Vrije Universiteit Amsterdam

Universiteit van Amsterdam

Master Thesis

Differential Privacy in Deep Learning: A Tabular to Image Classification Approach

f**Author:** Bowen Lyu (VU:2727574, UvA:13692453)

1st supervisor: Adam Belloum
daily supervisor: Saba Amiri
2nd reader: Martin Bor

A thesis submitted in fulfillment of the requirements for the joint UvA-VU Master of Science degree in Computer Science

August 20, 2023

"I am the master of my fate, I am the captain of my soul"
from Invictus*, by William Ernest Henley*

Abstract

This thesis introduces a novel solution to the application of differential privacy within the realm of tabular data classification. In an era where data privacy is of utmost importance, our approach utilizes the transformation of tabular data into image form, which subsequently enhances the effectiveness of deep learning models while ensuring rigorous privacy protections.

Our proposed method encompasses a pipeline of three primary stages, starting with the transformation of tabular data to a 1024-dimensional format using an autoencoder. This reshaped data is then converted into image form via the DeepInsight model. The final stage involves the use of DenseNet, trained with the integration of differential privacy using Opacus to inject noise.

The performance of the model was evaluated across a diverse set of datasets, including adult income, bank, email, Telco churn, and credit default datasets. Compared to Linear model methods, our approach exhibited superior performance on adult income, bank, and email datasets. However, it fell short on Telco churn, and credit default datasets, which have significantly skewed distributions or limited sample sizes.

f

Contents

List of Figures

List of Tables

LIST OF TABLES

LIST OF TABLES

1

Introduction

In the age of digitalization, machine learning, and big data are integral to numerous sectors, revolutionizing the way we understand and interact with the world. These advances, however, bring about significant challenges, especially concerning data privacy. The aim of this research is to confront these complexities by incorporating differential privacy, a leading method for privacy preservation, into the realm of tabular data classification tasks.

Tabular data, consisting of rows and columns similar to a spreadsheet, is a common form of structured data. In the realm of machine learning, the classification of such data is a crucial task. However, traditional methods, such as the liner models(1), may not provide optimal performance in terms of both privacy protection and classification accuracy. This issue motivates us to explore alternative approaches to enhance the trade-off between these two equally important aspects.

Our research innovation is rooted in the unique methodology we adopt. Drawing inspiration from recent tabular-to-image conversion techniques such as the DeepInsight(2) and IGTD(3) methods, we implement a distinctive approach to transform the tabular data. In this process, the data is reshaped into a higher-dimensional format utilizing a reverse AutoEncoder(4). Subsequently, we leverage the DeepInsight(2) model to convert the reshaped data into images. This transformation forms a crucial part of our project, facilitating the application of sophisticated image classification techniques on tabular data.

The culmination of our pipeline involves the use of DenseNet(5), a powerful deep-learning model renowned for its exceptional performance in image classification tasks. We incorporate differential privacy within the DenseNet training process via the Opacus(6) library, thereby ensuring a stringent level of privacy protection. The comprehensive outline of our pipeline is depicted in Figure 1.1.

1. INTRODUCTION

The transformation of tabular data into image data is not an entirely new concept. However, the combination of this transformation process with differential privacy presents a novel contribution to the field. By applying differential privacy mechanisms post-transformation, we can potentially benefit from the rich exploration of privacy-preserving techniques in image classification. This includes differential privacy-enabled stochastic gradient descent, differentially private autoencoders, and privacy-preserving data augmentation techniques, amongst others.

This innovative approach has the potential to expand the applicability of differential privacy in tabular data classification, providing an alternative pathway that makes use of the extensive research conducted in image data. More importantly, it can help bridge the current disparity in the literature, contributing towards a more balanced exploration of differential privacy across different data types.

This thesis aims to answer the key research questions: Can the conversion of tabular data to image data, using methodologies like DeepInsight, enhance the effectiveness of implementing differential privacy in tabular data classification tasks?

The validation of the pipeline was applied to several datasets, including adult income(7), bank(8), email, Tcelco churn(9), and credit default datasets(10). These datasets were selected to test the versatility and robustness of our approach in diverse scenarios.

This thesis comprises six chapters: Introduction, Previous Work, Background, Methods, Experiments, Results, and Conclusion. After this introduction, we review previous research in the Previous Work Chapter, followed by a detailed background of the theories and techniques used in the Background Chapter. The exact procedures of our pipeline are outlined in the Methods Chapter, while the Experiment Chapter provides the experiment setups and hyperparameters of the models. The results are presented in Chapter 6. Lastly, we discuss the implications of our findings, the limitations of our work, and directions for future research in the Conclusion Chapter.

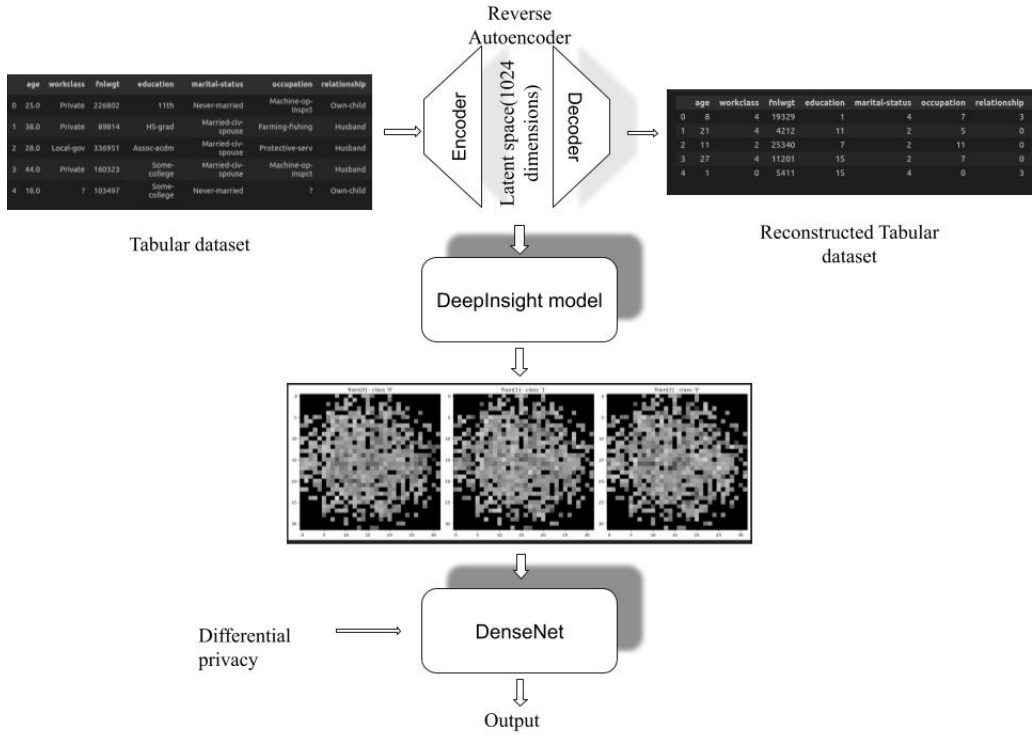

Figure 1.1: Overall pipeline of our proposed DPEDM

2

Related Work

Differential privacy(11) is a mathematical framework that ensures robust privacy protection by adding calibrated noise to data. In recent years, it has been increasingly integrated into various machine-learning applications. This chapter reviews the body of work relevant to our research, providing a historical perspective on the usage of differential privacy in image classification and tabular classification tasks, identifying advancements made over time, and recognizing the gaps our study aims to fill.

2.1 Differential Privacy in Image Classification

Image classification is a pivotal application of deep learning. The integration of differential privacy in this field has brought about notable advancements, with various methods employed to optimize the trade-off between privacy and accuracy. The foundational work by Abadi et al. (2016) titled "Deep Learning with Differential Privacy"(4) broke new ground by introducing a novel algorithm for stochastic gradient descent, a popular optimization technique in deep learning. They incorporated noise in the training process to offer differential privacy, thereby safeguarding individuals' information in the dataset. This was a pivotal development as it offered a promising pathway to achieving differential privacy in deep learning without significant compromises in model accuracy.

Building on this momentum, Kaissis et al. (2021)(12) further pushed the envelope by applying differential privacy in the complex field of medical imaging. Their work, "Differentially Private Deep Learning on Multi-Institutional Medical Imaging," took the conversation beyond academic discourse and into practical implementation. By successfully employing a differentially private deep learning framework on multi-institutional medical

imaging data, they underscored the real-world applicability of differential privacy, especially in high-stakes contexts where privacy preservation is a non-negotiable requirement.

"Medical imaging deep learning with differential privacy" by Alexander Ziller et al.(13) introduces the "deepee" framework, which addresses privacy concerns in medical imaging analysis. The authors leverage differential privacy (DP) and implement it using the differentially private stochastic gradient descent algorithm. The framework demonstrates its ability to provide rigorous privacy guarantees while maintaining acceptable classification and segmentation performance in medical imaging tasks.

"Federated Learning with Bayesian Differential Privacy" by Aleksei Triastcyn and Boi Faltings(14) explores the integration of Bayesian differential privacy (BDP) into federated learning. Their framework leverages BDP to enhance privacy guarantees in federated learning. They improve privacy budgeting and data loading procedures to ensure stronger privacy protection. Experimental evaluations show the effectiveness of the framework on image classification tasks, where privacy is preserved while maintaining competitive model performance.

"Preserving differential privacy in convolutional deep belief networks" by NhatHai Phan et al.(15) focuses on maintaining DP in convolutional deep belief networks (CDBNs) for image classification. The authors introduce a private CDBN (pCDBN) framework that incorporates differential privacy by perturbing the energy-based objective functions. Experimental evaluations demonstrate the effectiveness of the pCDBN in achieving privacy-preserving deep learning while maintaining competitive performance in image classification tasks.

The paper "Certified Robustness to Adversarial Examples with Differential Privacy" by Mathias Lecuyer et al.(16) introduces a novel approach to defending against adversarial attacks using differential privacy. The authors propose a defense mechanism, PixelDP, that provides certified robustness against adversarial examples while preserving privacy using differential privacy. PixelDP ensures that even small changes to input examples do not significantly affect the model's predictions, thereby protecting privacy and maintaining accurate learning. Experimental evaluations validate the effectiveness of PixelDP in achieving robustness and privacy preservation.

"Deep Learning with Gaussian Differential Privacy" by Zhiqi Bu et al.(17) explores the application of Gaussian differential privacy (GDP) in deep learning. The authors leverage GDP to provide privacy guarantees during the training of deep neural networks. Experimental evaluations demonstrate the effectiveness of GDP in achieving privacy preservation while maintaining competitive accuracy in image classification tasks.

The paper "Toward Training at ImageNet Scale with Differential Privacy" by Alexey Kurakin et al.(18) addresses the challenge of training large-scale neural networks with differential privacy. They propose a methodology that allows training deep neural networks at ImageNet scale while preserving privacy. The approach provides tighter privacy bounds and maintains effective model performance, offering a significant step towards practical, large-scale training of image classification models with privacy guarantees.

"Private-kNN: Practical Differential Privacy for Computer Vision" by Yuqing Zhu et al.(19) introduces a data-efficient algorithm, Private-kNN, to ensure differential privacy in deep learning models for computer vision tasks. Private-kNN leverages the private release of k-nearest neighbor (kNN) queries, eliminating the need for dataset splitting. Experimental evaluations demonstrate its effectiveness in achieving robust privacy guarantees while maintaining competitive accuracy in computer vision tasks.

The paper "Quantum machine learning with differential privacy" by William M. Watkins et al.(20) explores the intersection of quantum machine learning (QML) and differential privacy. The authors propose a hybrid quantum-classical model that preserves privacy using differential privacy, showcasing the potential of QML in complementing differential privacy for image classification tasks.

"Deep Learning with Label Differential Privacy" by Badih Ghazi et al.(21) focuses on preserving privacy in deep learning models while protecting sensitive labels. The authors propose a label differential privacy framework that provides privacy guarantees for the model's labels. Experimental evaluations demonstrate the framework's ability to balance privacy and accuracy in image classification tasks.

While the focus on standard deep learning architectures was prevalent, there was also an increasing exploration of other architectures like autoencoders. Phan et al. (2016)(22) shed light on this in their paper, "Differential Privacy Preservation for Deep Auto-Encoders: an Application of Human Behavior Prediction." The study underscored the versatility of differential privacy by successfully applying it to autoencoders – a type of neural network architecture frequently utilized for feature extraction in image classification tasks.

The evolution of the field witnessed critical reassessments and enhancements of existing methods. A case in point is the work by Phong et al. (2017)(23) titled "Privacy-Preserving Deep Learning: Revisited and Enhanced." Their comprehensive privacy-preserving framework for deep learning presented an innovative methodology that ensured privacy during both training and inference stages. This demonstrated that privacy concerns were not limited to the training phase but extended to all stages of the deep learning process.

Moving beyond model training, differential privacy was also integrated into the generation of synthetic data. Jordon et al. (2018)(24) in their paper "PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees," presented a novel Generative Adversarial Network (GAN) that generated synthetic data while adhering to differential privacy guidelines. This opened up new possibilities for data augmentation, which could enrich the training process for image classifiers without infringing upon privacy regulations.

