Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-16768-117991

Hlib Kokin

Hyperparameter Tuning in Privacy-Preserving Machine Learning

Bachelor thesis

Thesis supervisor: Ing. Oleksandr Lytvyn

January 2024

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-16768-117991

Hlib Kokin

Hyperparameter Tuning in Privacy-Preserving Machine Learning

Bachelor thesis

Study Program: Informatics

Degree Course: Informatics

Department: Institute of Informatics and Software Engineering, FIIT STU, Bratislava

Thesis supervisor: Ing. Oleksandr Lytvyn

January 2024

Annotation

Slovak University of Technology Bratislava
Faculty of Informatics and Information Technologies
Degree Course: Informatics

Author: Hlib Kokin
Bachelor's Thesis: Hyperparameter Tuning in Privacy-Preserving Machine Learning
Supervisor: Ing. Oleksandr Lytvyn
January 2024

The research for this bachelor's thesis focuses on the state of differential privacy technologies today, which is used to protect sensitive data privacy in machine learning. The following steps are taken in order to complete the work. An introduction to machine learning, deep learning, hyperparameter tuning, and differential privacy is provided in the first step. This prompts an examination of the frameworks and instruments that are currently accessible for enabling differential privacy-based privacy preservation solutions. In the second step, the models are set up and the motivation to use them is discussed, along with the description of the sensitive heart disease dataset and the tensorflow-privacy specifications to use for the experiments. The process that has been put into practice then yields outcomes for every model that has been selected. These outcomes are examined to determine which hyperparameter configurations are optimal and which are unsatisfactory; the optimal configurations are those that ensure privacy while maintaining a high degree of accuracy. The evaluation of the hyperparameter tuning patterns, discovered during the experiments, provides insight into the best method for setting up the model with maintaining accuracy and privacy.

Anotácia

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií
Študijný program: Informatika

Autor: Hlib Kokin
Bakalárska práca: Ladenie ML hyperparametrov so zachovaním súkromia
Vedúci bakalárskeho projektu: Ing. Oleksandr Lytvyn
January 2024

Výskum tejto bakalárskej práce sa zameriava na súčasný stav technológií diferencovaného súkromia (Differential Privacy), ktoré sa používajú na ochranu súkromia citlivých údajov v strojovom učení. Na dokončenie práce sú vykonané nasledujúce kroky. V prvom kroku je uvedený úvod do strojového učenia, hlbokého učenia, ladenia hyperparametrov a diferenciálneho súkromia. To podnecuje skúmanie rámcov a nástrojov, ktoré sú v súčasnosti dostupné na umožnenie riešení založených na ochrane súkromia na základe diferenciálnej ochrany. V druhom kroku sú nastavené modely a diskutuje sa o motivácii na ich použitie spolu s opisom citlivého súboru údajov o ochoreniach srdca a špecifikáciami tensorflow-privacy, ktoré sa majú použiť na experimenty. Postup, ktorý bol zavedený do praxe, potom prináša výsledky pre každý vybraný model. Tieto výsledky sa skúmajú s cieľom určiť, ktoré konfigurácie hyperparametrov sú optimálne a ktoré sú nevyhovujúce; optimálne konfigurácie sú tie, ktoré zabezpečujú súkromie pri zachovaní vysokej miery presnosti. Vyhodnotenie modelov nastavenia hyperparametrov, ktoré sa zistili počas experimentov, poskytuje prehľad o najlepšej metóde výberu hyperparametrov a nastavenia modelu s cieľom zachovať presnosť a súkromie modelu.

Strojové učenie umožňuje dosahovať významné výsledky vo viacerých oblastiach. Použiteľnosť výsledkov priamo závisí na kvalitu a reprezentatívnosť dát použitých na tréningovej fáze, ktoré často môžu obsahovať súkromne informácie. Diferenciálne súkromie (angl. Differential Privacy) je jedným zo spôsobov na ochranu súkromia údajov pre strojové učenie. Analyzujte problematiku nastavenia hyperparametrov v strojovom učení s použitím diferenciálneho súkromia. Preskúmajte dostupné riešenia. Implementuje modul na trénovanie vami zvolených modelov strojového učenia na dátach s aplikovaním diferenciálneho súkromia. Vyhodnoťte úspešnosť natrénovaného modelu strojového učenia pomocou dostupných metrík s ohľadom na bilanciu medzi mierou ochrany súkromia a praktickou použiteľnosťou dat. Literatúra: - Papernot, Nicolas, and Thomas Steinke. "Hyperparameter tuning with renyi differential privacy." arXiv preprint arXiv:2110.03620 (2021) - Priyanshu, Aman, et al. "Efficient Hyperparameter Optimization for Differentially Private Deep Learning." arXiv preprint arXiv:2108.03888 (2021). - Abadi, Martin, et al. "Deep learning with differential privacy." Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016.

Acknowledgements

I would like to express my deepest appreciation to my project advisor, Ing. Lytvyn Oleksandr, for being my guide, his patience and feedback.

Contents

Chapter 1

Introduction

Chapter 2

Objectives

- Figure out hyperparameters influence on ε value.
- Find out if and how ε affects accuracy of the model.
- Distinguish patterns to stick to to efficiently set up differentially private model.

Chapter 3

State Of The Art

3.1 Machine Learning & Deep Learning

3.1.1 Introduction

Machine learning (ML) strives to independently identify significant connections and patterns from instances and observations. Advances in ML have paved the way for the recent development of intelligent systems that exhibit cognitive capabilities resembling those of humans. These systems have become integral in both our professional and personal lives. They play a crucial role in shaping the interconnected dynamics of electronic markets by improving decision-making processes for increased productivity, engagement, and employee retention in businesses. Furthermore, trainable assistant systems have the ability to customize themselves according to individual user preferences, while trading agents are causing disruptions in traditional finance markets [15].

The success of these systems in addressing intricate issues, commonly known as Artificial intelligence (AI) (AI), depends on analytical models that produce pre-

dictions, rules, responses, recommendations, or comparable results. Historically, attempts to construct these analytical models required the explicit programming of established relationships, procedures, and decision logic into intelligent systems using manually designed rules, such as expert systems for medical diagnoses. Due to the convenience offered by contemporary programming frameworks, ready access to data, and the widespread availability of computational power, analytical models are now more frequently created through a widely recognized approach referred to as ML. ML alleviates the workload on humans by removing the necessity to explicitly articulate and formalize knowledge in a format that machines can comprehend. This, in turn, streamlines the development of intelligent systems, making it more efficient [15].

Over the past few decades, the domain of ML has witnessed substantial advancements in improving sophisticated learning algorithms and applying effective preprocessing techniques. A notable accomplishment in this progression is the development of artificial neural networks (ANNs) towards more complex architectures, ultimately resulting in enhanced learning capabilities known as Deep learning (DL) [15].

3.1.2 Difference between ML & DL

Three core categories of ML exists: supervised learning, unsupervised learning, and reinforcement learning. Table 3.1 contains a description of each of the three categories. The ML field provides numerous classes of algorithms tailored to specific learning tasks, with each class having multiple specifications and variations. These include regression models, instance-based algorithms, decision trees, Bayesian methods, and artificial neural networks (ANNs) [15].

The versatility of the artificial neural network family is noteworthy, as its adaptable

structure renders it suitable for a broad spectrum of applications within all three categories of machine learning. Mimicking the information processing principle observed in biological systems, artificial neural networks (ANNs) are composed of interconnected processing units known as artificial neurons. Like synapses in the brain, the connections between neurons relay signals, and the intensity of these signals can be adjusted by weights that undergo continuous modification throughout the learning process. Signals are processed by successive neurons only when they exceed a particular threshold, as dictated by an activation function. Neurons are commonly arranged into networks featuring distinct layers: an input layer that receives data input (such as product images), an output layer responsible for generating the final result (for instance, product categorization), and zero or more hidden layers tasked with learning non-linear mappings between the input and output [15].

DL proves particularly effective in domains with large and high-dimensional data, surpassing shallow ML algorithms for tasks involving text, image, video, speech, and audio data. However, for low-dimensional input data, especially with limited training data availability, shallow ML can still yield superior and more interpretable results compared to those generated by Deep neural network (DNN) [15].

Basic artificial neural networks and various machine learning algorithms fall into the category of shallow machine learning as they lack such capabilities. Since there is no precise boundary between the two concepts in the literature, we represent it with a dashed line in the Fig 3.1. While certain shallow machine learning algorithms are inherently interpretable to humans and are considered "white boxes," the decision-making process of most advanced machine learning algorithms is inherently untraceable unless explicitly explained. Therefore, it inherently consti-

tutes a "black box" [15].

