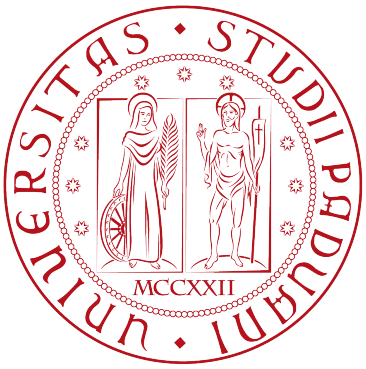

Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Master Degree in ICT for Internet and Multimedia

Differential Privacy Techniques in Federated Learning: Application to Diabetic Retinopathy Image Processing

Supervisor:
Prof.ssa Federica Battisti

Co-Supervisor:
Prof. Luis Alberto da Silva Cruz

Candidate:
Mahsa Shahbazi
2049588

Academic Year 2023-2024

Abstract

The advent of federated learning has opened new frontiers in the privacy-preserving analysis of medical data, enabling collaborative model training without direct data sharing. This is particularly critical in the realm of healthcare, where patient confidentiality and data protection are paramount. Traditional data anonymization techniques are often insufficient to protect privacy against sophisticated attacks that can re-identify individuals from anonymized datasets. Therefore, federated learning is needed as it allows model training on decentralized data, mitigating the risk of data leakage.

This thesis explores the integration of *Differential Privacy* (DP) techniques into federated learning frameworks, focusing on the application to *Diabetic Retinopathy* (DR) image processing, a critical area in medical diagnostics where the early detection and classification of disease stages can significantly impact patient outcomes.

I present a study comparing four models: centralized non-private machine learning and non-private federated learning as baseline models, alongside two differentially private federated learning models utilizing the Gaussian and Laplace mechanisms. My goal is to establish a trade-off between model utility and privacy preservation, which is crucial for deploying machine learning models in sensitive domains. For the differentially private models, I identify the optimal noise values for both the Gaussian and Laplace mechanisms that offer the best balance between accuracy and privacy.

Additionally, I undertake a critical evaluation of the system's security through the simulation of an inversion attack, which tests the robustness of the DP-enhanced federated learning models against potential attempts to reconstruct individual data points from aggregated data. This simulation considers a worst-case scenario where the attacker has high-level access, providing insights into how added noise affects the reconstructed images.

Experimental results demonstrate that the DP-enhanced federated learning models I developed achieve competitive accuracy in classifying diabetic retinopathy images while ensuring better privacy guarantees and resilience against inversion attacks. The results show that by

adding noise, the reconstructed images become less informative, yet the accuracy trade-offs remain relatively close to those of the baseline models. This research contributes to the field by providing empirical evidence of the feasibility of deploying differential privacy in federated learning for medical image analysis, suggesting that privacy-preserving federated learning can be both practical and effective, balancing the need for data security with the imperative of maintaining high-quality medical diagnostics.

Keywords: Federated Learning, Differential Privacy, Diabetic Retinopathy, Medical Image Processing, Gaussian Mechanism, Laplace Mechanism, Inversion Attack, Data Anonymization.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Professor Federica Battisti at the University of Padova and Professor Luis Alberto da Silva Cruz at the University of Coimbra, for their invaluable guidance, support, and expertise throughout the course of this research. Their wisdom, encouragement, and insightful critiques have been instrumental in shaping this thesis.

My time as an Erasmus student at the University of Coimbra has been a pivotal period in my academic journey. I am especially thankful to Professor Cruz for welcoming me into his department and providing me with the opportunity to work in an environment that was both challenging and nurturing.

I am equally grateful to Professor Battisti for her constant support and constructive feedback, which were crucial in refining my research objectives and methodologies. Her commitment to academic excellence has been a constant source of inspiration.

I would also like to extend my appreciation to the staff and faculty members of both the University of Padova and the University of Coimbra. Their assistance and support have been invaluable.

A special thank you goes to my peers and colleagues, who have contributed to my personal and professional growth during this journey. Their camaraderie and insights have been a source of strength and encouragement.

Lastly, I want to thank my family and friends for their unwavering support and belief in me. Their love and encouragement have been my anchor throughout this challenging yet fulfilling endeavor.

This thesis would not have been possible without the collective support and encouragement of all these individuals. I am deeply grateful to each one of them.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to Federated Learning in Healthcare

Federated Learning is a paradigm change in machine learning where localized data samples in decentralized devices or on servers can communicate to train algorithms without necessarily exchanging the data itself. This kind of decentralized model training is valuable in situations requiring data privacy, security, and governance. Unlike traditional centralized machine learning, which aggregates data in a central server, FL enables model training directly on user devices while aggregating updates of the model without aggregating the raw data itself. This preserves users' privacy, reduces latency, and enhances scalability.

This capability is critical in the area of medical image analysis, where patient privacy is paramount. For instance, diabetic retinopathy (DR) diagnosis relies on sensitive retinal images, so the ability to train models without sharing raw medical data mitigates privacy concerns. Thus, FL is an ideal solution in healthcare applications.

However, despite its clear benefits, federated learning faces significant challenges, particularly in privacy-sensitive fields like healthcare. Issues like data variability, privacy risks, and difficulties in model convergence can emerge in this context.

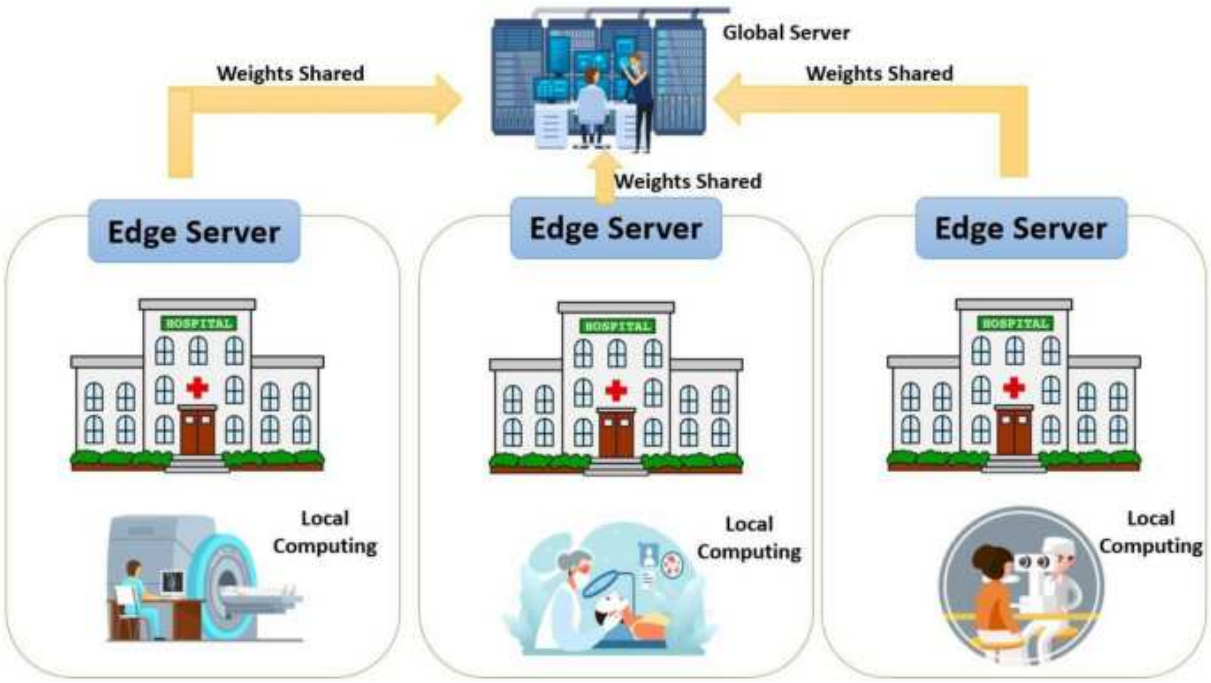

Figure 1.1: Federated Learning System Architecture in Healthcare

Figure 1.1 illustrates the federated learning system architecture in a healthcare setting. In this setup, multiple hospitals (represented as edge servers) perform local computing on medical data to train a model without sharing the raw data with a central server. Instead, only model weights are shared between the edge servers and a global server, which aggregates these updates to refine the model centrally. This architecture enhances patient data privacy by keeping sensitive information localized while still enabling collaborative learning across institutions.

1.2 Motivation and Related Work

This thesis is inspired by foundational research, particularly a prior dissertation titled *Ophthalmology Applications of Federated Learning*, conducted by a student at the University of Coimbra. That work developed a FL system aimed at diabetic retinopathy detection, demonstrating the potential of FL in decentralized medical diagnostics. Specifically, it explored and compared three aggregation algorithms—Federated Averaging, Federated Proximal, and *stochastic controlled averaging for federated learning* (SCAFFOLD)—to provide insights into model convergence and diagnostic accuracy. However, the research did not address implementing privacy-preserving mechanisms or analyzing privacy-utility trade-offs, nor did it account for security threats such as inversion attacks that attempt to reconstruct sensitive data from aggregated model updates.

Building on this conceptual foundation, my thesis shifts focus to privacy enhancement within FL systems, specifically through DP mechanisms. Unlike the previous study, this research does not reuse any specific settings, models, or code from that work; instead, it aims to introduce and evaluate DP techniques within a similar application context. By embedding DP mechanisms into the federated learning framework, this study aims to strengthen data protection in diabetic retinopathy detection without compromising diagnostic accuracy. A particular focus is given to understanding the balance between privacy and utility by testing these methods against simulated inversion attacks—an approach intended to assess the resilience of DP-enhanced FL systems against data reconstruction threats.

The direction for this study is further informed by the work *Dopamine: Differentially Private Federated Learning on Medical Data* [1], which proposes a privacy-preserving FL approach using *Differentially Private Stochastic Gradient Descent* (DP-SGD) with the Gaussian mechanism, supported by secure multi-party aggregation. Dopamine has shown notable success in achieving privacy guarantees and model accuracy. Building on Dopamine's demonstrated effectiveness, this thesis applies a similar framework, integrating the Gaussian mechanism but also introducing the Laplace mechanism to allow a comparative analysis. Rather than employing secure multi-party aggregation or homomorphic encryption, this study takes a simplified approach, focusing on contrasting the Laplace and Gaussian mechanisms in terms of privacy-accuracy trade-offs.

To further validate the robustness of these privacy mechanisms, this research conducts simulations of inversion attacks, analyzing how well differentially private FL systems withstand data reconstruction threats. By examining the strength of these DP mechanisms in safeguarding sensitive medical information, this study seeks to extend the previous research in a direction that emphasizes security and privacy resilience in federated learning for healthcare applications.

1.3 Federated Learning: Key Benefits and Challenges

Federated Learning offers a promising approach to machine learning in privacy-sensitive and distributed environments, especially in fields like healthcare, finance, and education. However, FL also presents distinct challenges, particularly around data heterogeneity, model convergence, and balancing privacy with model performance. In the next two sections, I explore both the benefits and limitations of FL.

1.3.1 Advantages of Federated Learning

- **Privacy Preservation:** FL minimizes the need to share raw data by keeping it localized on each device, reducing risks associated with data breaches and meeting regulatory standards. This decentralized approach is especially important in sensitive fields like healthcare, where privacy protection is paramount.

- **Reduced Latency and Communication Costs:** With local training on edge devices, FL reduces the frequency and volume of data transmission to a central server, resulting in lower latency and communication costs. Only model updates, not raw data, are shared, which makes FL a cost-effective option for devices with limited bandwidth or power.

- **Scalability and Flexibility:** FL enables large-scale deployment across a variety of devices, from mobile phones to *Internet of Things* (IoT) systems, accommodating distributed environments and allowing diverse organizations to collaboratively train models. This scalability is essential for processing vast amounts of decentralized data across heterogeneous sources.

- **Enhanced Personalization:** By allowing model updates based on local data, FL supports personalized models that better capture individual user patterns and preferences without compromising privacy. This advantage is valuable for applications like personalized healthcare, where individualization enhances clinical relevance.

- **Data Diversity and Robustness:** FL can leverage data from diverse sources, including different geographies, institutions, and demographic groups. This diversity can improve model robustness and generalizability, as the model can learn from a wide range of real-world data without requiring centralized storage.

- **Compliance with Data Sovereignty and Localization Laws:** FL complies with data sovereignty regulations by keeping data local while enabling collaborative model development, making it feasible to work across institutions and countries with strict data governance requirements (e.g., *General Data Protection Regulation* (GDPR), *Health Insurance Portability and Accountability Act* (HIPAA)).

- **Improved Fault Tolerance:** FL's decentralized structure allows for greater fault tolerance. If certain devices drop out or experience connectivity issues, the model can continue training with the updates from remaining devices, supporting applications on devices with intermittent connectivity.

- **Enhanced Security Through Decentralized Training:** By avoiding centralized storage, FL reduces the risk of large-scale data breaches. Even if a device is compromised, the data remains segmented, making it more challenging for a malicious actor to access a complete dataset.
- **Continuous Learning and Adaptability:** FL supports on-device learning, allowing models to be updated in real-time as new data becomes available. This adaptability is beneficial in healthcare, where models can adjust to new patient data, improving accuracy over time.
- **Preservation of Data Context and Local Knowledge:** Since FL allows data to remain within its originating environment, it preserves the local context, which can be valuable for applications requiring location-specific insights, such as regional variations in medical conditions.

1.3.2 Challenges in Federated Learning

- **Data Heterogeneity and *Non-Independent and Identically Distributed* (non-IID) Data:** In FL, data is often decentralized and varies significantly between devices, meaning it is not independently and identically distributed (non-IID). This data heterogeneity can lead to difficulties in training, as local updates may diverge due to the differences in data distributions across clients. Consequently, models trained through FL may struggle to generalize effectively, especially in scenarios like healthcare, where patient data may vary widely based on demographic and geographic factors.
- **Model Convergence Issues:** Ensuring efficient convergence in FL is challenging due to the asynchronous nature of updates from multiple devices. With heterogeneous data and varying computational resources, client devices may introduce noise or bias in the aggregated updates, potentially slowing down convergence and affecting overall model performance. This issue becomes more pronounced in large, distributed networks where devices may have irregular participation.
- **Communication Overheads and Limited Resources on Edge Devices:** While FL reduces the need for raw data transmission, it still requires frequent communication of model parameters, which can be costly in terms of bandwidth, especially for devices with limited connectivity. Moreover, edge devices used in FL often have constrained computational resources and battery life, limiting the complexity of models that can be

trained locally. These constraints can hinder the application of FL in environments with limited or intermittent connectivity, reducing the potential benefits of scalability.

- **Privacy-Utility Trade-Offs with Differential Privacy Mechanisms:** Although FL aims to protect privacy, adding DP mechanisms introduces a trade-off between privacy and model utility. DP mechanisms, like the Gaussian and Laplace noise injections, can degrade model accuracy, especially when stringent privacy budgets are applied. Finding an optimal balance between preserving privacy and maintaining sufficient model performance remains a key challenge, particularly in sensitive fields like healthcare where high accuracy is critical.

- **Vulnerability to Adversarial Attacks and Malicious Clients:** FL can be susceptible to adversarial attacks, including model poisoning and inference attacks. Malicious clients can manipulate model updates, degrading the model's performance, or attempt inversion attacks to infer sensitive information from model parameters. Although differential privacy can mitigate some risks, FL requires additional security mechanisms, such as robust aggregation techniques, to protect against adversarial behavior. These extra safeguards can increase computational demands and complexity.

- **Lack of Standardization and Interoperability:** The deployment of FL across diverse devices and systems faces challenges due to a lack of standardized protocols and interoperability. Variations in device specifications, data storage formats, and privacy regulations across jurisdictions can complicate model training and integration. In healthcare, for instance, the diverse range of *Electronic Health Record* (EHR) systems across institutions can hinder collaborative model training, making cross-institutional FL implementations challenging.

1.4 Introduction to Differential Privacy

Differential Privacy is a framework that ensures privacy during the analysis of sensitive data by adding controlled noise to query outputs. This approach is particularly relevant in healthcare, where patient data must remain confidential while allowing valuable insights to be derived.

