Universidad ORT Uruguay
Facultad de Ingeniería

Estudio de modelos de privacidad de datos

Entregado como requisito para la obtención del título de Master en Ingeniería (Por Investigación)

Ramiro Visca - 178417

Tutor: Sergio Yovine

2021

Yo, Ramiro Visca declaro que el trabajo que se presenta en esta obra es de mi propia mano. Puedo asegurar que:

- La obra fue producida en su totalidad mientras realizaba el Proyecto Final requerido por el Master en Ingeniería (por Investigación).
- Cuando he consultado el trabajo publicado por otros, lo he atribuido con claridad.
- Cuando he citado obras de otros, he indicado las fuentes. Con excepción de estas citas, la obra es enteramente mía.
- En la obra, he acusado recibo de las ayudas recibidas.
- Cuando la obra se basa en trabajo realizado conjuntamente con otros, he explicado claramente qué fue contribuido por otros, y qué fue contribuido por mi.
- Ninguna parte de este trabajo ha sido publicada previamente a su entrega, excepto donde se han realizado las aclaraciones correspondientes.

Ramiro Visca

30-08-2021

Dedicado a mi familia y amigos, quienes fueron de gran soporte en esta etapa.

Agradezco a mi tutor, Dr. Sergio Yovine, por las largas discusiones que dieron fruto a este trabajo y por su enorme soporte.

La investigación que da origen a los resultados presentados en la presente publicación recibió fondos de ICT4V - Information and Communication Technologies for Verticals bajo el código `POS_ICT4V_2016_1_15`.

Abstract Español

El presente trabajo surge como una investigación motivada por la necesidad de proteger la privacidad de los usuarios de sistemas en contextos de análisis estadístico, inteligencia artificial y publicación de datos. Para ello se ha llevado a cabo un estudio del estado del arte y se han explorado técnicas de privatización de datos basadas en Privacidad Diferencial.

Abstract

This work arises as an investigation motivated by the need of protecting systems users' privacy in the context of statistical analysis, artificial intelligence and data release. A study of the state of the art has been carried on and different privatization techniques based on Differential Privacy have been explored.

Keywords

Open Data; Privacy; Differential Privacy; Deep Learning; Machine Learning; PATE

Contents

1 Introduction

1.1 Open Data

Today's world demands great amount of data to solve a wide variety of problems. Some of which are important or belong to hard areas of problem solving where there could be potentially a large amount of restrictions, areas such as health, social development, economics, politics, among others. For that reason, it exists an implicit need for open data. By having organizations sharing openly specific data we might stand a better chance to solve some of those hard problems. Moreover, in the last couple of years there has been a recent push in the laws department for open government data. On the other hand, there has also been laws regarding the privacy of the individuals personal data within databases.

1.1.1 The value of the data

Along the existence of an organization, which could add up to many years, the organization itself gathers data from its own processes but also from its clients, users, beneficiaries, among the wide spectrum of stakeholders which it may have. This data may contain valuable information but it has to be extracted through different techniques. Today's computer systems allow this data to be stored and processed more easily as computers storage capacities expands, processing power increases and new data processing algorithms are invented. It is in the organization best interest for their existence and their capacity for innovation to be able to extract information and value from such data, as this allows to make better decisions for the future or to solve problems that the organization might be facing at certain moment of time. It is also remarkable, that the value of the data may signify a competitive advantage over other organizations that are trying to solve similar needs or that compete in the same market. For that reason, it exists a present disbelief between organizations about sharing data. This means, organizations

would not share their data with others, in fear of losing the chance of extracting their value before anyone else so as to be the first to have a competitive advantage. This disbelief, is partially true but not entirely true. Organizations should be able to share data they do not consider relevant while preserving data they consider valuable or might be valuable for them in the future. This is, in favour of allowing others to solve problems they can not solve on their own or they are simply not interested in solving. Luckily, there are new trending areas of research such as Differential Privacy, Open Data and Fairness which aims to solve these problems.

1.1.2 The utopia of Open data

Ideally an organization would openly share all their data, but as mentioned previously this goes against organization best interests. Second best case scenario we could aim for an organization releasing open data but filtering out what is relevant to them from what is not. And the last scenario, and this is where we are currently standing in terms of research, organizations should be able to provide control access either to certain data or to certain features of the data to other organizations while also preserving the desired value and the privacy of the individuals stored in such data.

1.1.3 What problems Open data creates

Open data has big problems to solve, already mentioned in previous sections. The first one is the undesirable leakage of value from the owner of the data, which prevents organizations from sharing data to solve hard problems. The second one, which can be seen as a specific case of the first one, it's privacy leakage, this mean, the data or a statistical analysis over such data also delivers sensible information about a particular individuals. Ideally, the value extracted form the data should be targeted or controlled, and secondly, the information released must be about the entire population and not about the individuals.

1.1.4 The five ways to Access Control

One could list five distinct ways on how organizations and humans overall can control how the data is accessed.

1. Access Denial: The most simple way of access control, it simply consists in

denying access to the data which holds all the value to the owner organization yet prevents any possible outsourced analysis of the data. This comes with the problem that the owner organization might not be particularly concerned about extracting value but other organizations may think otherwise.

2. Permission Agreement: This method does not really solve any of above issues, simply takes the data owner and those interested in the data and binds them into a behaviour agreement contract where restrictions, protocols and consequences are stipulated. In other words, the ownership of the data is shared through an extension.

3. Information Hiding: This method consists in hiding, denying or blocking access to certain parts of the data, that is, removing certain features which the organization considers valuable or where a possible privacy leakage may occur if the data release is carried on. However, this sort of techniques such as data anonymization, does not guarantee any privacy protection as they are vulnerable to re-identification attacks[1].

4. Changing the truth: This is where the most promising methods stands. It consists in basically lying or changing the truth of certain features in an individual with other information that does not affect the results over the population. In this area we find Differential Privacy (DP for short) as the state of the art technique[1]. Yet used wrongly, on one side it can degrade all value from the data and on the other side it can still leak privacy. However DP is the only technique, to our knowledge, to provide mathematical guarantees on what it does.

5. Encrypted communication: This is a technique that uses remote execution to compute where the data is stored, so as to avoid any data release. The data does not have to leave the trusted owner organization. This allows secret computations even in a foreign environment where one does not even have control of. However, in some cases, the results have to be decrypted to be valuable and in case of a statistical analysis or machine learning model this means the technique is still vulnerable to privacy attacks.

1.2 Context

1.2.1 General Context

The access to open data plays a fundamental role in developing more transparent and inclusive societies. In Uruguay, the initiative of carrying on public policies

about open data is regulated by the Law of Access Right over Public Information (Law Nº 18.381), which goal is to promote the availability of data produced or obtained under the power or control of public institutions.

In the mean time, the availability of data is subject to comply the current legislation of protection of private data. In case of Uruguay, this aspects are contemplated under the Law of Protection of Personal Data and Habeas Data Action (Law Nº18.331). In particular, this law defines the process of disassociation, as every processing of personal data in a way that the obtained information can not be linked back to a certain person.

Moreover, the publication of any organization, public or private in Uruguay, could also be reached by the normative framework of GDPR (General Data Protection Regulation)[2] which defines a set of regulations of data protection that applies to all organizations that operate in the European Union, independently of where they are located. This in regards to personal data related to citizen or residents of the European Union.

Furthermore, the access to data is not only motivated by the legislation of open data, but also by the need to make data available to actors, public or private, who holds the technical capacity to analyse them or use them as input to essential research projects and scientific-technological innovation.

As mentioned previously, granting access is not only an obligation to public institutions but also a need for any organization, due to that data is becoming the most valuable asset [3, 4, 5]. In fact, the big volumes of data available, together with the increasing processing computation, have enabled the research and development of algorithms of machine learning that learn from data with the goal of building predictive models, which are key to the decision making processes [6]. However, despite this fact, that data constitute an evident asset in organizations, they are faced against a bigger problem when they try to extract value from them, the data is not easy to publish nor to transfer, as in many cases, they contain personal private information that belongs to third parties [7].

Exists then, a clear tension between the ability to provide access to data and maintaining privacy. Therefore, it is essential to provide mechanisms that allow protection of private data that is made available to any third party so as to guarantee the compliance with the legislation in terms of privacy of confidential information. And it is also important to point out, that the privatized data need to hold any value or useful information to carry on successfully with the different tasks or analysis being carried away[8].

This thesis is motivated by the need of finding privatizing tools for discrete

sequential data, in particular security system logs, that might contain sensible information of the users, so as they can be released with the goal of carrying on cyber-security predictive experiments, specifically the training of neural networks dedicated to the detection and prevention of attacks[9, 10, 11, 12].

1.2.2 Particular Context

This summarizes the particular context where this thesis takes place.

1. Data: The nature of the data is Hadoop filesystem logs and Apache Logs saved from Mod Secuity Web Application Firewall (WAF). The first dataset consist on a sequence of numbers where each number represents a line of code with a logger line of code. There are logging sequences that are normal, while there are also some sequences where the behaviour is not expected and it could represent an attack or a failure of the system. The second dataset consists in URLs with query parameters logged by Mod Security in an Apache environment. Some of the queries represents normal user behaviour while others represent an attempt to attack the system, such as SQL Injection or password bruteforce attacks.
2. Restrictions: Uruguayan Laws (Law Nº18.331 and Law Nº 18.381)
3. The value or utility: Attack detection predictive models in logs
4. The problems:
   (a) Privacy Leak: Reidentification attacks of the domains contained in the logs and users' sensible data.
   (b) Mis-intended Use: leakage of other type of value not related to attack detection.

1.3 Objective

The primary objective of this thesis is to study the different models of privacy proposed by differential privacy as a way to define processes or deploy tools that allow access to data for secure release of such data or any statistical analysis carried out on it. These access mechanisms need to be compliant with stipulated restrictions and regulation of the particular context.

The secondary objective is to study, not only the privacy guarantees, but also the utility of the data release after privatization, in particular, how useful the privatized data is, to develop attack detection models on the nature of the data itself.

2 A general approach for securing open data

The problem of open data analysis with privacy preserving guarantees has a long history. As data about individuals can be stored and increases in detail, and as technology enables ever more powerful development analysis tools of these data, the need increases for a robust and mathematically rigorous definition of privacy. Differential Privacy is the promise of such definition.

2.1 Roles in data governance

1. The Data Owner: This entity is the true owner of the data. The data is based in it's characteristics, such as attributes or behaviours. This entity can be either a system, an individual, among others.

2. The Trusted Curator: An entity that is trusted by the Data Owner to store and manipulate the data.

3. Third parties: This includes any stakeholder interested in the data stored by the Trusted Curator but is not trusted by the Data Owner. This entity can be any Data Analyst, Statistician, etc.

2.2 Without privacy

In a context without privacy the individual who is owner of the data, gives away information to a Trusted Curator. Now, without privacy this last entity allows

queries to their databases allowing privacy leaks. The queries can be done by third parties with good or bad intentions. The privacy breach is not measurable and in most cases it's uncontrollable. The third parties has access to individual data and could potentially run attacks such us re-identification attacks.

