Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

Republic of Algeria Democratic and Popular
Ministry of Higher Education and Scientific Research
University of Abou Bekr Belkaïd - Tlemcen
Faculty of Sciences
Department of Computer Science

FINAL YEAR PROJECT THESIS

For the attainment of the Master's degree in Computer Science

Specialty: Intelligent and Decision Models (M.I.D)

Theme:

Differential Privacy in Deep Learning: A Web-Based Prediction Service for Breast Cancer Diagnosis.

Realized by:

- **KORBAS Fatima Zohra Narimene**
- **BENGRINE Rachida**

Presented on June 26, 2024, before the jury composed of:

Dr. BERRABAH Sid Ahmed	President
Dr. BENAMAR Abdelkrim	Examiner
Dr. HADJILA Fethallah	Supervisor
Dr. BENLEDGHAM Rafika	Co-Supervisor

Academic Year: 2023 / 2024

ACKNOWLEDGMENT

ACKNOWLEDGMENT

First of all, we would like to express our gratitude to Allah, the All-Merciful, for giving us the strength, health, willpower and patience to be able to complete this important and memorable work in our lives.

The completion of this thesis has been possible thanks to the help of several people to whom we would like to express our gratitude.

Firstly, we would like to thank our dissertation supervisor, Mr HADJILA, our teacher, and our co-supervisor, Mrs BENLEDGHAM, for their patience, availability and, above all, their sound advice, which helped to fuel our thinking. And our knowledge of a very interesting field.

Then we want to give our sincer gratitude and thank to the jury composed of Dr. BERRABAH Sid Ahmed as a President and Dr. BENAMAR Abdelkrim as an Examiner for accepting to assist and discuss our thesis.

Our parents, for their constant support and kindness, and our families for their encouragement.

To our friends and, above all, to some of our classmates who have been a constant support, who have been there from near and far with their different characters and humours, making us all a family.

To all these people, we offer our thanks, our respect and our gratitude.

Content

Content

Content

Content

CHAPTER 3
Differential Privacy in Deep Learning

CHAPTER 4
Implementing and deploying a Privacy-Preserving Model: A web-based prediction service

Content

LIST OF FIGURES

LIST OF FIGURES

ACRONYMS

ACRONYMS

AI	Artificial Intelligence
Adam	Adaptive Moment Estimation
CNN	Convolutional Neural Network
DL	Deep Learning
DNN	Deep Neural Network
ML	Machine Learning
ReLU	Rectified Linear Unit
ResNet	Residual Networks
FNN	Feedforword Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
GAN	Generative Artifial Network
SGD	Stochastic Gradient Descent
DP	Differential Privacy
HE	Homomorphic Encryption
FL	Federated Learning
SMC	Secure Multyparty Computation
DP-SGD	Differential Privacy S
DP-Adam	Differential Privacy Adaptive Moment Estimation

INTRODUCTION

INTRODUCTION

Context

Deep learning is revolutionizing numerous industries, but the widespread use of personal data for training these models raises significant privacy concerns.
This thesis addresses these challenges and offers guidelines for the responsible advancement of deep learning.
The primary focus of this thesis is to investigate the privacy issues associated with deep learning. It aims to align the advancements in deep learning with the principles of data privacy.
By increasing researchers' awareness of privacy issues in deep learning, the thesis seeks to promote the development of models that provide valuable insights while preserving the privacy of the data used.
Furthermore, the thesis proposes a framework that combines the benefits of deep learning with stringent privacy preservation. It includes practical recommendations for designing deep learning technologies responsibly and discusses potential legal and ethical challenges.

Problem Statement

The problem is that the widespread adoption of deep learning comes at the cost of increased risks to individual privacy, as sensitive personal information can be vulnerable to various attacks, such as reconstruction attacks, model inversion attacks, and membership inference attacks. This poses significant ethical, trust, and regulatory challenges that need to be addressed to ensure the responsible and trustworthy development of deep learning technologies.

Contributions of the Thesis

This thesis makes the following key contributions:
1. Provides a comprehensive review of the fundamentals of deep learning, its applications, and the privacy challenges associated with its widespread use.
2. Presents a detailed analysis of privacy threats and attack vectors in deep learning models, equipping readers with a thorough understanding of the privacy risks.
3. Conducts an in-depth study of differential privacy, a state-of-the-art privacy-preserving technique, and its application in the context of deep learning.

4. Explores the advantages, limitations, and future prospects of integrating differential privacy into deep learning models, offering insights for both researchers and practitioners.
5. Proposes a framework for balancing the benefits of deep learning with the imperative of protecting individual privacy, providing guidelines for the responsible development of deep learning technologies.

Outline of the Manuscript

Chapter 1: Fundamentals of Deep Learning This chapter provides a comprehensive overview of the fundamentals of deep learning, covering key concepts, architectures, and applications.
Chapter 2: Privacy Threats in Deep Learning This chapter examines the privacy threats inherent in deep learning, highlighting the risks associated with the use of personal data in training models.
Chapter 3: Differential Privacy in Deep Learning This chapter defines and details the architecture of how differential privacy is integrated into deep learning, explaining the mathematical foundations and implementation techniques.
Chapter 4: Implementing and Deploying Differentially private Model This chapter presents the implementation of a differentially private model that achieves a high level of privacy with moderate performance degradation under specific conditions. Additionally, it discusses the deployment of this model as a web-based prediction service for breast cancer diagnosis.

CHAPTER

1

FUNDAMENTALS OF DEEP LEARNING

1.1 Introduction

Lately, in the 21st century, machine learning or ML for short, has emerged as a crucial component of contemporary civilization, operating largely behind the sciences to drive groundbreaking technological advances. Its applications cover a wide range from medical, engineering and other fields. one of machine learning most powerful sub-field is deep learning (DP), which draws inspiration from the complex structure and functions of the human brain.

1.2 Background

“ As with trees, deep learning didn’t just appear one day out of thin air. It all started with a seed, a seed that grew and grew to create deep learning[1] ” The origins of neural networks traced to the 1940s, namely to 1943, when Walter Pitts and Warren McCullough created a computer model of the human brain based on a neural network. Henry J. Kelley is recognized for developing the principles of a continuous back-propagation model in 1960. In 1962, Stuart Dreyfus devel- oped a more basic version that was based just on the chain rule. in 1965,Alexey and Valentin Grigoryvich introduced the multilayer perceptron with a polynomial activation function for the first time[2] . in the early 1970s, Seppo Linnainmaa introduced back-propagation into computer code , and then in the intervening years, back-propagation research has advanced dramatically[3]. The era of 1990s marked the beginning of the development of convolutional neural networks, or CNN. Moreover,Yann Le Cunn was instrumental in bringing CNN to where it is now, using back-propagation to recognize handwritten digits[4]. New possibilities for deep learning in science, fashion and the arts have been opened up due to the ability of GANs, invented by Ian Goodfellow in 2014, to aggregate real data[5].

1.3 Definition

Deep learning (DL) is a sub-field of Machine learning(ML) that is a branch of Artificial intelligence(AI) (Figure 1). unlike traditional machine learning , deep learning does not require any human-designed instructions to work due to the presence of several more layers of neurons in the artificial neural networks inspired by the structure and function of biological neurons in the human brain ,giving us deep neural networks that learn and make intelligent decisions and extract meaningful patterns from large amounts of data. [6]

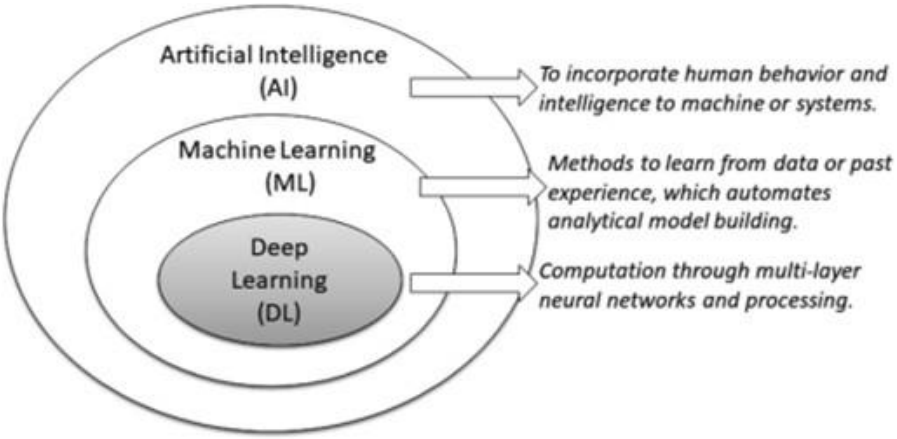


Figure 1.1: An illustration of the position of deep learning (DL), comparing with machine learning (ML) and artificial intelligence (AI) [6]

1.3.1 Differences Between Deep Learning and Traditional Machine Learning

As we explained earlier, Deep Learning is a sub-domain of ML, but one with different capabilities. Deep learning and traditional machine learning differ primarily in their approach to feature extraction, data requirements, computational demands, and application areas. Traditional machine learning relies on manual feature engineering, where the user must identify and extract the most important features from the input data, using simpler algorithms like decision trees and linear regression [7]. This makes it suitable for smaller datasets and more interpretable models, with lower computational power requirements. In contrast, deep learning uses deep neural networks that automatically learn and extract features from large amounts of raw data, making it highly effective for handling unstructured data such as images and text [7] . This approach requires significant computational resources, often utilizing GPUs, and excels in complex tasks like computer vision and natural language processing, although it comes with increased training time and reduced model interpretability. This difference in feature extraction is crucial, as depicted in the Figure bellow:

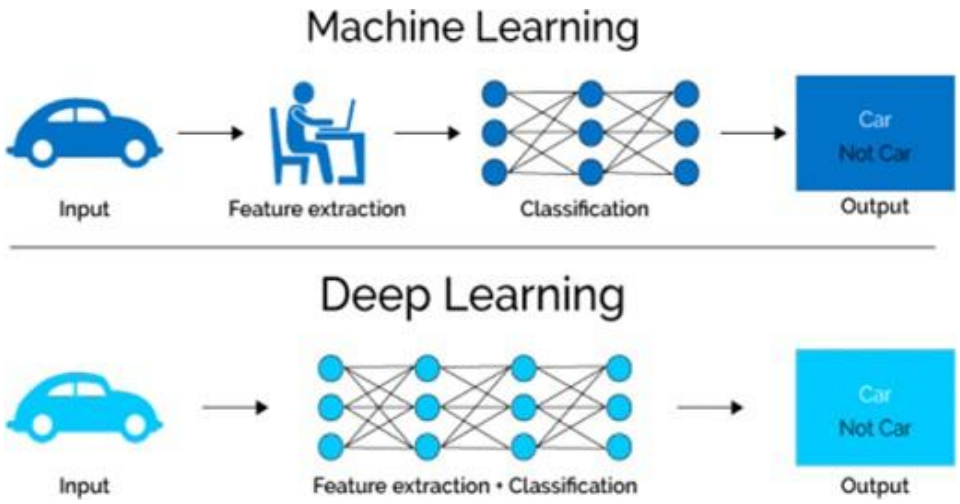


Figure 1.2: Learning Vs ML process [7]

1.4 Motivations for deep learning

This question addressed by a number of performance metrics, such as: [8]

1.4.1 Universal Learning Approach:

Because DL possesses the inclination to perform in about all application domains, this kind is often termed as universal learning.[8]

1.4.2 Robustness:

Generally speaking, DL approaches do not necessitate precisely constructed features. Rather, the optimum traits are automatically learned in relation to the task at hand. As a result, the input data becomes robust against typical changes. [8]

1.4.3 Generalization:

Alternatively, one type of data or a specific application can use that specific DL technique, a process commonly known as transfer learning (TL) . In addition, it is helpful in many problems that information is scarce. [9]

1.4.4 Scalability:

DL is also very scalable. ResNet [10], created by Microsoft, has 1202 layers and is often used at the supercomputer level. Another large-scale project that has developed new frameworks for networks is Lawrence Livermore National Laboratory (LLNL), where thousands of nodes can be implemented as well.
[9]

1.5 Structure of a Neural Network

1.5.1 Architecture

- **Neurons:** Artificial neurons are the elementary components of the artificial neural networks based on the structure and the action of neurons within the human brain Figure1.3 . An artificial neuron accepts inputs from other neurons, applies the activation function to calculate an output, and disseminates it to other neurons having connection.[10]
  Figure5 shows a typical artificial neuron.

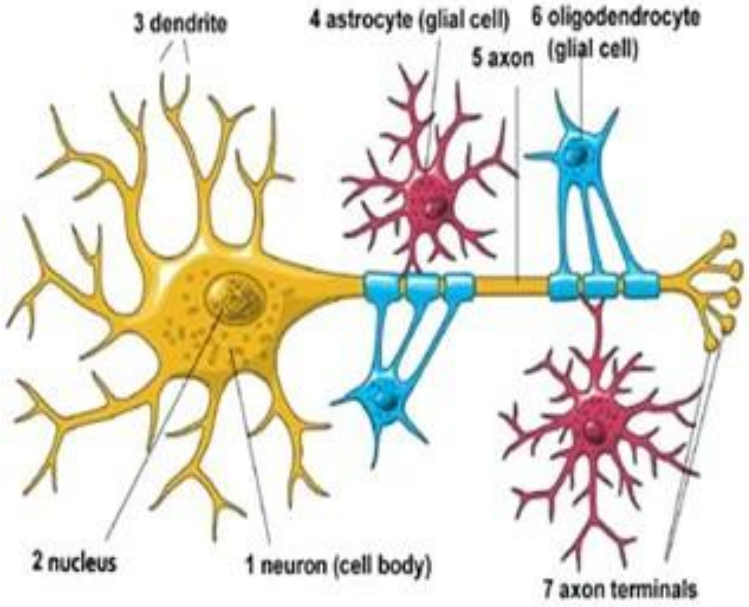


Figure 1.3: A biological neuron [10]

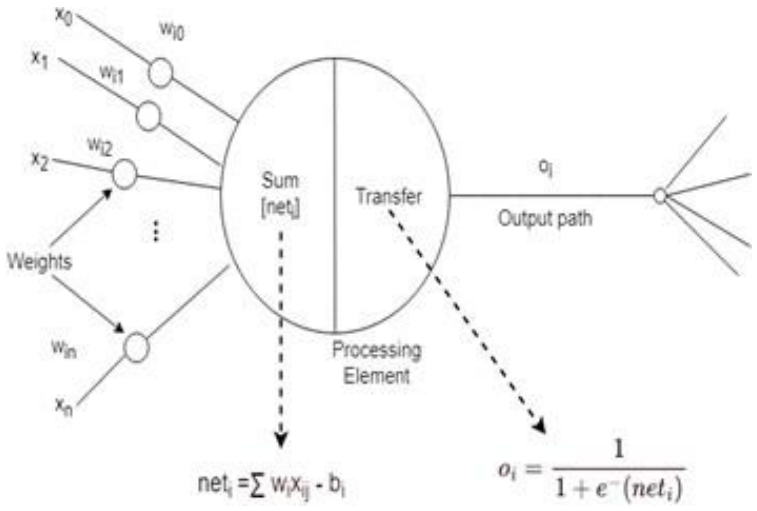


Figure 1.4: An artificial neuron
[10]

- **Layers:** Neurons are organized into layers - input, hidden (multiple in deep learning), and output layers: [10]
  - **Input:** Gets the input data for processing, the number of neurons equal to the number of features in the data.
  - **Hidden:** Allow the actual processing of data so that the network is able to learn complex relationships.The number and size of layers help determine the network's ability to learn and generalize information.
  - **Output:** Final layer or decision layer Gives the final output or the recommended result.

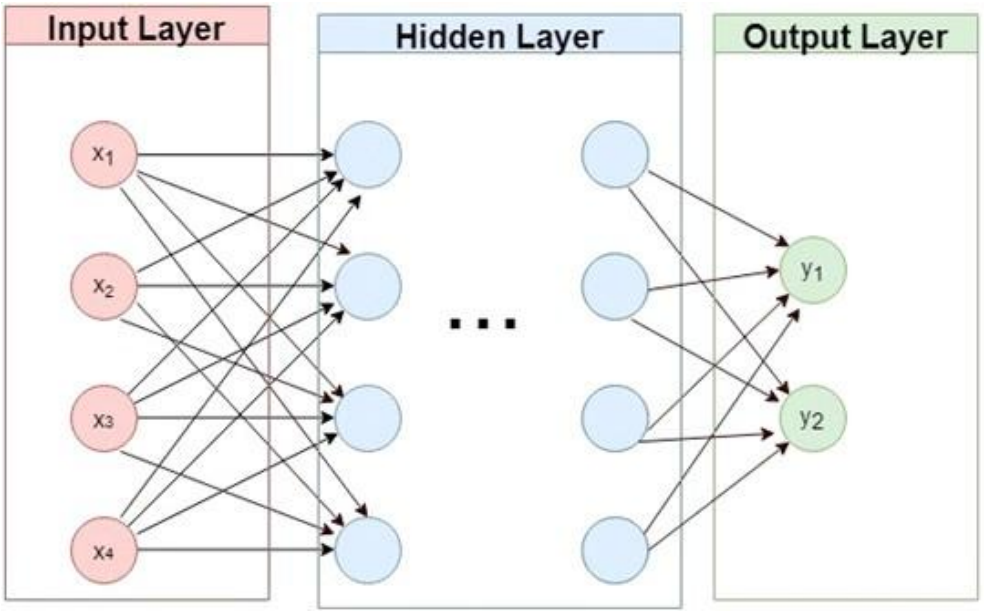


Figure 1.5: The three types of layers[10]

- **Weights:** Numerical values on connections between neurons influence information flow. [11]
- **Biases:** Constant values are added to neuron inputs, adjusting the activation function. [11]
- **Activation Functions:** Introduce non-linearity for complex pattern learning.Here are some examples about it:
  - **ReLU (Rectified Linear Unit):** Is defined as

    $$f(x) = \max(0, x)$$

    . It introduces non-linearity into the model, helps solve solve vanishing gradient problem and learn complex patterns[12]
  - **Sigmoid Function:** Usually utilized in two-class classification problems[12]
  - **ReLU (Rectified Linear Unit):** Is defined as

    $$f(x) = \max(0, x)$$

    . It introduces non-linearity into the model, helps solve solve vanishing gradient

problem and learn complex patterns. [12]

- **Tanh (Hyperbolic Tangent):** Better gradients than sigmoid but used only in cases where the output is zero-centered. [12]

Figure **1.6** shows some of the more common activation functions:

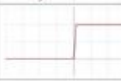
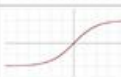
Name	Plot	Function, $f(x)$	Derivative of f, $f'(x)$	Range
Identity		x	1	$(-\infty, \infty)$
Binary step		$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$	$\begin{cases} 0 & \text{if } x \neq 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$\{0, 1\}$
Logistic, sigmoid, or soft step		$\sigma(x) = \frac{1}{1+e^{-x}}$ [1]	$f(x)(1-f(x))$	$(0, 1)$
tanh		$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$1 - f(x)^2$	$(-1, 1)$
Rectified linear unit (ReLU)[11]		$\begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$ $= \max\{0, x\} = x\mathbf{1}_{x>0}$	$\begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \\ \text{undefined} & \text{if } x = 0 \end{cases}$	$[0, \infty)$

Figure 1.6: the more common activation functions .

1.5.2 Training Process

- **Forward Propagation:** A sweep through the network is made in the forward direction, in order to produce the predictions. [12]
- **Loss Function:** Gives the difference between the data the network predicted and the real data it should have predicted. Typically, frequent loss functions encompass Mean Squared Error (MSE) in the regression type of problems and Cross-Entropy loss when the problem type is classification. [13]
- **Back-propagation:** The training core mechanism, which is encompassing of: [13]
  - **Computing Gradients:** Computing the derivatives and the relation between the weights and biases with the loss using the chain rule in calculus. [13]
  - **Updating Weights:** Bringing the weights and the biases closer to their optimal values by penalizing the weights and the biases in the gradient direction, It makes the process smooth by using optimization algorithms like Gradient descent or Stochastic gradient descent or Adam optimizer. [13]
    - **Stochastic Gradient Descent:** Stochastic Gradient Descent**SGD** is an optimization algorithm that minimizes a function by iteratively adjusting parameters using a randomly selected subset of data.
    - **Adaptive Moment Estimation:** Adaptive Moment Estimation (Adam) is an optimization algorithm that combines momentum and adaptive learning rates. It calculates personalized learning rates for each parameter based on gradient moments, leading to faster convergence and enhanced model performance.
  - **Iterative Process:** The two procedures of forward propagation and back-propagation are done over and over to enhance the performance of the network in achieving direct input/output mapping.

1.6 Classification of Deep Learning Approaches

Like machine learning, deep learning approaches can be classified as supervised, semi-supervised and unsupervised. In addition, there is another category of learning called reinforcement learning (RL) or Deep RL (DRL), which are often addressed as part of semi-supervised or sometimes unsupervised learning approaches. [14]

- **Deep supervised learning:** It is a learning technique that uses labelled data. In the case of supervised DL approaches, the environment has a set of inputs and corresponding outputs. It will then iteratively modify the network parameters to better approximate the desired outputs. There are various supervised learning approaches for deep leaning, including deep neural networks (DNN), Convolutional neural networks (CNN), recurrent neural networks (RNN), long-short-term Memory (LSTM) and closed recurrent units (GRU). [14]
- **Deep unsupervised learning:** This technique enables the learning process to be done without the use of labelled data since such data availability is limited at times (i.e., no labels are required). It also becomes apparent that here, the agent acquires information regarding the significant features or internal representation that contains sufficient information to identify the unknown structure or relation in the input data. Methods such as generative networks, dimensionality and clustering are often classified to the category of unsupervised learning. Many of the DL family have offered good results in non-linear DRR and clustering tasks; among these are RBMs, AEs and GANs which are relatively recent in development. Furthermore, many RNNs such as GRUs and LSTM approaches have also been used in a broad category of unsupervised learning applications as well. [15]
- **Deep reinforcement learning:** Reinforcement learning (RL) is a process of an agent's interaction with an environment or system whereby it identifies those actions that cause a gain or reward. In this process, an action that is perceived to be good earns the subject a positive consequence, while a bad action earns the subject a punishment. Neural reinforcement learning is the combination of neural networks with RL to make the agents learn in different simulations. While model-based RL attempts to model or mimic the environment, on the other hand, model-free RL directly learns through interactions, and Q-learning popular among them. DRL employs established deep learning architectures such as Deep Neural Networks (DNNs) and Convolution Neural Networks (CNNs). Some of the uses of reinforcement machine learning cover Robotics, video games and control problems. [6]

1.7 Training Deep Neural Networks

Training in deep learning is the process of feeding the model with data, testing its performance, and then making changes to the internal structure of the model to improve its prediction capability. Here's a breakdown of the key steps:

1.7.1 Data Preparation

This stage is very important in realizing the deep learning project. It involves:

- **Data Collection:** Acquiring data pertinent to the task which you want the model to accomplish. This could be in the form of images, text, audio or even numeric data. [16]

- **Data Cleaning:** Data correctness: the data should be clean, without any errors. This may include processes such as duplicate elimination, handling of missing values, healing of discrepancies, and standardization of inputs, etc. [17]

- **Data Augmentation:** Any process that results in an expansion of data or its heterogeneity, such as flipping images, adding noise to them, or paraphrasing the texts. Such strategies help the model to generalize better when faced with unseen data. [18]

1.7.2 Model Training Process

This is where the magic happens, The model learns from the data through steps that are follow:

- It is crucial to start the model weights correctly from scratch, i.e. to initialize it properly, for example using the Xavier initialization. [19]

- When training deep networks, implement batch normalization to make inputs to each layer more stable instead of a normal distribution to increase the learning rate and reduce dependence on aspects such as initialization. [17]

- Feed forward the inputs to perform computations on the network and calculate the output and loss. [16]

- You need to differentiate the loss with respect to the weights to get gradients, and this can be done with back-propagation. [16]

- Calculate new weights for model parameters are computed using gradient descent and thus can include techniques such as momentum and adaptive learning rates. [16][19]

- Iterate over many epochs, if you validate on a separate set, check performance and stop iterating when the validation set's performance ceases to improve significantly. [19]

1.7.3 Evaluation Metrics

- **Accuracy:** The ratio of correct previsions on the total number of sequences, i.e., the average of the models performance. Precision is a straightforward measure, yet may be deceptive if imbalanced data sets were under consideration. [20]

- **Precision:** Shall express the measure of true positive identification according to the given total number of positive predictions made by the model. Precision is the accuracy of the model meaning that the model is precise when if correctly points out the relevant features of the image passed to it. [20]

- **Recall:** The percentage of true positive instances out of all existing positive instances, and is calculated as the number of real positive instances, which have been also classified as positive, multiplied by 100 and divided by the number of all real positive instances. Recall is the measure of how the model is able to remember what all existent instances are. [20]

- **F1 Score:** The F1-score that represents the average of the measures of precision and recall, which would give an average measurement rather than a favoring one or another. [20]

- **ROC-AUC: :** The usually normalized measure of performance derived from the curve which is a graphical representation of Sensitivity/True Positive Rate against the plot of Fallacy/False Positive Rate at different classification boundaries. ROC-AUC evaluates the model based on the ranking of the outputs and should be used if the classes are of similar sizes. [20]

1.7.4 Deployment

Model deployment (release) is one of the process of deploying deep learning models to make decisions in production from real data. After it has been deployed, the model also requires validation to ensure that every step such as data acquisition, data preprocessing, training and testing and the likes is optimally adjusted to be executed automatically without human interference[21].

