AN ABSTRACT OF THE DISSERTATION OF

Zahir Alsulaimawi for the degree of Doctor of Philosophy in Electrical and Computer Engineering presented on August 11, 2021.

Title: Information-theoretic Approach to Design and Evaluate Privacy-preserving and Fair Frameworks for Continuous High-dimensional Data

Abstract approved: _____

Huaping Liu

Deep learning is becoming the latest trend in sensitive applications, such as healthcare, criminal justice, and finance. As these new applications emerge, adversaries are circumventing them. Further, there have been concerns about the possibility of bias and discrimination in predictive applications. In order to address these issues, we propose an information-theoretic approach to design a continuous high-dimensional data deep learning framework. We call this framework Gaussian privacy protector (GPP). Our proposed framework has many advantages: (1) it reduces the problem to the optimal compression of data about a measure of utility and privacy; (2) it can prevent adversaries from private mining information from the released data while simultaneously maximizing the amount of the utility's information revealed; (3) it adapts the idea of the information bottleneck (IB) based on the problem of revealing data, which is often sensitive; (4) it considers a privacy funnel (PF) problem inspired by utility data as the central part of

data to be revealed; (5) using a similar framework, we show how to achieve fairness in classification; and (6) this work illustrates the feasibility of creating a centralized platform to support this framework over distributed datasets. We utilize variational lower bounds of mutual information approximation implemented as supervised learning using an adversarial training algorithm. We use three datasets: hand-written digits (MNIST), celeb faces attributes (CelebA), and human activities and postural transitions' recognition using smartphone data (HAPT-Recognition) to evaluate our algorithms. The experimental results on these datasets demonstrate that the proposed approach effectively removes private information from the datasets while allowing non-private information to be mined effectively.

Information-theoretic Approach to Design and Evaluate Privacy-preserving and Fair Frameworks for Continuous High-dimensional Data

by

Zahir Alsulaimawi

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented August 11, 2021
Commencement June 2022

Doctor of Philosophy dissertation of Zahir Alsulaimawi presented on August 11, 2021.

APPROVED:

Major Professor, representing Electrical and Computer Engineering

---

Head of the School of Electrical Engineering and Computer Science

---

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Zahir Alsulaimawi, Author

ACKNOWLEDGEMENTS

"Success is not a random outcome, it is the result of predictable and powerful set of circumstances and opportunities. Successful people don't do success alone."

Outliers, Malcolm Gladwell.

Taking on this Ph.D. has been a profoundly life-changing experience for me, and I would not have accomplished this without the guidance and support I received from many people.

I would like to express my most profound appreciation to my advisor, Dr. Huaping Liu, for being an excellent academic advisor and guide me towards exciting and challenging research directions. His wise and aimed view in research made our works thrive, going beyond our limits. I would also like to extend my deepest gratitude to my computer science advisor, Dr. Xiaoli Fern, for giving me general insights into approaching machine-learning problems from different perspectives.

To my committee members, Dr. Ted Brekken and Dr. Eduardo Cotilla-Sanchez, I am profoundly grateful for their support, generosity, and encouragement on my Ph.D.

Pursuing a graduate degree at OSU was mainly due to the excellent professors I encountered in my life. I would like to thank Dr. Bella Bose, Dr. Weng-Keen Wong, Dr. Fuxin Li, and Dr. Xiao Fu for their fantastic courses. I would like to extend my sincere thanks to Dr. Glencora Borradaile and Mr. Calvin Hughes for their invaluable assistance.

I gratefully acknowledge the funding received towards my Ph.D. from the Silicon Valley Community Foundation (SVCF). I am also grateful to the Cisco Research University Funding Committee (CRUFC) funding to undertake my Ph.D.

I would also like to say a heartfelt thank you to my beloved father and mother and my supportive siblings. In particular, I want to thank my brothers Azhar and Zuhair for their sincere support throughout my studies. From my early childhood, Zuhair unconditionally supported me and taught me to pursue my dreams fearlessly. I would also like to thank my friend Dr. Mahmood A.K. for his constant support and assistance.