DP aims to minimize the impact of any individual's data on analysis results, achieved by adding noise proportional to the sensitivity of the query function [2]. This ensures that any single data point has minimal influence on the output, thereby protecting privacy [3]. DP is valuable in federated learning, where aggregated model updates, rather than raw data, are

shared, adding a privacy layer against data reconstruction attacks. In FL, two mechanisms commonly used to implement DP are the Laplace and Gaussian mechanisms, each providing specific trade-offs between privacy and accuracy.

Mathematically, a randomized algorithm $\mathcal{A}$ satisfies ϵ-differential privacy if, for any two datasets D_1 and D_2 differing by one record, and any subset of outcomes S:

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(D_2) \in S],$$

where ϵ is the privacy budget [4]. Smaller ϵ values indicate stronger privacy.

The Laplace mechanism, typically used for low-dimensional data, and the Gaussian mechanism, suited for high-dimensional tasks, align with the research's objectives in handling medical data. As privacy concerns grow, DP provides a robust solution for safeguarding individual privacy while retaining the utility of the dataset.

1.5 Differential Privacy Mechanisms

Differential Privacy is achieved through mechanisms that add noise to data, with the Laplace and Gaussian mechanisms being the most commonly used. These mechanisms vary in how they introduce noise and in the trade-offs they offer between privacy protection and data utility.

1.5.1 Laplace and Gaussian Mechanisms for Differential Privacy

The Laplace and Gaussian mechanisms are fundamental methods for achieving Differential Privacy (DP) by adding noise to query outputs. The choice between these mechanisms depends on data dimensionality, sensitivity, and the required trade-off between privacy and accuracy. In this thesis, both mechanisms are explored and compared to assess their respective impacts on privacy and utility in the specific context of diabetic retinopathy image processing in federated learning. In the next two sections, we will see how they work.

1.5.1.1 Laplace Mechanism:

The Laplace mechanism adds noise drawn from a Laplace distribution, making it particularly suitable for low-dimensional queries, such as sums or averages. The *probability density function* (PDF) for the Laplace distribution is:

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

where $\beta = \frac{\Delta f}{\epsilon}$, with Δf representing query sensitivity and ϵ being the privacy budget. This mechanism effectively maintains privacy without significantly distorting results for low-dimensional data [2]. Although the Laplace mechanism is generally more effective for low-dimensional data, it is included in this thesis to evaluate its performance and utility on high-dimensional diabetic retinopathy images, providing insights into its applicability in more complex data contexts.

1.5.1.2 Gaussian Mechanism:

The Gaussian mechanism introduces noise from a Gaussian distribution, which is more suited to high-dimensional tasks, such as machine learning, where the Gaussian distribution's properties provide a balance between privacy and utility [5]. Its PDF is:

$$f(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right),$$

where $\sigma = \frac{\Delta f \sqrt{2\ln(1.25/\delta)}}{\epsilon}$, and δ represents the failure probability. This mechanism is ideal for complex, high-dimensional data, as it introduces less distortion in such contexts [6]. Given that diabetic retinopathy images are high-dimensional, the Gaussian mechanism is expected to be better suited to this thesis, as it can potentially preserve the data utility necessary for accurate classification and diagnosis while maintaining privacy.

1.5.1.3 Rationale for Comparing Both Mechanisms:

Comparing the Laplace and Gaussian mechanisms in this thesis provides valuable insights into the privacy-utility trade-offs specific to federated learning in healthcare, where high data utility and strict privacy requirements coexist. While the Gaussian mechanism is often preferred for high-dimensional data, assessing the Laplace mechanism's performance on complex, image-based data may reveal its viability in scenarios where privacy requirements are particularly stringent.

1. **Privacy-Utility Trade-Offs**: This comparison allows us to understand how each mechanism impacts model performance in terms of classification accuracy and privacy preservation in diabetic retinopathy detection. The Laplace mechanism could provide adequate privacy with manageable accuracy loss, which would be advantageous in tightly controlled healthcare

settings.

2. **Evaluating Suitability for Medical Image Processing**: Since medical image analysis, especially for sensitive diagnoses like diabetic retinopathy, demands high accuracy, the Gaussian mechanism is anticipated to introduce less noise distortion in high-dimensional data, preserving model utility. However, the Laplace mechanism's impact on these image-based models is evaluated to explore its feasibility in complex tasks, allowing us to make informed recommendations for privacy mechanisms in federated medical applications.

3. **Implications for Federated Learning in Healthcare**: By comparing these mechanisms, this thesis aims to contribute practical insights on tuning differential privacy for federated learning in healthcare. If the Gaussian mechanism provides higher utility with sufficient privacy, it could support more accurate diagnosis models. Conversely, if the Laplace mechanism proves viable, it may offer a simpler privacy implementation in high-dimensional settings, helping healthcare providers balance privacy and model performance effectively.

Figure 1.2 illustrates the probability density functions of the Laplace and Gaussian distributions, both centered at 0 with unit scale. The Laplace distribution (solid green line) has a sharper peak and heavier tails compared to the Gaussian distribution (dashed blue line), which exhibits a smoother, more gradual slope. This visual comparison highlights that the Laplace mechanism tends to concentrate noise more tightly around the mean, whereas the Gaussian mechanism distributes noise more evenly. In the context of differential privacy, these characteristics affect how each mechanism balances privacy and accuracy, especially in high-dimensional data scenarios.

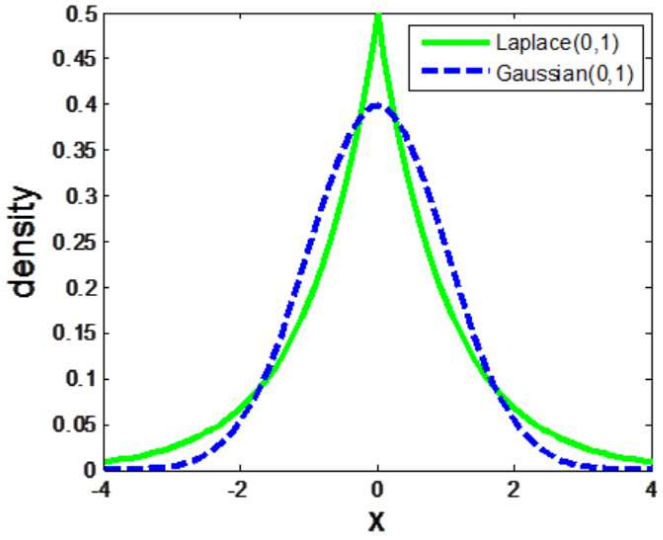

Figure 1.2: Comparison of Laplace and Gaussian Distributions

1.6 Inversion Attacks in Machine Learning

In healthcare, where patient confidentiality is critical, inversion attacks pose a significant privacy risk by potentially exposing sensitive medical data, such as retinal scans used in diagnosing diabetic retinopathy. Unauthorized access to such information can lead to privacy violations, identity theft, and ethical concerns, making data protection paramount. Federated learning mitigates some privacy risks by decentralizing data processing; however, it is not immune to vulnerabilities. Model updates or gradients exchanged during FL training can inadvertently reveal traces of the original data, enabling attackers to reconstruct sensitive information.

This thesis investigates how differential privacy mechanisms, specifically the Gaussian and Laplace mechanisms, can enhance privacy protections in FL by mitigating the risk of inversion attacks while preserving model utility. These DP mechanisms are evaluated for their effectiveness in maintaining both privacy and accuracy within federated learning models for healthcare applications.

Inversion attacks in FL target gradients shared between decentralized devices and a central server. By analyzing these gradients, attackers can reverse-engineer original input data, exploiting the information contained within gradients to reconstruct details of patient medical images or records. Gradient inversion attacks are particularly concerning in healthcare, as they can reveal identifiable patient information from shared model updates, compromising patient privacy despite the decentralized structure of FL.

An example of this vulnerability was demonstrated by Zhu et al. (2019), where researchers successfully reconstructed images from the CIFAR-10 dataset solely using shared gradients, highlighting the susceptibility of FL systems to such attacks [7]. This thesis builds upon these findings, assessing the robustness of DP-augmented FL models against gradient inversion attacks in healthcare.

1.6.1 Metrics for Evaluating Similarity

To evaluate the similarity between the original and reconstructed images after a simulated inversion attack, I utilize a set of quantitative metrics that measure image fidelity and perceptual quality. These metrics provide insight into how much of the original image's detail is retained in the reconstructed image, which directly correlates with the potential privacy leakage. The following metrics are used:

- ***Peak Signal-to-Noise Ratio* (PSNR)**: PSNR measures the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. In the context of image reconstruction, a higher PSNR value generally indicates better similarity to the original image. Lower PSNR values suggest that added noise has effectively degraded the reconstructed image, enhancing privacy.

- ***Structural Similarity Index Measure* (SSIM)**: SSIM assesses the structural similarity between two images, considering luminance, contrast, and structure. SSIM values range from -1 to 1, where values closer to 1 indicate higher structural similarity. For privacy protection, lower SSIM values imply that the reconstructed image has lost more structural details of the original.

- **Multiscale SSIM**: *Multi-Scale Structural Similarity Index Measure* (MS-SSIM) extends SSIM by calculating similarity at multiple scales, which allows for a more comprehensive evaluation of structural information across different levels of image detail. MS-SSIM is particularly useful for detecting differences in high-frequency details, such as edges, which are essential in retinal images. Lower MS-SSIM values indicate better privacy as they suggest reduced structural fidelity.

- ***Feature Similarity Index Measure* (FSIM)**: FSIM is designed to evaluate perceptual image quality by focusing on low-level features, such as phase congruency and gradient magnitude. Higher FSIM values indicate closer resemblance to the original image in terms of essential features. A lower FSIM value indicates that the added noise has effectively reduced feature retention, thereby enhancing privacy.

- **Perceptual Loss (*Visual Geometry Group* (VGG))**: Perceptual loss is computed by passing images through a pre-trained deep neural network (e.g., VGG) and calculating the difference between the feature maps of the original and reconstructed images. Lower perceptual loss values imply greater similarity in high-level features, whereas higher values indicate effective privacy preservation by obscuring detailed patterns.

- ***Gradient Peak Signal-to-Noise Ratio* (G-PSNR)**: Gradient PSNR focuses on preserving edges and fine details by calculating PSNR on the gradient of the images. It is particularly useful for medical images, where edges and structures are essential for diagnosis. Lower G-PSNR values indicate that the noise has degraded edge information in the reconstructed images, improving privacy.

These metrics collectively provide a robust framework for evaluating the quality of reconstructed images and assessing the effectiveness of different privacy-preserving mechanisms. Lower values in PSNR, SSIM, MS-SSIM, FSIM, and G-PSNR, coupled with higher perceptual loss, generally indicate better privacy protection due to reduced similarity between the original and reconstructed images.

1.6.2 Factors Influencing the Effectiveness of Inversion Attacks

The success of inversion attacks depends on several factors that determine how much information can be extracted from a model's outputs or gradients. These factors include:

- **Model Complexity:** Simpler models, or those trained on smaller datasets, may be more vulnerable to inversion attacks as they retain more specific patterns from the training data, making it easier to reverse-engineer the inputs [8].

- **Dimensionality of the Data:** Models trained on high-dimensional data, such as medical images or videos, tend to retain more detailed features of the input data. This makes inversion attacks more successful, as attackers can reconstruct recognizable versions of the original data [7]. For example, in diabetic retinopathy diagnosis, attackers could potentially reconstruct retinal images, revealing sensitive health information.

- **Auxiliary Information:** The availability of auxiliary information, such as public datasets or prior knowledge about the data distribution, can significantly enhance an attacker's ability to reconstruct inputs. For instance, if attackers know the general structure of the data (e.g., the shape of retinal blood vessels), they can guide the inversion process more effectively [9].

Understanding these factors is crucial for developing defenses against inversion attacks. The next section explores different types of inversion attacks and their impact on privacy.

1.7 Types of Inversion Attacks

Inversion attacks in machine learning can be categorized based on their methods and the specific components of the model they target. This section outlines the primary types of inversion attacks, their mechanisms, and their privacy implications.

1.7.1 Model Inversion Attacks

Model inversion attacks infer sensitive features of input data by analyzing the model's output predictions. These attacks exploit the relationship between input features and output predictions to reconstruct partial or approximate representations of data. For example, in facial recognition systems, attackers can use confidence scores to reconstruct images of individuals' faces by iteratively adjusting inputs to achieve a close approximation of the original. Fredrikson et al. (2015) demonstrated the feasibility of this technique in reconstructing facial images from a model trained on a facial recognition dataset, showing how even access to output probabilities can expose private information [8].

This type of attack is particularly concerning in settings where models provide detailed confidence scores or probability distributions, as attackers can iteratively refine inputs to reconstruct sensitive information. Model inversion attacks are especially relevant in centralized learning contexts, where overconfident or poorly generalized models can expose sensitive features.

1.7.2 Training Data Reconstruction

Training data reconstruction attacks aim to recover exact instances from the training dataset, rather than merely approximating features. Unlike model inversion, which infers approximate characteristics, training data reconstruction seeks to retrieve actual samples from the training data. In centralized learning, attackers can probe the model with various inputs to infer details or even reconstruct specific examples from the training dataset, especially if the model is overfitted.

In federated learning, attackers can exploit the gradients shared during training to infer details about the individual samples. Since gradients are computed based on local data, they may reveal sensitive characteristics of the training samples. This approach poses a privacy risk, particularly in federated learning, where gradients are regularly shared between clients and a central server [10].

1.7.3 Gradient Inversion Attacks

Gradient inversion attacks are particularly significant in federated learning environments, where gradients are frequently exchanged between local devices and a central server. By accessing shared gradients, attackers can potentially reconstruct or approximate the original input data, especially when they have additional knowledge of the model structure. Zhao et

al. (2020) demonstrated that detailed images from the CIFAR-10 dataset could be recovered from gradients, highlighting the vulnerability of gradient-based systems [11].

These attacks are more effective when the attacker has knowledge of the model architecture, allowing precise alignment of gradients to reconstruct sensitive features in input data, such as images or text, particularly in high-dimensional tasks like image recognition.

1.7.4 Gradient Matching with Known Model Architecture

In this thesis, I simulate inversion attacks using an advanced technique known as *Gradient Matching with Known Model Architecture*. In this approach, the attacker has access not only to shared gradients but also to a public dataset that resembles the original training dataset. Additionally, the attacker is fully aware of the model architecture used by the clients. This combination enables a more accurate approximation of the original input data by leveraging both the model's structure and the public dataset.

1.7.4.1 Gradient Matching Process

The gradient matching process involves iteratively refining inputs from a similar dataset until their gradients closely align with those observed in the actual inputs. The main steps are as follows:

1. **Initialization with Similar Dataset**: The attacker selects an initial input from a public dataset that closely resembles the original training dataset. This starting point facilitates gradient matching and improves reconstruction accuracy.

2. **Forward and Backward Passes**: Using the known model architecture, the attacker performs a forward pass to calculate the model's output, followed by a backward pass to compute the gradients. These gradients serve as an estimate of the actual gradients.

3. **Gradient Comparison**: The computed gradients of the selected input are compared to the observed gradients using a distance metric, such *Mean Squared Error* (MSE) or cosine similarity. A smaller distance indicates better alignment.

4. **Optimization**: To improve alignment, the attacker adjusts the selected input using gradient descent or another optimization method to minimize the difference between the selected and actual gradients. The optimization is represented as:

$$\mathbf{x}_{\text{chosen}}^{(t+1)} = \mathbf{x}_{\text{chosen}}^{(t)} - \eta \nabla_{\mathbf{x}_{\text{chosen}}} \mathcal{L}(\nabla_{\mathbf{x}_{\text{chosen}}}, \nabla_{\mathbf{x}_{\text{real}}})$$

where η is the learning rate, $\mathbf{x}_{\text{chosen}}^{(t)}$ is the current input from the similar dataset, $\nabla_{\mathbf{x}_{\text{chosen}}}$ and $\nabla_{\mathbf{x}_{\text{real}}}$ represent the gradients of the chosen and real inputs, respectively, and $\mathcal{L}$ is the loss function measuring gradient similarity.