- **Deployment Strategies:**
  - **Shadow Deployment:** This involves the ability to create a development replica of a model seamlessly alongside the current live server without impacting it. The advantage of this is that potential problems with the new release and how it will behave in the new environment can be identified before several organizations and people are affected [21].
  - **A/B Testing:** This involves balancing the traffic between two or more versions of a model with the goal of noting the one that performs better. It is then deployed in the version that showed the efficiency in response to the queries made of it [21].
  - **Multi-Armed Bandits:** This strategy directly applies the reinforcement learning technique in the context of a continuously changing traffic distribution function so as to control traffic towards the winning model versions, as well as seek further traffic changes by exploring other model versions [21].
  - **Canary Testing:** This is often done by updating a limited number of application instances or the traffic they receive before making a new model version available to all the users. Therefore, there is a way to monitor new version performance without making it fully operational [21].
- **Deployment Considerations:**
  - **Scalability and Performance:** Ensuring that is crucial for optimal model reliability, which is why it's necessary to determine the need for cloud, on-site, hybrid, and multi-cloud solutions to address changing traffic and resource loads.
  - **Cost and Budget:** Organizations must evaluate long-term usage patterns and calculate total cost of ownership (TCO) for deployment models. Cloud-based deployments, with their pay-as-you-go model, can be more cost-effective than on-premises options.
  - **Security and Compliance:** Effective collaboration is vital for deployment, focusing on data encryption, access controls, and regulatory compliance.
  - **Iteration, Testing, and Performance Comparisons:** Iteration and testing are crucial for model deployment, using strategies like shadow deployments, A/B testing, and CI/CD to ensure new models perform as well or better than existing ones.

1.8 Architecture of Deep Learning Models

The classification of deep learning involves having many models where each one is created to work best on a particular data type or problem. Here's an overview of some of them:

1.8.1 Feedforward Neural Networks

Feedforward Neural Networks (FNNs) is the easiest form of artificial neural network to understand and has a simple architecture that has the following elements: Input layer(s) , one or many hidden layers which are fully connected layers, and output layer.Her consist of a number of nodes and can only process information in one direction, that is from input layer to the output layer.FNNs are commonly used for basic tasks like classification and regression prob- lems. They are trained using gradient-based backpropagation algorithms to minimize a loss function. Backpropagation adjusts the weights and biases of the network after each pass through the network.[21] Few popular example areas where usage of FNNs is prevalent are, Hand writing recognition, simple image recognition, text recognition. [22] Figure 7 shows a typical feedforward neural network :

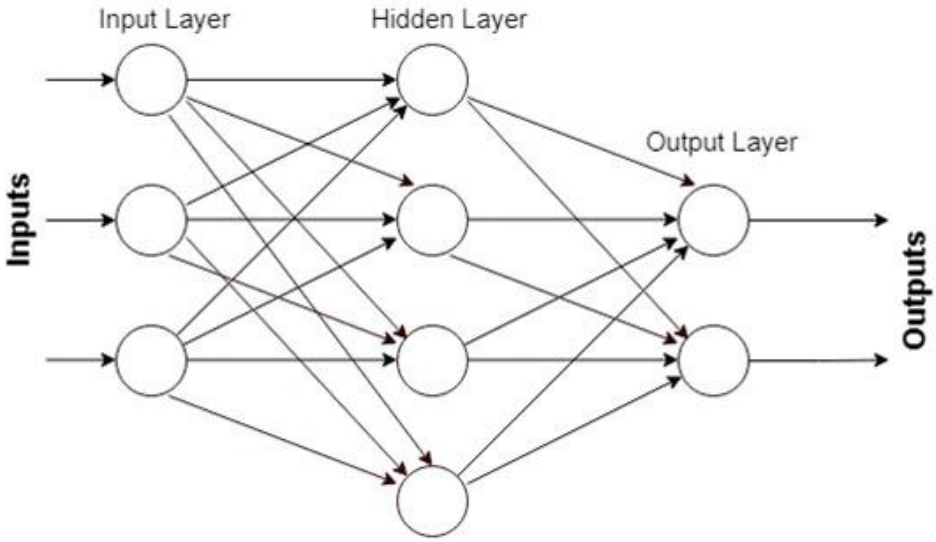


Figure 1.7: A typical feedforward neural network .

As an example of FNNs , we have convolutional neural networks (CNNs) and generative adversarial networks (GANs):

1. **Convolutional Neural Networks:** A convolutional neural network (CNNs) applies what is known as a convolution instead of matrix multiplication in at least one of the layers. These specialized networks are designed for the extraction of some relevant features from the data points, which are locally correlated. The feature maps delivered by the convolutional kernels are then passed into a non-linear processing element known as an activation function which enables the model to uncover representation and introduce non-linearity into the feature map. This is the reason why this function is non-linear, which creates different activation models, hence making it easier to learn semantics of the images. [23]

   A CNN topology consists of many layers of learnings including the convolutional layers, the non-linear processing, and the sub-sampling layers.[23]

- **Convolutional Layer:** The initial operation performed on an input image is called convolution, that helps in feature extraction. Convolution focuses on the associativity of the pixels by utilizing small rectangles of input data in feature learning. It is a mathematical operation that should be performed on two inputs, such as an image matrix and a filter or kernel. [24][25] Figure 10 shows a simple filter operation for a convolution step:

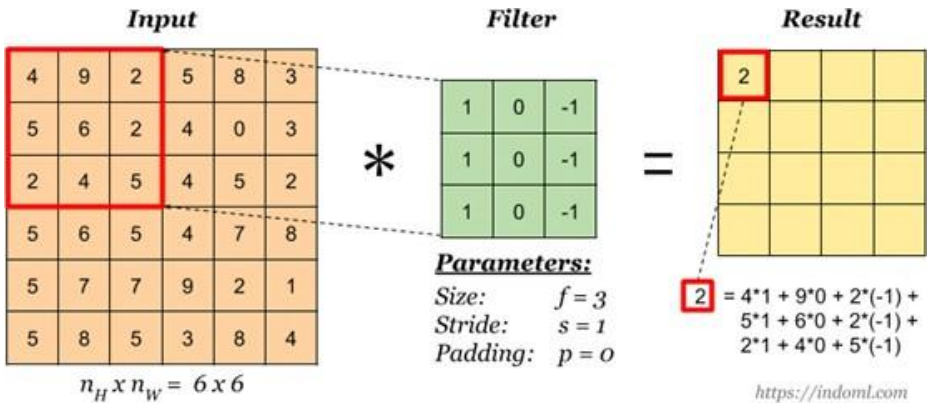


Figure 1.8: Explanation example of the convolution operation.[25]

- **Pooling Layer:** The pooling layer is typically placed between two consecutive layers of convolution and aims at diminishing the number of sub-sampling operations in cases where the images are too large. The process of pooling is also referred to as sub-sampling or down-sampling, whereby the new maps have reduced dimensions in comparison with the previous one; however, important information is not lost [22]. This is a simple operation that consists of replacing a square of pixels (generally 2x2 or 3x3) with a single value, depending on the type of pooling Figure1.9 :
  - **average pooling:** takes the average of all the pixels in the selection :
  - **maximum pooling:** Takes the pixel with the highest value of all the pixels in the selection.

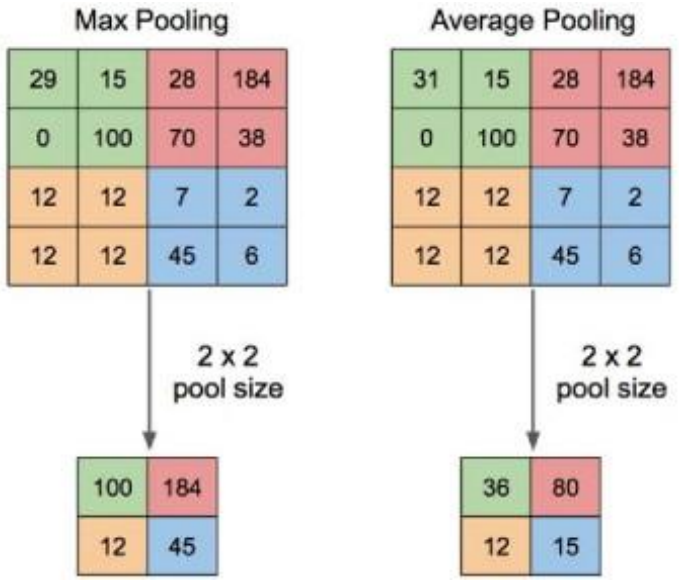


Figure 1.9: Max pooling and Average pooling illustration.[26]

- **Fully connected layer:** The last layer on the architecture is the fully connected layer, which is comparable to the fully connected network found in classical models. The result from the first stage, including unconventional convolution and repeated pooling, perform a fully connected layer working to calculate the dot product of the weight vector and the input vector to obtain the final output.[25] Figure 12 shows a fully connected layer:

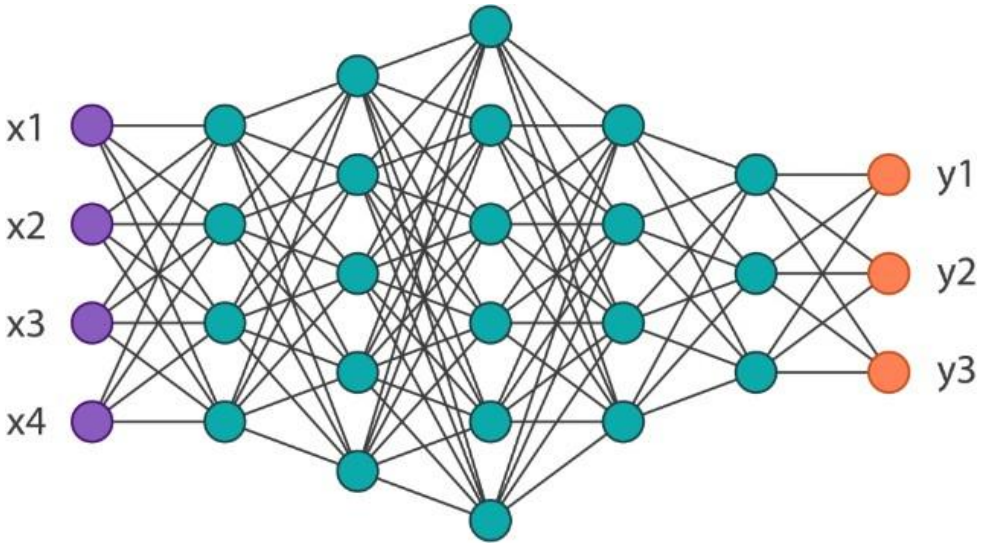


Figure 1.10: fully connected layer.[27]

Several CNN architectures exist, some of which are listed below:

- LeNet-5 (Famous CNN used to identify and recognise patterns in a series of handwritten postcodes [28])
- AlexNet [29]
- VGGNet (Visual Geometry Group) [30]
- GoogLeNet [31]
- ResNet [32]
- **Applications of CNN Networks:**
- Decoding Facial Recognition.
- Document rendering.
- Video Processing.
- Heart Disease Detection.
- Video Games.

1. **Generative Adversarial Networks** Generative Adversarial Networks (GANs) are a type of deep learning architecture that consists of two neural networks: As we can see in the beneath sections, a generator and a discriminator. The aim of a GAN is to create data points for a new distribution that is in some way related to the original data points.[33] The generator network is responsible for coming up with new samples of data, whereas the discriminator network is one that assesses the generated samples to determine if they come from the real input distribution or if are fake or synthetic data as produced by the generator.[33]
   In the process of training, we have the generator and discriminator both playing a game, a method used during training. The generator works to output new, believable data sets that will confuse the discriminator, whereas the discriminator attempts to accurately classify between the real and the generated data sets. This forms an adversarial loop that runs in a cycle until the generator has created a sample that looks like real data, such that the discriminator will not be able to set them aside as fake data.[33]

   There are numerous studies that have examined the use of GANs in diverse application areas, such as image synthesis, video creation, style transfer, and data enhancement. In particular, the VAEs have been reported to produce relatively good performances in creating realistic samples of data and have emerged as a common choice when carrying out generative modeling in deep learning.[33]

   Figure 14 shows how GAN implementation works:

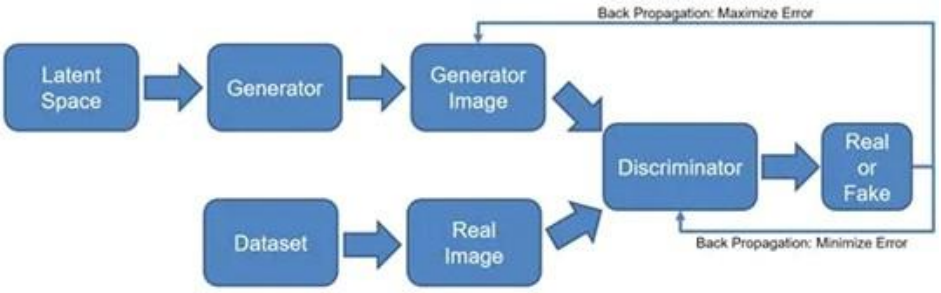


Figure 1.11: GAN implementation.[34]

1.8.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are another diverse variant of neural networks that are commonly employed in natural language processing. They are called recurrent since work on the same data, and the output for each element depends on the outcomes of the previous ones. RNNs can be viewed from another perspective as having a 'memory'- a memory of the computation that has been carried out thus far. Technically, RNNs have the ability to use information in sequences that have an arbitrary length, but in reality they can only see a few steps behind. It is a type of neural network which uses past predictions as input through the hidden states.[35]

They take the form shown in Figure1.12:

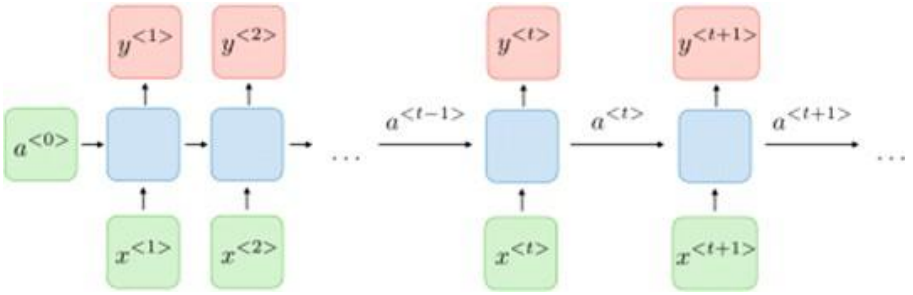


Figure 1.12: Architecture of RNN.[35]

- **maximum pooling:** Backpropagation in time (BPTT) is used to train RNNs because it is a modification of the normal backpropagation algorithm for input data in a sequence.BPTT recalculates the network weights in the manner of, updating, the loss function gradients with reference to the weights, as well as the fact that the data is sequential in nature. [33]

- **Applications of RNN Networks[34]:**

- Machine Translation.

- Text Creation.

- Captioning of images.

- Recognition of Speech.

- Forecasting of Time Series.

1. **Long Short-Term Memory Networks:** One of the disadvantages of standard RNNs is the disappearance of the gradient, and given the inefficiency of restoring it through optimal parameter setting in the first layers (wasted time and a high cost in terms of computing power), long-short-term memory (LSTM) networks have emerged. Invented by computer scientists Sepp Hocheriez and Jurgen Schmidhuber in 1997, they offer a solution to this problem. RNNs built with LSTM units classify data into short- and long-term memory cells. This enables RNNs to determine which data is important and needs to be stored and fed back into the network. It also enables RNNs to determine which data can be forgotten, giving the model even better performance.[36] The key components of an LSTM unit are [37]:

    - **Cell:** Recall values at any arbitrary time instant
    - **Input gate:** With the help of this property, one can select which values from the current input and the previous output need to be added to the cell state.
    - **Forget gate:** Determine which information is to be omitted from the previous cell state.
    - **Output gate:** which of the cell state parts will be utilized in mapping the output

They control the input and output of information within the cell, which helps LSTMs to memorize or output information as it passes through it.[37]

- **Applications of LSTM [37]:**
- Language Modeling
- Speech Recognition
- Time Series Forecasting
- Anomaly Detection
- Video Analysis

1.9 Deep Learning Tools and Frameworks

These frameworks make the use of complicated models easier and provide precise methodology to develop as well as train the Deeplearning models effectively, here is some of these frameworks [38]:

- **TensorFlow:** A machine learning software library which is used in Deep learning that is also known as Tensorflow. It allows the scale out of neural networks with many layers through data flow graphs.

- **PyTorch:** It is a deep learning library that offers an extensible execution framework with a dynamically constructed computation graph that has made it suitable for research purposes and applications. It is good for opting with Python and the Numpy stack, and further boosts computation on GPUs.

- **Keras:** A varied library that has been developed with the capability of recreating neural networks in relatively simple ways. It powers many applications, supports various backends including TensorFlow and provides pretrained computer vision models.

1.10 Challenges of Deep Learning

While deep learning offers remarkable capabilities, it also presents significant challenges such as:

- **Feature Engineering :** Use of meticulous textures makes the feature having low generality and possibility to pass from one system or fault type to another one. Feature engineering is also a manual process consisting of the expert's work, and thus cannot be easily integrated into models as the number of input parameters to monitor grows[39].
- **Privacy preservation :** Deep learning faces a critical hurdle in maintaining privacy due to its data-intensive nature. As these models delve into sensitive information like medical records or user behavior, safeguarding privacy becomes paramount].
- **Robustness and Generalization capability :** refers to a model's ability to maintain performance in the face of input variations and generalize well to new data. Deep learning models can struggle with overfitting and sensitivity to noise or attacks, necessitating techniques like regularization and robust training methods to improve their resilience and performance on unseen data.
- **Interpretability :** Interpreting the results to a level that is comprehensible to the field experts is an essential requirement for PHM; however, with developing deep learning models, it's quite a complex issue[39].
- **Overfitting :** When the model supposes to learn from training data to gain high accuracy and at the same time does not generalize when it is tested on unknown data, then that is called overfitting. It occurs when a model has an extended period learning noise or irrelevant features from the training dataset usually resulting in poor performance of the model on new dataset and consequently decreasing reliability of the given model [40]. So mitigate overfitting in deep learning models, various strategies can be employed, here is some of them:
- **Data augmentation:** Operations of rotation, flipping, and adjusting color allow creating new residuals of the training data set that helps the model to generalize.[40]
- **Early stopping:** This is because training is halted when the validation loss begins to rise; this reduced the risk of overfitting by the optimization process at the noise in the data.[40]
- **Hyperparameter Tuning:** The pre-specified hyperparameters, including learning rate and set size, seem to influence results and the occurrence of overfitting.[41]

1.11 Conclusion

In this chapter, we have exhibited an introductory illustration of deep learning by reviewing its history, concepts, and neural network structures. Among others, we reasoned about the rationale for such interest in deep learning, pointing to important benefits that set this approach apart from more conventional paradigms in machine learning. The chapter also went over the process thus involved in the training of neural networks inclusive of preparation of data, training the model and the final evaluation. Depending on quantized layers, we discussed different deep learning techniques and explored the structures of CNNs, GANs, and RNNs. Further, we looked at some of the common software tools and libraries available, those include TensorFlow and

PyTorch, and discussed some issues associated with deep learning including overfitting and the lack of interpretability. Looking ahead , deep learning will likely continue to progress at a very advanced rate as computational power scales and large datasets become more commonplace. More advanced and generalizable models may be developed in the future and there can be improvements on interpretability of models to be made in the future, thus its application can also diversify in fields such as self-driving cars and recommendation systems and even in the field of healthcare for instance in personalized medicine. All in all, as the research continues, deep learning is expected to be a significant key in tackling other more real life challenges in future, translating to its continued relevance to advancing technology and social aspects.

CHAPTER

2

DATA PRIVACY IN DEEP LEARNING

2.1 Introduction

Data privacy, a critical aspect of information security, refers to the handling and protection of personal data to ensure it is not misused or disclosed without consent. Deep learning models train on extensive datasets that often encompass sensitive personal information and critical details like: healthcare records, financial information, and personal identifiers, Thus, ensuring data privacy in deep learning is essential to protect individuals' confidential information from unauthorized access and misuse. [42][43]

2.2 Privacy vs. Security in Deep Learning

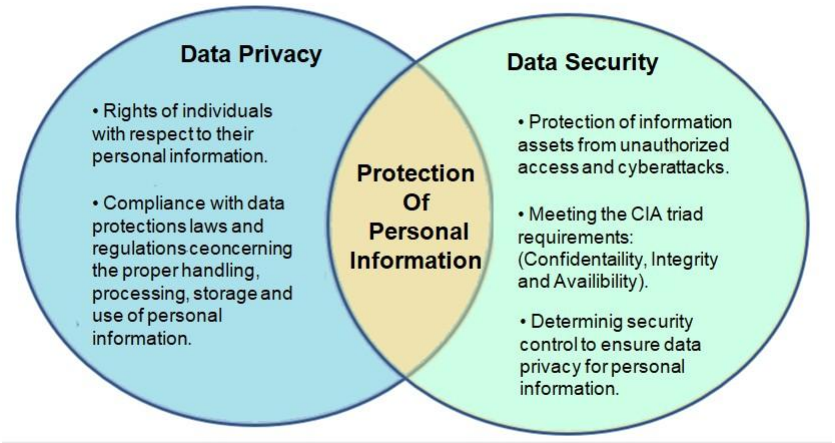


Figure 2.1: Data Privacy vs. Data Security.

2.2.1 PRIVACY:

Deep learning requires careful thought about privacy and security. Privacy concentrates on information about individuals and is the protection against unauthorised access or deduction of sensitive data used in DL models, such as medical imaging or proprietary training data. This is particularly important in areas such as national security, healthcare and finance, where sensitive and private data may be involved. The goal of privacy-preserving methods such as homomorphic encryption, secure multi-party computation and differential privacy is to enable collaborative learning of DL models without revealing private data. [42]

2.2.2 SECURITY:

On the flip side, security in DL is concerned with preventing harmful attacks on DL models. This includes guarding against adversarial attacks that attempt to modify the model's predictions via carefully developed perturbations to the input data, in order to discover the internal structure of the model, and robbery of the model parameters. These security risks may enable unauthorized access to confidential data, aid in financial fraud, or lead to the incorrect judgments being made by vital systems, among other grave outcomes. To protect DL models from these kinds of assaults, researchers have put forth a number of security techniques, such as adversarial training, input validation, and model watermarking. [43]

Privacy and security issues have been examined in great detail by the deep learning community, which has spent a lot of money to find realistic answers to these problems. Strict privacy and security measures will become increasingly essential as the use of DL expands in various segments and applications, in order to guarantee the dependability and credibility of these powerful artificial intelligence systems. [44]

When it comes to DL, there are a number of important privacy and security aspects to bear in mind:

- **Access control**
  Privacy protection is based on restricting unauthorized access to and use of data, which can only be achieved by implementing strict security measures. [45]

- **Data integrity**
  The accuracy and integrity of information must be guaranteed, in order to respect privacy and for security reasons. [46]

- **Responsibility**
  It is important to document and address privacy and security issues in line with the company's existing data policies. [47]
  As DL becomes more widespread, it will become increasingly crucial to ensure the security and confidentiality of these systems. In order to consolidate confidence in the use of DL technologies in different contexts, it will be essential to conduct in-depth research and adopt an original approach in this field. [48]

2.2.3 Cases from Deep Learning where security and privacy issues come up in different ways

The following situations clearly demonstrate that privacy and security issues arise differently in the field of DL:

- **On the privacy side: Medical imaging:** Private information about patients that is crucial to their health must be preserved by DL models trained on medical images, such as X-rays, mammography scans or MRI data. [47]

- **Financial transactions:** With DL models, we can also protect everything to do with records and their confidentiality, and a user's financial transactions, let's talk about this sector, we can also protect against credit risk and fraud detection. [47]

- **Personal assistant systems:** Since virtual assistants based on DL can have access to users' voice recognition, text and personal information, rigorous privacy precautions are needed to prevent misuse or unwanted access. [47]

- **On the security side:**
  Autonomous Vehicles: Self-driving cars use DL models for vision, decision-making and control. Unauthorized attacks can alter vehicle behavior and endanger public safety, so DL models need to be protected. [47]

- **Malware detection:**
  Trials at reverse engineering to either work around the detection system or create new malware that can evade model predictions must be avoided when using DL-based malware detection models. [47]

- **Biometric authentication:**
  DL models used for biometric recognition, such as fingerprint or facial detection, must be protected from attacks aimed at impersonating someone else or bypassing the authentication system. [47]

  Security and privacy issues may be particular to each of these situations. For example, patient privacy is an important concern in medical imaging, while safety of passengers and the safety of other drivers are the primary worries in the case of self-driving automobiles. The adoption of guidelines and mitigation approaches to address these challenges could differ as well depending on the DL application's characteristics and the domain.

