Towards a Holistic Framework for Machine Learning Trustworthiness

by

Elie Rizk

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master's of Science
(Computer and Information Science)
in the University of Michigan-Dearborn
2024

Master's Thesis Committee:

Assistant Professor Birhanu Eshete, Chair
Professor Di Ma
Assistant Professor Srijita Das
Assistant Professor Ang Li

Elie Rizk
elirizk@umich.edu
ORCID iD: 0009-0008-8551-0910

DEDICATION

For my Mom and Dad,
Thank you for everything.
And for Roxy,
You taught me how to love even though you were a dog

ACKNOWLEDGEMENTS

I would like to acknowledge my research supervisor and thesis advisor Professor Birhanu Eshete for his valuable guidance throughout this work. His lab, the Data-driven Security and Privacy (DSP) Lab in the Computer and Information Science department, offered me the computing resources needed to run my experiments.

I would also like to acknowledge the Department of State's Fulbright program for sponsoring my education and stay in the United States. Their scholarship allowed a Lebanese student like myself to pursue an American graduate education in Michigan.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

LIST OF ABBREVIATIONS

AI Artificial Intelligence

CNN Convolutional Neural Networks

COSMOS Conditioned One-Shot Multi-Objective Search

DDP Difference of Demographic Parity

DEO Difference of Equality of Opportunity

DNN Deep Neural Networks

DP Differential Privacy

DP-RFL Differentially Private - Robust and Fair Learning

DP-SGD Differentially Private - Stochastic Gradient Descent

DPFR Demographic Parity Fairness Regularizer

EPO Exact Pareto Optimal

ERM Empirical Risk Minimization

FGSM Fast Gradient Sign Method

IP Intellectual Property

KL Kullback–Leibler

LLM Large Language Models

MGDA Multiple Gradient Descent Algorithm

MIA Membership Inference Attack

ML Machine Learning

MNIST

MOO Multi-Objective Optimization

MTL Multi-Task Learning

PATE Private Aggregation of Teacher Ensembles

PFL Pareto Front Learning

PGD Projected Gradient Descent

PHN-HVI Pareto Hypernetworks with Hypervolume Indicator

SGD Stochastic Gradient Descent

TRADES TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization

ABSTRACT

Machine learning models are being heavily deployed in critical environments such as healthcare, transportation, and surveillance. However, most models still fall short in jointly satisfying different trustworthiness requirements, namely being simultaneously private, fair, and robust, while maintaining acceptable model accuracy. This work aims to bridge the gap between the trustworthiness objectives of machine learning models by proposing a unified framework to develop and assess holistically trustworthy models. We empirically confirm the existence of a four-dimensional trade-off among utility, privacy, fairness, and robustness and assess how pronounced the trade-off is for each subset of objectives. We further propose and evaluate different algorithms and loss functions in the context of generating a high quality multi-dimensional Pareto frontier.

CHAPTER 1

Introduction

Motivation: The field of Machine Learning (ML) has pushed the limits of Artificial Intelligence (AI), exceeding human expectations on what we previously thought achievable by computers. While some ML models have been deployed in low-risk environments like e-mail spam filtering or commercial chatbots, others are being used in critical and high-stakes settings. Today's ML models are behind applications ranging from self-driving vehicles [42] and credit loan approvals [65] to patient diagnosis [31] and recidivism risk prediction [62].

In recent years, ML models such as Convolutional Neural Networks (CNN) and Large Language Models (LLM) have reached impressive performance in terms of accuracy. However, they still fall short on other socially salient requirements, hindering their trustworthiness for high-stakes decisions [69, 17]. For example, while facial recognition models are widely deployed in surveillance systems, research has proven the elevated risk of misidentification for women and people of color [39] which is particularly concerning when used by law enforcement. This points to a larger problem: the growing gap between highly accurate ML and trustworthy ML satisfying wide-ranging objectives across robustness, privacy, transparency and fairness. For an ML model to be trusted by users, it must satisfy the following desiderata:

- **Utility**: Models have to be highly accurate to be useful on a given task. Various novel architectures and algorithms have been developed to ensure well performing models across application domains when evaluated via test-time accuracy.

- **Robustness**: At train-time, the training data needs to be clean of any adversarial condemnations that could be realized via data poisoning attacks [8]. At test-time, models shouldn't be easily fooled by slightly modifying their input through adversarial examples [21], or fall victim of adversarial approximation of their decision boundary via model extraction attacks [60].

- **Privacy**: When trained on confidential or privacy sensitive data, models shouldn't expose their training data via attacks such as membership inference [56], attribute inference [26], training point inference [9] and model inversion [19]. In addition, the model itself might be proprietary for the ML owner emphasizing the need to protect against model extraction attacks with implications on model privacy.

- **Fairness**: Models shouldn't discriminate against protected attributes of individuals: decision-making at inference time shouldn't be biased against a particular demographic attribute such as race, gender, or religion.
- **Interpretability**: In most high-stake applications where model decision impacts people's lives, models should be transparent regarding their decision-making process to inform their user or other stakeholders (e.g., regulators) the reasoning behind their output as opposed to being treated like a black-box system.

Given the evolving nature of the science and public discourse around trustworthy ML, the above list of trustworthiness objectives is by no means exhaustive. However, most critical usage of ML sits at the intersection of two or more of the aforementioned objectives. For example, in the context of automating patient diagnosis in a hospital, ML models shouldn't exhibit accuracy discrepancies for different demographics [53]. They should also remain private with respect to the training data (medical records are confidential information), robust against data perturbation, and transparent in their decision-making for a domain expert to intervene on demand. As additional examples, we provide below a summary of high-stakes applications with respect to the ML trustworthiness goals described earlier.

Domain	**Task**	**Utility**	**Robustness**	**Privacy**	**Fairness**	**Interpretability**
Healthcare	Patient diagnosis	✓	✓	✓	✓	✓
Education	Teacher substitute	✓	✓		✓	✓
Transportation	Self-driving vehicles	✓	✓			✓
Criminal Justice	Risk of recidivism	✓		✓	✓	✓
Surveillance	Facial Recognition	✓	✓	✓	✓	✓
Chatbots	Conversing	✓	✓	✓	✓	✓

Table 1.1: Desirable ML trustworthiness objectives for tasks in high-stakes applications

The need for machine learning to jointly satisfy multi-dimensional trustworthiness across different settings emphasize the urgency for approaches that reconcile the different objectives. Achieving trustworthy ML requires research dedicated to producing holistically optimal models with respect to the desiderata enumerated above as opposed to accuracy in isolation.

Previous Work and Limitations: Even though the public might require models to satisfy all the aforementioned properties, research still mostly focused on each objective separately. Since achieving each objective alone is in itself a long-standing challenge, little effort has been dedicated to studying their intersectionality let alone resolve them holistically. Research has attempted reconciling at most three objectives, namely utility, data privacy and fairness in [71] or utility, data privacy and test-time robustness in [70]. This motivates our present work to study and reconcile four of the desiderata of trustworthy ML. Towards this goal, this work focuses on the intersectionality of utility, test-time robustness, data privacy and fairness.

Approach Overview: This work develops a unified systematic framework to reconcile and assess the four trustworthiness objectives of utility, privacy, robustness and fairness, in an efficient and scalable manner through evaluating the quality of the Pareto optimal models produced. While we aim to train models satisfying all four objectives, our framework provides practitioners the freedom to modify the number of trustworthiness dimensions considered without significantly modifying our high-level approach. In order to satisfy the different desiderata of machine learning systems, we propose different algorithms to generate holistically trustworthy ML models, allowing us to select the Pareto-optimal models and approximate the Pareto front of the multi-objective problem. The Pareto front is a direct manifestation of the multi-faceted trade-off across the objectives: it provides us with a concrete way to visualize and establish the achievable compromise given the inherent constraints of the problem, i.e., model architecture, dataset, algorithm, objectives, etc.

Evaluation Highlights: We empirically compare the quality of the Pareto frontier generated through different algorithms and loss functions across the objective space. As such, our experiments confirm previous research on the nature of the multi-faceted trade-off. Since our research tackles four-dimensional objectives, our approach provides a novel method to compare the degree of trade-off among subset of lower-dimensional objective spaces. For example, we show that the Utility-Robustness-Privacy and Robustness-Fairness-Privacy trade-off is significantly more pronounced than the Utility-Fairness-Privacy trade-off for the dataset we considered. Our findings further demonstrate that applying Multi-Objective Optimization (MOO) algorithms in the context of machine learning outperforms plain ERM regularization, providing a direction for future research since the intersectionality of MOO and ML trustworthiness is still left mostly unexplored.

Contributions: Towards comprehensive assessment of ML trustworthiness, this work makes the following contributions:

1. We introduce the first work that tackles the four-dimensional trade-off of utility, privacy, fairness and robustness for generating holistically trustworthy ML models.

2. Our approach proposes a novel framework that treats the four dimensions of machine

learning trustworthiness in the context of a multi-objective optimization.

3. We propose surrogate loss functions to regularize the model's natural loss with robustness and fairness metrics in a differentially private manner.

4. Our work is the first to apply differential privacy and include robustness metrics in MOO algorithms such as multi-step gradient descent and Pareto front learning.

5. We provide experimental results that capture the shape of the trade-off across different objectives and obtain a set of Pareto-optimal models.

Thesis Organization: The rest of this thesis is organized as follows. Chapter 2 introduces ML preliminaries and an in-depth background discussion on various ML trustworthiness objectives as well as the state of the art frameworks for MOO. We cover related works and position this work with respect to existing literature in Chapter 3. Our main approach to resolving the different ML trustworthiness objectives is described in Chapter 4. The evaluation of our approach on different datasets is provided in Chapter 5. Finally, Chapter 6 concludes this work and provides future direction for further research. Some mathematical equations and algorithms are defined in Appendices A and B respectively as a reference for the reader.

CHAPTER 2

Background

2.1 Machine Learning Setting

We will consider a supervised machine learning model θ trained on a classification task: $\mathcal{X} \rightarrow \mathcal{Y}$ where $\mathcal{X} \in \mathbb{R}^d$ is the feature space of the input and $\mathcal{Y} \in \mathbb{R}^k$ corresponds to the probability distribution over K class labels or the one-hot encoded ground truth. In addition, a training sample has a sensitive attribute $s \in \mathcal{S}$ which may or may not be contained in the input space $\mathcal{X}$. The samples (x, y) follow an underlying data distribution $\mathcal{D}$. Without loss of generality, since s and y are categorical variables, we will denote $\mathcal{S} = [S] = \{1, 2, ..., S\}$ and $\mathcal{Y} = [K] = \{1, 2, ..., K\}$. We will also make use of the predicted label $\hat{y} = arg\,max\big(f(x)\big)$.

Traditional training algorithms aim to produce a predictor $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$, parameterized by θ, with minimal risk as measured by some non-negative loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+$. For classification tasks, the Cross-Entropy loss [54] is conventionally used to evaluate the model's performance defined as $L_{CE}\big(f(x), y\big) = -\log f(x)_k$ with k as the true label. The risk of a predictor f_θ is measured by the expectancy of the loss: $\mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(f(x), y)]$. Since the underlying probability distribution $\mathcal{D}$ is unknown, we resort to calculating the empirical risk which relies on the empirical expectation $\hat{\mathbb{E}}$ with respect to the training data. Empirical Risk Minimization [68] refers to the training algorithm which produces model parameters that minimize the empirical risk and can be written as:

$$\hat{\theta} = arg\,\underset{\theta}{min}\ \hat{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\Big[\mathcal{L}(f_\theta(x), y)\Big] \tag{2.1}$$

While ERM often succeeds in finding the optimal parameters to achieve high accuracy, it falls short in satisfying other objectives. Notably, it fails to consider real-world ML trustworthiness objectives throughout the training stage of the model or after deployment such as robustness against adversarial manipulations, privacy of the training samples and fair decision-making. This renders ML trustworthiness questionable particularly when deployed in high-stakes real-world settings. Next, we describe these trustworthiness objectives based on current state of the art.

2.2 Robustness

A large body of research has shown that ML models such as deep neural networks (DNNs) are vulnerable to adversarial manipulation attacks such as adversarial examples [28], training data poisoning [8], and model extraction [60]. There exists two main forms of robustness in the context of machine learning: train-time and test-time. The former indicates resilience of the model against perturbation on the dataset at train-time. This type of robustness provides a defense against data poisoning attacks [8], where an adversary intentionally worsens the model's performance by inserting maliciously crafted input into the training dataset, e.g. backdoor attacks. In contrast, test-time robustness ensures that the model isn't susceptible to data perturbation at inference time, i.e. adversarial attacks [21].

In this work, we focus on robustness against the widely studied adversarial example attack[1]. This type of attack fools the model by adding small calibrated noise δ, also called adversarial perturbation, from a perturbation set Δ to the input x which generates its adversarial version $x' = x + \delta$. We constrain the amount of manipulation that the attacker is allowed to make on the input by forcing x' to be close to x, i.e., the set Δ has to represent acceptable "imperceptible" noise. Typically, the set Δ is defined as the ℓ_p-norm ball: $\Delta = \mathbb{B}_p(\delta, r) = \{\delta \mid ||\delta||_p \leq r\}$. This restriction ensures that the distance between the adversarial example x' and its benign counterpart x does not exceed a perturbation budget r: $x' \in \mathbb{B}_p(x, r)$. Note that the ℓ_p-norm Ball is defined as $\mathbb{B}_p(x, r) = \{x' \in \mathbb{R}^d| \; ||x' - x||_p \leq r\}$ and the ℓ_p-norm is defined as $||\delta||_p = \left(\sum_i |\delta_i|^p\right)^{1/p}$ for $1 \leq p < \infty$ [33]. For $p = \infty$, the ℓ_∞ norm is defined as the maximum perturbation allowed on any feature: $||\delta||_p = \max_i |\delta_i|$.

As such, a successful untargeted adversarial attack crafts a perturbation $\delta \in \Delta$ that modifies the predicted label: $f(x + \delta) \neq f(x)$. A targeted adversarial example intends to fool the model into misclassifying the input as a different label y': $f(x + \delta) = \hat{y}'$ with $\hat{y}' \neq \hat{y}$. Multiple versions of adversarial attacks have been developed to date, and they can be classified at a high-level as white-box or black-box approaches. The former has access to the model parameters and crafts adversarial examples with the use of the model's gradients while the latter doesn't have access to the internal workings of the model and have to fool the model through the sole use of its queries' output. Our threat model will be limited to untargeted white-box attacks for the rest of this work. One of the simplest yet effective methods to

[1] We will use the term robustness to refer exclusively to test-time robustness for the rest of this paper

generate adversarial examples is the Fast Gradient Sign Method (FGSM) [22] which modifies the original input by stepping along the direction of the loss function's gradient with respect to the input. It is formalized as follows:

$$x' = x + \delta \cdot sign(\nabla_x L(f(x), y)) \tag{2.2}$$

Kurakin et al. [32] proposed an iterative FGSM method called Projected Gradient Descent (PGD) which iterates over the gradient of the loss while clipping the perturbation at each iteration. The equation to generate the adversarial input at the i^{th} iteration is the following:

$$x'_i = Clip_{x,\delta}\Big\{x'_{i-1} + \alpha \cdot sign(\nabla_x L(f(x'_{i-1}), y))\Big\} \tag{2.3}$$

α corresponds to the step size at each iteration and the procedure starts with the original input $x'_0 = x$.