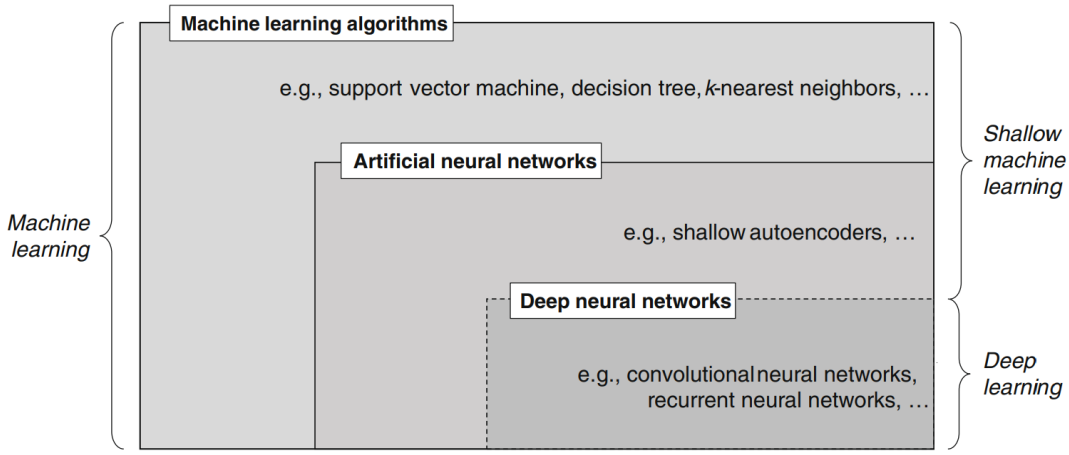

Figure 3.1: Landscape of Machine Learning concepts and methods [15].

Type	**Description**
Reinforcement learning	In a reinforcement learning system, rather than providing input and output pairs, we outline the current state of the system, establish a goal, present a list of actions that are permitted along with the environmental constraints that they are subject to, and then allow the machine learning model to go through the process of achieving the goal on its own. In order to optimize a reward, this is accomplished through the use of the trial-and-error principle. There has been a significant amount of success obtained by reinforcement learning models in confined contexts such as games. Their usefulness, on the other hand, extends to multi-agent systems, such as electronic marketplaces with several agents.

Supervised learning	A training dataset that includes instances for the input and labeled answers or target values for the output is required for supervised learning. This dataset must also include instances for the input. Taking into consideration the target variable, also known as the y variable, one example may be making a prediction regarding the amount of active users who will be subscribing to a marketplace platform in the next month. It is usual practice to refer to these qualities as input features or x variables. This prediction is based on a number of different input characteristics, such as the quantity of items that have been sold or positive customer reviews. The final step involves making use of the sets of input and output data that are contained within the training set in order to fine-tune the adjustable parameters of the machine learning model. After the training procedure for the model has been successfully completed, it will be able to make accurate predictions for the target variable y when it is given data points for the input characteristics x that it has not previously encountered. A more nuanced distinction can be made with regard to the category of supervised learning between regression problems, in which the objective is to forecast a numerical value (for example, the number of users), and classification problems, in which the anticipated outcome is a categorical class affiliation, such as "lookers" or "buyers."

Unsupervised learning	The process of unsupervised learning takes place when the learning system is given the goal of recognizing patterns without the existence of any pre-existing labels or requirements within the system. Within the scope of this discussion, the training data is comprised entirely of variables x, with the objective of elucidating structural information of interest. One example of this is the process of clustering, which involves discovering groups of items that share similar qualities. Another example is the process of dimensionality reduction, which involves producing data representations that are projected from a high-dimensional space into a lower-dimensional space. The use of clustering strategies is a noteworthy example of unsupervised learning in the context of electronic marketplaces. This strategy is utilized to categorize clients or markets into segments, which enables communication that is more targeted and specialized, as well as suited to the characteristics of each segment being targeted.

Table 3.1: Overview of types of machine learning. Taken from [15]

3.1.3 ML Applications

In discerning which machine learning method(s) to employ, the initial step involves precisely formulating one's research question. Hernán and colleagues' framework introduces three primary data science research tasks: description, prediction, and causal inference.[25]. Machine learning is applicable to each of these three tasks; however, depending on the particular research question, conventional statistical

methods might be adequate and more suitable.

- Description - Tasks related to description entail utilizing data to furnish a quantitative summary of specific variables. In description tasks, unsupervised machine learning methods prove especially beneficial as they strive to uncover relationships within a data structure without relying on a measured outcome. The objective of unsupervised learning is to discern underlying dimensions, components, clusters, or trajectories within a given data structure.

- Prediction - The second prominent data science task is prediction, encompassing a variety of specific objectives. Conventional statistical methods, such as logistic regression, can be employed to test this hypothesis. They facilitate the acquisition of interpretable estimates regarding the nature and statistical significance of associations between predictors and the outcome, for instance, odds ratio. In scenarios where the objective is the optimization of prediction within large or complex data structures, machine learning might be favored over traditional statistical methods. This preference arises from the fact that machine learning methods have fewer and less restrictive statistical assumptions when compared to traditional parametric methods. A correlated objective in prediction is the identification of variables that significantly contribute to the accuracy of predictions.

  Supervised learning emerges as the linchpin for articulating prediction tasks, driven by the ambition to adeptly forecast or classify specific outcomes of interest, such as discerning the presence or absence of a mental disorder. This strategic approach finds application in the expansive realm of data structures, weaving together a rich tapestry that spans demographic, clinical, and social predictors.

- Causal Inference - Causal inference encompasses the nuanced process of estimating effects through a meticulous comparison of outcomes among individuals exposed to a particular factor and counterfactual outcomes that hypothesize what would have transpired had they not been exposed. This methodological approach is rooted in the discernment of causality, striving to unravel the true impact of an exposure by scrutinizing the divergence between observed outcomes and those counterfactual scenarios that serve as a baseline for comparison. Causal inference finds practical application in analyzing observational data, which lacks the randomization inherent in experimental designs, by adopting an approach that mirrors a hypothetical randomized trial. Although causal inference methods have been a longstanding presence in research for decades, the incorporation of supervised learning techniques represents a more recent and innovative development [16].

3.1.4 DL Applications

1. Natural language processing - DL is used in natural language for voice translation, machine translation, and computer semantic comprehension, among other fields. In [6], Shi Dong concluded that DL excels in image and natural language processing. Shi Dong cited 2015 Google Word Lens technology for real-time phone and video translation. This powerful technology may quickly recognize and translate words into the target language. The phone may also translate without an internet connection. Beyond visual translation, current technology can translate over 20 languages. Google added an automatic email reply capability to Gmail, using a DeepL model to evaluate email text.

2. Speech recognition - Researchers worked hard to realize Human-Computer

Interaction. By building the first experimental system in 1952, Davis and his Bell Institute team made history. This innovative approach identified 10 English digital pronunciations. Several decades have been spent researching speech recognition technologies, especially voice recognition. Voice recognition became a dominant force in some fields, named one of the top ten computer development events by the US press. Speech recognition technology has advanced significantly in the previous two decades. With the advancement of DL models, several voice recognition devices and applications have moved from lab to market. Baidu [4] introduced Deep Speech in 2014, using DL technology to achieve 8% accuracy in noisy conditions. By February 2016, Baidu's Deep Speech 2 has lowered phrase recognition error rate to 3.7%.

3. Medical applications - DL's predicting and automated feature recognition make it a popular illness diagnostic approach. DL uses in medicine, regardless of frequency or species, are constantly improving. Google began developing a groundbreaking technology to detect early-stage eye disorders in 2016. They sought early prevention of diabetic retinopathy and age-related macular degeneration with Moorfields Eye Hospital. A month later, Google developed a radiation approach for head and neck cancer using DL methods. This novel method controlled radiation time and reduced injury risk. DL technology is set to make significant contributions to precision medical treatment as it develops.

4. Computer vision - Computer vision, a pivotal application within AI, represents an interdisciplinary field dedicated to enabling computers to attain a sophisticated understanding from digital images or videos. This involves utilizing computers and cameras to substitute for the human eye in tasks

such as target object recognition, tracking, measurement, and resolution of various visual challenges. Furthermore, it extends its purview to graphic processing, empowering computers to surpass the capabilities of the human eye in image processing.

Below are listed the directions of computer vision.

- Image segmentation - Image segmentation doesn't need visual ideas or object recognition, unlike image categorization or object recognition. An object classifier only defines labeled items. In addition, an optimum picture segmentation algorithm segments new or unrecognized items. In human-computer interaction, each video frame is segmented to help users interact with people and objects. In an airport, the security crew focuses on unsecured objects that may contain dangerous compounds. This requires investigating all left-behind items.

- Face recognition - Face recognition uses human face traits for biometric identification. The camera records video or picture data of the face, which is then used for automated image and face detection. The newest DL algorithm, like FaceNet, achieves 99.63% face recognition accuracy, exceeding eye recognition. DL excels at capturing key features that manual expressions lack. Strong face attribute and identification selectivity and robustness for local blocks result from its moderate sparsity. DL features are used for face recognition, with display limitations or post-processing incorporated. Integration of DL is a key factor in its widespread use in face recognition. CNN technology is crucial for facial recognition in DL.