2.3 The Imperative of Privacy in Deep Learning Systems

2.3.1 Ethical considerations

- **Equal opportunities and bias mitigation**
  It is possible to modify the data and algorithms used to train AI. In order to minimize

biased decisions, it is crucial to have diverse and inclusive programming teams as well as regular checks on algorithms. [49]

As for example, the medical and healthcare fields where AI algorithms are used are not carefully designed and validated, it is vital that they avoid perpetuating bias.

- **Transparency and explicability**
  To gain trust and empower users, it's essential that AI systems are transparent and explainable in their decision-making processes. [49]

- **Data security and confidentiality**
  In order to guarantee user confidentiality, trust and privacy, it is essential that AI systems are developed and deployed with strong data encryption, access controls and any other security measures needed to protect this data. [49]

- **Accountability and responsibility**
  Individuals must take responsibility for the creation, implementation and use of AI. Proper maintenance is crucial, as AI relies on sensors, data, algorithms and computing power. As healthcare providers in their field and AI developers and all parties involved must take responsibility for the ethical use of data,scrupulously complying with applicable laws, regulations and guidelines.

  In addition, they must take care to minimize the risks associated with the protection of information. [49]

- **Respect for patient autonomy**
  It is vital that users have full control over their data, including the ability to access, modify or delete it as required. On the other hand, it is imperative that they express their preferences regarding the use of artificial intelligence algorithms on their AI data, in order to foster trust, promote patient-centered care and respect ethical principles. [49]

2.3.2 User trust and acceptance

- **Patient trust**
  It is essential that users trust AI systems to be able to share their data in the event that this needs to be done to transparency as well as the opportunity for users to explain themselves which plays an important role in building user trust.

- **Trust of healthcare professionals**
  To ensure compatibility, professionals in any field need to be aware of legal standards and regulations. They also need to trust intelligent systems to deliver accurate, reliable results.

2.3.3 Compliance with regulations

- **A lack of clear legal rules**
  People affected by artificial intelligence find it difficult to obtain compensation due to the lack of clearly defined legislation concerning these technologies.

Corporate liability: Companies could be held legally responsible if their internal organization dehumanizes their employees and prevents them from taking full responsibility for their AI-related tasks.

- **New regulations**
  Financial institutions must provide information to US banking regulators on their use of AI, demonstrating that governments see this technology as a tool with a responsibility to use it.

2.4 Regulations and Legal Frameworks

The following sections discuss GDPR, CCPA, and PIPEDA, which are prominent regulations that govern the use of DL models.

2.4.1 General Data Protection Regulation(GDPR)

- **Consent**
  Consent of the concerned people is of major importance under the GDPR which is very crucial under the circumstances where DL systems demand the personal details of the concerned individuals. This means that the data of a particular individual has to be gathered properly, that the particular individual has to consent to the use of his data actively and more to the point, that this consent must be one that can easily be withdrawn. [50]

- **Data protection by design**
  The RGPD requires that an organisation to implement privacy by design, a measure that ensures that personal data is processed in a manner that does not infringe on privacy. This principle is crucial for the development of DL applications because it reveals that, at the design phase, privacy is incorporated. [50]

- **Data portability**
  According to the GDPR every person has the right to have without any cost their personal data in a structured format and in such format that can be transferred from one controller to another in a format that is customarily used. This right is significant for DL applications as it enables the patients to reclaim their data and migrate it to another firm if needed. [50]

- **Data deletion**
  The right to erasure is clearly enshrined in the GDPR as this allows the persons to request that the organizations delete their personal data in certain circumstances; for instance, where the information is no longer necessary to the purposes for which it was collected or where the person withdraws their consent. This right is essential for DL applications because it gives the right to individuals to provide control over their data and demand its erasure if needed. [50]

2.4.2 California Consumer Privacy Act (CCPA)

CCPA does not require consent, but focuses on the ability to inform and control, allowing individuals to opt out of having their personal information sold. This means that DL applications must provide information in a way where personal data will be used, and must offer a means by

Which individuals can opt out of the sale of their data.

- **Data portability**
  The law imposed by the CCPA requires companies to provide consumer data in useful formats if requested by co-consumers. This right is important when DL enforcement is present, as it gives individuals permission to control their data that has been shared with the company and to transfer it to other organizations. [49]

- **Data deletion**
  The deletion of records collected from corporate customers is a law imposed by the CCPA in cases where the customer has authorized its deletion and it is the organization that sends notifications to corporate customers informing them that the customer has requested the deletion of his or her shared record. This is important in the context of DL, as it allows individuals to know how their data has been processed and to request that it be deleted. [49]

2.4.3 Personal Information Protections and Electronique Documents Act (PIPEDA)

- **Consent**
  Implied and explicit consent fall under PIPED, this is rather challenging when developing deep learning apps that engage personal data. In other words, deep learning applications must be very perceptive around the legal basis on which the personal data of an individual is processed and the provision of reasonable opportunity to the concerned individual to withdraw their consent for the processing of their personal data. [45]

- **Data Protection by Design**
  PIPEDA explains the concept of fair treatment, which means minimizing any potential harm to the individuals, largely to maintain a reasonable expectation of the use of the information collected. This principle is particularly relevant to deep learning applications because, it makes people aware of how their data is being used and decide accordingly about the use of the data collected. [45]

- **Data Portability**
  Whereas, PIPEDA does not state that a candidate has the right to obtain his or her information in a structured format in a manner that they could transfer to another company. This specifies that deep learning applications must make sure that people own the information they share, and they can take it to another firm if they want to, yet this is not a well-identified right. [45]

- **Data Deletion**
  Personal information protection act also does not point out the right to erasure in an orderly manner. Users can ask the organization to amend the information they hold or remove the data which is misleading, but this right can be restricted according to the business requirements of the organization and the laws of the country in question. This implies that in relation to personal data, deep learning applications must make sure that they have a legal right to store the data and that individuals should be given a reasonable timeframe to ask for their data to be erased.
  Regulations and legal frameworks are important issues that influence deep learning practices since they point out rules and prohibitions related to the creation and implementation of the deep learning systems. Here are some key ways in which regulations and legal frameworks affect deep learning.
  [45]

- **Data Privacy and Protection**
  Some of the regulations include the General Data Protection Regulation (GDPR) in the EU, the California Consumer Privacy Act (CCPA) in the US, and the Personal Information Protection and Electronic Documents Act (PIPEDA) in Canada which has strict requirements over the collection, usage, and storage of personal data. Since deep learning models' training usually requires massive amounts of data that may contain PII, following these regulations is essential. Organizations must also have the permission of the individuals for the data collection, enable the data subjects to have an opportunity to access and erase their information, as well as ensure the protection of their data. [46]

Figure 2.2: Privacy Laws Around the World. [51]

2.5 Privacy Attacks in Deep Learning

The threats of data privacy in deep learning include various types of privacy attacks that are malicious efforts aimed at compromising the confidentiality and integrity of sensitive data used in training models.

The most essential attacks are the following ones:

2.5.1 Reconstruction Attacks

- **Definition**
  A privacy threat in deep learning where an attacker attempts to partially recreate a private dataset by analyzing publicly available aggregated data. [52]
- **Mechanism**
  Here's a general breakdown of the reconstruction attack mechanism : [53] [54]

1. **Target the Private Dataset:** The attacker chooses an unauthorized private dataset that he cannot directly access. This data normally includes the income, zip codes, health records, or browsing habits information that are normally considered sensitive.
2. **Gather Public Statistics:** The attacker searches for publicly available statistics derived from the private dataset. These statistics are not individual data points, but

rather summaries of the data. Are often released by government agencies, research institutions, or the organization holding the private dataset (but in a way that doesn't reveal individual information).

3. **Exploit Weaknesses in the Statistics:** The attacker analyzes the type and granularity of the public statistics available and looks for weaknesses in how these statistics are presented.
4. **Data Reconstruction Attempt:** by using the gathered statistics, the attacker attempts to reconstruct entries or characteristics of individual data points in the private dataset.

Figure 3 shows the process of the reconstruction attacks:

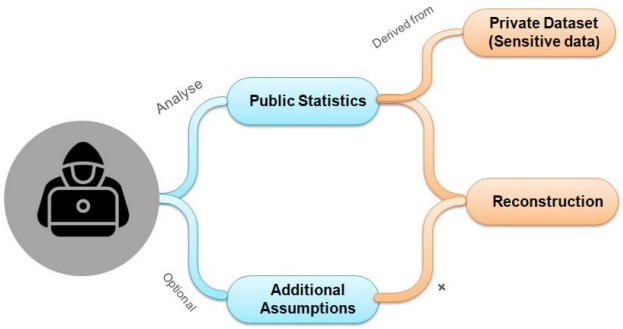


Figure 2.3: The reconstruction attack.

2.5.2 Model Inversion Attacks

- **Definition**

  A form of privacy violation in machine learning where the goal of the attacker is to develop a way to obtain the original training data or specific information that is used in the model's output. In simpler terms, the attacker tries to reverse-engineer the model to "see" the data it was trained on (Figure 4). [55] [56]

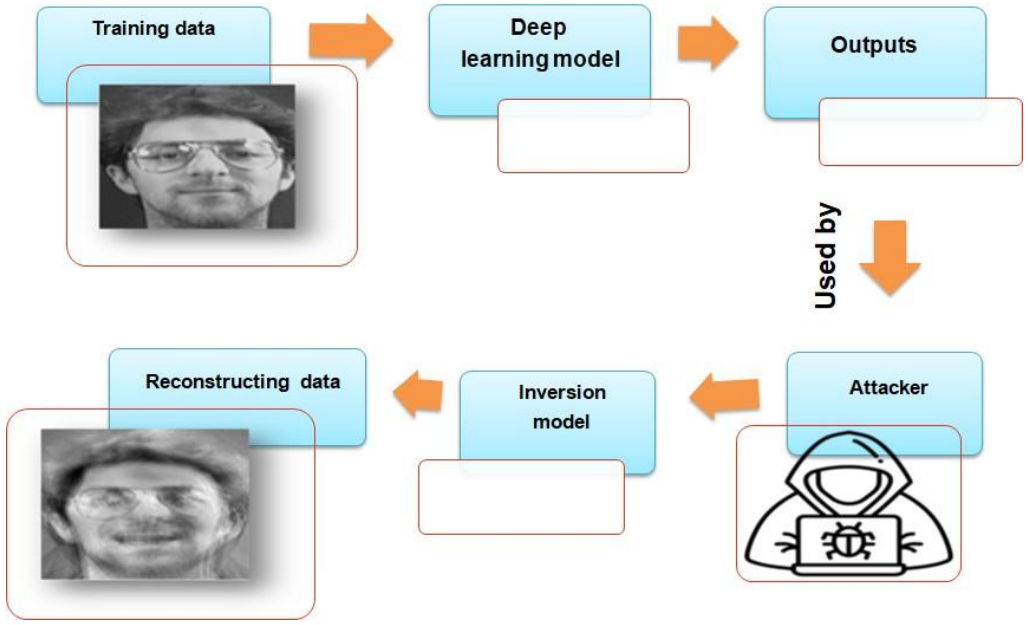


Figure 2.4: The model inversion attack. [57]

- **Mechanism:**
  Here is a simplified breakdown of the model inversion attack process :

  Target Model: The attacker has capabilities of accessing to a target deep learning model. This model is commonly trained with a dataset of the sensitive informationthe attacker intends to gain (e.g., a facial recognition system). [55] [56]

  Input Data: The attacker need some input data to pass on to the target model. These input data could be data that is accessible to the public, or could be data specially designed to influence the decision made by the model. [55] [56]

  Model Outputs: The attacker observes the outputs produced by the given target model when tested on the entered data by him. [55] [56]

  Inversion Model: The attacker creates a separate inversion model. This inversion model is actually another deep learning model trained to study the target model's outputs and, from these, try to deduce the initial input data used to produce these outputs. [55] [56]

  Information Recovery: The attacker tries to deduce what the input data (whichmay contain sensitive information) was by feeding the inversion model by the outputs obtained from the target model. [55] [56]

2.5.3 Membership Inference Attacks

- **Definition** [48] [58]
  A Membership Inference Attack (MIA) is a type of privacy risk in deep learning where an attacker strives to decide whether a data record was part of training data of a created deep learning model. In simpler terms, the attacker wants to know whether a particular sensitive information about an individual has been used in training the model (Figure 1).

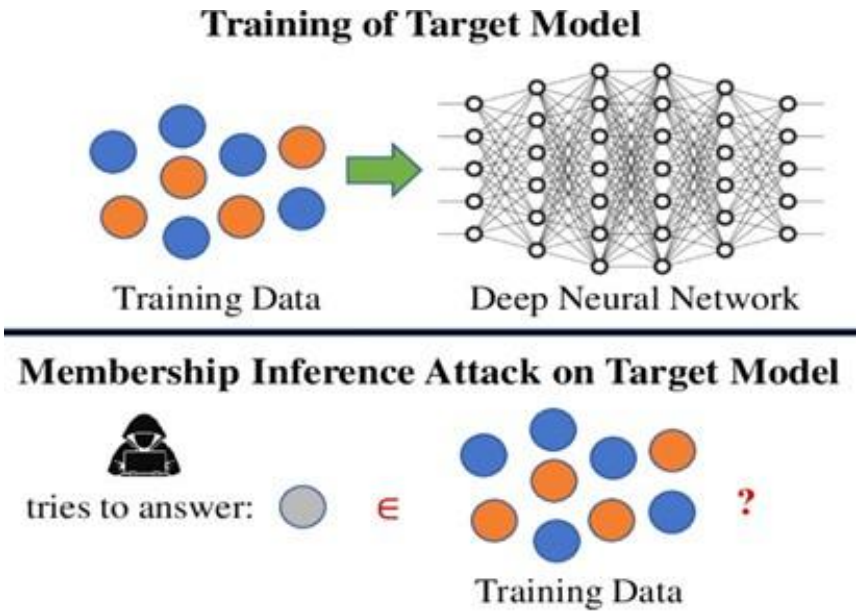


Figure 2.5: The membership inference attack (MIA). [58]

- **mechanism**
  This type of attacks in general involves the following stages:

  1. **Access to the Model:** The attacker gains black-box access to the target model. This means that they can input data into the model, and get results out, but they cannot modify the model, nor do they have direct access to the data it was trained on. [48] [58]
  2. **Crafting Inputs:** the attacker randomly generates a new set of data points that they are close to the training ones. [48] [58]
  3. **Observing Model Behavior: Observing Model Behavior:** An attacker provides these crafted inputs to the target model and analyzes the outputs given by the model which can be predictions or classifications. For instance, slight variations in the model's behavior to these inputs as compared to other normal inputs might suggest membership. [48] [58]
  4. **Membership Inference Model:** The attacker builds another model known as a membership inference model which feeds through the output of the target model and attempts to determine whether an input data point was training data. [48] [58]

2.6 Privacy-Preserving Techniques in Deep Learning

As highlighted earlier, deep learning is not without its drawbacks, in particular, privacy or as it's commonly called, privacy attacks, which pose a real threat to individuals' personal data. Hence, preserving data privacy becomes critical in order to protect these information from unauthorized access and misuse. [59]

Initially, as a primary solution, the so-called anonymization method was implemented.

2.6.1 The anonymization method

It's the procedure of modifying data by removing or masking Personally Identifiable Information (PII) in a manner that would not let people understand who the information belongs to within the context of a data set. Anonymization methods in deep learning involve techniques (randomization, generalization, suppression, and tokenization) That are useful for preserving individual privacy while enabling organizations to use important data for deep learning model training and analysis. [60]

Techniques of anonymization:

- **k-Anonymity**: A dataset is k-anonymous if each record cannot be distinguished from at least (k-1) other records with respect to some identifying attribute. It provides basic privacy protection, but does not provide protection against disclosure of attributes. [61][62]
- **l-Diversity:** This technique extends k-anonymity, which is a concept that offers better privacy protection. It requires that we should have at least 'l' distinguishable values in the sensitive attributes within each group of indistinguishable records. This further minimizes attribute disclosure since the chances of identifying the sensitive attribute are diminished. [61][62]
- **t-Closeness:** This builds upon l-diversity where it is necessary that the distribution of the lattice over the sensitive attribute values in any group should be fairly close to that over the entire dataset lattice. This ensures that there is no high density of attribute values in a specific group to enhance the data privacy. [61][62]

This figure provides an easy-to-understand explanation of these mechanisms:

Figure 2.6: Anonymization techniques.

Advantages of anonymization

- **Protection of privacy :** All three techniques ensure that the privacy of individuals is protected by making it difficult to identify specific individuals from a dataset. [63]
- **information Utility:** These allow the publication of data set statistics while maintaining

privacy. This is especially useful for information sharing and collaboration. [63]

Limits of anonymization

- **Loss of Information:** These technologies requires data to be aggregated or scrambled to ensure privacy, which can result in a loss of detail and accuracy of disclosed data. [63]
- **Vulnerability to Linkage Attack:** If the anonymized data is combined with other datasets containing the same individuals , it might be possible to re-identify them. [63]

  And the most common example about that is when Netflix conducted a data anonymization of the training set for the recommender challenge by making name values in the database to be random identity numbers. What is even more shocking is that by connecting it to public IMDB review ratings, people have been de-anonymized once again by researchers. [63]

  So if we intend to achieve complete privacy, we need to know nothing about the data. For that , there are other technologies that achieve this, which are the followings ones :

2.6.2 Federated Learning

Federated Learning (FL) is a distributed machine learning approach where multiple devices or servers collaboratively train a model under the administration of a central server but without necessarily sharing the underlying data itself [64]. Which is mean that Instead of sending raw data to that server, each device shares model updates computed locally (e.g., gradients) with that server that it aggregates them to improve the global model . [65]
Figure 7 shows the process of federated learning:

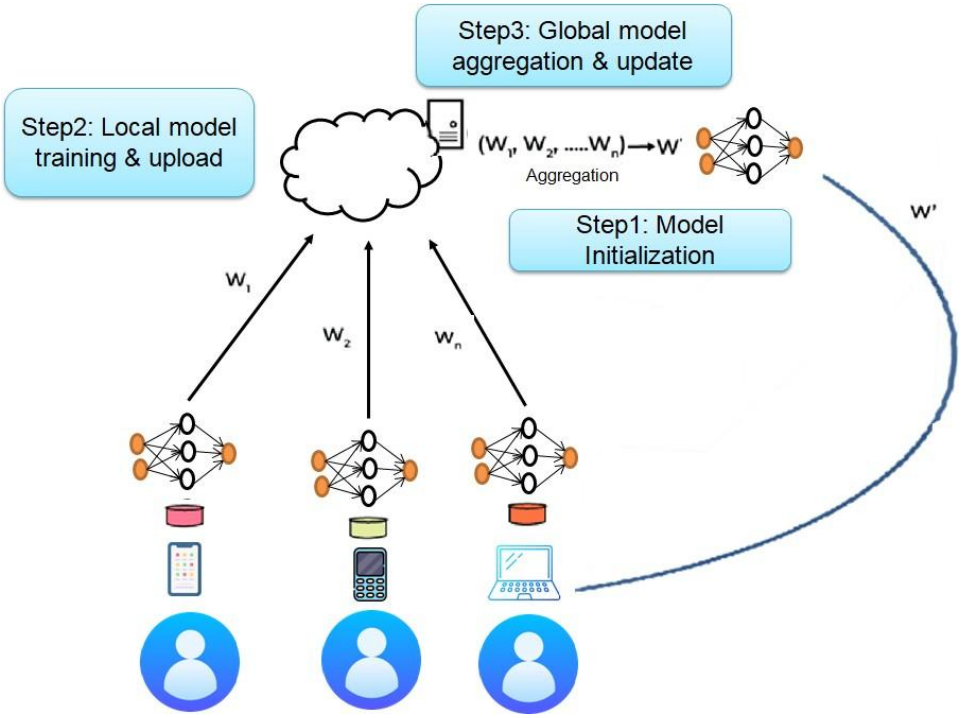


Figure 2.7: The Federated Learning Process. [66]

Advantages

- **Reduced Data Transfer**: As a result of sharing only model updates as opposed to raw data, the data communicated over the network at any one time is much lower. This result

in low communication expenses and optimal bandwidth utilization. [65]

- **Scalability** : Federated learning can scale to a large number of clients, making it scalable to vast networks like smartphones or Internet of Things devices . [64]

- **Privacy Preservation** : Federated learning keeps the training data local on client devices, minimizing the risk of data breaches and enhances privacy. [64]

Limits

- **Communication Overhead:** While aggregating less data, federated learning still needs to ensure that the devices and the central server are constantly exchanging model updates, which may be problematic in terms of bandwidth. [65] [66]

- **Aggregation Complexity** : Efficiently aggregating model updates from a large number of devices, while ensuring robustness against unreliable updates, poses significant technical challenges. [66]

- **Heterogeneity**: Client devices in federated learning can have varying computational capabilities, network conditions, and data distributions, leading to challenges in training an effective global model. [65] [66]

2.6.3 Homomorphic Encryption

Homomorphic Encryption (HE) is a kind of encryption where operations can be performed on encrypted data without decrypting it. This means that a client could send data to the server in an encrypted form may perform certain calculations over this data without decrypting it and then send back the encrypted result that the client would then decrypt. [67]

Figure 8 shows the process of Homomorphic Encryption :

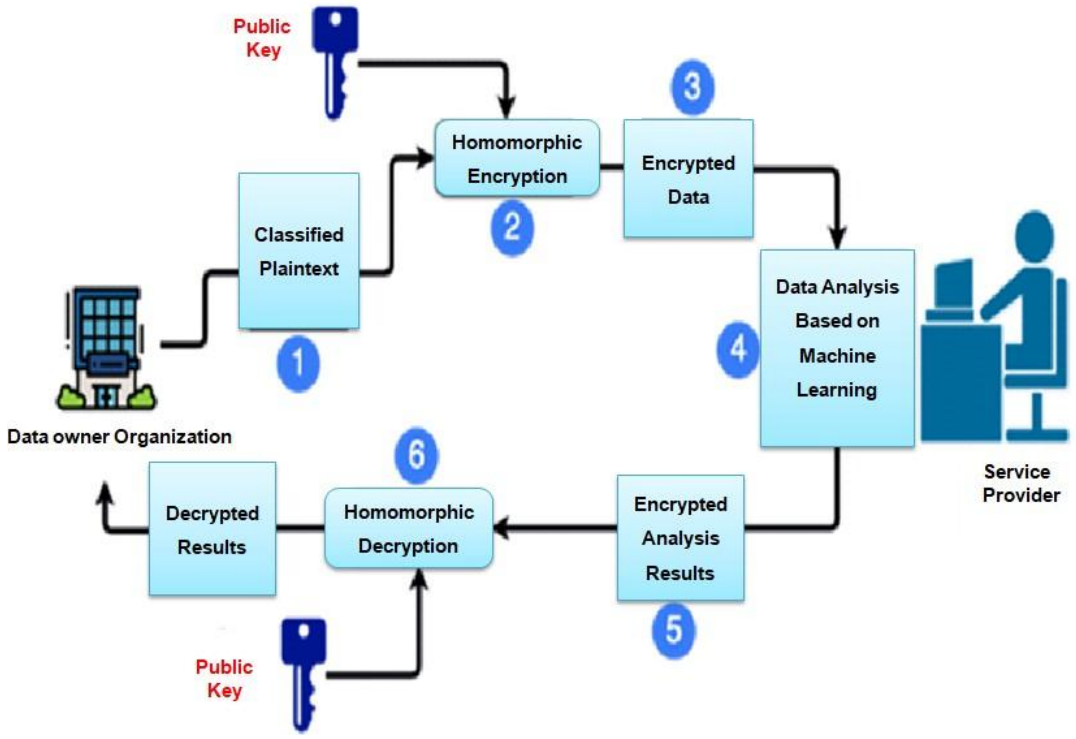


Figure 2.8: Homomorphic processing . [68]

Types of Homomorphic Encryption

- **Fully Homomorphic Encryption:** Fully Homomorphic Encryption (FHE) allows additions and multiplications that are performed on encrypted data with different rounds of execution that are limitless, which it means computations, that are arbitrary can be done on encrypted data. [67]

- **Strong (Somewhat) Homomorphic Encryption:** Strong (Somewhat) Homomorphic Encryption (SHE) allows multiple operations for homomorphism, although the count of such operations cannot be counted into infinity[19]. an example of SHE , the BGN cryptosystem (Boneh-Goh-Nissim) which supports both addition and multiplication but is limited to a certain number of operations. [68]

- **Partially Homomorphic Encryption:** Partially Homomorphic Encryption (PHE) enables addition or multiplication on cipher texts but not both .This limitation limits the nature of computations that can be made to a system compared to Fully Homomorphic Encryption , although the unlimited executions . [67]