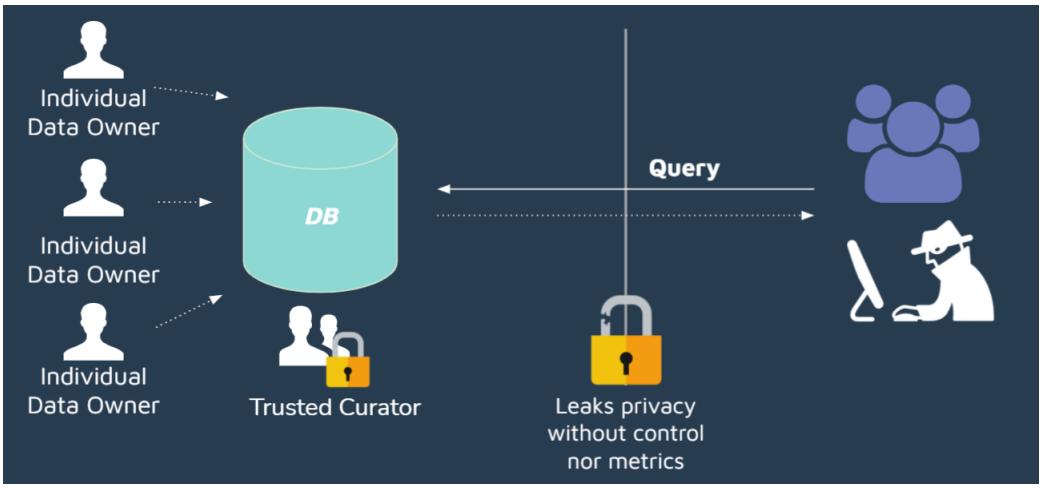

Figure 2.1: Context without privacy

The following tables depicts a simple example where an attacker with a couple of queries can reveal hidden information of the individuals, in a system where querying an individual is not allowed. By using range sum queries the attacker can make partial conclusions up to the point of revealing the health of each individual in the database.

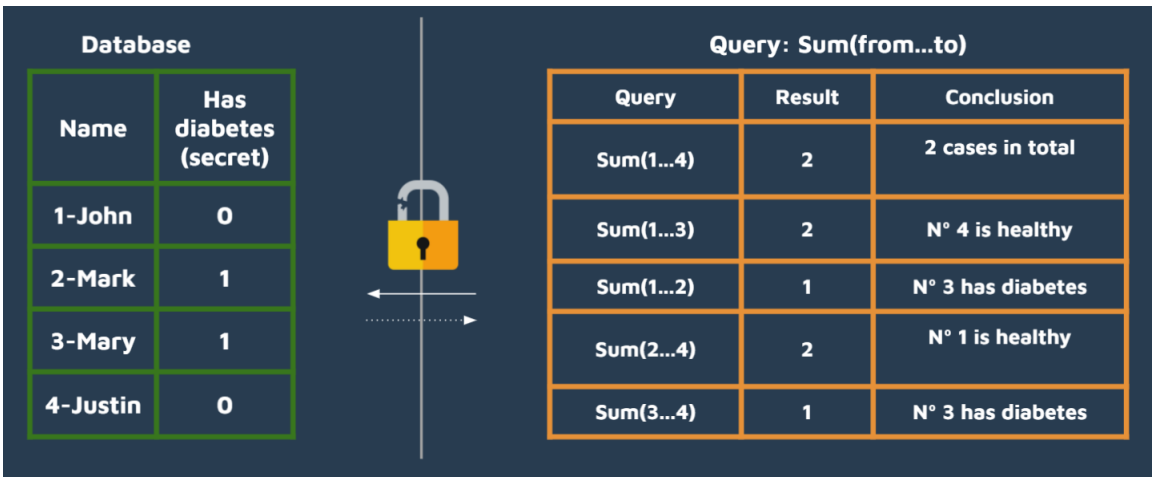

Figure 2.2: Simple example of query attacks

2.2.1 Why anonymization does not work

In the following example, a well known company gives away user ratings of movies in order to lunch a competition to design a recommender system. To no reveal users' usernames nor the movies they watch, the company uses anonymization techniques, such as replacing all usernames and movies by one way codenames. In their solution only them can trace back the username and movies. However, this turned to be false. An internet user recreated the same table using IMDB website and matching rows to the original data release. With this technique the user was able to traceback username and movie names, they preferences and even get more information by search them in Google. The anonymization failed.

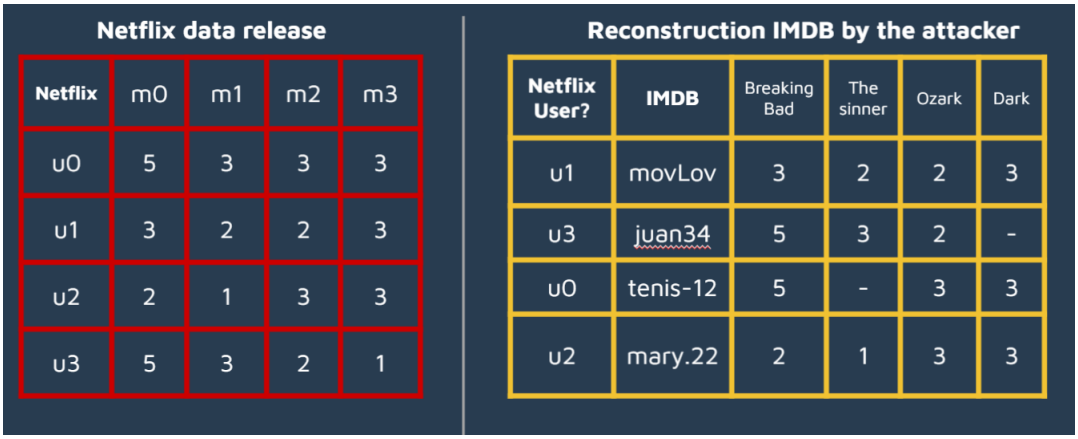

Figure 2.3: Users and Movies reconstruction

2.3 Differential Privacy

As Dwork defines in her work [1], Differential Privacy describes a promise made by the data curator to a data owner, data subject or individual. The promise is the following:

> You will not be affected, adversely or otherwise, by allowing your data to be used in any study or analysis, no matter what other studies, data sets, or information sources are available.

Differential Privacy (DP) and its mechanisms try to address the problem of learning nothing about individuals while learning useful and valuable information about the population. Under DP, a database of logs might teach us that there has

been an attack without compromising the privacy within the logs itself. Going a step even further, a database might tell us which logs are attacks without revealing anything else about the logs. This depends on how one frames the privacy preserving problem.

For interested readers, in [13] authors use intuitive illustrations and some mathematical formalism as an introduction to differential privacy for non technical practitioners. Those who are tasked with making decisions with respect to differential privacy. The document contains examples in which social scientists can understand the guarantees provided by differential privacy with respect to the decisions they make when managing data. This work also explains clearly what does differential privacy protects and what it does not.

2.4 Formalizing differential privacy

In this section a formal mathematical definition of Differential Privacy is given. The Differential Privacy definition and the prior definitions presented on this sections are taken from [1].

2.4.1 Probability Simplex

Given a discrete set B, the *probability simplex* over such set B is $\Delta(B)$:

$$\Delta(B) = \{x \in \mathcal{R}^{|B|} : x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{|\mathcal{B}|} |x_i| = 1\}$$

2.4.2 Randomized Algorithm

A *randomized algorithm* $\mathcal{M}$ with domain A and discrete range B is associated with a mapping $M : A \to \Delta(B)$. On input $a \in A$, the algorithm $\mathcal{M}$ outputs $\mathcal{M}(a) = b$ with probability $(M(a))_b$ for each $b \in B$. The probability is over the random process of the algorithm $\mathcal{M}$.

2.4.3 Distance between databases

To explain the general idea of Differential Privacy, it is often convenient to represent databases by their histograms: $x \in \mathbb{N}^{|\mathcal{X}|}$, in which each entry x_i represents the number of elements in the database x of type $i \in \mathcal{X}$, where $\mathcal{X}$ is the universe of all records of the databases x. Under this representation we can define a measurement of distance between two databases x and y, the ℓ_1 distance.

The ℓ_1 norm of a database x is defined to be $||x||_1$:

$$||x||_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i|$$

The ℓ_1 distance then of two databases x and y is $||x - y||_1$

Where $||x||_1$ is a measure of the size of the database x while $||x - y||_1$ is how many records differ between x and y.

Remark. In our work we use other representations of databases and different distance metrics such as hamming distance. These will be introduced later as it is required by the context.

2.4.4 Definition of Differential Privacy

A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $||x - y||_1 \leq 1$:

$$Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\epsilon) Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the random processes of the mechanism $\mathcal{M}$. If $\delta = 0$, then $\mathcal{M}$ is ϵ-differentially private.

2.5 Properties of Differential Privacy

In the context of this thesis there are 2 important properties to remark.

2.5.1 Post-Processing

Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ be a randomized algorithm that is (ϵ, δ)-differentially private. Let $f : R \to R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R'$ is (ϵ, δ)-differentially private.

The proof can be found in [1].

Intuitively, the Post-Processing property means that after querying a database for statistical analysis or for learning models, this property guarantees that no matter what someone does with the result, they can not obtain new information about the individuals. In other words, it is impossible to increase the privacy leakage, no matter what tools or other information are available. Once the data is privatized or accessed through a differentially private mechanism, it can no longer leak more information than what the mechanism was originally designed to leak. For instance, if a deep learning model is trained with DP-SGD[14], a mechanism for training deep learning models (more on that later), the leakage only happens when the model is trained. Once in production, the internals of the model cannot be attacked or analyzed to obtain new information about the individuals in the training dataset.

2.5.2 Composition theorem

Differential Privacy has the 'automatic' strength of having a composition theorem, in that the bounds obtained hold without any effort of the database curator nor of the types of queries or mechanisms.

Let $\mathcal{M}_i : \mathbb{N}^{|\mathcal{X}|} \to R_i$ be an (ϵ_i, δ_i)-differentially private algorithm for $i \in [k]$. Then if $\mathcal{M}_{[k]} : \mathbb{N}^{|\mathcal{X}|} \to \prod_{i=1}^{k} R_i$ is defined to be $\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), ..., \mathcal{M}_k(x))$ then $\mathcal{M}_{[k]}$ is $(\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private.

Intuitively, the Composition theorem actually means that if a database is consulted multiple times using a mechanism or a different set of mechanisms then it is guaranteed that the leakage is as big as the sum of the leakage of each mechanisms. For instance, if a person is registered in two different databases with the same information and a data analyst queries both databases with the same or with different mechanisms, the leak is at worst case, the sum of each leakage. Moreover, if the data analyst queries the same database multiple times, the leakage adds up with every single trial. In practice, one can find and demonstrate lower bounds of this general composition value, however this is a task that requires carefully designed differential privacy mechanisms and mathematics to back them up. For

instance, taking the example of DP-SGD, each batch or lot used to train a neural network leaks some ϵ privacy, after k many batches the leak adds up to $k\epsilon$. However, using moments accountant mathematics, one can find a lower bound than $k\epsilon$.

2.6 Mechanisms

Differential Privacy does not define a particular mechanism for privacy. In contrary, it propose a formal definition that a mechanisms must satisfy in order to be differentially private. In the next section the Laplacian Mechanism is presented, one of the first mechanisms and one of the building blocks for more complex mechanisms.

2.6.1 Laplace Mechanism

Given any function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(x, f(.), \epsilon) = f(x) + (Y_1, ..., Y_k)$$

where Y_i are i.i.d random variables from Lap($\frac{\Delta f}{\epsilon}$), where Δf is de $\ell_1 - sensitivity$ of the function f.

$$\Delta f = max_{x,y \in \mathbb{N}^{|\mathcal{X}|}} ||f(x) - f(y)||_1$$

and where Lap is the Laplacian Distribution centered at zero ($\mu = 0$) with scale b, with probability density function:

$$Lap(x|b) = \frac{1}{2b} exp(-\frac{|x|}{b})$$

The variance of this distribution is $\sigma^2 = 2b^2$. The Laplace distribution is a symmetric version of the exponential distribution.