5. **Iterative Alignment**: Steps 2-4 are repeated, refining the input from the public dataset with each iteration until it closely resembles the actual input.

The use of a similar dataset and model architecture enhances gradient matching effectiveness, accelerating the optimization process and enabling highly accurate reconstructions. This approach reflects a realistic threat where an attacker could utilize publicly available data similar to the target data, improving attack feasibility.

In federated learning, where gradients are shared instead of raw data, gradient matching with known model architecture and a similar dataset poses a significant privacy risk. This scenario simulates a worst-case attack, where an attacker intercepts gradients and has access to both model knowledge and a similar public dataset, enabling them to approximate sensitive client data, potentially revealing private information, such as medical images in healthcare

1.8 Diabetic Retinopathy: An Overview

Diabetic Retinopathy is a major complication of diabetes and one of the leading causes of blindness among working-age adults worldwide. It results from prolonged damage to the retinal blood vessels, leading to two primary stages: *Non-Proliferative Diabetic Retinopathy* (NPDR) and *Proliferative Diabetic Retinopathy* (PDR) [12]. While NPDR can progress without symptoms, PDR represents an advanced stage, marked by abnormal blood vessel growth and the risk of retinal detachment, potentially leading to blindness if untreated.

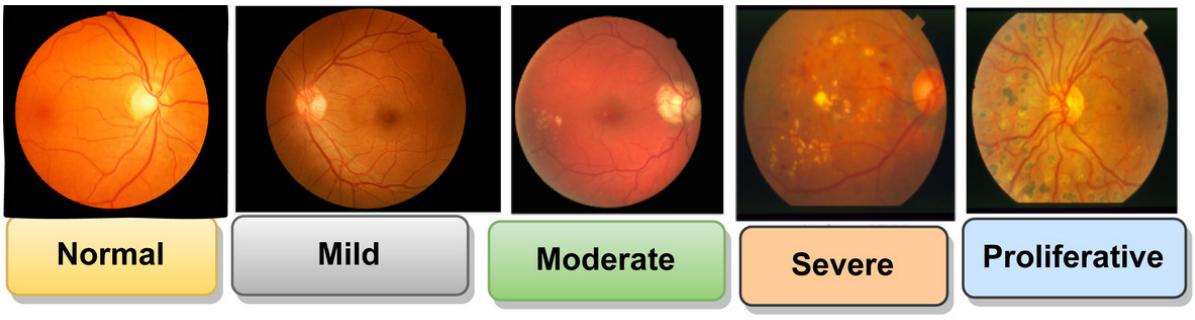

Figure 1.3: Stages of Diabetic Retinopathy

1.8.1 Challenges in Diagnosing Diabetic Retinopathy

Early detection of DR is critical to preventing vision loss. Traditionally, retinal images are captured through fundus photography and manually assessed by ophthalmologists for signs of DR. However, diagnosing DR presents several challenges, particularly in resource-limited regions where access to specialized care is scarce. Moreover, diagnostic variability between practitioners and the subtle nature of early-stage DR, such as the appearance of microaneurysms, make consistent diagnosis difficult [13]. These challenges emphasize the need for scalable, accurate diagnostic tools. Given these challenges, machine learning, especially deep learning, has emerged as a promising tool to automate DR diagnosis while maintaining accuracy and scalability.

1.8.2 The Role of Machine Learning in Diagnosis

Machine learning, especially deep learning, has shown potential in automating the detection of DR through the analysis of retinal images. *Convolutional Neural Network* (CNN)s have demonstrated their ability to detect subtle features, such as microaneurysms and neovascularization, which are essential for accurate DR diagnosis [14]. These models provide consistent, scalable diagnostics, reducing the reliance on human expertise and increasing access to high-quality diagnostic tools, particularly in underserved areas. For example, a study by Gulshan et al. (2016) highlighted the capability of CNNs to match or surpass the performance of ophthalmologists in grading DR severity [15].

1.8.3 Sensitivity of Medical Data and the Importance of Privacy

Retinal images, like other medical data, contain sensitive information and are subject to strict privacy regulations, such as HIPAA in the U.S. and GDPR in Europe. Centralizing such data for machine learning model training introduces significant privacy risks, including potential data breaches. To mitigate these risks, federated learning allows collaborative model training without sharing raw data, reducing privacy concerns.

However, even FL is not entirely immune to privacy threats, as the gradients or model updates shared between institutions can still reveal sensitive information. This is where **Differential Privacy** comes into play. By introducing noise into model updates, DP ensures that sensitive data cannot be reverse-engineered, further enhancing privacy protection. For instance, Abadi et al. (2016) developed DP-SGD, which provides privacy guarantees while maintaining model utility [6].

1.9 Research Synthesis, Gaps, and Contributions

In federated learning systems, balancing privacy and model utility is paramount. FL aims to mitigate privacy risks by decentralizing data processing, but it does not completely eliminate vulnerabilities, as gradients exchanged during training may still reveal sensitive information, such as patient medical images. Inversion attacks, which exploit shared gradients to reconstruct input data, pose a significant threat to privacy in FL systems.

This thesis examines the integration of differential privacy mechanisms, specifically the Gaussian and Laplace mechanisms, within an FL framework designed for diagnosing diabetic retinopathy. The research focuses on evaluating these mechanisms' effectiveness in maintaining privacy without sacrificing diagnostic accuracy, especially when faced with gradient inversion attacks. Gradient matching is used to simulate inversion attacks, providing a practical assessment of the DP-enhanced FL model's robustness against data reconstruction threats.

Key research gaps and contributions identified in this study include:

- **Application to Medical Imaging Data**: Limited empirical research has specifically addressed the vulnerability of FL systems to inversion attacks in medical imaging contexts, such as DR diagnosis. This study contributes by examining the effectiveness of privacy-preserving methods in protecting sensitive medical images.

- **Comparative Analysis of DP Mechanisms in FL**: This thesis compares Gaussian and Laplace DP mechanisms, analyzing the privacy-utility trade-offs to determine optimal noise levels that safeguard privacy while preserving model performance. The study clarifies these trade-offs by quantifying the impact of DP mechanisms on diagnostic accuracy, which is particularly critical in healthcare where accuracy losses can directly affect outcomes.

- **Robustness Against Gradient-Based Inversion Attacks**: There is limited evaluation of DP mechanisms' resilience to gradient inversion attacks within FL settings. This thesis contributes by simulating inversion attacks to test the robustness of DP-enhanced FL models, analyzing how DP-induced noise affects the reconstruction of sensitive data and providing insights for protecting medical images in federated settings.

- **Healthcare-Focused Validation and Privacy-Utility Analysis**: Conducted with a specific focus on DR, this research offers empirical evidence for deploying privacy-preserving FL systems in healthcare, validating the applicability of DP mechanisms in scenarios where both data privacy and diagnostic accuracy are essential.

This thesis advances the field of privacy-preserving machine learning by providing a comprehensive evaluation of DP techniques within FL systems for medical applications. The findings contribute to the development of secure, accurate, and scalable FL solutions suited to privacy-sensitive domains, supporting enhanced protection of healthcare data.

Chapter 2

Related Work

2.1 Advances in Privacy-Preserving Machine Learning

2.1.1 Early Privacy-Preserving Techniques and Limitations

Early methods for privacy protection, such as anonymization, focused on removing or masking identifiable information to prevent linkage to individual identities. However, Narayanan and Shmatikov (2008) demonstrated that anonymized data could often be re-identified through cross-referencing with publicly available sources, exposing significant privacy vulnerabilities in sensitive domains like healthcare [16].

In response to these limitations, cryptographic techniques such as Secure Multi-Party Computation *(MPC)* and Homomorphic Encryption *(HE)* emerged. MPC enables collaborative computations without revealing individual inputs [17], while HE supports computations on encrypted data without exposing plaintext. However, both approaches face computational overhead challenges, making them impractical for large-scale machine learning tasks [2].

These challenges spurred the development of *DP*, which offers formal privacy guarantees with lower computational complexity. DP became a scalable solution by adding controlled noise to outputs, limiting the influence of individual data points on analytical results.

2.1.2 Emergence and Evolution of DP in Machine Learning

Introduced by Dwork et al. in 2006, *DP* provided a mathematical framework to ensure privacy by adding noise proportional to the sensitivity of data queries [4]. DP has been widely adopted in privacy-preserving machine learning due to its balance of privacy guarantees and analytical utility.

A major milestone was the development of *DP-SGD* by Abadi et al. (2016), which incorporates DP into model training by adding noise to gradient updates [6]. This advancement enabled privacy-preserving large-scale machine learning. Subsequent work introduced mechanisms like *Gaussian Noise* for high-dimensional tasks and *Smooth Sensitivity* for non-smooth functions, further enhancing the applicability of DP [5, 3].

These advances have cemented DP as a cornerstone of privacy-preserving research, particularly in sensitive areas such as healthcare and finance.

2.2 Federated Learning in Healthcare

2.2.1 Adoption and Applications

FL enables collaborative model training across decentralized datasets, preserving data privacy by transmitting only model updates rather than raw data. This makes FL particularly valuable in privacy-sensitive applications like healthcare, where patient data must remain confidential.

Prominent applications include:

- **Medical Imaging Analysis:** Sheller et al. (2020) demonstrated FL's potential in multi-institutional brain tumor segmentation studies, achieving high diagnostic accuracy while maintaining data privacy [18].
- **COVID-19 Diagnosis:** Nguyen et al. (2021) used FL for diagnosing COVID-19 from chest X-rays, highlighting its scalability and effectiveness in privacy-compliant collaborative learning [19].
- **General Healthcare Applications:** FL aligns with regulations like HIPAA in the U.S. and GDPR in the EU, making it a practical solution for cross-institutional model training without compromising patient privacy [20].

2.2.2 Challenges in Federated Learning for Healthcare

Despite its benefits, FL faces several challenges in healthcare:

- **Data Heterogeneity:** Variations in patient demographics, equipment, and diagnostic protocols lead to non-IID data, affecting model convergence and performance. Personalized Federated Learning *(PFL)*, proposed by Fallah et al. (2020), and adaptive optimization strategies like those by Reddi et al. (2021), address these challenges by tailoring models to local data distributions [21, 22].

- **Privacy-Utility Trade-offs:** While inherently privacy-preserving, FL still requires mechanisms like DP for robust protection. Adding noise for DP can degrade model accuracy, especially in high-stakes applications. Balle and Wang's (2018) adaptive noise mechanisms dynamically adjust noise levels to balance privacy and utility [5].

2.3 Differential Privacy Mechanisms in Federated Learning

2.3.1 Applications of Gaussian and Laplace Mechanisms in Research

The Gaussian and Laplace mechanisms have been integral to privacy-preserving frameworks in FL. These methods introduce noise to gradients or model outputs to achieve DP, offering robust defenses against adversarial attacks while preserving data utility.

2.3.1.1 Gaussian Mechanism

- **Enhancing Security in Distributed Systems:** The Gaussian mechanism has become a standard choice for high-dimensional data tasks. For example, Malekzadeh et al. (2021) integrated Gaussian noise into federated models for diabetic retinopathy diagnosis, achieving robust privacy without degrading diagnostic performance [1].

- **Advanced Use in Multi-Site Medical Data:** Tang et al. (2023) proposed privacy-preserving FL frameworks using Gaussian noise and domain adaptation to bridge site-specific variations in ocular disease diagnosis datasets [23].

- **Addressing Network Scalability Issues:** Dijk et al. (2020) demonstrated an asynchronous FL approach combining Gaussian noise with reduced communication rounds, enhancing scalability for real-time applications [24].

2.3.1.2 Laplace Mechanism

- **Resource-Constrained Applications:** The Laplace mechanism, with its computational simplicity, is frequently applied in resource-constrained scenarios. Papernot et al. (2018) highlighted its application in lightweight federated systems, prioritizing privacy while minimizing computational overhead [25].

- **Exploration in Trajectory Data Protection:** The work of Gu et al. (2018) showcased a novel application of the Laplace mechanism in trajectory data protection, preserving privacy while maintaining data usability in mobility analytics [26].

2.3.2 Comparative Studies and Trends

Comparisons of Gaussian and Laplace mechanisms have revealed significant trends in privacy-preserving FL:

- **Task-Specific Preferences:** Zhou et al. (2022) demonstrated that while Gaussian mechanisms are better suited for high-dimensional tasks like medical imaging, Laplace mechanisms excel in simpler data environments requiring lower computational resources [27].

- **Utility-Preserving Innovations:** He et al. (2023) analyzed privacy-utility trade-offs, highlighting that Gaussian noise addition offers scalable solutions for maintaining accuracy in multi-client FL systems [28].

- **Sector-Specific Applications:** Zia et al. (2020) explored DP implementations in healthcare data sharing, emphasizing tailored noise levels for balancing utility and regulatory compliance [29].

2.3.3 Historical Development and Innovations

The evolution of DP mechanisms in FL has progressed through incremental innovations aimed at optimizing privacy and utility:

- **Initial Theoretical Foundations:** The work by Dwork et al. (2006) formalized DP, introducing the mathematical underpinnings that have since guided privacy-preserving model design [30].

- **Scaling DP for FL:** Advanced mechanisms such as the adaptive Gaussian approach by Jiao et al. (2023) have refined noise addition strategies for large-scale distributed systems, reducing model variance without compromising privacy [31].

- **Hybrid Mechanisms:** Recent studies, including those by Liu (2016), have combined DP techniques with bounding constraints to enhance utility in bounded statistical datasets [32].

2.4 Inversion Attacks and Mitigation Strategies

2.4.1 Threats of Inversion Attacks

Inversion attacks exploit shared model updates or gradients to reconstruct sensitive data, such as patient images. Fredrikson et al. (2015) first demonstrated this vulnerability in facial recognition systems [8]. More recent studies by Zhu et al. (2019) and Zhao et al. (2020) showed that FL is also susceptible to gradient-based inversion attacks, underscoring the need for robust privacy mechanisms in healthcare [7, 11].

2.4.2 Mitigation Strategies

Key strategies include:

- **DP:** DP-SGD masks individual data contributions by adding noise to gradients, enhancing protection against inversion attacks [6].
- **Cryptographic Methods:** MPC and HE add layers of encryption to model updates, mitigating risks during data sharing [33].
- **Adaptive Techniques:** Dynamic noise adjustment further improves resilience against sophisticated attacks by balancing privacy and model utility [5].

2.5 DR as a Case Study in Privacy-Preserving ML

2.5.1 Early Machine Learning Approaches

DR detection has been a significant focus within medical image analysis, with ML techniques, particularly CNNs, demonstrating potential in automating DR diagnosis. Gulshan et al. (2016) showed that CNN-based models could achieve diagnostic performance on par with ophthalmologists, establishing the foundation for DR detection frameworks [34].

2.5.2 Federated Learning and Differential Privacy in DR

Recent frameworks, such as Dopamine (Malekzadeh et al., 2021), integrate Gaussian noise into FL for privacy-preserving DR diagnosis [1]. *PFL* further addresses data heterogeneity across institutions by customizing models for local variations [21].

2.5.3 Challenges and Future Directions

Challenges in DR diagnosis include:

- **Balancing Privacy and Accuracy:** Noise added for privacy can degrade model performance. Adaptive techniques, such as those by Balle and Wang (2018), offer promising solutions but require further validation in clinical settings [5].
- **Class Imbalance:** DR datasets often have an imbalance between healthy and diseased images, affecting model generalization. Reweighting and augmentation strategies are potential solutions.

2.6 Synthesis and Research Gaps

- **Empirical Validation:** DP-enhanced FL models need testing in real-world healthcare workflows.
- **Advanced Adversarial Resilience:** Current approaches address basic attacks but lack robustness against adaptive adversaries.
- **Data Heterogeneity:** Strategies for handling variability in healthcare datasets, particularly through PFL, require further exploration.