Defenses aimed at making ML models robust against adversarial examples have taken multiple avenues broadly focused on model hardening (e.g., via regularization, perturbation or distillation), input pre-processing (e.g., via data transformation or de-noising) and adversarial training. Regardless of the specifics, these defenses intend to produce a model with high accuracy on both natural and adversarial inputs. We formally define the robust classification error under the threat model of δ-bounded perturbations as $\mathcal{R}_{rob}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{1}\{\exists x' \in \mathbb{B}(x,r) | \hat{y} \neq y\}$ as opposed to the natural classification error $\mathcal{R}_{nat}(f) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\mathbb{1}\{\hat{y} \neq y\}$ [73].

Adversarial Training: Finding a model robust to adversarial example attacks reduces to training it on adversarial rather than original data which produces the variant of ERM defined in Equation (2.4) by Madry et al. [36].

$$\hat{\theta} = \underset{\theta}{arg\,min}\ \hat{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\Big[\max_{||\alpha||_p \leq r} L(f_\theta(x + \alpha), y)\Big] \tag{2.4}$$

Lipschitz Continuity: Alternatively to the robust definition in Equation (2.4), numerous research has been dedicated to study the relationship between Lipschitzness and robustness [29, 48, 72]. Formally, a function $f : \mathcal{X} \to \mathcal{Y}$ is globally Lipschitz continuous if there exists $\ell \geq 0$ such that

$$||f(x_1) - f(x_2)|| \leq \ell\ ||x_1 - x_2|| \qquad\qquad \forall x_1, x_2 \in \mathcal{X} \tag{2.5}$$

The smallest ℓ for which Equation (2.5) holds is called the Lipschitz constant of f. This constant constitutes an upper bound on the degree to which the output of f changes with respect to the change on the input space. Achieving small Lipschitz continuity has been proven to produce robust classifiers [64], because Equation (2.5) provides a bound on the sensitivity of a model to small perturbations of the data. However, calculating the Lipschitz constant has proven to be computationally hard and intractable for large DNNs. This persuaded researchers to provide estimates for the local Lipschitz constant to be used as a proxy for the sensitivity of the model to adversarial input. One such proxy is TRadeoff-inspired Adversarial DEfense via Surrogate-loss minimization (TRADES) which incorporates the Kullback–Leibler (KL) divergence of the output defined in Equation (A.1) as a regularization term [73]. This allows the training algorithm to limit the sensitivity of the model to slight perturbations. The corresponding ERM variant is formalized as:

$$\hat{\theta} = \underset{\theta}{arg\,min}\ \hat{\mathbb{E}}_{(x,y)\sim\mathcal{D}}\Big[L(f_\theta(x), y) + \beta \cdot \max_{||\alpha||_p \leq r} D_{KL}(f_\theta(x+\alpha), f_\theta(x))\Big] \tag{2.6}$$

Conflicting research has emerged on the relationship between robustness and accuracy. Tsipras et al. [63] showed that robustness might be incompatible with natural accuracy. Given a simple binary classification setting, the authors prove the nonexistence of a robust and accurate model: increasing robustness degrades accuracy on the prediction task. Surprisingly, this is due to the characteristic of the underlying data distribution as opposed to the size of the dataset which goes against the popular assumption that robust and accurate models can be achieved by collecting more data. They also argue that robust classifiers learn fundamentally different feature representations than standard classifiers. In particular, robustness pushes models to learn higher quality representations aligning with salient data characteristics and human perception which highlights the benefits of incorporating robustness even at the cost of accuracy.

On the other hand, Yang et al. [72] argue the opposite: the trade-off between accuracy and robustness isn't inherent and reconciling them should be achievable if the classes can be separated with finite margins. Their approach is based on rounding the function to be locally Lipschitz around the data, limiting its sensitivity to perturbation while remaining accurate due to the class-separable nature of the data distribution.

2.3 Privacy

Early research attempted to define data privacy in the realm of statistical databases[2] as a guarantee that query-based access to the database doesn't reveal any information about an individual that could not be learned without access to the query responses [10]. This

[2] ML models can be thought of as an interactive database

"nothing is learned" principle is analogous to the notion of semantic security in cryptosystems which states that nothing new should be learned about the plaintext whether or not the ciphertext is known to the adversary. While semantically secure cryptosystems have been widely developed, Dwork rigorously proved that data privacy under the former definition is unachievable because of the presence of auxiliary information [14]. For example, an analysis showing that smoking causes cancer coupled with the auxiliary information that Tom is a smoker reveals that that Tom is at high risk for cancer. Surprisingly, the unintentional effect on Tom's privacy resulting from publishing the study holds whether he participated in the original study or not. In addition, attempts to anonymize or remove personally identifiable information from the data are still vulnerable to linkage attacks which pose the risk of re-indentification for participating individuals.

Differential Privacy: In contrast, the notion of Differential Privacy (DP) was introduced to remedy the challenges mentioned above, namely resistance to auxiliary information, linkage attacks, and resilience to post-processing. Differential privacy ensures that nothing new will be learned about an individual whether or not they participate in a study[3], i.e. the individual will not be affected by providing their data.

Formally, differential privacy is defined as a property for a randomized algorithm M stating that the following inequality holds for any two neighboring[4] datasets D and D' and for any event $S \subseteq Range(M)$:

$$\mathbb{P}[M(D) \in S] \leq e^{\epsilon}\mathbb{P}[M(D') \in S] + \delta \tag{2.7}$$

As such, M is said to be (ϵ, δ)-DP [15]. The inequality above limits the extent to which an individual data point affects the model's behavior. The parameter ϵ (called the privacy budget) imposes an upper bound on the influence of a single record while the δ parameter[5] (called the failure probability) allows for a slight relaxation of the inequality. It is desired for a model to achieve a low ϵ bound as that would result in low privacy leakage of individual's data.

In the context of ML training, differential privacy can be achieved through Differentially Private - Stochastic Gradient Descent (DP-SGD): a modified version of Stochastic Gradient Descent (SGD) [1]. DP-SGD limits privacy leakage through (1) gradient clipping and (2) random noising. Per sample gradients are clipped to a maximum bound C to limit the sensitivity of each record's gradients which prevents individual records from contributing too much to the model's parameters. After clipping the gradients, Gaussian noise following

[3] For ML models participating means being part of the training data

[4] Neighboring conventionally means differing in one record

[5] Not to be confused with the noise δ of adversarial examples

the normal distribution $\mathcal{N}(0, \sigma^2 C^2 I)$ is added to enforce some randomness on top of the sample's gradients. Equation (2.8) calculates the new batch gradients used in DP-SGD with group size[6] L and noise $\xi \sim \mathcal{N}(0, \mathbf{I})$ at iteration t. The pseudo-code of the algorithm is present in Appendix B.

$$\tilde{g}_t \leftarrow \frac{1}{L} \sum_{(x,y) \in L_t} clip_c\Big(\nabla L\big(f_\theta(x), y\big)\Big) + \frac{\sigma C}{L}\xi \tag{2.8}$$

Typically, the noise added to the gradients considerably hurts the accuracy of the model which poses a significant barrier against the wide deployment of differentially private learning [59]. De et al. [11] remedy this problem by performing various types of data and model modifications, namely: group normalization, increased batch size, weight standardization, data augmentation, and parameter averaging. Their method succeeded in increasing the accuracy by 10% for a DP-SGD model trained on CIFAR-10. In particular, they perform K-fold data augmentation in a differentially private way by averaging the gradients of the augmented data before clipping it. Since this approach doesn't increase the sensitivity of the gradient per training sample, it doesn't incur any additional privacy cost. Their method is formalized as:

$$\tilde{g}_t \leftarrow \frac{1}{L} \sum_{(x,y) \in L_t} clip_c\Big(\frac{1}{K} \sum_{j \in \mathcal{K}_i} \nabla L\big(f_\theta(x_j), y\big)\Big) + \frac{\sigma C}{L}\xi \tag{2.9}$$

2.4 Fairness

Fairness in the context of ML aims to ensure that a sensitive attribute[7] doesn't influence the model's prediction. Because of the pervasive nature of bias within the training data and the data processing pipeline, the elimination of sensitive attributes doesn't necessarily lead to a fair model. This is because of the red-lining effect [6]: the presence of other features highly correlated with the sensitive attribute allows the model to learn spurious correlations. For example, if a bank has historically excluded marginalized communities from loan approvals, removing explicit racial information from the dataset will not necessarily make it less biased. If postal codes are highly correlated with ethnicity, the model might learn these spurious correlations and base its predictions on unfair criteria. This highlights the need for discrimination-aware classification where sensitive attributes are known for each sample during training and a fairness constraint is imposed on the model to ensure

[6] In DP, a group (or lot) is a training batch with additional Poisson sampling which randomly picks samples for training

[7] Examples include gender, race, religion, etc...

discrimination-free decision making.

Fairness can be imposed on ML models through pre-processing, in-processing or post-processing schemas. Pre-processing approaches attempt to eliminate any underlying discrimination within the dataset before training the model, e.g., removing class imbalances through undersampling. Post-processing methods impose discrimination-free classification at inference time, e.g., rejecting a query if answering it might violate a set fairness guarantee [71]. Our work will mainly focus on in-processing schemes which aim to significantly reduce discrimination during the training stage of the model. This is conventionally done by adding a fairness regularizer or changing the objective function [38, 12]. Multiple loss functions have been introduced to capture the degree of discrimination of a model with respect to a sensitive attribute, namely the difference of demographic parity, the difference of equality of opportunity and the demographic parity loss. We will first provide the definition of the fairness metrics for binary classification $y \in \{-1, 1\}$, where the positive label is regarded as the "desirable outcome."

Demographic Parity: A classifier f satisfies demographic parity if the probability of the outcome for a sample is independent of the value of the sensitive attribute:

$$\mathbb{P}[\hat{Y} = 1 | S = s] = \mathbb{P}[\hat{Y} = 1 | S \neq s]$$

Difference of Demographic Parity (DDP): the difference between the expected values of the demographic parity over the probability distribution:

$$DDP(f) = \mathbb{E}[\mathbb{1}_{\hat{Y}=1} | S = s] - \mathbb{E}[\mathbb{1}_{\hat{Y}=1} | S \neq s] \tag{2.10}$$

Equality of Opportunity is satisfied when the probability of a true positive remains the same regardless of the sensitive attribute:

$$\mathbb{P}[\hat{Y} = 1 | S = s, y = 1] = \mathbb{P}[\hat{Y} = 1 | S \neq s, y = 1]$$

Similarly to DDP, **Difference of Equality of Opportunity (DEO)** is defined as the difference in the expected values:

$$DEO(f) = \mathbb{E}[\mathbb{1}_{\hat{Y}=1} | S = s, Y = 1] - \mathbb{E}[\mathbb{1}_{\hat{Y}=1} | S \neq s, Y = 1] \tag{2.11}$$

Padh et al. [44] proposed a differentiable hyperbolic tangent relaxation of the above definitions to more accurately approximate their true values. Particularly, they use the following equations:

$$\widehat{DDP}(f) = \frac{1}{n_s}\sum_{S=s} t(c, f(x)) - \frac{1}{N - n_s}\sum_{S\neq s} t(c, f(x)) \tag{2.12}$$

$$\widehat{DEO}(f) = \frac{1}{n_s}\sum_{S=s,Y=1} t(c, f(x)) - \frac{1}{N - n_s}\sum_{S\neq s,Y=1} t(c, f(x)) \tag{2.13}$$

where n is the total number of samples in the batch, n_s is the number of samples with sensitive attribute s and $t(c, x) = tanh(c \cdot max(0, x))$. The above loss functions have been conventionally used to impose an additional fairness constraint on the training algorithm in a multi objective framework as opposed to being added to the natural loss to regularize ERM [44, 51].

Subgroup Fairness: Alternatively, Oneto et al. [43] frame fairness as a multi-task learning problem by dividing the population into subgroups based on the sensitive attribute and minimizing the average loss per subgroup as opposed to the average loss of the entire batch. This approach has the benefit that the model is trained to remain highly accurate for each subgroup as opposed to averaging its performance on the entire batch.

Yaghini et al. [71] extended the Equation of the difference of demographic parity (2.10) for multi-label classification by proposing to condition the demographic disparity by the class label and sensitive attribute. Accordingly, **demographic disparity** $\Gamma(s, k)$ is defined as the difference between the probability of predicting label k for a sample with sensitive attribute s and the probability of predicting k for any other subgroup: $\Gamma(s, k) = \mathbb{P}[\hat{y} = k|S = s] - \mathbb{P}[\hat{y} = k|S \neq s]$. The demographic disparity is estimated in practice by taking the empirical expectation of the above probabilities:

$$\widehat{\Gamma}(s, k) = \frac{|\{\hat{Y} = k, S = s\}|}{|\{S = s\}} - \frac{|\{\hat{Y} = k, S \neq s\}|}{|\{S \neq s\}}$$

A model is said to satisfy γ-bounded demographic parity (γ-DemParity) if $\Gamma(s, k) \leq \gamma$ for all class labels k and sensitive attributes s. This leads to the natural definition of a Demographic Parity Fairness Regularizer (DPFR)[8] defined as the maximum demographic disparity:

$$\begin{aligned} DPFR(\theta) &= \max_k \max_s \widehat{\Gamma}(s, k) \\ &= \max_k \max_s \left\{ \frac{|\{\hat{Y} = k, S = s\}|}{|\{S = s\}} - \frac{|\{\hat{Y} = k, S \neq s\}|}{|\{S \neq s\}} \right\} \end{aligned} \tag{2.14}$$

[8] It is also referred to as the Demographic Parity Loss

The above loss functions can be added to the natural loss to produce a regularized form of ERM that accounts for discrimination. When privacy is an objective that we aim to simultaneously achieve without further consuming from the privacy budget, the fairness loss can be computed over a public dataset as opposed to over every training batch. This method doesn't incur any additional privacy budget consumption while training.

Research studying the effect of imposing fairness on model accuracy has empirically shown the existence of a trade-off between the two: imposing bias mitigation strategies on a model might compromise its accuracy [45]. This led numerous research to frame the problem of obtaining a fair and accurate model as a multi-objective optimization problem where they consider a fairness loss[9] as one of the objectives to be minimized [67, 44, 51].

2.5 Multi Objective Optimization in ML

Recent research has developed new approaches to solving multi-task learning in Deep Neural Networks (DNN) by framing it as a MOO problem [52, 34]. Each task's objective t is represented by a loss function $\mathcal{L}_t : \mathcal{Y}_t \times \mathcal{Y}_t \to \mathbb{R}_+$. As such, we obtain a vector of per task losses to be minimized:

$$min_\theta \mathcal{L}(\theta) = (\mathcal{L}_1(\theta), \mathcal{L}_2(\theta), ..., \mathcal{L}_m(\theta))^T \tag{2.15}$$

Equation (2.15) requires an MOO algorithm to minimize across its objectives in order to learn the tasks jointly. However, the tasks usually conflict with each other, preventing a single optimal solution. Instead, we obtain a set of dominant solutions called Pareto optimal representing the degree of trade-off achieved across the objectives.