Advantages of HE

- **Outsourcing:** HE allows computations to be safely outsourced to untrusted servers , elimination the need for sensitive information to be stored locally. [67]

- **Flexibility:** HE can be used to perform numerous calculations on the encrypted data, scenarios such as the operations of neural networks or more generally signal processing. [69]

- **Secure Computation:** HE assists in performing computations on the data without the data being exposed to unauthorized persons in a plaintext format, therefore maintaining security of the data during computation and transmission. [67]

Limits of HE

- **Latency**: It can also cause high latency due to the need to encrypt data before transmission and then decrypt it at the receiving end, so it can be a challenge for real-time applications. [69]

- **Key Management :** Ensuring that keys are managed properly is important for protecting Homomorphic Encryption systems, which add complexity to implementation and maintenance . [67]

- **Computational Intensity:** HE is computationally heavy thus taking a lot of time to process as compared to one which is not encrypted . [67]

2.6.4 Secure Multi-Party Computation

Secure Multi-Party Computation (SMPC) is an efficient cryptographic protocol which allows the structured multiple parties to have a joint computation on the data own but the data is not disclosed to the other parties. This is done using secret sharing and homomorphic encryption to ensure that no party gets to see the actual data but a copy that has been encrypted and the outcome of the computation. [70]

In another way ,SMPC is inspired from secret sharing technique, wherever each participant splits his input randomly into parts and sends it to other participants. Subsequently, the parties respectively take specific actions to perform some arithmetic computation on their actions such as addition, multiplication or comparison, employing respective protocols that maintain the secrecy of the actions. Last but not at least, there is synchronization of the actions of the parties to get the result of the function, but without any information about the inputs. [71]

Figure 9 shows the process of secure multi-party computation:

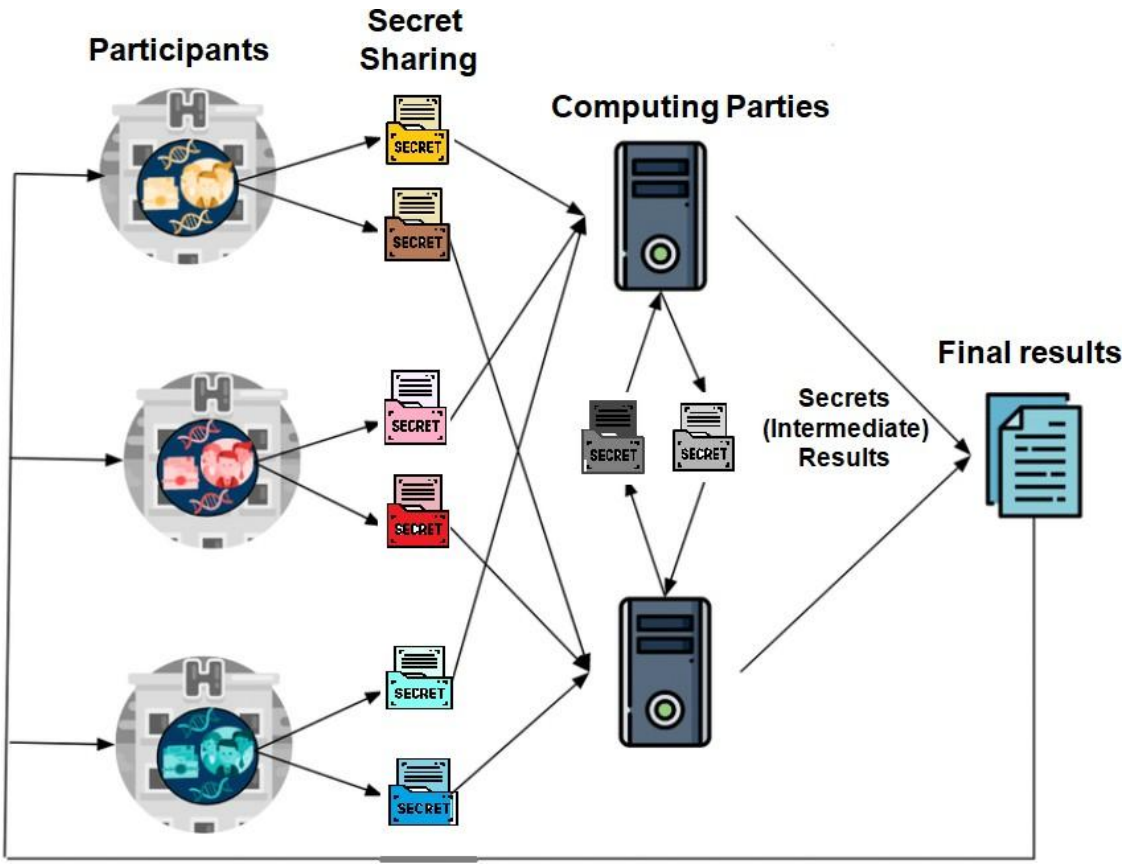


Figure 2.9: Secure Multi-Party Computation processing [72]

advantages of SMPC

- **Distributed Trust:** SMPC is a beneficial approach because members do not need to trust each other completely to work cooperatively and perform computation tasks for decentralized applications.

- **Flexibility:** SMPC has some versatility in the sense that it can accommodate a broad class of computations, from basic and more complex ones.

- **Privacy Preservation:** The most important consideration in SMC is the privacy preservation that guarantees that input of any party or subgroup remains unrevealed even when some of them are malicious or have agreed to leak the information. [71]

limits of SMPC

- **Rust Requirements:** The SMC approach relies on the assumption that at least one party can still be treated as being trustworthy in keeping the computation integrity and can be a disadvantage in a situation where all the parties can be untrusted. [71]

- **Communication Overhead:** Due to the fact that Secure Multi-party Computation entails massive communication among the parties it an be a problem in distributed computing

environment.

- **Computational Complexity:** SMC may be time consuming particularly for those computations which require computational power and this may lead to either slowness or expensive computation. [71]

2.6.5 Differential Privacy

Differential privacy (DP) is a mathematical framework for preserving the privacy of individuals in a dataset allowing for meaningful analysis. The basic idea is to add random noise to the data or survey results to mask individual contributions. This makes sure that the inclusion or non inclusion of any singl data point does not significantly affect the performance of the analysis. [72]

In other word, or according to the formal definition, a randomized mechanism M applied to a data set D satisfies differential privacy if and only if it holds for every subset of outputs S and two adjacent inputs d and d' : [72]

$$\Pr[M(D_1) \in S] \leq e^{\epsilon} \Pr[M(D_2) \in S]$$

Where:

- ϵ is the privacy budget that sets the amount of privacy.
- e^{ϵ} provides for a tiny chance that the privacy guarantee will be violated.

Advantages of differential privacy

- **Flexibility :** The differential privacy can be used for various kinds of data and for various kinds of queries, which makes it a flexible approach to the protection of data privacy in the different cases. [72]

- **Post-processing Invariance:** it means any transformation of a differentially private output can be an arbitrary transformation and still remain private. This implies that the result of a differentially private algorithm can go through post processing and transformation and all of these will not affect any person's privacy. [73]

- **Strong Privacy Guarantees:** protects data by introducing carefully controlled noise which mathematically caps the likelihood that individuals in the dataset can be uniquely reconstructed. [72]

Limits of differential privacy

- **Complexity :** It can be challenging to get right and indeed often calls for the added computation of days or even weeks to correctly set the right noise for the correct level of privacy and to adequately understand the data and the queries involved. [74]

- **Cumulative Privacy Loss :** The privacy loss adds up when answering multiple queries using differential privacy, thus an efficient management is necessited to preserve the overall privacy.[74]

- **Utility-Privacy Trade-off:**There is an inherent trade-off between data utility and privacy. Higher privacy guarantees often result in more noise and lower data utility. [74]

2.7 Case Studies and Real-World Applications

- Practical examples of privacy-preserving deep learning.

- Analysis of successful implementations and challenges faced.

On the other hand, privacy-preserving deep learning (PPDL) mainly concern itself with the most efficient way that deep learning models could be trained without compromising personal information. Here are some practical examples and an analysis of their implementations and challenges. In the following section, there are some examples of the application of the social media marketing and the assessment of each implementation and difficulty.

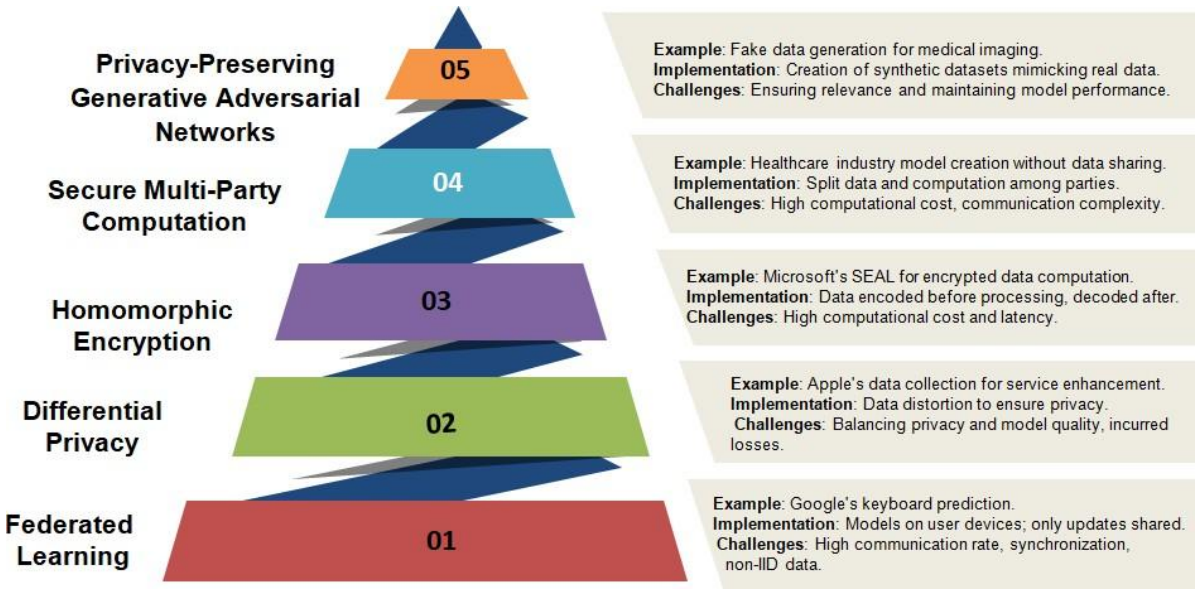


Figure 2.10: Case Studies and Real-World Applications.

2.7.1 Federated Learning:

- **Example**: Federated Learning (FL) as a solution by Google to improve the prediction of keys on keyboards when typing without forwarding the information of users to main servers.

- **Implementation:** Models created are at user end and only update models are passed between training partners.

- **Challenges:** The challenges that come with a high model communication rate, the problem of synchronizing a model, and handling non-IID data across the devices.

2.7.2 Differential Privacy:

- **Example**: In this case, Differential Privacy (DP) was employed in collection of necessary data by Apple in the enhancement of services delivery without infringing on the users' rights to personal privacy.

- **Implementation:** Sometimes it is necessary to ensure that data is distorted in a manner that no specific data point belongs to another in terms of a probability density function.

- **Challenges:** Sacrifices in loss incurred across a set of models, and the balancing of users' privacy and high-quality models.

2.7.3 Homomorphic Encryption:

- **Example**: An example of a software tool is Microsoft's SEAL that provides ability to compute on encrypted data using a library.

- **Implementation:** Data outsourced pre-processes through encoding and then decoded after feeding it to the model and receiving the result.

- **Challenges:** High computational cost and high latency since the operations that are carried out on the cipher text data are time-consuming and complex.

2.7.4 Secure Multi-Party Computation:

Secure Multi-Party Computation (SMPC) refers to the process by which a number of parties cooperate in computing a function on their private inputs without the need to exchange the values of the inputs.

- **Example**: Privacy-preserving methods for creating the model with the help of several parties in the healthcare industry while the parties never share their data with each other but develop the model together.

- **Implementation:** It has been pointed out that one party has only half of the data set while the other party carries a half of the computation on the data set and is not aware of it.

- **Challenges:** In this approach comprehensiveness issues, high computational cost and thus high communication complexity is another drawback is this approach.

2.7.5 Privacy-Preserving Generative Adversarial Networks:

- **Example**: In the first place designed to develop data which is very similar to real data but does not depict actual people's data. For instance, GANs will support the medical imaging in which one can develop fake patient data for analysis.

- **Implementation:** The outcome of a GAN is a fake data set identical in all the attributes of the data set in question.

- **Challenges:** This process enables one to check whether the generated data is relevant and maintain the model working.

- Nevertheless, the above-said techniques have put light on protection of privacy during deep learning but with constraints like computational load, communication overhead, and the flip side of sacrificing privacy for a better model.

CHAPTER

3

DIFFERENTIAL PRIVACY IN DEEP LEARNING

3.1 Introduction

As we discussed earlier, privacy of data is important since it helps guard data that is crucial to individuals from being accessed and used by unauthorized persons. In the context of deep learning, this involves ensuring that data used for training models does not inadvertently expose private information. In that regard, several methods have been considered to help enhance data privacy including anonymization, federated learning, homomorphic encryption, secure multi-party computation, and differential privacy. These methods are meant to ensure information security or some of it cannot be accessed or even identified by anyone else [63]. This chapter will expand further on a technique that has emerged in the recent past and has elicited a lot of interest, which is the differential privacy, a mathematical theory that has strong privacy provisions in the sense that the privacy of an individual cannot be violated given that the inclusion or exclusion of an individual data point does not have a material impact on the analysis being conducted.

3.2 Historical Background of Differential Privacy

The concept that became known as differential privacy (DP) has origins in the 1970s when Tore Dalenius distillated the cell suppression mathematics [67]. This work emphasized the theme of non-disclosure of information or anything about a person that one might come across in their day-to-day lives. It emerged from a paper by Dorothy Denning, Peter J Denning and Mayer D Schwartz in 1979[67] in which a concept of Tracker an adversary, who could create queries to read all confidential contents, was defined. This led to the understanding that even privacy properties in a database cannot remain intact without necessarily taking into account previous queries as a query is posed, and this makes query privacy NP-hard [67]. We can cite the breakthroughs ofthe century at the turn of the 2000s. In 2003, Kobbi Nissim and Irit Dinur identified that, it is actually impossible for one to freely publish questions on a statistical database which refersto the database owner's private information while going further to show that by using random

questions, one can reveal the entire content of the database[3]. This fundamental limitation in turn gave rise to DP as semantics, a definition of privacy that is meaningful and mathematically well-founded. The necessity of DP in present day data analysis has surged in recent years with a heightened need for protection of privacy. As it satisfy the definition of privacy and algorithms associated with this definition that appeared as electronic data about individuals becomes increasingly detailed, and technology enables powerful collection and curation of these data.

3.3 Definition of Differential Privacy

Differential privacy is a powerful mathematical framework that provides a rigorous and quantifiable way to protect the privacy of individuals in statistical databases. The key idea is to ensure that the presence or absence of any single individual's data in the database does not significantly affect the outcome of any analysis performed on the data. This is achieved by carefully adding noise to the data or the analysis results, in a way that preserves the overall statistical patterns while obscuring individual-level details. [77]

The formal definition of ϵ -differential privacy states that for any two adjacent databases (i.e., databases that differ in at most one record), the probability of any given output from the analysis algorithm must be almost the same, regardless of whether a particular individual's data is included or not. The parameter ϵ controls the level of privacy protection, with smaller values of ϵ indicating stronger privacy guarantees. [78]

Differential privacy offers several key advantages over traditional privacy-preserving techniques. It provides strong, provable privacy protections that hold even in the face of powerful adversaries with access to auxiliary information. It also enables the release of useful aggregate statistics about the data while rigorously bounding the privacy loss. This makes differential privacy a valuable tool for a wide range of applications, from census data publication to private machine learning. [79]

The practical implementation of differential privacy involves the use of specialized noise-adding mechanisms, such as the Laplace mechanism or the Gaussian mechanism. These mechanisms carefully calibrate the amount of noise added based on the sensitivity of the analysis task, ensuring that the desired level of privacy is achieved without excessively degrading the utility of the data. [80]

In summary, differential privacy is a transformative concept that has the potential to revolutionize the way we think about and protect individual privacy in the era of big data and data-driven decision making. By providing a mathematically rigorous and provable privacy guarantee, differential privacy offers a principled approach to balancing the competing demands of data utility and individual privacy. [81]

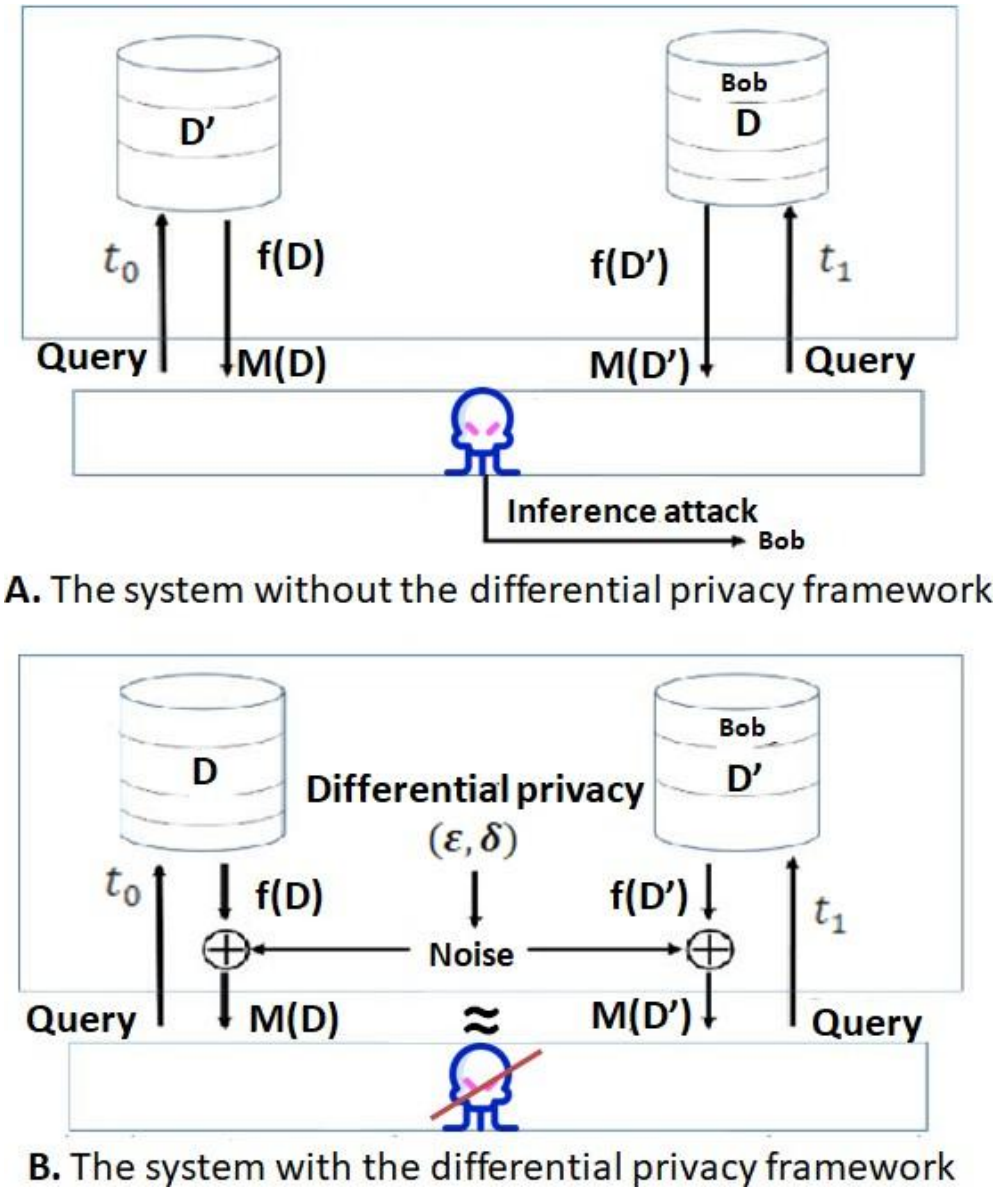


Figure 3.1: Overview of the differential privacy framework. [82]

3.4 Why DP is chosen over other privacy-preserving techniques

DP provides rigorous and long-lasting protective measures owing to the mathematical nature of the concept. It also preserves the nature of privacy guarantees as strong and measurable that does not depend on the adversary's knowledge and computational ability. Therefore, the privacy guarantees given by DP hold good in cases of multiple processing sequences of data and even after post-processing.

3.4.1 Mathematically Guaranteed Privacy and Post-Processing

This robustness makes certain that the privacy protections afforded by DP mechanisms are preserved even in instances where an adversary has more external knowledge about the dataset.

The fundamental idea is that noise injected through DP prevents any individual record from being extracted from the data set by any knowledgeable adversary .

3.4.2 Robustness against auxiliary information

This robustness makes certain that the privacy protections afforded by DP mechanisms are preserved even in instances where an adversary has more external knowledge about the dataset. The fundamental idea is that noise injected through DP prevents any individual record from being extracted from the data set by any knowledgeable adversary.

3.4.3 Composability

DP composability is a feature that means that it is possible to get the overall result keeping privacy in mind when using multiple DP mechanisms in your calculations. In the contexts of applied DP algorithms, either serial or parallel, the overall privacy loss is therefore equal to the total of the individual privacy losses of each applied mechanism. This guarantees that with every final result the sum total of contribution to the privacy risk being inflicted on an individual will not overshoot a predetermined limit no matter the number of private computations made. For DP to be practical in systems that consist of a number of components processing the sensitive data repeatedly in various stages of their analysis or computation, it has to be compostable.

3.5 Types of Differential Privacy (Global & Local)

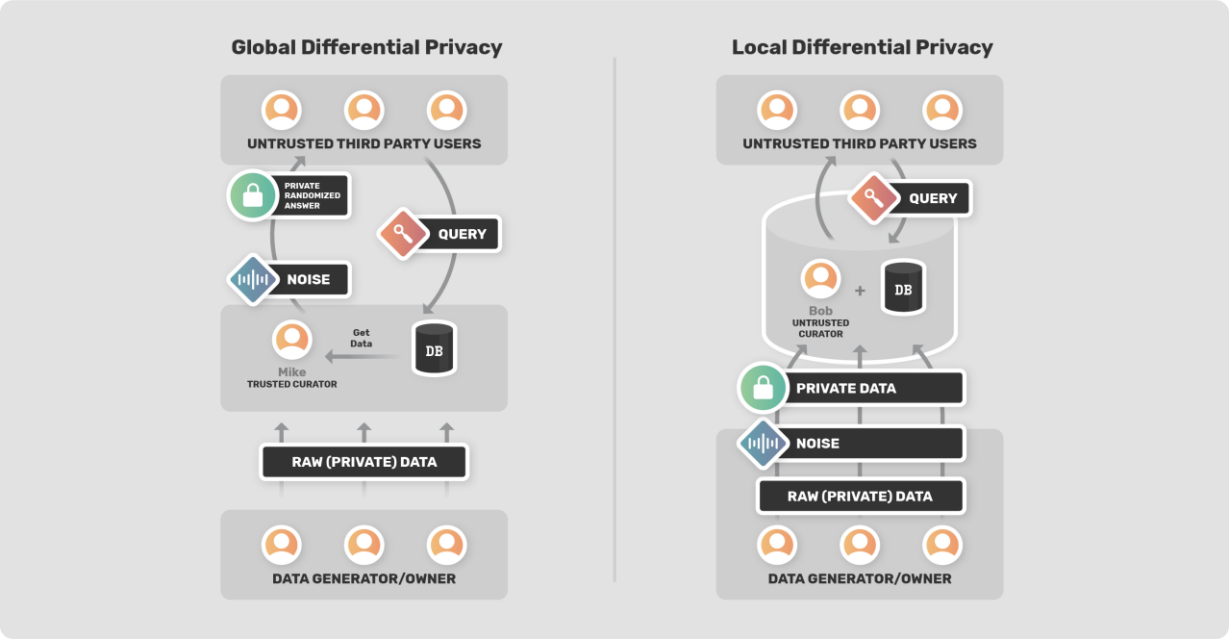


Figure 3.2: Local Differential Privacy and Global Differential Privacy. [83]

Differential Privacy describes a promise, made by a data holder, or curator, to a data subject (owner), and the promise is like this: “You will not suffer any adverse or other consequences as a result of your data being used in any study or analyses; however other studies, datasets and or information sources are available or used.” To sum it up, differential privacy guarantees that an attacker cannot influence a specific participant in the database query with overwhelming probability, even if the attacker has unbound computational power and has seen all the others participants of the database except for the specific targeted participant’s data.