In the end, no words can sufficiently express my gratitude to my dear wife, Wardah. Your love, prayers, patience, understanding, and compassion helped me stay focused during these difficult times. Thank you for everything you have done for me. I am also grateful to my darling Ali (Sajjad), Ahmed, and Zahra for making the past six years a joy and allowing me to succeed.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF ALGORITHMS

Chapter 1: Introduction

The size and availability of datasets have significantly increased in the past few decades; private information is becoming more prevalent. Controlling privacy risks has become a priority for organizations that rely on this data [94]. A preprocessing step is also needed to compensate for existing biases in decision-making systems. For instance, a method that hides sensitive data from prying eyes is an excellent way to prevent privacy leaks [44]. Although it might not appear effective when dealing with high-dimensional data, such as data on a patient's health status, microarray gene expression data, and images [10].

Distributed computing involves designing and studying algorithms that allow a network of computers or devices to solve a common problem. It often fundamentally depends on message passing over device networks to coordinate state variables or decisions. Distributed computing networks have become pervasive in the industrial and consumer internet of things (IoT) networks [60], distributed learning systems for medical purposes [99], and financial applications [151]. These systems increasingly handle private and sensitive information, such as medical and financial information. Thus, we must protect distributed computing systems from privacy attacks by adversaries. Consider a scenario in which adversaries have corrupted or taken control of devices in the network. Such adversaries may store and exploit the observed information to estimate the private data associated with non-corrupt devices [63].

In this work, we propose a variational approach for learning public representations for high-dimensional data. In presenting this data, we aim to maintain a prescribed level of relevant information that is not shared by private or sensitive data while minimizing the remaining information they hold. In other words, we will discuss how to prepare a dataset so that the privacy of the individuals in it is not compromised and that the prepared dataset is still useful in certain circumstances. We will study in particular a private data release mechanism based on the idea of optimal compression by projecting the data onto low-dimensional space, motivated by the idea that low-dimensional representations would have a smaller sensitivity than the raw data itself. Thus, the amount of privacy leakage could be significantly lower. This approach also addresses the distributed multiple data sources' privacy as training in a centralized framework.

The proposed approach is summarized as follows:

(1) Provides a demonstration of the "similarity solution" as a means of preserving privacy and achieving fair classification.

(2) Can be comfortably incorporated into standard representation learning algorithms, including information bottleneck (IB) and privacy funnel (PF).

(3) Enables the tradeoff between utility and privacy by controlling the dimensions of data representation.

(4) Proposes a distributed private learning framework for protecting the privacy of sensitive demographic data.

1.1 Motivation Behind the Research

Privacy-preserving: In the privacy-preserving era, it is crucial to ensure that the provided data, while delivering utility, does not expose critical sensitive information. Many data owners, such as hospitals, cannot share data due to privacy and confidentiality concerns. Additionally, users who want to utilize continuous high-dimensional data face the lack of suitable privacy-preserving machine learning frameworks. One way to achieve privacy would be to remove sensitive information from a dataset. Despite this, the utility data can be lost due to the correlation between sensitive and valuable data. A good example is removing confidential information from an image such as race but retaining gender. Several approaches to privacy-preserving computations for data mining are available, including anonymization and differential privacy [149], involving some form of perturbation of the data. The perturbation techniques provide privacy guarantees for datasets with categorical attributes. However, they may not be suitable for datasets of continuous high-dimensional data such as images, videos, and audio clips [116] [133]. Also, in some cases, it is possible to reconstruct a part of training data by only observing the predictions, such as recovering images from a facial recognition system through model-inversion attack [59]. In addition, it is possible to deduce whether a particular training point is involved in the model's training data by observing only the predictions of the model through a membership-inference attack [136].

Fairness in Classification: The act of discrimination is the unjustified treatment of individuals based on their membership in, or perception of membership in, a particular group, and is often a result of the group being treated less favorably than others. In light

of the increasing use of machine learning algorithms in many areas of our daily lives, including salary prediction, hiring, and criminal risk assessment, fair machine learning algorithms must be developed [111]. A majority of current discussions are focused on improving supervised learning algorithms with fairness requirements, namely that sensitive information such as gender or race not unfairly affect a learning algorithm's performance [51] [85]. The naive approach is to discard sensitive information and use the other data as raw input. This approach has some proponents in terms of process; however, the raw data might include important information about sensitive information. Individual home address information, for example, may be a good indicator of race. In this regard, it is important to actively decorrelate the mutual relationship between the target and sensitive information. Accordingly, fairness research aims to design objective functions that approximate specific fairness notions while simultaneously maximizing predictive accuracy.

