Adv Syst Sci Appl 2023; **04**; 41-59
Published online at https://ijassa.ipu.ru.

Balancing Accuracy, Fairness and Privacy in Machine Learning through Adversarial Learning

Alexander Eponeshnikov[1], Rustem Sabitov[2], Gulnara Smirnova[2*], Shamil Sabitov[1]

[1] *Kazan Federal University, Kazan, Russia*

[2] *Kazan National Research Technical University named after A.N. Tupolev – KAI, Kazan, Russia*

Abstract: This paper investigates balancing accuracy, fairness and privacy in machine learning through adversarial learning. Differential privacy (DP) provides strong guarantees for protecting individual privacy in datasets. However, DP can impact model accuracy and fairness of decisions. This paper explores the effect of integrating DP into the adversarial learning framework called LAFTR (Learning Adversarially Fair and Transferable Representations) on fairness and accuracy metrics. Experiments were conducted using the Adult income dataset to classify individuals into high vs low income groups based on features like age, education etc. Gender was considered a sensitive attribute. Models were trained with different levels of DP noise (controlled by the epsilon hyperparameter) added to different modules like the encoder, classifier and adversary. Results show that adding DP consistently improves fairness metrics like demographic parity and equalized odds by 3-5% compared to an unfair classifier, albeit at a cost of 1-3% reduction in accuracy. Stronger adversary models further improve fairness but require careful tuning to avoid instability during training. Overall, with proper configuration, DP models can achieve high fairness with minimal sacrifice of accuracy compared to an unfair classifier. The study provides insights into balancing competing objectives of privacy, fairness and accuracy in machine learning models.

Keywords: machine learning, differential privacy, adversarial learning, fairness, accuracy, privacy-preserving models

1. INTRODUCTION

In the modern world, a huge number of different industries use machine learning to automate processes such as credit ratings, spam filtering. Machine learning plays an important role in preventing financial losses in the banking industry. Perhaps the most urgent task of forecasting is the assessment of credit risk (the risk of default on debt). Such risks can lead to losses of billions of dollars annually. A good result of machine learning mainly depends on the huge amount of data in which biases can be found in favor of certain attributes that are not fair in reality [8]. For example, by learning from unfair data, a classification model of an automated recruitment system is more likely to hire candidates from certain racial or gender groups or to favor candidates of a certain age.

Algorithmic bias is a growing subject of much discussion and debate in the use of AI. This is a complex topic due to the potential complexity of the mathematical definition of what it means to be “fair” in decision making. Fairness depends on the situation and is not only a reflection of values, ethics and legal norms. However, there are clear ways to approach AI fairness issues.

Quite often, individual and sensitive data (for example, financial transactions or tax payments) are taken to solve machine learning problems. Because of this, algorithms must

* Corresponding author: seyl@mail.ru

guarantee privacy. One of the methods is differential privacy. Differential privacy (DP) is a mathematical definition of the loss of confidential data of individuals when their personal information is used to create a product. Differential privacy allows to find a balance between privacy and accuracy using a positive value ε. If ε is small, then we keep more privacy, but we degrade accuracy. If ε is big, then privacy suffers for the sake of accuracy. Meaning of ε varies from 0 to infinity. To train models with DP, use DP-SGD. The core idea is that training a model can be done through access to its parameter gradients, i.e., the gradients of the loss with respect to each parameter of your model. If this access preserves differential privacy of the training data, so does the resulting model, per the post-processing property of differential privacy.

Summarizing all of the above, machine learning models must guarantee data privacy while avoiding discrimination and ensuring fair decision-making. However, the use of these methods may have an impact on the accuracy of the model. The research questions below in this work aim to examine the impact of privacy on fairness and accuracy, as well as to compare fair metrics and accuracy in different approaches of privacy protection in different machine learning models.

- What is the effect of privacy during encoding process on fairness in machine learning models?
- How does privacy during encoding process impact the accuracy of machine learning models?
- How do different privacy strategies impact fairness metrics in machine learning models?
- How do the different privacy strategies impact on accuracy in machine learning models?
- How do the different approaches in models impact on ability to balance fairness and accuracy while privacy in machine learning models?
- How do different datasets configurations impact on fairness and accuracy results?

2. BACKGROUND

In this section, we discuss the fundamental privacy and fairness concepts used throughout the work.

2.1. Differential privacy

Differential privacy is a set of methods that provide the most accurate queries to a statistical database while minimizing the possibility of identifying individual records in it. Let ε be a positive real number and A be a probabilistic algorithm that takes a set of data as input (represents the actions of a trusted party that has the data). Algorithm A is ε - differentially private if for all datasets D and D' the following expression is executed [11].

$$P[A(D) = t] \leq e^{\varepsilon} P[A(D') = t] \; \forall t$$

where D and D' are differing datasets by at most one element, and $P[A(D) = t]$ denotes the probability that t is the output of A.

According to this definition, differential privacy is a condition of the data publishing mechanism (that is, determined by the trusted party that releases information about the data set), not the data set itself. Intuitively, this means that for any two similar datasets, the differential private algorithm will behave approximately the same on both datasets.

2.2. Privacy preservation (DP-SGD)

Stochastic Gradient Descent (SGD) is an iterative method for optimizing differentiable objective functions. It updates the weights and biases by calculating the gradient of the loss function for small data packets [3]. DP-SGD [26] is a modification of the stochastic gradient

descent algorithm that provides provable privacy guarantees. It differs from SGD in that it limits the sensitivity of each gradient and works in tandem with the moments accountant algorithm to amplify and track the loss of privacy when the weight is updated. The moments accountant accumulates and tracks privacy spending while training deep neural networks. Moments accountant greatly improves on earlier SGD privacy analysis and provides meaningful privacy guarantees for deep learning trained on real-size datasets.

To ensure that the SGD is differentially private (i.e. DP-SGD), two modifications must be made to the original SGD algorithm. First, the sensitivity of each gradient must be limited. This is done by trimming the gradient in normal $l2$. Second, random noise is applied to the previously clipped gradient by multiplying its sum by the learning rate and then using it to update the model parameters.

2.3. Fairness

Fair modeling is an area of artificial intelligence that ensures that machine simulation results are not affected by protected attributes such as gender, race, religion, sexual orientation, etc. In this work we will use specific metrics that can be used to evaluate the fairness of a model. The most commonly used measures of fairness are statistical (demographic) parity and equalized odds [19, 27]. The definition of these metrics is given below.

Given a dataset $= \{X, Y, S\}$, where $X = \{x_i\}_{i=1}^n$, $\mathbf{x}_i \in \mathrm{R}^n$ is the feature vector, $Y = \{y_i\}_{i=1}^n$, $\mathbf{y}_i \in \{0,1\}$ the label, and S the binary protected attribute (e.g. gender, race, etc.). The goal is to learn a new representation X', such that X' will satisfy a certain fairness criteria such as:

- **Statistical parity**: A predictor $\hat{Y}$ trained on X' must satisfy: $P(S = 0) = P(S = 1)$
- **Equalized odds**: A predictor $\hat{Y}$ trained on X' must satisfy: $P(S = 0, Y = y) = P(S = 1, Y = y), \forall \mathbf{y} \in \{0,1\}$

3. RELATED WORK

The growing field of algorithmic fairness and differential privacy has led to the development of numerous methods for mitigating bias and ensuring data privacy. In recent research, the authors in [21] introduce a framework for evaluating and analyzing bias mitigation techniques by using a synthetic dataset and controlling different components of the data generation process. The study also analyzes the performance of several model architectures (such as MLP [22], CNN [13], LAFTR [17, 9], CFAIR [6], FFVAE [15]) on the Adult, CI-MNIST datasets, to understand the sources and levels of bias. As an extension to this article, this work is focused on combining the principles of algorithmic fairness and differential privacy in the training of models under adversary. In particular, this article considers the LAFTR model, which aims to provide a fair data while maintaining good predictive accuracy. The methodology is based on the introduction of differential privacy to model training. Furthermore, the examination of combining algorithmic fairness and differential privacy in training models in the presence of an adversary is extended in other studies.