- Object detection - Object detection identifies categories of objects in photos or videos, such as human faces and eyes. Object detection is es-

sential to image understanding and computer vision, tackling complex problems including scene comprehension, picture captioning, event detection, and activity recognition. Its uses include consumer electronics, robot vision, security, autonomous driving, human-computer interaction, and automated surveillance. Due of image size, traditional algorithms have trouble learning patterns.

5. DL on graphs - Recently, academics have begun developing new methods for discovering patterns from graph-structured data. DL on graphs has been effective in tackling many challenges. In 2017, Ktena et al. [21] used graph CNN to predict graph similarity, detecting brain diseases. It is typical to treat complicated illnesses with numerous medications targeting intricate diseased proteins. However, changing one medicine may not be noticeable in clinical settings, especially with multiple drugs.

6. Intelligent transport system (ITS) - ITS underpin smart cities and provide crucial infrastructure for governments across ages. Vehicle management, a major part of transportation networks, is becoming harder and requires robots. US people took 181,541 public transportation vehicles for 9.9 billion trips and 55.8+ billion kilometers in 2019. This high utilization emphasizes smart transportation's importance for large cities worldwide. Transportation data comes in several forms, including text, audio, photos, and videos [6].

3.1.5 Data input

In addition to conventional numerical data, there is considerable volume of diverse data, particularly unstructured and non-cross-sectional data, including time series, images, and text. This data can be utilized for the construction of analytical models, aiming to enhance decision support or facilitate business automation

purposes. Nevertheless, manually extracting patterns and relationships would surpass the cognitive capacity of human operators, underscoring the indispensability of algorithmic support, especially when dealing with large and high-dimensional datasets.

The latest advancements in DL enable the processing of diverse types of data in combination, commonly known as cross-modal learning. This capability is valuable in applications where content is presented in multiple forms of representation, as seen in e-commerce websites where product information is typically conveyed through images, concise descriptions, and additional text metadata. Once these cross-modal representations are acquired, they can be employed, for instance, to enhance retrieval and recommendation tasks or to identify misinformation and fraud [15].

3.1.6 Feature extraction

The extraction of features that may be used in the construction of models is an essential step in the process of automatically recognizing patterns and correlations within large amounts of data resources. In a general sense, a feature is a description of a characteristic that is derived from the initial data input with the intention of delivering an acceptable representation. As a result, the purpose of feature extraction is to maintain information that is discriminative and to discern between variables of variation that are pertinent to the overall learning goal.

The achievement of success in shallow machine learning is closely connected to the utilization of well specified features, and the performance of this technique is contingent upon the efficiency of the extraction process. The process of manually crafting features is a difficult task that often requires a significant amount of domain expertise inside a specialized engineering process that is suited to a par-

ticular application. It is considered to be laborious, time-consuming, and lacking in flexibility due to the fact that it requires a significant amount of effort.

Deep neural network (DNN) address this constraint associated with manually crafted feature engineering. Their sophisticated structure empowers them with the ability to autonomously learn features, extracting discriminative representations with minimal human intervention. Hence, DL is more adept at handling extensive, noisy, and unstructured datasets. The process of acquiring features typically advances in a hierarchical fashion, where more complex, high-level abstract features are constructed from simpler ones. However, based on the nature of the data and the selection of DL architecture, various methods of feature learning are employed in tandem with the model-building phase [15].

3.1.7 Model building

In the process of automated model construction, a learning algorithm utilizes the input to discern patterns and relationships that are pertinent to the specific learning task. In contrast, DL has the ability to directly work with high-dimensional raw input data for the purpose of model construction, leveraging its automated feature learning capability. Hence, architectures in DL are frequently structured as end-to-end systems that integrate both aspects into a single pipeline. Differences in architectural variants primarily stem from the kinds of layers, neural units, and connections they employ. Table 3.2 provides an overview of the three categories of CNN Recurrent neural network (RNN) and DNN [15].

Architecture	**Description**
Convolutional neural network	CNN are predominantly employed in tasks associated with computer vision and speech recognition. They can tackle tasks that involve datasets with spatial relationships, where the interchangeability of columns and rows is not applicable. Their network structure consists of multiple stages, facilitating hierarchical feature learning in accordance with the specific modeling task at hand. As an illustration, in the context of object recognition in images, the initial layers of the network are tasked with extracting fundamental features, such as edges and corners. These basic features are progressively combined in the final layers to form more intricate representations resembling the actual objects of interest, such as animals, houses, or cars.
Recurrent neural network	RNN are explicitly crafted for data structures that follow a sequence, such as time-series data, event sequences, and natural language. Their structure incorporates internal feedback loops, allowing for the learning of sequential patterns and the modeling of time dependencies by establishing a memory. Basic RNN architectures pose challenges as they encounter the issue of vanishing gradients, leading to minimal or no impact from early memories. More advanced architectures, like long short-term memory (LSTM) networks equipped with sophisticated attention mechanisms, address this issue.
Deep neural network	DNN are distinguished by the inclusion of a large number of hidden layers that are grouped in complex network designs. Their ability to handle raw input data and independently discover the representations that are necessary for the learning task is significantly improved as a result of their frequent utilization of complex processes, like as convolutions, or the incorporation of numerous activations inside a single neuron. DL is the term that is usually used to describe this kind of capacity.

Table 3.2: Overview of DL architectures. Taken from [15]

3.1.8 Model assessment

Evaluating the quality of a model requires considering various aspects, including its performance, computational requirements, and interpretability. Metrics based on performance assess how effectively a model fulfills the objective set by the learning task. In the realm of supervised learning, there exist established guidelines for this particular endeavor. In this context, it is a standard practice to employ k-fold cross-validation to mitigate the risk of overfitting and assess the model's performance on out-of-sample data that was not part of the training set.

- Regression - The evaluation of regression models involves assessing estimation errors, such as the root mean square error (RMSE) or the mean absolute percentage error (MAPE).
- Classification - The evaluation of classification models involves computing various ratios of correctly and incorrectly predicted instances, including accuracy, recall, precision, and the F1 score.

In choosing an appropriate prediction model for a given task, it is rational to compare different models with varying complexities. This involves considering competing model classes as well as alternative variants within the same model class. As previously mentioned, the complexity of a model can be described by several attributes, including the type of learning mechanisms (e.g., shallow machine learning vs. DL), the quantity and type of manually created or autonomously extracted features, and the count of trainable parameters (e.g., network weights in artificial neural networks). Less complex models typically lack the flexibility to capture pertinent (non-linear) regularities and patterns associated with the learning task. On the flip side, excessively intricate models come with an increased likelihood of overfitting. Moreover, comprehending their reasoning is more challenging (as discussed in the next section), and they are likely to entail higher computational

costs. The costs associated with computation are articulated through memory demands and the time taken for inference when executing a model on fresh data. These considerations are particularly crucial when assessing DNN, given that processing and storing millions model parameters impose distinctive requirements on hardware resources. Hence, in business environments with limited resources, especially those heavily reliant on mobile devices, it is imperative to select a model that achieves a harmonious balance between underfitting and overfitting. They should also assess a model's complexity in terms of additional trade-off relationships, such as the balance between accuracy, memory usage, and speed.
([15])

3.1.9 Balance in the model

When constructing intelligent systems with shallow ML and DL models, the possibilities for algorithms or architectures, hyperparameters, and training data are virtually limitless([12]). Simultaneously, there is a dearth of established guidelines regarding the construction of a model tailored for a specific problem. This lack extends to ensuring not only performance and cost-efficiency but also robustness and privacy. Furthermore, as described earlier, business environments with constrained resources frequently involve numerous trade-off relationships to contemplate, including the balance between prediction quality and computational costs. Hence, the process of constructing analytical models stands out as the most pivotal task, as it ultimately shapes the business success of an intelligent system. Comparisons between different implementations can be accurate only when altering one element of the triangle at a time and reporting identical metrics.
([15])

3.2 Differential Privacy

Differential privacy (DP) conceals the presence or absence of any individual or small group of individuals within a dataset. This is accomplished by ensuring that, for each individual, any conclusion reached from the analysis would have been essentially as likely to have been reached regardless of whether the given individual joined the dataset or refrained from joining the dataset. In the following paragraphs will explain what DP is, when it is appropriate to use it, how to implement it, and how to achieve it using frameworks.