1.2 Research Questions

In this dissertation, we examine real-world scenarios that present significant challenges when it comes to providing privacy guarantees in practice. The overall goal of this dissertation is to address the following three research questions:

(1) Is it possible to build and evaluate a privacy-preserving deep learning framework for high-dimensional data that incorporates public data to extract useful information with solid privacy protections?

(2) Is it possible to train deep learning models using aggregate statistics gathered from

many data sources without publicizing sensitive information?

(3) Is there a way to incorporate fairness in classification with the deep learning framework?

1.3 Research Scope

Increasing adoption of IoT technologies such as mobile computing and social networking will lead to both conveniences, and privacy concerns [134]. A good example of such a technology is remote patient monitoring, which has become the norm in patient care today. These technologies, however, also pose serious privacy concerns and concerns regarding the logging and transfer of data transactions. For instance, medical data privacy problems could result from a delay in treatment progress, even endangering the patient's life [49]. As another example, smart homes with many internet-connected devices continuously transmit information about the users' daily activities. Data like this could be used to infer consumer behavior, which raises privacy concerns [124]. Additionally, most IoT devices are resource-constrained, requiring low bandwidth and low computing power [119]. In this dissertation, we use a privacy-preserving framework for online data communication technology to address the issues mentioned above.

1.4 Research Objectives

The objectives of this study are as follows:

(1) Investigate privacy-preserving and fair classification models, as well as how to se-

cure sensitive data.

(2) Develop new privacy-preserving algorithms to ensure a high level of privacy protection while maintaining utility.

(3) Design a centralized training framework in which multiple participants train an accurate global model collaboratively.

(4) Extend the proposed approach to the IB and PF problems.

(5) Provide an optimization framework for determining a fair classifier in machine learning.

(6) Evaluate the proposed framework using real-life and synthetic datasets, as well as compare them to existing approaches.

1.5 Research Methodology

This research adopts an information-theoretic approach [29] [30] to preserve the privacy of continuous high-dimensional datasets, which aims to achieve a privacy-utility tradeoff between private and non-private data. This research considers a situation where users have access to two kinds of correlated data. The first type of data is private, while the second type is not private but is made publicly available. The main idea of our model is to map continuous high-dimensional data in a given input space to a sanitized version of that data in new representation space. This new representation conceals any sensitive information that could indicate whether private events have taken place or not while preserving as much utility information as possible. Our work is based on a proposed concept called *privacy-preserving data release mechanisms* [10] [18] [152] to

achieve a good tradeoff between data utility and data privacy under constrained data release mechanisms. In our work, we refer to this approach as Gaussian privacy protector (GPP). Specifically, GPP is a *probabilistic mapping* training along with an adversarial network that attempts to recover the private information from the sanitized dataset and utility network to derive utility for the released data.

The IoT is becoming increasingly popular in sectors such as energy, transportation, healthcare, and manufacturing. The result is the generation of unprecedented volumes of data, and this trend will continue. A central approach for analyzing IoT data is limited by privacy concerns and the requirement for fast processing, low latency, and sufficient bandwidth of the communications network. This research investigates the privacy problem as a distributed computing environment where multiple GPP can acquire data and learn together. Also, reduced communication costs were achieved by using low-dimensional exposed data.

1.6 Contributions

This research makes the following contributions.

- In comparison to previous studies, we introduce GPP framework to capture the best privacy-utility tradeoff problem on benchmark datasets.
- Our proposed framework can be modified to achieve different privacy-utility tradeoffs, including information bottleneck and privacy funnel.
- The GPP framework removes sensitive information from raw high-dimensional

data and creates representations as law-dimensional data. Due to this, low dimensional output space introduces a privacy-utility tradeoff.

- The GPP is a privacy-preserving machine learning method that can be applied to IoT applications as a possible means of reducing communication costs.
- Mutual information is an essential quantity for expressing and understanding the statistical dependence between random variables. We explicitly formulate the privacy-preserving issue by a lower bound estimator for the mutual information based on the variational method.
- We propose a more robust evaluation of real-life problems by considering multiple adversaries and multiple utility gains.
- With the proposed federated learning model, we offer improved features over traditional distributed machine learning by learning sensitive data locally and uploading the sanitized data to the central server.