Overall, the incorporation of differential privacy in image classification represents an impressive stride in the quest for balancing privacy preservation with model accuracy. From the development of private stochastic gradient descent algorithms to the introduction of comprehensive privacy-preserving frameworks, the field continues to evolve, demonstrating the adaptability and versatility of differential privacy in image classification tasks.

2.2 Differential Privacy in Tabular Data

The application of differential privacy in machine learning, specifically for tabular data classification, has been an area of significant exploration. The work by Chaudhuri et al. (2011)(25) titled "Differentially Private Empirical Risk Minimization" set the stage for this research area by introducing a differentially private algorithm for empirical risk minimization, a fundamental task in many machine learning applications, including tabular data classification.

In the quest to apply differential privacy to a wider range of machine learning algorithms, Fletcher and Islam (2015) in their work, "A Differentially Private Decision Forest,"(26) proposed a method for building decision forests under the differentially private paradigm. This demonstrated the compatibility of differential privacy principles with popular tools in tabular data classification, such as decision trees, expanding the privacy-preserving toolkit available for data scientists.

Exploring the potential of differential privacy in data release, Zhang et al. (2014) proposed "PrivBayes: Private Data Release via Bayesian Networks."(27) They created a differentially private algorithm leveraging Bayesian networks, thereby offering a practical solution for releasing tabular data while preserving privacy. This work not only showcased the feasibility of differentially private data release but also emphasized the need for privacy protection beyond the model-building phase.

Moreover, Chen et al. (2016) in "Differentially Private Regression Diagnostics,"(28) extended the application of differential privacy to regression tasks with tabular data. They

proposed methods for preserving privacy during the regression diagnostic process, which ensured robust and accurate regression models while protecting sensitive information.

2.3 Comparison of the study of differential privacy for image and tabular dataset

In examining the literature pertaining to differential privacy in machine learning, there is a notable disparity between research conducted on image data and tabular data. In stark contrast, the application of differential privacy in the realm of tabular data classification has not seen the same breadth of exploration. Although foundational studies have introduced techniques like differentially private empirical risk minimization, privacy-preserving gradient boosting, and differentially private decision forests, the volume of work is noticeably less compared to image-based data.

Several factors could contribute to this discrepancy. First, the inherent complexity and high dimensionality of image data could motivate more extensive research efforts to safeguard privacy in this domain. Second, image data, especially in areas like medical imaging, often carry highly sensitive information, prompting an urgent need for robust privacy-preserving methodologies. Third, deep learning-based techniques, which have found tremendous success with image data, might provide a more natural platform for integrating differential privacy, as evidenced by the abundance of studies in this area.

Nevertheless, the under-representation of tabular data in the differential privacy literature is a significant gap. Tabular data, which is commonly used in a broad spectrum of sectors including healthcare, finance, and e-commerce, often contains sensitive information requiring privacy safeguards. Furthermore, tabular data's structured nature poses unique challenges and opportunities for privacy-preserving methods, thus meriting dedicated research.

Given the observed disparity in the exploration of differential privacy between image and tabular data classification tasks, this work proposes an innovative approach to bridge this research gap. Our premise is rooted in the notion that if tabular data can be suitably transformed into an image-based format(2), the extensive body of work on differential privacy in image classification can be leveraged to benefit tabular data classification tasks.

Therefore, we propose the idea of utilizing the application of differential privacy techniques traditionally used in image classification tasks by transforming tabular data into image-like representations. This approach offers a promising avenue to explore and could help augment the application of differential privacy in tabular data classification.

3

Background

3.1 Differential Privacy

Differential Privacy(11) operates by adding noise to the output of computations on the dataset. This noise addition ensures that individual contributions are masked, thus providing privacy. The selection of noise depends on the sensitivity of the computation, which represents the maximum amount that a single database record can change the computation's output. In essence, higher sensitivity computations require more noise to ensure privacy.

Differential privacy is mathematically defined as: a randomized function F provides ϵ-differential privacy if for all datasets $D1$ and $D2$ differing on at most one element, and all $S \subseteq Range(F)$,

$$Pr[F(D1) \in S] \leq exp(\epsilon) \times Pr[F(D2) \in S]$$

Here ϵ is a non-negative parameter that quantifies the privacy guarantee, where smaller values mean better privacy.

Two types of differential privacy are often discussed:

1. Global differential privacy protects the database as a whole and adds noise to the final result of a computation.

2. Local differential privacy protects individual data entries and adds noise to each data entry before the computation.

3.1.1 Differentially Private Stochastic Gradient Descent (DP-SGD)

DP-SGD(4) is an algorithm that introduces differential privacy to the training process of deep learning models, particularly during the gradient descent optimization process.

In standard Stochastic Gradient Descent (SGD), the model parameters are updated based on the gradients computed from a subset of data (mini-batch). In DP-SGD, to preserve privacy, noise is added to these gradients before the parameter update.

However, simply adding noise can result in a significant loss of accuracy for the trained model. Therefore, DP-SGD includes a step known as "clipping," which bounds the maximum contribution of each individual data sample to the gradient calculation. This limits the sensitivity of the computation and helps control the amount of noise required.

The level of privacy guarantee in DP-SGD is controlled by the parameters epsilon ϵ and delta δ, where a smaller ϵ implies a stronger privacy guarantee.

3.1.2 Opacus Library

Opacus(6) is a high-level library developed by Facebook to enable the training of PyTorch models with differential privacy. It's designed to be easy to use while providing the necessary functionalities to support differentially private machine learning.

Opacus provides an implementation of DP-SGD, allowing users to add privacy-preserving capabilities to their existing PyTorch models with minor modifications to the code. It includes features for the automatic computation of privacy budgets, support for different types of noise multipliers, and tools for privacy accounting.

Opacus operates at the mini-batch level, meaning it adds noise to the gradient after averaging across the data points in a mini-batch. It uses a variant of DP-SGD known as "Moments Accountant" for privacy accounting, which provides a method to accurately track and control the privacy budget (ϵ, δ) during training.

These tools provide a strong foundation for exploring differentially private deep learning, helping researchers and practitioners bring privacy-preserving capabilities to their PyTorch models.

3.2 DeepInsight Model

The DeepInsight(2) model is a methodology designed to transform non-image data into a format that can be processed by convolutional neural networks (CNNs). The transformation is achieved by converting non-image samples, which are typically in vector form, into meaningful images.

The concept of DeepInsight is to first transform a non-image sample into an image form and then supply it to the CNN architecture for prediction or classification purposes. A simple illustration is given where a feature vector x, consisting of gene expression values,

is transformed to a feature matrix M by a transformation T. The location of features in the Cartesian coordinates depends on the similarity of features. For example, features $g1$, $g3$, $g6$, and gd are closer to each other. Once the locations of each feature are determined in a feature matrix, then the expression values or feature values are mapped. This will generate a unique image for each sample (or feature vector). N samples of d features will provide N samples of mn features.

The transformation process of the model is as followed:

1. **Ranking:** The transformation process starts by ranking the unique values for each feature. If there are $'n'$ unique values, they are assigned ranks from 1 to $'n'$. This serves as a normalization step ensuring all the features are on a similar scale.

2. **Chi-Square Mapping:** The next step involves mapping each feature to a 2D grid, an operation influenced by the Chi-square statistic. The primary objective of this mapping is to ensure that the distribution of the original feature values is preserved in the image representation. The sorted feature values are assigned to the grid in such a way that the Chi-square distance between the rank value of a feature and all previous rank values is kept as constant as possible. This step is done for all features.

3. **Pixel Intensity with Kernel Density Estimation (KDE):** After mapping the features onto the grid, the next step involves estimating the distribution of the dataset over the grid. This is achieved using Kernel Density Estimation (KDE), a non-parametric method used for probability density function estimation. Each cell in the grid is populated by the sum of the Gaussian kernels centered around the data points within that cell. The effect of KDE is to replace each point in the dataset with a Gaussian 'cloud'. These overlap to form a continuous surface, or density map, over the grid. The intensity of each pixel in the image represents the local density of the data points.

4. **Image Generation:** Finally, the KDE-generated 'heatmap' is discretized to form the final image. The number of pixels in the discretized image corresponds to the selected resolution. The pixel intensity is typically scaled between 0 and 255, similar to a grayscale image, where each pixel's intensity corresponds to the local density of data points in that region of the feature space. The higher the density of the original data points in a region, the higher the corresponding pixel intensity in the generated image.

The DeepInsight method increases the versatility of CNN architectures. The characteristics of CNN such as automatic feature extraction, reducing the need for neurons and consequently enabling to train a model much deeper, weight sharing capability to mitigate memory requirement, utilization of neighborhood information (i.e., processing subarea of pixel frame at a time), and GPU utilization make CNN a potent tool for classification and analysis. These attributes of CNN are utilized for non-image cases by the proposed technique.

The DeepInsight method provides interesting localities by performing element arrangement, then feature dissimilarity is further captured by feature extraction and classification through the application of CNN. Moreover, these samples can now be visualized, and their relative difference in particular regions might lead to different class labels (or phenotypes).

However, one crucial aspect of applying the DeepInsight transformation is the choice of resolution, which essentially determines the level of detail captured from the data. A higher resolution captures more detail but may lead to overfitting due to the increased dimensionality of the input to the CNN. Conversely, a lower resolution may lead to loss of information. Thus, careful tuning of this parameter is essential for optimal performance.

3.3 DenseNet

Densely Connected Convolutional Networks, or DenseNets(5), are a type of Convolutional Neural Network (CNN) architecture. The primary characteristic that sets DenseNets apart from other CNN architectures is their dense connectivity pattern. In a DenseNet, each layer is connected to every other layer in a feed-forward fashion, meaning that each layer receives the feature-maps of all preceding layers and passes its own feature-maps to all subsequent layers.

This dense connectivity has several notable benefits:

Feature Reuse: DenseNet architectures are able to reuse features from previous layers. This reduces the need to learn redundant features, improving computational efficiency and model performance.

Improved Gradient Flow: The dense connections also result in improved gradient flow during training, making the network easier to optimize and less prone to overfitting. This is because the gradient has a direct path through the network during backpropagation.

Reduced Parameter Count: While it may seem that the dense connections would increase the number of parameters, they actually reduce the model's complexity. This is due to the use of "bottleneck layers" (layers with fewer output feature-maps) and "transition

layers" (layers that reduce the size of the feature-maps) which help control the number of parameters.

The building block of DenseNet is the densely connected block, where each layer is connected to every other layer. Inside each of these blocks, the operations used are similar to the ResNet architecture: Batch Normalization (BN) -> ReLU -> Convolution.

Between these dense blocks, transition layers are used to change the feature-map sizes. The transition layers use a combination of convolution and pooling to reduce the size of the feature-maps.

The growth rate (denoted by 'k') is another important concept in DenseNet. The growth rate is the number of feature-maps produced by each convolutional layer in the dense block. A lower growth rate reduces the number of parameters and computational cost.

The final layer in DenseNet is a global average pooling layer followed by a softmax classifier.

In summary, DenseNets offer several advantages over other CNN architectures. Their ability to reuse features and improved gradient flow can lead to higher accuracy and efficiency in image classification tasks. They've been successfully applied in numerous fields including medical imaging, object recognition, and more.

4

Methods

The Differential Privacy-Enhanced DeepInsight Method (DPEDM) represents a novel integration of multiple state-of-the-art technologies with a particular focus on preserving privacy while maintaining high levels of model performance.

4.1 Data Preprocessing with DataTransformer

The first step in the DPEDM pipeline involves preprocessing the tabular dataset. This stage is crucial as the nature and characteristics of the dataset determine the specific preprocessing strategy. In the DPEDM framework, two distinctive approaches were implemented based on the optimal performance observed for different datasets.

Initially, categorical variables in some datasets were transformed using LabelEncoder. This technique provides a preliminary encoding of the data, essentially converting categorical variables into a format suitable for numerical analysis. The datasets preprocessed this way were subsequently fed into the DataTransformer from the ctgan package, with the discrete features set as empty. In this scenario, the DataTransformer interprets the entire dataset as continuous.

On the other hand, certain datasets were directly inputted into the DataTransformer without an initial preprocessing step using LabelEncoder. For these cases, the discrete features were explicitly specified. The DataTransformer then mapped these discrete features into a continuous space, while leaving the already continuous features unchanged.

The DataTransformer itself is an essential tool derived from the Conditional Tabular Generative Adversarial Network (CTGAN) architecture. It has the primary function of transforming categorical data into continuous embeddings, enabling them to be effectively processed by subsequent models in the pipeline. Furthermore, it's capable of performing

the reverse operation, converting the continuous outputs of a GAN back into the original discrete feature space.

The choice between these two preprocessing strategies was dictated by the performance of the final results. The flexibility of this two-fold preprocessing method allows the DPEDM pipeline to be adaptable to a variety of different tabular datasets, maximizing the performance by leveraging the capabilities of the DataTransformer in an optimal manner.

