better fit.

5.5.2. (α, ε)-Rényi Differential Privacy

In 2017 Mironov introduced another relaxation based on (ε, δ)-DP using Rényi divergence: (α, ε)-Rényi Differential Privacy (RDP) [83]. Shannon's entropy and Kullback-Leibler (KL; also known as relative entropy) divergence are the fundamental basis of information entropy. Rényi divergence generalizes the concepts of information entropy in a parameterized expression, the parameter $\alpha \geq 0$ is called the order [84, 85]. Special cases of Rényi divergence exist for $\alpha = 0$, 1, 2, and ∞ resulting in max-entropy, Shannon's entropy, collision entropy, and min-entropy, relatively. Additionally, Rényi divergence can simplify computation and proofs

throughout information theory. For RDP this results in an easier description and composition of Gaussian mechanisms.

Definition 7 ((α,ε)-Rényi Differential Privacy)**.** A randomized algorithm M is (α, ε)-Rényi differentially private if for all $x, y \in S$, such that $||x-y||_1 = 1$

$$Pr[M(x) \in S] \leq (e^{\varepsilon} Pr[M(y) \in S])^{(\alpha-1)/\alpha} + \delta$$

Moreover, (α,ε)-RDP satisfies $(\varepsilon + \frac{\log 1/\delta}{\alpha-1}, \delta)$-DP for any $0 < \delta < 1$. Using RDP brings advantages, for example, during composition the α's do not add up, that is composing two mechanisms of (α,ε_1) and (α,ε_2) results in $(\alpha,\varepsilon_1+\varepsilon_2)$. This and other Rényi divergence related properties allows for tighter, advanced composition theorems in respect to (ε,δ)-DP.

5.6. Trade-offs

To differential privacy as a whole, or between the different approaches presented in this chapter, there are several trade-offs to consider. To help assist in finding the right approach for a DP XAI system, previous work that evaluated DP mechanisms is considered.

5.6.1. Applications

As described in the Section 5.4, there are two main architectures to achieve differential privacy, LDP and GDP. LDP provides local privacy by adding the noise to each data point, where GDP adds noise to computed results. In more complex combinations of functions, like used in ML, GDP can also be used to perturb any intermediary results to obtain a DP model and maintain DP due its closure to post-processing (Section 5.2). Considering ML models consist of multiple computations, noise can be added to (a combination of) intermediary computations or the actual result. These two options are respectively called *private training* and *private prediction*. In private training, the hyperparameters or the inner workings, e.g. the loss function, of the model are made differentially private. Therefore any inferences made by the model after private training are private. Private prediction relates to the models remaining unperturbed (also: pure) and only the model's final prediction is made private. Each of these architectures and variations come with their own trade-offs.

Van der Maaten [86] evaluated the trade-offs for the various methods. Two private prediction methods were considered: prediction sensitivity and subsample and aggregate. Prediction sensitivity adds noise only to the model's output. A subsample and aggregate method trains multiple models, for which the output of each model is aggregated and used to produce a differentially private output. A promising implementation of this method is Private Aggregation of Teacher Ensembles (PATE) [87]. The three private training methods are model sensitivity, loss perturbation, and *differentially private stochastic gradient descent* (DP-SGD). These respective methods achieve DP through adding noise to the model's parameters after training, to the loss function during training, or limiting the contribution of each batch by adding noise to the derived gradients.

The work shows that there are numerous complex trade-offs, depending on the parameters, training examples, inference budget. For the linear models (all except DP-SGD) prediction sensitivity has the worst performance in almost any setting. Subsample and aggregate performs relatively well in $(\varepsilon,0)$-DP, especially with a lower amount of training samples ($\leq 10^4$), or with a low inference budget. For the private training methods loss perturbation consistently outperforms model sensitivity, and subsample and aggregate depending on the parameter δ and the training examples. DP-SGD for convolutional networks performs similar or better to the other private training methods. In comparison to subsample and aggregate, where $\delta > 0$, subsample and aggregate only outperforms DP-SGD on very low inference budgets.

5.6.2. Fairness & Bias

Work has also been done regarding the fairness of in systems using differential privacy. Bagdasaryan [88] has proven that DP-SGD can increase the unfair impact on the accuracy of minorities in the data-set, as its accuracy decreases more in respect to the majority class or might even be omitted from classification. Further research by Farrand [89] has shown that even a moderate imbalance in the training data can result in unfair learning behavior and can even occur with low privacy levels. They argue the clipping performed during DP-SGD bounds the strong contribution minority groups need to make to be well represented.

Kuppam [90] investigates concrete real-world examples such as providing appropriate information about elections in minority languages, allocation of public funds based on the US Census, and misrepresentation

due to an inaccurate apportionment of political representatives. The results of these real-world examples demonstrate the large negative impact the use of differential privacy can have on social welfare.

5.7. Differentially Private XAI

The previous sections have introduced the concept of differential privacy and the different methods that can be used to obtain it and a short overview regarding the trade-offs to be considered. In this final the we try to answer the question of how DP can be best applied to obtain DP XAI.

Training an XAI model consists of successive elements:

- The data
- The classifier
- The XAI model

First, the data is used to train a classifier. Second, the XAI model is fitted to the previously trained classifier. With regard to privacy and ownership of this process, several possible stakeholders can be identified. In the simplest form all of the steps or responsible entities fall under the same umbrella organization, therefore requiring less or no privacy up to the final output. However, in many situations the data is sourced from an external party. In this case it is likely there is a wish to keep the data private for the data to remain private to both other entities. Based on this organizational structure the available approaches to apply DP vary.

With this in mind, two cases can be defined. The data-owning entity owns solely the data, or one or both of the consecutive steps. In the simplest case, the ownership of the data, the classifier, and the XAI model belong to a single entity. In this situation, none of the intermediate steps need to be private. As there is no need for privacy between separate steps, the noise can be added at any point between the data and the final output.

If the data-owning entity does not own the next step, either the model or its output must be made private to prevent a possible leak of information through the classifier. This limits the approaches of where to add noise to at the latest, during the training of a classifier. But under the assumption efficient cryptographic PPML techniques are available, cryptographic techniques can help to consolidate the privacy between the separate entities and train a model without output privacy. Keeping the entire model and its output encrypted, allows for privacy to be introduced at a later stage, before decrypting. This effectively allows for DP to be introduced at a later stage, similar to the previous case.

Under the assumption these cryptographic methods are available, t is the option to introduce noise at any part of the pipeline, its application is determined by the following trade-offs. Introducing noise to the data means using a local differentially private (LDP) approach. Adding noise to the classifier or the XAI model can be done through global differentially private (GDP) approaches, private training or private prediction. Between these options there are multiple trade-offs. An LDP approach is not strongly considered as it is typically outperformed by GDP approaches. Between the GDP approaches of private training and private prediction, private prediction performs better at lower privacy budgets (ε) but spends a little of the budget on every prediction made. Except for low privacy budgets private training, especially DP-SGD, outperforms private prediction and results in an infinite amount of predictions. Private prediction requires a part of the budget to be spent for each inference made by the model, limiting the total uses the system can provide. Practical speaking, this means that at some point, once its budget is spent, the model becomes useless. Therefore, the more favorable option is private training.

Private training can be achieved in either the classifier or the XAI model. Applying a private training approach, introducing DP as late as possible in the pipeline means applying it in the XAI model, keeps the initial classifier pure. This approach is very interesting but calls for the creation of a new implementation. Creating a completely new approach to train a DP Foil Tree is difficult and not without risks. As DP is so new, it is not unheard-of for flaws to be found in the DP design or the implementations voiding the strong, sought-after DP guarantee. Therefore the development of a new DP model is outside the scope of this thesis but is a highly interesting topic for future studies. Instead, relying on existing libraries providing DP classifiers that have been thoroughly tested and reviewed, provide the most secure and easy to implement strategy.

The remaining question is how large the difference in the performance of using a DP classifier as the basis for an XAI model is, and how these private results can be interpreted with confidence.

6

Methods

This section describes the setup of the experiment with the goal of evaluating the effects of using a differentially private classifier as the basis of an XAI model. First, the methods of training and evaluating a Foil Tree on a DP classifier using existing DP implementations, as described in Section 5.7, are explained. After that, a method to be used in the interpretation of DP values is described.

6.1. Models

The choice has been made to obtain differential privacy through DP-classifiers with the use of existing libraries for to avoid making a mistake that can break the DP guarantee. Currently, there are only a few reliable sources that have published libraries that offer such implementations. IBM[1] has a Differential Privacy Library named *diffprivlib* [91]. However, much of the current research is focused on deep learning with DP-SGD, which has shown very promising results regarding classifier performances. A popular and state-of-the-art library that provides a DP-SGD implementation is *Opacus*[92], by FacebookAI[2], which is to be use as an extension on a PyTorch model. Evaluating multiple models might show an insight whether the type of model is also an important factor to consider.

It should be taken into account that both libraries provide different guarantees. Diffprivlib provides ε-DP, while Opacus uses (α, ε)-RDP. Therefore, a comparison between the models with the same ε value, is in the advantage of the Opacus model as its guarantee is somewhat looser.

Diffprivlib Diffprivlib is an actively maintained library that contains DP mechanisms, tools, and fully implemented classifiers and regressions that are extended from sklearn (also: scikit-learn)[93]. The available classifiers are *Gaussian Naive Bayes* and *Logistic Regression.*

Gaussian Naive Bayes classifier (GNB) requires two parameters to be set: the total privacy budget ε and the bounds for each feature. The classifier uses the epsilon in three separate mechanisms. First, a third of the privacy budget is used to perform a noisy count of the records belonging to each class to maintain DP in later mechanisms. The two next mechanisms add noise to both the mean and variance of each class's feature distributions. The bounds are used to determine the sensitivity of the latter two mechanisms. These bounds should be estimated using domain knowledge and not be computed using the data to preserve DP.

The Logistic Regression (LR) requires the privacy budget ε and the *data norm* as parameters. The privacy budget is evenly divided between each feature and used to compute a Laplace distributed random vector that is added to the optimizable objective. The data norm specifies the maximum for which DP is guaranteed.

Opacus Opacus extends PyTorch with Rényi differential privacy through an easily accessible API. It defines a *PrivacyEngine* which handles every step of differential privacy, as attaching the PrivacyEngine to PyTorchs' traditional stochastic gradient descent (SGD) optimizer transforms it into DP-SGD. The backpropagation is modified so that each batch clips the gradients and adds noise before updating the model parameters. The

[1] https://www.ibm.com
[2] https://ai.facebook.com/

clipping ensures a maximum ℓ_2-norm and binds the sensitivity to a maximum. Finally, a calibrated amount of Gaussian noise is then added to perturb each gradient.

As this model uses Rényi differential privacy, obtaining the right amount of epsilon is slightly more complicated. During the training of the model is spent on each training epoch. SGD slowly converges, but as epsilon is spent each epoch turning the parameters to converge as fast as possible is crucial. The parameters that have an effect on the spent epsilon are batch size, size of the data-set, delta, clipping factor, and noise multiplier. The parameter δ will be set to 10^{-5}, which is more secure than the normally suggested inverse order of the data-set. Since this data-set is rather small with 768 samples, following this suggestion would result in a rather high risk. Additionally, a lower value slightly evens out the playing field comparing to ε-DP models. The batch size, clipping factor, and noise multiplier will be determined in an experimental setting to approach the target levels of epsilon.