DP functions in a way that introduces noise in the form of statistics to the data (or it can be added in the input data or even in the output data). Depending on the location, where the noise is introduced in the execution of the DP algorithm, it is mainly divided into two categories; Local DP and Global DP. [83]

3.5.1 Global Differential Privacy

In the global differential privacy approach, the process works as follows: In the global differential privacy approach, the process works as follows: [83]

Raw (Private) Data

The accumulation of huge volumes of data, together with algorithm development and learning operations, requires the data generator/owner to be in control of the raw and sensitive Private data they wish to share with untrusted third-Party Users in a privacy-preserving manner. [83]

Trusted Curator (Mike)

The data generator/owner can trust Mike, who will be the curator of the raw private data and will analyze them with the necessary differential privacy techniques. [83]

Differential Privacy Techniques

Using the data gathered and synchronizing with the trusted curator Mike, differential privacy techniques are applied to the raw data. It also entails creating noise (NOISE) to the data in a controlled manner, where some amount of uncertainty is affixed making it almost impossible to distinguish the data values. [83]

Query Interface

The trusted curator, Mike, then forms a query interface that gives the untrusted third-party users a means of accessing the data from the privacy-preserved views. This interface also allows the users to submit the queries, abbreviated as QUERY. [83]

Private Randomized Answer

Private randomized answer when an untrusted third-party user has asked the question, the trusted curator, Mike, completes the query and produces an anonymous answer. The noise is included in this answer; this makes it difficult for a specific data point to be clearly distinguished. [83]

Untrusted Third-Party Users

Concerning the work-flow, the untrusted third-party users have an ability of accessing the privacy-preserved data via the query interface implemented by the trusted curator, Mike. In this context, individuals submitting queries and get the authorized and convincible private and randomized answers without premeditating the privacies of the original data. [83]

- **The global differential privacy advantages**

  The conclusion is that the data generator/owner benefits from using a trusted curator to process the raw data and apply the necessary privacy-preserving data analysis techniques. In this way, while the data are made available to the untrusted third-party users, their access is constrained and can be done anonymously. [83]

3.6 Variants of DP

3.6.1 Epsilon Differential Privacy (ϵ-DP)

Epsilon-Differential Privacy or as it's known Pure Differential Privacy , is a specific mathematical abstraction of privacy that offers exact measures on the protection level provided.
Specifically, a randomized algorithm M provides -DP for any two datasets D and D' that differ by at most one element, and for any set of outputs S :

$$P(M(D) \in S) \leq e^{\epsilon} \cdot P(M(D') \in S)$$

(ϵ-DP) Advantages:

Simplicity: ϵ-DP is straightforward to understand and implement where it provides a clear privacy loss parameter ϵ that allows for easy quantification and communication of the level of privacy protection . [84]

Strong Privacy Guarantees: ϵ-DP offers a stringent mathematical privacy guarantee with no failure probability, ensuring strong privacy protection. [84]

(ϵ-DP) Limitations:

Overly Conservative: The strict privacy guarantees of ϵ-DP can lead to excessive noise addition, reducing the utility of the data. It might be too restrictive for practical applications, where some trade-off between privacy and data utility is necessary. [84]

No Tolerance for Failures: ϵ-DP does not account for any probability of failure, making it impractical in real-world scenarios where a small margin of error (ϵ) might be acceptable . [84]

Difficulty in Setting : Choosing an appropriate value for ϵ is challenging, as too small reduces data utility, while too large ϵ weakens privacy guarantees. [84]

3.6.2 Epsilon-Delta Differential Privacy (ϵ,δ-DP)

Approximate Differential Privacy or epsilon-delta differential privacy (ϵ,δ-DP) is an upgrade of ϵ-DP by allowing a small probability δ of privacy failure.
A randomized algorithm M provides (ϵ,δ)-DP if, for any two datasets D and D that differ by at most one element, and for any set of outputs S:

$$P(M(D) \in S) \leq e^{\epsilon} \cdot P(M(D') \in S) + \delta$$

- **(ϵ,δ)-DP Advantages:**

  Flexibility: (ϵ,δ)-DP is more practical in formulating the privacy by introducing a small probability δ, in which the privacy guarantee can be breached while solving the practical privacy issues encountered in real-life applications. [84]

  Balanced Trade-Off: It has always been true that by allowing failure to occur at some minimal level, As it is illustrated, the identified (ϵ,δ)-DP model provides an optimal middle ground between asserting high levels of privacy and achieving valuable data utility. This trade-off is critical in order to keep the representativeness of the data and maximize the level of privacy [84].

Scalability: The complexity of the system, aggregate calculation, or extensive data exchanges can make (ϵ,δ)-DP more scalable. The featured flexibility of the method with regards to adjusting individual components contributes to better management of the total privacy loss and thus proving more feasible for use at a large scale[84].

- **(ϵ,δ)-DP Limitations:**

  Complexity: δ makes the implementation and understanding of privacy guarantees more complicated. This can complicate the process of expressing and enforcing the privacy promises when compared to the basic ϵ-DP model. [84]

  Potential Misinterpretation: This flexibility may however be the reason why (ϵ,δ)-DP lacks accuracy in the formulation of the provided privacy guarantee. Some users might fail to appreciate the protection offered by ϵ while at the same time not paying attention to the role of δ. [84]

  Setting Parameters: It is difficult to identify the right values for ϵ and δ. If δ is too large, it means the privacy guarantees are compromised, while if δ is too small then it results in adding too much noise, like the problem with ϵ-DP. [84]

3.6.3 F-Differential Privacy (F-DP)

f-Differential Privacy (f-DP) extends the privacy loss function in order to offer a more versatile form of protection. Unlike ϵ or (ϵ,δ) that are fixed parameters of the privacy budget, f-DP uses a function *f* to quantify the distance of the probability distributions of the outputs from two consecutive datasets. This enables application and data distributions to be accounted for when it comes to the privacy guarantees to be offered .

F-DP Advantages

- **More Versatile Privacy Guarantees:** Compared to the standard DP, f-DP is more flexible and can offer better guarantees for privacy by considering the distribution of the application and data. This can be especially helpful in cases when basic ϵ-DP might not be enough unless of course, ϵ is low enough to allow for accurate approximations of the true data. [85]

- **Improved Accuracy:**

  This is because the privacy guarantees can be strengthened by a function *f* to measure the distance between two probability distributions and can offer better privacy than what ϵ-DP does. [86]

- **Enhanced Flexibility:**

  f-DP makes it possible for the application of different functions *f* that measure the distance between probability distributions, leading to effective enhancements in privacy preserving mechanisms. [87]

F-DP Limitations

- **Increased Complexity:** ϵ-DP is quite straightforward, thus making f-DP a more general type of DP which uses a function f to measure the distance between probability distributions. This is due to the fact that it becomes more complex to analyse and implement f-DP mechanisms when the complexity of the data increases. [85]

- **Higher Computational Requirements**

  While f-DP is generalisable to more complex tasks than ϵ-DP, it usually demands more computational resources than the latter, particularly when f is complicated or data is massive. [86]

- **Potential for Overfitting:**

  This is simply because the use of a function *f* to measure the distance between two probability distributions may at times result to overfitting if the chosen function is not regulated or when the data used to train the model does not represent a true distribution of the actual data set. [87]

- **Interpretability Challenges:**

  fϵ-DP is generally easier to interpret than f-DP since the later requires the employment of a function *f* for measuring the proximity of probability distributions. This can be problematic when reasoning about the privacy guarantees offered by f-DP mechanisms.[88]

- **Potential for Bias:**

  Therefore, the risk of bias arises if the function f is not chosen appropriately or if the data are not sampled from the distribution P or if neither f nor P are chosen appropriately.

  All things considered, f-DP provides more flexible privacy guarantees than conventional ϵ-DP, but it also has more complexity, greater processing overhead, and possible interpretability and bias issues. [89]

3.6.4 Renyi Differential Privacy (RDP):

RDP employs the Renyi divergence as a metric for the privacy loss, which is used to estimate the divergence of probability distributions. The Renyi divergence is defined as:

$$D_\alpha(P \parallel Q) = \frac{1}{\alpha - 1} \log \int \left(\frac{dQ}{dP} \right)^\alpha dQ$$

RDP Definition

Given a differentially private mechanism M and a dataset D, the RDP of M is defined as:

$$\rho_\alpha(M, D) = \sup_{D', D \sim D'} \frac{1}{\alpha - 1} \log \int \left(\frac{dM(D)}{dM(D')} \right)^\alpha dM(D')$$

$$\rho_\alpha(M, D) = \sup_{D', D \sim D'} \frac{1}{\alpha - 1} \log \int \left(\frac{dM(D')}{dM(D)} \right)^\alpha dM(D')$$

Where D' is an adjacent dataset to D, and M(D) and M(D') are the probability districutions of the output of M on D and D', respectively.
Less Pessimistic: RDP may be even more relaxing than ϵ-DP, meaning that it might offer more privacy protection for specific scenarios.
Simpler Accounting: However, it is still easier to apply RDP for analyzing the privacy loss in the complicated mechanisms compared to ϵ-DP with α.

[90] [91] [92]

RDP Limitations:

- **Computational Complexity:** It is to be noted that RDP can be quite computationally intensive especially when the underlying data is sizeable or when the mechanism under consideration is elaborate. This is usually the case especially when working with large data sets as it becomes difficult to establish practical RDP mechanisms and then analyze the results that are obtained. [91] [92]

- **Interpretability Challenges:** RDP can be slightly difficult to analyze compared to the standard ϵ-DP since it employs a more complicated privacy leakage metric. This could complicate the task of evaluation of privacy assurances given by RDP mechanisms. [91] [92]

- **Parameter Selection:** The selection of the parameter α can influence the privacy measures provided for RDP. It is not always easy to determine the best value for α, and a wrong choice may lead to the compromise of a subject's privacy. [91] [92]

- **Data Distribution Sensitivity:** RDP is only sensitive to the distribution of the data. Sometimes the data distribution cannot be described well or is complicated and in these situations it becomes difficult to accurately estimate the privacy loss in terms of RDP. [91] [92]

- **Mechanism Complexity:** RDP may be more difficult to employ for the analysis and implementation of particular systems, especially those that include a variety of interactions or nonlinear conversions. It can make the actual maintenance of the privacy guarantees of RDP mechanisms more complicated. [91] [92]

- **Limited Support:** RDP is only applicable to those data sets where the sample sizes are the same for both classification variables. There is a potential problem with RDP if the datasets have different supports, then RDP will not be able to accurately measure the privacy loss. [91] [92]

- **Lack of Standardization:** Despite the numerous research works that have been done, RDP is still in its development phase and there is no well-established definition of RDP nor is there a uniform way through which it is practiced. Presumably, this can cause some difficulties when it comes to comparison and integration of various mechanisms of RDP. [91] [92]

- **Limited Theoretical Understanding:** RDP has been proved to offer guaranteed level of privacy; however, there is a lack of theory to explain its characteristics and performance. This can make it difficult to prove the privary protection of RDP mechanisms in the real world. [91] [92]

Applications

- **Machine Learning:** RDP has been used in machine learning when traditional ϵ-DP mechanics are insufficient, for example, in deep learning or natural language processing.

- **Data Publishing:** RDP has been applied in the data publishing where it offers more accurate level of privacy for statistical disclosures.

- **Privacy-Preserving Data Analysis:** This tool has also been used for privacy-preserving data analysis, where RDP allows for the release of statistical analysis without compromising the identity of specific data values.

3.7 Differential Privacy Mechanisms

3.7.1 Laplace Mechanism

This mechanism injects Laplace noise to the output of a function to realize differential privacy. The noise which is the second term of the right hand side is accrued from the sensitivity of the function and is Laplace distributed. The sensitivity of a function measures the maximum amount that the output can change given the inclusion or exclusion of one person from any conceivable set of inputs. [93] [94]

Advantages:

- Laplace noise is added to the query output to offer good protection to the privacy of the individuals involved.
- It can be easily implemented and statistically analyzed compared to some other models.
- It can however be employed for any type of query functions .

  [97] [98]

Limitations:

- The noise to be added depends on the global sensitivity of a query which can be large for some queries.
- May not be suitable for high sensitivity queries as the amount of noise needed to generate the answer can have a devastating effect on the usefulness of the answer.

  [97] [98]

3.7.2 Gaussian Mechanism

The first of this mechanism involves adding Gaussian noise to the data in order to guarantee differential privacy. In contrast to the Laplace mechanism, the Gaussian mechanism does not guarantee only pure ϵ-differential privacy but rather implements the (ϵ,δ) –differential privacy. [95] [96]

Advantages:

- Actually can offer stronger privacy assurances than Laplacian mechanism for some queries.
- The noise added depends on the L2 sensitivity of the query where this value can be lower than the Laplacian sensitivity in L1.
- It can be used for queries that are likely to be sensitive in nature because it involves natural language processing.

  [97] [98]

Limitations:

- This is an area that needs to be approached with some caution, so as to guarantee that the privacy that one wants to be provided is the one that is actually being offered.
- It might be less flexible than the Laplacian mechanism, especially since it is only applicable to queries with L2 sensitivity that does not exceed a certain value.

  [97] [98]

3.7.3 Boolean Mechanism:

According to the availability of information in the given sources, there is no particular Boolean process which has been defined. ”Boolean mecanisms” are not typical for differential privacy. [93] [94]

Advantages:

- Offers privacy assurances for boolean-value-sought queries.
- May be more efficient than the Laplacian or Gaussian mechanisms for specific search terms.

[97] [98]

Limitations:

- Constrains the analysis to only boolean-valued queries, making it less useful in more flexible scenarios.
- The privacy guarantees may be weaker compared to the Laplacian or Gaussian mechanism, though.

  [97] [98]

3.7.4 Geometric Mechanism:

The sources disclosed do not mention any explicit Geometric mechanism. In the area of differential privacy, geometric mechanisms are not very popular at all. [93] [94]

Advantages:

- Only applicable for queries that return discrete values, for example count results, or histograms.
- For specific types of queries, it can offer stronger bounds on privacy than the Laplacian mechanism.

  [97] [98]

Limitations:

- Lacking clear extensibility to other forms of queries, particularly those that involve range- or frequency-based results.
- It can be weaker than Laplacian or Gaussian mechanisms in terms of privatization guarantees for the added noise.

  [97] [98]

3.7.5 Exponential Mechanism:

This mechanism draws its sample from a problem dependent set of distributions in order to achieve differential privacy. It is used for designing various algorithms that require differential privacy. [93] [94]

Advantages:

- May be employed for carrying out any kind of query functions, even if the data that it is working on is non-numeric.
- Offers a high level of privacy preservation owing to its propensity for sampling from a predetermined probability distribution.

[97] [98]

Limitations:

- It can sometimes be more computationally demanding than some of the other methods such as the Laplacian or Gaussian mechanisms.
- Difficult for achieving the intended privacy-utility balance, hence, the need to achieve the right utility function.

  [97] [98]

These mechanisms are applied towards gaining differential privacy in numerous statistical analysis operations such as machine learning and synthetic data computation. It offers robust privacy assurances to respondents since an antagonist cannot gain knowledge of the particulars of an individual even if they have access to unrestricted computing resources and know the specifics of the algorithm and system that is used in data acquisition and analysis.

Probability distribution	Noise	Use cases	Privacy leakage
Laplace	Introduces real-valued perturbations sampled from the Laplace distribution Lap (0, $\Delta f/\varepsilon$).	Safeguards query results, datasets, and gradients.	ε-DP
Gaussian	Adds real-valued noise drawn from the Gaussian distribution N (0, $\Delta^2 f/2\varepsilon^2$).	Shields query results, datasets, and gradients.	(ε, δ)-DP
Binomial	Incorporates discrete values drawn from the Binomial distribution (Bin(N, p) - Np)s.	Shields query results and datasets.	(ε, δ)-DP
Geometric	Adds a discrete value δ with probability $P(\Delta = \delta) = (\frac{1-e^{-\epsilon}}{1+e^{-\epsilon}} e^{-\epsilon\|\delta\|}$.	Protects query results and datasets.	ε-DP
Exponential	Selects a random output with probability $e^{(\frac{\varepsilon\mu(D,y)}{2\Delta\mu})}$	Safeguards learning models.	ε-DP

Figure 3.3: Summary of Probability Distributions with their satisfaction and their use Cases. [63]

3.8 Differential Privacy in Deep Learning

3.8.1 Overview of Adding Noise in Deep Learning :

Here is a concise overview of noise injection for training neural networks, written for a general audience:

Motivation for Adding Noise

They said that noise injection is one of the methods that aids in enhancing the performance and also the reliability of the neural networks. This is typically achieved by inserting slight independent random noise to the inputs or hidden layers of the model during training making it less likely to over-fit training data and thus capable of predicting unseen data accurately.[99]

The main benefits of adding noise are:
Improved Generalization: This type of noise aids the model in learning the features of the training set in a more generalized manner than the straightforward training of the aleatory data.
Increased Robustness: Noise allows the model to generalize and is beneficial for the next steps when there can be small perturbations or distortions of the input data.
Regularization: Noise in neural networks serves an important function as a source of regularization, which helps in preventing the model from memorizing data from the training set. [99] [103]

Practical Considerations

Choosing the right technique for noise injection and fine-tuning of the hyperparameters such as the amount and type of noise added can become crucial for achieving the result. High levels of noise may reduce performance while low levels may not bring about the necessary emanation of regularization effects.

The correct way of how the noise should be injected can, therefore, take more testing before correct validation on a validation set, since the method can be very dependent on the concrete utilization of the method as well as the data and architecture of the utilized models. [99] [103]

Different points where noise can be added in deep learning:

- Input Noise:

  Motivation: We can also increase the training set by applying some noise on the data which can assist the model not to overfit the input data and this is very vital due to real life applications.

  Techniques: Some of the regular input noise types are Gaussian noise, salt and pepper noise and also typical data augmentation approaches such as cropping, flipping, and rotation.

  Example: Some researchers apply Gaussian noise to the inputs in order to improve classification when a great variation of the input is observed.

  [100] [101]

- **Advantages:**

  Enhances the quality of a model by making it less sensitive to minor modifications of the input data.

  Could help the model to learn more generalistic attributes.

  Other than that, it can be employed alongside data augmentation approaches to boost the performance.
  [102] [104]

- **Limitations:**

  The kind of noise and the quantity of noise employed required to be optimal in order to not deteriorate the model quality.

  Different types of noise may be more appropriate in different contexts or indeed may be completely inapplicable to certain types of input data (for example, gausian noise may be

unsuitable for text or audio inputs).

Introducing noise into the input can be beneficial in terms of augmenting the data but it can also cause more computation during the training.

[102] [104]

- Gradient Noise:

  Motivation: Applying noise to the gradients during optimization implies the possibility of avoiding sharp local minima and allowing the model to jump out of such states and explore more of the local space, thus improving generalization performance of the model.

  Techniques: Additive noise can be imposed through the injection of Gaussian noise or other stochastic noise distributions into the gradients within the backpropagation phase.

  Example: Despite approaches such as canceling gradient noise through Gradient Noise Injection (GNI), or Stochastic Gradient Descent with Warm Restarts (SGDR), added noise to the gradients to enhance the optimization procedure.

  [100] [101]

- **Advantages:**

  Could potentially assist the model in the avoidance of ending up at a suboptimal solution in local minima and possibly cover more areas of the search domain.

  Faster convergence may help in achieving better generalization performance.

  Very important and can be used in conjunction with other optimization techniques such as stochastic gradient descent.
  [102] [104]

- **Limitations:**

  The location and intensity of noise ought to be thoroughly adjusted because of unfavorable effects.

  This often occurs when the system is introduced to too much noise, an occurrence which may cause the optimization to fluctuate at some point and return to give substandard solutions.

  However, complexity of adding gradient noise as a form of regularization may be a drawback since it may consume more computational power as compared to other forms of regularization.
  [102] [104]

- Output Noise:

  Motivation: It is also proposed here that applying output noise could make the model less sensitive to changes in input decisions in a similar manner to input noise. It can also be used for uncertainty quantification because the quality of data is often unknown.

  Techniques: Output noise can be produced by applying Dropout or individuals Gaussian noise to the last layer of the neural network.

  Example: In a regression task, noise is frequently incorporated into the model's output to reflect the uncertainty he had over the given results.

  [100] [101]

- **Advantages:**

  It increases model stability with the input data or, in other words, makes a model less sensitive to small changes in the input data.

Can be used for quantification of uncertainties in the model, as it provides probabilistic interpretation of the results.

It can also be combined with other noise injection methods such as Dropout with ease.

[102] [104]

- **Limitations:**

  Several considerations can be avowed when working on the robustness of Random Forests, including the type of noise and the amount of noise whereby measures need to be taken to ensure that the noise does not worsen the model's prediction capability.

  As it is applied at the end of the network, the output noise may not be as efficient as the input noise for generalization of architectures.

  Due to the model capacity to learn initial, robust features, the presentation of output noise may not influence the model's performance significantly.
  [102] [104]

- Label Noise

  Motivation: Augmenting the target labels during training can also assist the model, since by making the labels of the training examples more noisy, the model will not be affected by a given noise or corruption in the real data set which may contain erroneous labels.

  Techniques: Label noise can be generated artificially by either flipping random samples or introducing noise in the target labels with some fixed probability.

  Example: In a typical binary classification scenario where introducing high percentage of label noise degrades the performance superb features resulting from flipping of ground-truth labels by even a small percentage during training can be very helpful. [100] [101]

- **Advantages:**

  This makes the model less sensitive to noise or data corruption where some of the labels in the training set may be incorrect.

  Somewhat useful when dealing with real-world data where some labels might have been assigned inaccurately.

  It can be used with other basic methods of regularization for high accuracy of the model.
  [102] [104]

- **Limitations:**

  The amount of label noise that should be added to the training data requires a correct evaluation, as adding too much noise will impose a significantly negative impact on the model.

  Perhaps, the label noise is not as powerful as input or gradient noise in managing to improve the model's ability in the generalization of datasets.

  Though the idea of applying label noise is less complex in implementation, it could be daunting at times as the model may require extra interventions to learn the true mapping of the symbols and features used in the recognition of objects.
  [102] [104]

In conclusion, while each noise injection method has relative pros and cons, the decision of where to apply noise has to depend on the nature of the problem, the structure of the model under consideration, and the expected benefits of the regularization in terms of accuracy, stability, and calibration.

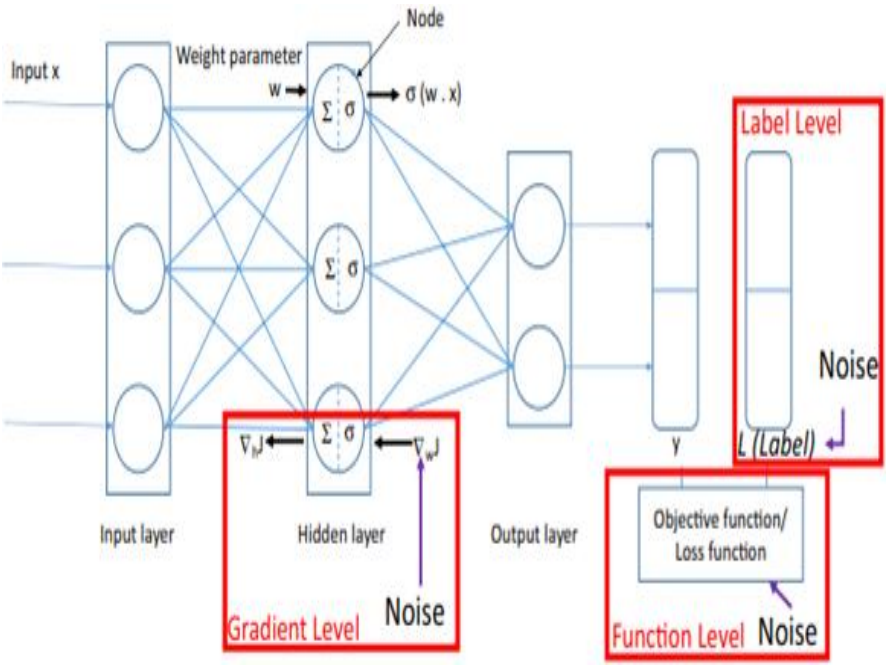


Figure 3.4: Protect privacy in the deep learning model. [105]

3.9 Focus on Gradient Noise

3.9.1 Importance of Adding Noise to Gradients

A common and preferred method of adding noise to gradients is used in most deep learning applications since it enables the enhancement of the optimization phase and generalization of the model. There is need to explain the relevance of this technique and then analyse the benefits of the technique and the possible demerits.

3.9.2 The reason why adding noise to gradients is a preferred approach in many scenarios

Pay Attention to Gradient Noise, in many situations, adding noise to the gradients of deep learning models is the better course of action for the following reasons:

Improved Optimization:

By including noise in the gradients, the optimization process can more successfully explore the loss landscape and break free from local minima. By adding stochasticity, the noise enables the model to overcome "transient plateau" stages in its early learning and come

Reduced Overfitting:

By reducing the model's ability to retain particular patterns, gradient noise helps keep the model from overfitting to the training set. This enhances the model's ability to generalize to fresh, untested data.

Sturdiness Against Disappearing Gradients:

The vanishing gradients problem, in which gradients get incredibly small and learning becomes ineffective, can arise in very deep neural networks. In order to prevent this problem and maintain enough gradient values to propagate throughout the network, noise can be added to the gradients.