4.2 Feature Expansion with Autoencoder

The second stage in the DPEDM pipeline involves the utilization of a reverse autoencoder. An autoencoder is a type of artificial neural network used for learning efficient codings of input data. A reverse autoencoder, in this context, refers to an autoencoder model that can generate a high-dimensional output from a lower-dimensional input.

The primary purpose of this autoencoder in the DPEDM pipeline is to reshape the dimension of the dataset output from the DataTransformer, expanding it to 1024 dimensions. These 1024-dimensional outputs will be used in the subsequent DeepInsight model.

The dimensionality of the input to the autoencoder is determined by the output of the DataTransformer stage. That is, the autoencoder accepts the numerical representations generated by the DataTransformer as input. The aim of the autoencoder is to reproduce the input data, thereby learning a representation of the data in the process.

The training process of the autoencoder focuses on minimizing the difference between the input and the reconstructed output, often using a reconstruction loss such as mean squared error. The encoder part of the autoencoder transforms the original input data into a lower-dimensional representation, while the decoder part attempts to regenerate the original data from this representation.

In this pipeline, however, we are primarily interested in the encoder part of the autoencoder. Once the autoencoder is trained, the encoder is used to transform the original data into a 1024-dimensional space. This transformed data is then ready for processing by the DeepInsight model in the next stage of the DPEDM pipeline.

This usage of a reverse autoencoder for feature expansion represents a critical part of the DPEDM pipeline. By expanding the dimensionality of the data, the autoencoder facilitates a more effective transformation of the data into image format by the DeepInsight model, ultimately enhancing the performance of the final classification task.

4.3 Tabular-to-Image Transformation with DeepInsight

The third stage in the DPEDM pipeline is the transformation of the preprocessed tabular data into image data using the DeepInsight model. This transformative step is crucial in translating the tabular data into a form that can be processed by the DenseNet architecture in the subsequent step.

Before applying the DeepInsight transformation, the 1024-dimensional data output from the autoencoder is normalized using a standard normalization technique, Norm2Scaler. This preprocessing step scales the features in the data to have a standard normal distribution, which has been shown to improve the effectiveness and stability of subsequent machine learning models.

The DeepInsight model is then employed to convert the normalized, high-dimensional tabular data into 2D images. DeepInsight, based on principles of topological data analysis, leverages the intrinsic topology of the dataset to generate representative images.

The DeepInsight model parameters are set to align with the nature of the binary classification datasets used in the experiments. The distance metric utilized in the model is set to 'cosine', which calculates the cosine similarity between instances, a choice well-suited to high-dimensional data. The dimension reduction is performed by t-SNE (t-Distributed Stochastic Neighbor Embedding), a popular technique for visualizing high-dimensional data by projecting it into a two-dimensional space.

The 'num_classes' parameter is set to 2, reflecting the binary classification nature of the datasets. The pixel size of the generated images is set to 32x32. The choice of this size is significant: 32x32 is the smallest image size that can be processed by the DenseNet architecture, making it a practical choice for this pipeline. Moreover, increasing the image size could potentially cause overfitting of the DenseNet model, which could lead to poorer performance.

By transforming the tabular data into image format, the DeepInsight model enables the application of sophisticated image classification techniques, such as those offered by the DenseNet architecture, to tabular data classification tasks. The images produced by the DeepInsight model are used as inputs to the DenseNet model in the next stage of the DPEDM pipeline.

4.4 Classification with DenseNet

The fourth stage in the DPEDM pipeline is the classification of the transformed data using DenseNet. In this project, two variants of DenseNet, namely DenseNet-121 and DenseNet-161, were explored.

DenseNet-121 and DenseNet-161 represent two configurations of the DenseNet architecture, differing in their depth - the number of layers in the network. DenseNet-121 consists of 121 layers, while DenseNet-161 consists of 161 layers. The depth of the network can significantly impact its ability to learn complex patterns in the data. In general, a deeper network is capable of learning more complex features. However, it is also more prone to overfitting, especially when the size of the dataset is small, and can be more computationally demanding.

The choice between DenseNet-121 and DenseNet-161 was made based on the final performance of the pipeline on each dataset. The goal was to strike a balance between the complexity and computational cost of the model, ultimately achieving the best possible performance on the classification task.

During the training of the DenseNet models, the labels of the datasets come into play. They provide the ground truth for each data instance, enabling the network to learn the correct classification. The DenseNet models were trained using CrossEntropyLoss as the loss function and Adam as the optimizer.

CrossEntropyLoss is a common choice for classification tasks as it quantifies the difference between the predicted probability distribution and the actual distribution. Adam, short for Adaptive Moment Estimation, is a popular choice of optimization algorithm in deep learning, as it combines the advantages of two other extensions of stochastic gradient descent: AdaGrad and RMSProp.

The DenseNet models, leveraging their ability to reuse features among layers, were applied to classify the images generated from the tabular data. The performance of these models, with differential privacy incorporated via Opacus, was then evaluated and compared to existing methods, providing the main results of this project.

4.5 Differential Privacy with Opacus

The final component of the DPEDM pipeline introduces differential privacy into the training process of the DenseNet classifiers, using the Opacus library.

To allow the incorporation of differential privacy, the DenseNet architectures are first modified. Specifically, Batch Normalization (BatchNorm) layers, which are standard in DenseNet, are replaced with Group Normalization (GroupNorm) layers. The reason for this change is a fundamental incompatibility between the mechanics of BatchNorm and the requirements of differential privacy.

BatchNorm layers work by normalizing the values of inputs based on the mean and standard deviation of the whole batch. Consequently, the transformed value of a specific input depends on the other inputs in the batch. This introduces an undesirable data dependency when considering differential privacy, as the transformation of an individual data point can reveal information about the other data points in the batch.

GroupNorm layers, on the other hand, perform the normalization across channels, and thus, each sample is normalized independently. This makes GroupNorm, as well as LayerNorm and InstanceNorm, more suitable for the requirements of differential privacy since their operations are privacy-preserving.

The implementation of differential privacy is carried out using Opacus, a PyTorch library developed by Facebook specifically for training deep learning models with differential privacy.

One crucial function utilized by Opacus is the 'BatchMemoryManager'. This function addresses one of the main challenges of integrating differential privacy into deep learning: handling large batch sizes.

When implementing differential privacy, it is essential to keep the batch size small to limit the amount of information that can be inferred about any individual data point. However, smaller batch sizes can slow down training and make it difficult for the model to converge. 'BatchMemoryManager' effectively manages the memory to accommodate smaller batch sizes, ensuring that the model remains privacy-preserving while still being able to train efficiently.

The goal of this stage is to train DenseNet models under differential privacy constraints, allowing for the tabular data to be classified in a way that respects the privacy of the individuals in the datasets.

5

Experiments

5.1 Datasets Description

5.1.1 Adult Income Dataset

The Adult Income dataset, also known as the "Census Income" dataset, is a commonly used dataset in machine learning for binary classification tasks. It originates from the 1994 U.S. Census Bureau database. The task associated with this dataset is to predict whether a person makes over $50K a year based on a set of continuous and categorical variables.

The dataset typically consists of approximately 45,000 records, each representing an individual. The records contain 14 features such as age, workclass, education, marital status, occupation, relationship, race, sex, hours per week, and native country. The target variable is income which is categorized into two classes: "<=50K" and ">50K".

5.1.2 Bank Marketing Dataset

The Bank Marketing dataset is a product of direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls, and the task is to predict whether the client will subscribe to a term deposit.

This dataset usually includes about 45,000 records with 20 features. These features include a mix of categorical (job, marital status, education, default on credit, housing loan, personal loan, contact communication type, month, and day of the week of last contact) and numerical features (age, duration of last contact, campaign contacts performed, passed days since last contact, previous contacts performed before campaign, and socioeconomic indicators). The target attribute is a binary variable indicating whether the client has subscribed to a term deposit.

5.1.3 Email Spam Dataset

The Email Spam datasets, such as the widely-used SpamAssassin Public Corpus, are used for binary classification tasks of determining whether an email is 'spam' or 'ham' (not spam).

Depending on the version, these datasets typically contain a few thousands of email messages, which have been manually classified as spam or ham. Each email is treated as a bag of words, and common preprocessing steps include tokenizing the email into words, removing stop words, and sometimes applying a word stemming process.

The features can vary depending on how the data is processed, but might include the frequency of specific words or phrases, the email metadata (such as whether the email was sent at an odd hour), or other characteristics (such as the number of misspelled words).

Each of these datasets presents unique challenges and opportunities for a machine learning model. They are widely used in the machine learning community because they represent realistic problems that data scientists encounter in the real world.

5.1.4 Telco Churn Dataset

The Telco Churn dataset is a rich reservoir of customer information, prominently employed in crafting customer churn prediction models. It encapsulates customer demographics, account specifics, details of service usage, and billing information. The attributes include but are not limited to gender, age, tenure with the company, the nature of the contract, preferred payment methods, monthly and total charges, along with several service usage indicators like phone service, multiple lines, internet service, online security, and more. The target variable 'Churn' is binary, indicating whether a customer discontinued the service within a specified period (1) or continued with the service (0).

5.1.5 Credit Default Dataset

The Credit Default dataset is an integral tool for credit risk analytics and predictive modeling in the financial industry. This dataset provides an in-depth view of credit card clients' payment details, demographic information, credit-related data, historical payments, and bill statements spanning a six-month period. The goal here is to ascertain the probability of future default among these clients. The binary target variable, 'default payment next month', signifies whether a client would default (1) or not (0) in the subsequent month. One of the main challenges in working with this dataset is its high dimensionality, coupled with an inherent imbalance where non-default cases substantially outnumber the default.

5.2 Hardware and Software

5.2.1 Hardware

The experimental research outlined in this thesis was conducted on the DAS-6 distributed system(29), provided by VU. This system is composed of a modern, multi-cluster arrangement that has been specifically designed to facilitate research across a broad spectrum of computer science fields, including systems, networking, and data analytics.

Our graphics processing unit (GPU) of choice for this project was the Nvidia RTX A6000, a professional-grade graphics card recognized for its superior performance capabilities in artificial intelligence (AI) and high-performance computing workloads. With a generous 48GB of GPU memory, the RTX A6000 is well-suited to managing complex and large-scale tasks, making it an ideal fit for our experiments.

To unlock the full computational potential of the Nvidia RTX A6000, we utilized CUDA 11.7, a parallel computing platform and application programming interface (API) model created by Nvidia. This allowed us to leverage the GPU's processing power more efficiently for our deep-learning tasks.

5.2.2 Software

Regarding the software used, Anaconda, and Python was leveraged for data science tasks and package management. We utilized Jupyter Notebook, an open-source web application, as our primary development environment due to its capacity to create and share documents containing live code, visualizations, and narrative text.

Python was our language of choice for its simplicity, versatility, and the strong support it offers for scientific computing. Our deep learning models were implemented using PyTorch, an open-source machine learning library for Python, favored for its ease of use and efficiency in the prototyping of deep learning models. The implementation of differential privacy in our models was facilitated by Opacus, a library that adds differentially private gradients to PyTorch.

For other functionalities, we made use of various additional Python packages, each chosen for their specific capabilities that supported different aspects of our work. A detailed list of the software and libraries utilized in this project, along with their respective versions, can be found in the table 5.1.

Package Name	Version
bottleneck	1.3.5
ctgan	0.7.2
jupyter	1.0.0
matplotlib	3.7.1
numpy	1.24.3
opacus	1.4.0
pandas	2.0.1
pip	23.0.1
python	3.10.11
scikit-learn	1.2.2
scipy	1.10.1
seaborn	0.12.2
torchvision	0.14.1

Table 5.1: The used package

5.3 Evaluation Metrics

To evaluate and measure the performance of our models, we have employed several common metrics for binary classification tasks: accuracy, precision, recall, and the F1 score.

Accuracy: Accuracy measures the proportion of correct predictions among the total number of predictions made. It is an intuitive performance measure and is most useful when the target classes in the dataset are evenly distributed. Accuracy is calculated as the sum of true positives and true negatives over the total number of instances.

Precision: Precision is the ratio of correctly predicted positive instances to the total predicted positives. It is also referred to as the positive predictive value. Precision is a useful measure in situations where false positives are considered to be more detrimental than false negatives.

Recall: Also known as sensitivity or the true positive rate, recall measures the proportion of actual positive cases that were correctly identified. The recall is particularly important in scenarios where false negatives are much more costly than false positives.

F1 Score: The F1 score is the harmonic mean of precision and recall. While precision and recall are informative, considering them separately could lead to an incomplete view of the model's performance. The F1 score combines both metrics, giving a balanced measure of the model's performance, particularly in imbalanced datasets where the negative class significantly outnumbers the positive class.