Pareto Dominance: A solution θ_a dominates another solution θ_b if $\mathcal{L}_i(\theta_a) \leq \mathcal{L}_i(\theta_b)$ for all objectives i and $\mathcal{L}(\theta_a) \neq \mathcal{L}(\theta_b)$, i.e., there exists one task where the inequality is strict.

Pareto Optimality: A solution θ^* is said to be Pareto optimal if there exists no other solution θ that dominates θ^*. The set of Pareto optimal solutions is called the Pareto set $\mathcal{P}$.

Pareto Front: The m-dimensional manifold of the Pareto set in the objective space is called the Pareto front $\mathcal{F} = \{\mathcal{L}(\theta) \in \mathbb{R}^m_+ | \theta \in \mathcal{P}\}$.

The Pareto front represents the ultimate trade-off of the objectives allowing practitioners to examine the achievable compromise under the different constraints of the problem and compare it with the desirable solutions sought. For example, when two loss functions are conflicting, producing solutions on the Pareto front is more tractable than finding one solution minimizing both losses (as that solution usually doesn't exist).

[9] Typically DDP, DEO or difference in false positive rates

The main approaches that have been developed to solve Equation (2.15) for DNNs fall under three categories: linear scalarization, gradient-based approaches, and Pareto front learning.

2.5.1 Linear Scalarization

The most common approach to solve an MOO problem is through linear scalarization which reduces the problem to a single objective optimization by performing a weighted sum of each objective. Given a weight vector $w \in \mathbb{R}_+^m$, the new surrogate loss obtained can be expressed as:

$$\mathcal{L}(\theta) = \sum_{i=1}^{m} w_i \mathcal{L}_i(\theta) \tag{2.16}$$

where w_i is the weight assigned to task i. Note that the weights have to be set a priori before optimization and their specific values highly influence the convergence of the model. In addition, this method only converges to the convex part of the Pareto front and is unable to find solutions on the concave part of it [34]. Theorem (1) proves that the solutions found using Equation (2.16) under risk minimization are Pareto optimal.

Theorem 1. *Let $w \in \mathbb{R}_+^m$ any preference vector, $\mathcal{L}$ the vector of losses, and θ_w^* the solution of the following equation which applies risk minimization on the surrogate loss:*

$$\theta_w^* = arg\,\underset{\theta}{min}\ \mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\sum_{i=1}^{m} w_i \mathcal{L}_i(f_\theta(x), y)\Big]$$

Then θ_w^* is Pareto optimal.

Proof. We will adopt a proof by contradiction. Assume that the solution of the equation in Theorem 1 (θ_w^*) isn't Pareto optimal. This implies that there exists another solution θ' which dominates θ_w^*, i.e., $\mathcal{L}_i(\theta') \leq \mathcal{L}_i(\theta_w^*)$ for all objectives i and $\mathcal{L}(\theta') \neq \mathcal{L}(\theta_w^*)$. Therefore, for any vector $w \in \mathbb{R}_+^m$, it should hold that $\sum_{i=1}^{m} w_i \mathcal{L}_i(\theta') < \sum_{i=1}^{m} w_i \mathcal{L}_i(\theta_w^*)$ which is a contradiction of our original assumption that θ_w^* is a solution of equation (1). □

However, the loss functions of DNNs are non-convex with respect to the model parameters θ and the empirical expectation is used since the true distribution is unknown. As such, solutions to the above equation only yields approximate solutions $\hat{\theta}$ which form an approximation to the Pareto set $\hat{\mathcal{P}}$. This in turn requires training multiple models with random preference vectors r to discover the Pareto front, making it intractable at the scale of a DNN as the number of objectives increase. Despite its limitations, linear scalarization serves as a foundational technique in MOO, providing a starting point for more advanced schemes.

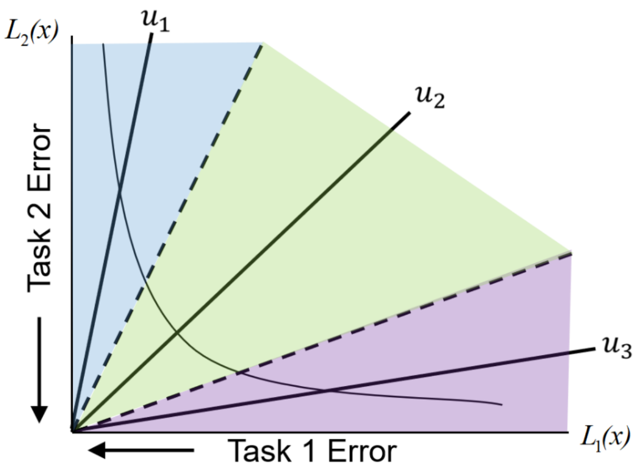

Figure 2.1: Pareto MTL decomposes the objective space into different preference vectors u_i and finds one Pareto solution inside the preference region [34].

2.5.2 Gradient-based Methods

To overcome the limitation of linear scalarization, researchers have attempted to reach Pareto optimal models by first calculating the separate gradients of each loss function $\mathcal{L}_i$ before solving an optimization problem aiming at finding the final gradient to use for the batch. Early research has applied the Multiple Gradient Descent Algorithm (MGDA) [16] to train DNNs on multiple tasks at once [52]. However, this early approach failed to produce models on the entire Pareto front and instead converges to a specific region of the front regardless of the number of iterations of the algorithm.

To resolve this problem, Lin et al. [34] introduced their Pareto MTL algorithm which trains a set of models and modifies its behavior according to pre-selected n_r preference rays r_i as shown in Figure 2.1. The rays divide the objective space into different regions of the Pareto front. Accordingly, the algorithm trains n_r different models θ_i to converge to the region of the Pareto front specified by the ray r_i. The algorithm is able to produce a diverse set of Pareto-optimal models representing different trade-offs on the objective space.

Lin et al. [34] also proved that gradient-based methods are equivalent to an adaptive linear scalarization of the loss vector. In other words, the method they developed reduces to performing a weighted sum of the losses with dynamically varying weights at each iteration according to the gradients calculated.

Further research has focused on improving the efficiency of the multiple gradient descent approach to obtain models converging closer to the preference rays. For example, Mahapatra and Rajan [37] developed the Exact Pareto Optimal (EPO) search algorithm. Their approach combines gradient descent with controlled ascent to traverse the Pareto front and obtain models closest to the given preference vector regardless of their initialization.

2.5.3 Pareto Front Learning

Pareto Front Learning (PFL) has emerged as a recent approach to estimate the entire Pareto front with a single model conditioned on the preference vector. Effectively, this approach expands the input space to include the preference vectors: $\mathcal{X} \times \mathbb{R}^m_+ \to \mathcal{Y}$. In practice, this is done by concatenating the input x with a vector r and training the model on this joint feature space. The model effectively conditions its behavior on the given preference ray, allowing for users to specify their preference at decision-time as opposed to fixing one model per preference vector at train-time. After training, the Pareto front is estimated by running the model on the same input with different preference rays.

Ruchte and Grabocka [51] developed a Conditioned One-Shot Multi-Objective Search (COSMOS) algorithm to train a conditioned model on the preference vectors. During training, the preference rays are sampled from a Dirichlet distribution with parameter α to ensure their elements are positive and sum up to one. The authors also used a novel loss function that consists of a weighted sum of the loss and preference vectors regularized with a penalty term. The penalty term corresponds to maximizing the cosine similarity between the loss vector and the preference ray, effectively decreasing the angle between them and forcing the solutions obtained to be well-spread across the objectives.

$$\theta^* = arg\,\underset{\theta}{min}\ \mathbb{E}_{r\sim\mathrm{Dir}(\alpha);(x,y)\sim\mathcal{D}}\Big[r \cdot \mathcal{L}(\theta) - \lambda \frac{r \cdot \mathcal{L}(\theta)}{||r|| \cdot ||\mathcal{L}(\theta)||}\Big] \tag{2.17}$$

In addition, Hoang et al. [25] developed their Pareto Hypernetworks with Hypervolume Indicator (PHN-HVI) algorithm to learn the Pareto front using multi-sampled hypernetworks. The model trained is a hypernetwork representing multiple solutions for different preference rays. The model is further enhanced using hypervolume maximization to improve the quality of the solutions by making them more diverse and spread out.

MOO Method	**Number of Models**	**Pareto Front Region**	**Example**
Linear Scalarization	$\geq n_r$	Convex part	Weighted Sum
Gradient-Based	$= n_r$	All	Pareto MTL, EPO
Pareto Front Learning	1	All	COSMOS, PHN-HVI

Table 2.1: Comparative summary of multi-objective optimization methods

CHAPTER 3

Related Work

Going beyond fulfilling accuracy and one trustworthiness objective, prior work has focused on at most two objectives to examine how they fair against the accuracy of the model and other objectives. In this chapter, we review related work that studied the interplay (tension) between a pair of trustworthiness objectives.

In fact, developing a unified approach to achieve trustworthiness objectives described in Chapter 2 has thus-far proven to be challenging for researchers and practitioners. The complex interplay of different countermeasures and trustworthiness pitfalls produce unintended and often surprising consequences on the ML model. Hence, most research still focuses on each objective individually, and we have yet to witness considerable progress on training and deploying holistically trustworthy models in high-stakes applications.

Duddu et al. [13] conjectured that the unintended interactions among the objectives and their countermeasures boil down to two properties of ML models: overfitting and memorization. The authors study how different ML defenses affect diverse ML risks through their influence on generalization error (as a proxy for overfitting) and memorization of training data points. For example, adversarial training as a robustness countermeasure favors a model with a smoother decision boundary which increases the generalization gap and thus decreases accuracy [50, 13].

Additionally, Gittens et al. [20] argue for the use of causal models to learn the direct causal relationships between different features and the label in order to achieve trustworthiness. Causal ML permits reasoning on top of the true causal links as opposed to any spurious ones, making them ideal for modeling anti-discriminatory and robust behavior.

3.1 Privacy vs. Fairness

Numerous research has studied the impact of implementing differential privacy on the discriminatory behavior of a model. In effect, it has been empirically shown that the gradient noising process of DP increases the demographic disparity of the model. Bagdasarya et al. [3] concluded that underrepresented classes tend to lose more accuracy than the dominant class. The authors described this disproportionate impact on the minority group as "the poor becoming poorer." They conjecture that DP-SGD amplifies the model's bias towards the most represented group in the training data distribution. For instance, models trained with DP-SGD for gender and age classification of facial images exhibit larger performance discrepancy with higher accuracy on light-skinned faces over dark-skinned ones compared to a non-DP setting.

Tran et al. [61] formally proved why the noise addition and gradient clipping process of DP-SGD cause discriminatory behavior. The authors attributed unfairness in differentially private models to the average gradient norms of the demographic groups, the clipping bound and the Hessian loss. In fact, demographics with large input norms result in large gradient norms leading the privately trained model to disproportionately favor their loss over that of the underrepresented groups.

Yaghini et al. [71] developed different training algorithms to incorporate fairness constraints within differentially private learning. In particular, they propose two algorithms that combine fairness and privacy. In FairPATE, they combine demographic parity with Private Aggregation of Teacher Ensembles (PATE) [46] while in FairDP-SGD they combine demographic parity with DP-SGD. Their loss function included the DPFR defined in Equation (2.14) applied on a public dataset to limit demographic disparity without increasing the privacy budget. Their algorithm also implemented a post-processing mechanism to ensure γ-DemParity at inference time. They were able to guarantee a set demographic disparity budget by disregarding queries that would violate this guarantee at inference-time. As such, they introduced the query coverage of the model during testing as an additional metric to represent the accuracy-fairness trade-off: ensuring a higher fairness guarantee results in answering fewer queries. Their FairDP-SGD algorithm uses the following equation to compute the model's gradients for batch B:

$$\frac{1}{B}\sum_{x\in\mathcal{B}} clip_c\Big(\nabla_\theta\Big[\mathcal{L}\big(f_\theta(x), y\big) + \lambda DPFR(\theta_{public})\Big]\Big) + \frac{\sigma C}{B}\xi \tag{3.1}$$

By modifying the hyper-parameters of their algorithm (λ and σ), they are able to establish the Pareto front of utility, privacy and fairness by training different models representing different trade-offs.

3.2 Robustness vs. Privacy

While a theoretical trade-off hasn't been established yet, research has empirically found an inhibiting relationship between data privacy and robustness. One one hand, multiple research studying the effect of implementing differential privacy on adversarial robustness has found that DP models are more prone to adversarial examples than their non-private counterparts. Tursynbek et al. [66] experimentally showed that DP-trained models exhibit higher error on FGSM and PGD generated adversarial examples. Interestingly, this gap grows wider as the noise multiplier increases, i.e., models with a lower privacy budget tend to be more vulnerable to adversarial examples. They attributed this trade-off to the fact that the decision boundary of DP models possess a higher curvature leading to fragmented decision regions which makes it easier to find adversarial perturbations to fool the model. Boenisch et al. [5] corroborated the previous findings and expanded it to a broader range of attacks: they empirically showed that an increase in privacy reduces adversarial robustness among all attacks considered.

On the other hand, research has attempted to establish the impact of adversarially robust models on privacy leakage. Song et al. [58] assessed the success rate of Membership Inference Attack (MIA) on six different forms of adversarially trained models. They showed that adversarial training indeed makes the model more susceptible to MIA compared to a naturally trained model (between 2× to 4.5× more vulnerable on popular image datasets). The authors also demonstrated that the MIA success rate increases as the model becomes more adversarially robust. In addition, they proposed novel attacks for inferring membership that exploits the nature of training for robust models. The authors suggest the use of temperature scaling and regularization (like parameter norm penalties and dropout) as countermeasures to reduce the risk of MIA for robust models.

However, Hayes [23] argues for the lack of an inherent privacy-robustness trade-off: it is merely due to the degree of overfitting robust models exhibit rather than a fundamental limitation of both objectives. The author theoretically shows that robust models are capable of overfitting more or less than a standard model depending on the perturbation size and the size of the training dataset. As such, Hayes empirically shows that robust models are capable of being more secure against MIA as long as their generalization gap is smaller than standard models.

Hayes et al. [24] studied the difficulties for a model to satisfy differential privacy and be robust. They showed that the size of adversarial perturbation and the clipping norm of DP jointly increase the curvature of the loss landscape which hurts the generalization performance of the model. Phan et al. [49] developed their StoBatch algorithm to achieve

DP and robustness by implementing DP adversarial examples with ensemble and distributed training to achieve tight privacy bounds.

Wu et al. [70] proposed an approach to reconcile certified robustness with differential privacy. They wrap the TRADES loss function with differential privacy (by clipping and noising) and incorporate adversarial smoothing for certified robustness. Hence, their loss equation for a batch B becomes the following:

$$\frac{1}{B}\sum_{x\in\mathcal{B}} clip_c\Bigg(\nabla_\theta\Big[\mathcal{L}\big(f_\theta(x),y\big)+\beta\max_{||\alpha||_p\leq r}D_{KL}\big(f_\theta(x+\alpha),f_\theta(x)\big)\Big]\Bigg)+\frac{\sigma C}{B}\xi \tag{3.2}$$

3.3 Robustness vs. Fairness

Nanda et al. [41] defined the notion of robustness bias which ensures that all subgroups of a dataset are equally robust. In other words, different demographics shouldn't be more vulnerable to adversarial example attacks than others. A model satisfying low robustness bias exhibit a similar curvature in its decision boundary across different subgroups. The authors observe that the phenomenon of robustness bias is common across different model architectures and dataset, emphasizing the need for explicitly considering the robustness gap among groups during training.