3.2.1 Definition and characteristics of DP

DP is a precise mathematical notion that provides a clear definition for protecting privacy. The story explores the interplay between a data analyst and a curator who has a dataset. The stipulation is a specific constraint on the probability of interactions across datasets that differ in the data of a single individual. This limitation hinders the exploration of new information on an individual's data, but it permits the gathering of statistical insights about all individuals in the dataset. Statistical learning reveals insights into cohorts of individuals, perhaps exposing information that members of the group may like to keep confidential. To do this, it is necessary to include a meticulously regulated level of unpredictability into the calculation. Therefore, the result of a differentially private analysis depends not only on the data itself but also on the added randomness. Essentially, after the dataset is completed, every possible outcome is linked to a probability that represents the chance of that outcome being seen. From the standpoint of individuals whose future is influenced by the algorithm's results, it may be stated that although it may be difficult to determine the likelihood of a subjectively negative outcome, the choice to participate or abstain from the data set will not substan-

tially alter the risk. The term "significantly" is determined by the parameter ε, with a smaller value of ε indicating less variation and so greater privacy. The change, whether it is an increase or a decrease, is multiplied by a factor of at most e^{ε}. When the value of ε is significantly smaller than 1, the result is about $1 + \varepsilon$. When ε equals 1, the value is around 2.71. However, when ε is more than 1, the value increases rapidly. For instance, when ε is equal to 3, the value is about 20, and when ε is equal to 5, it is approximately 148.4. When the value of ε is equal to 15, it exceeds 3,269,017. When the value of the bound e is close to one (i.e., e is close to 0), any information obtained about a participant is about as likely to be obtained about a non-participant [7].

The following four characteristics of DP create an atmosphere that is particularly advantageous for data-driven and data-rich environments.

1. An objective measurement of the loss of privacy that occurs as a result of DP solutions may be made using. This allows for a comparative analysis of the risks to privacy posed by different systems. Although DP may not always provide a clear distinction between high-quality and low-quality implementations, it does provide a chance to transform at least this aspect of an organization's privacy policies into a quality that can be evaluated by those who are not affiliated with the organization. Comparative analysis between different institutions is made possible by this quantification.

2. Because of the requirement to choose ε, there is a chance to contemplate one's core beliefs. With the option to tweak ε, organizations are compelled to choose a privacy-utility mix and document it. This provides a database controller with the opportunity to modify her choice for privacy and utility. The controller might be the entity that owns or physically manages the database, or it could be a regulator that possesses logical control through a

regulatory structure. Both of these possibilities are possible. In the process of algorithmic decision making, ε generates a symbolic representation of social values as a placeholder.

3. According to the mathematical definition of DP, being differentially private is independent of what a privacy opponent may or might not know, as well as to which additional sources of information such an adversary might or might not have access, at any given moment, even in the future. This is the case regardless of whether the adversary is aware of the information or not. DP, in other words, is not dependent on any one adversary and is future-proof.

4. The mathematics of DP makes it possible to monitor and manage the cumulative loss of privacy that occurs across many data uses [7].

3.2.2 The Opposition of DP in Action

In addition to gaining a grasp of what DP does give, it is essential to have an understanding of what it does not supply, even with a very small ε. DP is not a single-solution solution. When it comes to studying outliers, DP is not the appropriate instrument to utilize since it conceals the presence or absence of outliers. For the purpose of studying tiny datasets, this tool is not appropriate. It is possible for DP to conceal significant disparities in tiny populations or subpopulations of interest, depending on the choice of epsilon. However, despite the fact that this may be seen as a restriction, it is actually a feature. It is important to keep in mind that DP assures that any result that is made from the analysis would have been essentially as likely to have been reached with or without the data of any individual than it would have been otherwise. When compared to the situation with small datasets, we anticipate that statistical estimators that are run on large datasets will be more resistant to the addition or deletion of a data point. This

is because large datasets include more information. When compared to situations in which the dataset is big, the likelihood of a substantial change in the value of the statistical estimator being brought about by the addition or removal of single individual from a small dataset is considerably higher. Despite the fact that it is less accurate, DP is, in fact, functioning as it was designed to do, which is to conceal the presence or absence of an individual over the results of the study. Legal limits on data access, usage, and sharing can occasionally promote the correct mix between discovery and privacy in situations where tiny datasets need to be evaluated. This is the case if the intended application supports the confidentiality risks that are involved. Consider the possibility of a federal database that contains personally identifiable health information that has been gathered without the individual's agreement for the sake of public health, such as detecting persons who are infectious. Unlike informational privacy, this non-consensual collecting is not addressed by DP, which is a violation of informational privacy. The additional information that is obtained from the analysis of the data that protects individuals' privacy may be used to modify the information that is provided to persons who are making significant decisions regarding their health care (for example, a decision regarding reproduction). This is another possible cause for worry. DP does not in any way restrict the manner in which the information obtained about the population as a whole is utilized to impact the treatment of particular people, regardless of whether or not the data collected from those individuals were included in the database [7].

3.2.3 Exactly What Does Epsilon Mean?

3.2.3.1 An Uncertainty Regarding the Optimality

There is a lack of clarity on the appropriate level of privacy loss in a particular setting, which works to weaken the pressure that is being put on for adequate DP implementations. To begin, it is feasible that a procedure that may extract the highest potential utility for any given value of may simply not be known. This is true for a particular analytical activity in general as well as for a very specific type of data. Through the use of technical research, this issue can be resolved over time. The second issue is that we do not have a formula that can determine, given a particular privacy-utility tradeoff, what the most prudent choice of epsilon would be. As was seen in the flossing example, not all big epsilon are the same. This is especially relevant in light of the fact that. If we do not know how small an epsilon is possible, which means how much privacy can be offered, consistent with a given analytical utility (the optimal privacy-utility mix), or if we do not know a fair privacy price to pay for what we learn, then the measurement has less meaning. This is because we are able to cap the amount of privacy loss that a particular algorithm can cause [7].

3.2.3.2 Understanding the Meanings of Epsilons of a Smaller Size

Although all tiny epsilons are similar, it is important to keep in mind that the value of e^{ε} is approximately equal to the sum of ε and one, provided that ε is less than one. This is based on an analogy to the first phrase of Anna Karenina, which states, "All happy families are alike, but an unhappy family is unhappy after its own fashion." This makes it difficult to reason about the huge ε, which states that each large ε is enormous after its own fashion. Imagine that there are two different kinds of entities that live in this world: ghosts and humans.

This means that both kinds of creatures behave in the same manner, interact with other people in the same manner, blog, study, work, laugh, love, mourn, reproduce, become ill, recover, and age in the same manner. There is just one distinction between ghosts and people, and that is that ghosts do not have any recordings in statistical databases. The enemy of privacy seeks to ascertain if a certain individual, who is fifty years old and is referred to as the "target," is a ghost or a human being. The enemy is, in fact, allowed the whole fifty years to do this. It is not necessary for the adversary to remain passive. For instance, she can organize clinical trials and enroll patients of her choosing. She can also create humans to populate databases, effectively creating the worst-case scenario (for privacy) databases that were discussed earlier. Furthermore, she can expose the target to chemicals at the age of 25 and again at the age of 35. Her knowledge of the target encompasses every single piece of information that could ever be recorded into any database. If the target were human, she would be able to determine which databases the target would be in. This would mean that the only potential violation of privacy that would be left to worry about would be the human/ghost bit of the target. In this particular scenario, every assurance of a constraint of ε on privacy loss is similar; the particular method and the kind of data are immaterial [7].

When ε is small, the human/ghost bit remains hidden, regardless of how much or how little additional information the adversary has about the target and about the rest of the world. This is true whether we are discussing a single computation or the cumulative privacy loss associated with multiple computations performed on the same database or across multiple databases. In this particular scenario, a tiny ε guarantees that the database does not disclose any information about the target (beyond what would be disclosed about the target based on the outcome of the calculation in the event that the target had chosen to opt out of the dataset).

We have seen that this unification is unsuccessful when the value of ε is high, as demonstrated by the flossing example. From a mathematical standpoint, this is due to the fact that dealing with a big ε simply does not succeed in concealing the human/ghost bit. As a result, we are unable to draw any conclusions from the assumption that this bit is concealed, just as we did in the case of a small ε. When ε is small, decent reasoning regarding the semantics of DP, including making sure that we are legitimately comparing apples to apples, is sound. However, when ε is big, this reasoning is no longer sound [7].

3.2.4 DP Mechanisms

This is the time when we will talk about several well-known techniques of DP.

- Randomized Response Mechanism - The randomized response is an example of a non-interactive technique, in which the data of each user is affected in an individual manner based on the outcome of coin flips. This technique affords the respondent the opportunity to maintain "plausible deniability" [29].

- Laplace Mechanism - In the field of DP, the Laplace mechanism is among the mechanisms that are utilized the most frequently. According to this approach, the random noise that conforms to the Laplace distribution with a mean of zero and a sensitivity of $GS(f)/\varepsilon$ is incorporated into the answer of every query in order to ensure that it is disturbed in an acceptable manner [3]. When calculating noise with l1 sensitivity, the Laplace technique is often utilized in ε DP. This is mostly due to the fact that ε is the sole parameter that is of relevance. In addition, the bounded Laplace mechanism is utilized by GRAM-DP in order to incorporate noise into the query answer.