Chapter 3

Proposed Method

3.1 Overview of the Proposed Method

The proposed framework introduces a privacy-preserving Federated Learning (FL) system tailored for the diagnostic analysis of diabetic retinopathy (DR) through medical image processing. This method integrates differential privacy (DP) mechanisms to address privacy-utility trade-offs, ensuring robust patient data protection without compromising diagnostic accuracy. By applying both the Gaussian and Laplace mechanisms, this framework provides a comparative analysis of privacy-preserving techniques within federated learning.

In this setup, four models are employed to assess varying levels of privacy and performance:

1. **Non-Private Machine Learning (ML):** A centralized baseline model where all client data is aggregated on a central server, allowing model training without privacy-preserving mechanisms. This setup provides a benchmark for evaluating diagnostic accuracy in non-private centralized conditions and establishes a baseline for comparing the performance of federated approaches.

2. **Non-Private Federated Learning (FL):** In this decentralized setup, client devices independently train on their local data and only share model updates with a central server. By keeping raw data on client devices, this approach serves as a federated learning baseline, providing insight into the effects of decentralization on model accuracy without any additional privacy mechanisms.

3. **Private FL with Gaussian Mechanism:** This model applies Gaussian noise to client updates before they are sent to the central server, controlled by a noise multiplier (σ).

Adjusting σ allows for testing different privacy-utility balances, enabling an analysis of how Gaussian noise impacts both model performance and privacy.

4. **Private FL with Laplace Mechanism:** In this model, Laplace noise is added to model updates, regulated by a privacy budget (ϵ). Different ϵ values are tested to identify optimal configurations where privacy and diagnostic accuracy are balanced.

After training, each model is rigorously evaluated and compared in terms of both performance and privacy guarantees. For the two private FL models, this includes:

- **Parameter Tuning and Trade-Off Analysis:** Various values for ϵ in the Laplace mechanism and noise multipliers in the Gaussian mechanism are tested. By plotting and analyzing these results, the framework identifies configurations that offer the best privacy-utility trade-off relative to baseline models.
- **Selection of Optimal Trade-Off:** The analysis seeks to identify configurations that achieve maximum privacy with minimal impact on accuracy. The goal is to pinpoint the smallest ϵ or highest σ values that effectively balance privacy and utility.
- **Inversion Attack Simulation:** To evaluate the robustness of the privacy mechanisms, simulated inversion attacks are performed to attempt reconstruction of dataset images from model gradients. By observing how effectively the Gaussian or Laplace noise conceals this information, the study gains insights into the effectiveness of each privacy approach. This simulation measures the impact of noise on the quality of reconstructed images, providing a practical assessment of privacy protection under potential adversarial conditions.

The final results are presented in comparative visualizations, showcasing the optimal privacy configurations for each mechanism and demonstrating the practical viability of privacy-preserving FL for medical image analysis. This framework ultimately aims to establish a balance between robust data privacy and high diagnostic utility within sensitive healthcare applications.

3.2 Dataset

Diabetic retinopathy (DR) is a severe eye condition that can lead to vision loss in diabetic patients due to damage to the retinal blood vessels. Diagnosing this condition requires analyzing

retinal images to identify signs of damage, framing it as an image classification task. For this study, we use a diabetic retinopathy dataset introduced by Choi et al. (2017), which is publicly available through the APTOS 2019 Blindness Detection competition on Kaggle.[1] The task is to classify retinal images into one of five categories representing different levels of diabetic retinopathy severity:

- **No DR**: No signs of diabetic retinopathy.
- **Mild DR**: Early signs of retinopathy.
- **Moderate DR**: More prominent signs, with potential progression.
- **Severe DR**: Significant progression, with risk of severe vision impairment.
- **Proliferative DR**: Advanced stage with substantial risk of vision loss.

The dataset includes 2,931 training images and 731 testing images, each with variable dimensions. To prepare the images for model training and ensure uniformity, all images were resized to 224 x 224 pixels during pre-processing. This resizing step standardizes the input, optimizing it for the SqueezeNet model used in this study.

3.3 Model Architecture

This section provides an overview of the selected model architecture, SqueezeNet, detailing its suitability for the study, specific customizations for diabetic retinopathy classification, and consistent initialization across federated learning clients. The chosen model, SqueezeNet, is a compact and efficient neural network that balances accuracy with computational efficiency, making it well-suited for medical image analysis where high-resolution images are common. SqueezeNet's lightweight structure and small memory footprint enable deployment across multiple clients with limited computational resources, a critical factor in distributed healthcare environments where federated learning is implemented. SqueezeNet's architecture includes a series of Fire modules, which reduce computational costs while maintaining accuracy. Each Fire module contains a *squeeze* layer that minimizes the number of input channels via a 1x1 convolution, followed by an *expand* layer that applies both 1x1 and 3x3 convolutions. This structure maximizes parameter efficiency, achieving a compact model size without sacrificing performance. The small parameter count helps mitigate overfitting, an important consideration when working with a relatively small dataset, such as one for diabetic retinopathy.

[1] https://www.kaggle.com/c/aptos2019-blindness-detection/data

To tailor SqueezeNet for diabetic retinopathy diagnosis, the model's output layer was customized to accommodate five classes: No DR, Mild DR, Moderate DR, Severe DR, and Proliferative DR. This was achieved by modifying the classifier layer, replacing the final convolutional layer with a 1x1 convolution layer that outputs five channels. This change enables the model to classify retinal images into these specific diagnostic categories, thus enhancing its utility in medical image analysis. Additionally, *Rectified Linear Unit* (ReLU) activation functions were employed, which are particularly effective in deep convolutional neural networks used in medical image analysis, contributing to faster convergence and improved model performance.

Consistency in model initialization across federated learning clients was ensured by starting each client with an identical SqueezeNet model. The model was initialized with pretrained ImageNet weights (SqueezeNet1_1_Weights.IMAGENET1K_V1), providing a robust starting point that leverages prior knowledge from a large-scale general image dataset. This shared initialization supports straightforward aggregation of client updates during each round of federated learning, ensuring uniform starting conditions for all participating clients.

3.4 Privacy-Preserving Mechanisms and Training Process

This study secures sensitive data in federated learning through Differential Privacy (DP) mechanisms, specifically using Laplace and Gaussian methods to add controlled noise to model gradients. These mechanisms aim to obscure individual data contributions, safeguarding privacy while retaining model utility during training.

3.4.1 Differential Privacy Mechanisms: Laplace and Gaussian Approaches

The privacy parameter, ϵ, is central to DP mechanisms as it determines the balance between privacy and model utility. Smaller ϵ values indicate higher privacy but introduce greater noise, potentially impacting accuracy. Based on differential privacy theory, three primary ranges for ϵ were evaluated:

- **High Privacy ($\epsilon < 1$)**: Strong privacy protection with significant noise addition, suitable for highly sensitive data but can impact accuracy.

- **Moderate Privacy ($1 \leq \epsilon \leq 3$)**: A balance between privacy and accuracy, offering reasonable protection while preserving utility.

- **Low Privacy ($\epsilon > 3$)**: Minimal privacy protection with less noise, enhancing accuracy but less suitable for confidential settings.

In this study, testing began with $\epsilon = 0.5$ for high privacy and was incrementally increased to $\epsilon = 3$ to monitor how privacy adjustments affected model performance. This iterative testing helped determine values that achieved the desired privacy-utility balance.

3.4.1.1 Laplace Mechanism

The Laplace mechanism, implemented via a custom `LaplaceOptimizer`, applies noise directly to gradients based on the specified ϵ value. This approach provides flexible tuning of privacy levels, and gradients are clipped before noise is added to prevent any single data point from disproportionately influencing the model. Observations showed that, at moderate ϵ values (e.g., 1–3), the model retained reasonable accuracy while maintaining a secure privacy threshold, balancing the privacy-utility trade-off.

3.4.1.2 Gaussian Mechanism

For the Gaussian mechanism, the Opacus library's `PrivacyEngine` was used to automate gradient clipping, noise addition, and batch sampling. Controlled by a `noise_multiplier` parameter, Gaussian noise effectively obfuscates sensitive data in gradients. The noise multiplier, σ, for the Gaussian mechanism was calculated according to:

$$\sigma = \frac{\sqrt{2\ln(1.25/\delta)}}{\epsilon}$$

where δ was set to 10^{-4}, representing an acceptable probability of privacy failure, and Δf (sensitivity) was set to 1.

To ensure a fair comparison between the Laplace and Gaussian models, we calculated the noise multiplier values for the Gaussian mechanism based on the corresponding epsilon values, thereby achieving consistent noise levels across both mechanisms.

By systematically varying the ϵ values in the Laplace model and adjusting σ in the Gaussian model, this study identified optimal configurations for each mechanism that balanced privacy protection with diagnostic accuracy. These empirical observations, combined with theoretical guidelines, validated the selected parameter ranges and supported the effectiveness of privacy-preserving federated learning in healthcare applications.

3.4.2 Training Process Overview

The training process varies among centralized non-private Machine Learning (ML), non-private Federated Learning (FL), and private FL models that utilize Differential Privacy mechanisms. Each approach is tailored to balance privacy, computational efficiency, and diagnostic accuracy.

1. **Centralized Non-Private ML Training**

The centralized ML model serves as a baseline, with all data aggregated on a single server for unified training, maximizing accuracy but offering no privacy protection:

- Data is aggregated centrally, processed in batches, and updated through gradient optimization.

2. **Non-Private FL Training**

The non-private FL model enables decentralized training across clients without noise addition, preserving data localization but without additional privacy mechanisms:

- Each client trains locally, and only model updates are shared with a central server.
- The central server aggregates these updates using weighted averaging.

3. **Private FL Training with Differential Privacy Mechanisms**

Private FL training incorporates DP by applying Gaussian or Laplace noise to gradients before sharing them with the central server. This ensures that model updates do not expose sensitive information from individual datasets:

- Each client trains locally, applying DP to gradients.
- **Private FL with Gaussian Mechanism**: Gradients are clipped, and Gaussian noise is added through Opacus, balancing privacy with the `noise_multiplier` parameter.
- **Private FL with Laplace Mechanism**: Using a custom `LaplaceOptimizer`, gradients are clipped, and Laplace noise, controlled by ϵ, is applied.

3.4.3 Global Model Aggregation

In both private and non-private FL models, global updates are achieved by aggregating client model updates:

- **Weighted Averaging of Model Weights**: The server aggregates updates via weighted averaging to integrate insights from all clients.

- **Enhanced Privacy in Private FL**: In the private FL setup, noise addition further strengthens privacy protection, particularly for sensitive healthcare applications.

3.5 Implementation Details

This section provides an overview of the key software tools, libraries, and workflow utilized in implementing the privacy-preserving federated learning framework for diabetic retinopathy diagnosis.

3.5.1 Software and Libraries

The implementation of the proposed federated learning system with differential privacy relies on several key software tools chosen for their robustness, flexibility, and specific support for privacy-preserving machine learning:

- **Jupyter Notebook:** Provides an interactive environment for iterative data analysis and model testing, integrating live code, visualizations, and documentation to enhance readability and reproducibility.
- **PyTorch:** A dynamic machine learning library that enables flexible model development and training with *Compute Unified Device Architecture* (CUDA) support, essential for handling large-scale datasets and optimizing federated learning setups.
- **Opacus:** Extends PyTorch with Differential Privacy capabilities, offering granular privacy controls and efficient *Graphics Processing Unit* (GPU)-optimized implementations that facilitate privacy-preserving model training with minimal adjustments.

These tools collectively support the development and evaluation of privacy-preserving federated learning models, underscoring our use of advanced technologies in medical machine learning applications.

3.5.2 General Workflow of Code Files

This section outlines the primary workflow for implementing and evaluating federated learning (FL) models with differential privacy for diabetic retinopathy diagnosis. The process progresses from dataset preparation to model training, privacy configuration, and validation through inversion attack simulations. In the next section, we will explain each part in more details.

- **Dataset Preparation and Distribution:** The files `dr_dataset_to_numpy.py` and `datasets.py` handle dataset loading, preprocessing, and conversion into a PyTorch-compatible format, with data distributed among clients to simulate real-world federated learning setups.
- **Configuration and Hyperparameters:** Experimental parameters, including epochs, batch sizes, learning rates, and privacy settings, are defined in `options.py`, ensuring consistent configuration across models.
- **Model Definition and Initialization:** `models.py` specifies the model architecture, SqueezeNet, for diabetic retinopathy classification, adaptable for both private and non-private FL, as well as non-private ML configurations.
- **Training Implementation for FL Models:** The primary training scripts—`update_s2.py` for centralized ML, `update_s3.py` for non-private FL, and `Laplace_update_s3.py` for private FL—implement the training loop tailored for each privacy requirement.
- **Model Aggregation and Performance Logging:** average_weights() in `utils.py` performs global aggregation of model updates, with accuracy and loss metrics tracked by `logging_results.py` to assess privacy-utility trade-offs.
- **Optimal Privacy Parameter Selection:** Post-training analysis identifies the optimal noise multiplier for Gaussian and epsilon for Laplace mechanisms, which are then used in inversion attack simulations for privacy validation.
- **Inversion Attack Simulation:** Using `Gradient_Matching.ipynb`, inversion attacks are simulated to test the privacy robustness of each model under worst-case conditions.
- **Privacy Robustness Evaluation:** Privacy robustness is quantitatively assessed using metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) on reconstructed images from the inversion attacks.

3.5.3 Detailed Explanation of Each Code File

This section provides a detailed breakdown of each code file used in the implementation. Each file is discussed in terms of its specific functionality, input-output structure, and its role within the overall workFLow described in the previous section. Additionally, the files' roles in preparing, training, evaluating, and validating models through inversion attacks are highlighted.

3.5.3.1 Data Processing and Preparation with `dr_dataset_to_numpy.py`

The `dr_dataset_to_numpy.py` script is crucial for automating the preparation and formatting of the diabetic retinopathy (DR) dataset, enabling efficient data loading and consistent formatting for federated learning experiments. By converting raw data into a preprocessed NumPy format, this script not only accelerates the loading process but also ensures reproducibility across iterative experiments with differential privacy.

Key elements of the script include:

- **Setting a Reproducible Environment**: The script begins by calling `set_seed()` to initialize a random seed, ensuring that operations involving randomness, such as dataset shuffling, produce consistent results in every run, a crucial factor for experimental reliability.

- **Configuring Device and Parsing Arguments**: Using `args_parser()`, the script configures parameters, such as GPU or *Central Processing Unit* (CPU) selection. If a GPU is available, the device is set to `cuda`, leveraging faster processing speeds advantageous for handling large image datasets in federated learning.

- **Loading the Dataset**: The diabetic retinopathy dataset is retrieved via `get_dataset()` from `datasets.py`, which provides:

  - `train_dataset` and `test_dataset`, each containing image-label pairs formatted for training and testing, respectively.
  - `user_groups`, a data index mapping for clients, which supports federated learning's distributed data requirements.

  In the next section, we will explain how `datasets.py` works.

- **Extracting and Saving Data**: The script iterates over each image-label pair in the training and testing datasets, converting the images into NumPy arrays for optimized storage. Processed data is saved as `dr_train_images.npy`, `dr_train_labels.npy`, `dr_test_images.npy`, and `dr_test_labels.npy`, ensuring that experiments can efficiently access preprocessed data without redundant processing steps.

- **Real-Time Progress Updates**: To facilitate monitoring, the script displays real-time progress during data processing, making it straightforward to track and troubleshoot the conversion process, especially when working with large datasets.

- **Advantages of Storing Data in NumPy Format**: Saving the dataset in a NumPy format is especially beneficial for federated learning. Distributed access to preprocessed data in a standardized format is more efficient and simplifies handling complex FL setups. By using the `--dr_from_np` flag, data can be loaded directly from NumPy files, bypassing re-downloads and preprocessing steps, which enhances computational efficiency for repeated experiments.

The `dr_dataset_to_numpy.py` script ensures that the diabetic retinopathy dataset is preprocessed, standardized, and readily accessible, streamlining FL experiments and supporting high-quality, reproducible research.

3.5.3.2 Dataset Management and Transformation with `datasets.py`

The `datasets.py` file is responsible for loading, processing, and distributing the dataset required for this study. Its flexible configuration options for data transformations and client distribution enable efficient use in both centralized and federated learning contexts.