Orthogonal to the notion of group fairness, multiple research has studied the discrepancy in accuracy and robustness of adversarially trained models with respect to the class labels (as opposed to separate sensitive attributes) [35, 27, 4]. For example, while an adversarially trained model might exhibit high overall accuracy and robustness, its performance per-class may vary wildly, causing some class labels to be more vulnerable and leading to a false sense of security. Surprisingly this holds true even if the data doesn't have any class imbalances and every label is equally represented. Though interesting, our work will be limited to the group notion of fairness as it relates to sensitive attributes rather than equal performance under adversarial settings across class labels.

3.4 Privacy vs. Interpretability

Before the advent of ML explanation methods, ML models such as DNNs have been mostly treated as black-boxes. End users are completely unaware of the 'reasoning' behind the model's prediction and must take them at face value. Considering that models are being used in high-stakes environments, ML developers face considerable pressure from both the public and governmental agencies to provide explanation frameworks on top of model predictions. However, this need for transparency within the decision-making process of the model seems to result in privacy leakage of the training samples. In fact, Shokri et al. [55] empirically showed that an adversary can take advantage of model explanations to infer information about the training data. The authors argue that backpropagation-based explanation methods expose statistical information regarding the decision boundaries of the input, which in turn enables an adversary to determine whether or not a data-point is a member of the model's training data. They also confirmed previous findings on privacy and fairness by showing that minority classes in the dataset are at higher risk of membership inference attacks via explanation methods. Patel et al. [47] proposed to reconcile the trade-off between data privacy and model transparency by developing an adaptive differentially private explanation algorithm.

In Table 3.1, we provide a summary of the multi-faceted interplay among countermeasures (rows) and trustworthiness objectives (columns). We denote with a red dot 🔴 the existence of research showing a theoretical or empirical trade-off between the two objectives and with a green dot 🟢 research arguing that the objectives don't directly conflict. As shown below, most research has found a conflicting relationship between implementing a countermeasure for one area of trustworthiness and its effect on other areas.

Countermeasure or Objective	Accuracy	Robustness	Data Privacy	Fairness
Adversarial Training	🔴[63] 🟢[72]	🟢[36]	🔴[58] 🟢[23]	🔴[41, 35, 27, 4]
Differential Privacy	🔴[59]	🔴[66, 5]	🟢[15]	🔴[3, 18, 71]
Fairness Constraints	🔴[45]	🔴 🟢[57]	🔴[7]	🟢[38, 45]

Table 3.1: Interplay of countermeasures and trustworthiness goals

CHAPTER 4

Approach

In Chapters 2 and 3, we established the complex interplay among the different ML trustworthiness objectives, i.e., most research has found a theoretical and/or empirical trade-off across different trustworthiness requirements. However, we also observe the following gaps that previous work on ML trustworthiness has either overlooked or under-explored:

- Most research in the area has studied the nature of the interaction between the trustworthiness objectives rather than aim to develop a unified framework to solve them jointly.
- Research has attempted to reconcile at most three objectives: utility, privacy and fairness in [71] and utility, privacy and robustness in [70].
- Research reconciling the different objectives has mostly relied on ERM regularization [71, 70, 73].
- Limited research has attempted to study the Pareto front with respect to the different loss functions and objectives (we are only aware of [71] which plotted the frontier of utility, privacy and fairness).

To fill the aforementioned gaps, we propose an approach to reconcile four trustworthiness objectives, namely utility, privacy, test-time robustness, and fairness. We do so in a simple, scalable, and efficient manner by restricting our approach to SGD-based algorithms (as opposed to ensemble methods) which scale at most linearly with the number of Pareto-optimal models needed. In addition, since our algorithms only rely on the choice of loss functions used, it provides a simple plug and play for practitioners wanting to incorporate less or more objectives by modifying the number and nature of loss functions utilized. In order to mitigate the challenges faced by plain ERM regularization[1], we leverage algorithms

[1] Hyperparameters set a priori, no systematic way to establish a diverse Pareto front, etc.

developed by the MOO literature to study their influence on the Pareto front generated. Our novel approach consists of reconciling trustworthiness objectives within the existing methods developed in the MOO-MTL literature.

4.1 Setup

In order to achieve and maintain trustworthiness throughout training, we propose to assign a loss function or regularization penalty for each objective. The loss functions made available for each dimension are listed below:

1. **Utility:** $\mathcal{L}_{Nat}$
   For the objective of natural accuracy, we use the conventional classification loss:, i.e., the cross-entropy (L_{CE}) with respect to the ground truth label defined in Section 2.1.

2. **Robustness:** $\mathcal{L}_{Rob}$
   In order for the trained model to remain robust against small perturbation of the input, we employ two main robustness mechanism for adversarial training. Let x' denote the adversarial version of x, we can either compute the KL divergence $D_{KL}\big(f(x'), f(x)\big)$ used in Equation (2.6) or the Madry loss, also known as the adversarial loss, $\mathcal{L}_{Adv} = L_{CE}\big(f(x'), y\big)$ used in Equation (2.4).

3. **Fairness:** $\mathcal{L}_{Fair}$
   Multiple losses can be used to mitigate unwanted bias against a demographic, namely the hyperbolic tangent relaxation of the DDP in Equation (2.12), its DEO version in Equation (2.13) and DPFR in Equation (2.14).

4. **Privacy:** ϵ
   As opposed to the other objectives, differential privacy will be wrapped on top of the final loss by clipping its gradient and adding random Gaussian noise. The noise multiplier σ and clipping norm C directly influence the total privacy budget ϵ.

The extensive list of losses above provide us with a proxy to measure the list of ML trustworthiness objectives outlined in Chapter 1. Additionally, our approach offers ML practitioners the freedom to pick any combination of loss functions more suitable for their business needs, and domain experts can leverage their subject matter expertise to inform the choice. We further insist on the dynamic nature of the above list: emerging research often develops novel loss variants to satisfy various requirements. Our approach can be customized with specific implementations of loss functions. At a high-level, we will provide two frameworks for reconciling the trustworthiness objectives above: ERM regularization and MOO algorithms, which we describe next.

4.2 ERM Regularization

A natural first-cut approach to combine the different loss functions is to produce a regularized variant of the ERM algorithm defined in Equation (2.1) to incorporate robustness and fairness metrics. The loss function will be regularized as follows:

$$L(\theta) = \mathcal{L}_{Nat}(\theta) + \lambda_{Rob}\mathcal{L}_{Rob}(\theta) + \lambda_{Fair}\mathcal{L}_{Fair}(\theta) \tag{4.1}$$

While modifying the hyperparameter values allows great freedom on the degree of regularization for each objective, it should be done carefully to avoid breaking the DP budget. For example, if the adversarial loss is used for robustness, calculating the gradient of both the benign and perturbed input might result in privacy leakage of the training sample. Cohen et al. [70] remedy this problem by averaging the losses first *before* clipping the gradient, bounding the sensitivity of the sample thus preserving differential privacy[2]. Therefore, if we need to incorporate the Madry loss with the natural loss in Equation (4.1) we would have to compute the average of $\mathcal{L}_{Nat}(\theta)$ and $\mathcal{L}_{Adv}(\theta)$ before clipping the gradients in order to preserve differential privacy. As such, ML practitioners have to ensure that their choice of hyperparameters doesn't accidentally break the DP guarantee.

In particular, we propose combining the loss functions used in Equation (3.1) and Equation (3.2) to satisfy utility, robustness, fairness, and privacy. The loss makes use of the KL divergence as a robustness metric and DPFR as a fairness regularizer[3]:

$$L(x, y) = L_{CE}(f_\theta(x), y) + \beta D_{KL}(f_\theta(x'), f_\theta(x)) + \lambda \text{DPFR}(\theta) \tag{4.2}$$

In addition, we compute the gradients of the above loss function in a differentially private manner to obtain our novel Differentially Private - Robust and Fair Learning (DP-RFL) algorithm. Its pseudo-code is provided in Program 4.1.

[2] Their approach draws similarity with the data augmentation in private learning developed in [11] and defined in Equation (2.9)

[3] We renamed the hyperparameters for consistency with their use in their original paper

```
**Require:** Model parameters $\theta$, loss function $\ell$, learning rate $\eta_t$, noise multiplier $\sigma$, group
    size $B$, clipping norm $C$, fairness regularizer $\lambda$
for $t \in [T]$ do
    Sample mini-batch $B_t$ with sampling probability $B/N$
    for $i \in B_t$ do
        Compute perturbed input $x'_i$
        Compute $g_t(x_i) \leftarrow \nabla_\theta \Big( L_{CE}\big(f_\theta(x_i), y_i\big) + \beta D_{KL}\big(f_\theta(x'_i), f_\theta(x_i)\big) + \lambda \text{DPFR}(\theta) \Big)$
    end for
    $\tilde{g}_t(x_i) \leftarrow g_t(x_i)/max(1, \frac{||g_t(x_i)||_2}{C})$
    $\tilde{g}_t \leftarrow \frac{1}{|L_t|}(\sum_i \tilde{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$
    $\theta \leftarrow \theta - \eta_t \tilde{g}_t$
    $\epsilon \leftarrow \text{PrivacyAccountant}(\theta, \sigma, C)$
end for=0
```

Program 4.1: DP-RFL: Differentially Private-Robust and Fair Learning

The hyperparameters β and λ in Equation (4.2) as well as the noise multiplier σ for DP-SGD provide the ML practitioner with control over the scale associated with each trustworthiness objective during training. Its disadvantage is that the relationship between the hyperparameters (β, λ) and the true objectives (robustness and fairness) aren't one-to-one, i.e., increasing one hyperparameter value doesn't necessarily translate into prioritizing the corresponding objective. For example, increasing the value of β too much will generate weak decision boundaries, rendering the model useless and thus not robust. In addition, the hyperparameters need to be set a priori and don't allow dynamic shifting of priorities during training. In turn, this forces practitioners to decide on the hyperparameter values which often proves to be a difficult task even for experienced professionals. The choice of hyperparameters can be subjective and require domain expertise or sensitivity analysis to determine how their values appropriately reflect subjective preference per objective. Additionally, hyperparameter tuning doesn't provide us with a feasible alternative as it is both time consuming and resource intensive.

4.3 Multi-Objective Optimization

Considering the established multi-faceted trade-off imposed on the different objectives, it is natural to frame our goal of producing trustworthy ML as a multi-objective optimization problem. The challenge is that most MOO frameworks for training a DNN have been conventionally developed for multi-task learning which entails a model architecture with separate shared and task-specific parameters [52]. As an example of MTL, models have been developed to simultaneously predict two partially overlapping digits in the muti-MNIST dataset, requiring a model with two separate final layer for each task (predicting left and right digits). As such, the model has to optimize the shared parameters for both tasks and fine-tune the task-specific parameters for their corresponding task.

In contrast to MTL, the quest for trustworthy ML inherently entails a single task, i.e., predicting the class label, with the additional requirement of simultaneously satisfying multiple trustworthiness objectives. In this context, the model cannot employ task-specific parameters per objective but rather rely on all of its parameters to jointly satisfy the objectives. Limited effort has been dedicated to applying MOO for satisfying different objectives within a single task. Although research has studied the application of MOO on accuracy and fairness [44, 51], we aren't aware of any that incorporated robustness. This challenge raises the question: **can the MTL-MOO frameworks be efficiently adopted for ML trustworthiness?** In what follows, we provide details of our approach for applying the MOO frameworks discussed in Section 2.5 for the problem of trustworthy ML.

Before discussing the technicalities of the different MOO algorithms, we have to choose a vector of the losses to be minimized in accordance with Equation (2.15). As such, we have the freedom of selecting different combinations of loss functions mentioned in Section 4.1. We provide a generic three-dimensional vector of the per-objective losses to capture utility, robustness, and fairness[4] below:

$$\mathcal{L} = \begin{pmatrix} \mathcal{L}_{nat} \\ \mathcal{L}_{rob} \\ \mathcal{L}_{fair} \end{pmatrix} \tag{4.3}$$

Regardless of the specific loss function utilized above, we need to take into consideration ϵ in order to include the privacy budget within the final list of objectives to be minimized.

$$\mathcal{O} = \begin{pmatrix} \mathcal{L}_{nat} \\ \mathcal{L}_{rob} \\ \mathcal{L}_{fair} \\ \epsilon \end{pmatrix} \tag{4.4}$$

Keeping with our loss selection in Equation (4.2), we have the option of utilizing the following

[4] Data privacy being satisfied as a wrapper around gradient calculation

loss vector.

$$\mathcal{L}_{Nat\_KL\_DP} = \begin{pmatrix} \mathcal{L}_{Nat}\big(f_\theta(x), y\big) \\ D_{KL}\big(f_\theta(x'), f_\theta(x)\big) \\ \mathrm{DPFR}(\theta) \end{pmatrix} \tag{4.5}$$

However, we don't limit ourselves with the loss vector above nor its fixed dimensionality and further propose novel loss combinations to capture ML trustworthiness in the context of MOO. This approach will allow us to evaluate different loss vectors implemented across trustworthiness metrics and MOO methods.

For example, we can start by replacing the second elemend of Vector (4.5) by the Madry loss to obtain the following vector:

$$\mathcal{L}_{Nat\_Adv\_DP} = \begin{pmatrix} \mathcal{L}_{Nat}\big(f_\theta(x), y\big) \\ \mathcal{L}_{Adv}\big(f_\theta(x'), y\big) \\ \mathrm{DPFR}(\theta) \end{pmatrix} \tag{4.6}$$

Unlike ERM regularization where we have to make sure that the hyperparameters don't break the differential privacy guarantee, MOO methods compute a weighted average of the loss vector *before* the gradient calculation of backpropagation (and thus before clipping). As such, as long as each sample is equally represented, the final surrogate loss L is computed as their weighted average which limits the sensitivity per-batch, providing us with a guarantee that L doesn't incur more privacy leakage compared to only considering the natural loss in plain DP-SGD. For instance, when using both $\mathcal{L}_{Nat}$ and $\mathcal{L}_{Adv}$, the surrogate loss will be their weighted average[5] which doesn't incur additional privacy budget consumption compared to only considering the natural loss. As for the fairness loss, we limit its calculation on a public dataset in accordance with previous work [71].