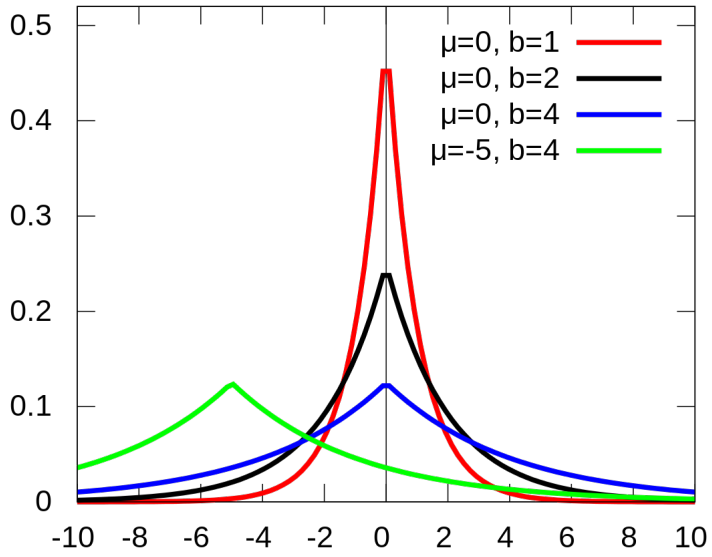

Figure 2.4: Laplacian distribution. Image taken from wikipedia.com

2.6.2 Exponential Mechanism

The exponential mechanism introduces a utility function u, which is used to balance the trade-off between privacy and utility. The exponential mechanism $\mathcal{M}_E(x, u, \mathcal{R})$ selects and outputs an element $r \in \mathcal{R}$ with a probability proportional to $\exp(\frac{\epsilon u(x,r)}{2\Delta u})$ where $\Delta u \equiv max_{r \in \mathcal{R}} max_{x,y:||x-y||_1 \leq 1}$.

2.7 Privacy Models

There are two main types of models of privacy proposed by differential privacy that we can take into consideration when implementing our system privacy compliant architecture. And we can include a variation of one of them as a third type of model.

1. The Local Model
2. The Centralized Model
3. Synthetic Databases

2.7.1 The Local Model

The model of privacy called the local model, also known as non-interactive or offline model consists on creating a database with data already privatized. This means, a randomized mechanism M is applied to the data recollected from the individuals before storing it into the database by the Trusted Curator. The privatization and it's leakage takes place when recollecting the individual information and not when querying the database. This model takes the advantage of the post processing property of differential privacy, so as, the data scientists can ask as many queries to the database as they desires without to worry about leakage composition. The database is privatized only once. This model allows the database to be released entirely under (ϵ, δ) - differentially private guarantees.

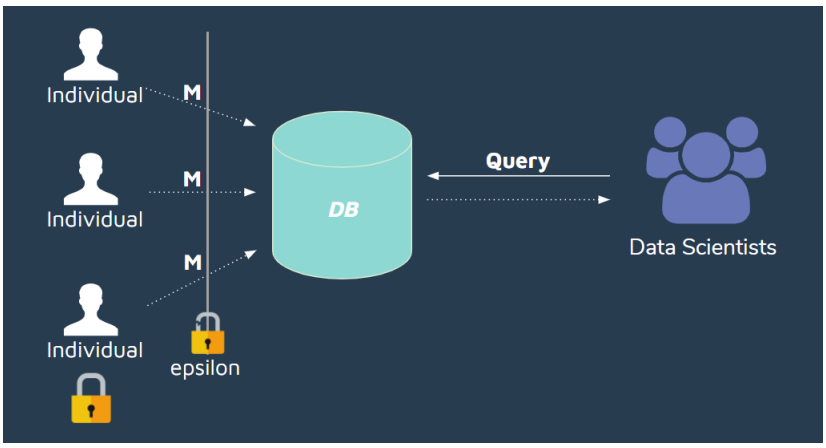

Figure 2.5: The Local Model

2.7.2 The Centralized Model

The model of privacy called centralized model, also known as interactive or online model, consist of asking n queries to the database by the data scientists. The database is owned and/or protected by a Trusted Curator. The query is a function applied to the database. Then the result of the function is privatized with some mechanism M, for instance some (ϵ, δ) - differentially private mechanism. This model allows to ask queries adaptively, for example it allows to query the database a second time based on the previous responses. However, each query done by the data scientists to the database has to be considered as a composition of mechanisms and the accumulated ϵ leakage has to be taken into account. Each query to the

database has an upper bound leakage of ϵ while k queries has an upper bound of $k\epsilon$ leakage due to composition. Besides, when all the queries are known in advance, a non-interactive approach should give a better answer than this method as it is able to correlate the noise added knowing the structure of the queries[1].

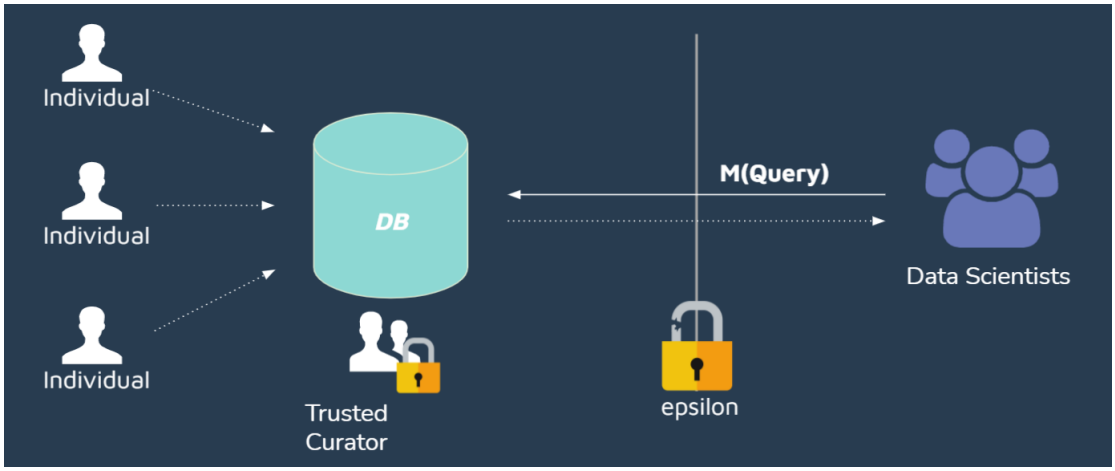

Figure 2.6: The Centralized Model

2.7.3 Synthetic Databases

This is not actually a privacy model but it works as a model of private data release. In this scenario the Trusted Curator transforms the database via a local randomized mechanism M in order to create a new privatized database. In simple words, is similar to applying the local model, querying the individuals of the database and storing the privatized result of such identity query plus a randomized mechanisms. In this case, the data scientists interacts with a privatized database protected by the post processing property with no risk of composition leakage. The privatization and it's leakage takes place when the Trusted Curator creates the new database.

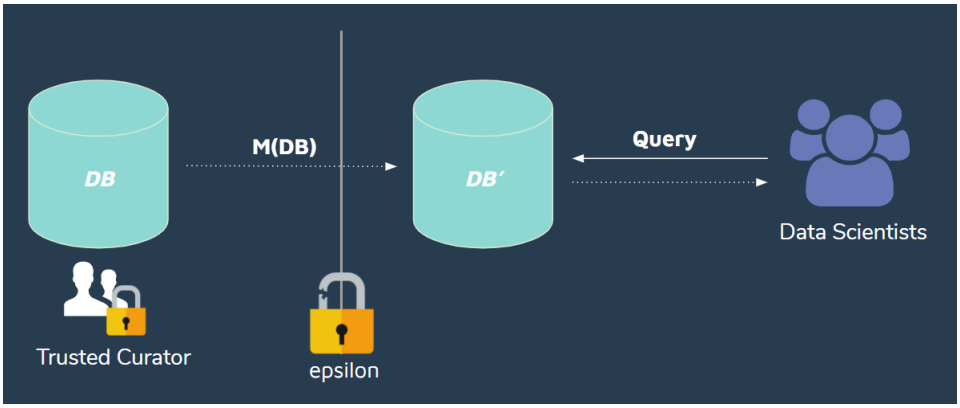

Figure 2.7: Synthetic Database

3 Privacy and learning

The following chapter presents a summary of publications that had used differential privacy to either protect a learned model, model protection, or to generate fake synthetic databases for data release. These works has been useful to understand how DP works in practice.

3.1 Model Protection

When Deep Learning models are trained with a dataset they may memorize the data even before reaching an overfitting state. This allows to attack the learnt models and recover part of the data used[15]. This behaviour is unwanted in terms of privacy for published models. For that reason, there is a field of study that combines Deep Learning and Differential Privacy to protect such models. The following works have been studied.

In [14] a new algorithmic technique for learning that introduces a differential privacy mechanism within Stochastic Gradient Descent algorithm (DP-SGD). Moreover, it introduces a refined analysis of privacy costs based on Moments Accountant. The experiments demonstrate that it is possible to train deep neural networks with non-convex objectives in this way, under a modest privacy budget and at a manageable cost in software complexity, training efficiency and model quality.

In [16] they apply Differential Privacy to improve the utility of outlier detection, even with new samples. They present a theoretical analysis on how Differential Privacy helps with the detection and then conduct extensive experiments using system logs such as Hadoop File Systems. In this work they use DP-SGD to train the models.

In [17] it is claimed that a model may store some of its training data and

a careful analysis may therefore reveal sensitive information. They propose a generally applicable approach to providing strong privacy guarantees for training data named PATE. The black-box fashion approach combines multiple models trained with disjoint datasets. Because they rely directly on sensitive data, these models are not published, although they are used as teachers for an student model. The student learns to predict the output chosen by noisy voting among all of the teachers. The student can not access the underlying data or parameters of the teachers models. The student privacy properties hold even if an adversary can not only query the student but also inspect its internal workings. This work claims to archive state of the art privacy/utility trade offs on MNIST and SVHN thanks to privacy analysis and semi-supervised learning.

In [18] authors take their previous work [17] and scale it to larger-scale learning tasks and real world datasets. The work shows that PATE can scale to learning tasks with larger numbers of classes and imbalanced data with errors. A new noisy aggregation mechanisms for teacher ensembles is introduced which adds less noise with tighter differential privacy guarantees. The teacher consensus is increased by using more concentrated noise and when lacking consensus no answer is given to the student.

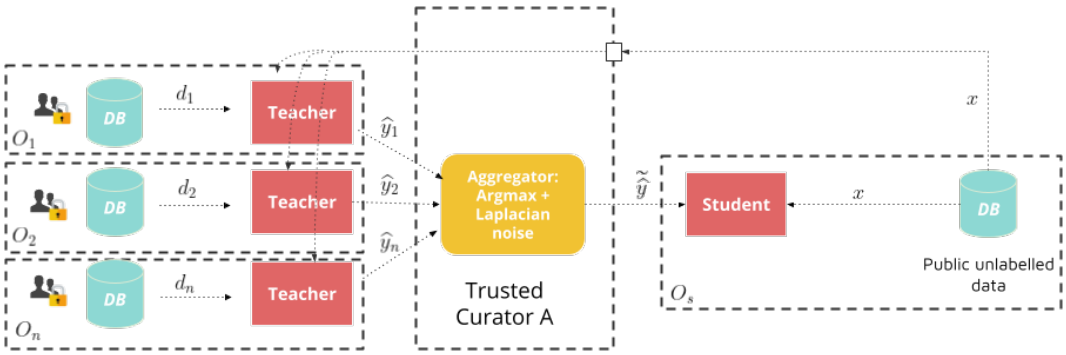

Figure 3.1: PATE

3.2 Synthetic Data Release

Another option that can be done with data is to release it publicly. However, this goes against the objective of privacy. For that reason, there is a field of study to create synthetic data to release instead of the original one. The following works have been studied.

In [19] shows in depth analyses of Differential Private algorithms in terms of accuracy and usability potential. They implement utility metrics on diferentially private synthetic dataset and compare mechanism utility on different categories.

In [20], the goal is to anonymize the author of a text while preserving its semantics by finding synonyms. For such purpose, they propose to augment the semantic information in the text by training a reward function using reinforcement learning and then applying the exponential mechanism to the output of a sequence-to-sequence RNN. This approach pays an important price in terms of privacy leakage because of the compositional theorem which entails that the overall loss in privacy is $k\epsilon$, where k is the length of the text. They do not propose any metric to measure the distortion between the original text and the synthetic one.