In [18] authors aim to show that it is possible to maintain good predictive accuracy while still providing a strong guarantee for the privacy of individuals' data. Their paper is structured around the concept of differential privacy, which is a mathematical framework for protecting the privacy of individuals' data by adding noise to the data. In [16] authors introduce a differentially private (DP) neural representation framework that safeguards user privacy during network computations, using a DP noise layer and robust training algorithm while maintaining performance. The framework offers formal privacy guarantees through sensitivity-based noise injection. Some other papers [28, 26, 2, 1] also learn how privacy affects accuracy in different models. The paper [28] proposes a new type of Generative Adversarial Network (GAN) called Differentially Private Generative Adversarial Network

(DPGAN) that addresses the issue of privacy protection in training data. The DPGAN adds carefully designed noise to gradients during the learning process to provide differential privacy, with a proof of privacy guarantee and empirical evidence to support its analysis. Another paper [26] compares the fairness implications of two differentially private deep learning algorithms, DP-SGD and PATE, and finds that PATE has higher utility on under-represented groups in imbalanced datasets. The impact of differential privacy on the accuracy of machine learning models is also examined [2], and it is found that the reduction in accuracy brought about by DP disproportionately affects underrepresented subgroups and subgroups with more complex data. The authors of next paper [1] present new algorithmic techniques and a refined analysis of privacy costs to train deep neural networks while ensuring privacy. The solution is evaluated on standard image classification tasks (MNIST and CIFAR-10) and demonstrates the feasibility of the approach against a strong adversary who has full knowledge of the training mechanism and access to the model's parameters.

There are some papers, which explore the effect of privacy introduction on fairness. The first paper [24] explores the impact of differential privacy algorithms on the fairness of machine learning systems. It focuses on two specific differential privacy methods and analyses the reasons for the disparities that arise among different groups of individuals in these methods. The paper proposes guidelines to mitigate the unfair impacts and contributes to the growing research at the interface between differential privacy and fairness. The second paper [25] considers importance of ensuring fairness in machine learning systems, specifically in the context of decisions that affect individuals, such as criminal assessment and hiring. The text highlights the trade-off between model accuracy and fairness and the importance of considering sensitive attributes in learning tasks to ensure non-discrimination. In this paper, the focus is on the integration of algorithmic fairness and differential privacy in the training process of machine learning models. Differential privacy, which is a mathematical framework for the protection of the privacy of individuals' data by the addition of noise to the data, is utilized as the main method for privacy preservation. The LAFTR model, designed to provide a fair data representation while maintaining good predictive accuracy, is evaluated. The performance of the LAFTR model is assessed and the fairness and accuracy indicators are evaluated in a classification problem, which is a common application in machine learning. Machine learning bias refers to a systematic error that occurs in the results due to incorrect assumptions. The objective of algorithmic fairness is to reduce this bias. This is achieved through three main categories of algorithms: pre-processing, in-processing, and post-processing. Pre-processing techniques aim to reweight training samples [15], edit features and labels [4], and resample datasets [5]. In contrast, post-processing methods aim to calibrate predictions [12, 7] by adjusting the learned predictor to remove discrimination based on the joint statistics of the predictor, target, and protected attribute. In-processing techniques [14, 18, 11, 29, 17, 6, 30] focus on removing sensitive information, such as racial or gender group, age, financial transactions or tax payments that may lead to discrimination or bias in decision making, from the data. There are various architectures of models for realizing fair representation in-processing, such as CFAIR [29], LAFTR [17], FFVAE [6], among others. One of the architectures of the adversarial bias-mitigation model consists of an encoder, adversary module, and classifier. The encoder takes in the data and considers the sensitive attribute while encoding the data into the latent space. The adversarial training setup, using a gradient-reversal layer and an attacker network, to train a classifier that accurately predicts main task labels while being oblivious to protected attributes [10]. Finally, the data from the preserved latent space is passed on to the classifier, which has an integrated architecture. This architecture is further described in [17].

In this work, the focus is on using the LAFTR model to generate data for a third-party classifier that is both fair and private. The LAFTR model is designed to provide a fair representation. Differential privacy, which adds noise to the data to protect individuals' privacy, is utilized as the main method for privacy preservation. The integration of

algorithmic fairness and differential privacy is a central focus in the training process of the LAFTR model, ensuring that the data generated is suitable for use by a third-party classifier. The paper suggests using a third-party classifier, such as logistic regression, to evaluate the fairness of the data generated by the LAFTR model. The logistic regression model will be trained on the data from the latent space, which has undergone in-processing methods, allowing us to compare the performance of models trained on both raw data and processed data. Mathematical definition of the fairness variate depends on tasks. In this paper we will use two popular metrics: Statistical (Demographic) parity and Equalized odds [20].

4. METHODOLOGY

In this article, we investigate the dependencies of the impact of privacy injection on accuracy and fairness. The LAFTR-DP and LAFTR-EOD [17] models with different privacy values ε in different modules of the model will be compared. In differential privacy ε represents the privacy budget that determines the amount of noise added to the gradients during training, where a smaller ε corresponds to stronger privacy guarantees but potentially higher noise levels, impacting the trade-off between privacy and utility in the learning process. These networks apply adversary training and optimization using DP-SGD to achieve fairness and guarantees of data confidentiality.

4.1. Dataset

Adult dataset: The Adult dataset is a widely used standard machine learning dataset for studying and demonstrating many common or specially designed machine learning algorithms for unbalanced classification. In total, the data set (train and test) contains 48842 samples. The dataset contains 16 columns, including a target field “Income” and 14 attributes that describe a person's demographics and other features. The income is divided into two classes: <=50K and >50K, which serves as the target variable for the classification task.

The dataset includes various personal information about individuals, such as their age, education level, gender, occupation, and so on. Given the characteristics of adult recruitment, the classification task is to determine whether a person earns more than 50K or less than 50K. Gender was selected as a protected attribute. The size of the test data was chosen to be equal to 30% of all data.

4.2. Model

The idea is to study a representation that satisfies a certain property of fairness while remaining differentially private. **Fig. 1** shows a network architecture that seeks to study a representation of data Z capable of reconstructing input data X, classifying target labels Y, and protecting a sensitive attribute S (**Fig. 1**). The loss function of the network is defined as a linear combination of three loss terms, the reconstruction loss $\left(L_{g,d}^{\text{textrec}}\right)$, the adversary loss $\left(L_{h}^{\text{textadv}}\right)$, and prediction loss $\left(L_{f}^{\text{textpred}}\right)$[9]:

$$L(g,d,f,h) = \alpha L_{g,d}^{\text{textrec}} + \beta L_{f}^{\text{textpred}} - \gamma L_{h}^{\text{textadv}}$$

where α, β and γ are hyperparameters controlling the weighting of the competing objectives. The decoder reconstructs input data X from the latent space. The classifier predicts the class label from latent space Z. The adversarial loss is to enforce the representation (latent space) to satisfy a certain fairness notion. Any of the first two requirements can be omitted by setting the hyperparameters to zero. To improve fair metrics, which defined in background section, loss functions can be defined as [9]:

$$L_{h}^{\text{textadv}} = E_{Z,S}\left[S \cdot \log\left(h(Z)\right) + (1-S) \cdot \log\left(1-h(Z)\right)\right],$$

$$L_{f}^{\text{textpred}} = E_{Z,Y}\left[Y \cdot \log\left(f(Z)\right) + (1-Y) \cdot \log\left(1-f(Z)\right)\right],$$

$L_{g,\,d}^{\text{textrec}} = \| X - d\big(g(X)\big) \|^2$, where h - adversary, f - classifier, g - encoder, d - decoder.

Also, the objective of the network is not only to make the representation Z fair, but also to be privacy preserving. In this regard, we propose to train the network with privacy preserving techniques such as DP-SGD.

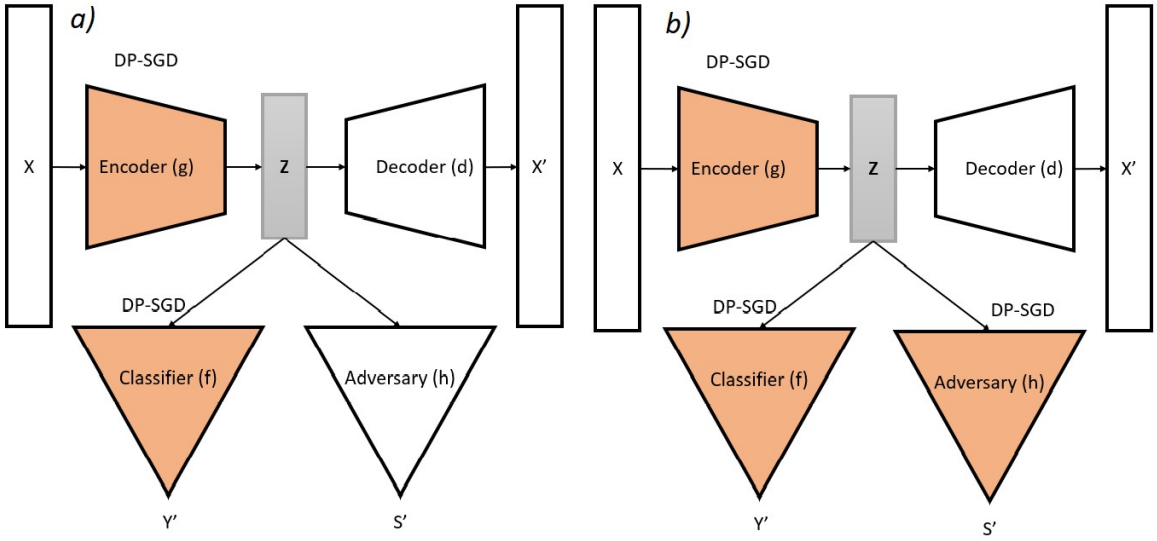

Fig. 1. LAFTR model with DP-SGD in (a) encoder/ classifier, (b) encoder/classifier/adversary

To make latent space confidential we need to introduce DP-SGD method in encoder. Also, in this work we will inject privacy in classifier and adversary modules (**Fig. 1**).