6.1.1. Reproducibility

Differential privacy randomizes its outcome through stochastic noise. By definition, this stochasticity is a random value, which means that each trained model is different. Usually, such randomness can be fixed by setting a seed, however, an ordinary *pseudorandom number generator* (PRNG) does not provide the perfect randomness that is required by DP. Therefore *cryptographic pseudorandom number generator* (CPRNG) should be used [94, 95]. Diffprivlib implemented this through the *secrets* library which is part of the Python Standard Library [96, 97]. Opacus uses a package extending PyTorch *torchcsprng* to obtain a CPRNG [98]. Opacus' documentation states the use of CPRNG can result in a significant slowdown and can therefore be disabled for performance reasons during experimentation through a parameter, but should not be disabled in a real implementation. Both of these CPRNG methods do have seeds, but can not be fixed. Instead, the seeds are set using random numbers provided by the operating system providing the highest level of randomness. For this reason, the computed DP model will be different every time.

Nevertheless, for the sake of obtaining reproducible and representative results that provide an overview of all possible outcomes these models can have, the experiment will be repeated for multiple runs. By repeating the model, the evaluation shows the spectrum of both the lucky and unlucky models due to randomness. All other randomnesses, such as the training split of the data or the training of the Foil Tree will be seeded to limit the number of random parameters. Fixing the other points of randomnesses the remaining randomness can be contributed solely to the effect of differential privacy.

6.1.2. Epsilon

The most important parameter to set is the epsilon determining the maximum loss of privacy. As explained in Chapter 5, there is no easy answer to this question and approaching one would require a complicated and lengthy socio-economic assessment, which is beyond the scope of this work. However, in the active community there is a general consensus which is backed by the theoretical increase of suspicion, that an epsilon of 1 and 8 can be seen as high and low privacy settings, respectively. For this reason, a range of 0.1, 1, 2, 4, and 8 epsilon is evaluated to capture the change in effects over different levels of privacy.

6.2. Data

The used data-set is the *Pima Indians Diabetes data-set*. This data-set is originally from the National Institute of Diabetes and Digestive and Kidney Diseases[3] and is freely available through Kaggle[4].

The data-set consists of 768 rows containing several medical predictor variables and one target variable: Outcome. All patients in the data-set are females at least 21 years old of Pima Indian heritage. Table 6.1 provides an overview of all the features and their domain. Most features are well known values, except for the Diabetes Pedigree Function (DPFunc), a function computing the likelihood of having diabetes based on their family history. As indicated by the domains for many features starting from 0, the data-set contains missing values that are denoted with 0 values. The data-set is also slightly imbalanced as a minority of 268, or 34.9%, of the records are positively classified with diabetes, indicated with a 1. For training and testing of the ML models in the experiment this data is split 80/20 into a training- and testing-set.

[3] https://www.niddk.nih.gov/
[4] https://www.kaggle.com/uciml/pima-indians-diabetes-database

Type	Field	Range
Integer	Pregnancies	[0, 17]
	Glucose	[0,199]
	Blood Pressure	[0, 122]
	Skin Thickness	[0, 99]
	Insulin	[0, 846]
	Age	[21, 81]
Float	Body Mass Index (BMI)	[0, 67.1]
	Diabetes Pedigree Function (DPFunc)	[0.08, 2.42]
Categorical	Outcome	{0, 1}

Table 6.1: Summary of the features available in the Pima Indians Diabetes data-set

6.2.1. Processing

As the PyTorch model requires the input to be scaled, the data will be scaled using min-max scaling operation to fit all features to the range of [0, 1]. This does result in an additional problem as the resulting boundaries will also be scaled and indiscernible, therefore unusable for human-form explanations. Simply inverse transforming the results using the true scaling factor breaks the privacy guarantee made by DP. The only method which will keep the DP guarantee is to estimate the reverse scaling using domain knowledge. While this is possible, it does add another layer of uncertainty and complexity, and is therefore ignored in this work. To obtain results comparable with the other two models any boundary results of GNB and LR will be transformed to match the Opacus results using the same scaling as post-processing. By doing so the DP guarantee would not be affected while obtaining comparable results.

The missing values that are replaces with zeros are not modified. These values could be replaced by a value from the features real domain, but this might introduce other unwanted correlations in the data. The data is not overly imbalanced that it needs to be resolved, but approaches to address the imbalance is also more difficult due to differential privacy. Typically there are 3 approaches to handle a class imbalance: *undersampling*, *oversampling*, or *synthetic data generation*. Undersampling will discard records of the already small data-set and therefore unfavorable. Oversampling and synthetic data generation (e.g. SMOTE [99]) affect the DP guarantee. Duplicating rows due to oversampling will require the use of a higher epsilon to maintain the property of *group privacy*. Synthetic data generation generates new data points based on the existing minority class, however this also affects the DP guarantee. Differentially private generation of synthetic data is still an active field of research. For this reason, the data is kept as is, and no further processing is applied to address the imbalanced data.

6.3. Metrics

To evaluate the differences as a result of using differential privacy will be measured in multiple aspects. First, the decay of the classifier is evaluated. This is the stage where differential privacy is applied, and it is trivial that any degradation will also influence the underlying XAI method. Secondly, the Foil Tree will be evaluated in various scopes. The Foil Tree will be evaluated as a whole, on the level of the generated decision rules, and finally the effect on the generated explanation.

6.3.1. Classifier

The classifier is expected to degrade as the level of differential privacy is increased. To quantify this degradation and evaluate what levels of privacy, if any, still provide a high enough performance to be usable, the following classifier statistics are computed:

- Accuracy
- Sensitivity
- Specificity
- Area Under ROC Curve (AUC)

The accuracy measures the overall ratio of correct predictions by dividing the true positive (TP) and true negative (TN) results by the sum of the TP, TN, and the false positive (FP) and false negatives (FN) to sum all cases ($(TP+TN)/(TP+TN+FP+FN)$). The sensitivity and specificity is a measure of how often true cases ($TP/(TP+FN)$) or negative cases are correctly identified ($TN/(TN+FP)$), respectively. The AUC measures

the area below the receiver operating characteristic (ROC) curve. The ROC shows the performance of a binary system for various discrimination thresholds. The area under this curve illustrates an aggregate measure for all possible thresholds. This set of metrics allows to monitor the overall change in accuracy and detect a disproportionate difference for either diabetic or health classifications.

6.3.2. Foil Tree

The Foil Tree's goal is to fit an explainable decision tree to capture as many decisions as possible. It accomplishes this by building Random Forests and deducing decision rules. Once completed, the decision rules are filtered to specification. More specifically, the generated decision rules are tested and should satisfy a specified recall and precision. Additionally, the rules are selected not to have overlap.

To observe differences within the Foil Tree the following metrics are recorded:

- Number of decision rules
- Fidelity
- Number of explanations

The above metrics are observed to look for indications the Foil Tree might be substantially different. If either the total or the filtered amount of generated decision rules significantly deviates as the privacy increases, this could be a possible cause for a change in the explanations. The fidelity is the ratio of how often the Foil Tree deduces the same outcome as the underlying classifier that is to be explained. Lastly, the Foil Tree requires a predetermined level of certainty to make a prediction. This results in the Foil Tree not being able to give a prediction for each case. Even if the explanations stay similar, there can be an effect on the number of explanations.

6.3.3. Rules

Zooming in on the Foil Tree, the set of decision rules that make up the Foil Trees internal decisions come into the picture. Upon request of a prediction, it searches for a match within a filtered set of learned rules. As the rules will always be different as a result of the differences within the classifier, making a one-to-one comparison impossible. Instead, in an attempt to quantify the changes the decision rules are split by their class label and evaluated on the following points:

- Precision
- Recall
- Surface

Precision and recall are both metrics based on relevance. Precision measures how many of all positively labeled predictions actually were positive ($TP/(TP+FP)$). Recall is, in a binary setting like is the case in this data-set, identical to sensitivity. The final metric is the surface the rule encompasses computed as the product of the distance between the bounds for each feature. An increase in the surface of decision rules could be an indicator the explanations are becoming more generic.

6.3.4. Explanation

The Foil Tree creates an explanation based on a few key metrics based on the decision rule. In order to evaluate the effect of DP on these key metrics, there are a few hurdles. These metrics by themselves are hard to judge on correctness. Additionally, not every model is able to generate explanations for the same cases.

- Feature importance
- Rule boundaries
- Feature evidence

Finally, these metrics are not a direct measurement of the quality of the explanation, as an explanation might simply have changed without being worse. If significant deviations are observed the more important question is whether the quality of the explanation has changed. This will require an additional human-interaction or questionnaire-based study.

6.4. Interpretation of DP mechanisms

Due to the stochasticity of a DP-mechanism a range of results is possible. Without knowing the domain of a result, meaning how far off the result is from the true value, it loses much of its usability and trustworthi-

ness. Computing many repetitions will give an overview of the results' domain, but in a real setting, multiple computations like in the previous experiment is usually not an option, as epsilon is spent and compounds on each computation. As the noise to obtain DP is drawn from a distribution, the range of the domain of the noise can be approached using a confidence interval (CI).

The amount of stochastic noise used in a ε-DP mechanism depends on the mechanism's distribution, the privacy budget ε and the sensitivity Δf. Recall the scale parameter of the distribution is defined as $\frac{\Delta f}{\varepsilon}$. Using these parameters a CI can be obtained to probabilistically define the noise's domain. Knowing the domain of the noise, and therefore the amount of perturbation, could return a sense of meaning to the DP value. But raises the question if a DP value accompanied by a CI still differentially private. How a CI could help to interpret the usability of DP value, is explained using the following example.

A DP-query is made requesting the average age, and returns 55, a single statistic perturbed with noise drawn from the Laplace distribution. Based on the data the sensitivity of the query is b_1 resulting in a 95% confidence interval of [-5, 5]. This means that with 95% certainty the value of 55 has at most a distance of 5 to the real value, between 50 and 60. Alternatively, if the data-set is different such that the sensitivity is b_2, resulting in a 95% CI of [-20, 20]. With this CI the returned value of 55 the real value can be anywhere between 35 and 75. It is trivial the usability of the outcome is reduced with a larger CI.

The privacy budget is a public variable that is needed to declare the level of privacy that is guaranteed. The mechanism's distribution is defined in the implementation, and is in many cases public in the form of open-source software. The sensitivity is more complex depends on two aspects: the query and the data. Only the data is to remain private. The performed query is public knowledge or might even be executed by an external person returning only the DP result. As the sensitivity represents the largest difference between the neighboring databases present, it holds at least some meta-information about the data. This raises the question of whether sensitivity can be a public parameter.

In order to uphold the same privacy guarantee for the accompanying CI, the sensitivity should be made differentially private. In order to create a private sensitivity, two approaches come to mind: adding DP-noise to the sensitivity or using the true sensitivity and creating a perturbed distribution on which the CI is based. Both approaches will be explained in more detail.

6.4.1. Private Sensitivity

To make a non-private value differentially private noise is added based on the sensitivity. In order to add noise to the sensitivity, the noise should therefore be based on the sensitivity of the sensitivity, the *sensitivity's sensitivity* (SS). The sensitivity is the maximum change to the query's output by removing any record from the data set. The SS would therefore be the maximum change between the sensitivity and the sensitivity of removing a second record from the database, for any initial record. This results in an expensive computation with a complexity of $\mathcal{O}(n^2)$. Alternatively, one could reason the SS should only be computed for the record which results in the largest sensitivity, keeping the complexity to a linear $\mathcal{O}(n)$.