- Gaussian Mechanism - Another technique that has garnered a lot of interest

in recent times is the Gaussian mechanism, which is designed to create (ε, δ)-DP. The answer of the query is now being supplemented with a certain quantity of zero-mean Gaussian noise. The quantity of noise that was added to the result corresponds to the sensitivity of the l2 scale. Along with ε, another component that plays a significant role in this context is δ, which is responsible for determining the likelihood of one's privacy being compromised. The demonstration of an extended Gaussian method for DP was carried out by Liu et al. [22].

- Exponential Mechanism - There are some queries that do not supply their output with numerical numbers. Therefore, McSherry and Talwar [24] developed a mechanism that can be utilized to make non-numerical searches differentially private. This method may be employed. Since the Exponential mechanism assigns exponentially larger odds of being picked for the higher outcomes, the end result would be near to optimal answers after applying it for nonnumeric inquiries. This is because the probability of being selected for the higher outcomes is exponentially higher.

3.2.5 DP Frameworks

Next table 3.3 describes different DP frameworks.

Framework	**Description**
Opacus	In addition to being developed on top of PyTorch, the DP library for ML services that Facebook uses is called PyTorch Opacus [26]. OpenMined is an open-source community that is committed to researching privacy strategies for machine learning and AI. It is being developed in conjunction with Facebook AI Research, the PyTorch team, and OpenMined. Because of its broad and specialized qualities, the service offered by Opacus is geared for both professional DP researchers and practitioners of machine learning practice.
Google DP	An open-source version of Google's DP library, which is used to power several of the company's major products, was recently published by Google [9]. This library, which is available in Java and Go, is a collection of years' worth of developer experience from Google. It provides practitioners and organizations with the opportunity to reap possible advantages from its implementation, and it requires a relatively minimal degree of skill in DP to get started.

OpenDP	OpenDP Smartnoise [11] can trace its origins back to the Privacy Tools Project initiated by Harvard University. Through the course of this project, they were able to acquire expertise in the construction and deployment of PSI [14], which is a system that was built to exchange and analyze privacy-sensitive datasets while maintaining the privacy safeguards of DP. Ultimately, this project helped to their efforts toward Smartnoise. Other distributed processing (DP) technologies, such as PinQ [23], ektelo [30], PrivateSQL [20], Fuzz [10], and LightDP [31], have also been merged into OpenDP Smartnoise. Even though the majority of the tools, such as PSI and PinQ, are research prototypes, OpenDP Smartnoise is now putting up efforts to further develop DP principles into solutions that are ready for production utilization. As a result of these characteristics, OpenDP Smartnoise has garnered a great amount of popularity within the developer community, as evidenced by the fact that its open-source software repository has more than two hundred stars.

Tensorflow Privacy	The work of Abadi et al. [1], who constructed a comparable optimizer for TensorFlow and a privacy cost tracker, served as the initial inspiration for the development and distribution of TensorFlow Privacy [27] (TFP), which is a machine learning framework developed and distributed by Google. The emergence and adaptation of DP mechanisms to TensorFlow is what makes it possible for users to take advantage of DP when training machine learning models. Additionally, TFP is adjustable, and developers have the ability to create their own machine learning models. With these models, developers are able to develop their own operators and integrate them in their applications. Because of its adaptability and DP services, TFP has evolved into an open DP tool that is utilized by a sizable developer community and to which they provide significant contributions.

Diffprivlib	The industrial giant IBM is responsible for the development of Diffprivlib [13], which enables DL in machine learning tasks such as classification, regression, clustering, dimensionality reduction, and data regularization. A tool that may be used for a variety of purposes, including performing experiments, investigations, and application development while maintaining DL is what it is intended to be. During their interactions with Diffprivlib, practitioners of varying levels will have no trouble locating the information they want thanks to the comprehensive product handbook that can be found at https://diffprivlib.readthedocs.io/en/latest/. This manual was accessed on December 31, 2023. The open repository that Diffprivlib maintains [5] has garnered a significant amount of attention from developers as a consequence of this.

Chorus	Chorus [18, 17] uses a cooperative architecture to conduct DP statistical queries. It uses industrial-grade database management systems (DBMS) for data processing and queries that need to be altered or repeated. This design focuses on rewriting, analysis, and postprocessing. The post-processing component processes query output, whereas the rewriting component changes queries for clipping. The analysis component analyzes queries to find features such DL noise. Chorus' clipping summing procedure illustrates this. The rewriting component can modify the initial query such that the database management system (DBMS) clips and sums, while the analysis and post-processing component completes the summation. Chorus differs from previous efforts because of its DBMS independence. Chorus does not require database change or a server-based database engine, unlike an integrated solution. When working with large datasets, Chorus can employ DMBS to ensure scalability. Chorus deployment may require additional measures to prevent hostile actors from getting sensitive data. Chorus' Uber implementation restricted privacy-sensitive data to a central query interface. The UI, privacy budget account, and DMBS were protected against tampering.

Table 3.3: Description of DP Frameworks [32]

Chapter 4

Methodology

In this section, the study design will be detailed, along with the methods that were utilized to accomplish the goals that were established.

4.1 Research design

In light of the fact that this study requires obvious patterns, the experimental technique has been selected as the appropriate approach to take. This is because clear patterns can be obtained through appropriate experiments with the tuning of the hyperparameters.

4.2 Procedure used

The next procedure has been selected in order to carry out experiments of this nature that will fulfill the objectives that were established.

First things first, dataset with sensitive data will be selected so the experiment is close enough to reality, since DP is suitable to use on those data that could reveal

some sensitive data.

Secondly, DNN will be selected as the type of the model because its hidden layers can be robustly selected and configured to achieve preferable learning speed and quality.

Thirdly, different configurations of models will be considered to tryout each possible way to achieve Differential privacy and so we can examine the difference between them.

Lastly, grid search approach will be used to find the best possible combination of the parameters for each of the models solely and then to choose the best model configuration out of all.

Chapter 5

Experiments

5.1 Context of the experiment

Experiment has been conducted in order to derive patterns of successful hyper-parameters tuning for ML models on a sensitive dataset with preserving high accuracy of the model.

5.2 Tensorflow-privacy usage

Tensorflow-privacy has been selected due to its robust model setup possibilities. There are three DP-model setups plus two baseline models in following experiments. Two of them (DP-models) use different optimizers from tensorflow-privacy framework and Sequentional model from Keras, while third one uses DPSequentional model from tensorflow-privacy and common optimizers from tensorflow. Such choice of the models allows us to make observations and statements on the difference and avant-gardism of some kind of those models after proper evaluation.

5.2.1 Differentially private stochastic gradient descent

This method, known as Differentially private stochastic gradient descent (DP-SGD), is based on modifying the gradients used in Stochastic gradient descent (SGD), the main algorithmic building block of nearly all deep learning systems. For their input data, models trained with DP-SGD offer provable differential privacy guarantees. The original SGD algorithm has two changes made to it:

1. Each gradient's sensitivity must first be limited. Put differently, you must restrict the amount that each unique training point sampled in a minibatch can affect the gradient calculations and the updates that are applied to the model parameters as a result. Each gradient that is computed on each training point can be clipped to achieve this.

2. To make it statistically impossible to determine whether a specific data point was included in the training dataset by comparing the updates SGD applies when it operates with or without this specific data point in the training dataset, random noise is sampled and added to the clipped gradients [28].

5.2.2 Hyperparameters in DP-SGD

Three privacy-specific hyperparameters for DP-SGD are available, and one hyperparameter is already in place and will be adjusted in the upcoming experiments:

- l2_norm_clip (float) - The greatest Euclidean (L2) norm of every gradient used to update the parameters of the model. The sensitivity of the optimizer to specific training points is constrained by this hyperparameter.

- noise_multiplier (float) - The volume of noise that is introduced to gradients and sampled during training. Generally speaking, increased noise levels

improve privacy (sometimes, but not always, at the price of decreased usefulness).

- microbatches (int) - Every data batch is divided into smaller segments known as microbatches. Every microbatch ought to include a single training example by default. As a result, gradients may now be clipped per example instead of being averaged over the minibatch. As a result, utility is usually maximized and the (negative) impact of clipping on the signal detected in the gradient is reduced. However, by enlarging microbatches to contain more training instances, computational costs can be minimized. After that, the average gradient from all of these training instances is trimmed. The total number of samples used in a batch, or one gradient descent step, doesn't change. The batch size should be divided equally by the number of microbatches.

- learning_rate (float) - In vanilla SGD, this hyperparameter is already there. Every update counts more the greater the learning rate. In the event that the updates are noisy—that is, the additive noise is greater than the clipping threshold—a low learning rate might facilitate the training process' convergence [28].