In addition, in this work, two types of LAFTR models (LAFTR-DP and LAFTR-EOD) are considered. The only difference between LAFTR-DP and LAFTR-EOD models is that in LAFTR-EOD approach we pass to adversary not only Z, but we pass class label Y in addition.

4.3. Training

LAFTR seeks to study the encoder that gives reliable representations, i.e. the output of the encoder can be used by third parties with confidence that their naively trained classifiers be fairly fair, private and accurate. Algorithm below describes the training process of LAFTR with privacy preservation. The detailed pseudocode is described in Algorithm 1.

Algorithm 1: Differentially private SGD on learning under adversary

Input: Dataset $(X_i, Y_i, Z_i)\mathrm{i} = 1 \ldots \mathrm{N}$; batch size b; learning rate l; K iterations; noise σ; clipping bound C; hyperparameters controlling the weighting (α, β and γ); component for privacy preservation (θ (for g and d) or φ (for f) or ψ (for h)) as m; loss function $L(\theta(X), \varphi(X), \psi(X), \mathrm{X}, Y, S) = \alpha L_{g,d}^{\text{textrec}} + \beta L_f^{\text{textpred}} - \gamma L_h^{\text{textadv}}$

$\theta_0, \varphi_0, \psi_0 \leftarrow$ initialize network parameters (encoder, classifier, adversary respectively)
$U \leftarrow$True (Boolean indicating whether to update parameters θ, φ, ψ)
for k $\in [K]$ **do**
  if U **then**
    Select next batch;
  foreach (X_i, Y_i, Z_i) *in* batch **do**
    Compute gradients for each module:
    $g_{\theta,\varphi}^i \leftarrow \nabla_\theta L(\theta_k(X_i), \varphi_k(X_i), \psi_k(X_i), \mathrm{X}_i, Y_i, S_i)$
    $g_\psi^i \leftarrow \nabla_\psi \gamma L_h^{\text{textadv}}(\psi_k(X_i), S_i)$
    Clip gradients:

$$g_\theta^i \leftarrow g_\theta^i / \max\left(1, \frac{\|g_\theta^i\|_2}{C}\right)$$

$$g_\varphi^i \leftarrow g_\varphi^i / \max\left(1, \frac{\|g_\varphi^i\|_2}{C}\right)$$

$$g_\psi^i \leftarrow g_\psi^i / \max\left(1, \frac{\|g_\psi^i\|_2}{C}\right)$$

end

Add noise to selected component:

$$\tilde{g}_m \leftarrow \frac{1}{b}\left(\sum_i g_m^i + N(0, \sigma^2 C^2 I)\right)$$

Sum other gradients:

$$\tilde{g}_{\text{textnot m}} \leftarrow \frac{1}{b}\sum_i g_{\text{textnot m}}^i$$

Descent:

if U **then**

$$\theta_{k+1} \leftarrow \theta_k - l\tilde{g}_\theta$$

$$\varphi_{k+1} \leftarrow \varphi_k - l\tilde{g}_\varphi$$

else

$$\psi_{k+1} \leftarrow \psi_k + l\tilde{g}_\psi$$

end

$U \leftarrow \text{not U}$

end

In short, it is a combination of adversarial learned fair representations [9] and DP-SGD [1]. We learn encoder and classifier with gradient clipping and adding noise if necessary. Then we freeze the learned encoder and classifier and learn the adversary part with gradient clipping and adding noise if necessary.

The overall loss function is a combination of these loss functions with hyperparameters controlling the weighting. The training process consists of iterating over a number of epochs, each time selecting a random batch of data from the training set. The gradients for each component are computed and clipped to ensure that they do not exceed a certain bound. Then, a noise is added to one of the components to achieve differential privacy. Finally, the parameters of the corresponding component are updated by performing stochastic gradient descent with the computed gradients. This process is repeated until the desired number of epochs is reached. The training algorithm of LAFTR with privacy preservation is a crucial component of this approach, allowing for the training of complex models that can achieve both fairness and privacy.

4.4. Evaluation metrics

As we said in Section 4.3, we try to train encoder which will give reliable representation. Training encoder several times and taking the outputs from the encoder, we will pass it into a simple binary classifier (logistic regression), average them and calculate standard deviation. After that we will use fairness metrics which was mentioned in Section 0 to evaluate these bias-mitigation strategies. Also, we will measure accuracy of all models for comparison with bias-mitigation strategies and unfair strategies.

To evaluate accuracy and fair metrics we will use logistic regression. Logistic regression will learn on generated by encoder new data several times and return mean and standard deviation of accuracy, difference of demographic parity - $\Delta\text{DP} = P(S=0) - P(S=1)$ and difference of equalized odds - $\Delta\text{EOD} = P(S=0, Y=y) - P(S=1, Y=y)$.

During the training of different model configurations, the fair metrics and accuracy will change. To understand advantages of provided models we need to compare accuracy-fairness trade-off. For evaluation trade-off we will use next function:

$F(a,f) = \frac{a}{1+f}$, where a - accuracy, f - fair metric. Next we will call Acc/Fair.

4.5. Model and hyperparameters

At the outset of the experiment, the initial step was to define a set of hyperparameters that remained constant throughout all subsequent experiments. Hyperparameters are parameters that cannot be learned directly from the training data and need to be set manually before the training process commences.

Table 1. Model fixed hyperparameters

Model hyperparameters	**Values**
Encoder MLP depth as in depth*[width]	2 layers
Classifier MLP depth as in depth*[width]	2 layers
Encoder MLP width	32 neurons
Classifier MLP width	32 neurons
Latent (Z) space dimension	8 neurons
α– reconstruction loss weight	0
β– prediction loss weight	1
γ– adversary loss weight	1
Activation function in hidden layers in autoencoder	LeakyReLU
Activation function in hidden layers in classifier	LeakyReLU
Activation function in hidden layers in adversary	LeakyReLU
Activation function after last hidden layer of encoder	LeakyReLU
Activation function after last hidden layer of classifier	Sigmoid
Activation function after last hidden layer of adversary	Sigmoid

In **Table 1**, we can observe the various model hyperparameters and their corresponding values that were used in our experiments. We trained our model without taking into account the reconstruction loss, as our primary goal was to obtain a good representation in the latent space. Therefore, we did not require a good reconstruction of input data. Additionally, we used the same architecture for all modules of the network, except for the activation function after the last hidden layer of the encoder. This difference in the activation function was necessary since the encoder generates a new representation of input data $X = \{x_i\}_{i=1}^{n}$, $x_i \in \mathrm{R}^n$ whereas the classifier and adversary try to predict label Y or attribute S, which have binary nature.

The choice of parameters in the model was based on the findings from prior research, particularly [21]. The model architecture was configured to have a two-layer Multi-Layer Perceptron (MLP [22]) for both the encoder and classifier, with a width of 32 neurons in each layer, following the recommendation from [21]. The adversary module was set up in two different configurations. The latent space dimension was set to 8 neurons.

Having completed the previous step, the subsequent action is to establish the values of the training parameters. These parameters are essential in controlling the learning process and determining the behavior of the model during the training phase. The selection of suitable values for these parameters is a crucial process as it can significantly influence the performance of the model.

After determining the appropriate values for the training parameters, they were fixed and used throughout the training process to optimize the model's performance on the given task. Notably, these parameters were carefully chosen to produce results that could be compared to those reported in a previous study [17].

The process of defining these constant training parameters is complex, and various factors must be considered. The size of the dataset, the complexity of the model, and the available computational resources – factors which were taking into account during empirical

choosing hyperparameters. The most important hyperparameters which affect to results - learning rate, and number of epochs, optimizer. Finding the best combination of these parameters helps achieve optimal results.

Once the appropriate values for the training parameters have been determined, they are fixed and used throughout the training process to optimize the model's performance on the given task. Regular monitoring and tuning of these parameters may also be required during the training process to ensure that the model continues to improve its performance. In some cases generally founded hyperparameters cannot provide satisfiable results. This iterative process involves observing the model's performance during training, making adjustments to the parameters, and retraining the model to ensure that it is optimized for the task at hand.