Furthermore, we are able to remove the natural loss of the loss vector (4.6) to produce $\mathcal{L}_{Adv\_DP}$ without reducing the number of trustworthiness objectives considered. In fact, Zhang et al. [73] decomposed the robust error as the sum of the natural error and the boundary error:

$$\mathcal{R}_{rob}(f) = \mathcal{R}_{nat}(f) + \mathcal{R}_{bdy}(f) \tag{4.7}$$

By taking the Madry loss and the KL divergence as a proxy for the robust and boundary errors respectively, we are able to reduce the first two elements of Equation (4.5) as the Madry loss. In other words, the classification loss on adversarial data can capture both the natural error on the data (utility) and sensitivity to adversarial perturbation (smoothness),

[5] The weights are positive and sum up to 1

effectively reducing the dimensionality of Equation (4.5) to two dimensions.

$$\mathcal{L}_{Adv\_DP} = \begin{pmatrix} \mathcal{L}_{Adv}\big(f_\theta(x'), y\big) \\ \text{DPFR}(\theta) \end{pmatrix} \tag{4.8}$$

Similarly, we introduce a demographically partitioned natural loss by dividing the batch by the sensitive attribute before computing the per-group natural loss.

$$\mathcal{L}_{DPL} = \begin{pmatrix} \mathcal{L}^1_{nat} \\ \ldots \\ \mathcal{L}^s_{nat} \end{pmatrix} \tag{4.9}$$

While the above definition provides us with a simple way to contrast the performance of the model across different demographics, it assumes that the selected batch provides us with fairness metrics representative of the population. It also doesn't exhibit a natural trade-off: while unfair models tend to perform better for certain demographics, this doesn't necessarily translate into a conflicting relationship among the natural loss across the demographics (a model performing better for one group doesn't have to perform worse on another). We will study the effectiveness of this loss vector in the next section. In addition, we append either the KL divergence or the Madry loss to the above vector to represent the robustness objective and obtain $\mathcal{L}_{KL\_DPL}$ or $\mathcal{L}_{Adv\_DPL}$, respectively.

While different weights might be assigned to the losses in Equation (4.3) before or during training, the privacy budget ϵ doesn't offer the luxury of being manipulated dynamically. Instead, ϵ is fully determined by the noise multiplier σ and the number of epochs. Since ϵ gradually increases at each epoch, we propose running the model on a large number of epochs while saving its parameters and evaluating its objectives in Equation (4.4) at the end of each epoch. This approach allows us to choose a lower privacy budget ϵ by reverting to a model generated at an earlier epoch instead of retraining the model from scratch with a new value of σ.

We will discuss the methodology we follow to apply the MOO frameworks defined in Chapter 2 to achieve the desiderata we outlined previously in Chapter 1. For the rest of this section, we will refer to the number of preference rays desired as n_r.

4.3.1 Linear Scalarization

We generate n_r preference rays R either uniformly across a 2-dimensional circle or randomly for higher dimensions. We assign a total of n_r models for each ray and train them for e epochs. The surrogate loss of each model m_i corresponds to a weighted sum of the loss vector $\mathcal{L}$ with the model's respective ray r_i. In total, we obtain $n_r \cdot e$ model checkpoints across the different preference vectors and training epochs.

$$L(\theta) = R \cdot \mathcal{L}(\theta) \tag{4.10}$$

$$L(\theta) = r_1\mathcal{L}_{Nat}(\theta) + r_2\mathcal{L}_{Rob}(\theta) + r_3\mathcal{L}_{Fair}(\theta) \tag{4.11}$$

We note that the above formula is similar to the one in Equation (4.2) except for the fact that the rays r_i sum up to 1 and are positive. This requirement provides us with a systematic way to generate models at a particular region of the Pareto front without considering any potential privacy leakage: we are able to scale the weights r_i within the range $(0, 1)$ rather than deal with hyperparameters in the range $(0, \infty)$. Moreover, instead of setting the hyperparameters a priori and train one model, we train separate models for each preference vector, allowing us to obtain a representative set of Pareto optimal models.

4.3.2 Gradient-based Methods

While gradient-based approaches for multi-gradient descent have been conventionally used for MTL, we propose applying the algorithm for the problem of generating holistically trustworthy ML models. However, since the weights W are generated dynamically according to the per-loss gradients, naively applying them for our problem will violate the DP guarantee. Accordingly, we make use of the post-processing property of DP and perform our computation on top of the private gradients instead of the original ones. Since the gradients are clipped and noised, the sensitivity of the sample is bounded as required for the sampled gaussian mechanism of DP-SGD [40] and differential privacy remains preserved. Let $F(.)$ be the gradient based approach used (e.g., Pareto MTL, EPO), we compute the dynamic weights w_i at every iteration from the per-objective private gradients as follows.

$$\nabla_\theta \tilde{\mathcal{L}} = \begin{pmatrix} clip_c(\nabla_\theta \mathcal{L}_{nat}) + \frac{\sigma C}{B}\xi_1 \\ clip_c(\nabla_\theta \mathcal{L}_{rob}) + \frac{\sigma C}{B}\xi_2 \\ clip_c(\nabla_\theta \mathcal{L}_{fair}) + \frac{\sigma C}{B}\xi_3 \end{pmatrix}$$

$$W = F(\nabla_\theta \tilde{\mathcal{L}})$$

$$L(\theta) = W \cdot \mathcal{L}(\theta) = w_1\mathcal{L}_{nat}(\theta) + w_2\mathcal{L}_{rob}(\theta) + w_3\mathcal{L}_{fair}(\theta)$$

4.3.3 Pareto Front Learning

For PFL we require a model with input space $\mathcal{X} \times \mathbb{R}_+^m$ which we achieve by upsampling the original data to contain multiple channels for each weight. As opposed to other methods we train a single model for e epochs, resulting in e model checkpoints. The COSMOS method relies on weights sampled randomly from a Dirichlet distribution with parameter α and a cosine similarity term to decrease the angle with the preference ray. The loss function is defined below

$$L(\theta) = r \cdot \mathcal{L}(\theta) - \lambda \frac{r \cdot \mathcal{L}(\theta)}{||r|| \cdot ||\mathcal{L}(\theta)||} \qquad \text{with } r \sim \mathrm{Dir}\,(\alpha) \tag{4.12}$$

Since ML trustworthiness doesn't have per-task parameters, Pareto front learning methods will struggle to learn multiple trustworthiness trade-offs at once. Hence, we train multiple models each with different hyperparameters α and λ to provide us with a higher quality Pareto front[6].

[6] For our experiments, we only trained two models using COSMOS which is significantly less than the $n_r = 10$ models trained for the other methods

CHAPTER 5

Evaluation

Our evaluation is guided by the following research questions:

- **RQ1**: Given that research is often conflicting regarding the existence of a direct conflicting relationship between two objectives, e.g., whether the accuracy-robustness or robustness-privacy trade-off is inherent (as shown in Table 3.1), to what extent can we empirically study the nature of the multi-faceted relationship across trustworthiness objectives?
- **RQ2**: If trustworthiness objectives appear to be mutually conflicting, how pronounced is this trade-off when comparing any subset of our original objectives?
- **RQ3**: Can Multi-Objective Optimization frameworks developed for Multi-Task Learning be efficiently adopted for ML trustworthiness? If so, how do the different algorithms fair against each other in achieving holistically trustworthy models?
- **RQ4**: How does the combination of different loss functions impact the final Pareto front generated?
- **RQ5**: What is the best training algorithm applied to machine learning trustworthiness that produces a diverse set of optimal solutions?

5.1 Datasets

We will conduct our experiments on the Colored MNIST dataset. The dataset has been developed by Arjovsky et al. [2] for their invariant risk minimization algorithm. It consists of the original MNIST images expanded into two color channels: red and green[1]. The image pixels are present in one of the channels while the other has its value set to zero. The

[1] Unlike the original paper, we stay consistent with the digit label (no label flipping)

training, validation and testing dataset have different proportions of red and green images representing different environments, e.g., data acquired from different sources, data drift, etc. The dataset contains 50,000 images for training and 10,000 for validation and testing each. We further consider 512 images from the training split as public data to be used for calculating the fairness loss.

We present the configuration of our experiments below, including the dataset and model architecture.

Dataset	Class Label	K	Sensitive Attribute	S	Architecture	# Parameters	σ
ColorMNIST [2]	Digit	10	Color	2	CNN (Table 5.2)	431,580	0.8

Table 5.1: Datasets and architecture used

5.2 Metrics

In addition to the value of the losses introduced previously, we also make use of the metrics below:

1. **Validation accuracy**: Utility is evaluated as the benign accuracy of our model on a validation dataset $A_{nat} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(f_\theta(x_i) = y_i)$.

2. **Adversarial validation accuracy**: Robustness is measured by the accuracy on adversarial data from a validation dataset $A_{rob} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(f_\theta(x'_i) = y_i)$ where x'_i is the adversarial perturbation of x_i (approximated with PGD), i.e., $x'_i = \underset{x_{adv} \in \mathbb{B}(x,\epsilon)}{arg\,max}\ \mathcal{L}\big(f_\theta(x_{adv}), y\big)$.

3. **Local Lipschitzness**: We estimate the local Lipschitzness of the model using the empirical formula derived in [29] as a proxy for the sensitivity of the model for small perturbations.

4. **Demographic Parity Loss**: We compute the average DPFR per batch on a validation dataset. DPFR corresponds to the maximum demographic disparity across all predicted labels and sensitive features, and it is used as a proxy for evaluating the degree of fairness of the model $DPFR(\theta) = \underset{s,\hat{y}}{max}\ \widehat{\Gamma}(s, \hat{y})$.

5. **Privacy budget**: We compute the privacy budget ϵ spent while training for a fixed $\delta \ll 1/n$.

6. **Hypervolume**: In order to evaluate the different pareto fronts generated by an MOO method, we compute the hypervolume of the Pareto front with the point (3,3,3) as reference for the three dimensional objectives of natural, adversarial and fairness losses respectively.

5.3 Experimental Setup

We use a simple LeNet model architecture as shown in Table 5.2 for the Colored MNIST dataset adapted from [71]. We also note that when applying the COSMOS approach [51], we have to upsample the image input to add the preference ray as a feature for the model to learn. This is done through transposed convolutions which increase the number of input channels in the first convolutional layer of the model by the dimensionality of the objective space (i.e., the number of elements in the preference ray).

Layer	**Dimensions**
Conv2D	2 × 20 × 5 × 1
Relu	_
MaxPooling	2 × 1
Conv2D	20 × 50 × 5 × 1
Relu	_
MaxPooling	2 × 1
Flatten	_
Fully Connected	800 × 500
Relu	_
Fully Connected	500 × 10
Sigmoid	_

Table 5.2: ColorMNIST CNN Architecture

As for the hyper-parameters of the experiment, we train the model with an initial learning rate of 0.006 for 20 epochs and a batch size of 128. We further make use of an exponential learning rate decay of 0.95 at every third epoch. As for the DP-SGD algorithm parameters, we use a clipping norm of 1.2 and the noise multiplier σ is calculated before training to ensure a privacy budget ϵ of 2 for the last epoch[2]. For adversarial training, we perform 10-iteration PGD perturbation with a step size of 0.01 and an adversarial budget of 0.1 on the ℓ_∞ norm. For the MOO framework, we employ 10 preference rays to approximate the Pareto front of an approach. To remain fair across the comparison, we also train 10 different models with the ERM regularization approach with hyperparameter values ranging from 0.5

[2] Its value is approximately 0.79

to 5 for both β and λ. Furthermore, when we implement the COSMOS algorithm, we make use of different values for α and λ for the Dirichlet distribution and cosine term respectively. Specifically we use $\alpha = [1.7, 1.2, 0.5]$ with $\lambda = 0.9$ and $\alpha = [0.8, 0.8, 0.8]$ with $\lambda = 0.5$ when training for the $\mathcal{L}_{Nat\_KL\_DP}$ loss vector.

5.4 Results

We will first evaluate the two-dimensional Pareto front generated through private training. Afterwards, we will consider the three-dimensional Pareto front which will allow us to evaluate the different loss vectors proposed in Chapter 4. Finally, we display the Pareto front generated in the four dimensional objective space and compare the methods we previously proposed across different metrics.

5.4.1 Influence of Private Training on Trustworthiness

We will start with evaluating the influence of differentially private training (through DP-SGD) on different trustworthiness domains. Since we fixed the noise multiplier and gradient norm on all the experiments, the privacy budget ϵ can be regarded as a proxy for the training epoch where the user can achieve a higher privacy guarantee by stopping at an earlier epoch. We will contrast how the different methods proposed in Chapter 4 fair against each other: how optimal is the Pareto front generated by adopting ERM regularization or one of the MOO algorithms? In order to fairly compare the methods in this section, we use the same combination of loss functions, namely natural loss, KL divergence and DPFR, except for the Pareto MTL approach which only uses the Madry loss and DPFR as the algorithm only works for two dimensions.

Our experiments displayed in Figure 5.1 back up our previous motivation to expand our algorithm beyond ERM regularization as it performs poorly compared to the other MOO algorithms (**RQ3**). In particular, Linear Scalarization and EPO are able to achieve a more optimal front closer to the origin and provide a diverse set of Pareto optimal solutions. Furthermore, Figure 5.2 shows that the results hold when considering the trade-off across different pairs of objectives. When plotting the natural loss against the KL divergence, the ERM approach performs better than the others at the early stages of training. However, it fails to converge properly by the end of training, achieving a minimal natural loss of around 1.6 as opposed to the loss of 1.55 achieved by other approaches. We conjecture that this performance is due to the fact that ERM regularization doesn't provide us with a systemic approach to evaluate different regions in the Pareto front: the hyperparameters have a wide

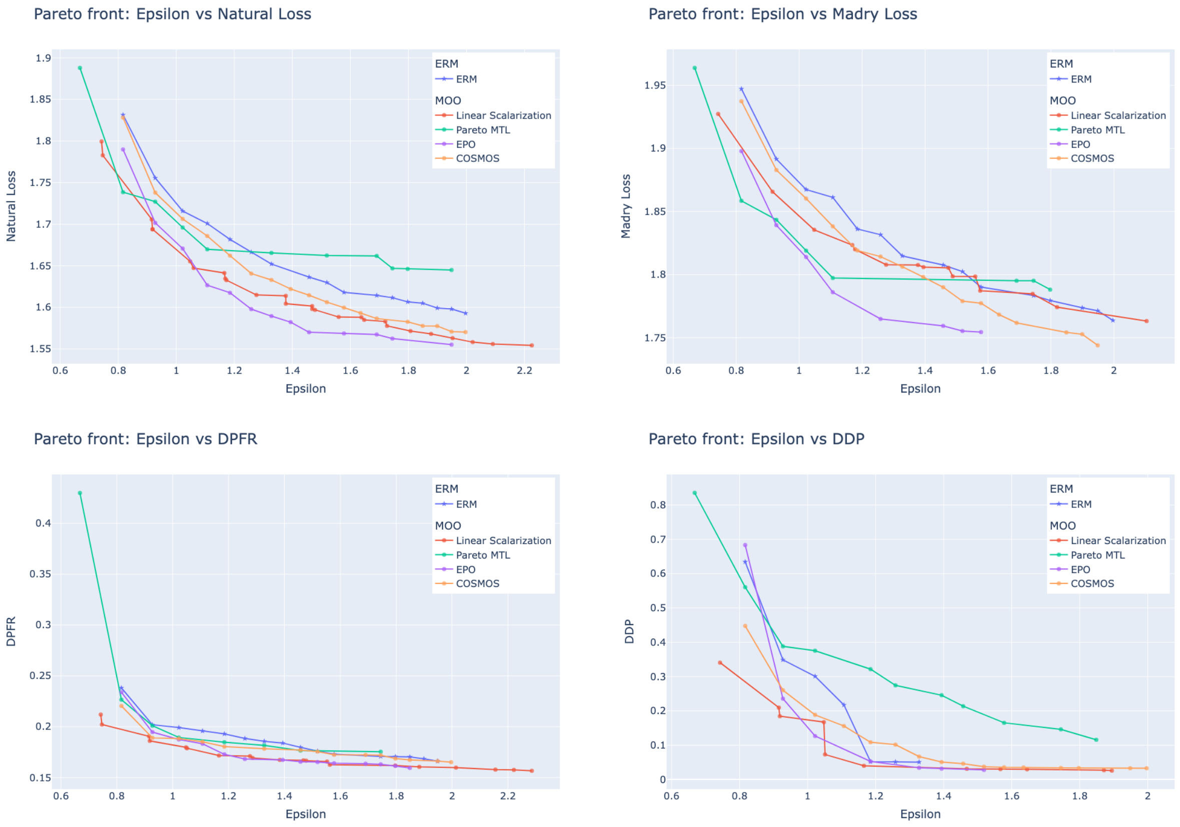

Figure 5.1: Privacy-X Pareto front

range of possible values with no proper way to set them such that the model converges towards a specific point on the Pareto front. MOO frameworks on the other hand provide us with a systemic approach to train different models on the regions of the Pareto front: the weights are set in a manner to optimize convergence towards a given preference ray.