5.3 Dataset

Heart disease dataset from 2022 (https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease) has been selected, such choice is sufficient in light of its sensitive data. Even though it has been anonymized, it doesn't change the fact that personal confidential information could be compromised by the attacker, thats what makes it suitable for further experiments.

5.3.1 Feature selection

Decide whether patient has heart disease we can on two columns: "HadHeartAttack" and "HadAngina", those two columns were combined in one boolean "HeartDisease" and it will take 1 if at least one of those columns is true.

5.3.2 Data distribution

There are 30 categorical columns and 7 numerical ones, other two were used as features. In further refinements visualization of the correlations, outliers, etc. will be added.

5.3.3 Train - test split

To achieve optimal training speed and sample size it has been deduced to choose batch size of 30, so after calculations it happen to be split 73,17074%(train) - 26.82926%(test)

5.4 Models setup

In this section will be described which models choices have been made and why, containing rationalization of baseline and experiment's main models.

5.4.1 Hidden layers

For current dataset next hidden layers has been chosen for each model with hidden layers, except for linear regression:

1. Input layer;
2. Dropout layer - rate of 0.2;

3. Dense layer - 5 neurons, ReLu activation function;

4. Dropout layer - rate of 0.2;

5. Dense layer - 1 neuron, Sigmoid activation function.

Due to long training time of linear regression model, Deep neural network models were used, but with this dataset they were too powerful even with such relatively narrow layers, so it has been decided to use dropout layers[1], so training has more optimal time.

5.4.2 Baseline models

For baseline models purposes Binary classification DNN and linear regression has been chosen. Binary classification baseline was set with the hidden layers discussed in the previous subsection, while linear regression model was set with single Dense layer with one neuron and linear activation function.

Their accuracy had quite constant progression through all possible sets of batch sizes.

[1]The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting [19].

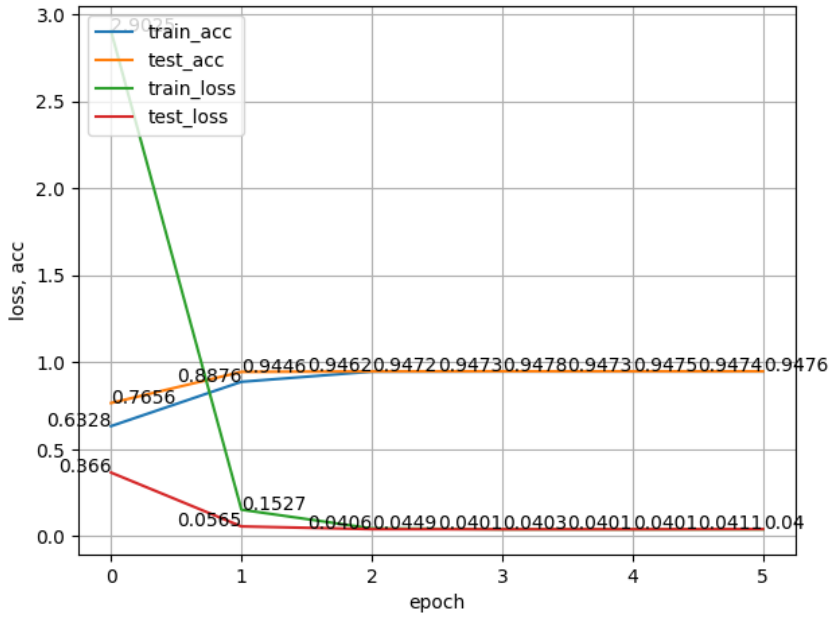

Figure 5.1: Plot of the baseline Linear Regression model training without Differential privacy

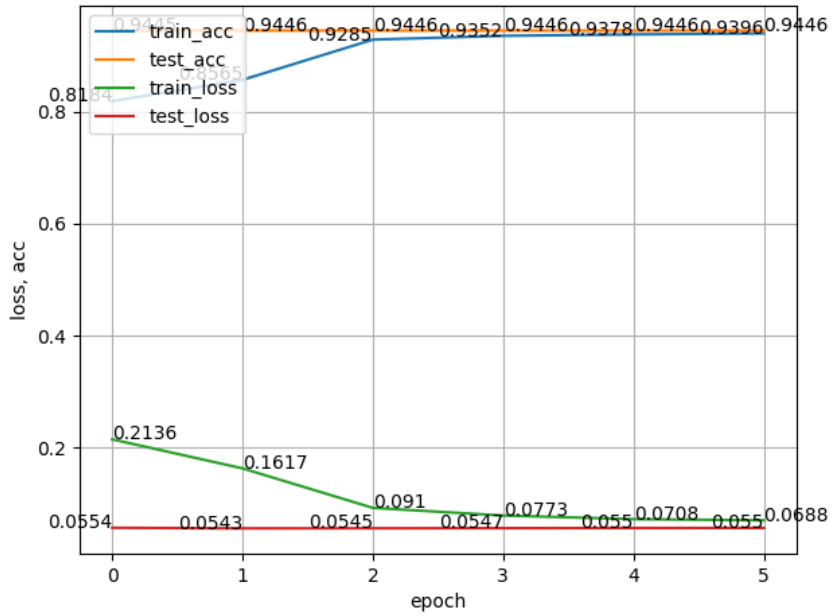

Figure 5.2: Plot of the baseline Binary Classification model training without Differential privacy

5.4.3 Grid search approach

Grid search is such a technique that builds a model with multiple possible sets of hyperparameters and produces output based on which we can decide what hyperparameters set is better to use for our purposes.

It was implemented using set of coefficients and a static value of each of parameters. For instance, last set of coefficients were

$$.25, .5, 1, 1.25, 1.5, 2$$

, they where consequently multiplied by static value of noise-multiplier = 1 and l2-norm-clip = 0.6 for each of chosen microbatch sizes (1, 15, 30 for batch size of 30). This way we can observe edge values that hyperparameters will produce to get a range of results where, ideally, we could see peak observations at some points and decide if there is some patterns to write down.

5.4.4 Experiment models

Let's take a closer look on the models that were used in experiments with hyperparameters tuning.

There were used two tensorflow-privacy optimizers:

1. DPSparseKerasSGDOptimizer
2. DPKerasSGDOptimizer

They were used in Sequential DNN model from Keras and provide privacy by applying noise to the gradients while training (precise mechanism is described in [2]), so all of the dp hyperparameters were set into those optimizers.

Another model, that was crafted, was DPSequential model from tensorflow-privacy

with regular Keras optimizers, which implements standard Gaussian mechanism introduced by C. Dwork [8]

5.5 Preliminary results

Collecting multiple tests results to yield average values graphs were created which show us dependency of accuracy, loss and time on noise_multiplier, l2_norm_clip and number of microbatches.

5.5.1 Analysis

Let's take a look at experiments results with Sequential model with DPKerasSGDOptimizer and 25 batch size, 25 microbatches and ranges of noise_multiplier - (0.25, 0.5, 1, 1.25, 1.5, 2.0) and l2_norm_clip - (0.3, 0.6, 0.9, 0.15, 0.75, 1.2)

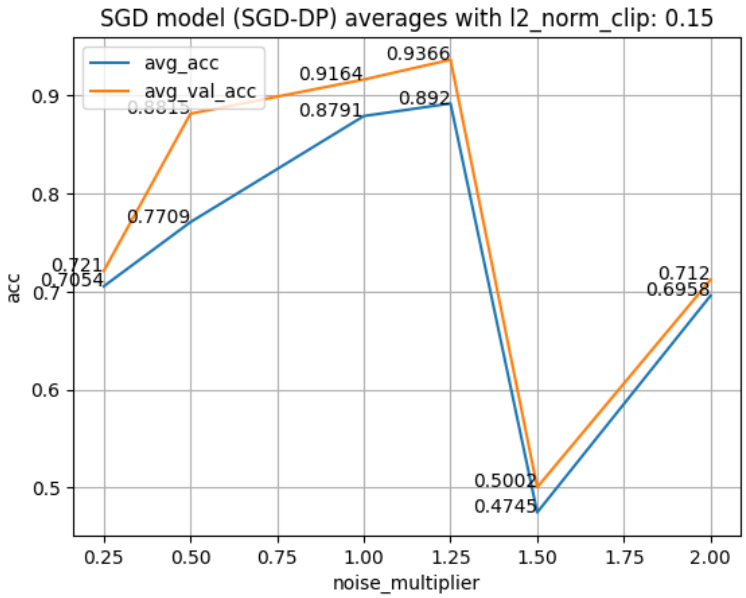

Figure 5.3: Plot of the SGD model with SGD-DP optimizer accuracy with batch size 25 and l2_norm_clip 0.15