In [21] is claimed to be the first method to fulfills differential privacy and so guarantee provable plausible deniability of documents' authorship. They use the most common representation of documents, a vector space model where each document is a vector typically containing its term frequencies or related quantities. They produce synthethic term frequency vector for the input documents that can be used in replacement of the original vectors. This work claims not only to have a low impact on its accuracy but also it strongly affects authorship attribution techniques to levels that make authorship attribution become unfeasible.

In [22] authors propose a privacy-preserving sensing framework for accessing time-series data in order to assure certain utility while protecting individuals' privacy. The approach consists in a Replacement Autoencoder, an algorithm that learn how to transform discriminative features of data that correspond to sensitive inferences, into some features that have been more observed in non-sensitive inferences. The replacement method not only eliminate the possibility of recognizing sensitive inferences, it also eliminates the possibility of detecting the occurrence of them. It is also evaluated the efficacy of the algorithm with an activity recognition task using extensive experiments.

In [23] authors provide two contributions. First, they compare different datasets under different techniques against different utility metrics. Second, they use deep learning to generate differentially private synthetic datasets with higher data util-

ity. The deep learning models can capture relationships among multiple features and use these models to generate differentially private synthetic datasets. According to the authors this approach in conducted on multiple datasets showing a robust approach.

The work [24] introduces a framework based on the advances of generative adversarial networks (GANs) to model rich semantic data maintaining both the original distribution of the features and the correlations between them. The output of the framework is a deep network, a generator, able to create new data on demand.

In [25] claims that one common issue in GANs is that the density of the learned generative distribution can easily remember training samples. This becomes a major concern when GANs are applied to private or sensitive data. Authors propose a differential private GAN (DPGAN) model, in which they achieve differential privacy in GANs by adding carefully designed noise to gradients during the learning phase in a similar fashion to DP-SGD. They demonstrate that their method can generate high quality data points at a reasonable privacy levels, and authors provide a rigorous proof for the privacy guarantee.

3.3 Utility

In [26] a privacy-utility trade off is presented for an arbitrary set of finite alphabet source distribution. Privacy is quantified using differential privacy and utility is quantified using expected Hamming distortion maximized of the set of distributions. The family of source distribution sets is categorized into three categories. Last, authors claim Differentially private leakage is an upper bound on mutual information leakage, the two criteria are compared analytically and numerically to illustrate the effect of adopting a stronger privacy criterion.

In [27] they try to reduce the difference in utility between local differential privacy and centralized differential privacy in the case of counting queries. They propose a new local differential approach called the truncated geometric mechanism. They claim their approach obtains better results than other local differential privacy methods known in the literature. They consider a particular d-private mechanism based on the geometric noise distribution, they explore it's properties and they show that the proposed mechanism is better than the typical k-Randomized-Responses (kRR)[1].

4 Investigated Approaches

In this work two approaches were investigated. First approach, to generate a new synthetic database, capable of maintaining high level of utility in terms of value while preserving privacy of the individuals. Second approach, explore a way to protect deep learning models while protecting the original data used to create such models.

4.1 First approach: Synthetic Databases

The idea behind the first approach is to maintain the sequence representation and generate new sequences as a replacement for those in the original dataset. For this particular approach each sequence is considerated as a database. RNN based Generative models were trained in order to generate a sequences from a probability distribution, that is, each token of the sequence is picked following the learnt distribution. In that sense, a few ideas were tested.

4.1.1 Deeplog Hadoop File System

In all of the experiments bellow Deeplog dataset[28] was used, which consists in many logged sequences of numbers. Each number can be mapped to a line of code of a log in a Hadoop file system. A sequence represents a transaction with the system. For control, a language model classifier made out of RNNs was trained to detect abnormal sequences on the dataset. The control average accuracy was around 0.97, for a balanced subset of 8000 sequences.

```
5 5 5 22 11 9 11 9 11 9 26 26 26 23 23 23 21 21 21
22 5 5 5 11 9 11 9 11 9 26 26 26
22 5 5 5 26 26 26 11 9 11 9 11 9 2 3 23 23 23 21 21 21
22 5 5 5 11 9 11 9 11 9 26 26 26
22 5 5 5 26 26 26 11 9 11 9 11 9 4 3 3 3 4 3 4 3 3 4 3 3 23 23 23 21 21 21
22 5 5 5 26 26 26 11 9 11 9 11 9 3 3 4 3 4 3 3 3 4 4 3 3 23 23 23 21 21 21
5 22 5 5 26 26 11 9 11 9 11 9 26 23 23 23 21 21 21
22 5 5 5 26 26 26 11 9 11 9 11 9 4 4 3 2 23 23 23 21 21 21
5 22 5 5 11 9 11 9 11 9 26 26 26 23 23 23 21 21 21
5 5 5 22 11 9 11 9 11 9 26 26 26 23 23 23 21 21 21
5 22 5 5 11 9 11 9 11 9 26 26 26 3 3 4 3 3 4 23 23 23 21 21 21
```

Figure 4.1: Deeplog Normal Sequences Example

```
5 5 5 22 11 9 11 9 26 26 26 9 11 23 23 23 21 21 20 21 20
22 5 5 5 26 26 26 11 9 11 9 11 9 23 23 23 21 21 28 26 21
5 5 5 22 11 9 11 9 11 9 26 26 26 23 23 23 21 21 21 20
5 22 5 5 13 11 9 13 11 9 13 11 9 26 26 26 23 23 23 21 21 21
5 5 5 22 11 9 11 9 11 9 26 26 26 23 23 23 21 20 21 21
5 5 22
5 5 5 22 11 9 11 9 26 26 11 9 26 23 23 23 21 21 20 21
5 5 5 22 11 9 11 9 11 9 26 26 26 23 23 23 21 20 21 21
22 5 5 5 26 26 26 11 9 11 9 11 9 3 3 4 3 3 4 3 4 3 23 23 23 21 21 20 21
5 5 22 5 11 9 11 9 11 9 26 26 25 18 5 26 16 6 26 21 23 23 23 21 21 21
5 5 5 22 11 9 11 9 11 9 26 26 25 18 5 6 16 26 26 23 23 23 23 21 21 21 21
```

Figure 4.2: Deeplog Abnormal Sequences Example

For this work, in terms of privacy, *each entry in the dataset (each sequence) represents a database*. The hamming distance is used then in the definition of Differential Privacy. The goal is to protect each sequence while still be able to detect whenever is normal or abnormal behaviour.

4.1.1.1 Idea 1 - Embedding and Exponential Mechanism

A *simple RNN* model with an *embedding layer* is trained to predict the class of the sequence. Then the embedding layer is used together with the cosine similarity to calculate the distance between symbols, then this works as the utility function of the exponential mechanism. In practice we calculate a Δ_u of 1.96 and the exponential mechanism is instantiated. Then a synthetic database is generated applying this mechanism for each symbol in the sequence individually and only using the full original sequence just once. This last bit means we are using a local model of $(\epsilon, 0)$ - differential privacy.

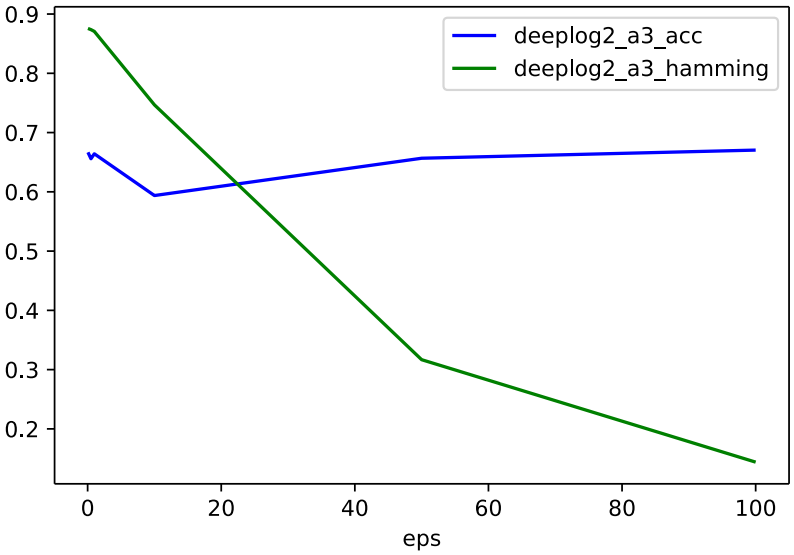

Figure 4.3: Idea 1

One can observe in Fig. 4.3 that as ϵ increases accuracy slowly increases as well, in the mean time, the hamming distance decreases substantially. This last part means the sequences are very similar to the original sequences, however the new ones still have the plausible deniability that differential privacy guarantees with the respective ϵ. Overall, the accuracy remains quite stable but lower than the control.

4.1.1.2 Idea 2 - Seq2Seq and Exponential Mechanism

An *autoencoder Seq2Seq* is trained to generate the same sequence considering a sequence as a database, then the probability distribution learned is used as the *utility function* of the *exponential mechanism*. In this case applies the composition theorem because the utility function depends on the entire sequence. The network receives the whole sequence, obtains a latent vector (LSTM context) and then it generates fake sequences using the vector and an exponential mechanism at the end of the network. The utility function of the exponential mechanism is the probability function at the end of the network. This means for each symbol requires access to the entire vector that represents the sequence. Which means that this method is a centralized model and composition theorem applies. So the final privacy budget spent is ϵ times the length of the sequence.

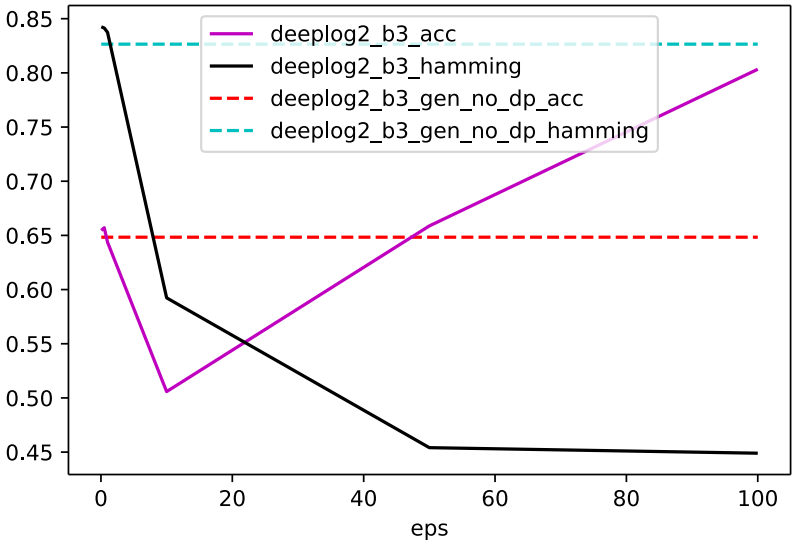

Figure 4.4: Idea 2

Looking at the graph in Fig. 4.4 one could say that for $\epsilon = 100$ the accuracy is high enough and the hamming distance is still above 0.4, however this graph does not contemplate the composition theorem as each sequence length is not easy to represent in a comparable manner. In practice there are sequences up to length 20, which means the final privacy budget for some sequences could be up to 2000.

4.1.1.3 Idea 3 - Seq2Seq and noisy hidden state

An *autoencoder Seq2Seq* is trained to generate the same sequence. However, when it is used to generate fake sequences in test phase, a *laplacian noise vector* with the same shape as the *hidden state* of the latent vector of the network is added to this latent vector, then the vector is normalized with norm 1, then the sequence is generated by sampling from the probability vector at the end of the network. This means there is no composition and we are facing a local model because the sequence without privacy is only queried once. The autoencoder has two parts (encoder-decoder), the encoder sees the entire sequence and outputs the latent vector, then this one is privatized using laplacian mechanism resulting in a privatized latent vector. The decoder takes the privatized vector and generate fake sequences. Due to the postprocessing property the decoder or anything after it cannot leak more privacy.