Key elements of the script include:

- **Class Definition for Diabetic Retinopathy Dataset** (`DRDataset`): This class inherits from PyTorch's `Dataset` class to manage the specific needs of diabetic retinopathy images.
  - **Attributes** include:
    - `data_label`: Stores image IDs and labels in a pandas DataFrame.
    - `data_dir`: Specifies the directory containing image files.
    - `transform`: Applies a series of transformations like resizing, cropping, normalization, and augmentation to standardize input for model training.
  - **Methods** include:
    - `__len__`: Returns the number of images.
    - `__getitem__`: Retrieves and transforms images by index, returning each with its label.
- **Dataset Loading with `get_dataset`**: This function manages dataset retrieval, transformations, and client distribution:
  - **Diabetic Retinopathy Dataset Modes**:

        * **Loading from `.npy` Files**: If preprocessed images exist, they load directly from NumPy files, bypassing redundant preprocessing.
        * **Loading from Raw Images**: If `.npy` files are unavailable, the function downloads, extracts, and saves images and labels, applying consistent transformations to prepare the data for input to the model.
    - **Federated Learning Data Distribution**:
        * ***Independent and Identically Distributed* (IID) Configuration**: Ensures a balanced distribution across clients for simpler FL setup.
        * **non-IID Configuration**: Distributes data unevenly, simulating real-world scenarios with heterogeneous client data, adding complexity to model training.

- **Preprocessing and Augmentation**: To ensure data compatibility with model input requirements, several transformations are applied:
    - **Normalization**: RGB channels are normalized with a mean and standard deviation of 0.5 for training stability.
    - **Resizing and Cropping**: Images are resized to 265x265 pixels and then center-cropped to 224x224 pixels to align with SqueezeNet's input size.
    - **Data Augmentation**: Random horizontal flipping increases data variability, supporting model generalization on unseen images.

The `datasets.py` file is essential for organizing and preparing the diabetic retinopathy dataset, ensuring data is readily available in the correct format for both centralized and federated learning experiments, particularly those with differential privacy mechanisms.

3.5.3.3 Experiment Configuration with `options.py`

The `options.py` file defines command-line arguments and configuration parameters that enable flexible and reproducible experimentation. This script allows users to adjust key aspects such as model architecture, federated learning settings, differential privacy parameters, and dataset selection.

- **Federated Parameters**: Defines core federated learning parameters including the number of global training rounds, number of users, fraction of clients selected per round,

local epochs, and batch size. These settings shape the federated learning process by controlling user participation frequency and local training depth.

- **Model and Optimizer Parameters**: Specifies model type, optimizer, learning rate, and momentum, providing flexibility to experiment with different model architectures and optimization strategies suited to the data and privacy requirements.
- **Differential Privacy (DP) Parameters**: Includes settings for gradient clipping threshold, noise multiplier, epsilon, and delta values, which together control the level of privacy during training. By adjusting these parameters, the user can fine-tune the noise added to gradients, balancing privacy with model utility.
- **Dataset and Miscellaneous Parameters**: Offers options for dataset selection, number of classes, device (CPU or GPU), data distribution (IID or non-IID), and the use of preprocessed data specifically for diabetic retinopathy. These parameters make it easy to experiment with diverse datasets and configurations across various federated learning scenarios.

By using command-line arguments, `options.py` promotes efficient exploration of parameter impacts on model performance and privacy, facilitating reproducible and adaptive experimentation.

3.5.3.4 Utility Functions with `utils.py`

The `utils.py` file contains essential utility functions that support federated learning tasks, model evaluation, and training management. These functions handle critical tasks, including model testing, weight aggregation, and training optimization, contributing to the efficiency and organization of the training process.

- **`test_inference()`**: Evaluates a trained model's performance on a test dataset, calculating accuracy and loss.
  - **Process**: Sets the model to evaluation mode to prevent gradient calculations, moves data to the appropriate device, and computes loss and accuracy over test batches.
  - **Output**: Returns accuracy and loss, serving as key metrics for model evaluation.
- **`average_weights()`**: Aggregates weights from multiple client models to update the global model in a federated learning setup.

- **Process**: Deep-copies the first client's weights as a baseline and iteratively averages these across all clients to form a unified global model.
- **Usage**: Ensures the global model reflects the collective updates, maintaining consistency across participating clients.

- **`exp_details()`**: Displays experimental details, such as model type, optimizer, and training parameters.
  - **Usage**: Provides a quick reference for experiment configurations, aiding in tracking and debugging.

- **`EarlyStopping` (Class)**: A class to implement early stopping, helping to prevent overfitting by halting training if validation loss does not improve over a specified patience period.
  - **Attributes**: Includes `patience`, `delta`, and `path`, setting criteria and location for model checkpoints.
  - **Methods**:
    - `__call__`: Monitors validation loss improvement, incrementing a counter if no improvement occurs. Stops training after reaching the patience limit.
    - `save_checkpoint`: Saves the model whenever validation loss improves.
  - **Usage**: Optimizes training by reducing unnecessary epochs and preserving the best-performing model.

3.5.3.5 Data Sampling for Federated Learning with `sampling.py`

The `sampling.py` file is integral to dividing datasets among clients, enabling both IID (Independent and Identically Distributed) and non-IID distributions. This simulation of real-world federated learning setups allows each client to receive data in line with the assigned configuration, supporting a fair evaluation across different models.

- **`dist_datasets_iid()`**: Ensures even data distribution across users, creating an IID setup.
  - **Process**: Calculates the number of samples per client and randomly assigns these to each client. This process ensures that data classes are uniformly represented across users.

- **Output**: A dictionary mapping each client to a unique set of data indices, enabling a balanced, IID-based federated learning environment.

- **`dist_datasets_noniid()`**: Distributes data unevenly to clients, creating a non-IID configuration where each client's dataset may reflect distinct data distributions.

  - **Process**: Organizes data by labels, divides it into shards, and assigns these subsets to different clients. This setup mimics realistic federated learning environments, where client data varies in content, creating a challenging scenario for model training.

  - **Output**: A dictionary mapping each client to a specific subset of data indices, representing a unique data distribution for each user in the non-IID setup.

This variation in data patterns is crucial for testing model robustness under heterogeneous conditions, mirroring real-world federated learning challenges.

3.5.3.6 Logging Experiment Results with `logging_results.py`

The `logging_results.py` file is responsible for capturing and organizing key training metrics across experiments, including accuracy, loss, and privacy-related parameters, such as epsilon values in differentially private (DP) models. These logs enable effective tracking of model performance and analysis of privacy-utility trade-offs over time, providing critical insights for selecting optimal noise levels for inversion attack simulations.

- **Logging Functionality**: The main function, `logging()`, records training loss, test accuracy, and DP metrics (e.g., epsilon values) for each epoch. It saves these metrics in dedicated directories, such as `train_log`, `test_log`, and `privacy_log`, with uniquely named experiment files for structured tracking.

- **Output**: By storing metrics in organized text files, this file provides a comprehensive record of model performance over time, including accuracy and convergence rates, as well as privacy-accuracy trade-offs. This historical data is essential for analyzing different model configurations and their impacts on both utility and privacy.

This organized logging framework supports comparison across experiments, enabling an in-depth analysis of model convergence and privacy-utility trade-offs in federated learning settings.

3.5.3.7 Training and Model Update Approaches for Different Privacy Mechanisms

The files `update_s2.py`, `update_s3.py`, and `Laplace_update_s3.py` implement varied training loops tailored to centralized non-private machine learning (ML), decentralized non-private federated learning (FL), and private FL models utilizing Gaussian and Laplace differential privacy mechanisms. This design accommodates different levels of data decentralization and privacy, with unique methods for managing gradients and privacy-preserving noise.

Each file shares structural elements like the `DatasetSplit` class, which partitions datasets by client, a `train_val_test()` method that creates train and test data loaders, and an `update_weights()` method to perform model updates. The `update_weights()` method serves as the core training loop, managing gradients and updates specific to each privacy mechanism.

- **Centralized Non-Private ML (`update_s2.py`)**: This file executes a training loop for centralized, non-private ML, serving as a baseline for performance comparison. Privacy mechanisms are unnecessary in this configuration.

  - **Model Training**: The `update_weights()` method initiates training over a specified number of epochs. For each batch, gradients are computed with a loss function and directly applied via the optimizer, bypassing any privacy protections.
  - **Output Gradients**: The file captures gradients from the output layer's weights and biases, later used to assess noise effects in private models.
  - **Logging and Epsilon Tracking**: While differential privacy is not applied, the script logs metrics like training loss and accuracy per epoch, with a placeholder for `epsilon_log` should differential privacy be enabled.

- **Federated Learning with Optional Gaussian Privacy Mechanism (`update_s3.py`)**: This file supports both non-private FL and FL with Gaussian differential privacy, activated by setting `withDP`.

  - **Privacy-Enhanced Training**: When `withDP` is enabled, the `PrivacyEngine` from Opacus adds Gaussian noise to gradients, protecting data against inference attacks. Key privacy parameters, such as `noise_multiplier` and `max_grad_norm`, manage the trade-off between privacy and utility.

- **Virtual Batch Rate**: To optimize efficiency, a virtual batch rate splits the batch size into smaller units, allowing privacy-enhanced updates at set intervals. This method stabilizes updates while controlling computational costs of noise addition.
- **Epsilon Tracking**: After each training round, the `PrivacyEngine` calculates and logs the privacy budget ϵ, which quantifies privacy loss for comparison with other mechanisms.

- **Federated Learning with Custom Laplace Mechanism (`Laplace_update_s3.py`)**: This file implements FL with a custom Laplace noise addition. Unlike the Gaussian method, it uses a Laplace distribution tailored to the defined `epsilon` and sensitivity.
  - **Custom Laplace Noise Addition**: A dedicated function, `add_laplace_noise()`, generates Laplace noise based on sensitivity and `epsilon` values. This noise is directly applied to gradients, ensuring privacy through obfuscation.
  - **Laplace Optimizer**: The `LaplaceOptimizer` class inherits from PyTorch's `Optimizer` and overrides the `step()` method to perform gradient clipping and noise injection. This custom optimizer clips gradients at a defined `max_grad_norm` before applying Laplace noise.
  - **Output Gradients**: The script logs output layer gradients, facilitating comparisons between Laplace, Gaussian, and non-private models.
  - **Epsilon Logging**: Unlike Opacus, which automatically tracks `epsilon` for Gaussian noise, `epsilon` for Laplace is statically defined in the arguments and logged manually to monitor privacy levels.

The `update_s2.py` file represents a centralized ML baseline without privacy, while `update_s3.py` and `Laplace_update_s3.py` introduce Gaussian and Laplace noise, respectively, for a range of privacy-utility evaluations.

3.5.3.8 Analysis of Model Implementation Notebooks

The study evaluates performance and privacy-utility trade-offs in federated learning (FL) using four configurations. Each notebook implements a distinct model setup: centralized non-private ML, decentralized non-private FL, private FL with Gaussian noise, and private FL with Laplace noise. Consistent structure across the notebooks allows for specific modifications in privacy requirements and data distribution. Below is an overview of each configuration:

1. **Centralized Non-Private ML: non_private_ML.ipynb**

This notebook serves as a baseline with a single centralized model trained on the entire dataset without privacy mechanisms.

- **Data Loading and Preparation**: The `get_dataset()` function loads and preprocesses the diabetic retinopathy dataset.
- **Model Setup**: SqueezeNet, pretrained on ImageNet, is modified for diabetic retinopathy classification with five output classes.
- **Training and Evaluation**: The model trains on the full dataset with a standard optimizer, with metrics logged for performance comparison.

2. **Decentralized Non-Private FL: non_private_FL.ipynb**

This setup enables decentralized model training across multiple clients, simulating a federated environment without differential privacy.

- **Data Distribution**: Clients receive IID data splits, specified by `sampling.py`.
- **Federated Averaging and Logging**: Client weights are averaged to update the global model, with performance metrics recorded after each epoch.

3. **Private FL with Gaussian Noise: Gaussian_FL.ipynb**

This notebook extends non-private FL by incorporating Gaussian differential privacy.

- **Privacy Engine**: Opacus's `PrivacyEngine` adds Gaussian noise to gradients based on set parameters, allowing privacy control.
- **Privacy Monitoring and Experimentation**: Privacy budget ϵ is logged for trade-off analysis, with multiple noise multiplier values tested to optimize privacy-utility balance.

4. **Private FL with Laplace Noise: Laplace_FL.ipynb**

The Laplace FL model uses custom Laplace noise instead of Gaussian, with distinct privacy-utility trade-off characteristics.

- **Custom Laplace Mechanism**: The `LaplaceOptimizer` applies noise post-gradient clipping, with flexibility in tuning ϵ values.
- **Experimentation with Epsilon Values**: Varying ϵ values are tested, providing insights into the Laplace noise impact on model utility.

6. **Comparison of Model Configurations**

Each configuration adheres to a similar training pipeline, facilitating performance comparisons.

- **Baseline Models**: Centralized ML and decentralized FL serve as benchmarks without privacy mechanisms.
- **Privacy Mechanisms**: Gaussian and Laplace setups introduce privacy while enabling analysis of differential noise effects.
- **Experimentation with Privacy Parameters**: Both private configurations vary settings to explore privacy-utility balances.

3.5.3.9 Inversion Attack Simulation for Privacy Validation

This section outlines the process and evaluation of inversion attack simulations to validate the privacy robustness of differentially private federated learning models. By reconstructing data from shared model gradients, the inversion attack helps assess the effectiveness of privacy-preserving mechanisms like Gaussian and Laplace noise in federated learning.

The inversion attack simulation tests models trained both with and without differential privacy (DP), evaluating the extent to which Gaussian and Laplace noise obfuscate sensitive data.

To simulate a worst-case scenario, the attacker has access to both model architecture and gradients. Using gradient matching, the attack iteratively adjusts noise-based inputs until their gradients align with the shared model gradients. In our simulation, SqueezeNet is used for both original and attack models, simplifying the reconstruction process by maintaining consistent architectures.

The inversion attack implementation is custom-built to accommodate specific experimental needs. Below are key functions in this process:

- **Gradient Loading and Preparation**: The function `load_gradients` retrieves gradients from saved files, each representing model updates under different DP configurations. To maintain consistency, gradients correspond to a limited training epoch count (defined by `num_epochs_to_load`). Variations in gradient tensor shapes across layers present a technical challenge, requiring careful reshaping to ensure compatibility with the attack model.

- **Gradient Reshaping for Model Compatibility**: The `reshape_gradients` function adjusts gradient tensors to align with the model's output shapes. By reshaping, truncating, or repeating elements, this function enables seamless application of gradients during the matching process. This step is crucial for accurately matching gradients across layers without dimensional mismatches.

- **Gradient Matching and Inversion Process**: The core of the inversion attack is the `perform_inversion_attack` function. Here, random noise inputs are iteratively optimized to match model-generated gradients closely. Using Mean Squared Error (MSE) as a loss function, the process minimizes gradient discrepancies, iteratively reconstructing an image that reveals privacy vulnerabilities.

The Python code developed for evaluation metrics automates the process of loading images, computing similarity metrics, and performing an in-depth analysis of reconstructed images. Below is an explanation of the main components of this code:

- **Image Loading and Preprocessing**: The code begins by loading pairs of images (original and reconstructed) in grayscale format for standard metrics (such as PSNR and SSIM) and in RGB format for perceptual loss calculations, which require color information. This structure ensures that each metric is computed using the optimal input format, enhancing the accuracy of the results.

- **Pre-trained VGG Model for Perceptual Loss**: To compute perceptual loss, a pre-trained VGG19 model from PyTorch is used. This model extracts high-level features from both original and reconstructed images. The perceptual loss is calculated by comparing these feature representations, providing a measure of perceptual similarity at a higher level than pixel values alone.

- **Multiscale SSIM Calculation**: A custom function `compute_multiscale_ssim` computes MS-SSIM by repeatedly downsampling the images and calculating SSIM at each scale. This approach captures structural fidelity across multiple resolutions, making it useful for assessing fine-grained image details that may be lost due to noise.