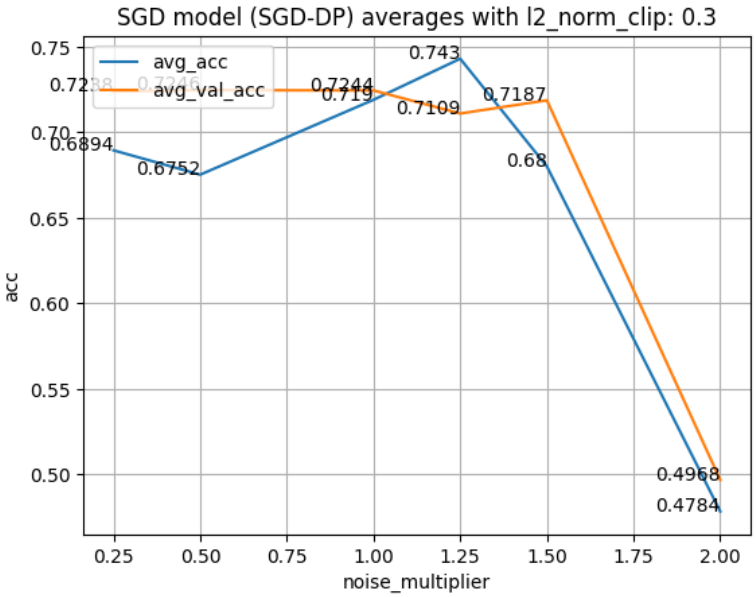

Figure 5.4: Plot of the SGD model with SGD-DP optimizer accuracy with batch size 25 and l2_norm_clip 0.3

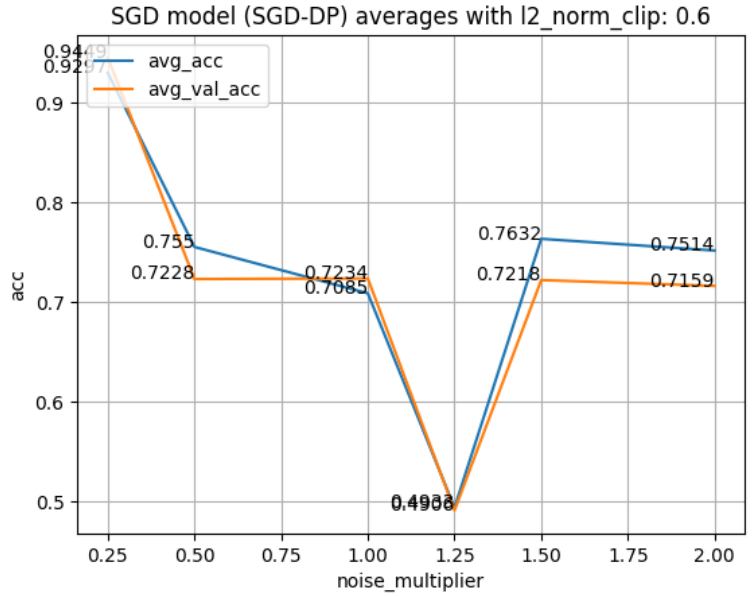

Figure 5.5: Plot of the SGD model with SGD-DP optimizer accuracy with batch size 25 and l2_norm_clip 0.6

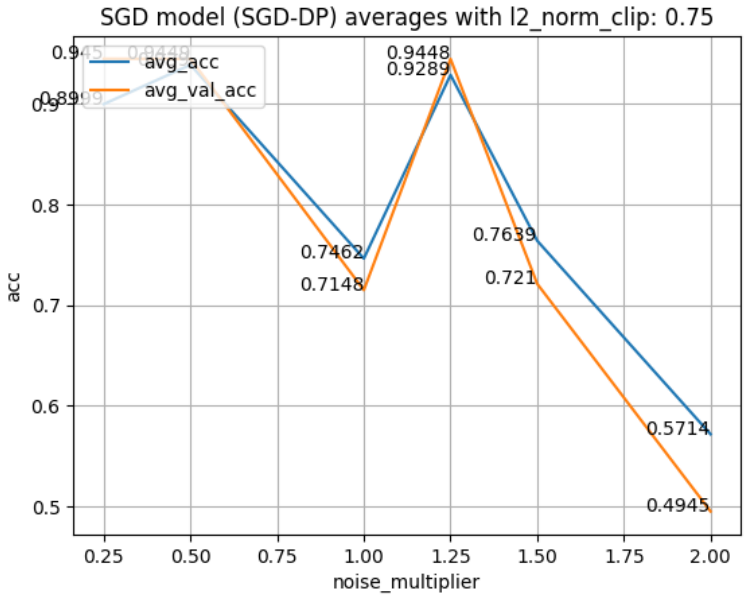

Figure 5.6: Plot of the SGD model with SGD-DP optimizer accuracy with batch size 25 and l2_norm_clip 0.75

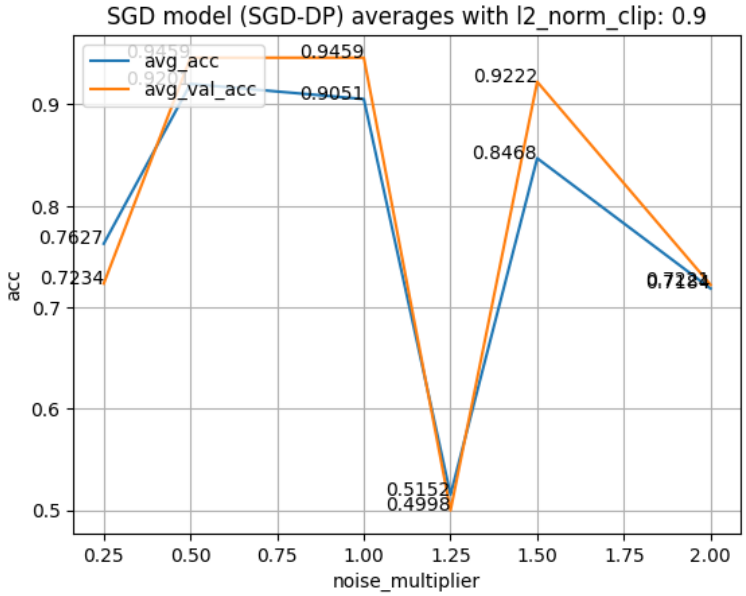

Figure 5.7: Plot of the SGD model with SGD-DP optimizer accuracy with batch size 25 and l2_norm_clip 0.9

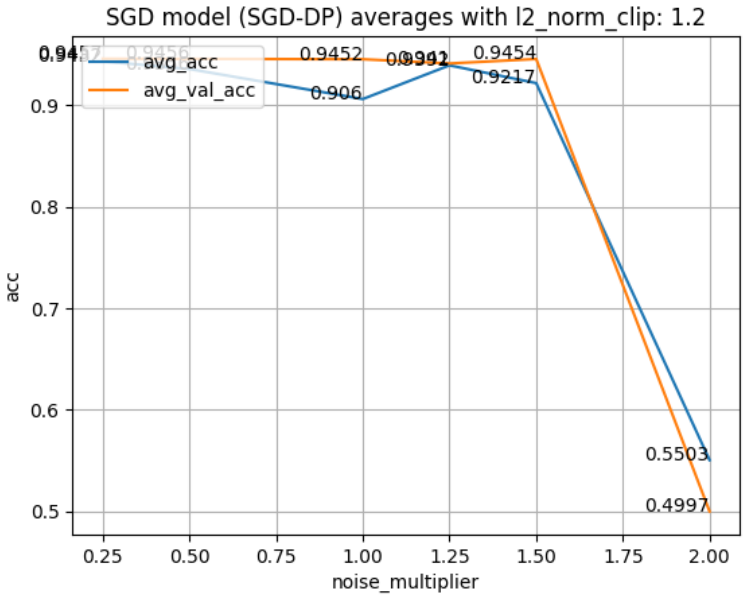

Figure 5.8: Plot of the SGD model with SGD-DP optimizer accuracy with batch size 25 and l2_norm_clip 1.2

From such graph we can deduce that there are some kind of right proportions to stick to to get stable and high accuracy while setting L2 and noise. Some of the pairs are so-so, they are keeping themselves at 0.7 level. Some of the pairs are a disaster, they sit at 0.5 level of accuracy not being able to get higher due to unlucky setting of the noise and clipping, where clipping clips not noisy data but real and leaves to us not realistic prediction.

Such procedure can be described as metallic smelting, where right proportions get us a extremely durable metal but a slight deviation from such formulas could ruin everything.

For instance, graph 5.4 illustrates us that highest possible accuracy was around 0.74, that means that over 4 experiments with the same parameters averages are still sitting low, while baseline is over 0.9. It is possible to state now that such clipping is not satisfactory for noise multiplier up to 1.5 and catastrophic for higher

noise.