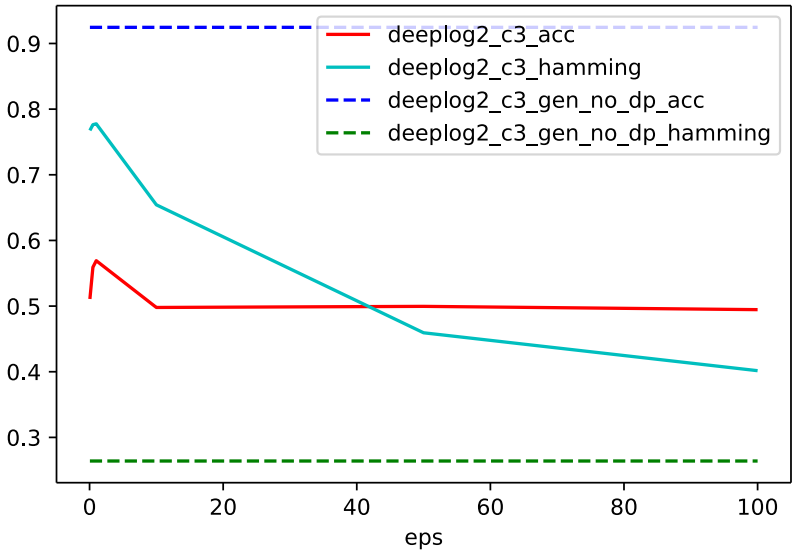

Figure 4.5: Idea 3

In this case, in Fig. 4.5, it can be observed that the sequences generated are more similar to the original ones as ϵ increases, the hamming distance starts high and then drops. This is not good from the privacy perspective. It also can be observed a constant behaviour on how accuracy stall independently of the ϵ around 0.50 or the flip of a coin.

4.1.1.4 Idea 4 - Seq2Seq and noisy output

In this case a *laplacian noise vector* is added to the probability output of the Seq2Seq network. Which means in this case the composition theorem applies. The probability output vector that represent the sequence is queried for every output token of the privatized sequence.

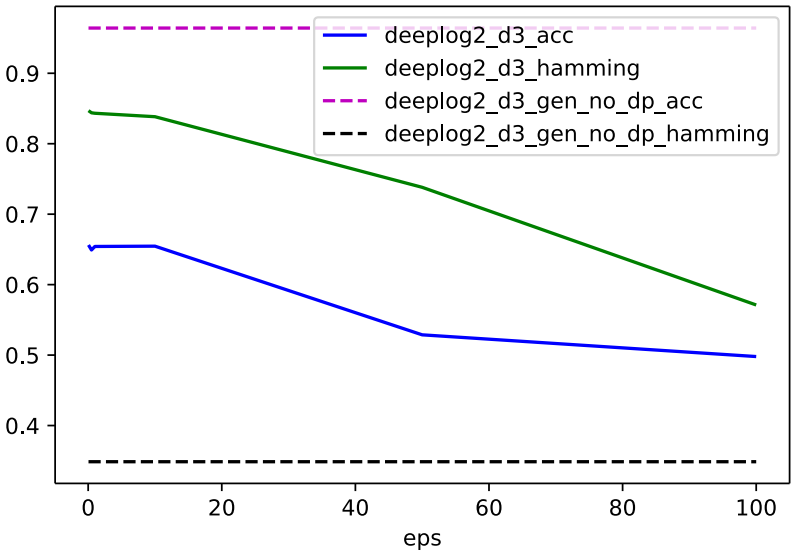

Figure 4.6: Idea 4

However if we observe, in Fig. 4.6 the results, this idea turns out to be a bad privatizer, both hamming distance and classification utility decreases as ϵ increases.

4.1.1.5 Idea 5 - Seq2Seq and Classifier

After many trials and ideas the conclusion drawn was that it is required to preserve as much as possible the features of the sequence that helps solve the utility problem. Otherwise any uncontrolled noisy mechanism with respect to the utility problem will end up in poor results or very high ϵ. So for this last idea a model composed of a *Seq2Seq* followed by a *Binary Classifier* was tested. The idea here is that the Classifier helps train the Seq2Seq so as it develops certain noise tolerance and learn the important features for the utility/ classification problem. The model is also trained adding noise to the latent vector of the Seq2Seq.

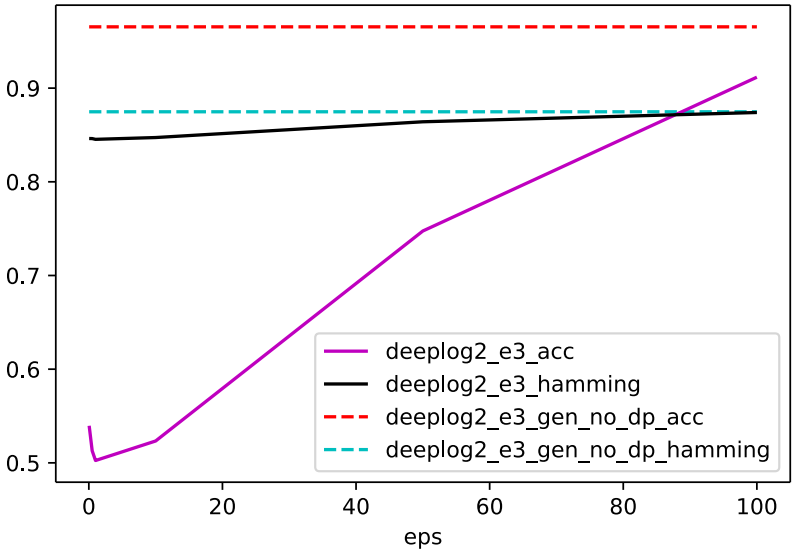

Figure 4.7: Idea 5

As one can see in Fig. 4.7, this idea works, the Seq2Seq generator model not only increases its accuracy as ϵ increases but also the distortion or hamming distance increases. This means that the model learned to generate normal and abnormal sequences that are very different to the original ones, yet the accuracy of the classifier remains very high. In practice, looking at the output sequences one can observe that they were generated using a small subset of the original tokens. This means the network found the right features to generate normal sequences and abnormal sequences using a subset of symbols. From the distortion point of view this is a great result as comparing original sequences to the resultant ones, the sequences are very different and hard to trace back to the originals, and from privacy point of view we still have the guarantees of plausible deniability. As a side note, the classifier used for training the Seq2Seq sequences was tested after training and its resulted in poor performance, however for the language model classifier used for utility purposes the results were the ones wanted as shown in the graph.

4.2 Second approach: PATE

As previously mentioned, Private Aggregation of Teacher Ensembles (PATE) [17, 18], is a technique that enables the training of machine learning models of arbitrary architecture in a way that its privacy guarantees can be described through

differential privacy. The technique proposes to train multiple *teacher* models on sets of sensitive private data, and then use an ensemble of these teachers to guide the training of a *student* model with public, unlabeled data. The student training data is sent through each teacher model to obtain a label prediction, and a noisy aggregation of predictions is used as the training sample label (Fig. 3.1).

The intuition behind PATE's privacy guarantees is that if multiple distinct teacher models agree on an input label, no private data of their training examples was leaked since the conclusion was arrived as a consensus and no particular model is revealing too much information. If, however, there's a strong disagreement among the teachers and the most probable class is likely to be defined by a single model's prediction, the random noise added by the aggregation mechanism will play a bigger role in defining the output, therefore protecting the individual model predictions.

The aggregation mechanisms employed can vary, although the general idea often consists in counting how many teacher models predict each class as being the most probable, adding noise to this count, and then picking the most probable one. The aggregation mechanism employed in this work is the one proposed in [17] which consists in adding noise sampled from a Laplace distribution to the teachers' class prediction count. For a given student training sample x, given the label count of teacher predictions $N_c(x)$ for class c, the aggregation mechanism that outputs the noisy prediction of the ensemble is defined as follows:

$$pred(x) = \arg\max_c \left\{ N_c(x) + Lap\left(\frac{1}{\gamma}\right) \right\} \tag{4.1}$$

4.2.1 Analysis of PATE privacy loss

PATE with the aggregation mechanism given in Eq. 4.1 provides $(2\gamma, 0)$-differential privacy [17]. Therefore, a direct application of DP composition theorem results in that T queries to the teacher ensemble yield $(2T\gamma, 0)$-DP. However, the privacy leakage could be reduced if we accept to reduce the confidence in the DP guarantees, that is, to have $\delta > 0$. The way of doing it is fixing the desired bound $\delta > 0$ on the tail probability of the privacy loss.

To analyze PATE, it is convenient to revisit the formalization of DP given in Sec. 2.4 by defining the *privacy loss* as a random variable. For a given mechanism $\mathcal{M}$, databases $d, d' \in \mathcal{D}$, and output $o \in \mathcal{O}$, the privacy loss at o, denoted $\ell(o)$, is:

$$\ell(o) = \log \frac{P[\mathcal{M}(d) = o]}{P[\mathcal{M}(d') = o]} \tag{4.2}$$

Given $\varepsilon, \delta \in [0, 1]$, $\mathcal{M}$ is said to be (ε, δ)-*differentially private* if for all adjacent databases $d, d' \in \mathcal{D}$ it holds that:

$$P_{o\sim\mathcal{M}(d)}[\ell(o) \geq \varepsilon] \leq \delta \tag{4.3}$$

To simplify the notation, we denote L the random variable distributed as $\mathcal{M}(d)$ whose values are given by evaluating ℓ at outcomes sampled from $\mathcal{M}(d)$, and write:

$$P[L \geq \varepsilon] \leq \delta \tag{4.4}$$

This definition is equivalent to the one given in Sec. 2.4.

Now, the analysis method consists in finding the *smallest* ε that satisfies Eq. 4.4. To do this, the moment generating function method is applied to derive the following bound on the tail probability:

$$P[L \geq \varepsilon] \leq \exp(\phi_L(\lambda) - \lambda\varepsilon) \tag{4.5}$$

where $\phi_L(\lambda)$ is the logarithm of the moment generating function M_L of L:

$$\phi_L(\lambda) = \log M_L(\lambda) = \log \mathbb{E}[\exp(\lambda L)] \tag{4.6}$$

This means that $P[L \geq \varepsilon]$ is ensured to be smaller than any δ such that:

$$\exp(\phi_L(\lambda) - \lambda\varepsilon) \leq \delta \tag{4.7}$$

Now, the above equation can be rewritten as follows:

$$\frac{1}{\lambda}\Big(\phi_L(\lambda) - \log\delta\Big) \leq \varepsilon \tag{4.8}$$

Hence, by fixing δ, it can be obtained the *minimum* bound of the privacy loss which could be ensured with such δ:

$$\varepsilon^* = \min_{\lambda} \frac{1}{\lambda}\Big(\phi_L(\lambda) - \log\delta\Big) \tag{4.9}$$

It follows from [17] that PATE with the aggregation mechanism defined in Eq. 4.1, satisfies:

$$\phi_L(\lambda) \leq 2\gamma^2\lambda(\lambda + 1) \tag{4.10}$$

By the composability theorem of [17], we have that the moment generating function of the mechanism obtained by applying PATE T times is $T\phi_L(\lambda)$. Therefore, it follows that after T queries, we can get a data independent privacy guarantee of $(\varepsilon^*_{ind}, \delta)$, where:

$$\varepsilon^*_{ind} = \min_{\lambda} \frac{1}{\lambda}\Big(2T\gamma^2\lambda(\lambda + 1) - \log\delta\Big) \tag{4.11}$$

The quantity ε^*_{ind} is called the *data independent epsilon*. Fig. 4.8 gives an example of the data independent epsilon for $\gamma = 0.05$, $\delta = 10^{-5}$ and $T = 1000$, computed using Wolfram Alpha.