Takeaways: Our findings emphasize the need to apply MOO algorithms in the context of holistic ML trustworthiness as opposed to plain ERM regularization. Since MOO frameworks are developed with the assumption of conflicting objectives, it makes sense for their performance to be superior over ERM.

5.4.2 Loss Vector Evaluation

In order to evaluate the losses properly, we fix the MOO training framework to be the Linear Scalarization algorithm before diving into the comparison of the other approaches. We move to comparing the performance of the model when it is trained according to the different loss vectors defined previously in Chapter 4. Note that while the specific loss functions

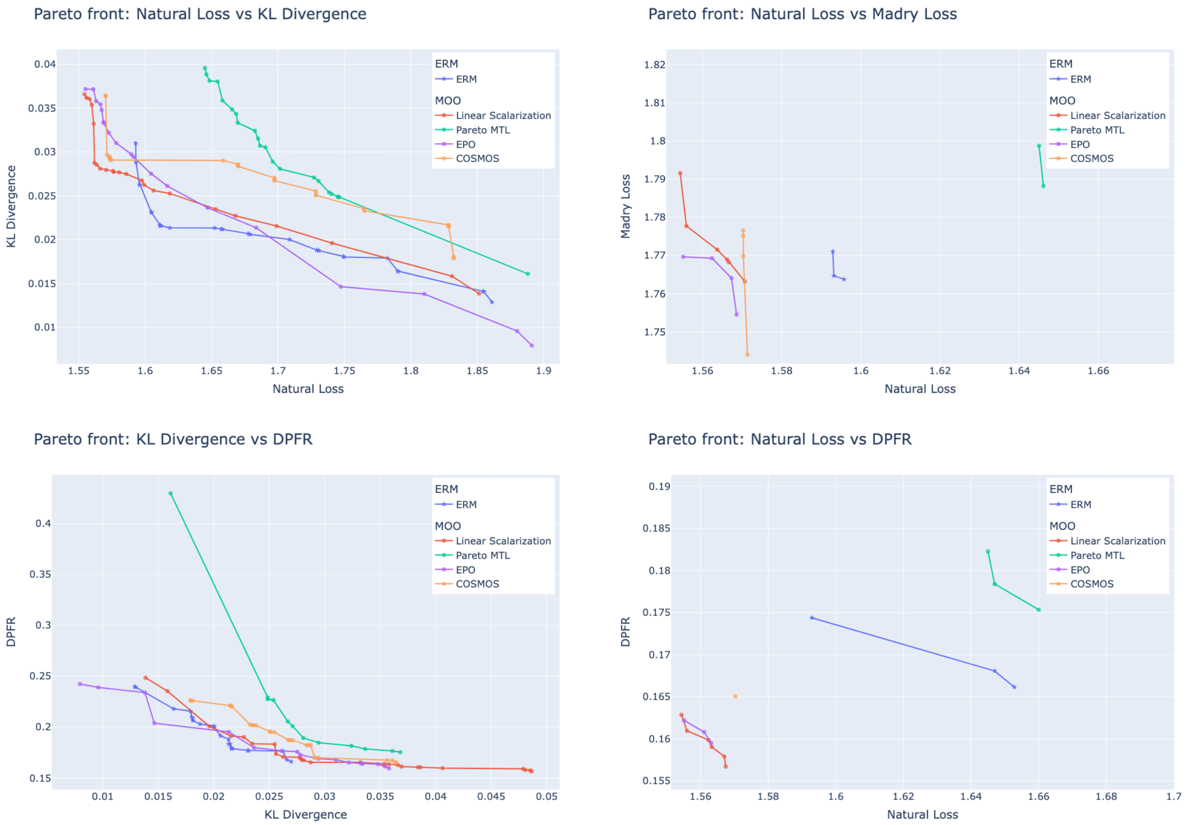

Figure 5.2: Two-Objective Pareto front under private training

used during training differ, the model is evaluated on the same list of metrics[3] (Section 5.2) at the end of each epoch. This allows us to compare different models across an extensive list of metrics even if they weren't explicitly part of the training objectives. For example, a robust model trained on the $\mathcal{L}_{Nat\_KL\_DP}$ vector will be evaluated on the Madry loss too in order to comprehensively assess its level of robustness. In addition, while some of the models are trained on three loss functions, we will plot a two-dimensional scatter plot, which effectively projects the Pareto front onto two dimensions. This projection doesn't guarantee that the Pareto front has a smooth curvature leading us to only select the Pareto optimal models for the two dimensions considered. However, our experiments show that the projected Pareto front remains in a“good” shape permitting a useful analysis. We will plot the three-dimensional Pareto front later in the section. Some of the models considered are only trained on three objectives rather than four[4], we include them in our figures in order to compare them to the other models incorporating all trustworthiness domains.

[3] the metrics also include all the loss functions mentioned previously

[4] utility, privacy and robustness or utility, privacy and fairness

Utility-Robustness-Privacy: We will first study the utility-robustness-privacy trade-off achieved by the different loss functions. As shown in Figure 5.4, the model that isn't explicitly trained for robustness shows an increase in its sensitivity across its decision boundary as the natural loss and privacy budget increase. However, when a robust loss is included within the training algorithm, the model learns to remain insensitive to small perturbations. Note that in the early epochs, the model has a low KL divergence because it hasn't yet converged to a useful model so its decision boundaries are still very smooth which isn't a reflection of its true robustness. At every further epoch, the KL divergence remains stable across the different models in the same region of the Pareto front.

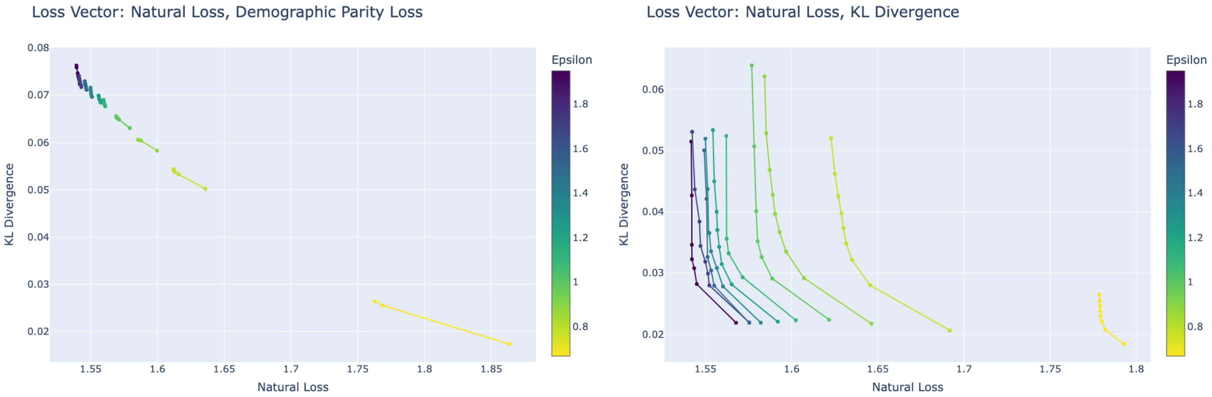

Figure 5.3: Utility-Robustness-Privacy Pareto front for a 3-objective trained model

In figure 5.4, we evaluate different losses used to capture the four trustworthiness objectives on the Utility-Robustness-Privacy trade-off. On the y-axis, we display the KL divergence in the top row and the Madry loss in the middle row. Note how the Pareto front gradually moves closer to the origin (more accurate and robust) as we train the model for more epochs (less private). Our result also indicates that using either the KL divergence or the Madry loss represent good proxies for robustness: a model trained with $\mathcal{L}_{Nat\_KL\_DP}$ still produces a good Pareto front when evaluated on the Madry loss and vice versa with $\mathcal{L}_{Nat\_Adv\_DP}$. We also note how the below graphs compare to the right plot in Figure 5.3 that is only trained on $\mathcal{L}_{Nat\_KL}$, the range of KL divergence spanned slightly increases from a minimum of 0.02 to 0.025, indicating the challenge of becoming more robust when additional fairness constraints are imposed on the training process. The bottom row displays the respective Pareto fronts when the model is trained using $\mathcal{L}_{Adv\_DP}$. While theoretically this loss should capture utility and robustness within its $\mathcal{L}_{Adv}$ component, the model struggles to establish a properly diverse Pareto front, especially when plotting the natural loss against its adversarial version (bottom right plot in Figure 5.4). This highlights the need to explicitly

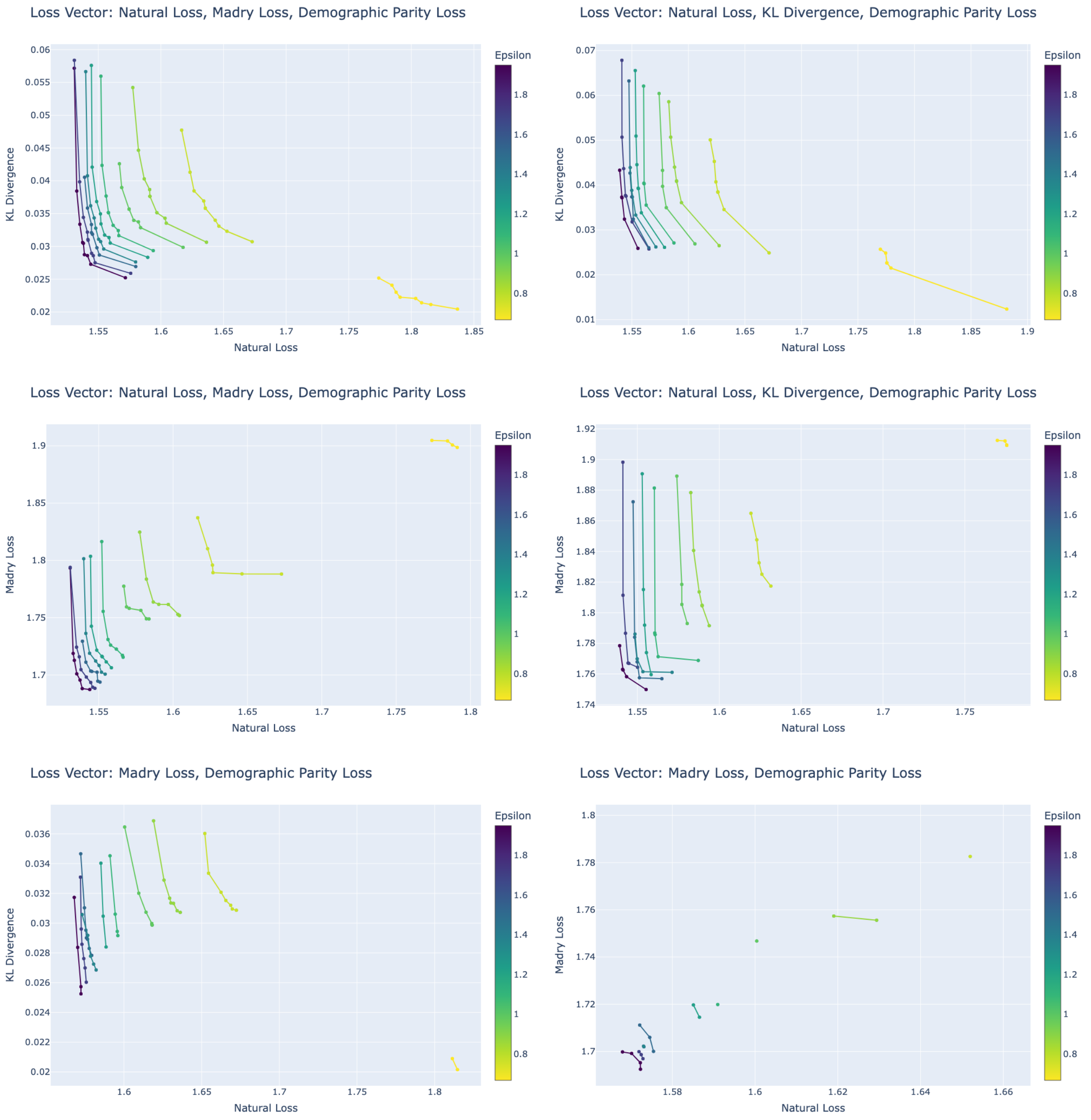

Figure 5.4: Utility-Robustness-Privacy Pareto front for a 4-objective trained model

consider the natural loss within the training objectives as opposed to combine it with its robust form.

Utility-Fairness-Privacy: We will start with plotting the Pareto front of a private model trained solely on accuracy and fairness (without robustness) and contrast its performance when robustness is added. As shown in figure 5.5, more Pareto optimal models (with respect to the natural loss and demographic disparity) are generated when the loss only captures accuracy and fairness: increasing the dimensionality of the objectives reduces the quality of the Pareto front projected on the two dimensions of utility and fairness. However, the optimal models generated remain consistent with their achieved range: DDP value around 0.3 (except for the $\mathcal{L}_{Adv\_DP}$-trained model which seem to exhibit a far wider range).

It is also interesting to note the difference in the Pareto fronts generated for Utility-Robustness-Privacy in Figure 5.4 versus Utility-Fairness-Privacy in Figure 5.5: the former seems more pronounced and diverse than the latter (**RQ2**). Our experiments show that accuracy and robustness exhibit a higher degree of "conflict" than accuracy and fairness. Intuitively, this makes sense particularly for the Colored MNIST dataset: we expect an accurate model to completely disregard the color of the image and only rely on other features for digit prediction. The same can't be argued for utility and robustness as multiple research has shown an empirical and theoretical conflict for a model to be both accurate and resilient to adversarial attacks [63].