1.7 Organization

The rest of the dissertation is organized as follows. In Chapter 2, we provide an overview of the key concepts and tools discussed throughout the dissertation. Chapter 3 discusses relevant related work aimed at protecting privacy and ensuring fairness in classification. We introduce a privacy-preserving algorithm for balancing data privacy and utility in Chapter 4. We also propose a distributed learning algorithm for learning utility representations without sharing raw data. New approaches to building information bottle-

necks and privacy funnels are explored in Chapter 5. In Chapter 6, we present a fairness framework for a machine-learning classification system. The results of the experiment and analysis of the proposed algorithms are presented in chapter 7. In Chapter 8, we conclude the dissertation and discuss our contributions.

1.8 Publications

1. "Sequential Game Network (SEGANE) with Application to Online Data Sanitization," *IEEE Global Conference on Signal and Information Processing*, Anaheim, CA, USA, 2018.

2. "Gaussian Privacy Protector for Online Data Communication in a Public World," *6th IEEE Big Data Security*, Baltimore, MD, USA, 2020.

3. "A Privacy Filter Framework for Internet of Robotic Things Applications," *41th IEEE Symposiumon security and privacy workshops*, San Francisco, CA, US, 2020.

4. "Variational Bound of Mutual Information for Fairness in Classification," *IEEE 22nd International Workshop on Multimedia Signal Processing*, Tampere, Finland, 2020.

5. "A Non-Negative Matrix Factorization Framework for Privacy-Preserving and Federated Learning," *IEEE 22nd International Workshop on Multimedia Signal Processing, IEEE MMSP*, Tampere, Finland, 2020.

6. "Distributed Variational Information Bottleneck for IoT Environments," *IEEE International Workshop on Machine Learning for Signal Processing, IEEE MLSP*, Gold Coast, Queensland, Australia, 2021.

Chapter 2: Background

In this chapter, we present the background material on which this dissertation is based. In the following section, we will provide formal definitions of privacy-related terms. We then survey data models and relevant techniques. In addition, the most common terminology and mathematical formulations of federated learning are discussed.

2.1 Definitions

Data Holder (Client): Several parties would like to develop a data model (or several models) to protect sensitive information [138].

Privacy: Companies, organizations, or individuals decide when, how, and what disclosure information to protect [62].

Privacy-preserving: Organizations and individuals can protect their privacy in hostile environments using privacy-preserving techniques [62].

Private Information (Sensitive or Protected Features): Information that is intended to remain confidential includes social security numbers, credit card numbers, and health information [108] [88].

Utility Information (Non-sensitive Features or Public Information): Data such as names, salary information, and telephone directories that are the result of privacy-preserving models as outputs [152] [88].

Adversary: When working with an adversary model, you're trying to learn sensitive information from the data holder [34].

Privacy-Utility Tradeoff: The system designer aims to construct a privacy-preserving model that provides the desired privacy and utility levels while achieving the optimal tradeoff [152].

2.2 Information Theory

We adopt the same notation for information-theoretic as used quantities used in [42]. In particular, H stands for entropy, I for mutual information, and KL for Kullback-Leibler divergence. This dissertation will use the concepts outlined in the following paragraphs.

Entropy: Given a random variable (RV) X and probability mass function P, entropy is:

$$H(X) = -\sum_{X} P(X)\log P(X) = \mathbb{E}_X[-\log P(X)]. \tag{2.1}$$

By measuring this, the information content is measured, i.e., expected uncertainty in X. The following are its properties:

- $H(X) \geq 0$, entropy is always non-negative.
- $H(X) = 0$ if and only if X is deterministic.

Conditional Entropy: Based on a RV X and a RV Y, the conditional entropy is:

$$H(Y|X) = \mathbb{E}_X \mathbb{E}_{Y|X}\left[-\log P(Y|X)\right] = E_{X,Y}[-\log P(Y \mid X)]. \tag{2.2}$$

It has the following properties:

- $H(Y \mid X) \neq H(X \mid Y)$.
- $H(Y \mid X) \geq 0$.