Table 2 Training fixed parameters

Training parameters	Values
Max gradient norm in encoder	10
Max gradient norm in classifier	10
Max gradient norm in adversary	10
Encoder/classifier optimizer	NAdam
Adversary optimizer	NAdam
Encoder/classifier learning rate scheduler	PolynomialLR(2)
Adversary learning rate scheduler	PolynomialLR(2)

The **Table 2** provided outlines the specific training parameters and their corresponding values that were used during the training process. The values were carefully selected based on various factors, such as the size of the dataset and the complexity of the model.

The maximum gradient norm in the encoder, classifier, and adversary was set to a value of 10. This parameter controls the maximum magnitude of the gradients during the training process, preventing the model from diverging or becoming unstable.

The optimization algorithm used for both the encoder and classifier was NAdam, while the adversary used the same algorithm with modifications to ensure differential privacy. NAdam was selected over other popular optimization algorithms due to its superior performance in handling noisy or sparse gradients. NAdam is a variant of the Adam optimization algorithm that combines the benefits of adaptive learning rates and momentum-based optimization techniques [23]. It has been shown to outperform other popular optimization algorithms such as Adagrad, RMSprop, and Adam. In detail, DP-NAdam works similarly to DP-SGD, with some modifications that introduce the NAdam algorithm itself. These modifications include the calculation of the adaptive learning rates and momentum parameters, which are adjusted based on the first and second moments of the gradient, as well as the inclusion of noise to ensure differential privacy. The choice of NAdam over SGD was made based on its superior performance on the given task, as well as its ability to handle noisy or sparse gradients that are common in differential privacy settings. Furthermore, the use of DP-NAdam provides additional benefits such as faster convergence rates.

The learning rate scheduler used for both the encoder and classifier was PolynomialLR(2), which gradually reduces the learning rate quadratically over the course of the training process to prevent the model from overfitting to the training data. Similarly, the adversary also used the same scheduler.

It is worth noting that the choice of parameters was not solely based on the findings from [21], as modifications were made to ensure the model could achieve similar results as reported in both [21] and [17]. While [21] provided valuable insights into the choice of architecture, latent space dimension, and the adversary module configurations, [17] provided guidance on the selection of the optimization algorithm and its modifications to ensure differential privacy. The modifications made to the optimization algorithm were necessary to achieve differential privacy while maintaining the model's performance. It should be noted that even with the use of DP-NAdam, the learning rate still had to be increased dramatically (~0.1-0.15) to achieve the desired level of performance. Despite the need to increase the learning rate dramatically, the model was able to achieve the desired level of performance

without sacrificing its stability or accuracy. Overall, the parameters were carefully selected based on a mixture of insights from both [21] and [17], with modifications to ensure that the model could achieve the desired level of performance while maintaining differential privacy.

Additionally, based on a large number of parameter tuning attempts, it was discovered that using a larger batch size improves model stability. Therefore, the maximum feasible batch size was used for training, which could fit in VRAM. For the Adult dataset, the entire dataset was used, for the German dataset, the entire dataset was used, and for CelebA, 20,000 samples were used.

Empirically, it has been observed that the model may cease to be adequate under certain conditions. Therefore, after training the model, a validation of its adequacy was conducted. The classifier within the neural model should yield accuracy values above 50%, while fair metrics should not exceed 2%. In the event that any of these requirements were not met, the model was retrained with different initial weights.

5. EXPERIMENTS

In this section we will compare different approaches of privacy preservation and/or bias-mitigation strategies. All models with privacy preservation and/or bias-mitigation was trained several times to calculate mean and standard deviation of accuracy and fair metrics. This work considers models with various configurations as described in **Table 3**. Since each configuration was trained multiple times, in addition to the histogram, a t-test table with a p-value of 5 was used for comparison.

Table 3 Changing parameters

Parameter	Values
Dataset	Adult
Model architecture	LAFTR-DP, LAFTR-EOD
Adversary module	2 layers * 32 neurons; 4 layers * 64 neurons
Privacy in	No privacy; Encoder/Classifier; Encoder/Classifier/Adversary
ε	1, 3, 10, 30

To ensure that the results were robust a diverse set of 37 models were trained for each dataset. These models varied in terms of their architecture, hyperparameters, and level of privacy. This approach allowed for a comprehensive evaluation of the performance of the models under different configurations and helped to identify the most effective configurations for each dataset.

A t-test will be utilized to compare the performance of models. T-test a statistical hypothesis test that is used to determine if there is a significant difference between the means of two groups. In the context of metrics, t-tests can be used to determine if there is a statistically significant difference in the performance of two models, or if a certain modification or intervention has had a significant effect on the performance of a model. By using t-tests, we can make informed decisions about the effectiveness of different strategies for improving the performance or fairness of machine learning models.

There are several conventions in the model names. “Unfair|No privacy” - logistic regression learned on raw data without any modifications. “Adversary = Classifier” means that modules of LAFTR have the same structure, “Adversary > Classifier” means that adversary part stronger (2 times more layers and 2 times more neurons in each layer).

5.1. Adult dataset

The Adult dataset is a well-known benchmark dataset in machine learning that contains demographic and employment-related information of individuals to predict whether their income is above or below a certain threshold. In this case each configuration of models was trained 10 times. Number of epochs for each training was set to 250. In all next datasets, based on t-test, protected models demonstrated similar performance across different levels of

privacy, regardless of the level of noise in the gradients during training and the neural network modules in which this noise was injected. Therefore, we will consolidate the results with different epsilon values and different privacy injection configurations. Next, tables will be presented for each metric, comparing the minimum, maximum, and average values among all implementations. The results will be grouped by model type and adversary strength, taking into account the conditions described above.

5.2. Difference of demographic parity

There are results of measuring first fair metric - difference of demographic parity – ΔDP.

Based on the **Fig. 2**, the comparison models with different privacy levels, epsilon values, and adversary-classifier strengths we can say for sure that the "Unfair|No privacy" model is the most unfair.