Regularization Effect:

Weight regularization methods like L1 or L2 regularization have an effect akin to injecting noise into the gradients. It pushes the model to pick up more reliable and steady features.

Adaptability to Different Architectures:

Gradient noise can be used with a wide range of neural network architectures, such as convolutional networks, memory networks, and neural GPUs, as well as very deep fully-connected networks.

- Advantages and potential drawbacks:

  Escaping Local Minima: Gradient noise can assist in assisting the optimization process in escaping local minima and allowing it to investigate a larger portion of the parameter space.

  Better Generalization: By keeping the model from overfitting to the training set, gradient noise can improve generalization performance. Because it forces the model to investigate a wider range of solutions, the additional noise can aid in the model's learning of more reliable and generalizable features.

  Faster Convergence: Gradient noise has the potential to speed up the optimization process in certain circumstances, particularly when paired with methods like stochastic gradient descent (SGD).

  Better Exploration: During training, gradient noise can motivate the model to explore a larger portion of the parameter space, which may result in the discovery of superior local minima.

- Potential Drawbacks:

  Hyperparameter Tuning: To balance the advantages of exploration and the danger of upsetting the optimization process, the ideal quantity and kind of gradient noise must be carefully adjusted.

  Selecting the proper noise magnitude and distribution: (such as Gaussian or Uniform) can be difficult and need a lot of trial and error.

  Computational Overhead: Because noise must be generated and applied to the gradients, adding noise to gradients may result in additional computational overhead during the training process. When employing computationally costly noise distributions or large-scale deep learning models, this overhead may be especially noticeable.

  Stability Issues: The optimization process may become unstable and the model may diverge or converge to a suboptimal solution if the gradient noise is too great or the noise distribution is inappropriate. To keep the training process stable, close observation and tweaking of the noise parameters are required.

Steps in DP-SGD:

1. **Gradient Clipping:** The gradients calculated for each training example are scaled by a maximum norm so that their sensitivity is limited. This makes it possible to avoid the situation where a single data point has a great impact on the model.

Algorithm 1 Gradient Descent with Privacy Accounting

Initialize ϑ_0 randomly
for $t = 0$ to L **do**
    Compute gradient
    for each i **do**
        Compute $g_t(x_i) = \nabla_\vartheta L(\vartheta_t, x_i)$
    end for
    Clip gradient
    $g_t(x_i) = g_t(x_i) / \max(1, \|g_t(x_i)\|_2 / c)$
    Add noise
    $g_t = (1/L)(\sum_{i=1}^{L} g_t(x_i)) + N(0, \sigma^2 C^2 I)$
    Descent
    Update parameter: $\vartheta_{t+1} \leftarrow \vartheta_t - \eta_t \cdot g_t$
end for
Output ϑ_T and compute the overall privacy cost (ϵ, δ) using a privacy accounting method.

[106]

2. **Noisy Gradient Averaging:** The clipped gradients are then averaged and Gaussian noise is added to the averages. This noise addition is crucial for providing differential privacy and thereby hiding the contribution made by each point in the data set.

3. **Moment Accounting:** The Moments Accountant is an approach that is applied in the computation of the total privacy loss up to a certain iteration. It tracks the total noise that has been added to the gradients and helps the algorithm stay within the privacy budget set.

Importance of the Moments Accountant: The Moments Accountant serves two critical functions in DP-SGD:

1. **Privacy Budget Tracking:** It enables the algorithm to monitor the privacy budget, which is the amount of privacy incurred in the training process. This tracking helps to avoid the leakage of privacy information and to keep the algorithm's privacy budget within the appropriate limits.

2. **Gradient Update:** In the following formula, the information stored in the Moments Accountant is used to adjust the gradient update step. Such an adjustment guarantees that the model parameters are updated in a way that preserves differential privacy.

Information Stored in the Moments Accountant The Moments Accountant typically tracks:

- The total amount of Gaussian noise that is added on the gradients.
- The total number of training steps, which is the number of steps taken in the training process.
- The privacy parameters the target privacy budget and the privacy parameter (ϵ).

By keeping track of this information, the DP-SGD algorithm can decide on the amount of Gaussian noise that needs to be added to the gradients in the current iteration by calculating the remaining privacy budget. This dynamic adjustment optimizes the model performance and the privacy guarantee requirements simultaneously.

To conclude, the Moments Accountant is an important part of DP-SGD which gives a precise way of measuring and controlling privacy loss. It allows for the use of differential privacy in deep learning by making sure that the privacy budget is used optimally and by preserving privacy parameters up to the final iteration. [99] [100] [101] [103] [105]

3.10 Implementation of Differentially Privacy-SGD

Differential Privacy Stochastic Gradient Descent (DP-SGD) is a technique used to train machine learning models while ensuring that the privacy of individual data points is preserved. This is achieved by adding noise to the gradients during the training process. In this section, we will discuss the implementations of DP-SGD in two popular deep learning frameworks: TensorFlow and PyTorch. We will also highlight the differences between their implementations. [63] [106] [107]

3.10.1 DP-SGD in TensorFlow

TensorFlow, developed by Google, offers a comprehensive implementation of DP-SGD through the tensorflow-privacy library. This library provides tools and functions that allow developers to easily integrate differential privacy into their models. - Key Features of TensorFlow's DP-SGD Implementation: Ease of Integration: The tensorflow-privacy library is designed to be integrated seamlessly with existing TensorFlow models. It extends the existing tf.keras API, making it easy for developers to add differential privacy to their models. Gradient Clipping and Noise Addition: The library includes functions for gradient clipping and noise addition, which are essential for DP-SGD. Gradient clipping ensures that the sensitivity of the model to individual data points is bounded, while noise addition provides the differential privacy guarantee. Configurable Privacy Parameters: TensorFlow's DP-SGD implementation allows users to configure privacy parameters such as the noise multiplier, clipping norm, and the privacy budget (epsilon). Performance Optimization: The library includes optimizations to minimize the performance overhead introduced by differential privacy, ensuring that models can be trained efficiently. [63] [106] [107]

3.10.2 DP-SGD in PyTorch

PyTorch, developed by Facebook, provides a different approach to implementing DP-SGD through the Opacus library. Opacus is designed to be flexible and easy to use, allowing developers to add differential privacy to their PyTorch models with minimal changes to their existing code. - Key Features of PyTorch's DP-SGD Implementation: Modular Design: Opacus is built to be modular, allowing developers to plug differential privacy into their models without significant modifications. It supports various privacy-preserving optimizers and can be easily extended. Per-Sample Gradient Computation: Unlike TensorFlow, Opacus computes per-sample gradients, which allows for more fine-grained control over the privacy guarantees. This approach also enables more efficient gradient clipping. Configurable Privacy Parameters: Similar to TensorFlow, Opacus allows users to configure privacy parameters such as the noise multiplier, clipping norm, and the privacy budget (epsilon). Support for Different Optimizers: In addition to DP-SGD, Opacus supports other privacy-preserving optimizers, giving developers more flexibility in choosing the best optimizer for their specific use case. [63] [106] [107]

3.10.3 Differences Between TensorFlow and PyTorch Implementations

Library Integration: TensorFlow uses the tensorflow-privacy library, which extends the tf.keras API, making it easier to integrate with existing TensorFlow models. PyTorch uses the Opacus

library, which is more modular and flexible, allowing for integration with various optimizers and models.
Gradient Computation: TensorFlow's implementation focuses on batch-level gradient clipping and noise addition, whereas PyTorch's Opacus computes per-sample gradients, allowing for more fine-grained control over the privacy guarantees.
Performance Optimization: Both libraries include performance optimizations to minimize the overhead introduced by differential privacy. However, PyTorch's per-sample gradient computation can be more computationally intensive compared to TensorFlow's batch-level approach.
Configurable Privacy Parameters: Both implementations allow users to configure privacy parameters such as noise multiplier, clipping norm, and privacy budget. However, PyTorch's Opacus provides more flexibility in choosing different privacy-preserving optimizers.
In conclusion, both TensorFlow and PyTorch offer robust implementations of DP-SGD, each with its own strengths and weaknesses. TensorFlow's implementation is easier to integrate with existing models, while PyTorch's implementation provides more flexibility and fine-grained control over the privacy guarantees. [63] [106] [107]

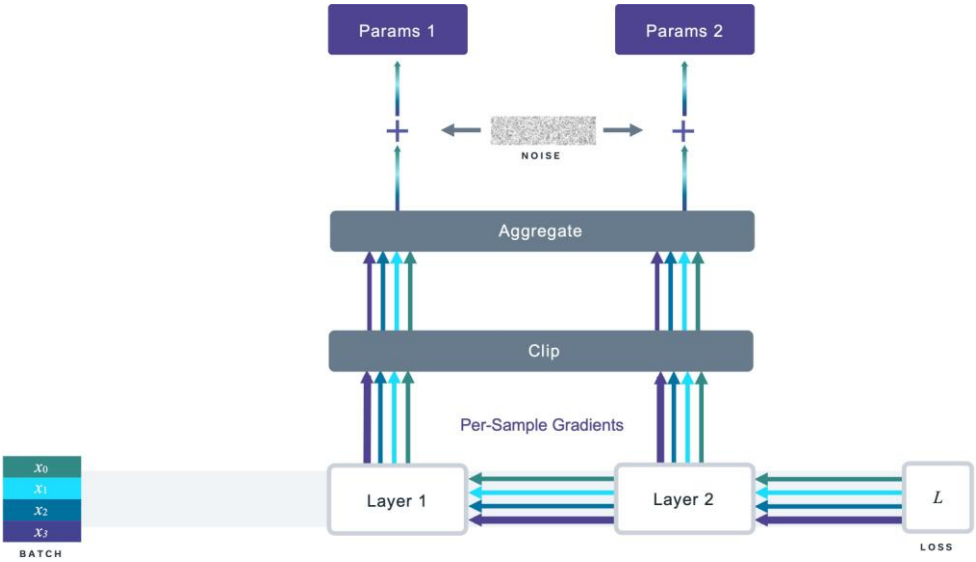


Figure 3.5: Pytorch-Opacus

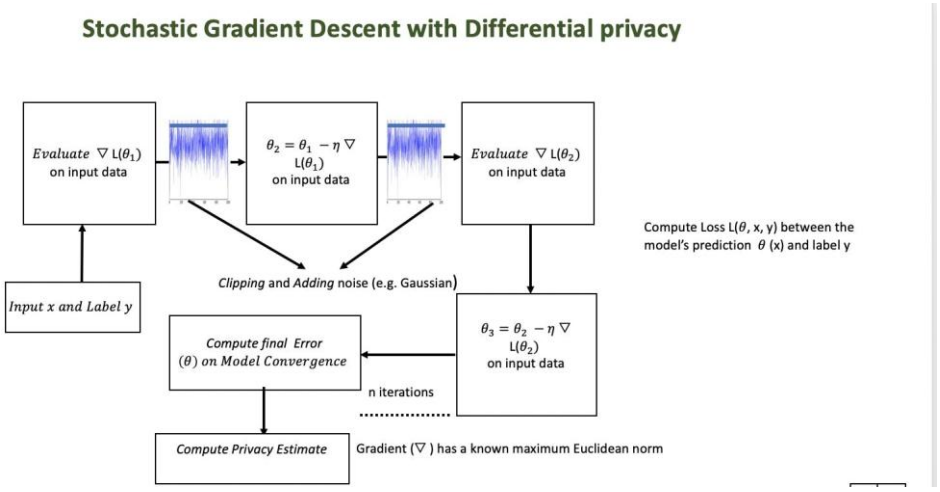


Figure 3.6: TensorFlow-privacy

3.11 Conclusion

In conclusion, this chapter covers the essential aspects of differential privacy (DP) within the context of deep learning. It highlights how DP techniques help in safeguarding sensitive information within training data, thereby preventing data breaches and misuse. The chapter underscores the significance of DP by illustrating its ability to maintain the balance between data utility and privacy, which is paramount in modern data-driven applications.

CHAPTER

4

IMPLEMENTING AND DEPLOYING A PRIVACY-PRESERVING MODEL: A WEB-BASED PREDICTION SERVICE

4.1 Introduction

This chapter investigates the performance of deep learning models for breast cancer diagnosis, evaluating both non-private and differentially private settings. We employed a comprehensive approach to train and fine-tune two versions of the model, aiming to balance accuracy, generalization, and privacy. For the non-private model, we focused on optimizing the key hyperparameters: learning rate, batch size, and optimizer. We utilized grid search to systematically explore a wide range of hyperparameter values and determine the optimal configuration. Additionally, we implemented early stopping cross- validation as a strategy to prevent overfitting, which can occur when the model becomes too tailored to the training data. This technique monitors the model's performance on a validation set and halts training when the performance stops improving, ensuring that the model retains its ability to generalize well to new, unseen data.

The differentially private model built upon the configuration of the non-private model by incorporating privacy-preserving mechanisms. Specifically, we added privacy-specific hyperparameters such as target epsilon, delta, and maximum gradient norm. These parameters are crucial for controlling the trade-off between privacy and utility in the model. By fine-tuning these additional hyperparameters, we aimed to achieve a model that maintains high accuracy while adhering to the principles of differential privacy, thereby protecting sensitive patient information.

We applied the grid search method to both models, allowing us to exhaustively search through the hyperparameter space and identify the best-performing configurations. This methodical approach ensured that we could optimize the performance of both models to their fullest potential. To further enhance the reliability of our models, we utilized early stopping as a method to fine-tune the number of training epochs. This approach helps to avoid the risk of over-fitting by

terminating the training process once the model's performance on the validation set begins to degrade, rather than continuing to train until a predetermined number of epochs is reached. The ultimate goal of this work was to develop robust models that could be deployed in a web-based prediction service. This service aims to provide accurate breast cancer predictions to both individual patients and medical institutions. By deploying the best-performing non-private and private models, we ensure that users can receive reliable predictions while their sensitive information is safeguarded. This dual approach underscores our commitment to delivering high-accuracy medical predictions without compromising on privacy.

Overall, this chapter provides a detailed examination of our methodology and the steps taken to ensure that our deep learning models are both effective and privacy-preserving.

4.2 Methodology

4.2.1 Dataset

Breast Cancer Wisconsin (Diagnostic) is a large public dataset and well known dataset among the datasets used for machine learning and healthcare research [1]. The dataset used in the study was compiled by **Dr.William Wolberg**, **W. Nick Street**, and **Olvi Mangasarian** from the University of Wisconsin Hospitals, Madison–it contains fluid samples extracted from patients with solid breast tumours[2].It helps to analyze cell nuclei characteristics for more effective diagnosis of breast cancer[1].
This real world dataset is obtained from the digitization of fine needle aspirate **FNA** of breast masses and it includes features derived from the digits of **FNA** images[1]. Figure 4.1 shows the first five rows of the breast cancer dataset:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mea
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.2776
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.0786
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.1599
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.2839
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.1328
...	...	...	...	...	...	...	...	..
564	926424	M	21.56	22.39	142.00	1479.0	0.11100	0.1159
565	926682	M	20.13	28.25	131.20	1261.0	0.09780	0.1034
566	926954	M	16.60	28.08	108.30	858.1	0.08455	0.1023
567	927241	M	20.60	29.33	140.10	1265.0	0.11780	0.2770
568	92751	B	7.76	24.54	47.92	181.0	0.05263	0.0436

569 rows × 33 columns

Figure 4.1: Section of breast cancer dataset showing first five rows [3] .

A. Characteristics of the Dataset

The dataset of breast cancer comprises of a total of 569 samples. In total, there are 32 features that are used to describe samples, the first of which is the ID of the sample and the second is its class, while the rest of them, which are 30, contain some information about the cells. The class label of our samples can be malignant(**M**) or benign(**B**). These are medical terms that relate to tumors that are non cancerous and those that are cancerous in nature. As for the properties, there are no any missing values. The samples of this real world dataset is split by 357 samples

of benign and 212 samples of malignant (Figure 4.4) [4]. In Figures 4.2 and 4.3 , the images that make up our dataset are the originals [4] :

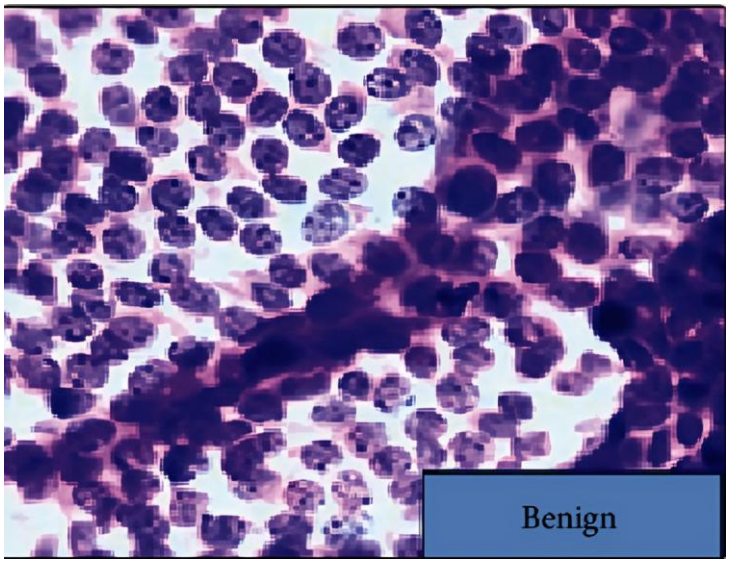


Figure 4.2: Image of a benign tumor cell.

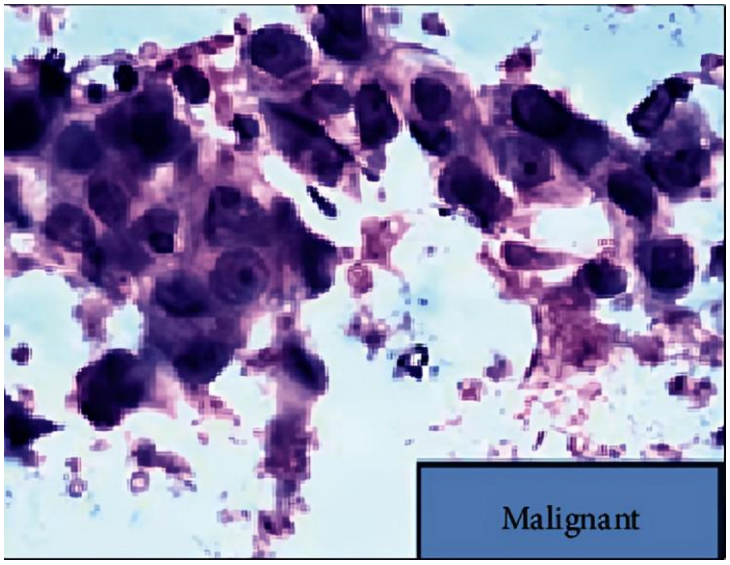


Figure 4.3: Image of a malignant tumor cell .

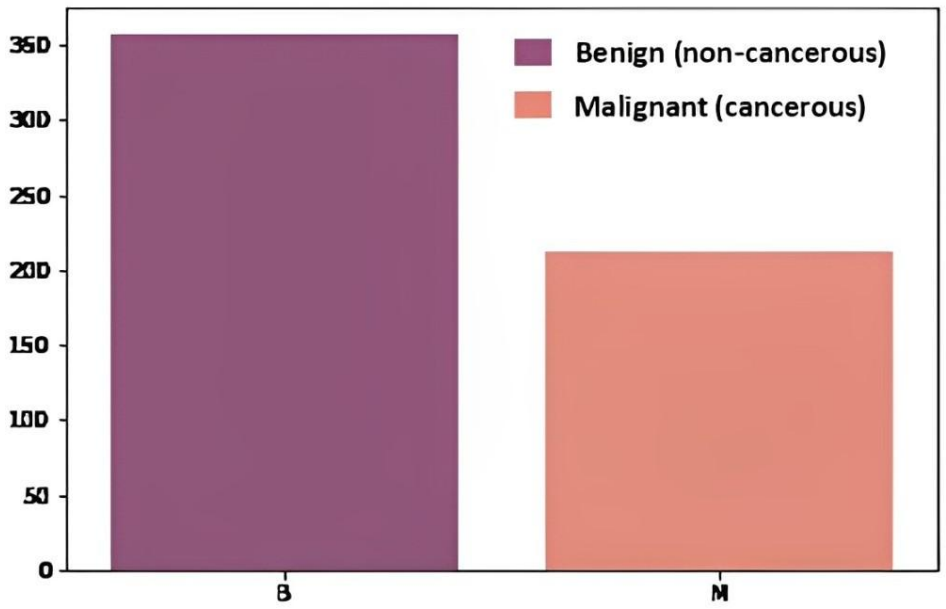


Figure 4.4: Distribution of benign and malignant patients of breast cancer .

B. Features of the Wisconsin Breast Cancer Dataset

The dataset contains ten main features and each feature is computed for the mean, standard error, and worst (largest) values, resulting in a total of 30 features [4][5] as it shown in the table (Figure 4.5) bellow:

Feature Group	Mean Feature	Standard Error Feature	Worst Feature
Radius	Radius_mean	Radius_se	Radius_worst
Texture	Texture_mean	Texture_se	Texture_worst
Perimeter	Perimeter_mean	Perimeter_se	Perimeter_worst
Area	Area_mean	Area_se	Area_worst
Smoothness	Smoothness_mean	Smoothness_se	Smoothness_worst
Compactness	Compactness_mean	Compactness_se	Compactness_worst
Concavity	Concavity_mean	Concavity_se	Concavity_worst
Concava Points	Concava Points_mean	Concava Points_se	Concava Points_worst
Symmetry	Symmetry_mean	Symmetry_se	Symmetry_worst
Fractal Dimension	Fractal Dimension_mean	Fractal Dimension_se	Fractal Dimension_worst

Figure 4.5: A summary table of the breast cancer dataset features .

C. Usage of the Features

- **Mean**: The average value of each feature across all cells in the sample.

- **Standard Error**: The standard error of the mean, providing an estimate of the variability of the feature.
- **Worst**: The maximum value of the feature observed across all cells in the sample.

4.2.2 Models Architecture: Simple Models

A. Models implementation details

Both models employ a simple feedforward neural network , this model is designed for classification tasks and consists of three fully connected layers. Below are the details of each layer in the architecture:

Components

- **Input Layer**
  Input Features: The input layer consists of 30 features. These features are the attributes from the dataset that describe each data point.
- **First Hidden Layer**
  Layer: This layer contains 20 neurons.
  Description: This layer takes the input features , maps them to its neurons and then applies a linear transformation to the incoming data allowing the model to learn a variety of features .
  Activation Function: After the linear transformation, a **ReLU** activation function is used to introduces non-linearity into the model and helps it learn complex patterns.
- **Second Hidden Layer**
  Layer: This layer contains 10 neurons.
  Description: This layer further processes the representation learned by the first hidden layer (20 features) mapping it to the 10 neurons , enabling the model to capture more abstract features.
  Activation Function: **ReLU** activation function is applied again to the output of this layer.
- **Output Layer**
  Layer: This layer contains 2 output features.
  Description: The final layer maps the 10 features from the second hidden layer to the output features, which corresponds to the number of output classes (for binary classification).
  No Activation Function: The raw output from this layer is typically passed to a loss function , which applies a **softmax**function internally

Forward pass

The forward method defines how the data flows through the network . The input data is passed through the first hidden layer . Then the output from the first hidden layer is passed through the second hidden layer. After that, the output from the second hidden layer is passed through the final output layer . finally , the final output is then returned , which will be used for further computation (calculating loss and updating the model weights during training).

B. Data Preparation

- Read CSV file into a Pandas Data Frame.
- Extracts features and labels from the Data Frame.

- Encodes the categorical labels into numerical values.
- Splits the data into training and testing sets.
- Converts the data into PyTorch tensors and creates datasets for training and testing.
- Creates data loaders for the training and testing datasets to enable batch processing.

C. Training the Models

For training a deep learning model, training data is provided to the model and the parameters of the model modified in an attempt to minimize prediction error on different data. To improve this process, we used the grid search to select parameters, early stopping technique to avoid overfitting, cross validation to improve generalization.

1. **Training with Grid Search**

   - **Grid Search Technique**: developed as an exhaustive search method, evaluates all possible combinations of hyperparameters to identify the optimal set.

   - **Grid Search Process:**It involves the following steps: - Define the hyperparameter space. - Train the model on each combination of hyperparameters. - Evaluate model performance using cross-validation. - Select the combination that yields the best performance based on predefined criteria.

2. **Training with Early Stopping**

   - **Early Stopping:** Stops training when the validation loss does not improve for a set number of epochs(patience=10). **Patience:** the number of consecutive epochs with no improvement in the validation metric before training is halted.
   - **Training Loop:** Trains the model, calculates losses and accuracies for training and validation sets, and updates the model parameters.
   - **evaluate model:** Function to evaluate the model's performance on a given dataset (validation or test) without updating the model parameters.

3. **Training with Cross-Validation**

   - **cross-validation:** Performs k-fold cross-validation to evaluate different hyperparameters.
   - **KFold**: Splits the data into k folds for cross-validation(k=5).
   - **Hyperparameters:** Evaluates different combinations of optimizers, learning rates, and batch sizes.