Kullback-Leibler Divergence: Assuming two probability distributions on the same alphabet, $P(X)$ and $Q(X)$, KL divergence $KL(P|Q)$ is a measure of their discrepancy as:

$$KL(P||Q) = \mathbb{E}_P \left[\log \frac{P(X)}{Q(X)} \right]. \tag{2.3}$$

Mutual Information: One of the fundamental quantities of information theory is mutual information, which measures how much information one RV conveys to another. In other words, the mutual information $I(X;Y)$ of two RVs X and Y is a measure of the dependence between the two RVs, satisfying $I(X;Y) \geq 0$, with equality if and only if, X and Y are mutually independent. It is defined as the KL divergence between the joint distribution $P(X,Y)$ and the independent distribution $P(X)P(Y)$ generated by the marginal ones:

$$I(X;Y) = KL(P(X,Y)||P(X)P(Y)) = \mathbb{E}_{X,Y} \left[\log \frac{P(X,Y)}{P(X)P(Y)} \right]. \tag{2.4}$$

Mutual information and entropy are related as follows:

$$I(X;Y) = H(X) - H(X \mid Y). \tag{2.5}$$

Note: $I(X;Y) = I(Y;X)$ (symmetry).

2.3 Classification Task

A classification task involves predicting an outcome Y from label space $\mathscr{Y}$, based on some observation X from a feature space $\mathscr{X}$. Let $P(X,Y)$ be a distribution over $\mathscr{X} \times \mathscr{Y}$. For such problems, we might have access to a data set of n pair $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$. A model, $f : \mathscr{X} \to \mathscr{Y}$, is called a classification function, and its true risk w.r.t. P is $R(f) \triangleq \mathbb{E}_{P(X,Y)}[\mathcal{L}(f(\mathbf{x}), \mathbf{y})]$, where $\mathcal{L} : \mathscr{Y} \times \mathscr{Y} \to \mathbb{R}$ is a given loss function, for example, the cross-entropy (CE) loss or the squared loss. Then $\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y})$ quantifies the cost of classifying $\hat{\mathbf{y}}$ when the true outcome is $\mathbf{y}$ and the aim is to ensure that $\mathcal{L}(f(\mathbf{x}), \mathbf{y})$ is small. For instance, given a dataset $D_n = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n \subseteq (\mathscr{X} \times \mathscr{Y})^n$ sampled i.i.d. from $P(X,Y)$, and the empirical risk of a neural network classifier f is $\hat{r}(f \mid D_n) \triangleq \frac{1}{n} \sum_{i=1}^n \mathcal{L}\left(f\left(\mathbf{x}_i, \theta\right), \mathbf{y}_i\right)$, with learnable weights θ, the classifier f aim is to classify an $\mathbf{x}$ into one of a finite number of classes (that is, the label space $\mathscr{Y}$ is finite) by minimizing the empirical risk $\min_f \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathscr{D}}[\mathcal{L}(f(\mathbf{x}, \theta), \mathbf{y})]$. If all mistakes are equally bad, we could define f as a binary classifier as follows,

$$\mathcal{L}(f(\mathbf{x}), \mathbf{y}) = \mathbf{1}(f(\mathbf{x}) \neq \mathbf{y}) = \begin{cases} 1 & \text{if } f(\mathbf{x}) \neq \mathbf{y} \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

where $\mathbf{1}(\cdot)$ is the indicator function whose value is 1 if its argument is true and 0 otherwise.

2.4 Autoencoder

An autoencoder is a type of artificial neural network (ANN) designed for unsupervised machine learning. It consists of an encoder and a decoder, as depicted in Figure 2.1. Basically, it reconstructs the original input while compressing the data to produce a more efficient and compressed representation [76]. As long as the hidden units include the good features, the autoencoder can minimize the reconstruction error [142].

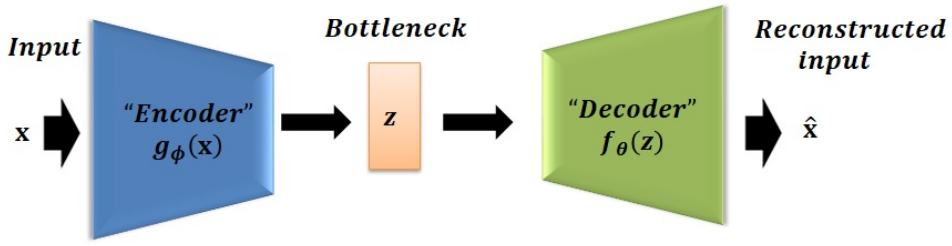

Figure 2.1: Autoencoder architecture.