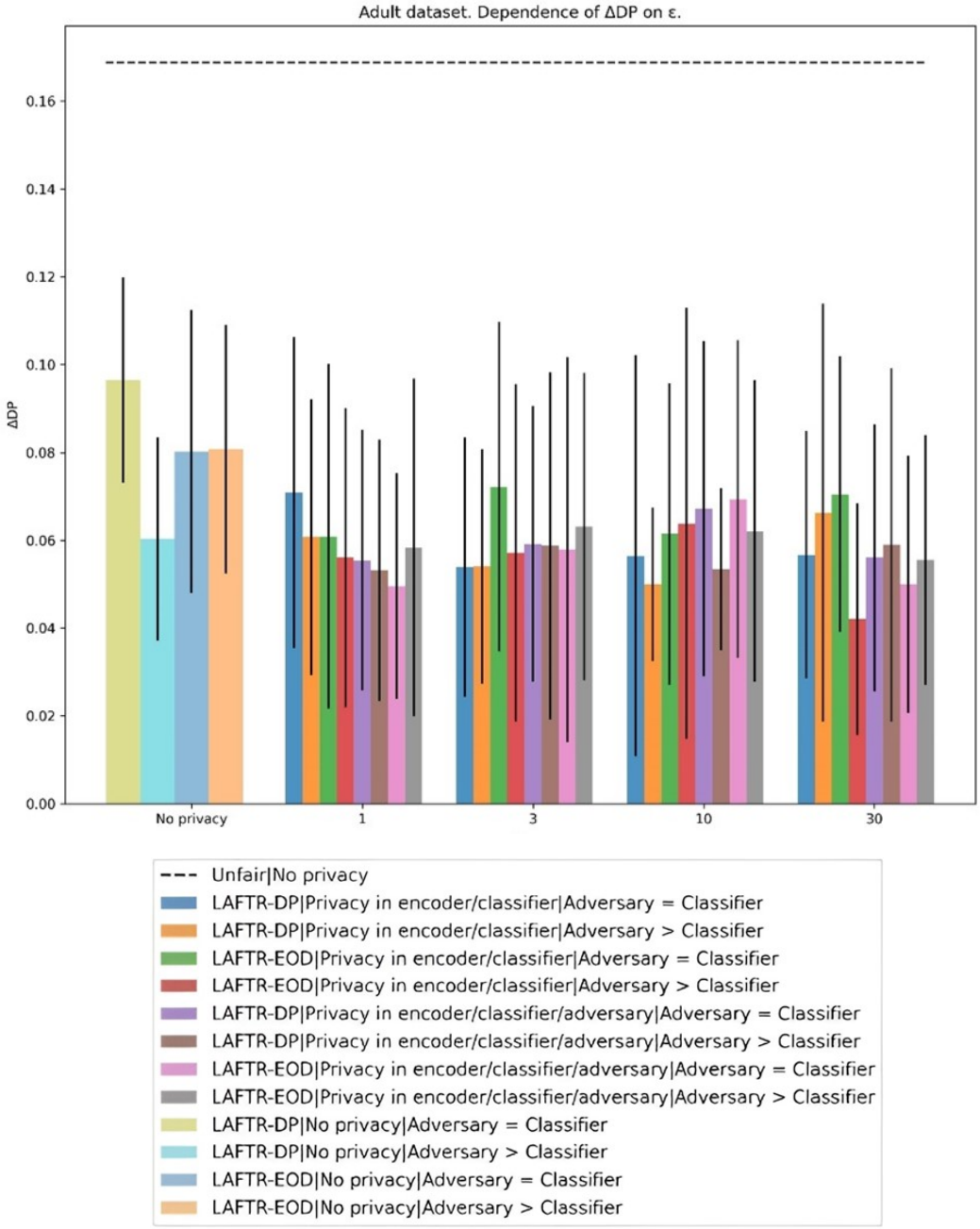

Fig. 2. Dependence of ΔDP on ε for Adult dataset

As observed in **Table 4**, for LAFTR-DP, both the minimum and average values of ΔDP are very low, regardless of the strength of the adversary. However, for the private models,

the minimum values are nearly zero, while for the unprotected models, they range from 6% to 9%. Additionally, in the case of protected models, the average value is independent of the adversary's strength, with a difference of only 3%. The maximum value does not exceed that of the unfair approach in any case, but the difference ranges from 0.6% to 7%.

Regarding LAFTR-EOD, the behavior of the results is comparable to LAFTR-DP, but with slight variations and more stability in the values. The maximum difference between LAFTR-EOD and the unfair model ranges from 3.5% to 2.5%. On average, the minimum values are slightly lower for this architecture. In the absence of protection, the difference can reach up to 5% in the case of a weak adversary. Additionally, the strength of the adversary has a stronger impact on the average value for the protected models compared to the unprotected ones.

Table 4 ΔDP comparison. Adult dataset.

ΔDP					
Model	**Privacy**	**Adversary**	**min**	**max**	**mean**
LAFTR-DP	No privacy	Adversary = Classifier	0.064	0.136	0.097
		Adversary > Classifier	0.020	0.097	0.060
	Privacy in..	Adversary = Classifier	0.000	0.163	0.059
		Adversary > Classifier	0.001	0.140	0.057
LAFTR-EOD	No privacy	Adversary = Classifier	0.017	0.133	0.080
		Adversary > Classifier	0.046	0.143	0.081
	Privacy in..	Adversary = Classifier	0.005	0.145	0.061
		Adversary > Classifier	0.000	0.152	0.057
Unfair	No privacy	-	0.169	0.169	0.169

5.3. Difference of equalized odds

Next step – results of measuring second fair metric - difference of equalized odds - ΔEOD.

It is very difficult to draw any conclusions based on the

Fig. 3.

Based on **Table 5**, in the case of ΔEOD, the situation is slightly different in comparisaon with ΔDP. The strengthening of the adversary continues to have a positive impact on fairness and often allows for an additional 1-2% improvement on average. Furthermore, all protected models can achieve better results than their unprotected counterparts. However, in the case of ΔEOD, techniques such as adversary strengthening or privacy injection can significantly deteriorate the outcomes. For instance, in the LAFTR-DP model, the addition of a strong adversary can lead to results that are 5% worse than the unfair model, and introducing noise can increase the difference up to 22% in the worst case. Nevertheless, the average values are still lower than those of the unfair approach, but the introduction of noise brings the average values closer to those of the unfair solution, with a difference of 1-2%. In contrast, for non-private models, the difference ranges from 3.6% to 2.6%.

The behavior of the results in LAFTR-EOD models is similar to LAFTR-DP, but they exhibit less stability. The minimum values are lower in almost all cases, while the maximum values are consistently high, even in the case of a weak adversary without protection (unfair approach performs better by 7%). In other cases, the difference can reach up to 19%. Due to the higher instability and a larger number of unfair results, the fairness metric's average value for LAFTR-EOD is approximately 2.5% worse than that of LAFTR-DP.

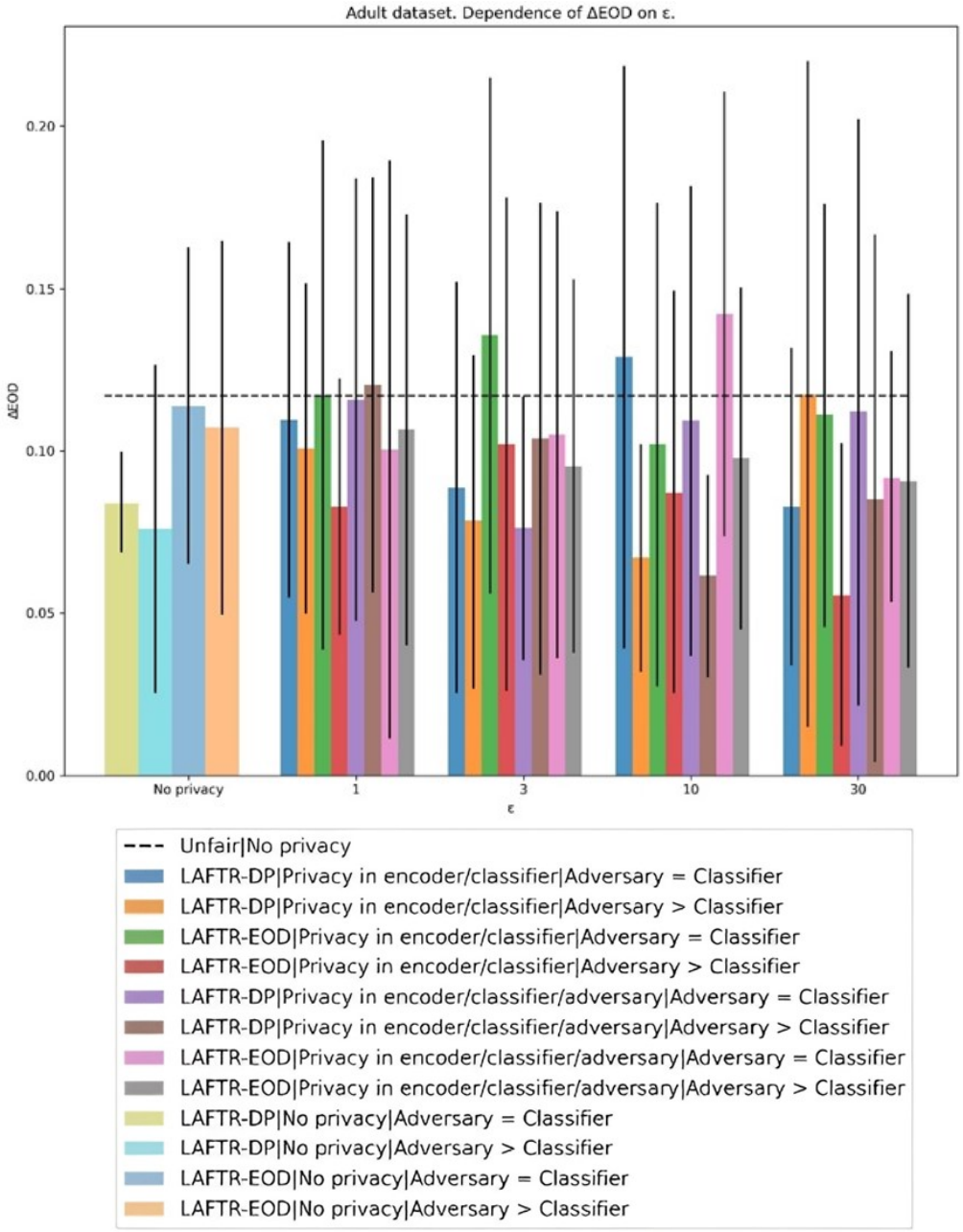

Fig. 3. Dependence of ΔEOD on ε for Adult dataset

Table 5 ΔEOD comparison. Adult dataset.