Having failed in trying to extend the original proof that noise based on the sensitivity upholds the DP guarantee, to this approach of SS upholding the same guarantee, the approach computing the worst case for any two initial records is the most encompassing and secure.

The SS will vary based on the data and the query. Where the sensitivity of a sum query depends on the maximum value in the data, the SS of a sum query depends on the two highest numbers. For different queries like computing an average is much less affected by data dependencies. Because of this, some DP queries confidence intervals might be more interpretable than others. Using 3 common queries: max, sum, and avg, the SS is evaluated on data-sets drawn from a uniform and a normal distribution with various parameters. Thereafter is evaluated what the effect of adding noise based on the SS has on the bounds of a CI.

6.4.2. Perturbed Distribution

Leaving the sensitivity to be unperturbed, DP can be guaranteed by introducing noise in a later stage, such as creating a perturbed distribution. A distribution could be perturbed by discretizing it into a histogram using samples drawn from the true distribution.

There are many existing methods in order to select the number of bins for a histogram [100]. The optimal amount of bins should capture the shape of the distribution, while being large enough to squash any errors due to random sampling of the distribution [101]. For DP histograms choosing the number of bins causes another effect. The optimal amount of bins is dependent on the original data and therefore the structure also

reveals information and simply adding noise to each bin count does not satisfy differential privacy [69]. As histograms are a popular tool in research previous studies already provide multiple algorithms to compute a DP histogram [69, 102]. The idea behind the two main strategies of DP histograms is to add noise to the original data and then optimize (NoiseFirst) or optimize the histogram's structure first and then add the noise (StructureFirst). As the data itself is noisy in the in NoiseFirst approach, the optimal structure can never depend on the original data and therefore satisfies DP. StructureFirst computes the histogram normally using the real data and then uses a portion of the privacy budget ε to move the histogram boundaries. Both NoiseFirst and StructureFirst methods will be evaluated with the goal of obtaining differentially private confidence intervals.

7

Experiments and Results

This section describes the process of running experiments and the obtained results, as outlined in Chapter 6. The code used in these experiments is dependent on a Foil Tree implementation from TNO[1] which has not been made publicly available. In order to request permission please reach out to us at `wouter.zirkzee@gmail.com` or `jasper.vanderwaa@tno.nl`.

7.1. Opacus

The current implementation of Foil Tree is based around *sklearn* models. To easily train a Foil Tree, a PyTorch implementation with an Opacus PrivacyEngine needs to be extended to implement the sklearn API. One library that wraps PyTorch models such that it is compatible with sklearn is *skorch*. This worked well with normal PyTorch models, but errors occurred when used in combination with Opacus. This seems to be because it references some libraries that do not work with the parameter clipping of Opacus. Instead, a relatively simple wrapper was developed, wrapping only the essential features required to work with Foil Tree.

7.1.1. Parameters

As explained in Section 6.1, before being able to train a DP model with Opacus, some hyperparameters need to be evaluated to approach the desired levels of epsilon. First, a simple PyTorch model without differential privacy is trained. As the data-set is not highly dimensional or complex, a feed-forward linear neural network with one hidden layer of 200 nodes activated with commonly used rectified linear units (ReLU) should suffice. In anticipation of DP-SGD, a normal SGD optimizer is used with a batch size of 10 and a learning rate of 0.05 as a starting point.

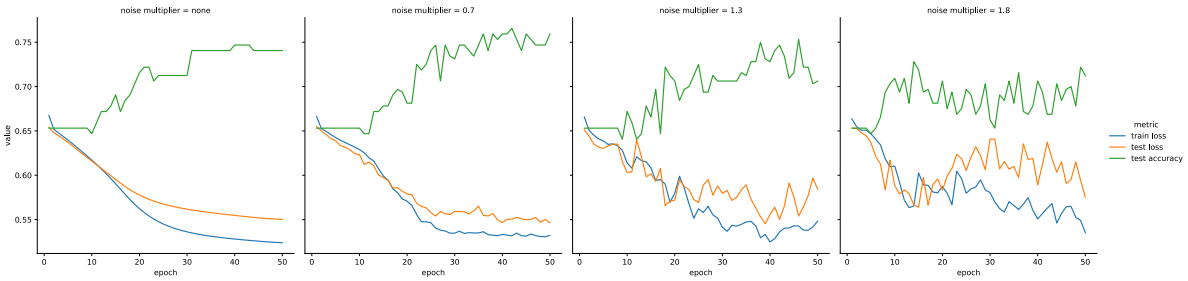


Figure 7.1: Performance degradation of a DP-SGD implementation using PyTorch with an Opacus PrivacyEngine. Using an increasing noise multiplier of 0.7, 1.3 and 1.8, 1 to 50 epochs represent an spent epsilon of [4.18, 12.51], [0.97, 3.59], and [0.5, 2.25], respectively.

Next, a *PrivacyEngine* is attached to the optimizer, which handles the clipping and addition of noise to the gradients before updating. With the PrivacyEngine an array of hyperparameters are tried. The initially chosen parameters seemed to work rather well. Figure 7.1 shows the training and test loss, as well as the test accuracy, a model without DP, and a noise multiplier of 0.7, 1.3, and 1.8. For the 50 shown epochs these

[1] https://www.tno.nl/en/

chosen noise multipliers reflect a range of epsilon of [4.18, 12.51], [0.97, 3.59], and [0.5, 2.25]. As each extra epoch increases the spent epsilon, but the models converge quite fast. To limit the spent epsilon in the chosen settings the number of epochs is limited to 20, after which the parameters seem to converge. Empirically the noise multipliers of 2.1, 1.28, 0.9, and 0.69 are found to result in a spent epsilon of 1.02, 1.99, 4.02, and 7.98, respectively, after 20 epochs. Note that a classifier with an epsilon of 0.1 has been omitted as this turned out very difficult to achieve. Attempts to achieve a 0.1 epsilon classifier do not converge well as there is likely too much noise to effectively find an SGD, and as a result, often predicts a single class.

7.2. DP Classifiers

As explained in Section 6.3, the initial point of interest was the degradation classifiers. Similar to the experience with Opacus, both the Gaussian Naive Bayes (GNB) and the Logistic Regression (LR) classifier have issues learning a model which predicts both classes at least once with an epsilon of 0.1. However, this only occurs in a few of the 50 trained models. These models have been retrained to have 50 working classifiers.

First, as a reference, similar models are trained without any differential privacy. As shown in Table 7.1, all models obtain relatively similar performances across all metrics. All randomness has been seeded such that there is no variance between multiple runs.

Model	Accuracy	Sensitivity	Specificity	AUC
GNB	0.708	0.752	0.600	0.780
LR	0.786	0.786	0.784	0.815
PyTorch	0.747	0.780	0.667	0.805

Table 7.1: Performance of Gaussian Naive Bayes, Logistic Regression and a PyTorch model without differential privacy, but otherwise similar settings as are available to the DP models.

The hyperparameters of the DP classifiers consist of feature boundaries and the norm of the data for GNB and LR, respectively. In a real-world implementation, these values should be picked independently of the actual data but instead using domain knowledge, as is often the case with differential privacy. To limit the scope of this work the real values are computed and used, but some minor preliminary tests showed the results do not seem significantly different with estimated parameters.

The results (Figure 7.2) give a representation of the degradation that occurs as a result of ensuring differential privacy. Expectedly, both diffprivlib (DPL) classifiers show a trend of decreasing performance metrics and increased variance as the privacy increases. Overall the GNB shows a stronger decrease over LR, but GNB still has better or equal results in terms of absolute values for every epsilon. As expected, the increased levels of differential privacy also display an increased variance between the data points for GNB. The variance seems to decrease using the LR for all metrics, except sensitivity. The Opacus clearly performs better than both DPL classifiers. For lower epsilons, the average accuracy actually stays the same, with a rise in sensitivity and a decrease in specificity. Except for sensitivity the variance is also quite limited in comparison to the other classifiers and does not increase much for higher levels of privacy. For all models the sensitivity seems to be most affected. In medical systems this means the system more often fail to identify a diabetic patient as one. Obviously, this is not desired behavior and should be heavily considered.

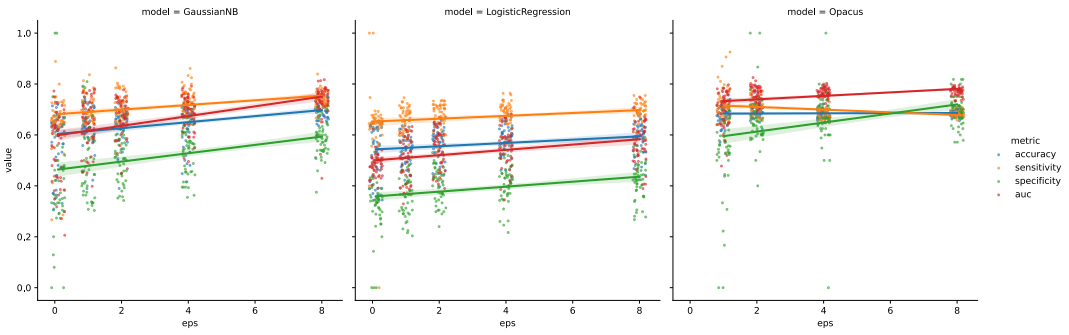


Figure 7.2: Degradation of diffprivlib's classifiers on the Pima Indians data-set through various levels of privacy on Accuracy, Sensitivity, Specificity and the Area under the Curve.

7.3. Effect on Foil Tree

Secondly, the performance of the Foil Tree using DP classifiers is evaluated. To do so a Foil Tree is trained to fit each of the previously trained models with its default parameters. In one case an LR classifier with the lowest epsilon value of 0.1, the classifier turned out to be so skewed the fitted Foil Tree's predictions only consist of a single outcome, rendering it unusable. For this reason the set LR fitted Foil Trees of 0.1 epsilon only consists of 49 models rather than 50. First, the overall metrics are evaluated by the number of rules, fidelity, and the ratio of explanations of the Foil Tree. Second, the generated rules of the Foil Tree are evaluated in more detail for each class and model. The number of rules is re-evaluated per class, the precision, recall, and surface the rules cover. Then lastly, the effect on each individual explanation is measured.

7.3.1. Overall

The number of rules (Figure 7.3) show a large variation to the total amount of rules generated, but overall a strong decrease with increased privacy. The filtered rules stay more similar with a much tighter variance but also display a slightly decreasing trend. The reason why this occurs is unclear, but it could be due to the fact that differential privacy limits individual input of data points which could lead to more generalized rules, for which fewer rules are required to mimic the classifier's decisions.