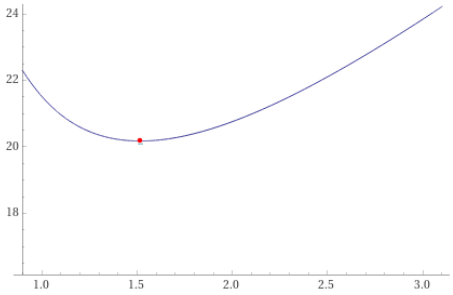

Figure 4.8: Graph of $\lambda^{-1}(2T\gamma^2\lambda(\lambda+1) - \log\delta)$. Data independent epsilon is $\varepsilon^*_{ind} \simeq 20.1743$ at $\lambda \simeq 1.51743$.

Indeed, the epsilon bound on the privacy loss could be made smaller provided we bring into the picture the actual predictions delivered by the ensemble of teachers. This bound is called the *data dependent epsilon* [17]. A tighter bound on the moment generating function could be computed if we take into account the fact that when quorum among teachers is strong, the majority outcome has overwhelming likelihood, so the privacy loss is smaller when this outcome occurs. The following theorem, proved in [17], provides a *data-dependent* bound on ϕ_L as a function ψ of the most probable predicted class c^* of the teacher ensemble:

$$\phi_L(\lambda) \leq \psi_L(\lambda; P[\mathcal{M}(d) \neq c^*]) \tag{4.12}$$

In order for this result to be applied, an upper bound of $P[\mathcal{M}(d) \neq c^*]$ is computed in [17]. For the sake of readability, we omit the details here. Thanks to this bound that depends on the teacher agreement, a tighter tail bound can be computed for specific responses of the ensemble to a sequence of T queries of the student:

$$\varepsilon^*_{dep} = \min_{\lambda} \frac{1}{\lambda}\left(\psi_L(\lambda) - \log\delta\right) \tag{4.13}$$

4.2.2 Sensitive student data scenario

This thesis is concerned with the case where the student does not have access to a public dataset but it has its own private data. In this scenario, the student is not able or not willing to share its private data with the teacher ensemble or

trusted curator (Trusted Curator A). For this case, this work proposes a framework where the student relies in another curator, called Trusted Curator B. The role of Trusted Curator B is to privatize student data by using a randomized mechanism, e.g., Laplace Mechanism, granting the student differential privacy guarantees over its data. Here, Trusted Curator A provides a centralized privacy model, which protects data used to train teachers, while Trusted Curator B provides a local privacy model, by granting DP guarantees for *each* individual data point in the student organization sent to Trusted Curator A to be labelled by the teacher ensemble.

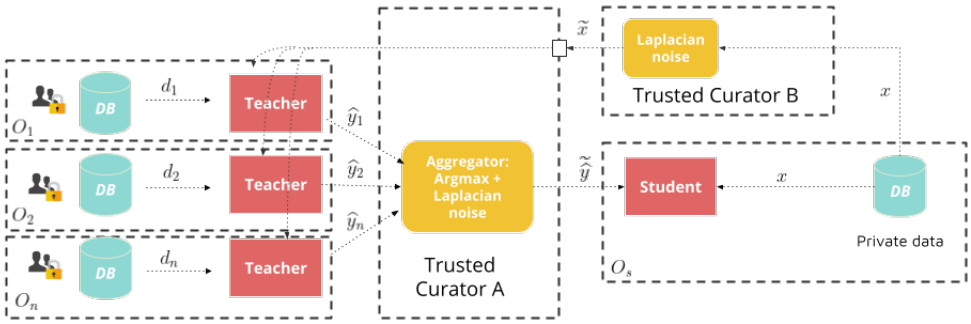

Figure 4.9: PATE with protected student's data.

4.2.3 Experimental results

This section describes the experimental setup and apply the approach presented in the previous section to two case studies of different domains, security and health [29]. Following the same strategy as the original PATE paper [17], teacher models were trained and used to generate labels for the student training samples, using an ensemble based on a Laplace aggregation mechanism with $\gamma = 0.05$. Every teacher $i \in [1, n]$ is presented with a labeled independent dataset $d_i = (x_i, y_i)$, which is used for training. The student is presented with an unlabeled independent dataset x. Trusted Curator B privatizes student data with a Laplace mechanism with distribution $Lap(1/\rho)$. To analyze this setting different values of ρ are used. In both case studies, database elements are vectors of real numbers having an

$l1$-norm equal to 1. Thus, the distance $\|\cdot\|$ is $l1$-norm. Moreover, the fact that vectors have a norm equal to 1 ensures that $\|\cdot\|$-*sensitivity* of the Laplace mechanism applied by Trusted Curator B is 2, resulting in a $(2\rho, 0)$-DP mechanism. For each value of ρ, ten student models were trained, each one on a different random sample of student data points labelled by the teacher ensemble. Each random sample was privatized by Trusted Curator B with noise from a Laplace distribution with scale ρ. Both student and teachers are assumed to have access to a labelled validation dataset, which is used with the only purpose of evaluating performance and privacy loss metrics in the context of this work. In a real world scenario such validation data may not be available. However, it does not pose any drawback to the applicability of the present approach.

4.2.3.1 Malicious Web Requests Detection

In order to classify web requests, a dataset of 651,602 labeled requests [30, 31] was assembled from several public datasets [32, 33, 34]. To construct the feature vector to train the networks, only the URI of each web request is analyzed. Each URI is tokenized in uni-grams following a bag-of-words approach. For each URI, the values of the uni-grams are computed using term frequency-inverse document frequency (TF-IDF) [35]. Each URI is represented by an $l1$-normalized vector composed of the 500 most frequent tokens across the entire dataset.

An ensemble of 250 teacher models was trained and used to generate labels for the student training samples, using the Laplace aggregation mechanism. Every teacher was trained with 930 datapoints and the validation dataset contained 500 samples. Given the unbalanced distribution of the training set where 95% of samples are not malicious, a threshold of 0.5 to split the model's output between positive and negative samples might yield poor accuracy results. Therefore, the receiver operating characteristic curve is calculated for a subset of samples, and the threshold that maximizes the difference between the true positive rate and false positive rate is picked as the best one. Every teacher used 800 samples for calculating the best threshold for considering the classifier's output as a positive prediction. For the student, random samples of 1,000 data points were used for training and 200 for calculating the optimal threshold. 5,000 data points were used for validation.

A simple fully connected neural network architecture was used for both the teacher and student models, with a single real-valued output (see Fig. 4.10).

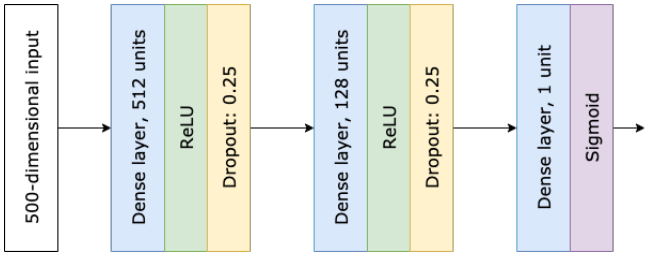

Figure 4.10: Neural network architecture used for teachers and student in Web Request example.

The data dependent privacy loss of the teacher ensemble is computed for every case as described above for $\delta = 10^{-5}$. The data independent privacy loss of the teacher ensemble computed using Wolfram Alpha resulted in a value of 20.1743.

As presented in Figs. 4.12-4.11, the median of both TPR and TNR performance metrics observed is similar for all values of ρ with relatively low dispersion in most cases. This shows that the predictive capacity of a student which privatizes its data is close to the one observed in student models trained with non-privatized data. That is, no significant loss in predictive value was evidenced in the experiments by privatizing student data.

On the other hand, Fig. 4.13 presents the data dependent privacy loss obtained for the different values of ρ. The dashed line in red represents the data independent privacy loss ε^*_{ind}. As it can be seen, the data dependent privacy loss ε^*_{dep} observed in some cases turned out to be higher than the one of the experiment without applying noise to student data. Actually, it happened to be even higher than the data independent privacy loss ε^*_{ind} in one case.

4.2.3.2 Cardiopathy Classification

The case study analyzed in this experiment consists in cardiopathy classification based on electrocardiogram (ECG) data. The ECG dataset contains a number of 109,446 ECG beats [36] extracted from ECG signals from the MIT-BIH Arrithmia Database [37]. The sampling frequency of each beat is 125Hz, and they can be categorized in one of five classes.

For simplicity, a multi layer perceptron architecture was used for both for teacher and student models, see Fig. 4.14. The number of teachers in the ensemble for this example was 200. Every teacher was trained with 5,000 data points. The

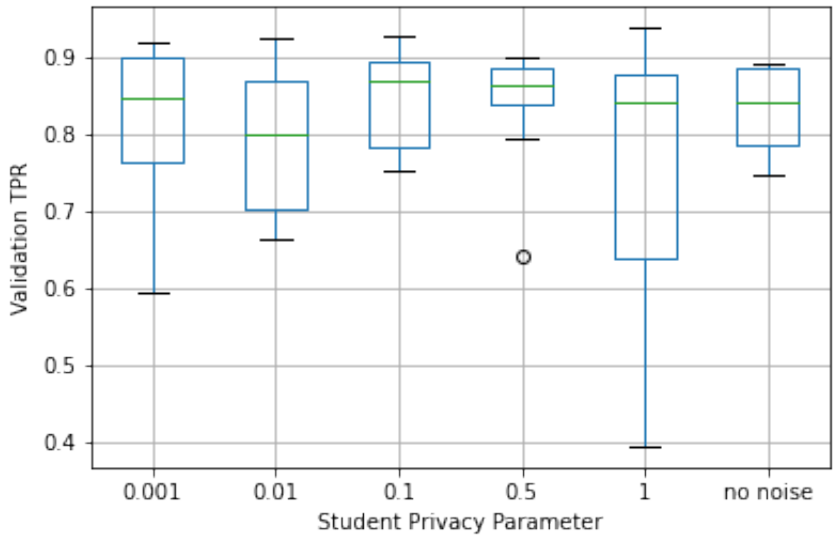

Figure 4.11: Validation TPR by student privacy parameter ρ in Web Requests dataset.

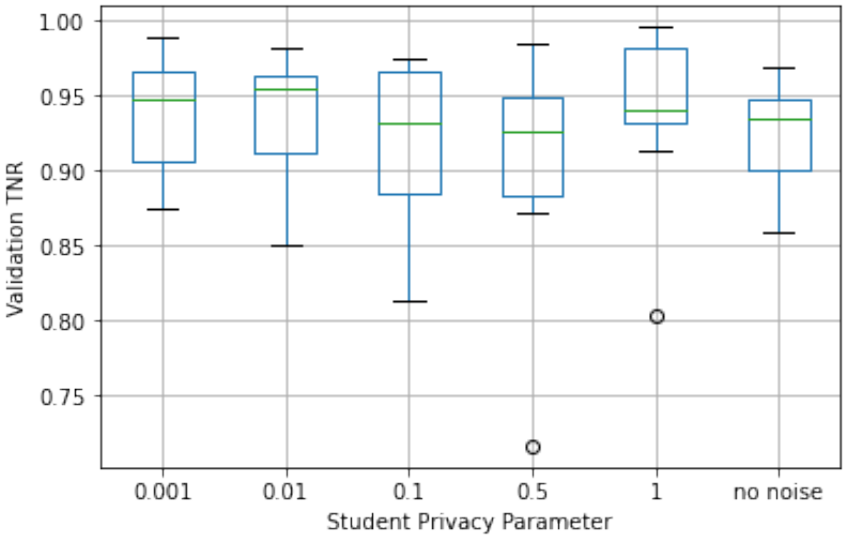

Figure 4.12: Validation TNR by student privacy parameter ρ in Web Requests dataset.