ΔEOD					
Model	**Privacy**	**Adversary**	**min**	**max**	**mean**
LAFTR-DP	No privacy	Adversary = Classifier	0.055	0.107	0.084
		Adversary > Classifier	0.015	0.165	0.076
	Privacy in..	Adversary = Classifier	0.008	0.335	0.103
		Adversary > Classifier	0.010	0.317	0.092
LAFTR-EOD	No privacy	Adversary = Classifier	0.027	0.183	0.114
		Adversary > Classifier	0.042	0.205	0.107
	Privacy in..	Adversary = Classifier	0.006	0.309	0.113
		Adversary > Classifier	0.007	0.276	0.090
Unfair	No privacy	-	0.117	0.117	0.117

5.4. Accuracy

To fully understand the behavior of the model, it is necessary to investigate patterns of accuracy.

As in demographical parity metric **Fig. 4** demonstrates that the "Unfair" model, which has no privacy protection and an unfair adversary much better predict labels and consistently outperformed by all other models in terms of accuracy.

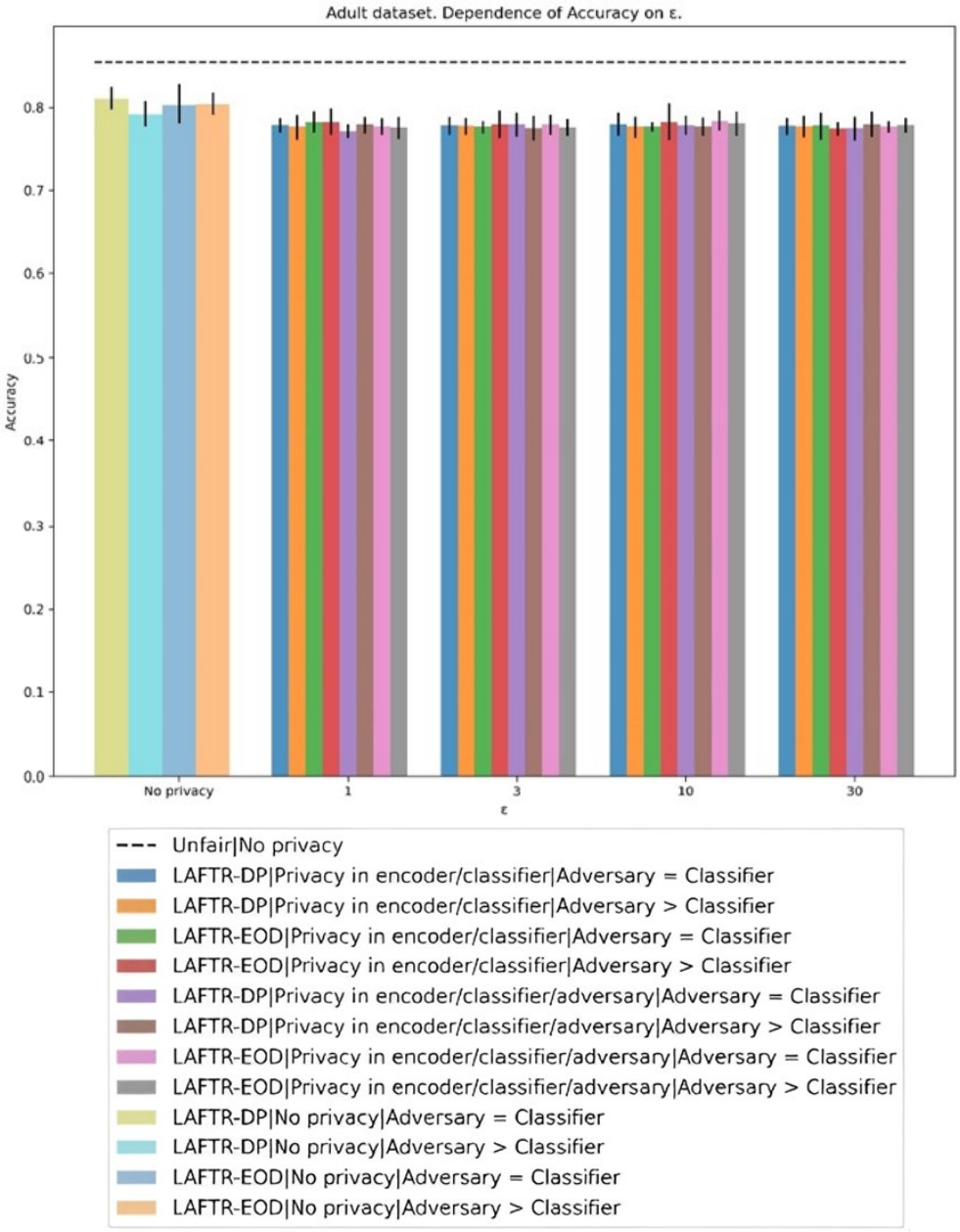

Fig. 4. Dependence of accuracy on ε for Adult dataset

As shown in **Table 6**, the average accuracy values are independent of the adversary's strength and the model type. The only factor that leads to a decrease in accuracy of around 2-3% is the introduction of privacy. The difference between the minimum and maximum accuracy values varies between 4% and 7%, and these values do not exhibit any clear correlation. In general, non-private models demonstrate a 5% deviation from the unfair solution, likely due to the reduced dimensionality of the latent z-space and the potential loss of information during data encoding. Protected models show an 8% deviation, which can be attributed to the addition of noise in the training process, affecting the latent z-vector.

To assess the trade-off between accuracy and fairness and understand the benefits of prioritizing fairness over accuracy, it is crucial to examine the compromise between these two metrics. By analyzing the relationship between accuracy and fairness measures, we can determine the extent to which sacrificing accuracy has resulted in improved fairness

outcomes. This analysis allows us to evaluate the value of prioritizing fairness and make informed decisions regarding the trade-off between these two important considerations.

Table 6 General accuracy comparison. Adult dataset.

Accuracy					
Model	**Privacy**	**Adversary**	**min**	**max**	**mean**
LAFTR-DP	No privacy	Adversary = Classifier	0.785	0.829	0.810
		Adversary > Classifier	0.767	0.810	0.792
	Privacy in..	Adversary = Classifier	0.749	0.803	0.777
		Adversary > Classifier	0.749	0.805	0.777
LAFTR-EOD	No privacy	Adversary = Classifier	0.758	0.830	0.803
		Adversary > Classifier	0.784	0.823	0.804
	Privacy in..	Adversary = Classifier	0.756	0.808	0.779
		Adversary > Classifier	0.754	0.822	0.778
Unfair	No privacy	-	0.853	0.853	0.853

5.5. Accuracy/Fairness trade-off

We compared fairness and accuracy of different approaches and explored that the bigger fairness the lower accuracy and vice versa. To understand advantages of provided models we need to compare accuracy-fairness trade-off using custom metric which was provided in Section 4.4.

The information presented in

Fig. 5 shows that "No privacy" approaches slightly better than "Unfair" and protected approaches.

But for a complete picture of understanding the compromise, it is also necessary to compare the accuracy-difference of equalized odds trade-off.

It is evident from **Fig. 6** that "Unfair" approach has better trade-off across all other models.

As evident from **Table 7**, in all cases of trade-offs between ΔDP, the average values are better than those of the unfair solution (with a difference of 0.4% to 1.7%), indicating consistent improvements in terms of this metric. It is important to note that the introduction of noise increases the variability between the minimum and maximum values (from approximately 3% for non-private models to around 10% for private models), but it also leads to an increase of approximately 2-3% in the maximum value. There are no differences observed between architectures, but strengthening the adversary may yield improvements of 1-2% in some cases.

When considering ΔEOD, the average values generally fall behind the unfair solution by 2-6%. However, in the maximum column, all values are equal to or greater than the unfair solution, indicating the need for more nuanced adjustments. On average, LAFTR-DP demonstrates values superior to LAFTR-EOD by 1%. It is worth noting the high stability of LAFTR-DP without privacy, with a variability of 4% and 7% between the minimum and maximum values for different adversaries, while the others range from 12% to 18%.

In general, for this dataset, it can be stated that with proper configuration, outstanding fairness results can be achieved while sacrificing less in terms of accuracy compared to gains in any fairness metric. Furthermore, based on all the results, it becomes evident that strengthening the adversary has a positive impact on the trade-off, but only when carefully fine-tuned; otherwise, it may lead to poor outcomes. Additionally, it is not possible to confidently assert that LAFTR-DP outperforms LAFTR-EOD, as it depends on the specific task, where LAFTR-EOD may perform better in certain cases. Finally, it can be concluded that by incorporating privacy at various levels, it is feasible to attain either a highly fair model or avoid significant accuracy losses while improving fairness.