- **Gradient PSNR Calculation**: Gradient PSNR (G-PSNR) is computed by first applying the Laplacian operator to both images to extract edge information and then calculating PSNR on these gradient representations. Since edge details are crucial for medical images, this metric provides insights into how well structural boundaries are preserved or obscured by noise.

- **Comprehensive Metric Calculation**: The code calculates six metrics: PSNR, SSIM, MS-SSIM, FSIM, perceptual loss, and G-PSNR. Each metric is returned for each image pair, allowing for a holistic evaluation of privacy preservation.

By quantifying these metrics (PSNR, SSIM, MS-SSIM, FSIM, Perceptual Loss, and G-PSNR), each privacy mechanism is systematically evaluated, offering a detailed understanding of the balance between privacy and data utility. This comprehensive assessment helps identify optimal privacy configurations for federated learning in sensitive healthcare contexts, ensuring both model utility and privacy protection.

Chapter 4

Experimental Results

4.1 Introduction

This chapter presents the findings of our study on integrating differential privacy (DP) mechanisms within a federated learning (FL) framework for diabetic retinopathy (DR) diagnosis. We aim to examine the impact of DP on privacy and diagnostic accuracy, focusing on two primary goals:

1. Privacy-Utility Trade-Off Analysis

The first goal is to evaluate the performance of four models: two baseline models (centralized non-private ML and decentralized non-private FL) and two privacy-preserving FL models (utilizing Gaussian and Laplace mechanisms). For the private FL models, we assess multiple privacy levels by varying ϵ values in the Laplace model and noise multipliers in the Gaussian model.

This analysis includes:

- **Laplace Model Plot**: A plot comparing the diagnostic accuracy of the baseline models with the Laplace model across different ϵ values to determine the optimal balance of privacy and accuracy.
- **Gaussian Model Plot**: A plot comparing baseline models with the Gaussian model at varying noise multipliers, providing insights into the most effective configuration for privacy and utility.

Finally, a composite plot will showcase the baseline models alongside the chosen Laplace and Gaussian configurations, highlighting the best trade-off points between privacy and accuracy.

2. Inversion Attack Simulation and Privacy Robustness Evaluation

The second goal is to test the robustness of the four models against inversion attacks to assess how effectively each configuration protects data privacy. This will include:

- **Visual Comparisons**: Images showing the original image alongside reconstructed versions from each model to visually assess noise-induced obfuscation.
- **Quantitative Comparison**: For a selected image, I compute multiple metrics to evaluate the fidelity of reconstructed images compared to the original, providing quantitative insights into privacy protection efficacy. These metrics include:
  - **Peak Signal-to-Noise Ratio (PSNR)**: Measures the ratio between the original and noise-induced variations, where lower values indicate higher privacy.
  - **Structural Similarity Index Measure (SSIM)**: Assesses the structural similarity between the original and reconstructed images, with lower values reflecting reduced resemblance and stronger privacy.
  - **Multiscale SSIM (MS-SSIM)**: Extends SSIM to multiple scales, capturing fine-grained differences in structural fidelity; lower values indicate better privacy protection.
  - **Feature Similarity Index (FSIM)**: Evaluates perceptual similarity based on image features like phase congruency and gradient magnitude, with lower values indicating greater privacy.
  - **Perceptual Loss**: Calculated using feature representations from a pre-trained neural network, this metric quantifies high-level perceptual similarity. Higher perceptual loss values indicate stronger privacy protection.
  - **Gradient PSNR (G-PSNR)**: Focuses on edge and detail preservation by calculating PSNR on the gradient of images. Lower values signify that edge details are more obscured, enhancing privacy.

These metrics collectively provide a comprehensive evaluation of privacy-preserving effectiveness, where lower values for PSNR, SSIM, MS-SSIM, FSIM, and G-PSNR, and higher perceptual loss values generally indicate better privacy due to reduced resemblance to the original data.

In the following sections, I detail the implementation setup, present the results for each goal, and interpret the findings with respect to both privacy and diagnostic accuracy.

4.2 Experimental Setup and Result for Privacy-Accuracy Trade-off Analysis

In this section, I describe the experimental setup used to evaluate the performance and privacy-utility trade-offs of four models: centralized non-private machine learning (ML), decentralized non-private federated learning (FL), and two differentially private federated learning models utilizing Gaussian and Laplace mechanisms. Each model is configured with specific parameters to optimize performance and privacy, with particular focus on the Differential Privacy (DP) parameters in the private FL models.

The configurations for each model, including shared and unique arguments, are presented in Figure 4.1. Key parameters, such as the number of training epochs, batch size, optimizer type, and differential privacy settings (e.g., epsilon values and noise multipliers), are specified for each model type. The values provided in Figure 4.1 ensure consistency in training while allowing for meaningful comparisons across the baseline and private models.

Argument	Centralized Non-Private ML	Non-Private Decentralized FL	Private Gaussian FL	Private Laplace FL
Epochs	100	100	100	100
Number of Users	1	10	10	10
Fraction of Clients (frac)	1.0	0.5	0.5	0.5
Local Epochs (local_ep)	1	5	5	5
Local Batch Size (local_bs)	50	50	50	50
Optimizer	sgd	sgd	sgd	sgd
Learning Rate (lr)	0.001	0.002	0.002	0.002
Momentum	0.9	0.9	0.9	0.9
Model	cnn	cnn	cnn	cnn
Activation	relu	relu	relu	relu
With Differential Privacy (withDP)	0	0	1	1
Dataset	dr	dr	dr	dr
Number of Classes	5	5	5	5
Device	cuda:0	cuda:0	cuda:0	cuda:0
IID Data Distribution	1	1	1	1
Unequal Data Splits	0	0	0	0
Sub-Dataset Size	-1	-1	-1	-1
Local Test Split	0.30	0.30	0.30	0.30
Retinopathy from NumPy (dr_from_np)	1	1	1	1
Experiment Name	exp_results	exp_results	exp_results	exp_results
DP Parameters	-	-	Noise Multiplier (varied)	Epsilon (varied)
Gaussian Noise Multiplier	-	-	8.69, 4.34, 2.90, 2.17, 1.74, 1.45	-
Laplace Epsilon	-	-	-	0.5, 1.0, 1.5, 2.0, 2.5, 3.0
DP Delta	-	-	10^-4	10^-4
Virtual Batch Size	-	-	50	50
Sampling Probability	-	-	0.001	0.001
Sensitivity	-	-	1	1
Max Gradient Norm	-	-	1	1

Figure 4.1: Experimental Setup for Model Configurations

For the differentially private models, privacy levels were systematically explored by adjusting the noise multiplier in the Gaussian mechanism and the epsilon value in the Laplace mechanism. To ensure a fair comparison between the Laplace and Gaussian models, we cal-

culated the noise multiplier values for the Gaussian mechanism based on the corresponding epsilon values, thereby achieving consistent noise levels across both mechanisms.

The noise multiplier σ was calculated using the formula:

$$\sigma = \frac{\sqrt{2\ln(1.25/\delta)}}{\epsilon}$$

where δ was set to 10^{-4}, representing an acceptable probability of privacy failure, and the sensitivity Δf was set to 1. This calculation provided noise levels equivalent to the specified epsilon values, allowing for direct comparison between the privacy-accuracy trade-offs in the Laplace and Gaussian mechanisms.

The selected epsilon values ranged from 0.5 to 3.0, with their corresponding noise multipliers calculated as follows:

- **Epsilon: 0.5 → Noise Multiplier: 8.69**
- **Epsilon: 1.0 → Noise Multiplier: 4.34**
- **Epsilon: 1.5 → Noise Multiplier: 2.90**
- **Epsilon: 2.0 → Noise Multiplier: 2.17**
- **Epsilon: 2.5 → Noise Multiplier: 1.74**
- **Epsilon: 3.0 → Noise Multiplier: 1.45**

These values represent various levels of privacy protection, with lower epsilon values and higher noise multipliers providing stronger privacy guarantees. To evaluate each model's privacy-utility trade-off, we tracked test accuracy across 100 epochs and plotted the results to visualize the relationship between privacy level and accuracy.

4.2.1 Results of Gaussian Mechanism

To evaluate the impact of differential privacy on model accuracy, we trained the private federated learning models with different noise multiplier values in the Gaussian mechanism. The noise multipliers ranged from 0.43 (indicating low privacy) to 8.69 (indicating strong privacy). The models were compared against non-private centralized ML and non-private decentralized FL as baselines. The test accuracy over 100 epochs for each model is plotted in Figure 4.2.

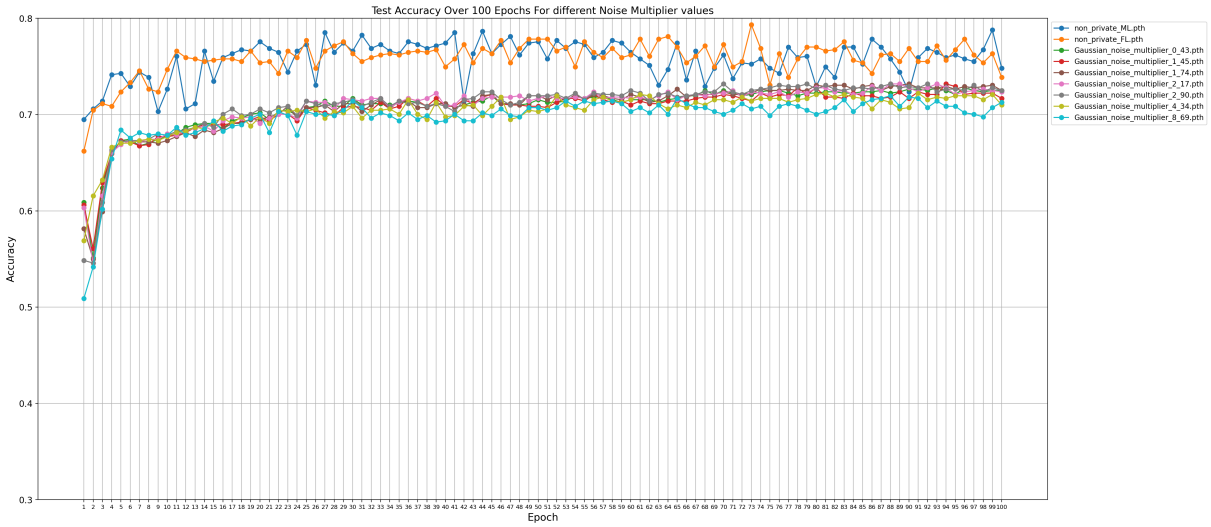

Figure 4.2: Test Accuracy Over 100 Epochs for Different Noise Multiplier Values in the Gaussian Mechanism

The plot shows the accuracy progression for each noise multiplier value, allowing for a clear comparison between different levels of privacy. The non-private ML and FL models serve as the benchmarks, achieving the highest accuracy without any privacy constraints.

From the plot, we observe that as the noise multiplier increases, the test accuracy generally decreases, which aligns with the expected trade-off between privacy and accuracy. Specifically:

- **Low Privacy (Noise Multiplier = 0.43)**: The model achieves a relatively high accuracy, comparable to the non-private models. However, the privacy level is low, making it less desirable in scenarios where strong privacy is required.
- **Moderate Privacy (Noise Multiplier = 1.45 and 1.74)**: These models offer a moderate level of privacy with only a slight decrease in accuracy compared to the low privacy setting.
- **Strong Privacy (Noise Multiplier = 2.17 and 2.90)**: These models demonstrate a stronger privacy guarantee with a minor trade-off in accuracy. Notably, the model with a noise multiplier of 2.90 achieves a very similar accuracy to the models with 1.74 and 2.17 while offering a higher privacy level.
- **Very Strong Privacy (Noise Multiplier = 4.34 and 8.69)**: As expected, these models offer the highest level of privacy but with a more significant reduction in accuracy. This reduction in performance indicates that very strong privacy comes at the cost of model effectiveness.

Given the theoretical guidance that suggests strong privacy when the noise multiplier is greater than 1.00, the model with a noise multiplier of 2.90 appears to provide the best trade-off. It maintains a level of accuracy close to that of models with lower privacy levels, while still offering strong privacy protection. Thus, for applications requiring a balance between privacy and model performance, a noise multiplier of 2.90 is recommended as the optimal choice.

These findings demonstrate that while very high privacy can significantly degrade model performance, a well-chosen noise multiplier, such as 2.90, can provide a strong balance between privacy and accuracy. This balance is crucial for real-world applications where both data privacy and model efficacy are paramount, fulfilling our first research goal of evaluating trade-offs in differential privacy.

4.2.2 Results of Laplace Mechanism

To further evaluate the impact of differential privacy on model accuracy, we trained the federated learning models with different epsilon values in the Laplace mechanism. The epsilon values ranged from 0.50 (indicating strong privacy) to 3.00 (indicating low privacy). The models were compared against non-private centralized ML and non-private decentralized FL as baselines. The test accuracy over 100 epochs for each model is plotted in Figure 4.3.

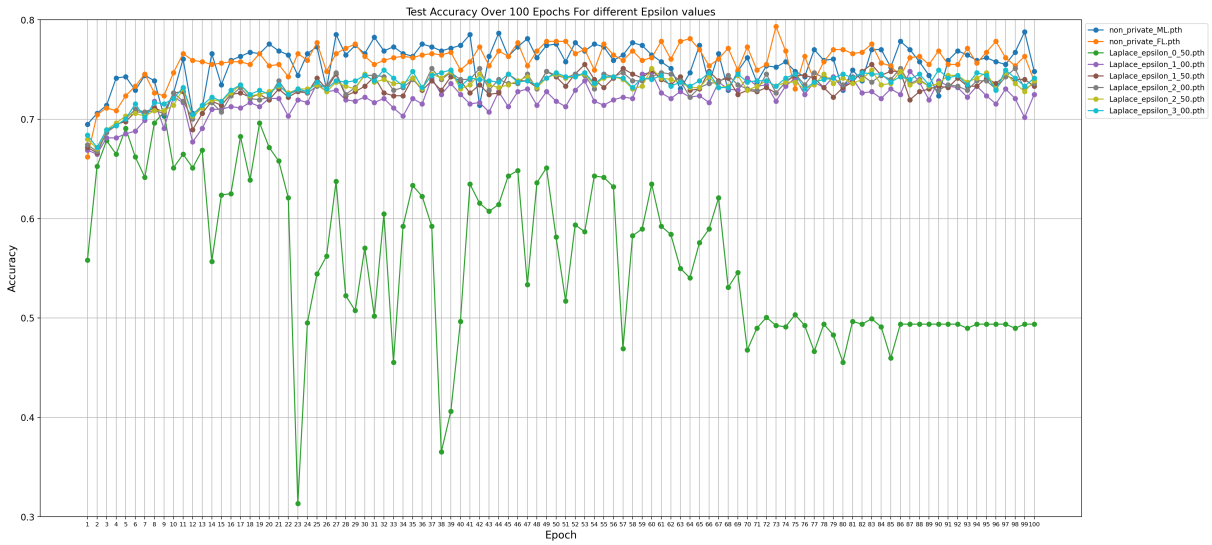

Figure 4.3: Test Accuracy Over 100 Epochs for Different Epsilon Values in the Laplace Mechanism

The plot illustrates the accuracy progression for each epsilon value, providing a clear comparison between different levels of privacy. The non-private ML and FL models serve as benchmarks, with the highest accuracy observed in the absence of privacy constraints.

From the plot, we observe the following trends:

- Strong Privacy (Epsilon = 0.50): The model with epsilon = 0.50 (represented by the green line) achieves a very high level of privacy but at the cost of significantly lower accuracy. While this setting maximizes privacy, it compromises the model's effectiveness, making it less suitable for applications where accuracy is paramount.

- Moderate Privacy (Epsilon = 1.00): The model with epsilon = 1.00 shows better accuracy compared to epsilon = 0.50 but still does not reach the accuracy levels of models with higher epsilon values.