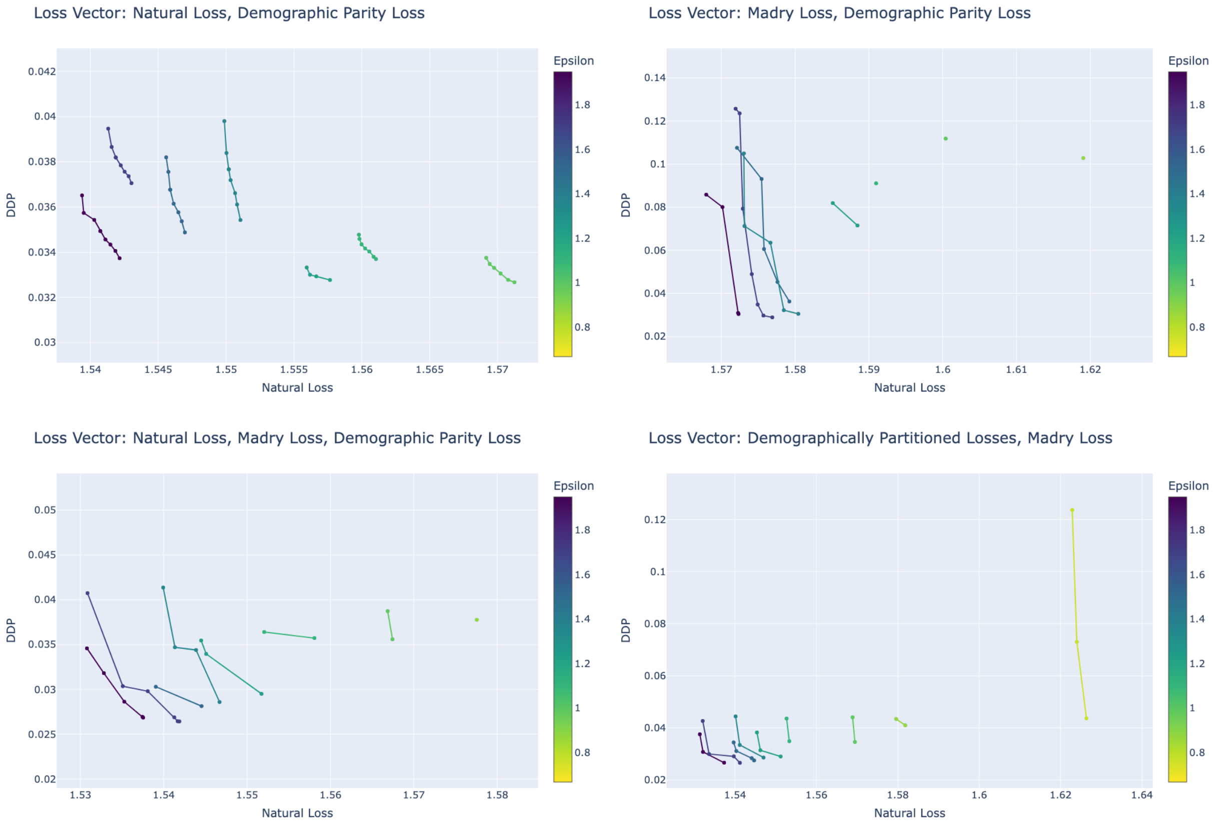

Figure 5.5: Utility-Fairness-Privacy Pareto front

Robustness-Fairness-Privacy: Keeping with our prior analysis, we will briefly evaluate the Robustness-Fairness-Privacy Pareto front. The experiments confirm previous findings on the inherent trade-off between robustness and fairness: robust models seem to exhibit performance discrepancy across different demographics and labels [41, 35, 27]. Our results further demonstrate the potential of models to converge closer to the origin (more robust and fair) at the expense of privacy. Unlike the Utility-Fairness-Privacy Pareto front in Figure 5.5, our experiments show a diverse set of Pareto-optimal models for the Robustness-Fairness-Privacy front. Robustness and fairness seem to exhibit a more pronounced trade-off than accuracy and fairness. Hence, our work confirms in the context of private learning what multiple research has shown for non-private training: adversarial training has the unintentional effect of worsening the model's performance with respect to fairness measures (**RQ2**) [27, 57].

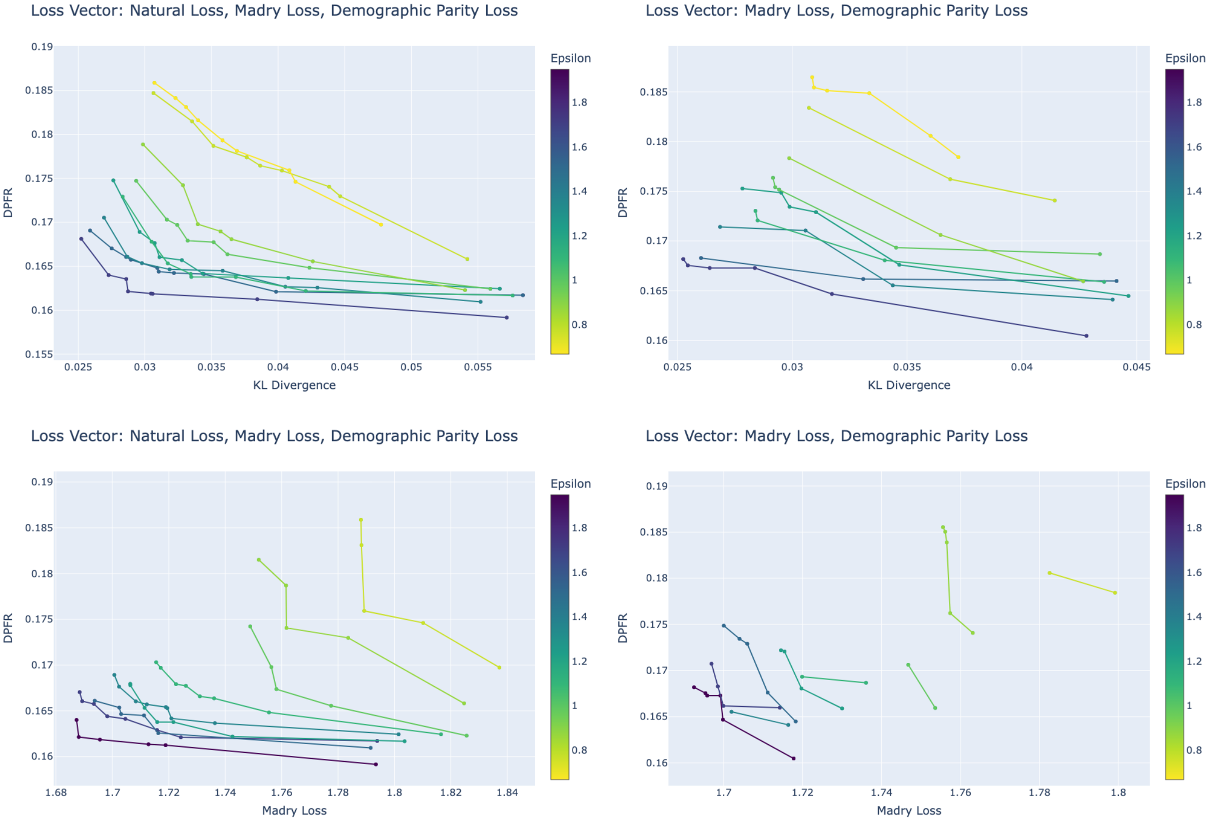

Figure 5.6: Robustness-Fairness-Privacy Pareto front

Takeaways: The above experiments allow us to empirically compare different combinations of loss functions for developing holistically trustworthy ML (**RQ4**). Accordingly, we favor the use of $\mathcal{L}_{Nat\_Adv\_DP}$ or $\mathcal{L}_{Nat\_KL\_DP}$ loss vector as it generated a wide and diverse Pareto optimal set across different metrics. Even though the $\mathcal{L}_{Adv\_DP}$ appealed to us as it reduced the dimensionality of the objective space while representing all our trustworthiness goals, this reduction comes at the cost of generating a well-shaped Pareto front particularly for the utility objective.

5.4.3 Four-Objective Pareto Front

We will now evaluate the Pareto front generated for all four trustworthiness dimensions: utility, fairness, robustness and privacy. To properly visualize it, we plot the natural loss, adversarial loss and DPFR in the x, y and z axis respectively. The privacy budget epsilon will correspond to the color of the scatter points. We also plot a cone pointing towards the origin to provide the relative direction of the diferent axis. We provide two views of the same scatter plot in Figure 5.7. The figure shows us the set of Pareto optimal models generated by all our methods with respect to the natural loss, adversarial loss, demographic disparity and differential privacy. As can be seen, the models converge to lower values of the former three losses at the expense of privacy leakage (**RQ1**). Interestingly, while the different methods start with a wide ranging set of initial models, they all converge in a similar direction and aggregate close to each other in the direction of the origin. We further divide this Pareto front by method in Figure 5.8 which highlights how MOO algorithms (Linear Scalarization, EPO and COSMOS) achieve a wider Pareto frontier closer to the origin than ERM and Pareto MTL.

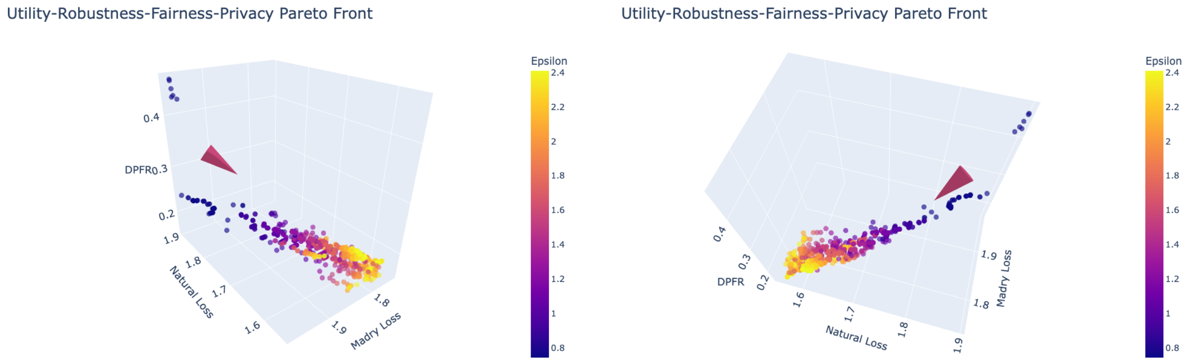

Figure 5.7: Utility-Robustness-Fairness-Privacy Pareto front - alternative view of the same plot

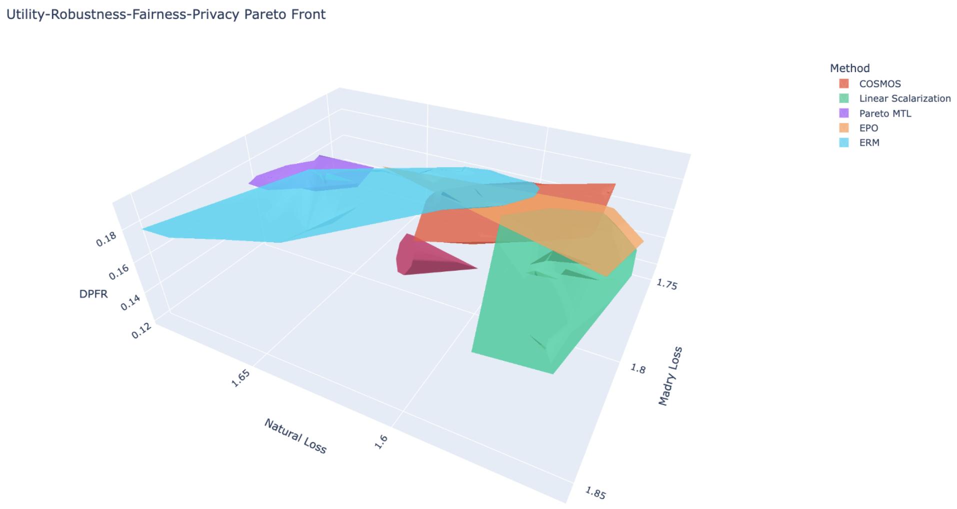

Figure 5.8: Utility-Robustness-Fairness-Privacy Pareto front per method

We provide a summary table of the metrics achieved across the different methods in Table 5.3. Note that all the methods are $(2, 10^{-5})$-DP after 20 epochs. While we provide the best metric achieved for each sample in the table, this shouldn't be mistaken as the existence of a single model achieving all of them. Rather, it provides us with a way to assess the range of values taken by the entire Pareto optimal set. Additionally, we compute the hypervolume of the entire Pareto set. This serves as an indicator of the quality of the Pareto front generated by a specific method: a Pareto front with a higher hypervolume implies an algorithm that is able to produce a diverse set of Pareto optimal models close to the origin. Table 5.3 shows that Linear Scalarization and COSMOS outperform other approaches in the context of producing holistically trustworthy ML models (**RQ5**).

Method	Accuracy	PGD Accuracy	Natural Loss	Madry Loss	KL Div.	Lipschitzness	DPFR	Hypervolume
ERM	89.52	55.08	1.5934	1.7849	0.0128	0.0096	0.1674	4.8414
Scalarization	**92.76**	**61.35**	**1.5462**	**1.7078**	0.0133	0.0100	**0.0880**	**5.4696**
Pareto MTL	86.49	56.51	1.6088	1.7544	0.0190	0.0142	0.0909	5.0409
EPO	91.75	59.00	1.5597	1.7443	**0.0091**	**0.0068**	0.1606	5.1322
COSMOS	89.33	57.10	1.5760	1.7324	0.0181	0.0135	0.0918	5.2481

Table 5.3: Performance Comparison of the Methods

Takeaways: We empirically found that MOO algorithms have the capacity to produce a diverse and representative Pareto front in the context of four-dimensional trustworthiness objectives. While the initial models generated are wide-ranging, they all converge in the same direction (towards the origin) to achieve higher accuracy, robustness and fairness at the expense of privacy. We further show that the Linear Scalarization approach was able to produce a Pareto front with the largest hypervolume compared to the other methods.

5.5 Limitations and Discussion

While our experiments provide us with valuable insights into how various algorithms fair in the context of producing trustworthy ML, they need to be replicated on larger datasets to confirm the findings, e.g., UTKFace [74] and CheXpert [30]. Evaluation on larger datasets and more complex model architectures would allow us to confirm the feasibility of our approach and further evaluate the quality of the Pareto front generated across different methods and loss vectors.

We also acknowledge that we only employed in-processing fairness schemes to reduce discrimination throughout the training process. However, this doesn't provide us with a precise fairness guarantee at inference time. In order to do that, we would have to develop a post-processing algorithm to analyze a query's prediction and determine whether answering it would violate the fairness guarantee [71]. Moreover, we provide robustness through adversarial training as opposed to randomized smoothing which would ensure certified robustness. Since smoothing would require a separate parent model and only provides an ℓ_2-norm guarantee, we favored the use of adversarial training for its simplicity and strength regardless of the norm chosen.

CHAPTER 6

Conclusion and Future Work

In conclusion, we have dedicated this work towards developing a holistic framework for assessing and evaluating trustworthy models in the face of conflicting objectives. Our work contributes to future research in developing trustworthy models to be deployed, especially since machine learning continues to be used in high-stakes environments. We used the notion of Pareto front in comparing multi-objective solutions as it provides us with valuable insight into the achievable trade-off across the different constraints. We further presented an extensive evaluation across the objective space of different losses utilized as well as different training algorithms. Moreover, we empirically showed that applying algorithms specifically designed for MOO generally outperforms plain ERM regularization.