Jumping over all other graphs, that show us a little bit of improving through increasing L2 clipping, let's focus attention at graph 5.8, with such clipping and all noise multipliers from 0.25 to 1.5 accuracy was high, although it dropped at 2.0, the maximum ε value that we can achieve with such parameters is 0.2590, which is considered a high privacy guarantee. Any ε over 10 is considered non-private, ε between 1 and 5 are considered merely private and ε lower than 1 is considered to be highly private. The general rule of thumb at work here is the closer to zero, the stronger the privacy

On the next graph 5.9 and in the next table 5.1 we see what epsilon values can be achieved with what noise multipliers

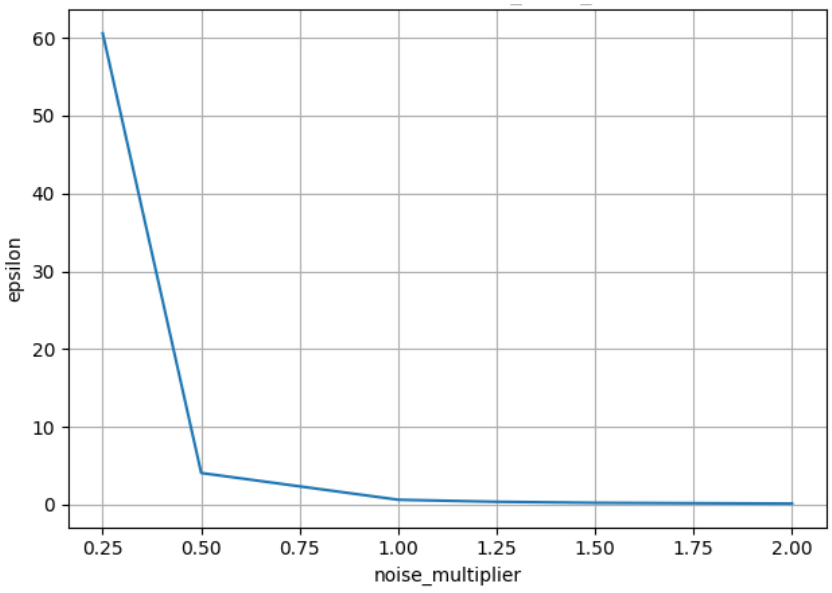

Figure 5.9: Plot of the epsilon values with L2 clip of 1.2 and batch-size of 25

Noise multiplier	Epsilon value
0,25	60,57
0,5	4,07
1	0,63
1,25	0,37
1,5	0,25
2	0,14

Table 5.1: Table Epsilon values with L2 clip of 1.2 and batch-size of 25

Chapter 6

Related Work

Chapter 7

Conclusion

7.1 Summary

7.2 Future Work

Resumé

Literature

[1] Martin Abadi et al. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318.

[2] Martin Abadi et al. "Deep learning with differential privacy". In: *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 2016, pp. 308–318.

[3] Rola I Al-Khalid et al. "A secure visual cryptography scheme using private key with invariant share sizes". In: *Journal of Software Engineering and Applications* 10.01 (2017), p. 1.

[4] *Baidu Research, howpublished=http://research.baidu.com/Index*. (accessed on 1 January 2024.

[5] *Diffprivlib, howpublished=https://github.com/IBM/differential-privacy-library*. (accessed on 31 December 2023.

[6] Shi Dong, Ping Wang, and Khushnood Abbas. "A survey on deep learning and its applications". In: *Computer Science Review* 40 (2021), p. 100379. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2021.100379.

[7] Cynthia Dwork, Nitin Kohli, and Deirdre Mulligan. "Differential Privacy in Practice: Expose your Epsilons!" In: *Journal of Privacy and Confidentiality*

9.2 (). Ed. by null. DOI: 10.29012/jpc.689. URL: https://par.nsf.gov/biblio/10217360.

[8] Cynthia Dwork et al. "Our data, ourselves: Privacy via distributed noise generation". In: *Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28-June 1, 2006. Proceedings 25*. Springer. 2006, pp. 486–503.

[9] Miguel Guevara. *Enabling developers and organizations to use differential privacy*. 2019.

[10] Andreas Haeberlen, Benjamin C Pierce, and Arjun Narayan. "Differential privacy under fire". In: *20th USENIX Security Symposium (USENIX Security 11)*. 2011.

[11] *Harvard University Privacy Tools Project, howpublished = https://privacytools.seas.harvard.edu/*. (accessed on 31 December 2023.

[12] Kai Heinrich et al. "Process data properties matter: Introducing gated convolutional neural networks (GCNN) and key-value-predict attention networks (KVP) for next event prediction with deep learning". In: *Decision Support Systems* 143 (2021), p. 113494. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2021.113494. URL: https://www.sciencedirect.com/science/article/pii/S016792362100004X.

[13] Naoise Holohan et al. "Diffprivlib: the IBM differential privacy library". In: *arXiv preprint arXiv:1907.02444* (2019).

[14] Justin Hsu et al. "Differential privacy: An economic method for choosing epsilon". In: *2014 IEEE 27th Computer Security Foundations Symposium*. IEEE. 2014, pp. 398–410.

[15] Christian Janiesch, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning". In: *Electronic Markets* 31.3 (2021), pp. 685–695.

[16] Tammy Jiang, Jaimie L. Gradus, and Anthony J. Rosellini. "Supervised Machine Learning: A Brief Primer". In: *Behavior Therapy* 51.5 (2020), pp. 675–687. ISSN: 0005-7894. DOI: https://doi.org/10.1016/j.beth.2020.05.002.

[17] Noah Johnson et al. "Chorus: a programming framework for building scalable differential privacy mechanisms". In: *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE. 2020, pp. 535–551.

[18] Noah Johnson et al. "Chorus: Differential privacy via query rewriting". In: *arXiv preprint arXiv:1809.07750* (2018), p. 30.

[19] *Keras, howpublished=https://keras.io/api/*. (accessed on 4 January 2024.

[20] Ios Kotsogiannis et al. "Architecting a Differentially Private SQL Engine." In: *CIDR*. 2019.

[21] Sofia Ira Ktena et al. "Distance Metric Learning Using Graph Convolutional Networks: Application to Functional Brain Networks". In: *Medical Image Computing and Computer Assisted Intervention MICCAI 2017*. Ed. by Maxime Descoteaux et al. Cham: Springer International Publishing, 2017, pp. 469–477. ISBN: 978-3-319-66182-7.

[22] Fang Liu. "Generalized gaussian mechanism for differential privacy". In: *IEEE Transactions on Knowledge and Data Engineering* 31.4 (2018), pp. 747–756.

[23] Frank McSherry. "Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis". In: *Commun. ACM* 53.9 (2010), 89–97. ISSN: 0001-0782. DOI: 10.1145/1810891.1810916. URL: https://doi.org/10.1145/1810891.1810916.

[24] Frank McSherry and Kunal Talwar. "Mechanism Design via Differential Privacy". In: *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*. 2007, pp. 94–103. DOI: 10.1109/FOCS.2007.66.

[25] John Hsu Miguel A. Hernán and Brian Healy. "A Second Chance to Get Causal Inference Right: A Classification of Data Science Tasks". In: *CHANCE* 32.1 (2019), pp. 42–49. DOI: 10.1080/09332480.2019.1579578. eprint: https://doi.org/10.1080/09332480.2019.1579578. URL: https://doi.org/10.1080/09332480.2019.1579578.

[26] *Openminded, howpublished=https://www.openminded.org/.* (accessed on 31 December 2023.

[27] Pranav Subramani, Nicholas Vadivelu, and Gautam Kamath. "Enabling fast differentially private sgd via just-in-time compilation and vectorization". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 26409–26421.

[28] *Tensorflow-Privacy DP-SGD Tutorial, howpublished=https://www.tensorflow.org/responsible$_a$i/p*
(accessed on 4 January 2024.

[29] Yue Wang, Xintao Wu, and Donghui Hu. "Using Randomized Response for Differential Privacy Preserving Data Collection." In: *EDBT/ICDT Workshops*. Vol. 1558. 2016, pp. 0090–6778.

[30] Dan Zhang et al. "EKTELO: A Framework for Defining Differentially-Private Computations". In: SIGMOD '18. Houston, TX, USA: Association for Computing Machinery, 2018, 115–130. ISBN: 9781450347037. DOI: 10.1145/3183713.3196921. URL: https://doi.org/10.1145/3183713.3196921.

[31] Danfeng Zhang and Daniel Kifer. "LightDP: Towards automating differential privacy proofs". In: *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*. 2017, pp. 888–901.

[32] Shiliang Zhang et al. "Evaluation of Open-Source Tools for Differential Privacy". In: *Sensors* 23.14 (2023), p. 6509.

Appendix A

First Appendix

Appendix B

Contents of Included CD–ROM

CD–ROM included to the thesis contains following files:

- `/file1` — First file
- `/file2` — Second file