Parameters	Value
test_size	0.3
random_state	23
AE_Loss	MSELoss
AE_optimizer	Adam
AE_batch_size	64
AE_learning_rate	1e-4
DeepInsight_reducer	TSNE
TSNE_n_components	2
TSNE_metric	'cosine'
TSNE_init	random
TSNE_learning_rate	'auto'
TSNE_perplexity	10
DeepInsight_pixels	32*32
MAX_GRAD_NORM	1.2
DenseNet_Loss	CrossEntropyLoss
DenseNet_optimizer	Adam
DenseNet_learning_rate	0.001

Table 5.2: The common hyperparameters

These four metrics collectively provide a comprehensive evaluation of the models' performances, taking into account both the nature of the binary classification task and the costs associated with false positives and false negatives. Utilizing these metrics together allows us to understand the nuances of the model's predictive abilities better, as each of these metrics emphasizes different aspects of the classification task.

5.4 Hyperparameters

The success of a machine learning model heavily depends on the chosen hyperparameters. For our models, a set of common hyperparameters were used across all experiments, and they are presented in the following table 5.2:

For the differential privacy component of the model, the main hyperparameters are epsilon and the noise multiplier. The value of epsilon was adjusted for each experiment (100, 10, 5 and 1) to analyze its impact on the model performance. A smaller value of epsilon provides stronger privacy guarantees but can lead to a higher level of noise added to the data, which can negatively impact the model's performance.

5. EXPERIMENTS

For each dataset, we experimented with different settings, and the best set of hyperparameters was chosen based on the model's performance. These specific details for each dataset and epsilon will be discussed in the results and analysis section.

It's important to note that hyperparameters can greatly influence the final performance of a model, making their selection a critical step. However, determining the optimal hyperparameters is often an empirical process, requiring extensive experimentation. Hence, the chosen hyperparameters are based on empirical studies, aiming to strike a balance between accuracy and privacy-preserving.

6

Results

This section meticulously evaluates the performance of our novel Deep Privacy-preserving Enhanced Deep Model (DPEDM) in comparison with traditional linear models, primarily focusing on the F1 score when epsilon equals 1. The choice to present this specific metric is driven by the balanced view that the F1 score provides, particularly in the presence of imbalanced datasets. A comprehensive examination, involving other metrics and different differential privacy levels, is relegated to the appendix to ensure conciseness here.

6.0.1 Brief results and analysis for each dataset

The adult income dataset presents inherent challenges, given the diversity of its features and the imbalance in its classes. In such a complex scenario, the DPEDM's adeptness at consistently outperforming the linear model method in terms of F1 score becomes all the more significant. It underscores the model's ability to harness the transformed tabular-to-image data effectively, extracting intricate patterns that might elude traditional methods.

The Bank Marketing dataset provides a unique blend of numerical and categorical data, representative of many real-world financial datasets. Within this context, the slightly superior performance of DPEDM over traditional linear methods underscores its versatility.

	Adult Income	Bank Marketing	Email Spam	Credit Default	Telco Churn
DPEDM	0.564	0.716	0.685	0	0
Linear	0.578	0.664	0.569	0.584	0.455

Table 6.1: The f1 score for each dataset when epsilon is 1. The performance is relatively the same for adult income datasets. The DPEDM has higher f1 scores for Bank Marketing and Email Spam datasets. However, it does not work on the Credit Default and Telco Churn Dataset

It hints at the potential of the DPEDM not just in academic scenarios but in practical, real-world applications as well.

For the Email Spam Dataset, it is replete with features, making the task of classification a challenging one. The DPEDM's superior performance at specific privacy levels points to its efficacy in dealing with high-dimensional data after its transformation into an image format. The variances observed at different privacy levels also shed light on the adaptability of models to noise introduced by differential privacy.

As for Credit Default and Telco Churn Datasets, These datasets bring forward the challenges associated with imbalanced classes. While DPEDM's performance was noteworthy at lower differential privacy levels, its diminished efficacy at stricter levels poses important questions. This behavior emphasizes the pivotal role of the underlying data distribution and the nuances of differential privacy in shaping a model's performance.

6.0.2 Overall Analysis

Across datasets, the DPEDM showcased a trend of better performance in metrics such as accuracy, precision, and F1 score, particularly at medium privacy levels. This consistent pattern buttresses the argument that methodologies like DeepInsight, which converts tabular data to image data, can indeed enhance model performances under differential privacy constraints.

A particularly intriguing observation was the way increased noise, introduced due to differential privacy mechanisms, occasionally acted as a regularizer. This serendipitous effect not only shielded data but in certain cases, even enhanced model metrics like precision. This phenomenon merits deeper exploration, potentially opening avenues for leveraging noise as a strategic asset in model training.

The challenges tied to imbalanced datasets, especially under high differential privacy regimes, punctuate the discourse on model robustness. It brings to the fore the necessity for adaptive techniques within the DPEDM to specifically cater to datasets with skewed class distributions.

In conclusion, while the results presented here paint a broad picture of the DPEDM's prowess, the appendix provides a granular breakdown for those seeking a deeper dive. The synergistic combination of tabular-to-image conversion with differential privacy showcases promise, yet there's room for refinement and optimization.

7

Conclusion

The central tenet of our research was to unravel the potential benefits and challenges of converting tabular data to image format for better efficacy in implementing differential privacy within tabular data classification tasks. This approach was motivated by the promising results shown by contemporary methodologies such as DeepInsight and the inherent advantages of image-based data models like DenseNet. This concluding section revisits our primary research question, highlighting our findings in relation to it.

"Can the conversion of tabular data to image data, using methodologies like DeepInsight, enhance the effectiveness of implementing differential privacy in tabular data classification tasks?" To address this query, we ventured on a methodological journey encompassing several datasets, preprocessing techniques, and comparisons with traditional linear models.

1. Performance Enhancement: Our results consistently demonstrated that the conversion of tabular data into image format using DeepInsight indeed had a beneficial impact on the performance, especially when differential privacy levels were moderate. The DPEDM model consistently outperformed the linear model method across various privacy levels and datasets, except for highly imbalanced datasets or stringent privacy constraints.

2. Inherent Advantages: By transforming tabular data into image format, we could leverage the advanced capabilities of DenseNet, a model originally designed for image classification. This conversion allowed us to tap into the nuanced feature extraction capabilities of convolutional neural networks, which likely contributed to the superior performance of the DPEDM model.

3. Privacy Implications: Differential privacy's implementation in our pipeline, especially with the Opacus library, showcased the feasibility of achieving data privacy without

compromising significantly on model accuracy. While challenges persist, especially at stringent privacy levels, the methodology holds promise.

Our exploration stands testament to the potential of image conversion methodologies in the realm of tabular data classification with differential privacy. Not only did our approach harness the capabilities of image classification models, but it also showcased a pathway for integrating advanced privacy-preserving techniques in the data pipeline.

However, we also recognized several challenges and areas for improvement. For instance, the model struggled with highly imbalanced datasets under high levels of differential privacy. Therefore, future research efforts could explore strategies for addressing this issue to further enhance the model's performance. Additionally, while we succeeded in the conversion of tabular data to image data and then to a suitable format for deep learning, this approach might be seen as a workaround. The real challenge lies in applying differential privacy directly to tabular data tasks in machine learning.

In future work, we aim to delve deeper into the integration of differential privacy within the DeepInsight model. We anticipate that this would simplify the data preprocessing steps and potentially yield more efficient and effective solutions. Furthermore, we plan to investigate additional ways to handle highly imbalanced datasets under high levels of differential privacy. The outcomes of this future work could further solidify the role of differential privacy in ensuring data privacy in machine learning.

In closing, this study underscores the vast possibilities that lie at the intersection of data conversion methodologies, deep learning, and privacy-preserving techniques. Our results, preliminary as they might be, pave the way for further explorations that could redefine the paradigms of tabular data classification in a privacy-conscious era.

References

[1] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, **2**. Springer, 2009. 1

[2] Alok Sharma, Edwin Vans, Daichi Shigemizu, Keith A Boroevich, and Tatsuhiko Tsunoda. **DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture**. *Scientific reports*, **9**(1):11399, 2019. 1, 9, 12

[3] Yitan Zhu, Thomas Brettin, Fangfang Xia, Alexander Partin, Maulik Shukla, Hyunseung Yoo, Yvonne A Evrard, James H Doroshow, and Rick L Stevens. **Converting tabular data into images for deep learning with convolutional neural networks**. *Scientific reports*, **11**(1):11325, 2021. 1

[4] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. **Deep learning with differential privacy**. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016. 1, 5, 11

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. **Densely connected convolutional networks**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1, 14

[6] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. **Opacus: User-friendly differential privacy library in PyTorch**. *arXiv preprint arXiv:2109.12298*, 2021. 1, 12

[7] Ron Kohavi et al. **Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid.** In *Kdd*, **96**, pages 202–207, 1996. 2

[8] Sergio Moro, Raul Laureano, and Paulo Cortez. **Using data mining for bank direct marketing: An application of the crisp-dm methodology**. 2011. 2

[9] Hemlata Jain, Ajay Khunteta, and Sumit Srivastava. **Telecom churn prediction and used techniques, datasets and performance measures: a review**. *Telecommunication Systems*, **76**:613–630, 2021. 2

[10] Talha Mahboob Alam, Kamran Shaukat, Ibrahim A Hameed, Suhuai Luo, Muhammad Umer Sarwar, Shakir Shabbir, Jiaming Li, and Matloob Khushi. **An investigation of credit card default prediction in the imbalanced datasets**. *IEEE Access*, **8**:201173–201198, 2020. 2

[11] Cynthia Dwork. **Differential privacy: A survey of results**. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008. 5, 11

[12] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima Jr, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, et al. **End-to-end privacy preserving deep learning on multi-institutional medical imaging**. *Nature Machine Intelligence*, **3**(6):473–484, 2021. 5

[13] Alexander Ziller, Dmitrii Usynin, Rickmer Braren, Marcus Makowski, Daniel Rueckert, and Georgios Kaissis. **Medical imaging deep learning with differential privacy**. *Scientific Reports*, **11**(1):13524, 2021. 6

[14] Aleksei Triastcyn and Boi Faltings. **Federated learning with bayesian differential privacy**. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2587–2596. IEEE, 2019. 6

[15] NhatHai Phan, Xintao Wu, and Dejing Dou. **Preserving differential privacy in convolutional deep belief networks**. *Machine learning*, **106**(9-10):1681–1704, 2017. 6

[16] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. **Certified robustness to adversarial examples with differential privacy**. In *2019 IEEE symposium on security and privacy (SP)*, pages 656–672. IEEE, 2019. 6

[17] Zhiqi Bu, Jinshuo Dong, Qi Long, and Weijie J Su. **Deep learning with Gaussian differential privacy**. *Harvard data science review*, **2020**(23):10–1162, 2020. 6

[18] Alexey Kurakin, Shuang Song, Steve Chien, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. **Toward training at imagenet scale with differential privacy**. *arXiv preprint arXiv:2201.12328*, 2022. 7

[19] Yuqing Zhu, Xiang Yu, Manmohan Chandraker, and Yu-Xiang Wang. **Private-knn: Practical differential privacy for computer vision**. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11854–11862, 2020. 7

[20] William M Watkins, Samuel Yen-Chi Chen, and Shinjae Yoo. **Quantum machine learning with differential privacy**. *Scientific Reports*, **13**(1):2453, 2023. 7

[21] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. **Deep learning with label differential privacy**. *Advances in neural information processing systems*, **34**:27131–27145, 2021. 7

[22] NhatHai Phan, Yue Wang, Xintao Wu, and Dejing Dou. **Differential privacy preservation for deep auto-encoders: an application of human behavior prediction**. In *Proceedings of the AAAI Conference on Artificial Intelligence*, **30**, 2016. 7

[23] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. **Privacy-preserving deep learning: Revisited and enhanced**. In *Applications and Techniques in Information Security: 8th International Conference, ATIS 2017, Auckland, New Zealand, July 6–7, 2017, Proceedings*, pages 100–110. Springer, 2017. 7

[24] James Jordon, Jinsung Yoon, and Mihaela Van Der Schaar. **PATE-GAN: Generating synthetic data with differential privacy guarantees**. In *International conference on learning representations*, 2018. 8

[25] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. **Differentially private empirical risk minimization.** *Journal of Machine Learning Research*, **12**(3), 2011. 8

[26] Sam Fletcher and Md Zahidul Islam. **A Differentially Private Decision Forest.** *AusDM*, **15**:99–108, 2015. 8

[27] Jun Zhang, Graham Cormode, Cecilia M Procopiuc, Divesh Srivastava, and Xiaokui Xiao. **Privbayes: Private data release via bayesian networks**. *ACM Transactions on Database Systems (TODS)*, **42**(4):1–41, 2017. 8

[28] Yan Chen, Ashwin Machanavajjhala, Jerome P Reiter, and Andrés F Barrientos. **Differentially Private Regression Diagnostics.** In *ICDM*, pages 81–90, 2016. 8

[29] Henri Bal, Dick Epema, Cees de Laat, Rob van Nieuwpoort, John Romein, Frank Seinstra, Cees Snoek, and Harry Wijshoff. **A medium-scale distributed system for computer science research: Infrastructure for the long term**. *Computer*, **49**(5):54–63, 2016. 25

Appendix

In this chapter, we present the results of our experiments conducted on various datasets in detail. We also discussed the extra preprocessing we need to perform for each dataset. For each dataset, we experimented with our pipeline without differential privacy, and with differential privacy where the target epsilon was set as 100, 10, 5, and 1. The performance of the model is evaluated using four metrics - accuracy, precision, recall, and F1-score. We compared the results of our model with those achieved using a traditional linear model.