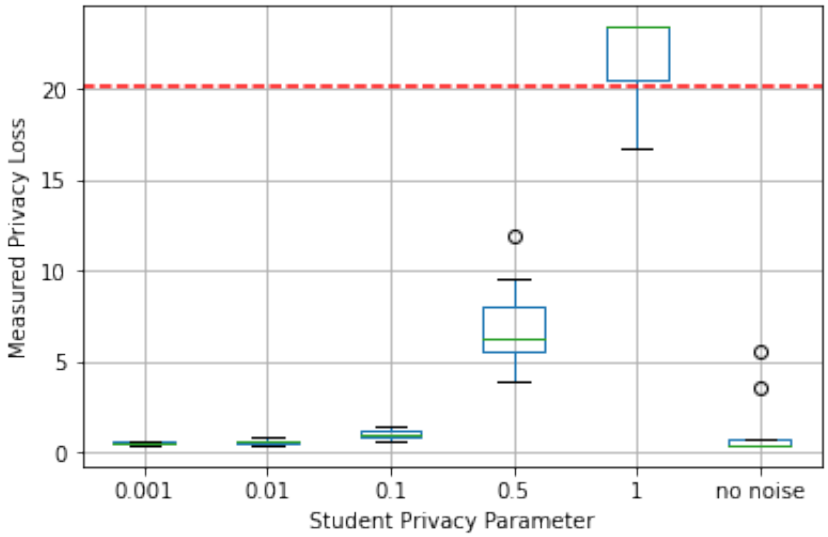

Figure 4.13: Privacy loss by student privacy parameter ρ in Web Requests dataset.

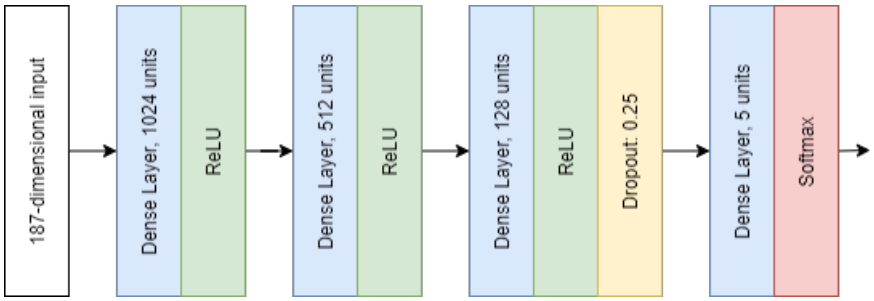

Figure 4.14: Neural network architecture used for teachers and student in ECG example.

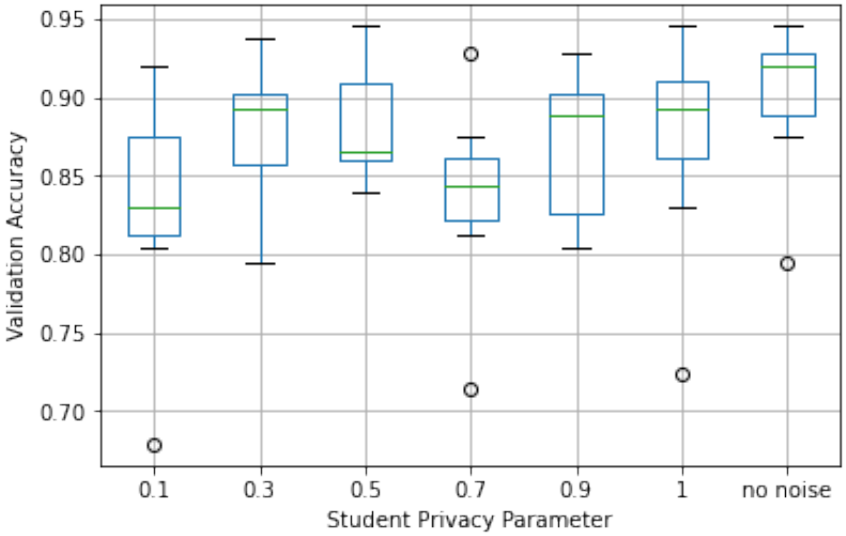

Figure 4.15: Validation accuracy by student privacy parameter ρ in ECG dataset.

validation dataset contained 500 samples. For the student, 900 data points were used for training and 100 for validation. A confidence parameter $\delta = 10^{-6}$ was used. The computed data independent privacy loss was $\varepsilon^*_{ind} = 20.2696$.

In Fig. 4.15 the accuracy observed in the validation set for different ρ values is plotted. As it can be seen, the median accuracy for all cases is not significantly smaller to the one observed in the case of no noise, with a reduction of about 7-8%. In particular, it becomes closer to the latter for larger values of ρ.

In Fig. 4.16 the data dependent privacy loss ε^*_{dep} for different ρ values is visualized. The dashed line in red represents the data independent privacy loss ε^*_{ind}. As it can be observed in Table 4.1, ε^*_{dep} presents more variability when the student does not privatize its data, with the worst case IQR for students with privatized data is 0.32 for $\rho = 0.1$, while the no-noise example presents a very large IQR of 13.12. At the same time, the median ε^*_{dep} for every ρ different to the no-noise version is larger than three times the median of the no-noise case, providing further empirical evidence of the phenomenon observed in the previous example.

4.2.3.3 Some comments about this approach

On one hand, the experiments showed that the introduction of noise in student data yielded no important reductions in predictive model performance.

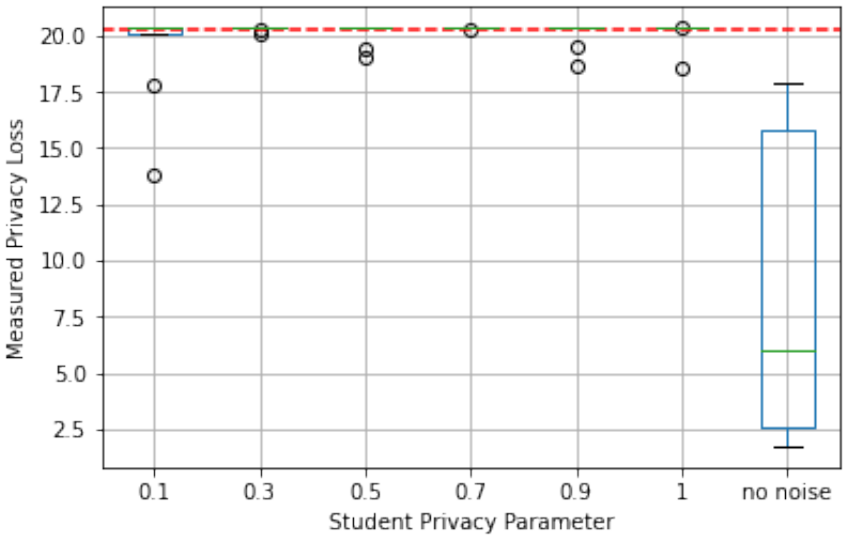

Figure 4.16: Privacy loss by student privacy parameter ρ in ECG dataset.

ρ	ε^*_{dep} **median**	ε^*_{dep} **IQR**
0.1	20.36	0.32
0.3	20.41	0.00
0.5	20.41	0.032
0.7	20.41	0.00
0.9	20.41	0.00
1	20.41	0.00043
no noise	5.96	13.15

Table 4.1: Median and IQR of data dependent privacy loss for student privacy parameter ρ

On the other, those analyses brought to light some issues with data dependent privacy loss. First, it was observed to suffer from high variance. Second, it presented evidence of being quite sensible to noise in data. It even resulted to be higher than the data independent privacy loss sometimes.

This phenomenon was not described prior to this work. It unveils a potential weakness of PATE as noise could be used as a means for tampering with the actual privacy loss.

5 Tool: DP-GEM

5.1 General description

DP-GEM is a Python tool created to replicate the experiments carried on in this thesis. The need of such tool is due to the complicated pipelines that brings transforming one database to other database by trying different privatization mechanisms, with different parameters and furthermore, testing the results against a control test for utility study.

The tool allows to define a series of modules that are executed in chain. The output of a module can be used as input of the next module. Moreover, each module can be run with many different parameters, named trials, which results in many different outputs. For this reason, modules may have sub-modules and those latter ones will be run as many times as many outputs given by the trials of the main module. This level of nesting goes on as required.

DP-GEM is flexible enough to allow for the definition of the experiments in a .json file, which helps with the replication requirement. This means, the modules and sub-modules, it's source codes and the parameters can be defined in a json format, including the number of trials per module, the nesting and the outputs logs and files.

The tool is also integrated with Wandb[38] for logging and saving many of the intermediate and final results in such service. And finally DP-GEM can be integrated with two Deep Learning framework Keras and PyTorch, as well as any functionality provided by scikit-learn package.

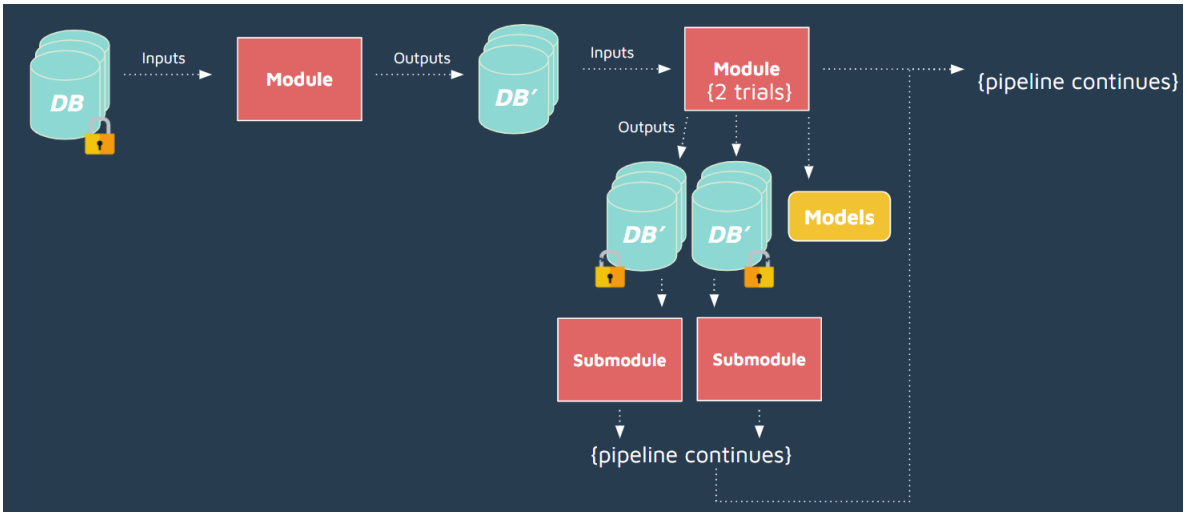

Figure 5.1: Structure of nested modules

5.2 Example

One example consists in three main modules.

1. Data preparation: In this module input files are taken merged, randomized and split into train, test and validation. This module was also used to tweak the data so as it works as an input for future modules in the pipeline. The output of this module are different data files for train, test and validation.

2. Control test: This module worked as a control test without any privacy mechanism. It takes the train, test and validation files as inputs and trains a classifier with training data, finds a proper threshold with the validation data and finally tests it against test data. The principal output of this module are different metrics such as a confusion matrix, accuracy, etc.

3. Generator: This is a module with nested modules and different trials. In creation phase it trains a generator neural network using train data. Then for each trial, with different privatization parameters, it privatizes train, test and validation datasets. Finally it runs different sub-modules for each privatized output dataset. The sub-modules include a similar classifier or the same as the control test and a similarity test module, which compares the original data with the privatized one to measure distortion.