D. Hyperparameters

The hyperparameters that are performed using the grid search technique to identify the best configuration for model performance are as follow:

1. **Non-private Model Hyperparameters**

   - **Optimizer:** methods to adjust neural network attributes like weights and learning rate to reduce losses.

- **Learning Rate:** controls the step size of the optimizer, determining how quickly or slowly the model updates its parameters to minimize the loss function.
- **Batch Size:** the number of training examples used in one iteration, determining how many samples are processed before updating the model's parameters.
- **Epochs:** a complete pass through the training dataset, with the number of epochs indicating how many times this process is repeated.

Types and Values of Hyperparameters

- **optimizer:** [**SGD** , **ADAM**]
- **learning Rates:** [0.001, 0.005, 0.01, 0.05, 0.1]
- **Batch Sizes:** [16, 32, 64, 128, 512, 1024]
- **Epochs:**500

2. Private Model Hyperparameters

The private model uses the same hyperparameters but includes additional ones to ensure differential privacy.

The private model includes additional hyperparameters to ensure differential privacy:

- **Privacy Budget (Epsilon):** Epsilon (ϵ) measures the privacy guarantee of the algorithm.
- **Privacy Loss(Delta):** Delta (δ) quantifies the probability of a privacy breach, measuring the likelihood of inferring sensitive information from a differentially private algorithm's output.
- **Clipping Norm(Max Gradient Norm):**a threshold to limit gradient magnitudes, preventing any single data point from disproportionately influencing the model's learning.

Types and Values of Hyperparameters

- **optimizer:** [**SGD** , **ADAM**]
- **learning Rates:** [0.001, 0.005, 0.01, 0.05, 0.1]
- **Batch Sizes:** [16, 32, 64, 128, 512, 1024]
- **Max grad norms:** = [1.2, 5.6]
- **Epsilons:** [1.0, 8.0]
- **Deltas:** [10^{-4}]
- **Epochs:**500

4.2.3 Experimental setup

To conduct our experimental study, we employed the computing platform: **Google Colab Pro**. The experiments were performed using the Python programming language, leveraging the **Pytorch librarie** for deep learning model development and evaluation.

a. Results and Discussion

1. Non-Private Model

The non-private model was trained using a range of hyperparameters. The best performance was achieved using the Adam optimizer with a learning rate of 0.001 and a batch size of 512, which resulted in a validation accuracy of 97.80% after 415 epochs and 96% of test accuracy. Figure 4.6 captures a sample of configurations, while the full table of all configurations can be found in the annex.

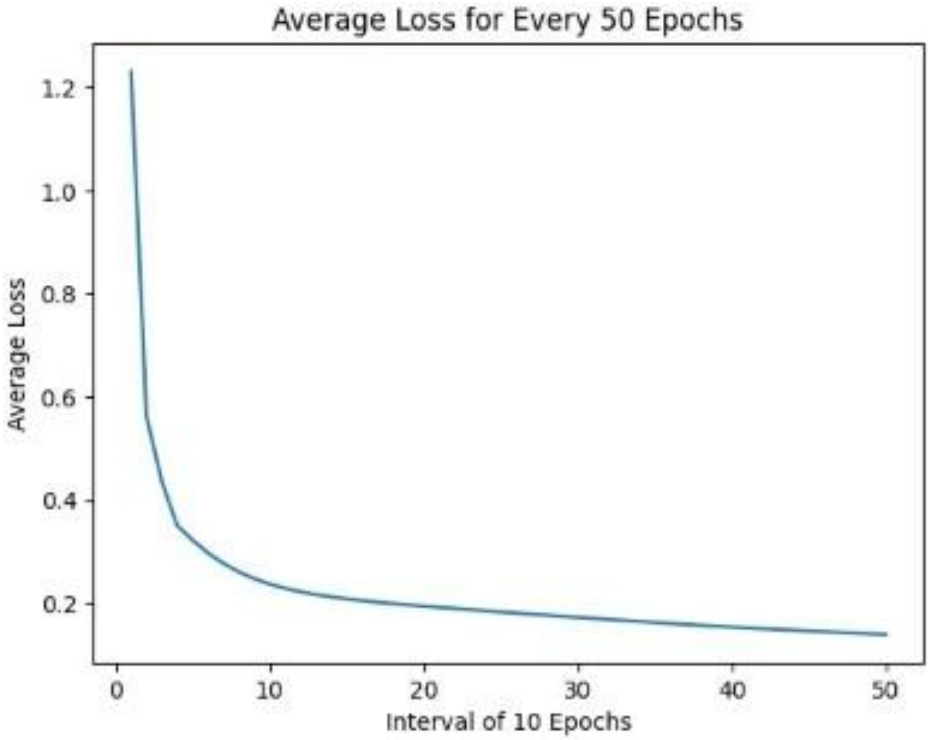
Optimizer	Learning Rate	Batch Size	Validation Accuracy	Epochs
Adam	0.001	512	97.80%	415
Adam	0.005	1024	94.50%	500
Adam	0.01	128	96.70%	92
SGD	0.001	32	95.60%	65
SGD	0.001	32	91.20%	66

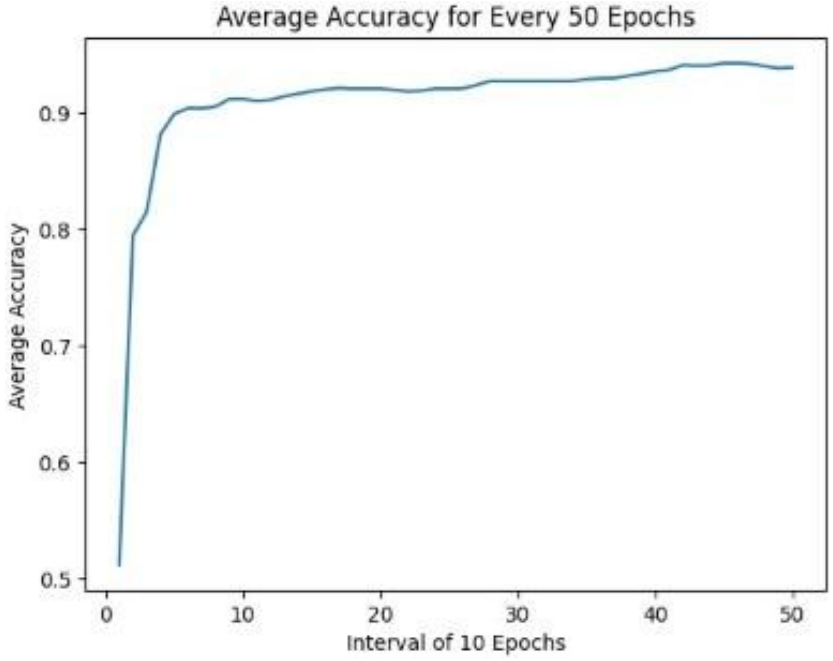
Figure 4.6: Non-Private Model Results.

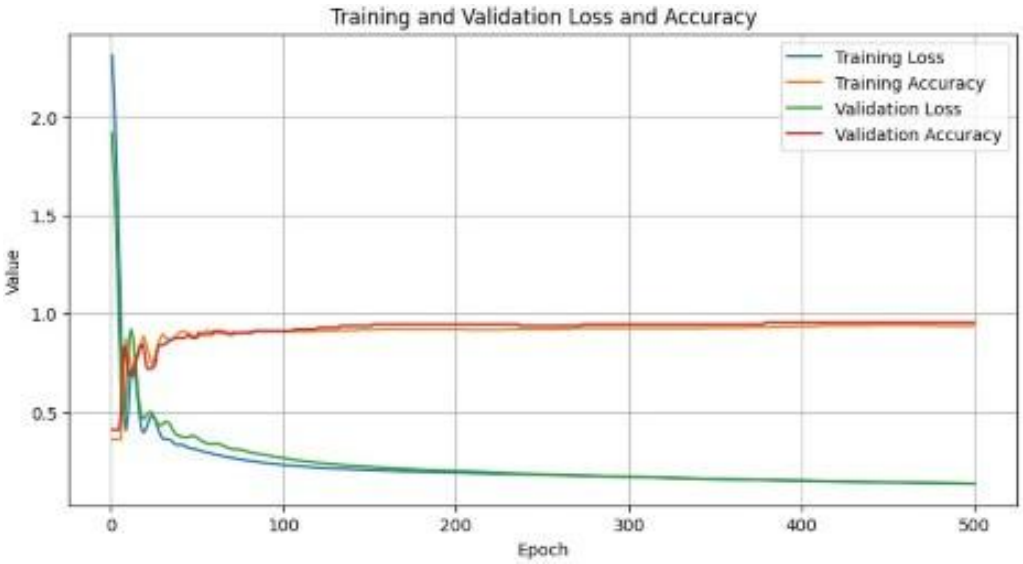
Classification Report: The classification report for the non-private model (with the best configuration) is as follows:

Class	Precision	Recall	F1-Score	Support
0	0.96	0.97	0.96	67
1	0.96	0.94	0.95	47
Accuracy			0.96	114
Macro Avg	0.96	0.95	0.95	114
Weighted Avg	0.96	0.96	0.96	114

Figure 4.7: classification report for the non-private model .


Average Loss for Every 50 Epochs
Average Loss
1.2
1.0
0.8
0.6
0.4
0.2
0
10
20
30
40
50
Interval of 10 Epochs


Average Accuracy for Every 50 Epochs
Average Accuracy
0.9
0.8
0.7
0.6
0.5
0
10
20
30
40
50
Interval of 10 Epochs



Confusion Matrix : The confusion matrix for the best performing non-private model configuration (**ADAM** optimizer, learning rate **0.001**, batch size **512**) is shown in Figure 4.8

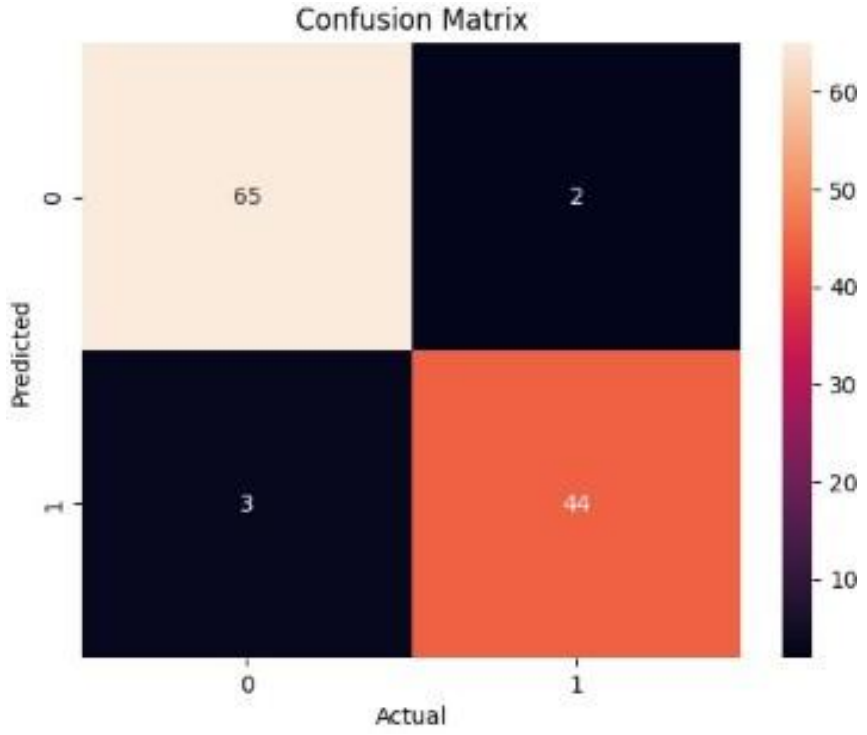


Figure 4.8: Confusion Matrix for non- private model .

Discussion

The results demonstrate the effectiveness of the non-private model in accurately classifying the Breast Cancer Wisconsin dataset. The **ADAM** optimizer consistently outperformed the **SGD** optimizer across various configurations. The highest accuracy was achieved with

the Adam optimizer, a learning rate of 0.001, and a batch size of 512. This configuration reached an impressive validation accuracy of 97.80% after 415 epochs, and when evaluated on the test dataset, the model achieved an accuracy of 96%. These findings highlight the significance of the chosen hyperparameters in achieving high validation and test accuracy, underscoring the robustness of the model under these specific conditions and its capability to generalize well to the test data.
The classification report reveals high precision, recall, and F1-scores for both classes, indicating that the model performs well in identifying both benign and malignant cases of breast cancer. The macro and weighted averages of the precision, recall, and F1-scores are all 0.96, underscoring the balanced performance of the model across different classes.
Additionally, the confusion matrix in Figure 1 shows that the non-private model has a high true positive rate and a low false negative rate, which is crucial for medical diagnoses where missing a positive case (false negative) can have serious implications. Specifically, the model correctly identified 65 out of 67 benign cases and 44 out of 47 malignant cases. These findings suggest that the non-private model, particularly with the Adam optimizer, is highly effective for the task of breast cancer classification.

2. **Private Model** The private model incorporated differential privacy parameters (ϵ, δ, σ, Max grad norm) along with the same hyperparameters as the non-private model. The best performance for the private model was achieved using the Adam optimizer with a learning rate of 0.05, a batch size of 512, sigma 72.5, epsilon of 1.0, delta of 10^{-4} , and max grad norm of 1.2, resulting in a validation accuracy of 96.70% after 17 epochs, and 80% of test accuracy.
We simulated a high level of privacy with epsilon = 1.0.
Figure 4.9 captures a sample of configuration, while the full table of all configurations can be found in the annex.

Optimizer	Learning Rate	Batch Size	Sigma	Epsilon	Delta	Max Grad Norm	Validation Accuracy	Epochs
Adam	0.05	512	72.5	1.0	0.0001	1.2	96.70%	17
Adam	0.001	1024	72.5	1.0	0.0001	1.2	75.82%	36
Adam	0.001	1024	72.5	1.0	0.0001	1.2	85.71%	87
SGD	0.1	32	3.588	8.0	0.0001	1.2	91.20%	31
SGD	0.05	64	5.02	8.0	0.0001	1.2	91.20%	18

Figure 4.9: (1.0,10^{-4})-DP model Fine-tuning results.

Classification Report : The classification report for the (1.0,10^{-4})-DP model (with the best configuration) is as follows:

Class	Precision	Recall	F1-Score	Support
0	0.94	0.95	0.94	67
1	0.93	0.92	0.93	47
Accuracy			0.94	114
Macro Avg	0.94	0.94	0.94	114
Weighted Avg	0.94	0.94	0.94	114

Figure 4.10: Classification Report for the Best Configuration of (1.0,10^{-4})-DP model.

Confusion Matrix : The confusion matrix for the best performing (1.0,10^{-4})-DP model configuration is shown in Figure 4.11:

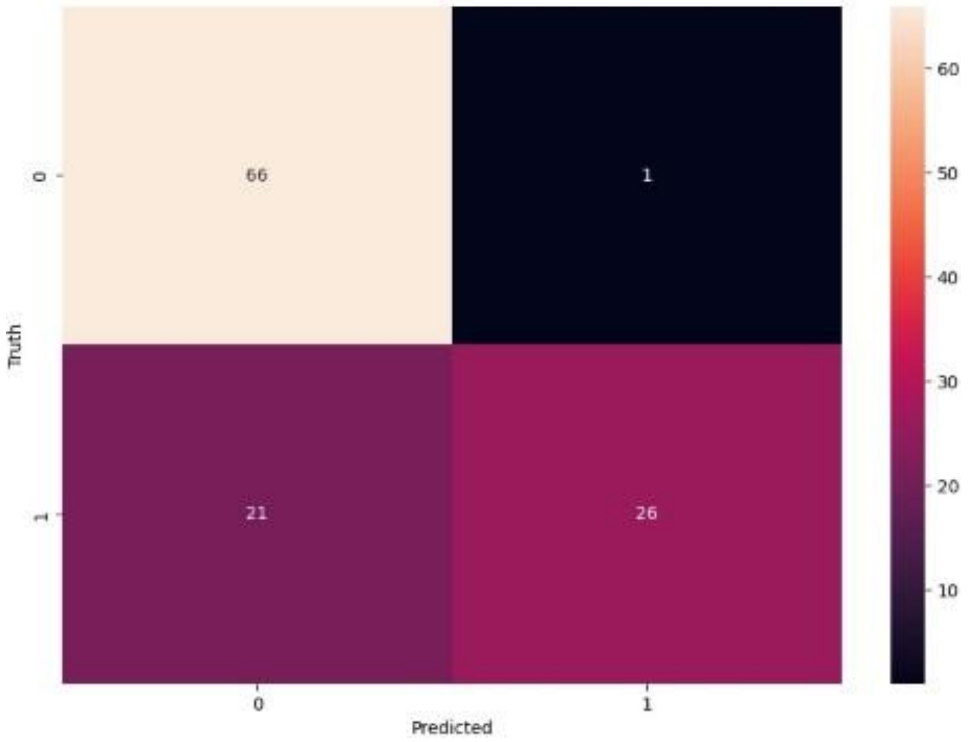


Figure 4.11: Confusion Matrix for private model .

Discussion

By fine-tuning hyperparameters of the differentially private model, we achieved a more balanced performance that ensures both high accuracy and enhanced privacy in breast cancer diagnosis. This optimization improves early detection and significantly benefits patient outcomes.

The private model results on the Breast Cancer Wisconsin dataset demonstrate its effectiveness in classifying benign and malignant cases while preserving data privacy. The best configuration, using the Adam optimizer with a learning rate of 0.05, batch size of 512, privacy budget of noise multiplier of 72.5 and privacy loss of 10^{-4} and gradient clipping at 1.2, achieved a validation accuracy of 96.70%.

The confusion matrix revealed 26 true positives, 66 true negatives, 1 false positive, and 21 false negatives, indicating a strong performance in identifying benign cases but a tendency to miss some malignant cases.

The model achieved a precision of 96% and a recall of 55% for malignant cases, and a precision of 76% and a recall of 99% for benign cases, with F1-scores of 70% for malignant and 86% for benign.

These metrics suggest a high reliability in predicting benign instances but highlight the need for better detection of malignant cases.

b. General Discussion

The results demonstrate the expected trade-off between accuracy and privacy when differential privacy techniques are applied. The non-private model achieved a slightly higher

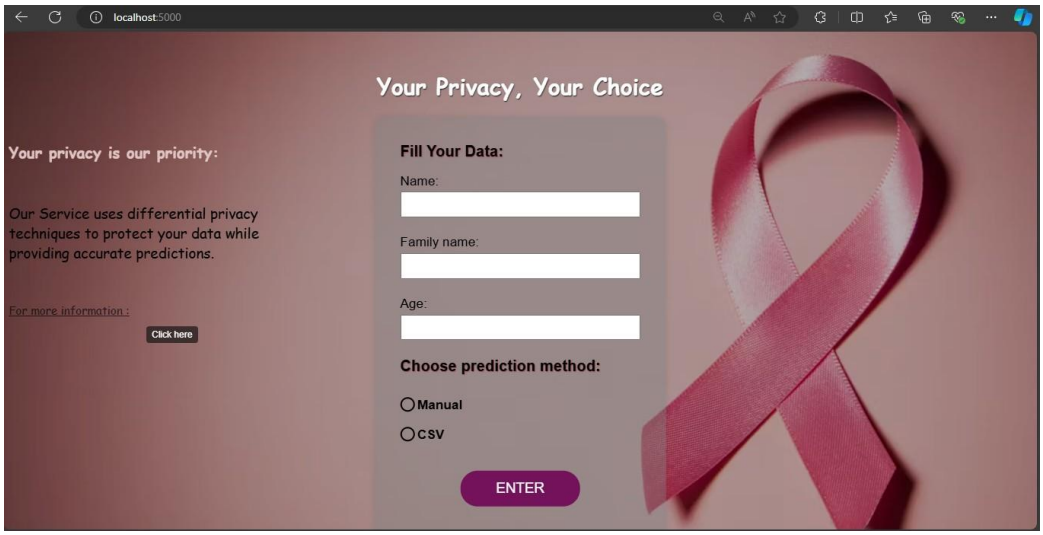
Validation accuracy (97.80%)(and test accuracy 96%) compared to the best private model (96.70%)(and test accuracy 80%). This small drop in accuracy is attributed to the noise added to ensure privacy, as governed by the differential privacy parameters.
Despite the slight reduction in accuracy, the private model maintained a high level of performance, validating the efficacy of differential privacy in protecting user data while retaining high accuracy. The results also indicate that tuning the differential privacy parameters (σ, ϵ, δ, max grad norm) is crucial in balancing the trade-off between privacy and accuracy.
The use of the **ADAM** optimizer consistently outperformed **SGD** in both non-private and private models. The highest accuracy for the non-private model was achieved with a learning rate of 0.001 and a batch size of 512, whereas for the private model, the best configuration was a learning rate of 0.01, batch size of 128,δ of 0.0001, and max grad norm of 1.2.

Synthetic data generation for making predictions:
To generate synthetic data for making predictions with our deep learning model, we utilized a synthetic data generation method. Specifically, we employed the CTGAN (Conditional Tabular GAN) library, which is a GAN-based approach for generating realistic tabular data based on our input dataset.

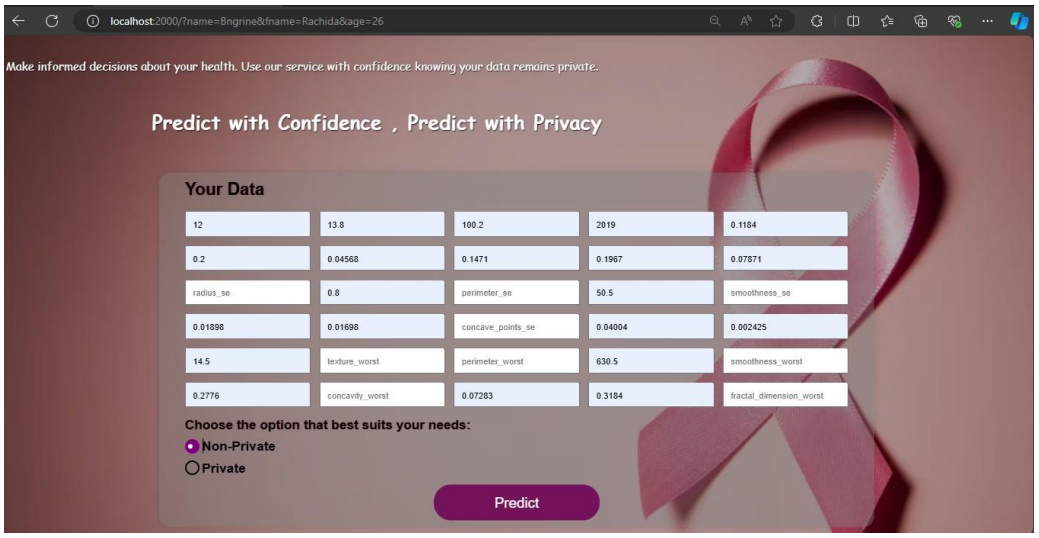
By using the CTGAN library, we ensured that the synthetic data generated for predictions retained the statistical properties and relationships of the original dataset, providing a robust input for our deep learning model.

4.2.4 Deployment of models

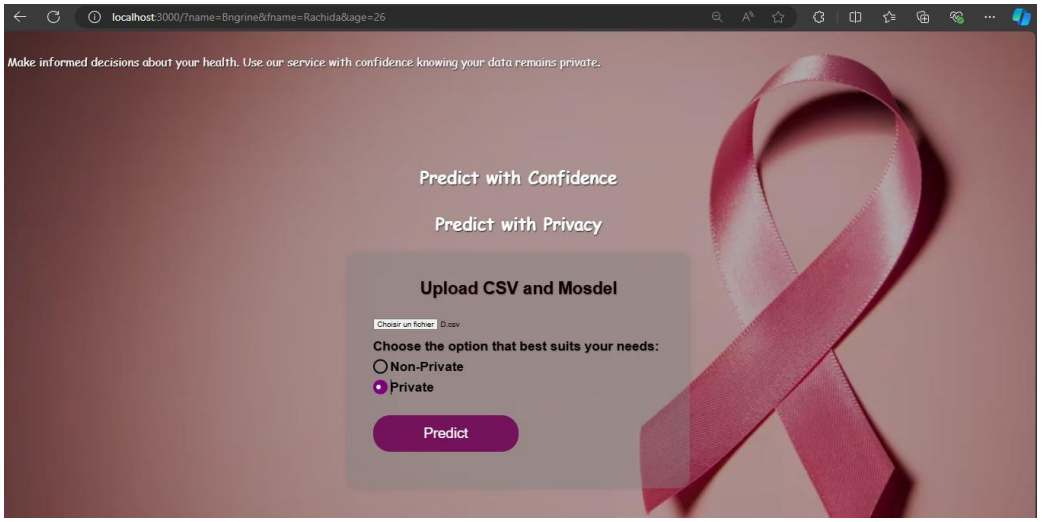
In this project, we designed a freshmen web application with the help of Flask for the integration of deep learning models with the HTML pages. This application allows users to input their personal information including name, family name, and age and interact with a breast cancer prediction model through two input methods: Manual data entry and uploading of CSV data files to the defined database.