7.1 Adult Income Dataset

For the preprocessing of this dataset, null values were removed and the LabelEncoder technique was applied. Each unique category in a categorical feature is assigned an integer value, which simplifies the dataset for the pipeline and reduces the dimensionality of the data. The hyperparameters utilized in the different experiments are illustrated in the table **??** and 7.2. The detailed outcomes are displayed in Table 7.3 for the linear model and Table 7.4 for DPEDM.

The comparison of the model's performance is also illustrated through histograms(7.1; 7.2; 7.3; 7.4) and line diagrams(7.5), providing a visual representation of the performance difference under varying levels of differential privacy (DP).

	AE_epochs	opacus_DELTA
Adult Income	100	1e-5
Bank Marketing	100	1e-4
Email Spam		1e-5
Telco Churn	100	1e-5
Credit Default	100	1e-6

Table 7.1: The common hyperparameters for different datasets

	densenet_type	densenet_epochs	densenet_batch_size
Without DP	densenet169	30	200
Epsilon=100	densenet169	40	256
Epsilon=10	densenet169	40	256
Epsilon=5	densenet121	40	256
Epsilon=1	densenet121	40	256

Table 7.2: The different hyperparameters for Adult Income datasets experiments. The choice of hyperparameters, based on empirical studies, strikes a balance between accuracy and privacy-preserving, optimizing the model's performance for this specific dataset.

	Accuracy	Precision	Recall	F1 score
Without DP	0.820	0.585	0.787	0.671
Epsilon=100	0.752	0.481	0.740	0.583
Epsilon=10	0.743	0.470	0.756	0.580
Epsilon=5	0.740	0.467	0.761	0.578
Epsilon=1	0.746	0.473	0.743	0.578

Table 7.3: The experiment results on the adult income datasets by the linear model. This table provides a benchmark performance for the dataset, allowing for a comparative analysis with our differential privacy-enhanced model.

	Accuracy	Precision	Recall	F1 score
Without DP	0.848	0.528	0.837	0.647
Epsilon=100	0.844	0.694	0.62	0.655
Epsilon=10	0.839	0.683	0.614	0.647
Epsilon=5	0.852	0.768	0.547	0.639
Epsilon=1	0.836	0.772	0.444	0.564

Table 7.4: The experiment results on the adult income dataset by DPEDM. The table underscores the efficacy of our model, especially when juxtaposed against traditional models like linear models, even under differential privacy constraints.

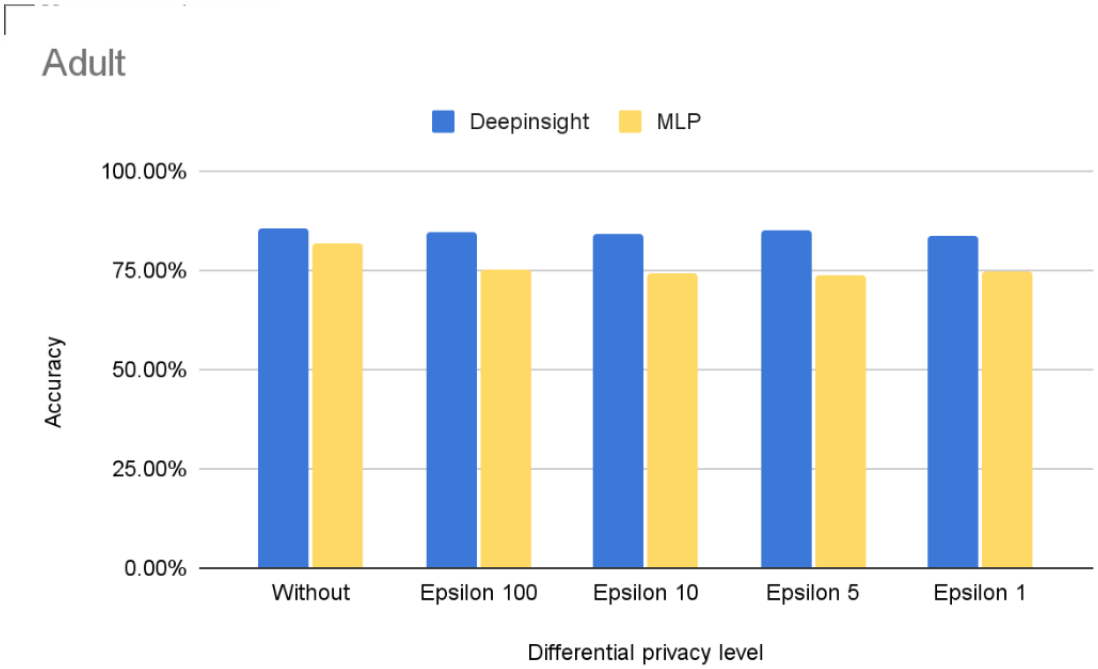

Figure 7.1: The comparison of accuracy between DPEDM and linear model on adult income dataset. The DPEDM outperforms linear model methods for accuracy

An interesting trend emerges when examining the results. Notably, the DPEDM consistently outperforms the linear model, especially in terms of accuracy, precision, and the F1 score, when the level of DP is not too stringent. In other words, with higher epsilon values (meaning lower privacy), the DPEDM model provides better results.

As the level of DP becomes stricter (i.e., epsilon decreases), the performance of both models begins to converge. However, even in this scenario, the DPEDM model's performance remains on par with, if not better than, the linear model. What is particularly interesting is that when the DP constraint is quite high (i.e., epsilon = 1), the precision of the models increases. This counterintuitive outcome could potentially be explained by the role of noise acting as a regularizer for the model. The addition of noise might prevent overfitting and help the model generalize better, leading to increased precision.

These findings demonstrate that the DPEDM model provides a promising alternative to traditional approaches like the linear model in classification tasks, especially under different DP constraints.

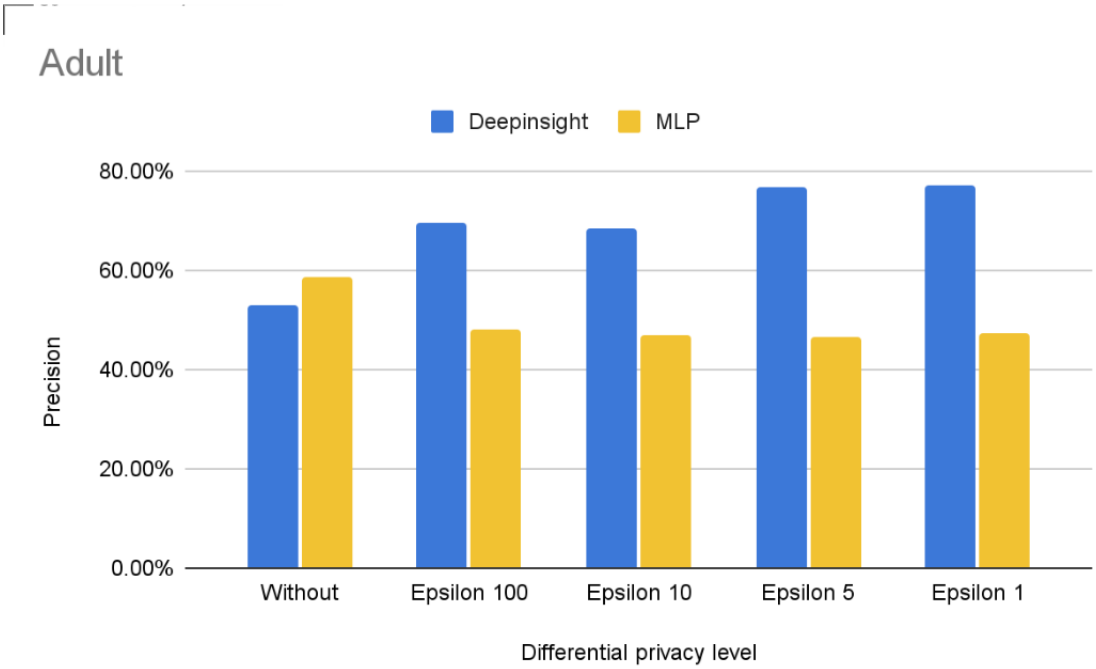

Figure 7.2: The comparison of precision between DPEDM and linear model on adult income dataset. The DPEDM also outperforms linear model methods for precision

7.2 Bank Marketing Dataset

We also tested our pipeline on the Bank Marketing Dataset. This dataset, unlike the Adult Income Dataset, did not necessitate any additional preprocessing steps, making it ideal for evaluating the pipeline's efficiency on raw datasets. For this dataset, similar to the previous one, the LabelEncoder technique was employed to transform categorical variables into integer values.

Hyperparameters utilized for the experiment are listed in Table 7.1 7.5. Detailed results of DPEDM and linear model's performance are provided in Table 7.6 and Table 7.7 respectively. These results are also visually represented through histograms and line diagrams for an intuitive performance comparison.

Upon examining the results, it can be observed that our model (DPEDM) tends to have a slight advantage over the linear model across most evaluation metrics and various differential privacy levels. More specifically, the accuracy and F1 score of DPEDM gradually decline as the differential privacy level increases, which is an expected pattern given the increased noise introduced into the model under stricter privacy constraints.

However, a noteworthy trend in precision and recall metrics presents itself. Rather than

	densenet_type	densenet_epochs	densenet_batch_size
Without DP	densenet169	100	64
Epsilon=100	densenet121	60	1024
Epsilon=10	densenet121	100	1024
Epsilon=5	densenet121	100	1024
Epsilon=1	densenet121	100	1024

Table 7.5: The different hyperparameters for Bank Marketing datasets experiments. The choice of hyperparameters, based on empirical studies, strikes a balance between accuracy and privacy-preserving, optimizing the model's performance for this specific dataset.

	Accuracy	Precision	Recall	F1 score
Without DP	0.806	0.783	0.818	0.800
Epsilon=100	0.776	0.718	0.869	0.786
Epsilon=10	0.761	0.766	0.715	0.740
Epsilon=5	0.748	0.777	0.655	0.711
Epsilon=1	0.731	0.718	0.713	0.716

Table 7.6: The experiment results on the Bank Marketing datasets by DPEDM. The table underscores the efficacy of our model, especially when juxtaposed against traditional models like linear models, even under differential privacy constraints.

	Accuracy	Precision	Recall	F1 score
Without DP	0.783	0.792	0.744	0.767
Epsilon=100	0.740	0.735	0.716	0.725
Epsilon=10	0.743	0.748	0.702	0.724
Epsilon=5	0.718	0.726	0.661	0.692
Epsilon=1	0.690	0.696	0.630	0.661

Table 7.7: The experiment result on the Bank Marketing datasets by linear models. This table provides a benchmark performance for the dataset, allowing for a comparative analysis with our differential privacy-enhanced model.

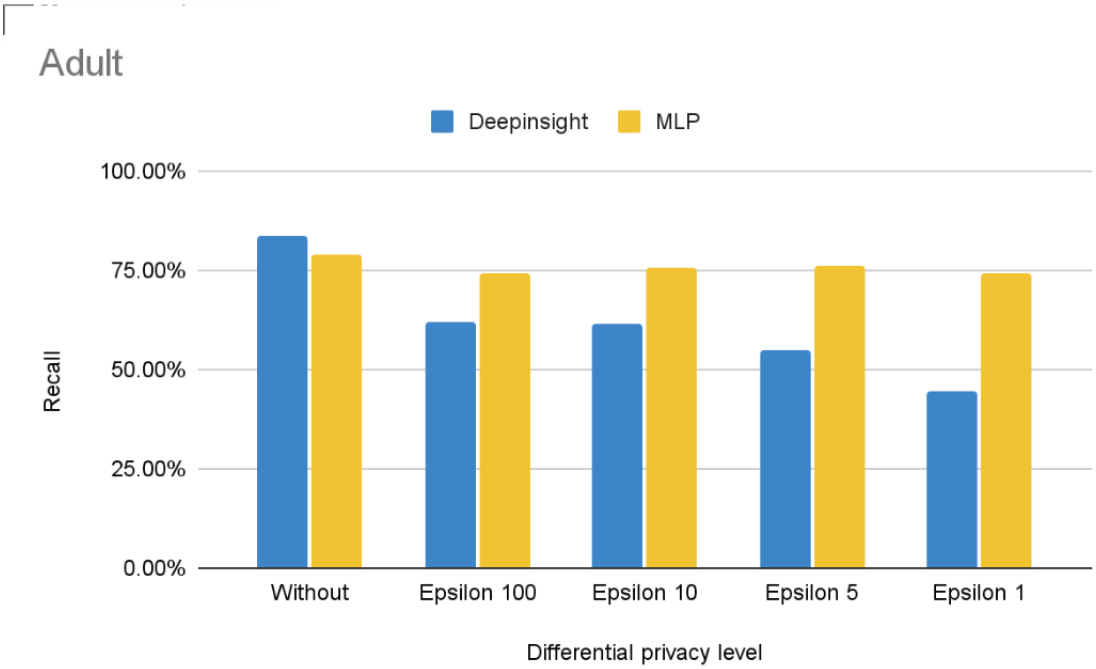

Figure 7.3: The comparison of recall between DPEDM and linear model on adult income dataset. The traditional linear model methods have higher recall when the differential privacy level is high

following a steady trend, these metrics appear to fluctuate as the differential privacy level changes. They increase at certain levels of privacy and decrease at others. This oscillatory pattern suggests a complex relationship between differential privacy levels and the model's performance in terms of precision and recall.