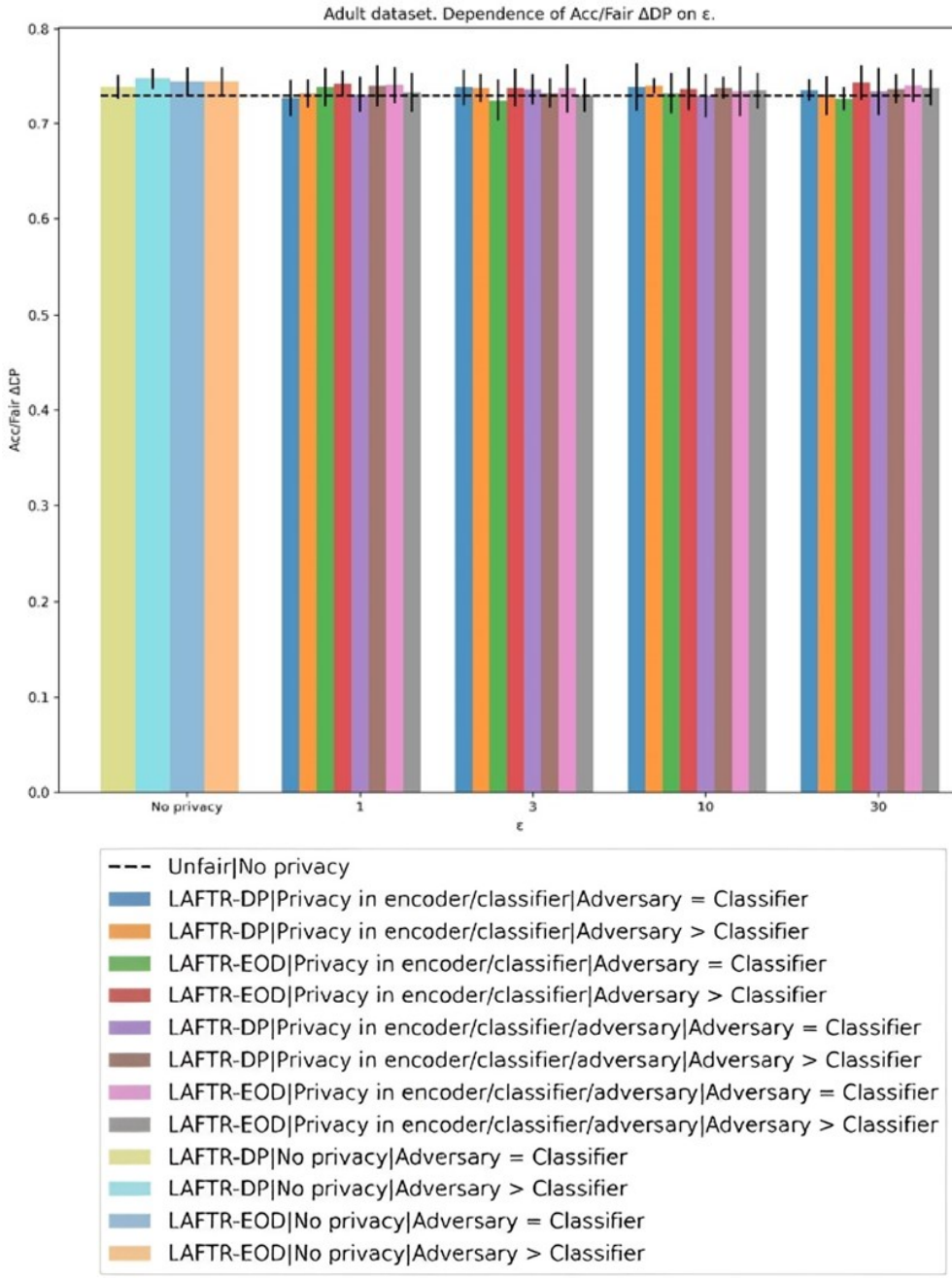

Fig. 5 Dependence of Acc/Fair ΔDP on ε for Adult dataset

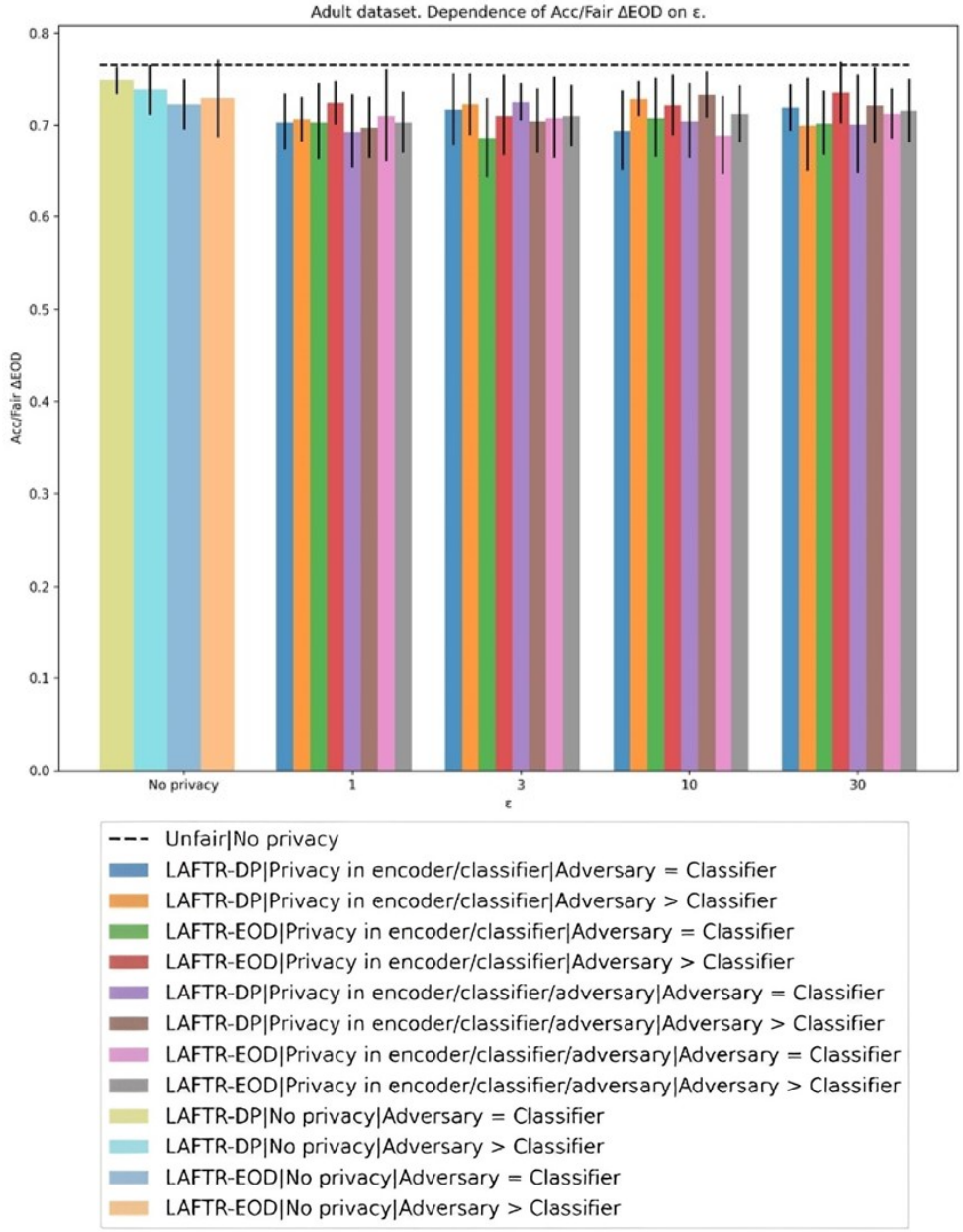

Fig. 6 Dependence of Acc/Fair ΔEOD on ε for Adult dataset

Table 7 General accuracy-fairness trade-off comparison. Adult dataset.