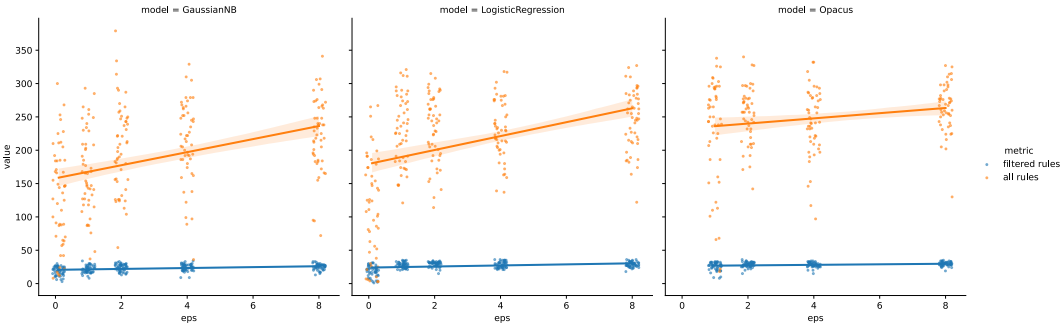


Figure 7.3: Amount of decision rules generated the form the Foil Tree, and the remaining rules after filtering

The fidelity (Figure 7.4) actually slightly increases with higher privacy for both GNB and LR, contrary to the Opacus model which decreases. All of the models show a drastic increase in variance as the available privacy budget is lowered. For LR and Opacus the variance remains relatively small for $\varepsilon = 4$ and $\varepsilon = 8$, while GNB has a very large variance for any privacy setting. The individual data points in the scatter plot show for both GNB and LR a cluster to form with fidelity of 1 for models with 0.1 epsilon.

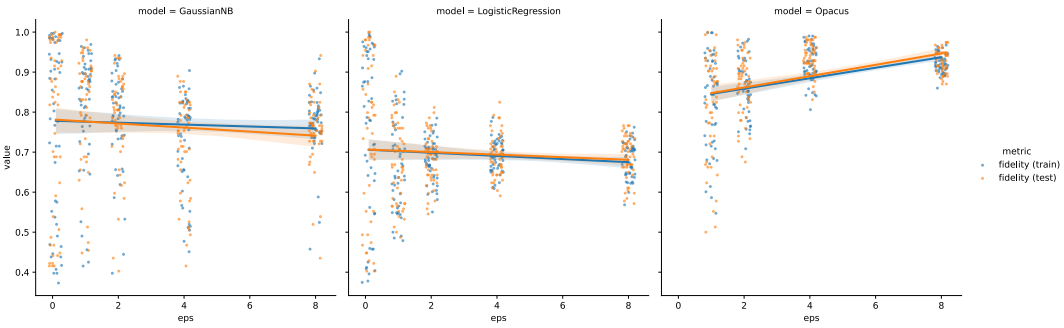


Figure 7.4: The fidelity of the Foil Tree's on both the train- and test-data-set

Finally, as the Foil Tree needs a predetermined level of certainty to give an explanation with some certainty, it can not generate an explanation for each of the test cases. The ratio of explanations Figure 7.5 shows a slight rise in both ratio and variance for GNB models. LR models show a strong rise in the number of explanations, especially between $0.1 \leq \varepsilon \leq 1$.

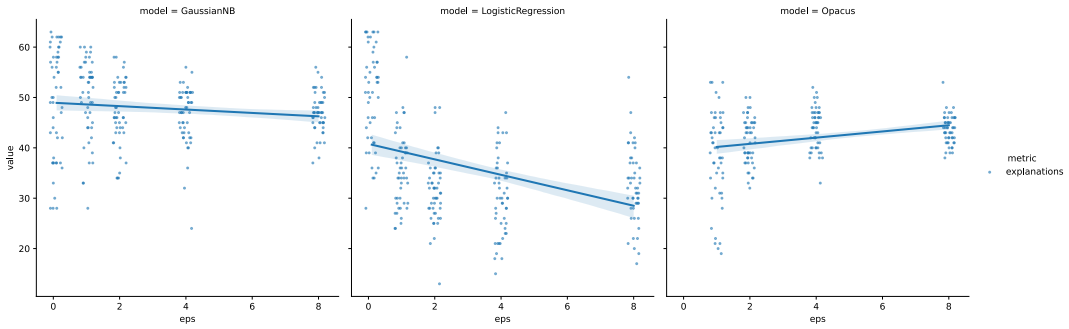
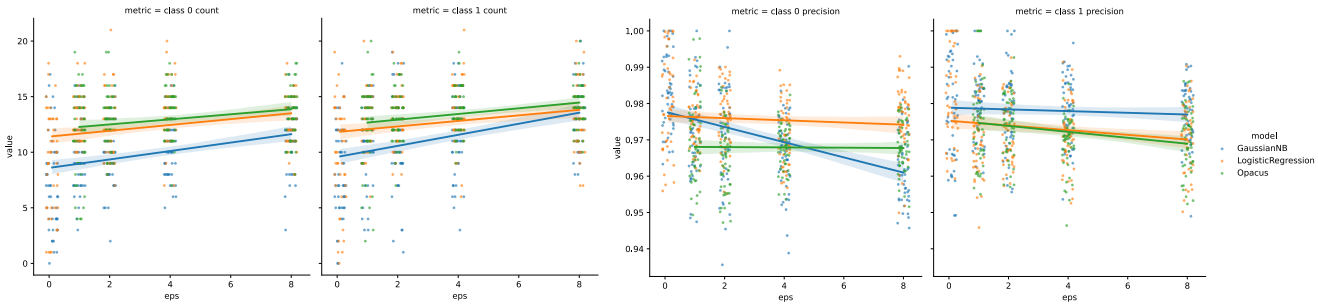


Figure 7.5: The ratio of explanations for different levels of epsilon.

7.3.2. Rules

The number of filtered decision rules has already shown to be relatively steady with a slight decrease, the rules are further evaluated to identify a possible bias. The results (Figure 7.6a) do not show any unfair decrease for either of the classes.

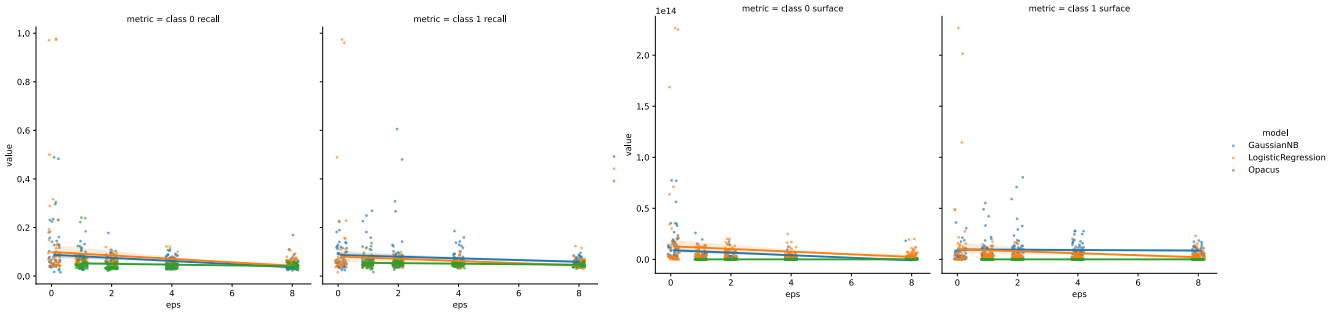


(a) Amount of generated decision rules in the Foil Tree for each classification after filtering.

(b) Precision for all Foil Tree models categorized by classification.

The precision of the rules (Figure 7.6b) shows that the precision and the variance of rules is not strongly affected in any manner, other than a slight increase in most cases. The only exception to this is for GNB's rules classifying the target of not having diabetes which shows a definitive increase, even though the absolute increase is only 0.02.

The average recall of the decision rules shows a split between the diffprivlib models, and the Opacus model (Figure 7.7a). Both of the GNB and LR show a slight increase in average recall and an increase in variance for $\varepsilon \leq 2$. Between the classes, for GNB the recall of the decision rules indicating diabetes (class 1) shows more outliers for $\varepsilon \leq 4$. Opacus' results stay similar throughout all levels of epsilon, with a relatively tight variance.



(a) Recall for all the Foil Tree models categorized by classification.

(b) Surface area captured by each Foil Tree's decision rules categorized by classification.

Figure 7.7b shows that the surface of the rules is also increased. This increase could be an explanation for the improved precision and recall, as the rule boundaries have typically become broader. This broadening

effect on the borders could possibly also be noted in the explanations. But as explained in Section 3.1, a good explanation should be *selective* and might not have the need to include all features in the explanation. To this end, evaluating the surface per feature could bring some more insights.

7.3.3. Explanations

The explanations are evaluated on features that are important for the explanation's generation: feature importance, rule boundaries, and feature evidence. These metrics are gathered for each feature per case in the test set. The results for the rule's boundaries are shown in Figure 7.8. At first look, it seems the lower privacy budgets have a higher distortion compared to a high privacy budget. The medium setting of $\varepsilon = 4$ also seems to perform well and seems the resemble the non-private explanation more often than the low privacy setting. For some features such as insulin and DPFunc the values are completely scattered over the entire plot, for all privacy budgets. Drawing conclusions are based on these results is difficult as some features seem to be less distorted for different values of epsilon. The other two features, feature importance and feature evidence, are omitted, and a more generalized approach is evaluated.

To obtain more comprehensible results, the difference between non-private explanation and the private explanation metrics are computed. The results on the individual explanations (Figures 7.9 to 7.12) show the difference between the explanation obtained by a pure model and the private models. By doing so, the expected results are for the variance of the distance between the DP and the pure explanation to expand as the epsilon decreases, but centered around the zero-axis, as DP's noise is obtained from zero-centered distributions. However, while many of the metrics, especially those from the Opacus model, follow this expected behavior, some show distinct deviations from the zero-axis.

The feature importance (Figure 7.9) does not show much of a difference between any of the models or features. Three features that do show more of a deviation are the Gaussian model for pregnancies, glucose, skin thickness, and insulin. For all of these features the noise is clearly non-zero-centered, and except for pregnancies, it seems to diverge further as the epsilon increases. Opacus seems to be slightly better than LR as it is the most zero-centered and least deviated for most of the features.

Looking at both the lower- and upper rule boundaries (Figures 7.10 and 7.11) there seem to be large differences between the features. With regard to the lower rule boundaries the pregnancy, insulin, DPFunc, age, and skin thickness show much less variance, in comparison to glucose, BMI and blood pressure. The high rule boundaries, the impact seems to have an inverse effect to that of the low boundaries, as the features that obtained a low variance in the lower boundaries, obtain significantly higher variances in the higher boundaries. Overall there does seem to be a slight increase in variance for lower privacy budgets, but the majority of the variance already occurs at a privacy budget of $\varepsilon = 8$.

The feature evidence (Figure 7.12) has a rather uniform result on all features. There is a high amount of variance for all features, which slightly increases as the privacy budget decreases.

7.4. Interpretation

In the previous experiment the range of possible outcomes has been evaluated by performing 50 repetitions to evaluate the range of outcomes as a result of the stochasticity. To interpret the usability of a result using only a single computation, the generation of confidence intervals in a private approach is evaluated.

7.4.1. Sensitivity's Sensitivity

As the sensitivity's sensitivity (SS) depends on the query and the data, multiple queries and randomly generated data-sets are used. For the queries max, sum, and avg the sensitivity and the SS are computed on 4 different randomly generated data sets to evaluate the data dependency. The data sets consist of numbers drawn from a uniform distribution between 0 and 100, 0 and 1000, and from a normal distribution with a variance of 12 and centered around 0, and a more spread distribution with a variance of 24 centered around 120. As the chosen method of computing the SS is a computationally intensive ($\Omega(n^2)$ approach, the size of the data set will be determined by the computational time.