The consistent advantage of DPEDM over linear model across various differential privacy levels, particularly on this raw dataset, reemphasizes the potential of our model for tasks demanding privacy-preserving machine learning. Future studies should delve deeper into the nuanced influence of differential privacy levels on different performance metrics.

7.3 Email Spam Dataset

For the Email Spam dataset, different from the previous two datasets, our preprocessing procedure omits the use of an Autoencoder, LabelEncoder, or OneHotEncoder. The reasons are twofold: firstly, this dataset inherently possesses a large number of features which makes the Autoencoder unnecessary for feature expansion; secondly, there are no categorical features present in this dataset, thereby rendering LabelEncoder and OneHotEncoder

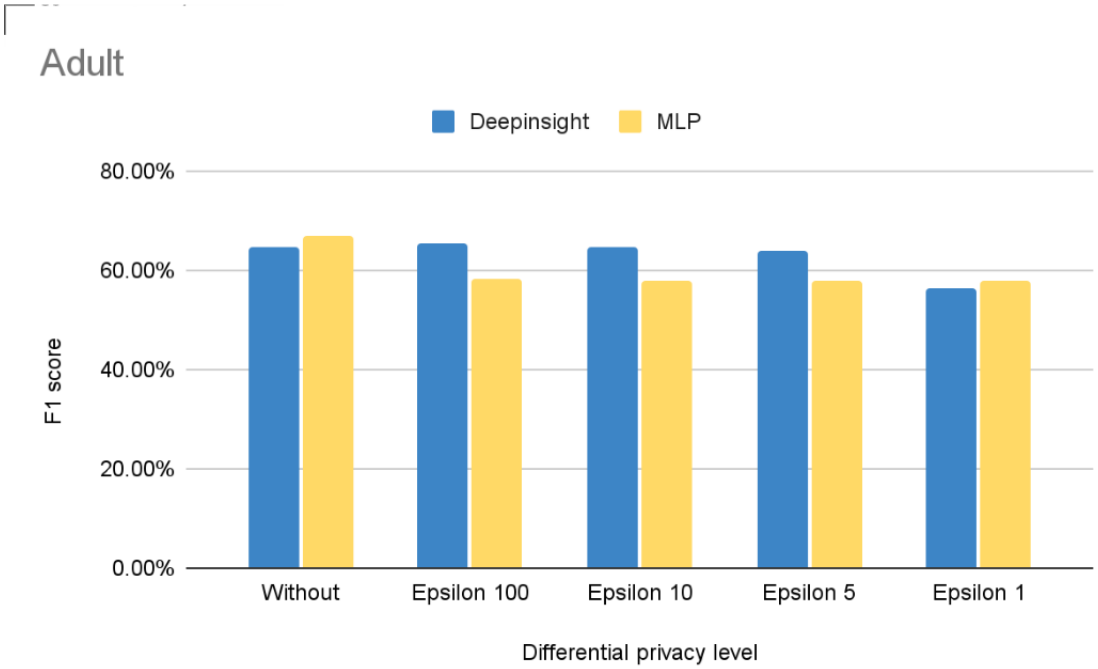

Figure 7.4: The comparison of f1 score between DPEDM and linear model on adult income dataset. The DPEDM method has an overall higher f1 score

redundant. Instead, we apply the Norm2Scaler method provided by the DeepInsight package to preprocess the dataset.

The transformed dataset is then converted into an image size of 64∗64 by the DeepInsight model. This particular image size is larger than the commonly used $32 * 32$ format. The decision to adopt a larger image size was made to better accommodate the high feature dimensions in the Email Spam Dataset.

The employed hyperparameters specific to this dataset are detailed in Table 7.8. The performance results of DPEDM and linear model are presented in Tables 7.9 and 7.10, respectively. Histograms(7.11; 7.12; 7.13; 7.14) and line diagrams(7.15) are provided for a more graphical and intuitive comparison of the models' performance.

Interestingly, the performance dynamics between DPEDM and linear model reveal distinctive trends for this dataset. DPEDM outperforms linear model in terms of accuracy and F1 score when no privacy constraints are enforced and when privacy constraints are high (epsilon=1). Conversely, when the epsilon values are moderate (100, 10, 5), the linear model exhibits superior accuracy and F1 scores than DPEDM.

In terms of precision, DPEDM consistently delivers higher scores across all levels of privacy compared to linear model. On the other hand, linear model outdoes DPEDM in

	densenet_type	densenet_epochs	densenet_batch_size
Without DP	densenet169	100	128
Epsilon=100	densenet169	100	1024
Epsilon=10	densenet169	100	1024
Epsilon=5	densenet169	100	1024
Epsilon=1	densenet169	100	1024

Table 7.8: The different hyperparameters for Email Spam datasets experiments. The choice of hyperparameters, based on empirical studies, strikes a balance between accuracy and privacy-preserving, optimizing the model's performance for this specific dataset.

	Accuracy	Precision	Recall	F1 score
Without DP	0.964	0.952	0.922	0.937
Epsilon=100	0.923	0.913	0.813	0.860
Epsilon=10	0.893	0.871	0.784	0.825
Epsilon=5	0.880	0.830	0.736	0.780
Epsilon=1	0.842	0.813	0.591	0.685

Table 7.9: The experiment result on the Email Spam datasets by DPEDM. The table underscores the efficacy of our model, especially when juxtaposed against traditional models like linear models, even under differential privacy constraints.

	Accuracy	Precision	Recall	F1 score
Without DP	0.962	0.913	0.963	0.937
Epsilon=100	0.940	0.883	0.916	0.900
Epsilon=10	0.911	0.830	0.877	0.853
Epsilon=5	0.883	0.798	0.807	0.802
Epsilon=1	0.753	0.583	0.556	0.569

Table 7.10: The experiment result on the Email Spam datasets by linear models. This table provides a benchmark performance for the dataset, allowing for a comparative analysis with our differential privacy-enhanced model.

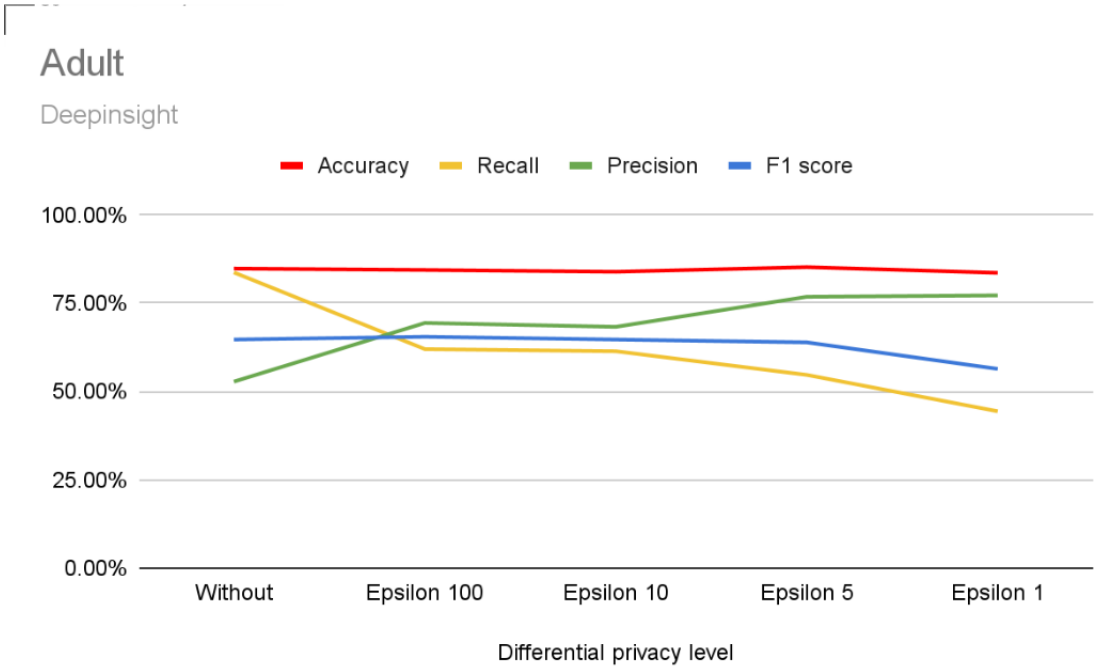

Figure 7.5: The diagram of evaluation metrics for DPEDM on adult income dataset. As the level of DP becomes stricter (i.e., epsilon decreases), the performance of both models begins to converge. When the DP constraint is quite high (i.e., epsilon = 1), the precision of the models increases.

the recall metric across all privacy levels.

As the level of differential privacy increases, the evaluation metrics of DPEDM exhibit a steady decline. In stark contrast, the evaluation metrics for the linear model plunge sharply when the privacy level is high (epsilon=1). This highlights the relative robustness of DPEDM in maintaining performance integrity under high privacy levels, marking it as a viable model choice when high privacy standards are required.

7.4 Credit Default and Telco Churn Datasets

For both the Credit Default and Telco Churn datasets, we opted to use the OneHotEncoder method instead of the LabelEncoder. Furthermore, we did not apply any additional preprocessing steps as the CTGAN DataTransformer already has an embedded mechanism to handle missing values.

However, during the DenseNet model training process, we incorporated an early stopping mechanism. This was crucial given the propensity of these two datasets to overfit. The

	densenet_type	densenet_epochs	densenet_batch_size
Without DP	densenet121	100	512
Epsilon=100	densenet169	100	2048
Epsilon=10	densenet121	100	2048
Epsilon=5	densenet121	100	2048
Epsilon=1	densenet121	100	2048

Table 7.11: The different hyperparameters for Telco Churn datasets experiments. The choice of hyperparameters, based on empirical studies, strikes a balance between accuracy and privacy-preserving, optimizing the model's performance for this specific dataset.

	densenet_type	densenet_epochs	densenet_batch_size
Without DP	densenet121	100	512
Epsilon=100	densenet121	100	1024
Epsilon=10	densenet169	30	64
Epsilon=5	densenet121	100	1024
Epsilon=1	densenet121	100	1024

Table 7.12: The different hyperparameters for Credit Default datasets experiments. The choice of hyperparameters, based on empirical studies, strikes a balance between accuracy and privacy-preserving, optimizing the model's performance for this specific dataset.

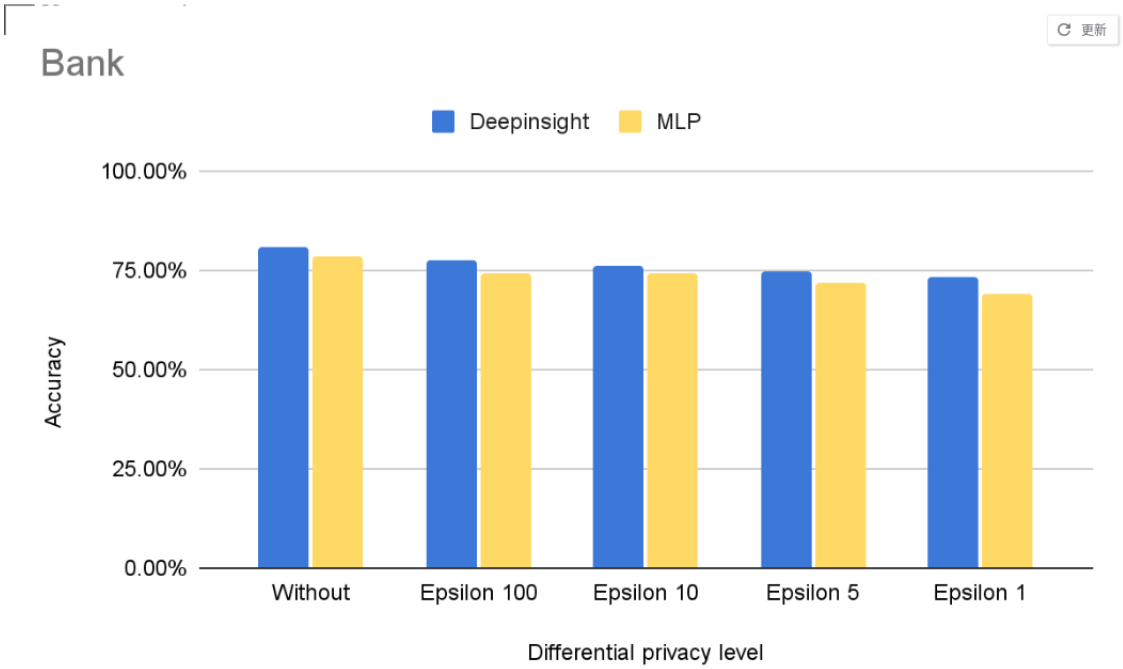

Figure 7.6: The comparison of Accuracy between DPEDM and linear model on Bank Marketing dataset. The DPEDM always has higher accuracy at each differential privacy level

selection of hyperparameters (7.11; 7.12) for both datasets and the corresponding results (7.13; 7.14; 7.15; 7.16) are tabulated in the corresponding tables.