Encoder Network: Using an encoder, high-dimensional inputs can be compressed into latent low-dimensional inputs, which a vector function can describe as:

$$g : \mathbf{x} \in \mathbb{R}^l \longmapsto \mathbf{z} \in \mathbb{R}^m, \tag{2.7}$$

where $g_\phi(\mathbf{x})$ is an encoder function parameterized by ϕ.

Decoder Network: Decoder networks recover the data from the code, probably with larger and larger output layers, which can be expressed as:

$$f : \mathbf{z} \in \mathbb{R}^m \longmapsto \hat{\mathbf{x}} \in \mathbb{R}^l, \tag{2.8}$$

where $f_\theta(\mathbf{z})$ is a decoder function parameterized by θ.

Loss Function: The parameters ϕ and θ are learned together to output a reconstructed data sample $\hat{\mathbf{x}} = f_\theta\left(g_\phi(\mathbf{x})\right)$ that is ideally the same as the original input $\mathbf{x}$. Various loss functions are used to quantify the error between the input and output, such as CE, or more specific mean squared error

$$\mathcal{L}(\theta,\phi) = \frac{1}{n}\sum_{i=0}^{n}\left(x_i - f_\theta\left(g_\phi\left(x_i\right)\right)\right)^2. \tag{2.9}$$

An autoencoder's sensitivity to input data is a crucial challenge when designing one. Autoencoders should learn a representation that embeds key data features correctly, encode features generalized beyond the training dataset, and capture similarly structured features for other datasets.

2.5 Deep Learning Based Generative Models

Deep learning has achieved impressive success in applications for which the goal is to model a conditional distribution $P(\mathbf{y}|\mathbf{x})$, with $\mathbf{y}$ being a label and $\mathbf{x}$ the features. While the conditional model $P(\mathbf{y}|\mathbf{x})$ may be highly accurate on inputs $\mathbf{x}$ sampled from the training distribution, there are no guarantees that the model will work well on $\mathbf{x}$'s drawn from some other distribution. One way to avoid such over-confidently wrong predictions would be to train a generative model $P_\theta(\mathbf{x})$, parametrize by θ, to approximate the proper distribution of training inputs $P^*(x)$ and refuse to predict any $\mathbf{x}$ that has a sufficiently low density under $P_\theta(\mathbf{x})$. The logical conclusion is that the discriminative model $P(\mathbf{y}|\mathbf{x})$ did not observe enough samples in that area to make a valid judgment for those inputs. The

generative model can produce or output new examples that could have been derived from the original dataset, i.e., a generative model aims to approximate the distribution of actual data [141] [114].

Deep generative models have demonstrated an impressive capacity to generate highly realistic pieces of content of various types, such as images, texts, and sounds, by relying on massive data, well-designed network architectures, and intelligent training techniques. Two significant families stand out among these deep generative models and deserve special attention: generative adversarial networks (GANs) [67] and variational autoencoders (VAEs) [90].

2.5.1 Generative Adversarial Networks

Originally, vanilla GAN was a two-player minimax game where a neural network represented each player. The one is a discriminator, while the other is a generator. Figure 2.2 depicts the design idea. The generator creates samples as close to the real data samples as possible by taking random noise as input. The discriminator takes samples from both real and generated data and tries to distinguish between them by reporting real or false output in either case. The optimization task aims to arrive at where the generator can produce samples that the discriminator cannot differentiate from real ones. In other words, the discriminator should produce a probability of 0.5 for either real or generated data. Another way to look at it is that the GANs find structure in the data, which helps to create more accurate data.

Minimax Loss: GAN can be trained with two loss functions: one for the generator

and one for the discriminator. In generator training, the generator minimizes the loss function, whereas the discriminator maximizes it

$$\min_G \max_D V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))], \tag{2.10}$$

where $D(\mathbf{x})$ is the discriminator's estimate of the probability that real data instance $\mathbf{x}$ is real, $G(\mathbf{z})$ is the generator's output when given noise $\mathbf{z}$, and $D(G(\mathbf{z}))$ is the discriminator's estimate of the probability that a fake instance $G(\mathbf{z})$ is real.