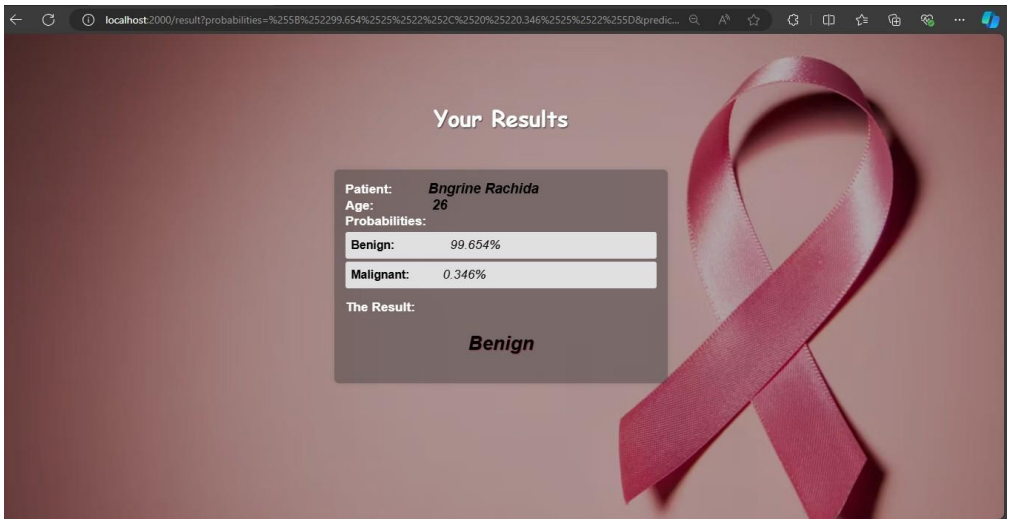
Then After choosing the way of interacting with our models , users get to the page where they upload more breast cancer related information by typing it in or uploading a CSV file and choose if they want private predictions or not : **Manual way:**



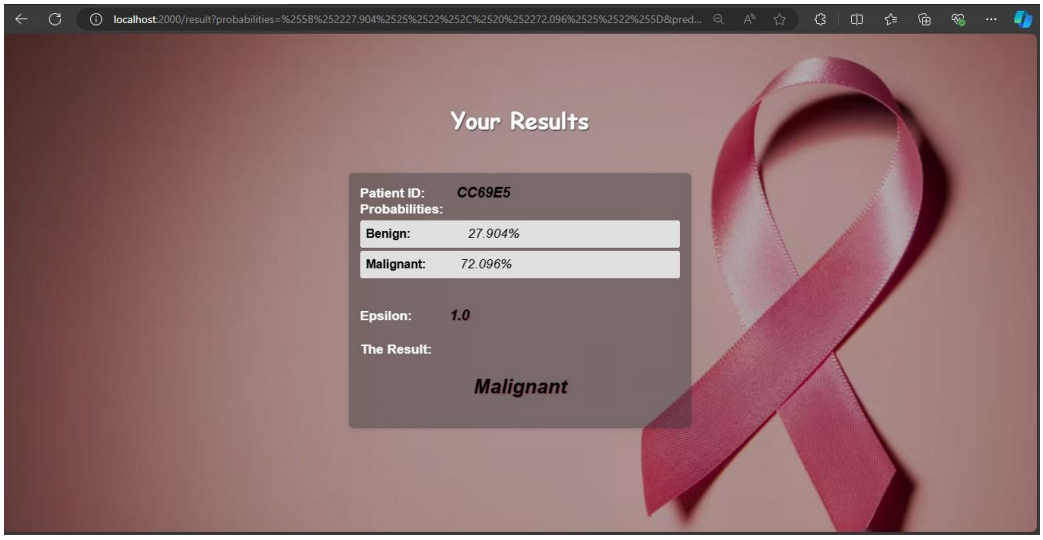
CVS way:



Finally they get their results . If they did choose the non-private model then sensitive data (name , family name , age), probabilities and the final predection will be displayed in the result page as it shown under:



Else, the application generate an identifier (ID) and display it insted of the sensitive data in addition to probabilities , privacy budjet (epsilon) and the final predection as results :



4.2.5 Conclusion

In this chapter, we explored the performance of deep learning models for breast cancer diagnosis, comparing non-private and differentially private settings. The non-private model achieved a validation accuracy of 97.80% and a test accuracy of 96%, demonstrating its effectiveness in accurately classifying the dataset. The differentially private model, designed to protect user data, achieved a slightly lower validation accuracy of 96.70% and a test accuracy of 80%. This reduction in accuracy was due to the noise added to ensure privacy.
Our findings indicate that differential privacy can protect sensitive patient information while maintaining high model performance. The trade-off between accuracy and privacy was carefully managed by fine-tuning the differential privacy parameters. The use of the **ADAM** optimizer consistently yielded better results compared to **SGD** in both model types. The best hyperparameters for the non-private model included a learning rate of 0.001 and a batch size of 512, whereas th
e private model performed best with a learning rate of 0.01, a batch size of 128, andspecific privacy parameters (δ of 0.0001 and max grad norm of 1.2).

CONCLUSION

CONCLUSION

This document has highlighted the growing importance of deep learning in a wide range of domains, while also underscoring the critical issues of security and privacy that must be addressed. The advancements in these AI technologies are paving the way for many promising applications, but their responsible deployment requires overcoming significant ethical and regulatory challenges.

Protecting the privacy and security of deep learning systems is essential to ensure public trust and enable the sustainable adoption of these technologies for the common good. Continuous research, regulation, and public awareness efforts will be necessary to achieve this goal.

As deep learning becomes more pervasive in sensitive areas such as healthcare, finance, and national security, safeguarding personal data and preventing misuse is of utmost importance. Techniques like homomorphic encryption, secure multi-party computation, and differential privacy must be leveraged to enable collaborative model training without revealing private information.

Beyond the technical aspects, this document has also emphasized the key ethical and regulatory considerations, such as transparency, accountability, and respect for user autonomy. Upholding privacy principles is a crucial element in building public confidence and promoting the responsible and sustainable deployment of these AI technologies.

Moving forward, the deep learning community must remain vigilant and proactive in addressing security and privacy challenges. Only by addressing these fundamental concerns can the transformative potential of deep learning be fully realized in a way that benefits individuals and society as a whole.

Future Perspectives

Future research could focus on several areas to further enhance the balance between privacy and accuracy in breast cancer diagnosis models:

- **Advanced Differential Privacy Techniques:** Investigating more sophisticated differential privacy mechanisms could help reduce the trade-off between accuracy and privacy.
- **Hybrid Models:** Combining different machine learning approaches and optimization techniques may improve model performance while maintaining privacy.
- **Dataset Expansion:** Applying these models to larger and more varied datasets could help generalize the findings and improve model robustness.
- Investigate algorithmic enhancements to address the challenges posed by imbalanced data, focusing on improving fairness in model outcomes.
- Optimizing hyper-parameters and exploring innovative approaches.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Julien, D. (2018, April). Natural solutions: Histoire du deep learning. Retrieved June 28, 2020, from https://www.natural-solutions.eu/blog/histoire-du-deep-learning

[2] Heaton, J. (2016). Artificial intelligence for humans, Volume 3: Deep learning and neural networks.

[3] Schmidhuber, J. (2015). Deep learning. Scholarpedia, 10(11), 32832

[4] Ajit, A., Acharya, K., & Samanta, A. (2020). A review of convolutional neural networks. In 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE).

[5] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial networks. Advances in Neural Information Processing Systems, 3.

[6] Sarker, I. H. (2021). Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science, 2(6), 420.

[7] Yadav, P. (2018, July). Machine learning vs. deep learning. Medium. https://medium.com/@mail2princeyadav/machine-learning-vs-deep-learning-b5c5a4fc5c

[8] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, 1-74.

[9] Van Essen, B., Kim, H., Pearce, R., Boakye, K., & Chen, B. (2015). LBANN: Livermore Big Artificial Neural Network HPC toolkit. In Proceedings of the workshop on machine learning in high-performance computing environments (pp. 1−6).

[13] Rivas-Blanco, I., Pérez-Del-Pulgar, C. J., García-Morales, I., & Muñoz, V. F. (2021). A review on deep learning in minimally invasive surgery. IEEE Access, 9, 48658-48678.

[14] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:1803.01164.

[15] Saeed, M. M., Al Aghbari, Z., & Alsharidah, M. (2020). Big data clustering techniques based on spark: a literature review. PeerJ Computer Science, 6, e321.

[17] Martí-Juan, G., Sanroma-Guell, G., & Piella, G. (2020). A survey on machine and statistical learning for longitudinal analysis of neuroimaging data in Alzheimer's disease. Computer methods and programs in biomedicine, 189, 105348.

[23] Khan, A., Sohail, A., Zahoora, U., & Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. Artificial intelligence review, 53, 5455-5516.

[24] Indolia, S., Goswami, A. K., Mishra, S. P., & Asopa, P. (2018). Conceptual understanding of convolutional neural network-a deep learning approach. Procedia computer science, 132, 679-688.

[26] Yani, M., Aprillianto, F., Sigit, D. V., Purwanto, E., & Farizqi, A. S. (2019). Application of transfer learning using convolutional neural network method for early detection of Terry's nail. Journal of Physics: Conference Series, 1201(1), 012052. https://doi.org/10.1088/1742-6596/1201/1/012052

[28] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324. https://doi.org/10.1109/5.726791

[29] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems (Vol. 25, pp. 1097-1105)

[30] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representations (ICLR)

[31] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2014). Going deeper with convolutions. arXiv preprint arXiv:1409.4842. https://arxiv.org/abs/1409.4842

[32] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385. https://arxiv.org/abs/1512.03385

[33] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[35] Amidi, A., & Amidi, S. (2018). Vip cheatsheet: Recurrent neural networks.

[36] Sundermeyer, M., Schlüter, R., & Ney, H. (2012). LSTM neural networks for language modeling. In INTERSPEECH 2012: ISCA's 13th Annual Conference (pp. 601-604), Portland, OR, USA, September 9-13.

[39] Musolf, A. M., Holzinger, E. R., Malley, J. D., & Bailey-Wilson, J. E. (2022). What makes a good prediction? Feature importance and beginning to open the black box of machine learning in genetics. Human Genetics, 141(9), 1515-1528.

[42] https://www.talend.com/resources/data-privacy/

[43] https://www.techtarget.com/searchcio/definition/data-privacy-information-privacy

[47] Anonymous. (2024). Cases from Deep Learning where security and privacy issues come up in different ways. Note: Demonstrates the different ways privacy and security issues arise in DL applications, including medical imaging, financial transactions, personal assistant systems, autonomous vehicles, malware detection, and biometric authentication.

[48] Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017, May). Membership inference attacks against machine learning models. In 2017 IEEE symposium on security and privacy (SP) (pp. 3-18). IEEE

[52] Tran, A. T., Luong, T. D., & Huynh, V. N. (2024). A comprehensive survey and taxonomy on privacy-preserving deep learning. Neurocomputing, 127345.

[53] Al-Rubaie, M., & Chang, J. M. (2019). Privacy-preserving machine learning: Threats and solutions. IEEE Security & Privacy, 17(2), 49-58.

[54] Garfinkel, S., Abowd, J. M., & Martindale, C. (2019). Understanding database reconstruction attacks on public data. Communications of the ACM, 62(3), 46-53.

[55] Kasichainula, K., Mansourifar, H., & Shi, W. (2020). Privacy Preserving Proxy for Machine Learning as a Service. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 4006). https://doi.org/10.1109/BigData50022.2020.9378377

[56] Yang, Z., Shao, B., Xuan, B., Chang, E. C., & Zhang, F. (2020). Defending model inversion and membership inference attacks via prediction purification. arXiv preprint arXiv:2005.03915

[57] Mireshghallah, F., Taram, M., Jalali, A., and Gupta, H. (2020). Privacy in Deep Learning: A Survey. arXiv preprint arXiv:2004.12254.

[58] Shi, Y., Davaslioglu, K., & Sagduyu, Y. E. (2020, July). Over-the-air membership inference attacks as privacy threats for deep learning-based wireless signal classifiers. In Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning (pp. 61-66).

[60] Punitha, N., & Amsaveni, R. (2011). Methods and techniques to protect the privacy information in privacy preservation data mining. IJCTA| NOV-DEC.

[61] Li, N., Li, T., & Venkatasubramanian, S. (2006, April). t-closeness: Privacy beyond k-anonymity and l-diversity. In 2007 IEEE 23rd international conference on data engineering (pp. 106-115). IEEE.

[62] Rajendran, K., Jayabalan, M., & Rana, M. E. (2017). A study on k-anonymity, l-diversity, and t-closeness techniques. IJCSNS, 17(12), 172.

[63] El Ouadrhiri, A., & Abdelhadi, A. (2022). Differential privacy for deep and federated learning: A survey. IEEE access, 10, 22359-22380

[64] Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., ... & Zhao, S. (2021). Advances and open problems in federated learning. Foundations and trends® in machine learning, 14(1−2), 1-210.

[65] Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 10(2), 1-19.

[67] Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing (pp. 169-178).

[68] Boneh, D., Goh, E. J., & Nissim, K. (2005). Evaluating 2-DNF formulas on ciphertexts. In Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005. Proceedings 2 (pp. 325-341). Springer Berlin Heidelberg.

[69] Gilad-Bachrach, R., Dowlin, N., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2016, June). Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In International conference on machine learning (pp. 201-210). PMLR.

[70] Makri, E., Rotaru, D., Smart, N. P., & Vercauteren, F. (2019). EPIC: efficient private image classification (or: Learning from the masters). In Topics in Cryptology−CT-RSA 2019: The Cryptographers' Track at the RSA Conference 2019, San Francisco, CA, USA, March 4−8, 2019, Proceedings (pp. 473-492). Springer International Publishing.

[71] Yao, A. C. (1982, November). Protocols for secure computations. In 23rd annual symposium on foundations of computer science (sfcs 1982) (pp. 160-164). IEEE.

[72] Narayanan, A., & Shmatikov, V. (2008, May). Robust de-anonymization of large sparse datasets. In 2008 IEEE Symposium on Security and Privacy (sp 2008) (pp. 111-125). IEEE.

[73] Trung Ha, Tran Khanh Dang, Hieu Le, & Tuan Anh Truong. (2020). Security and privacy issues in deep learning: A brief review. SN Computer Science, 1(253). https://doi.org/10.1007/s42979-020-00254-4

[74] Zhu, K., Van Hentenryck, P., & Fioretto, F. (2021, May). Bias and variance of post-processing in differential privacy. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 12, pp. 11177-11184).

[75] Nanayakkara, P., Bater, J., He, X., Hullman, J., & Rogers, J. (2022). Visualizing privacy-utility trade-offs in differentially private data releases. arXiv preprint arXiv:2201.05964.

[76] Aitsam, M. (2022, December). Differential privacy made easy. In 2022 International Conference on Emerging Trends in Electrical, Control, and Telecommunication Engineering (ETECTE) (pp. 1-7). IEEE.

[91] Redberg, R., Koskela, A., & Wang, Y. X. (2024). Improving the privacy and practicality of objective perturbation for differentially private linear learners. Advances in Neural Information Processing Systems, 36.

[105] Bae, H., Jang, J., Jung, D., Jang, H., Ha, H., Lee, H., & Yoon, S. (Year). Security and Privacy Issues in Deep Learning. IEEE Transactions on Artificial Intelligence, Vol. 00, No. 0, Month 2020, pp. 1.

[109] W. Wolberg and O. Mangasarian. “Multisurface method of pattern separation for medical diagnosis applied to breast cytology”. In: Proceedings of the National Academy of Sciences of the United States of America 87.1 (1991), pp. 9193−9196.

[110] W. T. Mohammad et al. “Diagnosis of Breast Cancer Pathology on the Wisconsin Dataset with the Help of Data Mining Classification and Clustering Techniques”. In: Applied Bionics and Biomechanics 2022 (2022). Retraction published Appl Bionics Biomech. 2023 Nov 29;2023:9759080. doi: 10.1155/2023/9759080, p. 6187275. doi: 10.1155/2022/6187275.

WEB REFERENCES

WEB REFERENCES

[10] https://www.baeldung.com/cs/neural-networks-neurons

[11] https://www.mathworks.com/help/deeplearning/ug/speed-up-deep-neural-network-training.html

[12] https://www.solver.com/training-artificial-neural-network-intro

[16] https://oden.io/glossary/model-training/

[18] https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/train-deep-learning-model.htm

[20] https://klu.ai/glossary/accuracy-precision-recall-f1

[25] https://www.mathworks.com/discovery/deep-learning.html

[27] https://www.linkedin.com/pulse/fitting-parameters-convolutional-neural-network-class-gil-gutierrez

[34] https://medium.com/@Packt_Pub/inside-the-generative-adversarial-networks-gan-architecture-2435afbd6b3b

[37] https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

[38] https://marutitech.com/top-8-deep-learning-frameworks/

[40] https://www.v7labs.com/blog/overfitting#h4

[41] https://www.linkedin.com/pulse/strategies-mitigate-overfitting-deep-learning-abdullah-al-rahman

[44] Tran, A. T., Luong, T. D., & Huynh, V. N. (2024). A comprehensive survey and taxonomy on privacy-preserving deep learning. Neurocomputing, 127345.

[45] Blueway. (2021). *Ensure Data Integrity: Definition, Issues, Risks, and Best Practices*. Retrieved from https://www.blueway.fr/en/blog/data-integrity

[46] National Academies. (n.d.). *Protecting Privacy and Confidentiality While Providing Access to Data for Research Use*. Retrieved from https://nap.nationalacademies.org/read/24652/chapter/7

[49] Cloudian. (2024). *What is Data Protection and Privacy?*. Retrieved from https://cloudian.com/guides/data-protection/data-protection-and-privacy-7-ways-to-protect-user-data/

[50] European Partnership for Personalised Medicine - EP PerMed. (2024). *Data privacy*. Retrieved from https://www.eppermed.eu/data-privacy/

[51] https://www.onlyoffice.com/blog/2024/01/data-privacy-day

[59] Author. (n.d.). Privacy-Preserving Techniques in Deep Learning. Retrieved from URL

[66] https://www.geeksforgeeks.org/collaborative-learning-federated-learning/

[68] https://www.spiceworks.com/it-security/data-security/articles/how-homomorphic-encryption-protects-data/

[77] Wikipedia contributors. (n.d.). Differential Privacy. In Wikipedia, The Free Encyclopedia. Retrieved from https://en.wikipedia.org/wiki/Differential_privacy

[78] (n.d.). Differential Privacy. In SpringerLink. Retrieved from https://link.springer.com/referenceworkentry/10.1007/978-1-4419-5906-5_752

[79] (n.d.). What is Differential Privacy? Definition, Mechanisms, Examples. In Statice. Retrieved from https://www.statice.ai/post/what-is-differential-privacy-definition-mechanisms-examples

[80] IEEE Digital Privacy. (n.d.). What is Differential Privacy?. Retrieved from https://digitalprivacy.ieee.org/publications/topics/what-is-differential-privacy

[81] (n.d.). Understanding Differential Privacy. In Towards Data Science. Retrieved from https://towardsdatascience.com/understanding-differential-privacy-85ce191e198a

[82] https://www.semanticscholar.org/paper/Differential-Privacy-in-Deep-Learning%3A-An-Overview-Ha-Dang/b386bf52898398e68560d295b5b0335dd443b701

[85] //en.wikipedia.org/wiki/Differential_privacy

[86] //ealizadeh.com/blog/abc-of-differential-privacy/

[87] //towardsdatascience.com/understanding-differential-privacy-85ce191e198a

[88] https://deepgram.com/ai-glossary/differential-privacy

[89] https://towardsdatascience.com/abcs-of-differential-privacy-8dc709a3a6b3
[90] https://programming-dp.com/ch8.html

[92] https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/46029.pdf

[99]. [arxiv.org - Auto DP-SGD: Dual Improvements of Privacy and Accuracy ...](https://arxiv.org/html/2312.02400v1)

[100]. [researchgate.net - Differential Privacy in Federated Dynamic Gradient ...](https://www.researchgate.net/publication/378509478_Differential_Privacy_in_Federated_Dynamic_Gradient_Clipping_Based_on_Gradient_Norm)

[101]. [github.com - thecml/dpsgd-optimizer](https://github.com/thecml/dpsgd-optimizer)

[102]. [researchgate.net - DPDR: Gradient Decomposition and Reconstruction ...](https://www.researchgate.net/publication/381190515_DPDR_Gradient_Decomposition_and_Reconstruction_for_Differentially_Private_Deep_Learning)

[103]. [onlinelibrary.wiley.com - Understanding adaptive gradient clipping in DP- SGD](https://onlinelibrary.wiley.com/doi/10.1002/int.23001)

[104]. [medium.com - Differential Privacy Series Part 1 | DP-SGD Algorithm ...](https://medium.com/pytorch/differential-privacy-series-part-1-dp-sgd-algorithm-explained-12512c3959a3)

[106] *Breast Cancer Wisconsin (Diagnostic) Dataset*. https://www.geeksforgeeks.org/ breast-cancer-wisconsin-diagnostic-dataset/.

[107] *Breast Cancer Wisconsin Diagnostics*. https://jovian.com/mitchell- odili/ breast-cancer-wisconsin-diagnostics.

[108] *UCI Machine Learning Repository: Breast Cancer Wisconsin (Diagnostic) Data Set*. https : / / archive . ics . uci . edu / ml / datasets / Breast + Cancer + Wisconsin + (Diagnosti

Abstract

In this thesis, privacy-preserving techniques, specifically differential privacy, are examined in the context of sensitive health information using breast cancer data as a case study.
The discussion begins with an overview of deep learning and its application in contemporary scenarios. The core focus of the thesis is on addressing privacy concerns associated with the use of healthcare data in machine learning models. It investigates how the mathematical framework of differential privacy, which quantifies and limits privacy risks, can enable the use of breast cancer data for training deep learning models without compromising patient confidentiality.
The thesis provides a detailed explanation of the implementation of differential privacy, incorporating various techniques and formulas. It also presents an analysis of the trade-off between utility and privacy, and evaluates the accuracy of privacy-preserving deep learning models for breast cancer prediction tasks.
The goal is to contribute to the existing body of knowledge on privacy-preserving deep learning, particularly in the ethical use of health data.

Résumé

Dans cette thèse, les techniques de préservation de la vie privée, en particulier la confidentialité différentielle, sont examinées dans le contexte des informations sensibles sur la santé en utilisant les données sur le cancer du sein comme étude de cas.
La discussion commence par un aperçu de l'apprentissage profond et de son application dans des scénarios contemporains. L'objectif principal de la thèse est d'aborder les problèmes de confidentialité associés à l'utilisation des données de santé dans les modèles d'apprentissage automatique. Elle étudie comment le cadre mathématique de la confidentialité différentielle, qui quantifie et limite les risques pour la vie privée, peut permettre l'utilisation de données sur le cancer du sein pour la formation de modèles d'apprentissage profond sans compromettre la confidentialité des patients.
La thèse fournit une explication détaillée de la mise en œuvre de la confidentialité différentielle, en incorporant diverses techniques et formules. Elle présente également une analyse du compromis entre l'utilité et la confidentialité, et évalue la précision des modèles d'apprentissage profond préservant la confidentialité pour les tâches de prédiction du cancer du sein.
L'objectif est de contribuer à l'ensemble des connaissances existantes sur l'apprentissage profond préservant la vie privée, en particulier en ce qui concerne l'utilisation éthique des données de santé.

ملخص

في هذه الأطروحة، يتم فحص تقنيات الحفاظ على الخصوصية، وتحديداً الخصوصية التفاضلية، في سياق المعلومات الصحية الحساسة باستخدام بيانات سرطان الثدي كدراسة حالة.
تبدأ المناقشة بلمحة عامة عن التعلم العميق وتطبيقه في السيناريوهات المعاصرة. ينصب التركيز الأساسي للأطروحة على معالجة مخاوف الخصوصية المرتبطة باستخدام بيانات الرعاية الصحية في نماذج التعلم الآلي. وتبحث الأطروحة كيف يمكن للإطار الرياضي للخصوصية التفاضلية، الذي يحدد مخاطر الخصوصية ويحد منها، أن يتيح استخدام بيانات سرطان الثدي لتدريب نماذج التعلم العميق دون المساس بسرية المرضى.
تقدم الأطروحة شرحًا تفصيليًا لتطبيق الخصوصية التفاضلية، مع دمج تقنيات وصيغ مختلفة. كما تقدم تحليلاً للمفاضلة بين المنفعة والخصوصية، وتقيّم دقة نماذج التعلم العميق التي تحافظ على الخصوصية لمهام التنبؤ بسرطان الثدي.
الهدف هو الإسهام في مجموعة المعارف الحالية حول التعلم العميق الذي يحافظ على الخصوصية، لا سيما في الاستخدام الأخلاقي للبيانات الصحية.