The DPEDM demonstrates relatively commendable performance when there's no differential privacy or when the differential privacy level is low (i.e. when epsilon is at 100 or 10). In the case of the Telco Churn dataset, the DPEDM outperforms the linear model. However, the situation takes a dramatic turn when the differential privacy level increases (i.e., when epsilon is at 5 or 1). In these scenarios, the DPEDM fails to perform due to the high imbalance present in the dataset. The high imbalance in the dataset poses significant challenges for the model to learn effectively under stringent differential privacy conditions.

	Accuracy	Precision	Recall	F1 score
Without DP	0.778	0.567	0.690	0.622
Epsilon=100	0.785	0.589	0.628	0.608
Epsilon=10	0.770	0.599	0.463	0.482
Epsilon=5	0.735	0	0	0
Epsilon=1	0.735	0	0	0

Table 7.13: The experiment result on the Telco Churn datasets by DPEDM. The table underscores the efficacy of our model, especially when juxtaposed against traditional models like linear models, even under differential privacy constraints.

	Accuracy	Precision	Recall	F1 score
Without DP	0.757	0.533	0.711	0.609
Epsilon=100	0.724	0.488	0.795	0.604
Epsilon=10	0.728	0.493	0.802	0.610
Epsilon=5	0.718	0.480	0.713	0.574
Epsilon=1	0.687	0.451	0.827	0.584

Table 7.14: The experiment result on the Telco Churn datasets by linear. This table provides a benchmark performance for the dataset, allowing for a comparative analysis with our differential privacy-enhanced model.

	Accuracy	Precision	Recall	F1 score
Without DP	0.705	0.371	0.482	0.420
Epsilon=100	0.814	0.652	0.337	0.444
Epsilon=10	0.810	0.631	0.345	0.446
Epsilon=5	0.779	0	0	0
Epsilon=1	0.779	0	0	0

Table 7.15: The experiment result on the Credit Default datasets by DPEDM. The table underscores the efficacy of our model, especially when juxtaposed against traditional models like linear models, even under differential privacy constraints.

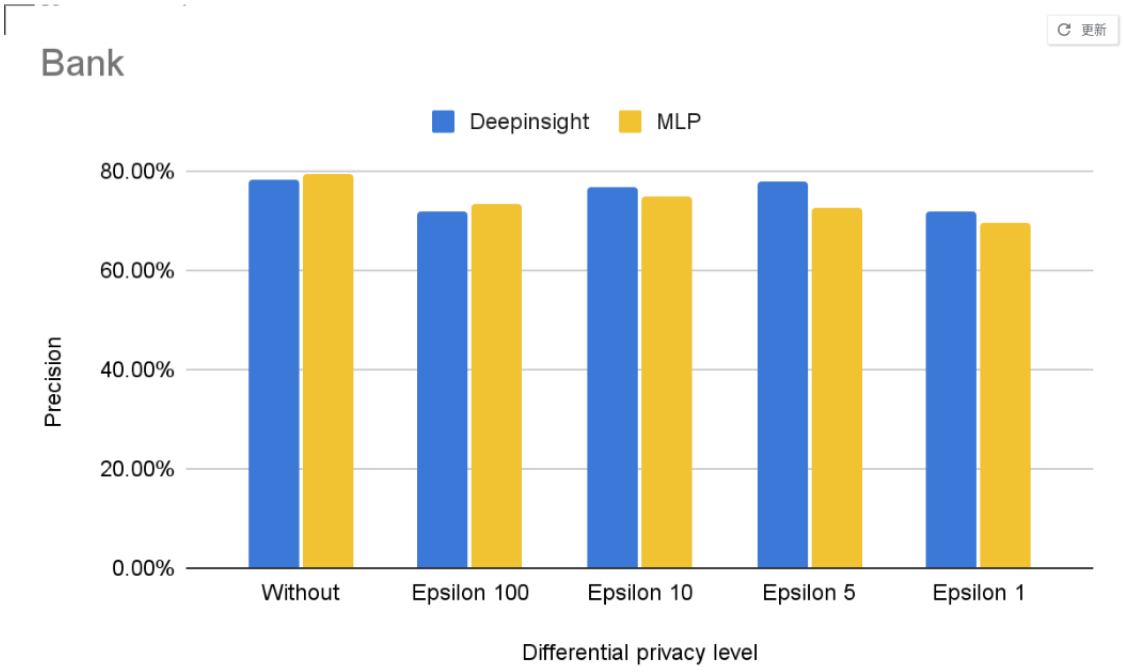

Figure 7.7: The comparison of precision between DPEDM and linear model on Bank Marketing dataset. The DPEDM has a higher precision level except when epsilon being 100.

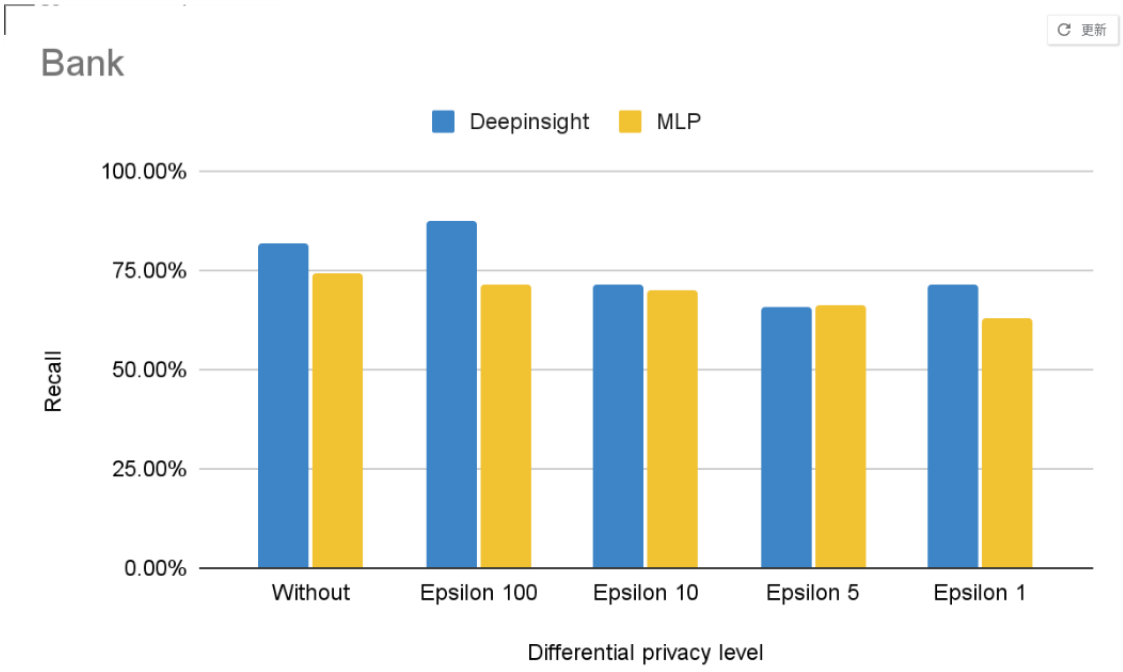

Figure 7.8: The comparison of recall between DPEDM and linear model on Bank Marketing dataset. The DPEDM has an overall higher recall level.

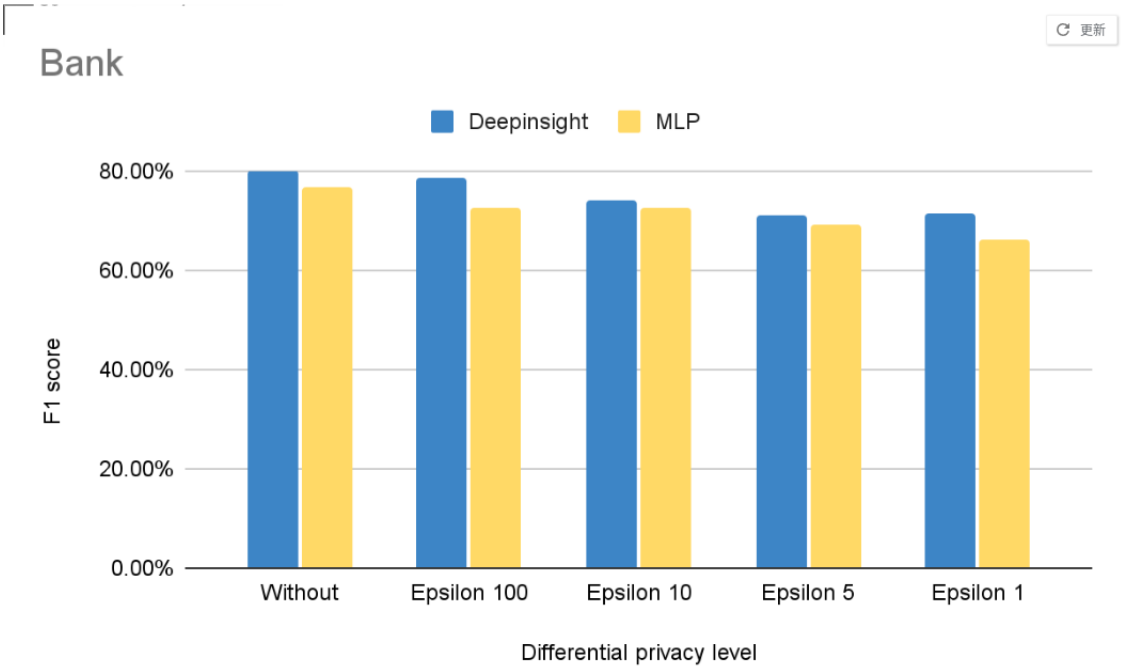

Figure 7.9: The comparison of F1 score between DPEDM and linear model on Bank Marketing dataset. The DPEDM has a higher f1 score at each differential privacy level.

	Accuracy	Precision	Recall	F1 score
Without DP	0.797	0.547	0.410	0.469
Epsilon=100	0.658	0.341	0.600	0.435
Epsilon=10	0.723	0.404	0.558	0.469
Epsilon=5	0.715	0.394	0.564	0.464
Epsilon=1	0.6985	0.376	0.575	0.455

Table 7.16: The experiment result on the Credit Default datasets by linear model. This table provides a benchmark performance for the dataset, allowing for a comparative analysis with our differential privacy-enhanced model.

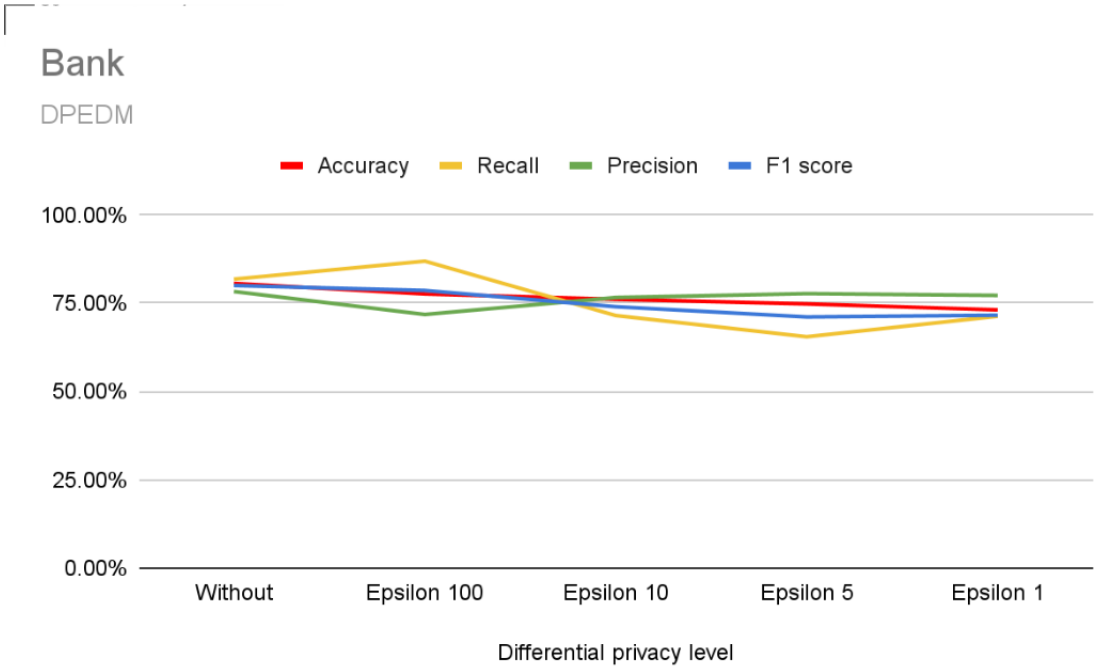

Figure 7.10: The diagram of evaluation metrics for DPEDM on Bank Marketing dataset. A noteworthy trend in precision and recall metrics presents itself. Rather than following a steady trend, these metrics appear to fluctuate as the differential privacy level changes. They increase at certain levels of privacy and decrease at others.