- Low Privacy (Epsilon = 1.50 and above): The model with epsilon = 1.50, although it offers lower privacy, provides the best trade-off between privacy and accuracy. It outperforms the models with stronger privacy while still maintaining a level of privacy protection. Given the importance of accuracy in medical data applications, epsilon = 1.50 is identified as the most suitable choice for achieving a balance between privacy and accuracy.

The Laplace distribution has heavier tails compared to the Gaussian distribution, which means that while most of the noise values are close to zero, there is a higher probability of encountering larger noise values. Due to these heavier tails, the Laplace mechanism can occasionally introduce larger noise spikes, which might impact the stability of the federated learning process. Conversely, the Gaussian mechanism tends to introduce more consistent and less extreme noise, leading to smoother and potentially more stable learning processes in federated learning. This difference underscores the importance of selecting the appropriate mechanism based on the specific stability and privacy needs of the application.

According to the theoretical framework, lower epsilon values in the Laplace mechanism indicate higher privacy but result in reduced accuracy. However, the results indicate that when accuracy is a critical concern, especially in sensitive domains like medical data, a higher epsilon value such as 1.50 offers a better balance, ensuring the model remains effective while providing an acceptable level of privacy.

In conclusion, for scenarios where accuracy is crucial and some reduction in privacy can be tolerated, epsilon = 1.50 appears to offer the optimal balance, despite it being categorized as low privacy.

4.2.3 Comparison of Selected Models

After evaluating the trade-offs between privacy and accuracy for both the Gaussian and Laplace mechanisms, we identified the optimal settings for each: a noise multiplier of 2.90 for the Gaus-

sian mechanism and an epsilon of 1.50 for the Laplace mechanism. These settings provided the best balance between maintaining model accuracy and ensuring privacy.

To compare these final models against the non-private baselines, we plotted the test accuracy of the following models over 100 epochs:

- Non-private Centralized ML
- Non-private Decentralized FL
- Private FL using the Gaussian mechanism with a noise multiplier of 2.90
- Private FL using the Laplace mechanism with an epsilon value of 1.50

The results are shown in Figure 4.5.

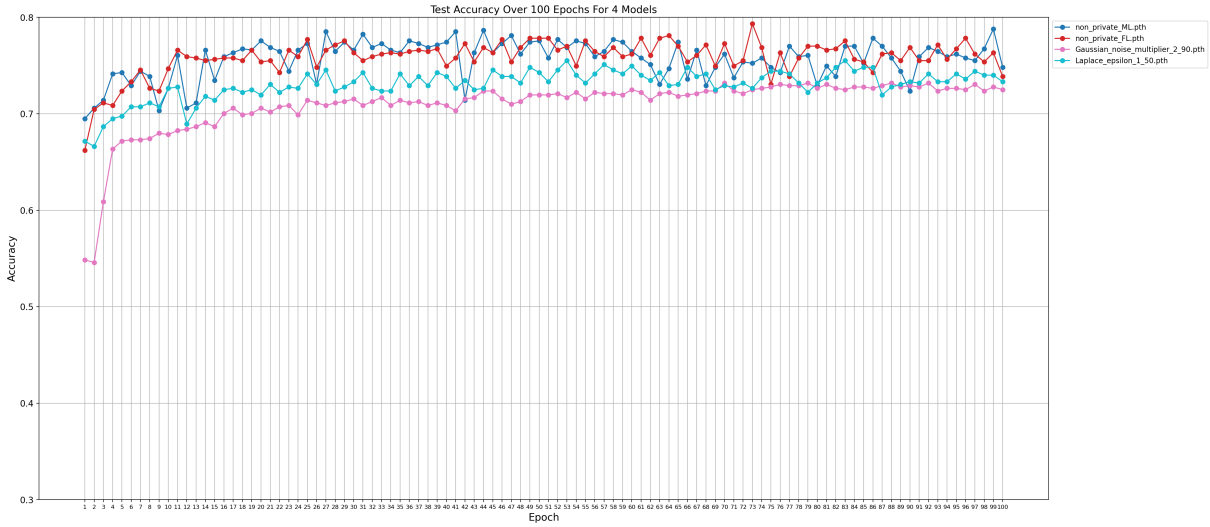

Figure 4.4: Comparison of Test Accuracy Over 100 Epochs for Non-private and Private Models (Gaussian with Noise Multiplier = 2.90, Laplace with Epsilon = 1.50)

The plot reveals the following insights:

- **Non-private Models**: As expected, the non-private centralized ML and decentralized FL models achieve the highest accuracy, with the centralized ML model performing slightly better.
- **Gaussian Mechanism (Noise Multiplier = 2.90)**: The private FL model using the Gaussian mechanism with a noise multiplier of 2.90 shows a slight decrease in accuracy compared to the non-private models. However, the trade-off is minimal, making it a viable option when privacy is a concern.

- **Laplace Mechanism (Epsilon = 1.50)**: The private FL model using the Laplace mechanism with an epsilon of 1.50 also exhibits a minor reduction in accuracy. Its performance is very close to that of the Gaussian mechanism with a noise multiplier of 2.90, reaffirming its suitability for scenarios where both privacy and accuracy are important.

The Laplace distribution has heavier tails compared to the Gaussian distribution, which means that while most of the noise values are close to zero, there is a higher probability of encountering larger noise values. Due to the heavier tails of the Laplace distribution, the Laplace mechanism can introduce larger noise spikes occasionally, which might impact the stability of the federated learning process. In contrast, the Gaussian mechanism tends to introduce more consistent and less extreme noise compared to the Laplace mechanism, which can lead to smoother and potentially more stable learning processes in federated learning.

In this study, we observed that the Gaussian model exhibited greater stability in accuracy, while the Laplace model had noticeable fluctuations, reflecting these distributional differences.

In summary, while the non-private models naturally outperform the private ones in terms of accuracy, the Gaussian mechanism with a noise multiplier of 2.90 and the Laplace mechanism with an epsilon of 1.50 offer the best trade-offs between privacy and accuracy. These models are recommended for use in sensitive applications, such as medical data processing, where privacy cannot be compromised but accuracy remains a priority. These findings not only demonstrate the effectiveness of differential privacy mechanisms in federated learning but also provide a foundation for future research into more robust and efficient privacy-preserving techniques.

Having identified the optimal trade-offs between accuracy and privacy for both Gaussian and Laplace mechanisms, we next evaluate how these settings withstand adversarial conditions. In the following section, we simulate inversion attacks to assess the robustness of the final models, particularly focusing on the impact of the selected noise multiplier and epsilon values on the reconstructed images.

4.3 Experimental Setup and Result for Inversion Attack Simulation

This section details the parameter values used in the inversion attack simulation to evaluate privacy robustness across various model configurations.

1. **Batch Size**: A batch size of 1 is used for each attack instance, ensuring that each

gradient corresponds to a single image. This configuration avoids gradient averaging, which could otherwise dilute the impact of the added noise. By using a batch size of 1, I achieve a clear representation of how each noise mechanism affects the reconstruction process on an individual image basis.

2. **Subset of Test Dataset**: A subset of 50 images from the diabetic retinopathy test dataset is used in the inversion attack, providing a representative sample for evaluating reconstruction accuracy.

3. **Model Architecture**: The inversion attack is conducted using a pretrained `SqueezeNet` model (`squeezenet1_1`) with the classifier layer modified to output 5 classes, consistent with the diabetic retinopathy dataset.

4. **Optimizer and Loss Function**: The `Adam` optimizer is used with a learning rate of `0.01`, and the `MSELoss` function is applied to measure the difference between model output gradients and target gradients.

5. **Gradient Files**: The following `.pth` files are used for inversion attacks, each representing a different model configuration:

- `non_private_ML.pth`
- `non_private_FL.pth`
- `Gaussian_noise_multiplier_2_90.pth` — (noise multiplier = 2.90)
- `Laplace_epsilon_1_50.pth` — (epsilon = 1.50).

6. **Model State Loading**: For each model configuration, `load_state_dict` is used to set the model to the corresponding gradient and weight state.

7. **Number of Images Saved**: Up to 6 pairs of original and reconstructed images are saved for each model configuration for visual analysis.

The next part of this section will present visual and quantitative evaluations, specifically focusing on Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) metrics, to quantify the effectiveness of each differential privacy mechanism.

4.3.1 Quantitative Comparison

To evaluate the privacy-preserving effectiveness of the different models, I computed Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Multiscale SSIM (MS-SSIM), Feature Similarity Index (FSIM), Perceptual Loss, and Gradient PSNR (G-PSNR)

for each reconstructed image against the original image. These metrics provide insights into the fidelity of the reconstructed images, where lower values for PSNR, SSIM, and MS-SSIM, and higher perceptual loss values generally indicate higher privacy due to reduced resemblance to the original data. The results for each model are summarized below, along with an in-depth analysis of how the privacy mechanisms affect image fidelity.

Figure 4.5: Comparison of Original and Reconstructed Images Across Different Model Configurations: Original Image, Non-Private ML, Non-Private FL, Private FL with Gaussian Mechanism, and Private FL with Laplace Mechanism.

- **Non-Private FL**:
  - **PSNR**: 27.90 dB
  - **SSIM**: 0.469
  - **MS-SSIM**: 0.773
  - **FSIM (as SSIM placeholder)**: 0.469
  - **Perceptual Loss (VGG)**: 0.064
  - **Gradient PSNR (G-PSNR)**: 17.14
  - **Interpretation**: The non-private FL model achieves the highest PSNR and SSIM among the FL models, with relatively high MS-SSIM and Gradient PSNR, indicating strong intensity and structural fidelity. These high values suggest that without noise, FL captures detailed information across diverse client updates, which is then retained in gradients. The relatively low perceptual loss value reflects good high-level feature retention, while a G-PSNR of 17.14 indicates reasonable preservation of edge details. In non-private FL, gradients remain largely unmodified, allowing the model to retain high-quality reconstructions with minimal information loss.
- **Non-Private ML**:
  - **PSNR**: 27.07 dB
  - **SSIM**: 0.457

- **MS-SSIM**: 0.766
- **FSIM (as SSIM placeholder)**: 0.457
- **Perceptual Loss (VGG)**: 0.068
- **Gradient PSNR (G-PSNR)**: 16.94
- **Interpretation**: The non-private ML model has slightly lower PSNR, SSIM, MS-SSIM, and G-PSNR values than non-private FL, indicating minor quality degradation. This is likely because centralized ML gradients lack the richness of multi-round, multi-client updates found in FL, resulting in less detail and lower reconstructive quality. The perceptual loss value is slightly higher, reflecting minor perceptual differences, while the G-PSNR of 16.94 indicates somewhat less edge preservation than non-private FL.

- **Private FL with Gaussian Mechanism**:
  - **PSNR**: 26.24 dB
  - **SSIM**: 0.421
  - **MS-SSIM**: 0.749
  - **FSIM (as SSIM placeholder)**: 0.421
  - **Perceptual Loss (VGG)**: 0.078
  - **Gradient PSNR (G-PSNR)**: 15.94
  - **Interpretation**: The Gaussian noise model shows reduced quality in all metrics compared to non-private models. The lower PSNR and SSIM suggest a loss of intensity and structural details, while the lower MS-SSIM and G-PSNR indicate reduced structural and edge preservation. The perceptual loss is higher (0.078), meaning that perceptual quality is impacted, introducing more noticeable visual distortions. The reduction in all these metrics demonstrates the privacy-preserving effect of Gaussian noise, which helps obscure fine details, effectively reducing the risk of data reconstruction.

- **Private FL with Laplace Mechanism**:
  - **PSNR**: 25.86 dB
  - **SSIM**: 0.477
  - **MS-SSIM**: 0.764

- **FSIM (as SSIM placeholder)**: 0.477
- **Perceptual Loss (VGG)**: 0.058
- **Gradient PSNR (G-PSNR)**: 18.49
- **Interpretation**: The Laplace noise model has a PSNR lower than non-private models, indicating intensity degradation. However, SSIM and MS-SSIM values are comparatively high for a privacy-preserving model, suggesting that structural features, particularly edges, are preserved better than with Gaussian noise. The low perceptual loss (0.058) suggests close high-level feature alignment with the original, and the higher G-PSNR (18.49) implies better edge retention. This aligns with the characteristic of Laplace noise, which can obscure pixel intensity while preserving edges, making it suitable where structural retention is prioritized.

To provide a clearer comparison of the fidelity and privacy-preserving effectiveness of each model configuration, we present the quantitative metrics for Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Multiscale SSIM (MS-SSIM), Feature Similarity Index (FSIM), Perceptual Loss, and Gradient PSNR (G-PSNR) in Table 4.1. This table allows for a side-by-side assessment of how each model impacts the reconstructed image quality relative to the original, offering insights into the trade-offs between image fidelity and privacy preservation. Lower PSNR and SSIM values, combined with higher Perceptual Loss, generally indicate a greater reduction in visual similarity, suggesting enhanced privacy.

Table 4.1: Quantitative Metrics for Reconstructed Images Across Model Configurations

Model Configuration	PSNR (dB)	SSIM	MS-SSIM	FSIM	Perceptual Loss	Gradient PSNR
Non-Private FL	27.90	0.469	0.773	0.469	0.064	17.14
Non-Private ML	27.07	0.457	0.766	0.457	0.068	16.94
Private FL (Gaussian)	26.24	0.421	0.749	0.421	0.078	15.94
Private FL (Laplace)	25.86	0.477	0.764	0.477	0.058	18.49

4.3.1.1 Why FL Has Higher Reconstructed Quality Than ML

The non-private FL model exhibits the highest quality among all FL setups, achieving the highest PSNR, SSIM, MS-SSIM, and perceptual similarity scores among the privacy-preserving models. This outcome is explained by factors inherent to the FL training dynamics:

- **Richer Gradients through Multiple Rounds and Client Diversity**: In FL, the model updates from diverse client data introduce a broader perspective on the global dataset. This diversity enriches gradients with detailed information, which aids in reconstructing images during an inversion attack.

- **Effect of Gradient Matching on Reconstruction**: Gradient matching in inversion attacks exploits gradient granularity, and FL gradients contain multi-client information, resulting in a richer signal than single-dataset ML. This richness explains the better reconstructive quality observed in non-private FL.
- **Privacy Implications of Non-Private FL**: Non-private FL lacks noise or secure aggregation, which means gradients may leak detailed information. Without privacy-preserving noise, FL gradients allow for higher-quality reconstructions, posing a potential privacy risk.

4.3.1.2 Solutions for Reducing Information Leakage in Non-Private FL

To mitigate leakage risks from gradients in non-private FL, the following techniques are beneficial:

1. **Gradient Compression and Sparsification**: Reduces shared information by retaining only significant gradient components.
2. **Gradient Clipping**: Prevents outliers from revealing identifiable data points by capping gradients.
3. **Adding Noise (Differential Privacy)**: Applying Gaussian or Laplace noise obscures details while preserving utility.
4. **Secure Aggregation**: Ensures that only aggregated gradients are accessible, reducing individual data exposure.
5. **Knowledge Distillation**: Shares only high-level outputs rather than raw gradients, complicating data reconstruction attempts.

By employing these methods, FL systems can achieve a balance between model performance and privacy requirements, shielding sensitive data from inversion attacks while retaining the advantages of federated training.
In summary, the **non-private FL model** demonstrates the highest reconstructed quality due to rich gradient information from multi-client aggregation, but poses potential privacy risks. Both **private mechanisms (Gaussian and Laplace)** effectively degrade reconstructed image quality, with **Gaussian noise consistently reducing quality** and **Laplace noise occasionally preserving structural details** due to edge retention properties. These findings illustrate the privacy-fidelity trade-offs for differentially private FL models.

4.3.2 Visual Comparison

To visually assess the impact of differential privacy mechanisms on image reconstruction, we compare six reconstructed images from each model configuration with the original dataset image. This comparison helps illustrate how well each model retains or degrades visual details, especially sensitive structures in the retinal images.