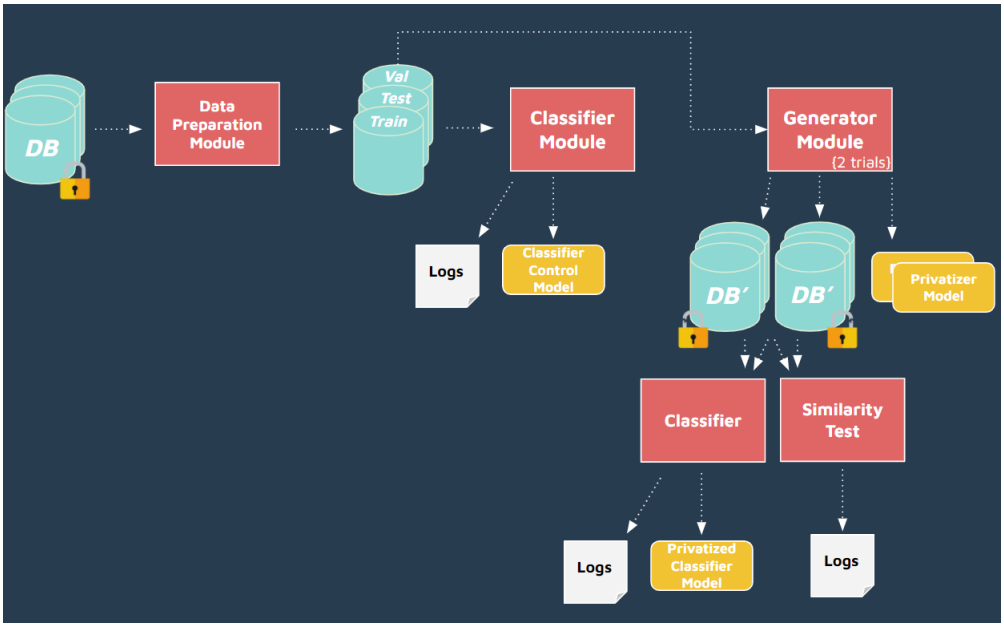

Figure 5.2: Example of DP-GEM Use with DP 2 trials

6 Conclusions

This work not only studied both fields of research, Deep Learning and Differential Privacy but also the synergy between them. Two aspects, privacy and utility has been studied with different techniques and case studies.

A dataset generator algorithm differential privacy compliant was successfully created in order to release a synthethic dataset of logs sequences in replacement of the original dataset, protecting its privacy and value. This generated dataset contains the right information required for a classifier task of anomaly detection. Moreover, an extension of PATE was successfully carried on and tested. Last but not least, a framework for machine learning experiments was also created with a high degree of freedom. It allowed to run the many different experiments for training models, applying differential privacy mechanisms, generating data, carry out classification tasks, similarity tests, among others.

For future work, the hypothesis that in order for an mechanism to return good results, it needs to identify which features contribute to the desired utility problem while identifying those which contribute to the privacy preserving problem and finally which features contribute to both, it can be studied more in depth. In that last case scenario a trade-off needs to be taken into consideration. Ideally one would like to learn which features have mutual information with the utility problem presented, agree on a trade-off for those and privatize the rest.

7 Bibliography

[1] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3-4, pp. 211–407, 2014. [Online]. Available: https://doi.org/10.1561/0400000042

[2] "General data protection regulation," https://gdpr-info.eu/, accessed: 2021-05-10.

[3] "The world's most valuable resource is no longer oil, but data," https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data, accessed: 2021-05-10.

[4] "Data: Your most ignored and valuable asset," https://www.forbes.com/sites/forbesagencycouncil/2018/02/12/data-your-most-ignored-and-valuable-asset/?sh=7d0cd6ad715b, accessed: 2021-05-10.

[5] "Treating information as an asset," https://www.gartner.com/smarterwithgartner/treating-information-as-an-asset/, accessed: 2021-05-10.

[6] "Ng, a. (2017). machine learning yearning." http://www.mlyearning.org.

[7] N. Gruschka, V. Mavroeidis, K. Vishi, and M. Jensen, "Privacy issues and data protection in big data: A case study analysis under GDPR," in *IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10-13, 2018*, N. Abe, H. Liu, C. Pu, X. Hu, N. K. Ahmed, M. Qiao, Y. Song, D. Kossmann, B. Liu, K. Lee, J. Tang, J. He, and J. S. Saltz, Eds. IEEE, 2018, pp. 5027–5033. [Online]. Available: https://doi.org/10.1109/BigData.2018.8622621

[8] B. Chen, D. Kifer, K. LeFevre, and A. Machanavajjhala, "Privacy-preserving data publishing," *Found. Trends Databases*, vol. 2, no. 1-2, pp. 1–167, 2009. [Online]. Available: https://doi.org/10.1561/1900000008

[9] J. Kim, J. Kim, H. L. Thi Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," in *2016 International Conference on Platform Technology and Service (PlatCon)*, 2016, pp. 1–5.

[10] L. Bontemps, V. L. Cao, J. McDermott, and N. Le-Khac, "Collective anomaly detection based on long short-term memory recurrent neural networks," in *Future Data and Security Engineering - Third International Conference, FDSE 2016, Can Tho City, Vietnam, November 23-25, 2016, Proceedings*, ser. Lecture Notes in Computer Science, T. K. Dang, R. R. Wagner, J. Küng, N. Thoai, M. Takizawa, and E. J. Neuhold, Eds., vol. 10018, 2016, pp. 141–152. [Online]. Available: https://doi.org/10.1007/978-3-319-48057-2_9

[11] N. N. Thi, V. L. Cao, and N. Le-Khac, "One-class collective anomaly detection based on lstm-rnns," *Trans. Large Scale Data Knowl. Centered Syst.*, vol. 36, pp. 73–85, 2017. [Online]. Available: https://doi.org/10.1007/978-3-662-56266-6_4

[12] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21 954–21 961, 2017.

[13] A. Wood, M. Altman, A. Bembenek, M. Bun, M. Gaboardi, J. Honaker, K. Nissim, D. R. OBrien, T. Steinke, and S. Vadhan, "Differential privacy: A primer for a non-technical audience," *Vanderbilt Journal of Entertainment & Technology Law*, vol. 21, no. 1, pp. 209–275, 2018. [Online]. Available: http://www.jetlaw.org/journal-archives/volume-21/volume-21-issue-1/differential-privacy-a-primer-for-a-non-technical-audience/

[14] M. Abadi, A. Chu, I. J. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM, 2016, pp. 308–318. [Online]. Available: https://doi.org/10.1145/2976749.2978318

[15] N. Carlini, C. Liu, Úlfar Erlingsson, J. Kos, and D. Song, "The secret sharer: Evaluating and testing unintended memorization in neural networks," 2019.

[16] M. Du, R. Jia, and D. Song, "Robust anomaly detection and backdoor attack detection via differential privacy," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=SJx0q1rtvS

[17] N. Papernot, M. Abadi, Ú. Erlingsson, I. J. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. [Online]. Available: https://openreview.net/forum?id=HkwoSDPgg

[18] N. Papernot, S. Song, I. Mironov, A. Raghunathan, K. Talwar, and Ú. Erlingsson, "Scalable private learning with PATE," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=rkZB1XbRZ

[19] C. M. Bowen and J. Snoke, "Comparative study of differentially private synthetic data algorithms from the nist pscr differential privacy synthetic data challenge," 2020.

[20] H. Bo, S. H. H. Ding, B. C. M. Fung, and F. Iqbal, "Er-ae: Differentially-private text generation for authorship anonymization," 2019.

[21] B. Weggenmann and F. Kerschbaum, "Syntf: Synthetic and differentially private term frequency vectors for privacy-preserving text mining," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, K. Collins-Thompson, Q. Mei, B. D. Davison, Y. Liu, and E. Yilmaz, Eds. ACM, 2018, pp. 305–314. [Online]. Available: https://doi.org/10.1145/3209978.3210008

[22] M. Malekzadeh, R. G. Clegg, and H. Haddadi, "Replacement autoencoder: A privacy-preserving algorithm for sensory data analysis," in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation, IoTDI 2018, Orlando, FL, USA, April 17-20, 2018*. IEEE Computer Society, 2018, pp. 165–176. [Online]. Available: https://doi.org/10.1109/IoTDI.2018.00025

[23] N. C. Abay, Y. Zhou, M. Kantarcioglu, B. M. Thuraisingham, and L. Sweeney, "Privacy preserving synthetic data release using deep learning," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part I*, ser. Lecture Notes in Computer Science, M. Berlingerio, F. Bonchi, T. Gärtner, N. Hurley, and G. Ifrim, Eds., vol. 11051. Springer, 2018, pp. 510–526. [Online]. Available: https://doi.org/10.1007/978-3-030-10925-7_31

[24] L. Frigerio, A. S. de Oliveira, L. Gomez, and P. Duverger, "Differentially private generative adversarial networks for time series, continuous, and discrete open data," in *ICT Systems Security and Privacy Protection - 34th IFIP TC 11 International Conference, SEC 2019, Lisbon, Portugal, June 25-27, 2019, Proceedings*, ser. IFIP Advances in Information and Communication Technology, G. Dhillon, F. Karlsson, K. Hedström, and A. Zúquete, Eds., vol. 562. Springer, 2019, pp. 151–164. [Online]. Available: https://doi.org/10.1007/978-3-030-22312-0_11

[25] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *CoRR*, vol. abs/1802.06739, 2018. [Online]. Available: http://arxiv.org/abs/1802.06739

[26] K. Kalantari, L. Sankar, and A. D. Sarwate, "Robust privacy-utility tradeoffs under differential privacy and hamming distortion," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 11, pp. 2816–2830, 2018. [Online]. Available: https://doi.org/10.1109/TIFS.2018.2831619

[27] N. Fernandes, L. Kacem, and C. Palamidessi, "Utility-preserving privacy mechanisms for counting queries," in *Models, Languages, and Tools for Concurrent and Distributed Programming - Essays Dedicated to Rocco De Nicola on the Occasion of His 65th Birthday*, ser. Lecture Notes in Computer Science, M. Boreale, F. Corradini, M. Loreti, and R. Pugliese, Eds., vol. 11665. Springer, 2019, pp. 487–495. [Online]. Available: https://doi.org/10.1007/978-3-030-21485-2_27

[28] "Deeplog dataset," https://github.com/wuyifan18/DeepLog, accessed: 2021-08-12.

[29] S. Yovine, F. Mayr, S. Sosa, and R. Visca, "An assessment of the application of private aggregation of ensemble models to sensible data," Submitted to MAKE, August 2021.

[30] D. Biardo, G. González, and S. Lanzotti, "Análisis y desarrollo de modelos predictivos con redes neuronales para web application firewall," Engineering Thesis, Universidad ORT Uruguay, 2020.

[31] S. Sosa, "Application of private aggregation teacher ensembles framework for malicious web request detection," Engineering Thesis, Universidad ORT Uruguay, 2021.

[32] F. Ahmad, "Using machine learning to detect malicious urls," https://github.com/faizann24/Using-machine-learning-to-detect-malicious-URLs, 2017.

[33] LIRMM, "Analyzing web traffic: Ecml/pkdd 2007 discovery challenge," http://www.lirmm.fr/pkdd2007-challenge/, 2007.

[34] C. Torrano-Gimenez, A. Perez-Villegas, and G. Alvarez, "An anomaly-based approach for intrusion detection in web traffic," *Journal of Information Assurance and Security*, vol. 5, no. 4, pp. 446–454, 2010.

[35] C. Salton, G.; Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management*, vol. 24, no. 5, pp. 513—523, 1988.

[36] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, "Ecg heartbeat classification: A deep transferable representation," in *2018 IEEE International Conference on Healthcare Informatics (ICHI)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2018, pp. 443–444. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICHI.2018.00092

[37] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[38] L. Biewald, "Experiment tracking with weights and biases," 2020, software available from wandb.com. [Online]. Available: https://www.wandb.com/