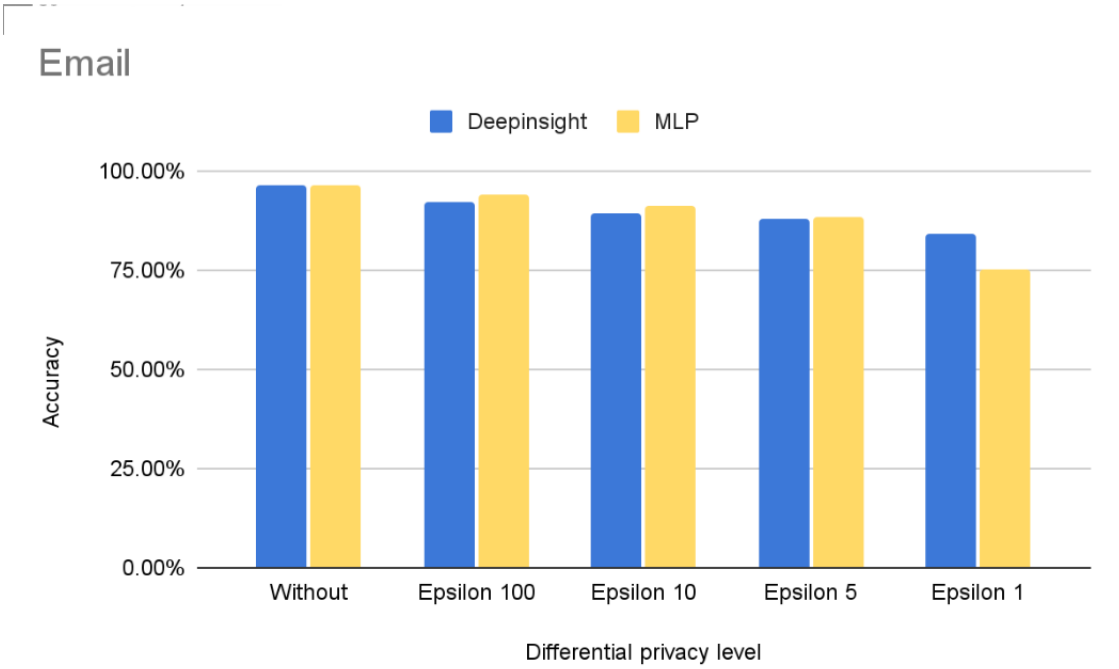

Figure 7.11: The comparison of Accuracy between DPEDM and linear model on Email Spam dataset. The DPEDM has a higher accuracy without privacy and when epsilon is 1.

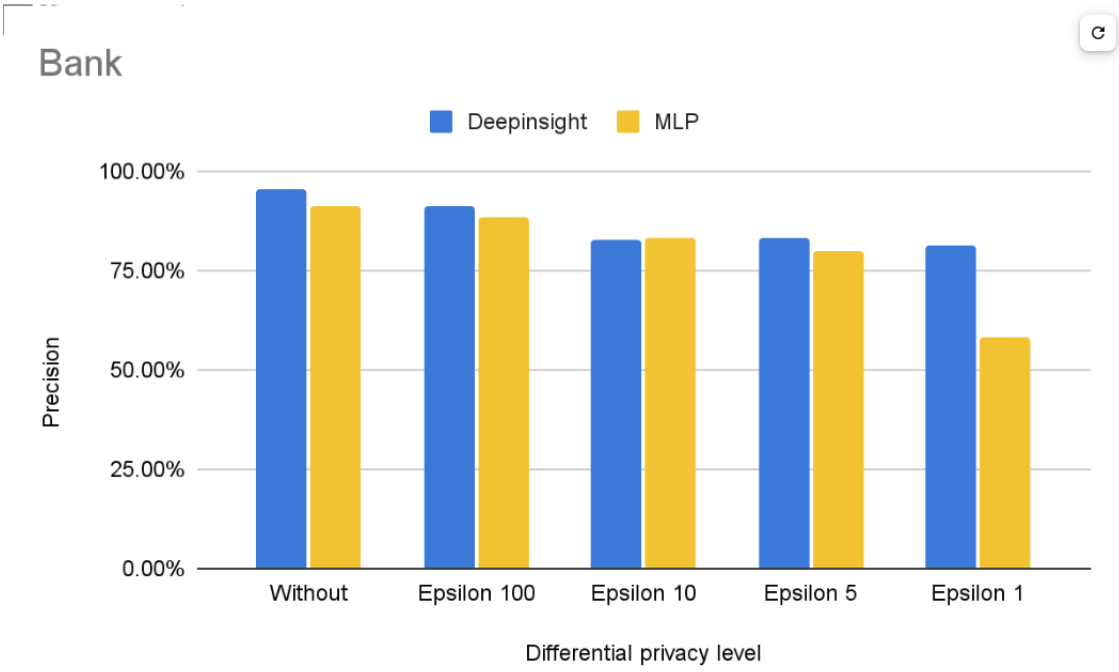

Figure 7.12: The comparison of precision between DPEDM and linear model on Email Spam dataset. The DPEDM has a higher precision at each differential privacy level.

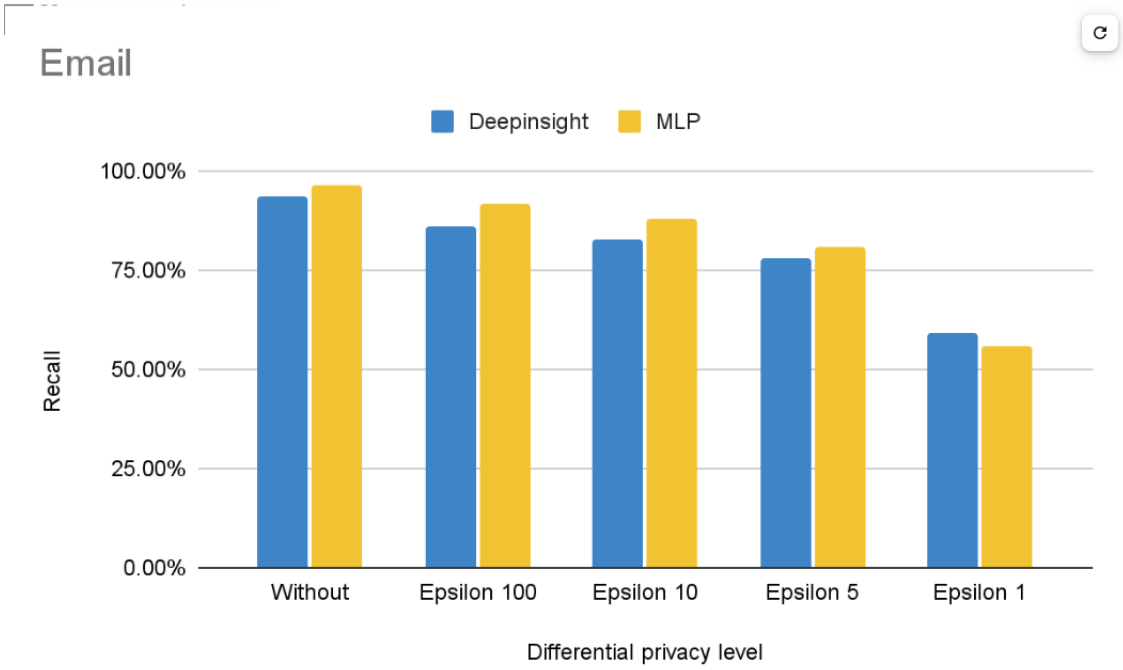

Figure 7.13: The comparison of recall between DPEDM and linear model on Email Spam dataset. The linear model has a higher recall except when epsilon being 1.

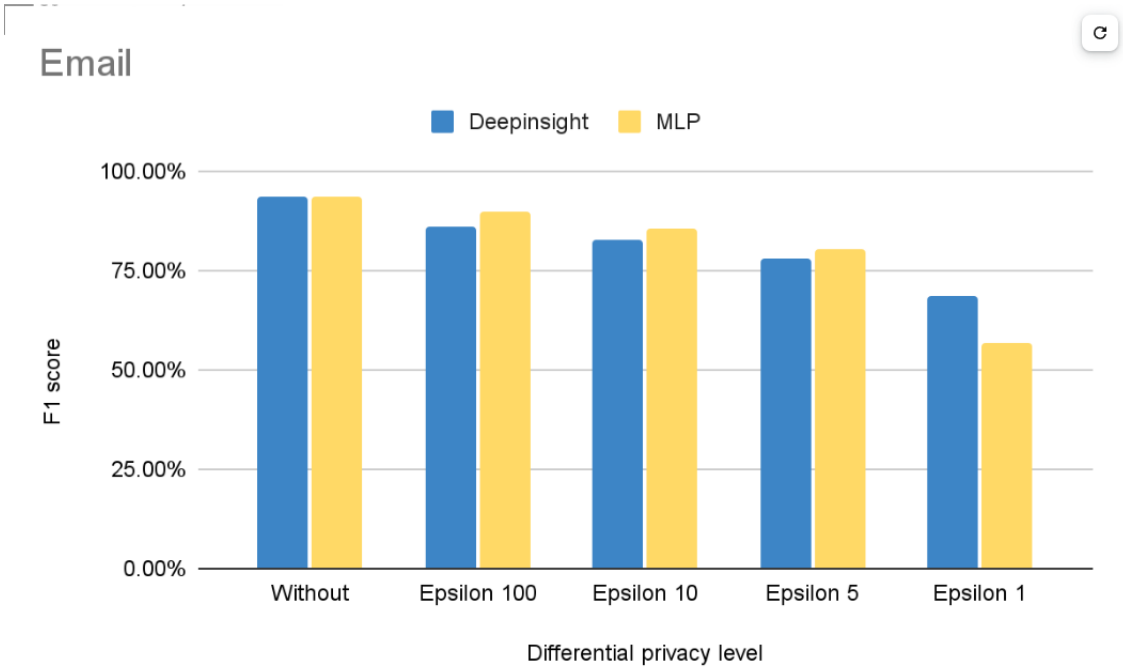

Figure 7.14: The comparison of F1 score between DPEDM and linear model on Email Spam dataset. The DPEDM has a higher f1 score without differential privacy and when epsilon being 1.

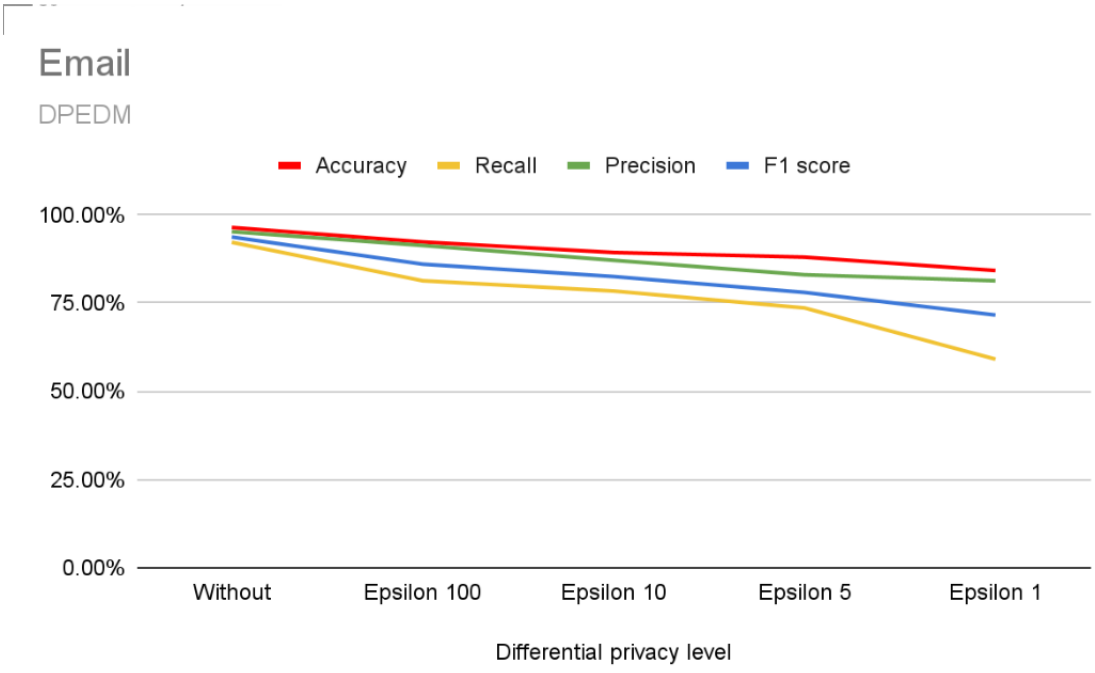

Figure 7.15: The diagram of evaluation metrics for DPEDM on Email Spam dataset. All metrics decrease when differential privacy level increases