Computing the SS for any combination of two records, already takes up to 8 minutes for only 200 records. Increasing the number of records would quickly increase the computational time due to the complexity. The results of the sensitivity and SS can be seen in Section 7.4.1. For the queries sum and avg the SS is significantly smaller than the normal sensitivity. The max query had a higher or almost equal SS in all cases. The results could be interpreted as an indication that the sensitivity carries more information for some queries

Figure 7.8: Scatter plot of the low and high boundaries for each explanation, per feature, at multiple levels of epsilon.

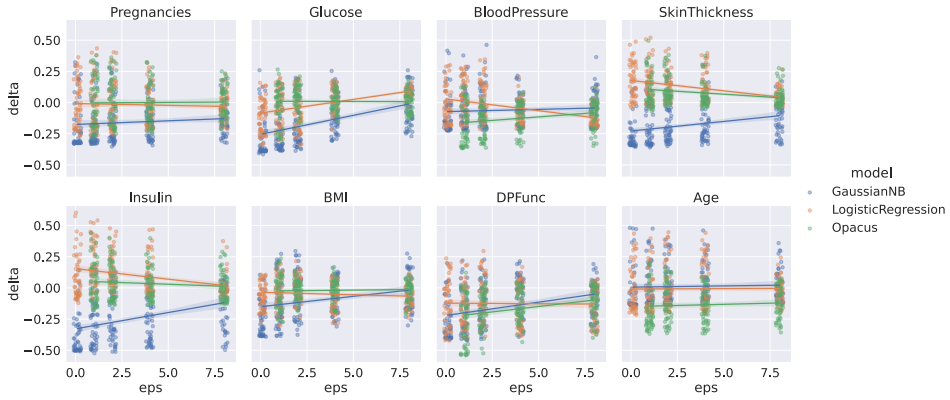


Figure 7.9: Results on the mean change of feature importance for all explanations per model in comparison to a model without differential privacy on Pima Indians data-set.

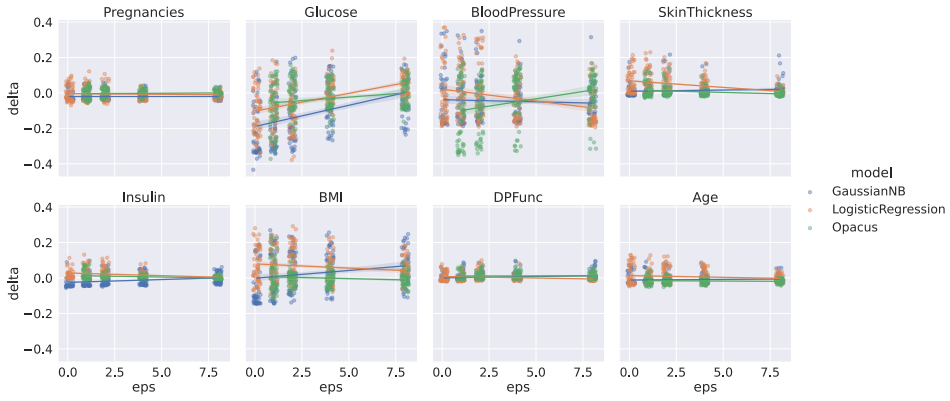


Figure 7.10: The mean change of the rule's lower bound for all explanations per model in comparison to a model without differential privacy on Pima Indians data-set.

than for others. The low SS for sum and avg is an indication there is not a high information loss and only needs a small amount of noise to be made private. The actual effect of this noise on the CI is evaluated next. For an epsilon of 0.01, 0.1, and 1 and a SS of 0.001, 0.01, 0.1, and 1, the change in a confidence interval is com-

	Rand(0,200)	Rand(0,1000)	Norm(0,12)	Norm(120,24)
sens(max)	1	2	1.87	5.12
sens(sens(max))	2	8	1.84	14.62
sens(sum)	200	989	36.24	184.35
sens(sens(sum))	1	2	6.31	5.11
sens(avg)	0.51	0.5	0.18	0.34
sens(sens(avg))	0.005	0.03	0.03	0.02

Table 7.2: The sensitivity and sensitivity's sensitivity computed on various generated data-sets.

puted. As this is once again this is a stochastic process, drawing noise to add to the sensitivity, it is repeated multiple times to find bounds.

The results (Table 7.3) show that many results would become completely unusable. For an epsilon of 0.01 only an SS of 0.001 shows a usable outcome. Increasing the epsilon from 0.01 to 0.1 an SS $\leq$ 0.01 produces somewhat usable results of 1.3% +- 12.6%. Similarly, for an epsilon of 1 the next level of SS, 0.1, becomes usable with a deviation of 0.4% +- 9.3%. As any epsilon spent on obtaining a private sensitivity can not be spent on the model, spending more than 1 epsilon here is considered too significant of a chunk of the privacy budget.

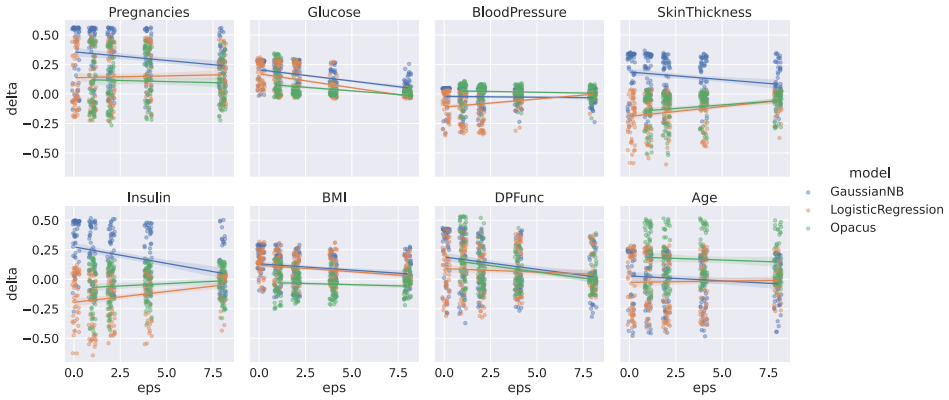


Figure 7.11: The mean change of the rule's upper bound for all explanations per model in comparison to a model without differential privacy on Pima Indians data-set.

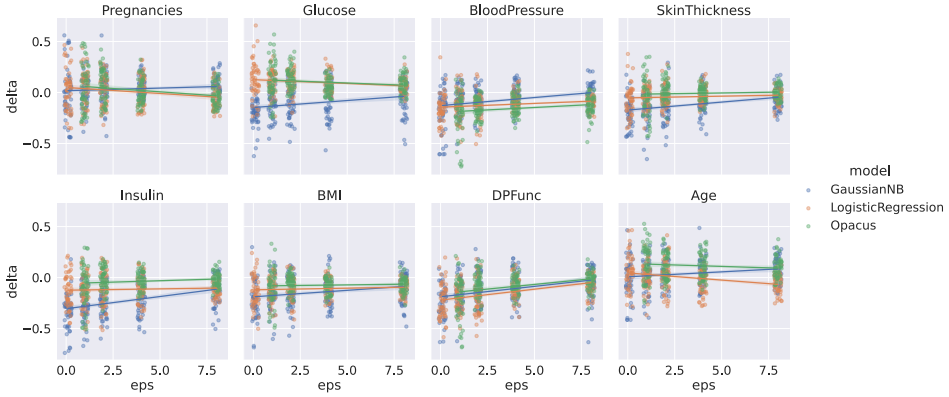


Figure 7.12: The mean change of the feature evidence over all explanations per model in comparison to a model without differential privacy on Pima Indians data-set.

	sens(sens())			
ε	0.001	0.01	0.1	1
0.01	0.6% +- 7.2%	19.8% +- 7.5%	568.1% +- 766%	6569% +- 6052%
0.1	0% +- 9.3%	1.3% +- 12.6%	121.1% +- 728%	728.9% +- 608%
1	0% +- 0%	0.1% +- 1.0%	0.4% +- 9.3%	112.6% +- 82.0%

Table 7.3: This table shows the difference in CI boundaries after perturbing sensitivity for the distribution after 100 rounds.

7.4.2. Oversights

During the evaluation of the first approach it became clear that the proposed methods are a real catch-22. Explaining the stochasticity of the original sensitivity with a confidence interval that is also has a stochastic nature, results in an infinite recursion requiring where the new stochasticity needs to be explained.

Moreover, there was a moment of realization that in many cases the sensitivity already is a private value. If the raw data is used to compute the true sensitivity, this value can not be publicized, used in the development, or used to compute (public) confidence intervals. But in many situations the raw data is not accessible, even for computing the sensitivity. Instead of using the sensitivity's sensitivity to obtain a private sensitivity, the sensitivity should be computed along with constraints that are to be applied to the data. The data constraints enforce the sensitivity to be an active upper limit, without knowing the true sensitivity. For example, the maximum sensitivity of a sum query depends on the largest value. Setting a maximum boundary for the data, the sensitivity is effectively limited.

8

Discussions and Future Work

This chapter goes into further discussion of the experiments, results, and limitations of the experiment. Finally, interesting aspects for future work are recommended.

8.1. Discussion

The results of the experiments are twofold. First, the effect of differential privacy, introduced at the classifier, has on the Foil Tree's explanations. And second, an approach to help interpret the stochasticity of a DP explanation.

8.1.1. Effect on Private Explanations

During the experiment several issues occurred with the low epsilon models of $\varepsilon = 0.1$. Such a low setting of epsilon has shown to be troublesome as 5% of the trained GNB and LR classifiers only predicted one class, or in the case of Opacus were near impossible to train. This shows the limit of what privacy budgets ε are even remotely possible using the current techniques.

The results of the experiments have been double-sided. The first results showed that the performance of the classifiers was reduced as the privacy level was increased, which was expected. Nonetheless, the performance was still on an acceptable level. But further results analyzing the effect on the Foil Tree have been worse, as in most metrics a large variance was found for all settings of epsilon. There was no significant difference of effect between the two classes.

Finally, evaluating the effect on the actual explanations, two aspects stood out. First, not all features were affected equally. For all of the metrics, the variance was large for all settings of epsilon. And second, even though the noise should be zero-centered in some cases the mean difference actually diverged away from the zero-axis.

From this, the conclusion is drawn that in this setting, the explanations react too volatile to small changes introduced by even a low notion of differential privacy, as it finds drastically different decision rules for each repetition.

8.1.2. Interpretation of Private Explanations

In order to help with the interpretation of the randomness that is present with differentially private explanations, the use of confidence intervals was explored. The main question in this problem was that the sensitivity of the computation holds some entropy about the data, which makes it not possible to share the sensitivity or a CI based on this sensitivity. Two approaches were conceived to make the sensitivity a private value.

The first approach added noise to the sensitivity, based on the sensitivity's sensitivity (SS). The results showed that some queries would be more suitable than others. The sum and max query showed to have a substantially higher than avg. Only queries with a low SS (< 0.1) would be usable to effectively perturb the sensitivity, without spending too much of the privacy budget ($\varepsilon \geq 1$) and still get a trustworthy confidence interval.

But this approach is a catch-22, as the noise added in the process of obtaining a private sensitivity would then need to be explained.

More importantly, it was realized that in many real-world applications, sensitivity actually is a private value and there is no need to spend any epsilon by using either of these approaches.

Only if the sensitivity is computed on all private attributes in the database, the resulting sensitivity is indeed a private variable. But as previously discussed, in an ideal case only the data owners can have access to the private attributes, and therefore the sensitivity should be computed in a private manner. In this case, where the sensitivity should be private to be published or to be used by developers without access to the private data, the sensitivity should be determined *without* looking at the data. This can be done by defining the maximum contributions and values. Therefore the sensitivity should be bounded by the imposed restrictions on the data, enforcing the sensitivity to hold. For example, this can be achieved through clamping the data to fall within specific ranges values or sub-sampling [103].