Accordingly, we encourage future work to continue evaluating and enhancing holistic machine learning trustworthiness in the context of MOO. While the intersectionality of MOO frameworks and ML trustworthiness has been left mostly unexplored by the research community, it provided our work with a systematic way to develop Pareto optimal solutions and assess the quality of the models generated. Future research should also attempt to increase the efficiency of our approach by incorporating various novel defenses for each domain in a joint manner, e.g., certified robustness, ensemble learning, PATE [46], and post-processing fairness guarantee. We also encourage the research community to incorporate additional trustworthiness requirements on top of the ones we considered, e.g., robustness against data poisoning, machine learning interpretability, etc.

APPENDIX A

Mathematical Equations

KL Divergence:
The KL divergence between two probability density functions P and Q defined over the sample space $\mathcal{X}$ is defined as:

$$D_{KL}(P||Q) = \sum_{x \in \mathcal{X}} \Big[P(x) \cdot \log\big(\frac{P(x)}{Q(x)}\big) \Big] \tag{A.1}$$

While the KL divergence isn't a distance metric (it doesn't satisfy the symmetry property nor the triangle inequality), it is conventionally used as a form of measure of how different two distributions p and Q are.

APPENDIX B

Algorithms

```
Require: $\theta$
  for $t \in [T]$ do
    Sample mini-batch $L_t$ with sampling probability $L/N$    {Poisson sampling}
    for $i \in L_t$ do
      Compute $g_t(x_i) \leftarrow \nabla_\theta \ell\big(f_\theta(x_i), y_i\big)$    {Compute gradient}
    end for
    $\tilde{g}_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{||g_t(x_i)||_2}{C})$    {Clip gradients}
    $\tilde{g}_t \leftarrow \frac{1}{|L_t|}(\sum_i \tilde{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$    {Add Gaussian noise}
    $\theta \leftarrow \theta - \eta_t \tilde{g}_t$    {Gradient Descent}
  end for=0
```

Program B.1: DP-SGD

```
Require: $\theta$
  for $t \in [T]$ do
    Sample mini-batch $L_t$ with sampling probability $L/N$
    for $i \in L_t$ do
      Compute $g_t(x_i) \leftarrow \nabla_\theta \Big(\ell\big(f_\theta(x_i), y_i\big) + \lambda \mathrm{DPFR}(\theta, \mathcal{D}_{public})\Big)$
    end for
    $\tilde{g}_t(x_i) \leftarrow g_t(x_i) / \max(1, \frac{||g_t(x_i)||_2}{C})$
    $\tilde{g}_t \leftarrow \frac{1}{|L_t|}(\sum_i \tilde{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$
    $\theta \leftarrow \theta - \eta_t \tilde{g}_t$
  end for=0
```

Program B.2: Fair DP-SGD

REFERENCES

[1] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 308–318. ACM, 2016.

[2] Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019.

[3] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 15453–15462, 2019.

[4] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Robustness may be at odds with fairness: An empirical study on class-wise accuracy. In Luca Bertinetto, João F. Henriques, Samuel Albanie, Michela Paganini, and Gül Varol, editors, *NeurIPS 2020 Workshop on Pre-registration in Machine Learning, 11 December 2020, Virtual Event*, volume 148 of *Proceedings of Machine Learning Research*, pages 325–342. PMLR, 2020.

[5] Franziska Boenisch, Philip Sperl, and Konstantin Böttinger. Gradient masking and the underestimated robustness threats of differential privacy in deep learning. *CoRR*, abs/2105.07985, 2021.

[6] Toon Calders and Sicco Verwer. Three naive bayes approaches for discrimination-free classification. *Data Min. Knowl. Discov.*, 21(2):277–292, 2010.

[7] Hongyan Chang and Reza Shokri. On the privacy risks of algorithmic fairness. In *IEEE European Symposium on Security and Privacy, EuroS&P 2021, Vienna, Austria, September 6-10, 2021*, pages 292–303. IEEE, 2021.

[8] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *CoRR*, abs/1712.05526, 2017.

[9] Christopher A. Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1964–1974. PMLR, 2021.

[10] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistik Tidskrift*, 15(429-444):2–1, 1977.

[11] Soham De, Leonard Berrada, Jamie Hayes, Samuel L. Smith, and Borja Balle. Unlocking high-accuracy differentially private image classification through scale. *CoRR*, abs/2204.13650, 2022.

[12] Michele Donini, Luca Oneto, Shai Ben-David, John Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2796–2806, 2018.

[13] Vasisht Duddu, Sebastian Szyller, and N. Asokan. Sok: Unintended interactions among machine learning defenses and risks. *CoRR*, abs/2312.04542, 2023.

[14] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[15] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[16] Jean-Antoine Désidéri. Multiple-gradient descent algorithm (mgda) for multiobjective optimization. *Comptes Rendus Mathematique*, 350(5):313–318, 2012.

[17] Birhanu Eshete. Making machine learning trustworthy. *Science*, 373(6556):743–744, 2021.

[18] Ferdinando Fioretto, Cuong Tran, Pascal Van Hentenryck, and Keyu Zhu. Differential privacy and fairness in decisions and learning tasks: A survey. In Luc De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*, pages 5470–5477. ijcai.org, 2022.

[19] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 1322–1333. ACM, 2015.

[20] Alex Gittens, Bülent Yener, and Moti Yung. An adversarial perspective on accuracy, robustness, fairness, and privacy: Multilateral-tradeoffs in trustworthy ML. *IEEE Access*, 10:120850–120865, 2022.

[21] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[22] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[23] Jamie Hayes. Provable trade-offs between private & robust machine learning. *CoRR*, abs/2006.04622, 2020.

[24] Jamie Hayes, Borja Balle, and M. Pawan Kumar. Learning to be adversarially robust and differentially private. *CoRR*, abs/2201.02265, 2022.

[25] Long P. Hoang, Dung D. Le, Tran Anh Tuan, and Tran Ngoc Thang. Improving pareto front learning via multi-sample hypernetworks. In Brian Williams, Yiling Chen, and Jennifer Neville, editors, *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, pages 7875–7883. AAAI Press, 2023.

[26] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S. Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Comput. Surv.*, 54(11s):235:1–235:37, 2022.

[27] Yuzheng Hu, Fan Wu, Hongyang Zhang, and Han Zhao. Understanding the impact of adversarial robustness on accuracy disparity. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 13679–13709. PMLR, 2023.

[28] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Comput. Sci. Rev.*, 37:100270, 2020.

[29] Yujia Huang, Huan Zhang, Yuanyuan Shi, J. Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*

34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pages 22745–22757, 2021.

[30] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Christopher Chute, Henrik Marklund, Behzad Haghgoo, Robyn L. Ball, Katie S. Shpanskaya, Jayne Seekins, David A. Mong, Safwan S. Halabi, Jesse K. Sandberg, Ricky Jones, David B. Larson, Curtis P. Langlotz, Bhavik N. Patel, Matthew P. Lungren, and Andrew Y. Ng. Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 590–597. AAAI Press, 2019.

[31] Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, 2001.

[32] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[33] Guofu Li, Pengjia Zhu, Jin Li, Zhemin Yang, Ning Cao, and Zhiyi Chen. Security matters: A survey on adversarial machine learning. *CoRR*, abs/1810.07339, 2018.

[34] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12037–12047, 2019.

[35] Xinsong Ma, Zekai Wang, and Weiwei Liu. On the tradeoff between robustness and fairness. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[36] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[37] Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6597–6607. PMLR, 2020.

[38] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6):115:1–115:35, 2022.

[39] Hanna F. Menezes, Arthur S. C. Ferreira, Eanes T. Pereira, and Herman M. Gomes. Bias and fairness in face detection. In *2021 34th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 247–254, 2021.

[40] Ilya Mironov, Kunal Talwar, and Li Zhang. Rényi differential privacy of the sampled gaussian mechanism. *CoRR*, abs/1908.10530, 2019.

[41] Vedant Nanda, Samuel Dooley, Sahil Singla, Soheil Feizi, and John P. Dickerson. Fairness through robustness: Investigating robustness disparity in deep learning. In Madeleine Clare Elish, William Isaac, and Richard S. Zemel, editors, *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pages 466–477. ACM, 2021.

[42] Jianjun Ni, Yinan Chen, Yan Chen, Jinxiu Zhu, Deena Ali, and Weidong Cao. A survey on theories and applications for self-driving cars based on deep learning methods. *Applied Sciences*, 10(8), 2020.

[43] Luca Oneto, Michele Donini, Amon Elders, and Massimiliano Pontil. Taking advantage of multitask learning for fair classification. In Vincent Conitzer, Gillian K. Hadfield, and Shannon Vallor, editors, *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES 2019, Honolulu, HI, USA, January 27-28, 2019*, pages 227–237. ACM, 2019.

[44] Kirtan Padh, Diego Antognini, Emma Lejal Glaude, Boi Faltings, and Claudiu Musat. Addressing fairness in classification with a model-agnostic multi-objective algorithm. In Cassio P. de Campos, Marloes H. Maathuis, and Erik Quaeghebeur, editors, *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021*, volume 161 of *Proceedings of Machine Learning Research*, pages 600–609. AUAI Press, 2021.

[45] Tiago Palma Pagano, Rafael Bessa Loureiro, Fernanda Vitoria Nascimento Lisboa, Rodrigo Matos Peixoto, Guilherme Aragao de Sousa Guimaraes, Gustavo Oliveira Ramos Cruz, Maira Matos Araujo, Lucas Lisboa dos Santos, Marco Cruz, Ewerton Lopes Silva de Oliveira, Ingrid Winkler, and Erick Giovani Sperandio Nascimento. Bias and unfairness in machine learning models: A systematic review on datasets, tools, fairness metrics, and identification and mitigation methods. *Big Data Cogn. Comput.*, 7(1):15, 2023.

[46] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with PATE. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[47] Neel Patel, Reza Shokri, and Yair Zick. Model explanations with differential privacy. In *FAccT '22: 2022 ACM Conference on Fairness, Accountability, and Transparency, Seoul, Republic of Korea, June 21 - 24, 2022*, pages 1895–1904. ACM, 2022.

[48] Patricia Pauli, Anne Koch, Julian Berberich, Paul Kohler, and Frank Allgöwer. Training robust neural networks using lipschitz bounds. *IEEE Control. Syst. Lett.*, 6:121–126, 2022.

[49] NhatHai Phan, My T. Thai, Han Hu, Ruoming Jin, Tong Sun, and Dejing Dou. Scalable differential privacy with certified robustness in adversarial learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7683–7694. PMLR, 2020.

[50] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8093–8104. PMLR, 2020.

[51] Michael Ruchte and Josif Grabocka. Scalable pareto front approximation for deep multi-objective learning. In James Bailey, Pauli Miettinen, Yun Sing Koh, Dacheng Tao, and Xindong Wu, editors, *IEEE International Conference on Data Mining, ICDM 2021, Auckland, New Zealand, December 7-10, 2021*, pages 1306–1311. IEEE, 2021.

[52] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 525–536, 2018.

[53] Laleh Seyyed-Kalantari, Guanxiong Liu, Matthew B. A. McDermott, Irene Y. Chen, and Marzyeh Ghassemi. Chexclusion: Fairness gaps in deep chest x-ray classifiers. In *Biocomputing 2021: Proceedings of the Pacific Symposium, Kohala Coast, Hawaii, USA, January 3-7, 2021.* WorldScientific, 2021.

[54] Claude E. Shannon. A mathematical theory of communication. *Bell Syst. Tech. J.*, 27(3):379–423, 1948.

[55] Reza Shokri, Martin Strobel, and Yair Zick. On the privacy risks of model explanations. In Marion Fourcade, Benjamin Kuipers, Seth Lazar, and Deirdre K. Mulligan, editors, *AIES '21: AAAI/ACM Conference on AI, Ethics, and Society, Virtual Event, USA, May 19-21, 2021*, pages 231–241. ACM, 2021.

[56] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017.

[57] Edward Small, Wei Shao, Zeliang Zhang, Peihan Liu, Jeffrey Chan, Kacper Sokol, and Flora D. Salim. How robust is your fair model? exploring the robustness of diverse fairness strategies. *CoRR*, abs/2207.04581, 2022.

[58] Liwei Song, Reza Shokri, and Prateek Mittal. Privacy risks of securing machine learning models against adversarial examples. In Lorenzo Cavallaro, Johannes Kinder, XiaoFeng Wang, and Jonathan Katz, editors, *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, CCS 2019, London, UK, November 11-15, 2019*, pages 241–257. ACM, 2019.

[59] Florian Tramèr and Dan Boneh. Differentially private learning needs better features (or much more data). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[60] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*, pages 601–618. USENIX Association, 2016.

[61] Cuong Tran, My H. Dinh, and Ferdinando Fioretto. Differentially private empirical risk minimization under the fairness lens. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 27555–27565, 2021.

[62] Guido Vittorio Travaini, Federico Pacchioni, Silvia Bellumore, Marta Bosia, and Francesco De Micco. Machine learning and criminal justice: A systematic review of advanced methodology for recidivism risk prediction. *International Journal of Environmental Research and Public Health*, 19(17), 2022.

[63] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[64] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 6542–6551, 2018.

[65] Praveen Tumuluru, Lakshmi Ramani Burra, M. Loukya, S. Bhavana, H.M.H. CSaiBaba, and N Sunanda. Comparative analysis of customer loan approval prediction using machine learning algorithms. In *2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, pages 349–353, 2022.

[66] Nurislam Tursynbek, Aleksandr Petiushko, and Ivan V. Oseledets. Robustness threats of differential privacy. *CoRR*, abs/2012.07828, 2020.

[67] Ana Valdivia, Javier Sánchez-Monedero, and Jorge Casillas. How fair can we go in machine learning? assessing the boundaries of accuracy and fairness. *Int. J. Intell. Syst.*, 36(4):1619–1643, 2021.

[68] Vladimir Vapnik. Principles of risk minimization for learning theory. In John E. Moody, Stephen Jose Hanson, and Richard Lippmann, editors, *Advances in Neural Information Processing Systems 4, [NIPS Conference, Denver, Colorado, USA, December 2-5, 1991]*, pages 831–838. Morgan Kaufmann, 1991.

[69] Kush R. Varshney. Trustworthy machine learning and artificial intelligence. *XRDS*, 25(3):26–29, 2019.

[70] Jiapeng Wu, Atiyeh Ashari Ghomi, David Glukhov, Jesse C. Cresswell, Franziska Boenisch, and Nicolas Papernot. Augment then smooth: Reconciling differential privacy with certified robustness. *CoRR*, abs/2306.08656, 2023.

[71] Mohammad Yaghini, Patty Liu, Franziska Boenisch, and Nicolas Papernot. Learning to walk impartially on the pareto frontier of fairness, privacy, and utility. In *NeurIPS 2023 Workshop on Regulatable ML*, 2023.

[72] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[73] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, 2019.

[74] Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 4352–4360. IEEE Computer Society, 2017.