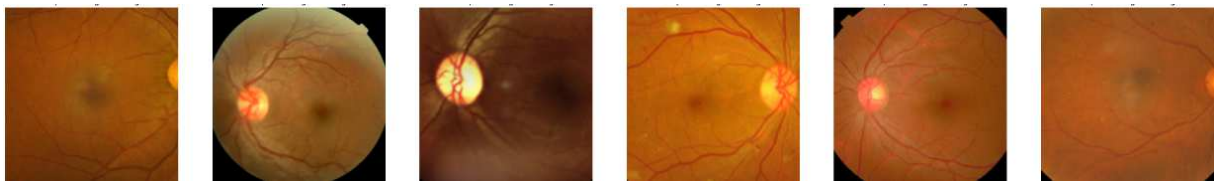

Figure 4.6: Original Image from the dataset

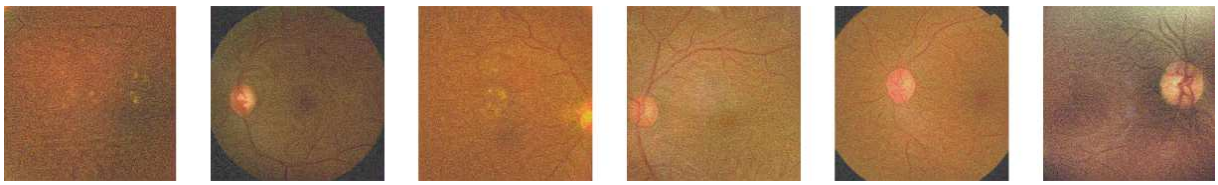

Figure 4.7: Reconstructed Image from Non-Private ML Model

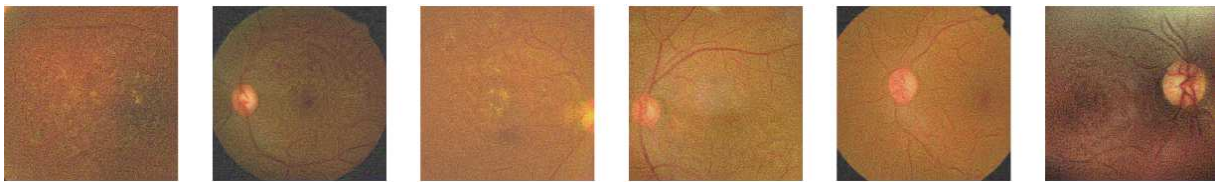

Figure 4.8: Reconstructed Image from Non-Private FL Model

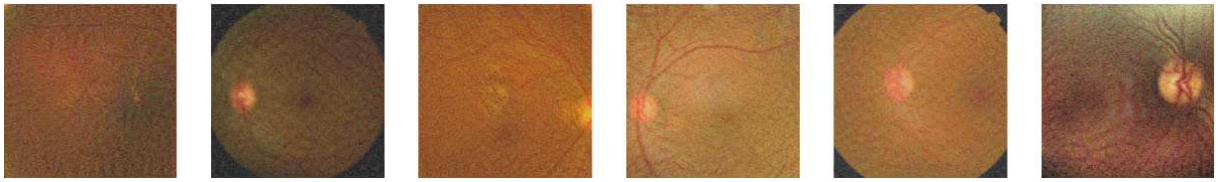

Figure 4.9: Reconstructed Image from Private FL with Gaussian Mechanism

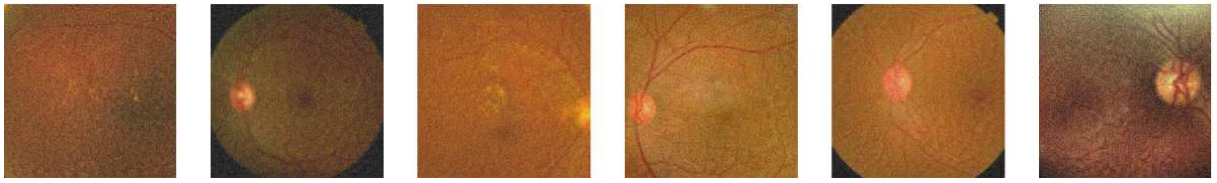

Figure 4.10: Reconstructed Image from Private FL with Laplace Mechanism

As demonstrated in the **Quantitative Comparison** section, we observed that each model's reconstructed image exhibits varying levels of fidelity to the original, influenced by the type

and level of privacy-preserving noise added. Higher levels of noise, as seen in the Private FL models with Gaussian and Laplace mechanisms, contribute to a greater reduction in image quality, which aligns with the lower PSNR and SSIM values. These quantitative metrics indicate a loss of similarity to the original image, supporting the privacy-preserving goals of the differentially private FL configurations.

Upon closer inspection of the visual images presented here, it becomes evident how these privacy mechanisms affect the retinal structures. The original image maintains all anatomical details, including fine blood vessels and retinal textures, which are critical for medical analysis. However, in the non-private ML and FL images, these structures are still relatively preserved due to the absence of noise, resulting in higher similarity to the original. In contrast, the images processed with Gaussian and Laplace noise show varying degrees of blurring and slight distortions, especially noticeable in smaller vessels and more delicate textures.

If we zoom in on the reconstructed images from the private FL models (Gaussian and Laplace), we can more clearly observe the visual impact of noise. Gaussian noise introduces a more uniform blurring effect, which effectively obscures intricate details while still preserving general shapes and structural features. Laplace noise, on the other hand, tends to retain edges better, meaning that while overall intensity and finer textures are affected, some structural details, such as the outlines of larger vessels, remain visible. This aligns with our quantitative analysis, where Laplace noise had slightly higher SSIM values than Gaussian noise, indicating a modest preservation of structural similarity.

These visual observations provide a complementary perspective to the quantitative metrics, illustrating that while the Gaussian and Laplace mechanisms effectively reduce the reconstructive quality of the image to enhance privacy, they each impact the image in distinct ways. This combined quantitative and qualitative approach reinforces the understanding of privacy-fidelity trade-offs in federated learning for medical imaging.

Chapter 5

Conclusions

5.1 Summary of Contributions

This thesis has presented a comprehensive investigation into privacy-preserving federated learning (FL) for diabetic retinopathy (DR) diagnosis, focusing on the integration of differential privacy (DP) mechanisms within a federated framework to achieve a balance between diagnostic accuracy and patient data protection. The research was motivated by the critical need for privacy-aware solutions in healthcare, where patient data is sensitive and highly regulated. By introducing and evaluating DP techniques—specifically, Gaussian and Laplace noise mechanisms—in FL, this study aimed to adDRess the dual objectives of privacy and model utility.

The major contributions of this thesis can be summarized as follows:

- **Development of a Privacy-Preserving FL Framework for Medical Imaging**: A federated learning framework was established, tailored for DR diagnosis using medical images. This setup allowed decentralized training across simulated client devices, preserving data locality and reducing privacy risks associated with centralizing sensitive patient information.

- **Application of Differential Privacy Mechanisms**: The study explored the efficacy of two DP mechanisms (Gaussian and Laplace) in mitigating privacy risks during model updates. By systematically adjusting privacy parameters (e.g., epsilon in Laplace and noise multiplier in Gaussian), this research demonstrated how different configurations impact both model accuracy and privacy levels.

- **In-Depth Analysis of Privacy-Utility Trade-Offs**: Extensive experiments were con-

ducted to quantify the impact of DP on model performance and privacy preservation, identifying optimal parameter settings for each mechanism. Visual and quantitative assessments of inversion attacks further validated the privacy protection offered by these mechanisms, showing how noise impacts image reconstruction quality and enhances data security.

- **Implementation of Robust Evaluation Techniques for Privacy and Accuracy**: By employing inversion attacks and assessing reconstructed images with metrics such as PSNR, SSIM, and perceptual loss, this study provided a robust methodology for evaluating the privacy efficacy of DP mechanisms in FL, offering a replicable framework for similar studies.

5.2 Key Findings and Implications

The findings from this research have important implications for both academic and practical applications of privacy-preserving machine learning, particularly in healthcare:

- **Privacy-Accuracy Trade-Off**: The results reveal that a careful balance can be achieved between privacy and model utility by selecting appropriate noise levels in DP mechanisms. The Gaussian mechanism, with a noise multiplier of 2.90, and the Laplace mechanism, with epsilon set to 1.50, were shown to maintain diagnostic accuracy within an acceptable range while effectively protecting privacy.

- **Mechanism-Specific Characteristics**: The distinct characteristics of Gaussian and Laplace noise mechanisms highlight that the choice of privacy technique can depend on the specific privacy-utility requirements of the application. The Gaussian mechanism provided more stable noise distribution and accuracy, while the Laplace mechanism preserved structural details better in certain high-frequency image areas, an advantage for specific medical imaging tasks.

- **Applicability to High-Stakes Domains**: The successful implementation and evaluation of privacy-preserving FL for DR diagnosis demonstrate that these techniques can be applied to other sensitive domains within healthcare. This framework could serve as a foundation for further research and development, particularly for applications where high model accuracy and strict privacy regulations intersect.

5.3 Limitations and Future Work

While this study provides a solid foundation for privacy-preserving FL in healthcare, there are limitations that suggest directions for future research:

- **Scalability to Real-World FL Environments**: This research simulated an FL environment with limited client diversity and simplified data distributions. Future studies should expand to more realistic settings with heterogeneous data distributions and larger client pools, to better evaluate scalability and robustness.
- **Exploration of Alternative DP Mechanisms**: While Gaussian and Laplace mechanisms were effective, alternative DP techniques, such as Rényi differential privacy or hybrid DP models, may offer enhanced privacy-utility trade-offs. Future research could investigate these approaches within FL frameworks.
- **Adaptive and Context-Sensitive Noise Application**: Implementing adaptive noise mechanisms that vary according to data sensitivity or model update frequency could further optimize privacy and accuracy. Integrating such adaptive mechanisms could help balance privacy requirements with minimal accuracy degradation over time.

5.4 Concluding Remarks

This thesis has addressed a pressing challenge in the era of data-Driven healthcare: how to leverage advanced machine learning models while preserving patient privacy. By implementing and evaluating differential privacy within a federated learning framework for DR diagnosis, this research has demonstrated that robust privacy protections can coexist with high model utility, even in the sensitive context of medical imaging. The findings contribute to the field of privacy-preserving machine learning and lay the groundwork for deploying secure, efficient, and privacy-aware FL systems in real-world healthcare applications.

As healthcare continues to adopt machine learning at scale, privacy-preserving techniques will play an essential role in ensuring that these advances are sustainable, ethical, and compliant with regulatory standards. The results of this research underscore the feasibility and importance of privacy-preserving FL, marking a significant step toward trustworthy *Artificial Intelligence* (AI) applications that can safeguard individual rights while advancing healthcare innovation.

Acronyms

AI *Artificial Intelligence*

CNN *Convolutional Neural Network*

DP *Differential Privacy*

DR *Diabetic Retinopathy*

FL *Federated Learning*

ML *Machine Learning*

GDPR *General Data Protection Regulation*

HIPAA *Health Insurance Portability and Accountability Act*

GPU *Graphics Processing Unit*

CPU *Central Processing Unit*

IID *Independent and Identically Distributed*

non-IID *Non-Independent and Identically Distributed*

ReLU *Rectified Linear Unit*

NPDR *Non-Proliferative Diabetic Retinopathy*

PDR *Proliferative Diabetic Retinopathy*

PSNR *Peak Signal-to-Noise Ratio*

SSIM *Structural Similarity Index Measure*

FSIM *Feature Similarity Index Measure*

MS-SSIM *Multi-Scale Structural Similarity Index Measure*

G-PSNR *Gradient Peak Signal-to-Noise Ratio*

DP-SGD *Differentially Private Stochastic Gradient Descent*

VGG *Visual Geometry Group*

GPU *Graphics Processing Unit*

CUDA *Compute Unified Device Architecture*

MSE *Mean Squared Error*

IoT *Internet of Things*

EHR *Electronic Health Record*

HE *Homomorphic Encryption*

MPC *Secure Multi-Party Computation*

PFL *Personalized Federated Learning*

PDF *probability density function*

IoT *Internet of Things*

SCAFFOLD *stochastic controlled averaging for federated learning*

Bibliography

[1] Mohammad Malekzadeh, Burak Hasircioglu, Nitish Mital, Kunal Katarya, Mehmet Emre Ozfatura, and Deniz Gündüz. Dopamine: Differentially private federated learning on medical data. *The Second AAAI Workshop on Privacy-Preserving Artificial Intelligence (PPAI-21)*, 2021.

[2] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 2014.

[3] Kobbi Nissim, Sofya Raskhodnikova, and Adam D. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, 2007.

[4] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference (TCC)*, pages 265–284, 2006.

[5] Borja Balle, Gilles Barthe, Marco Gaboardi, and Justin Hsu. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning (ICML)*, 2018.

[6] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016.

[7] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[8] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[9] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[10] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep models under the gan: Information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017.

[11] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. idlg: Improved deep leakage from gradients. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020.

[12] World Health Organization. Diabetic retinopathy: A leading cause of blindness, 2020.

[13] Waleed Nazih, Ahmad O. Aseeri, Osama Youssef Atallah, and Shaker El-Sappagh. Vision transformer model for predicting the severity of diabetic retinopathy in fundus photography-based retina images. *IEEE Access*, 2023.

[14] Michael A. Gulshan and colleagues. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*, 316(22):2402–2410, 2018.

[15] Feng Shi and colleagues. Performance of machine learning models in detecting diabetic retinopathy: A meta-analysis of the literature. *Ophthalmology*, 126(8):1100–1110, 2019.

[16] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. *IEEE Symposium on Security and Privacy (SP)*, 2008.

[17] Craig Gentry. Fully homomorphic encryption using ideal lattices. *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, 2009.

[18] Micah J Sheller, Guido A Reina, Brandon Edwards, Jeffrey Martin, and Spyridon Bakas. Multi-institutional deep learning modeling without sharing patient data: A feasibility study on brain tumor segmentation. In *International MICCAI Brainlesion Workshop*, 2020.

[19] Duy Nguyen, Ming Ding, Pubudu N Pathirana, and Aruna Seneviratne. Federated learning for covid-19 detection with generative adversarial networks in chest x-ray images. *IEEE Internet of Things Journal*, 8(7):4938–4951, 2021.

[20] Georgios Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. Secure, privacy-preserving and federated machine learning in medical imaging. *Nature Machine Intelligence*, 2020.

[21] Asuman Ozdaglar Alireza Fallah, Aryan Mokhtari. Personalized federated learning: A meta-learning approach. *34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, 2020.

[22] Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and H Brendan McMahan. Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[23] Pingfan Tang, Venkata Gandikota, Yang Liu, and Haishan Wang. Privacy-preserving federated learning framework for multi-site ocular disease diagnosis using gaussian noise. *IEEE Transactions on Medical Imaging*, 2023.

[24] Jan Dijk, Shuo Xu, and Li Wu. Asynchronous federated learning with reduced number of rounds and with differential privacy from less aggregated gaussian noise. In *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2020.

[25] Nicolas Papernot, Shuang Song, Ilya Mironov, Aditi Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.

[26] Xiaoming Gu, Yijun Li, and Tian Zhang. Efficient trajectory data protection using laplace mechanism in mobility analytics. *Journal of Big Data Analytics*, 2018.

[27] Jinhao Zhou, Zhou Su, Jianbing Ni, Yuntao Wang, Yanghe Pan, and Rui Xing. Personalized privacy-preserving federated learning: Optimized trade-off between utility and privacy. *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022.

[28] Hua He and Zheng He. Minimum gaussian noise variance of federated learning in the presence of mutual information based differential privacy. *IEEE Access*, 2023.

[29] Hesham El-Sayed Muhammad Talha Zia, Manzoor Ahmed Khan. Application of differential privacy approach in healthcare data – a case study. *14th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates*, 2020.

[30] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference (TCC)*, 2006.

[31] Sanxiu Jiao, Lecai Cai, Xinjie Wang, Kui Cheng, and Xiang Gao. A differential privacy federated learning scheme based on adaptive gaussian noise. *Computer Modeling in Engineering & Sciences*, 2023.

[32] Fang Liu. Statistical properties of sanitized results from differentially private laplace mechanism with univariate bounding constraints. *Trans. Data Priv.*, 2016.

[33] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[34] Varun Gulshan, Lily Peng, Marc Coram, Markus C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22):2402–2410, 2016.