While these are the main two types of sensitivity, there are interesting other methods. A new method for privacy accounting that has been suggested is *individual sensitivity* [104]. In this approach, the sensitivity can be seen as a combination of public and private sensitivity. Instead of a global parameter, this new composition method allows for individual setting of parameters resulting in a more flexible approach. Setting the privacy on an individual level also gives a tighter bound than private sensitivity as the true privacy loss is considered rather than an average while remaining private.

To conclude, confidence intervals can be obtained in a private approach, and have the potential to aid with the interpretation of DP values. In a setting where a single query is used, its application is obvious. In more complicated approaches that use a combination of multiple DP-mechanisms, such as many classification models, more research is required on the combination of multiple confidence intervals to use for an interpretation of the final prediction.

8.2. Conclusion

This study distinguished two main research questions: (1) can PPML and XAI be combined to form private explanations, and (2) how does PPML alter a explanation? First, a literature study was conducted on XAI and PPML, showing that these two rather new research fields provide diverse models and methods at different levels of maturity. To our best knowledge this the first work trying to combine these two fields, in order to obtain private explanations. Based on the outcomes of the literature study, we identify Foil Tree as the only known method to generate contrastive explanations and DP as the strongest privacy guarantee that can be obtained regarding the privacy of a data-set. With this additional information the research questions can be made more specific: (1) can Foil Tree and DP be combined to form private explanations, and (2) how does DP alter a Foil Tree's explanations? In an experiment, several DP-classifiers were trained with a diabetes data-set to analyze their effects on the Foil-Tree-based explanations. These analyses provided the following insights on our research questions.

Private explanations can be obtained as a result of PPML, especially differential privacy, and XAI. In essence, DP introduces noise scaled to match the set privacy goal, resulting in a trade-off between accuracy and privacy. There are various strategies of where this noise can be applied, of which one has been evaluated empirically. The results of the experiment show that the chosen approach, even for a low level of privacy, has a major effect on the resulting private explanations, likely rendering them unusable. An additional challenge of using differential privacy to form private explanations is preserving the trustworthiness of the explanations. The stochastic nature of differential privacy results in a variable outcome. The use of confidence intervals seems to be a good candidate to explain the stochasticity and to restore the credibility of an explanation.

To conclude and answer the imposed research questions, PPML and XAI several approaches to obtain private contrastive explanations are possible. However, in the evaluated setting, the introduced stochasticity as a result of differential privacy results in a level of randomness that is too high to be considered usable. Even if this approach did not turn out in a usable scenario, this does not mean usable private explanations are impossible. Even using the same approach, differential privacy is still heavily researched and implementations might undergo significant changes. Furthermore, there are still other approaches to explore.

Based on the insights gained from our analyses, we formulated the following recommendations for the research & development of private explanations:

- Develop a new DP Foil Tree implementation (instead of a DP classifier). One example is to train a Foil Tree in a distributed manner using PATE [87]. PATE is a differentially private student/teacher voting mechanism to unanimously build a single classifier. In this approach all student classifiers are completely noise-free, but each result is aggregated and used in a DP vote to decide the boundaries of a Foil

Tree.

- Adapt Foil Tree to integrate the interpretation of the confidence intervals into each decision node in the form of soft decision boundaries. However, this might not work well for hard classification problems like diabetes in which the feature domains of both healthy and diabetic patients have a lot of overlap. Therefore, it is likely that many classified cases will fall within the uncertainty of the soft boundaries.

8.3. Limitations and Unknowns

There are a number of limitations and uncertainties that might have had a notable impact on the experiment and its results, of which the most important aspects will be explored in more detail.

8.3.1. Data

One of the limitations of this work is that the used data-set of *Pima Indians Diabetes* is rather small with a total of 728 data records. Furthermore, some of these records are only partially filled where the missing values have been replaced with 0's. These 0 values are clearly wrong in features such as age or blood pressure. It is trivial that a more complete data-set could positively impact the overall system, but replacing the missing values is an additional problem. Replacing the missing values with a value from the feature's real domain (e.g. average), can introduce other unwanted effects if the features are correlated.

The second issue is the size of only 728 records, as differential privacy typically shines on large data-sets. In large data-sets a single data point is far less contributing to an aggregate result. Because of this, the sensitivity can often be lower, therefore requiring less noise to obtain the same notion of privacy. However, medical data-sets are often quite small due to its limited availability as a result of privacy concerns. A well-founded PPML ecosystem with differential privacy as a part of it, would drastically improve the availability of sensitive data-sets. With the privacy guarantees it brings, there would no longer be a need for highly limited data-sets such as the used *Pima Indians Diabetes* data-set. Having access to much larger and diverse data-sets will result in being able to solve real-world problems like diabetes more efficiently, but requires progress in the privacy domain to be made first.

8.3.2. Models

Due to the complexity and dangers associated with creating a novel DP model, this work has relied on existing libraries and models. As differential privacy is still very much an active research field, there are not many libraries available to use in such a setting. The used *diffprivlib* offers two but does not allow for many of the parameters that are normally available for optimization as each setting has an influence on the DP guarantee. For example, the Logistic Regression Classifier only works with Limited Memory BFGS as a solver.

The diffprivlib classifiers allocate an evenly splits the available epsilon among the features. Assigning different amounts of epsilon to the features, or using fewer features, imply the most important features can be less perturbed and more effective.

8.3.3. Epsilon Bounds

There are still a lot of new developments surrounding DP that can affect the obtained results. DP gives an upper bound, however especially for more complex mechanisms used in (ε, δ)-DP and RDP, these might be too conservative. New research indicates that these bounds might be too conservative and could be tighter, resulting in less performance loss and higher usability. In this light, privacy budgets that are now discarded due to too much degradation or too high of a privacy budget could become a usable alternative.

8.4. Future Work

Even though this scenario did not turn out to be usable, differential privacy is a highly promising technique in the early stages of development, and there is are many aspects that could serve as the basis for future work. For starters, the previously given limitations can easily be translated into suggestions for future work. For example by repeating this experiment with a larger or more complete data-set, rather than the small and incomplete Pima Indians data-set. Or evaluating other DP base models, as there was an observable difference of GNB having a significantly higher variance for most metrics in comparison to LR.

Most importantly, in this work, the DP has been applied to the classifier. Currently, there are no true DP-XAI methods. The development of such an implementation could provide very different and potentially more stable results, as introducing noise at a later stage has shown to typically produce better results.

Bibliography

[1] S. L. Norris, J. Lau, S. J. Smith, C. H. Schmid, and M. M. Engelgau. Self-Management Education for Adults With Type 2 Diabetes: A meta-analysis of the effect on glycemic control. *Diabetes Care*, 25(7): 1159–1171, July 2002. ISSN 0149-5992, 1935-5548. doi: 10.2337/diacare.25.7.1159. URL http://care.diabetesjournals.org/cgi/doi/10.2337/diacare.25.7.1159.

[2] Jamie Ross, Fiona A Stevenson, Charlotte Dack, Kingshuk Pal, Carl R May, Susan Michie, Lucy Yardley, and Elizabeth Murray. Health care professionals' views towards self-management and self-management education for people with type 2 diabetes. *BMJ Open*, 9(7):e029961, July 2019. ISSN 2044-6055, 2044-6055. doi: 10.1136/bmjopen-2019-029961. URL http://bmjopen.bmj.com/lookup/doi/10.1136/bmjopen-2019-029961.

[3] Gojka Roglic and World Health Organization, editors. *Global report on diabetes*. World Health Organization, Geneva, Switzerland, 2016. ISBN 978-92-4-156525-7. OCLC: ocn948336981.

[4] Elizabeth M Heitkemper, Lena Mamykina, Jasmine Travers, and Arlene Smaldone. Do health information technology self-management interventions improve glycemic control in medically underserved adults with diabetes? A systematic review and meta-analysis. *Journal of the American Medical Informatics Association*, 24(5):1024–1035, September 2017. ISSN 1067-5027, 1527-974X. doi: 10.1093/jamia/ocx025. URL https://academic.oup.com/jamia/article/24/5/1024/3097264.

[5] Michael EJ Lean, Wilma S Leslie, Alison C Barnes, Naomi Brosnahan, George Thom, Louise McCombie, Carl Peters, Sviatlana Zhyzhneuskaya, Ahmad Al-Mrabeh, Kieren G Hollingsworth, Angela M Rodrigues, Lucia Rehackova, Ashley J Adamson, Falko F Sniehotta, John C Mathers, Hazel M Ross, Yvonne McIlvenna, Renae Stefanetti, Michael Trenell, Paul Welsh, Sharon Kean, Ian Ford, Alex McConnachie, Naveed Sattar, and Roy Taylor. Primary care-led weight management for remission of type 2 diabetes (DiRECT): an open-label, cluster-randomised trial. *The Lancet*, 391(10120):541–551, February 2018. ISSN 01406736. doi: 10.1016/S0140-6736(17)33102-1. URL https://linkinghub.elsevier.com/retrieve/pii/S0140673617331021.

[6] Wilma S. Leslie, Ian Ford, Naveed Sattar, Kieren G. Hollingsworth, Ashley Adamson, Falko F. Sniehotta, Louise McCombie, Naomi Brosnahan, Hazel Ross, John C. Mathers, Carl Peters, George Thom, Alison Barnes, Sharon Kean, Yvonne McIlvenna, Angela Rodrigues, Lucia Rehackova, Sviatlana Zhyzhneuskaya, Roy Taylor, and Mike E. J. Lean. The Diabetes Remission Clinical Trial (DiRECT): protocol for a cluster randomised trial. *BMC Family Practice*, 17(1):20, December 2016. ISSN 1471-2296. doi: 10.1186/s12875-016-0406-2. URL http://www.biomedcentral.com/1471-2296/17/20.

[7] Linda Penn, Martin White, Jaana Lindström, Annemieke Th. den Boer, Ellen Blaak, Johan G. Eriksson, Edith Feskens, Pirjo Ilanne-Parikka, Sirkka M. Keinänen-Kiukaanniemi, Mark Walker, John C. Mathers, Matti Uusitupa, and Jaakko Tuomilehto. Importance of Weight Loss Maintenance and Risk Prediction in the Prevention of Type 2 Diabetes: Analysis of European Diabetes Prevention Study RCT. *PLoS ONE*, 8(2):e57143, February 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0057143. URL https://dx.plos.org/10.1371/journal.pone.0057143.

[8] J. A. Osheroff, J. M. Teich, B. Middleton, E. B. Steen, A. Wright, and D. E. Detmer. A Roadmap for National Action on Clinical Decision Support. *Journal of the American Medical Informatics Association*, 14(2):141–145, March 2007. ISSN 1067-5027, 1527-974X. doi: 10.1197/jamia.M2334. URL https://academic.oup.com/jamia/article-lookup/doi/10.1197/jamia.M2334.

[9] Deepti Sisodia and Dilip Singh Sisodia. Prediction of Diabetes using Classification Algorithms. *Procedia Computer Science*, 132:1578–1585, 2018. ISSN 18770509. doi: 10.1016/j.procs.2018.05.122. URL https://linkinghub.elsevier.com/retrieve/pii/S1877050918308548.