Acc/Fair ΔDP					
Model	**Privacy**	**Adversary**	**min**	**max**	**mean**
LAFTR-DP	No privacy	Adversary = Classifier	0.716	0.755	0.739
		Adversary > Classifier	0.723	0.761	0.747
	Privacy in..	Adversary = Classifier	0.683	0.789	0.734
		Adversary > Classifier	0.694	0.783	0.735
LAFTR-EOD	No privacy	Adversary = Classifier	0.716	0.766	0.744
		Adversary > Classifier	0.720	0.766	0.744
	Privacy in..	Adversary = Classifier	0.678	0.779	0.734
		Adversary > Classifier	0.698	0.777	0.737
Unfair	No privacy	-	0.730	0.730	0.730
Acc/Fair ΔEOD					
Model	**Privacy**	**Adversary**	**min**	**max**	**mean**
LAFTR-DP	No privacy	Adversary = Classifier	0.723	0.764	0.747
		Adversary > Classifier	0.694	0.776	0.737
	Privacy in..	Adversary = Classifier	0.595	0.773	0.707
		Adversary > Classifier	0.604	0.763	0.714
LAFTR-EOD	No privacy	Adversary = Classifier	0.687	0.769	0.722

		Adversary > Classifier	0.668	0.785	0.728
	Privacy in..	Adversary = Classifier	0.594	0.776	0.702
		Adversary > Classifier	0.624	0.774	0.716
Unfair	No privacy	-	0.764	0.764	0.764

6. CONCLUSION

This research study aimed to develop a benchmark to evaluate differentially private fair representations across various model configurations and datasets and research results. The following key findings were obtained during the study:

- Firstly, the impact of privacy integration on fairness in the encoding process was examined. It was demonstrated that with successful weight initialization, any model could generate nearly fair representations. Additionally, statistical tests confirmed the equality of privacy-enabled models in all cases. The level of noise, as well as the number of parts it is injected into, did not affect the fairness outcome. The introduction of privacy improved fairness by up to 5%.
- Secondly, it was observed that privacy integration had a negative impact on model accuracy, with a typical decline of 1-3% compared to unprotected counterparts.
- Furthermore, the analysis of results revealed a pattern where strengthening the adversary could have a positive effect on the outcomes. However, careful fine-tuning and monitoring of the training process were necessary due to increased instability.
- Moreover, it was shown that any private fair models could achieve up to 5% advantage in the compromise between accuracy and fairness compared to the "unfair" model.

Based on the study's findings, it can be concluded that it is possible to train a fair and private model with acceptable accuracy and a high level of protection. However, achieving such results requires fine-tuning or fortunate circumstances, as privacy integration reduces stability in an already unstable model. Increasing fairness may require strengthening the adversary, which further adds to the instability.

REFERENCES

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., et al. (2016). Deep Learning with Differential Privacy, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security – CCS'16*. Vienna, Austria, 308–318, https://doi.org/10.1145/2976749.2978318

2. Bagdasaryan, E., & Shmatikov, V. (2019). *Differential Privacy Has Disparate Impact on Model Accuracy*. arXiv:1905.12101, [Online]. Available: https://arxiv.org/abs/1905.12101

3. Baig, S. M. (2022, February 2) *Stochastic Gradient Descent Algorithm (SGD).* [Online]. Available: https://www.researchgate.net/publication/358769475_Stochastic_Gradient_Descent_Algorithm_SGD

4. Calmon, F., Wei, D., Vinzamuri, B., Ramamurthy, K. N., & Varshney, K. R. (2017). Optimized Pre-Processing for Discrimination Prevention, *Proc. of Neural Information Processing Systems 31 (NIPS 2017)*. Long Beach, CA, https://doi.org/10.48550/arXiv.1704.03354

5. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique, *Journal of Artificial Intelligence Research*, **16**, 321–357, https://doi.org/10.1613/jair.953

6. Creager, E., Madras, D., Jacobsen, J.-H., Weis, M. A., Swersky, K., et al. (2019). *Flexibly Fair Representation Learning by Disentanglement*. arXiv:1906.02589, [Online]. Available: https://arxiv.org/abs/1906.02589

7. Sabitov, R. A., Smirnova, G. S., et. al. (2017). The concept of intelligent tutoring for enterprise staff as a component of integrated manufacturing control system development, *Advances in Systems Science and Applications*, **17**(1), 1–8.

8. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012). Fairness Through Awareness, *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS'12)*. Cambridge, USA, 214–226, https://doi.org/10.1145/2090236.2090255

9. Edwards, H., & Storkey, A. (2016). *Censoring Representations with an Adversary*. arXiv:1511.05897, [Online]. Available: https://arxiv.org/abs/1511.05897

10. Elazar, Y., & Goldberg, Y. (2018). *Adversarial Removal of Demographic Attributes from Text Data*. arXiv:1808.06640, [Online]. Available: https://arxiv.org/abs/1808.06640

11. Farrand, T., Mireshghallah, F., Singh, S. & Trask, A. (2020). *Neither Private Nor Fair: Impact of Data Imbalance on Utility and Fairness in Differential Privacy*. arXiv:2009.06389, [Online]. Available: https://arxiv.org/abs/2009.06389

12. Hardt, M., Price, E., & Srebro, N. (2016). *Equality of Opportunity in Supervised Learning.* arXiv:1610.02413, [Online]. Available: https://arxiv.org/abs/1610.02413

13. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. (2018). *Densely Connected Convolutional Networks*. arXiv: 1608.06993, [Online]. Available: https://arxiv.org/abs/1608.06993

14. Jagielski, M., Kearns, M., Mao, J., Oprea, A., Roth, A., et. al. (2019). *Differentially private fair learning*. arXiv:1812.02696, [Online]. Available: https://arxiv.org/abs/1812.02696

15. Kamiran, F., & Calders, T. (2011). Data preprocessing techniques for classification without discrimination, *Knowledge and Information Systems*, **33**, 1–33, https://doi.org/10.1007/s10115-011-0463-8

16. Lyu, L., He, X., & Li, Y. (2020). *Differentially Private Representation for NLP: Formal Guarantee and An Empirical Study on Privacy and Fairness*. arXiv:2010.01285, [Online]. Available: https://arxiv.org/abs/2010.01285

17. Madras, D., Creager, E., Pitassi, T., & Zemel, R. (2018). *Learning Adversarially Fair and Transferable Representations*. arXiv:1802.06309, [Online]. Available: https://arxiv.org/abs/1802.06309

18. McMahan, H. B., Ramage, D., Talwar, K., & Zhang, L. (2018). *Learning Differentially Private Recurrent Language Models*. arXiv:1710.06963, [Online]. Available: https://arxiv.org/abs/1710.06963

19. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K. & Galstyan, A. (2019). *A Survey on Bias and Fairness in Machine Learning*. arXiv:1908.09635, [Online]. Available: https://arxiv.org/abs/1908.09635

20. Pereira, M., Kshirsagar, M., Mukherjee, S., Dodhia, R., & Ferres, J. L. (2021). *An Analysis of the Deployment of Models Trained on Private Tabular Synthetic Data: Unexpected Surprises*. arXiv:2106.10241, [Online]. Available: https://arxiv.org/abs/2106.10241

21. Reddy, C., Sharma, D., Mehri, S., Romero-Soriano, A., Shabanian, S., et al. (2021). *Benchmarking Bias Mitigation Algorithms in Representation Learning through Fairness Metrics*. [Online]. Available: https://openreview.net/pdf?id=OTnqQUEwPKu

22. Singh, J., & Banerjee, R. (2019). A Study on Single and Multi-layer Perceptron Neural Network. *Proc. of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*. Erode, India, 35–40, https://doi.org/10.1109/ICCMC.2019.8819775.

23. Tato, A., & Nkambou, R. (2018). Improving Adam Optimizer. *Workshop track – ICLR 2018*. Vancouver, BC, Canada. https://doi.org/10.13140/RG.2.2.21344.43528

24. Tran, C., Dinh, M., & Fioretto, F. (2021). Differentially Private Empirical Risk Minimization under the Fairness Lens, *Advances in Neural Information Processing Systems (NeurIPS)*, **34**, 27555–27565, https://doi.org/10.48550/arXiv.2106.02674

25. Tran, C., Fioretto, F., & Van Hentenryck, P. (2021). Differentially Private and Fair Deep Learning: A Lagrangian Dual Approach, *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**(11), 9932–9939, https://doi.org/10.1609/aaai.v35i11.17193

26. Uniyal, A., Naidu, R., Kotti, S., Singh, S., Kenfack, P. J., & et al. (2022). *DP-SGD vs PATE: Which Has Less Disparate Impact on Model Accuracy?*. arXiv:2106.12576, [Online]. Available: https://arxiv.org/abs/2106.12576

27. Verma, S., & Rubin, J. (2018). Fairness definitions explained. *Proceedings of the 2018 IEEE/ACM International Workshop on Software Fairness (FairWare'18)*. IEEE, Los Alamitos, CA, 1–7, https://doi.org/10.1145/3194770.3194776.

28. Xie, L., Lin, K., Wang, S., Wang, F. & Zhou, J. (2018). *Differentially Private Generative Adversarial Network*. arXiv:1802.06739, [Online]. Available: https://arxiv.org/abs/1802.06739

29. Zhao, H., Coston, A., Adel T., & Gordon, G.J. (2020). *Conditional Learning of Fair Representations. arXiv:1910.07162*, [Online]. Available: https://arxiv.org/abs/1910.07162

30. Smirnova G. S., Sabitov R. A., Korobkova, E. A., Sabitov, Sh. R. (2017). Modeling production facility as a dynamic integrated interacting objects system, *Procedia Computer Science*, **112**, 965–970, https://doi.org/10.1016/j.procs.2017.08.136