[10] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, June 2020. ISSN 15662535. doi: 10.1016/j.inffus.2019.12.012. URL https://linkinghub.elsevier.com/retrieve/pii/S1566253519308103.

[11] Amina Adadi and Mohammed Berrada. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2870052. URL https://ieeexplore.ieee.org/document/8466590/.

[12] Jasper van der Waa, Marcel Robeer, Jurriaan van Diggelen, Matthieu Brinkhuis, and Mark Neerincx. Contrastive Explanations with Local Foil Trees. *arXiv:1806.07470 [cs, stat]*, June 2018. URL http://arxiv.org/abs/1806.07470. arXiv: 1806.07470.

[13] Institute of Medicine (US) Committee on Health Research and the Privacy of Health Information: The HIPAA Privacy Rule. *Beyond the HIPAA Privacy Rule: Enhancing Privacy, Improving Health Through Research*. The National Academies Collection: Reports funded by National Institutes of Health. National Academies Press (US), Washington (DC), 2009. ISBN 978-0-309-12499-7. URL http://www.ncbi.nlm.nih.gov/books/NBK9578/.

[14] Pierangela Samarati and Latanya Sweeney. Protecting Privacy when Disclosing Information: k-Anonymity and Its Enforcement through Generalization and Suppression. page 19.

[15] Luc Rocher, Julien M. Hendrickx, and Yves-Alexandre de Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. *Nature Communications*, 10, July 2019. ISSN 2041-1723. doi: 10.1038/s41467-019-10933-3. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6650473/.

[16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating Noise to Sensitivity in Private Data Analysis. page 20.

[17] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013. ISSN 1551-305X, 1551-3068. doi: 10.1561/0400000042. URL http://www.nowpublishers.com/articles/foundations-and-trends-in-theoretical-computer-science/TCS-042.

[18] Yichen Wang, Heather Zhou, Oksana Palyha, and James Mu. Restoration of insulin receptor improves diabetic phenotype in T2DM mice. *JCI Insight*, 4(15), . ISSN 2379-3708. doi: 10.1172/jci.insight.124945. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6693840/.

[19] World Health Organization and International Diabetes Federation. *Definition and diagnosis of diabetes mellitus and intermediate hyperglycaemia: report of a WHO/IDF consultation.* 2006. ISBN 978-92-4-159493-6. URL http://www.who.int/diabetes/publications/diagnosis_diabetes2006/en/. OCLC: 918994236.

[20] May 2020. URL https://www.nhs.uk/conditions/type-2-diabetes/symptoms/.

[21] Type 2 diabetes - symptoms, May 2020. URL https://www.diabetes.org/diabetes/type-2/symptoms.

[22] Standards of Medical Care in Diabetes: Summary of Revisions. *Diabetes Care*, 40(Supplement 1):S4, January 2017. doi: 10.2337/dc17-S003. URL http://care.diabetesjournals.org/content/40/Supplement_1/S4.abstract.

[23] Type 2 diabetes in children and adolescents. American Diabetes Association. *Diabetes Care*, 23(3):381–389, March 2000. ISSN 0149-5992, 1935-5548. doi: 10.2337/diacare.23.3.381. URL http://care.diabetesjournals.org/cgi/doi/10.2337/diacare.23.3.381.

[24] R. Li, P. Zhang, L. E. Barker, F. M. Chowdhury, and X. Zhang. Cost-Effectiveness of Interventions to Prevent and Control Diabetes Mellitus: A Systematic Review. *Diabetes Care*, 33(8):1872–1894, August 2010. ISSN 0149-5992, 1935-5548. doi: 10.2337/dc10-0843. URL http://care.diabetesjournals.org/cgi/doi/10.2337/dc10-0843.

[25] Ian Duncan, Tamim Ahmed, Qijuan (Emily) Li, Barbara Stetson, Laurie Ruggiero, Kathryn Burton, Dawn Rosenthal, and Karen Fitzner. Assessing the Value of the Diabetes Educator. *The Diabetes Educator*, 37(5):638–657, September 2011. ISSN 0145-7217, 1554-6063. doi: 10.1177/0145721711416256. URL http://journals.sagepub.com/doi/10.1177/0145721711416256.

[26] Andrew D Selbst and Julia Powles. Meaningful information and the right to explanation. *International Data Privacy Law*, 7(4):233–242, November 2017. ISSN 2044-3994. doi: 10.1093/idpl/ipx022. URL https://doi.org/10.1093/idpl/ipx022.

[27] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, February 2019. ISSN 00043702. doi: 10.1016/j.artint.2018.07.007. URL https://linkinghub.elsevier.com/retrieve/pii/S0004370218305988.

[28] Tjeerd A J Schoonderwoerd, Wiard Jorritsma, and Mark A Neerincx. User Requirements and Design Patterns for Explanations of Clinical Decision Support Systems. page 46.

[29] Steven K Feiner and Kathleen R McKeown. 1990-Coordinating Text and Graphics in Explanation Generation. page 8.

[30] Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(1):57–87, March 1997. ISSN 13513249. doi: 10.1017/S1351324997001502. URL http://www.journals.cambridge.org/abstract_S1351324997001502.

[31] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM Computing Surveys*, 51(5):1–42, January 2019. ISSN 0360-0300, 1557-7341. doi: 10.1145/3236009. URL https://dl.acm.org/doi/10.1145/3236009.

[32] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *arXiv:1602.04938 [cs, stat]*, August 2016. URL http://arxiv.org/abs/1602.04938. arXiv: 1602.04938.

[33] Scott Lundberg and Su-In Lee. An unexpected unity among methods for interpreting model predictions. *arXiv:1611.07478 [cs]*, December 2016. URL http://arxiv.org/abs/1611.07478. arXiv: 1611.07478.

[34] Adil Hussain Seh, Mohammad Zarour, Mamdouh Alenezi, Amal Krishna Sarkar, Alka Agrawal, Rajeev Kumar, and Raees Ahmad Khan. Healthcare Data Breaches: Insights and Implications. *Healthcare*, 8 (2), May 2020. ISSN 2227-9032. doi: 10.3390/healthcare8020133. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7349636/.

[35] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated Optimization: Distributed Machine Learning for On-Device Intelligence. *arXiv:1610.02527 [cs]*, October 2016. URL http://arxiv.org/abs/1610.02527. arXiv: 1610.02527.

[36] Nguyen Truong, Kai Sun, Siyao Wang, Florian Guitton, and Yike Guo. Privacy Preservation in Federated Learning: Insights from the GDPR Perspective. *arXiv:2011.05411 [cs]*, November 2020. URL http://arxiv.org/abs/2011.05411. arXiv: 2011.05411.

[37] Mohammad Al-Rubaie and J. Morris Chang. Privacy-Preserving Machine Learning: Threats and Solutions. *IEEE Security & Privacy*, 17(2):49–58, March 2019. ISSN 1540-7993, 1558-4046. doi: 10.1109/MSEC.2018.2888775. URL https://ieeexplore.ieee.org/document/8677282/.

[38] Privacy and machine learning: two unexpected allies?, April 2018. URL http://cleverhans.io/privacy/2018/04/29/privacy-and-machine-learning.html. Library Catalog: www.cleverhans.io.

[39] Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert, and Rickmer F. Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2(6):305–311, June 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-0186-1. URL https://www.nature.com/articles/s42256-020-0186-1. Number: 6 Publisher: Nature Publishing Group.

[40] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! A Survey of Attacks on Private Data. *Annual Review of Statistics and Its Application*, 4(1):61–84, March 2017. ISSN 2326-8298, 2326-831X. doi: 10.1146/annurev-statistics-060116-054123. URL http://www.annualreviews.org/doi/10.1146/annurev-statistics-060116-054123.

[41] Gregory S Nelson, ThotWave Technologies, and Chapel Hill. Practical Implications of Sharing Data: A Primer on Data Privacy, Anonymization, and De-Identification. page 24.

[42] Latanya Sweeney. Simple Demographics Often Identify People Uniquely. . *Pittsburgh*, page 34.

[43] Arvind Narayanan and Vitaly Shmatikov. Robust De-anonymization of Large Sparse Datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125, Oakland, CA, USA, May 2008. IEEE. ISBN 978-0-7695-3168-7. doi: 10.1109/SP.2008.33. URL http://ieeexplore.ieee.org/document/4531148/. ISSN: 1081-6011.

[44] Michael Barbaro and Tom Zeller. A Face is exposed for AOL searcher no. 4417749. *New York Times*, January 2006.

[45] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Towards Demystifying Membership Inference Attacks. *arXiv:1807.09173 [cs]*, February 2019. URL http://arxiv.org/abs/1807.09173. arXiv: 1807.09173.

[46] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, San Jose, CA, USA, May 2017. IEEE. ISBN 978-1-5090-5533-3. doi: 10.1109/SP.2017.41. URL http://ieeexplore.ieee.org/document/7958568/.

[47] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting Unintended Feature Leakage in Collaborative Learning. *arXiv:1805.04049 [cs]*, November 2018. URL http://arxiv.org/abs/1805.04049. arXiv: 1805.04049.

[48] Yue Wang, Cheng Si, and Xintao Wu. Regression Model Fitting under Differential Privacy and Model Inversion Attack. page 7, .

[49] Michael B. Hawes. Implementing Differential Privacy: Seven Lessons From the 2020 United States Census. *Harvard Data Science Review*, 2(2), April 2020. ISSN ,. doi: 10.1162/99608f92.353c6f99. URL https://hdsr.mitpress.mit.edu/pub/dgg03vo6/release/2. Publisher: PubPub.

[50] Census' privacy method, Sept 2020. URL https://www2.census.gov/programs-surveys/decennial/2020/program-management/planning-docs/privacy-methods-2020-census.pdf.

[51] Florian Tramer, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. page 19.

[52] Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High Accuracy and High Fidelity Extraction of Neural Networks. *arXiv:1909.01838 [cs, stat]*, March 2020. URL http://arxiv.org/abs/1909.01838. arXiv: 1909.01838.

[53] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv:1602.05629 [cs]*, February 2017. URL http://arxiv.org/abs/1602.05629. arXiv: 1602.05629.

[54] Caroline Fontaine and Fabien Galand. A Survey of Homomorphic Encryption for Nonspecialists. *EURASIP Journal on Information Security*, 2007:1–10, 2007. ISSN 1687-4161, 1687-417X. doi: 10.1155/2007/13801. URL http://jis.eurasipjournals.com/content/2007/1/013801.

[55] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, July 1985. ISSN 1557-9654. doi: 10.1109/TIT.1985.1057074. Conference Name: IEEE Transactions on Information Theory.

[56] Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, Lecture Notes in Computer Science, pages 223–238, Berlin, Heidelberg, 1999. Springer. ISBN 978-3-540-48910-8. doi: 10.1007/3-540-48910-X_16.

[57] L. T. Phong, Y. Aono, T. Hayashi, L. Wang, and S. Moriai. Privacy-Preserving Deep Learning via Additively Homomorphic Encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, May 2018. ISSN 1556-6021. doi: 10.1109/TIFS.2017.2787987. Conference Name: IEEE Transactions on Information Forensics and Security.

[58] Anamaria Vizitiu, Cosmin Ioan Nita, Andrei Puiu, Constantin Suciu, and Lucian Mihai Itu. Applying Deep Neural Networks over Homomorphic Encrypted Medical Data, April 2020. URL `https://www.hindawi.com/journals/cmmm/2020/3910250/`. ISSN: 1748-670X Pages: e3910250 Publisher: Hindawi Volume: 2020.

[59] A. C. Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167, October 1986. doi: 10.1109/SFCS.1986.25. ISSN: 0272-5428.

[60] A. C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 160–164, November 1982. doi: 10.1109/SFCS.1982.38. ISSN: 0272-5428.

[61] Joseph I. Choi, Kevin R. B. Butler, and Bela Genge. Secure Multiparty Computation and Trusted Hardware: Examining Adoption Challenges and Opportunities. *Security and Communication Networks*, 2019, January 2019. ISSN 1939-0114. doi: 10.1155/2019/1368905. URL `https://doi.org/10.1155/2019/1368905`.

[62] Yehuda Lindell. Secure Multiparty Computation (MPC). page 15.

[63] Ivan Damgård, Sebastian Faust, and Carmit Hazay. Secure Two-Party Computation with Low Communication. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, and Ronald Cramer, editors, *Theory of Cryptography*, volume 7194, pages 54–74. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-28913-2 978-3-642-28914-9. doi: 10.1007/978-3-642-28914-9_4. URL `http://link.springer.com/10.1007/978-3-642-28914-9_4`. Series Title: Lecture Notes in Computer Science.

[64] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. *L*-diversity: Privacy beyond *k*-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1):3–es, March 2007. ISSN 1556-4681. doi: 10.1145/1217299.1217302. URL `https://doi.org/10.1145/1217299.1217302`.

[65] Mehmet Ercan Nergiz, Maurizio Atzori, and Chris Clifton. Hiding the presence of individuals from shared databases. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, SIGMOD '07, pages 665–676, New York, NY, USA, June 2007. Association for Computing Machinery. ISBN 978-1-59593-686-8. doi: 10.1145/1247480.1247554. URL `https://doi.org/10.1145/1247480.1247554`.

[66] TiCC Tr. Dimensionality Reduction: A Comparative Review. page 36.

[67] Khaled Alotaibi, V.J. Rayward-Smith, Wenjia Wang, and Beatriz de la Iglesia. Non-linear Dimensionality Reduction for Privacy-Preserving Data Classification. In *2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*, pages 694–701, Amsterdam, Netherlands, September 2012. IEEE. ISBN 978-1-4673-5638-1 978-0-7695-4848-7. doi: 10.1109/SocialCom-PASSAT.2012.76. URL `http://ieeexplore.ieee.org/document/6406295/`.

[68] Hanumantha Rao Jalla and P. N. Girija. Probabilistic Dimension Reduction Method for Privacy Preserving Data Clustering. In Ajith Abraham, Paramartha Dutta, Jyotsna Kumar Mandal, Abhishek Bhattacharya, and Soumi Dutta, editors, *Emerging Technologies in Data Mining and Information Security*, Advances in Intelligent Systems and Computing, pages 545–554, Singapore, 2019. Springer. ISBN 9789811314988. doi: 10.1007/978-981-13-1498-8_48.

[69] Jia Xu, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, Ge Yu, and Marianne Winslett. Differentially private histogram publication. *The VLDB Journal*, 22(6):797–822, December 2013. ISSN 0949-877X. doi: 10.1007/s00778-013-0309-y. URL https://doi.org/10.1007/s00778-013-0309-y.

[70] Justin Hsu, Marco Gaboardi, Andreas Haeberlen, Sanjeev Khanna, Arjun Narayan, Benjamin C. Pierce, and Aaron Roth. Differential Privacy: An Economic Method for Choosing Epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pages 398–410, July 2014. doi: 10.1109/CSF.2014.35. ISSN: 2377-5459.

[71] Jaewoo Lee and Chris Clifton. How Much Is Enough? Choosing Epsilon for Differential Privacy. page 16.

[72] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. Differential Privacy in Practice: Expose your Epsilons! *Journal of Privacy and Confidentiality*, 9(2), October 2019. ISSN 2575-8527. doi: 10.29012/jpc.689. URL https://journalprivacyconfidentiality.org/index.php/jpc/article/view/689. Number: 2.

[73] Apple's differential privacy overview, Sept 2020. URL https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf.

[74] John M. Abowd. The U.S. Census Bureau Adopts Differential Privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 2867, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5552-0. doi: 10.1145/3219819.3226070. URL https://doi.org/10.1145/3219819.3226070.

[75] Lowering the cost of anonymization. *Desfontaines, Damien*. PhD thesis, ETH Zurich, 2021.

[76] Naoise Holohan, Douglas J. Leith, and Oliver Mason. Optimal Differentially Private Mechanisms for Randomised Response. *IEEE Transactions on Information Forensics and Security*, 12(11):2726–2735, November 2017. ISSN 1556-6013, 1556-6021. doi: 10.1109/TIFS.2017.2718487. URL http://arxiv.org/abs/1612.05568. arXiv: 1612.05568.

[77] Salil Vadhan. The Complexity of Differential Privacy. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer International Publishing, Cham, 2017. ISBN 978-3-319-57047-1 978-3-319-57048-8. doi: 10.1007/978-3-319-57048-8_7. URL http://link.springer.com/10.1007/978-3-319-57048-8_7. Series Title: Information Security and Cryptography.

[78] Björn Bebensee. Local Differential Privacy: a tutorial. *arXiv:1907.11908 [cs]*, July 2019. URL http://arxiv.org/abs/1907.11908. arXiv: 1907.11908.

[79] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnes, and Bernhard Seefeld. Prochlo: Strong Privacy for Analytics in the Crowd. In *Proceedings of the 26th Symposium on Operating Systems Principles*, pages 441–459, Shanghai China, October 2017. ACM. ISBN 978-1-4503-5085-3. doi: 10.1145/3132747.3132769. URL https://dl.acm.org/doi/10.1145/3132747.3132769.

[80] Borja Balle, James Bell, Adria Gascon, and Kobbi Nissim. Differentially Private Summation with Multi-Message Shuffling. *arXiv:1906.09116 [cs, stat]*, August 2019. URL http://arxiv.org/abs/1906.09116. arXiv: 1906.09116.

[81] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed Differential Privacy via Shuffling. *arXiv:1808.01394 [cs]*, 11476:375–403, 2019. doi: 10.1007/978-3-030-17653-2_13. URL http://arxiv.org/abs/1808.01394. arXiv: 1808.01394.

[82] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Shuang Song, Kunal Talwar, and Abhradeep Thakurta. Encode, Shuffle, Analyze Privacy Revisited: Formalizations and Empirical Evaluation. *arXiv:2001.03618 [cs]*, January 2020. URL http://arxiv.org/abs/2001.03618. arXiv: 2001.03618.

[83] Ilya Mironov. Rényi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275, Santa Barbara, CA, August 2017. IEEE. ISBN 978-1-5386-3217-8. doi: 10.1109/CSF.2017.11. URL https://ieeexplore.ieee.org/document/8049725/.

[84] Tim van Erven and Peter Harremoës. R\'enyi Divergence and Kullback-Leibler Divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820, July 2014. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2014.2320500. URL http://arxiv.org/abs/1206.2459. arXiv: 1206.2459.

[85] Albert Renyi. On Measures of Entropy and Information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 547–561, Berkeley, California, January 1961. University of California Press. URL https://digitalassets.lib.berkeley.edu/math/ucb/text/math_s4_v1_article-27.pdf.

[86] Laurens van der Maaten and Awni Hannun. The Trade-Offs of Private Prediction. *arXiv:2007.05089 [cs, stat]*, July 2020. URL http://arxiv.org/abs/2007.05089. arXiv: 2007.05089.

[87] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. SCALABLE PRIVATE LEARNING WITH PATE. page 34, 2018.

[88] Eugene Bagdasaryan and Vitaly Shmatikov. Differential Privacy Has Disparate Impact on Model Accuracy. *arXiv:1905.12101 [cs, stat]*, October 2019. URL http://arxiv.org/abs/1905.12101. arXiv: 1905.12101.

[89] Tom Farrand, Fatemehsadat Mireshghallah, Sahib Singh, and Andrew Trask. Neither Private Nor Fair: Impact of Data Imbalance on Utility and Fairness in Differential Privacy. *arXiv:2009.06389 [cs, stat]*, October 2020. URL http://arxiv.org/abs/2009.06389. arXiv: 2009.06389.

[90] Satya Kuppam, Ryan Mckenna, David Pujol, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Fair Decision Making using Privacy-Protected Data. *arXiv:1905.12744 [cs]*, January 2020. URL http://arxiv.org/abs/1905.12744. arXiv: 1905.12744.

[91] Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: The IBM Differential Privacy Library. *arXiv:1907.02444 [cs]*, July 2019. URL http://arxiv.org/abs/1907.02444. arXiv: 1907.02444.

[92] Opacus. Opacus PyTorch library. Available from opacus.ai.

[93] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[94] Yevgeniy Dodis, Adriana López-Alt, Ilya Mironov, and Salil Vadhan. Differential Privacy with Imperfect Randomness. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417, pages 497–516. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-32008-8 978-3-642-32009-5. doi: 10.1007/978-3-642-32009-5_29. URL http://link.springer.com/10.1007/978-3-642-32009-5_29. Series Title: Lecture Notes in Computer Science.

[95] Daniel Kifer, Solomon Messing, Aaron Roth, Abhradeep Thakurta, and Danfeng Zhang. Guidelines for Implementing and Auditing Differentially Private Systems. *arXiv:2002.04049 [cs]*, May 2020. URL http://arxiv.org/abs/2002.04049. arXiv: 2002.04049.

[96] diffprivlib. Random Number Generation · Issue #27 · IBM/differential-privacy-library. URL https://github.com/IBM/differential-privacy-library/issues/27.

[97] secrets — Generate secure random numbers for managing secrets — Python 3.9.2 documentation. URL https://docs.python.org/3/library/secrets.html.

[98] Pavel Belevich. torchcsprng: Cryptographically secure pseudorandom number generators for PyTorch. URL https://github.com/pytorch/csprng.

[99] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(1):321–357, June 2002. ISSN 1076-9757.

[100] Lucien Birgé and Yves Rozenholc. How many bins should be put in a regular histogram. *ESAIM: Probability and Statistics*, 10:24–45, February 2006. ISSN 1292-8100, 1262-3318. doi: 10.1051/ps:2006001. URL http://www.esaim-ps.org/10.1051/ps:2006001.

[101] Sai Venu Gopal Lolla and Lawrence L Hoberock. On Selecting The Number Of Bins For A Histogram. page 7.

[102] Xue Meng, Hui Li, and Jiangtao Cui. Different strategies for differentially private histogram publication. *Journal of Communications and Information Networks*, 2(3):68–77, September 2017. ISSN 2096-1081, 2509-3312. doi: 10.1007/s41650-017-0014-x. URL http://link.springer.com/10.1007/s41650-017-0014-x.

[103] Royce J. Wilson, Celia Yuxin Zhang, William Lam, Damien Desfontaines, Daniel Simmons-Marengo, and Bryant Gipson. Differentially Private SQL with Bounded User Contribution. *arXiv:1909.01917 [cs]*, November 2019. URL http://arxiv.org/abs/1909.01917. arXiv: 1909.01917.

[104] Vitaly Feldman and Tijana Zrnic. Individual Privacy Accounting via a Renyi Filter. *arXiv:2008.11193 [cs, stat]*, September 2020. URL http://arxiv.org/abs/2008.11193. arXiv: 2008.11193.