Training: The generator and discriminator are the two players in a GAN, where the weights of their models are updated alternately. The generator tries to reduce the log of the inverse probability predicted by the discriminator (i.e., minimize $\log(1–D(G(\mathbf{z})))$). On the other hand, the discriminator aims to maximize the log probability of real images and the log of inverted probabilities of fake images (i.e., maximize $\log D(\mathbf{x}) + \log(1–D(G(\mathbf{z})))$). Algorithm 1 is summarized the GAN training process taken from the original paper [67].

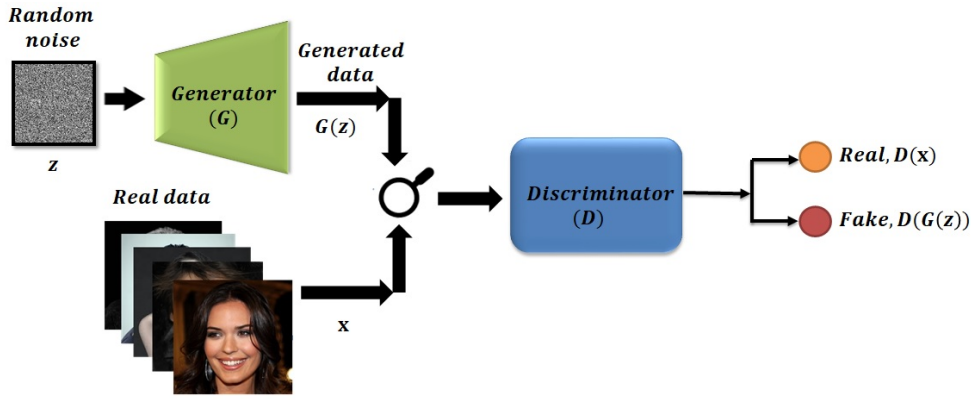

Figure 2.2: High-level description of the GAN.

norelsize 1 GAN training algorithm.

1: **for** number of training iterations **do**

2: **for** k steps **do**

- Sample minibatch of m noise samples $\left\{\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(m)}\right\}$ from noise prior $p_g(\mathbf{z})$.
- Sample minibatch of m examples $\left\{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)}\right\}$ from data generating distribution $p_{\text{data}}(\mathbf{x})$
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(\mathbf{x}^{(i)}\right)+\log\left(1-D\left(G\left(\mathbf{z}^{(i)}\right)\right)\right)\right]$$

3: **end for**

- Sample minibatch of m noise samples $\left\{\mathbf{z}^{(1)},\ldots,\mathbf{z}^{(m)}\right\}$ from noise prior $p_g(\mathbf{z})$.
- Update the generator by ascending its stochastic gradient:

$$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\left(1-D\left(G\left(\mathbf{z}^{(i)}\right)\right)\right).$$

4: **end for**

2.5.1.1 Information GAN

An information GAN (InfoGAN) is a form of GAN that learns interpretable and meaningful representations. The mutual knowledge between a fixed small subset of the GAN's noise variables and the observations is maximized in this way [167]. In other words, InfoGAN approaches this problem by splitting the generator input into two parts: the traditional noise vector $\mathbf{z}$ and a new "latent code" vector $\mathbf{c}$. The code vector $\mathbf{c}$ is then made meaningful by maximizing the mutual information between the code $\mathbf{c}$ and the generator output $G(\mathbf{z},\mathbf{c})$. This framework is implemented by simply adding a regular-

ization term $I(\mathbf{c}; G(\mathbf{z}, \mathbf{c})$ to the original GAN's objective function, as in below [39]:

$$\min_G \max_D V_{InfoGAN}(D, G) = V_{GAN}(D, G) - \lambda I(\mathbf{c}; G(\mathbf{z}, \mathbf{c})), \tag{2.11}$$

where λ is a weight parameter. Also, we can write the InfoGAN objective function combines the mutual information lower bound L_I with the standard GAN objective as

$$\min_{G,Q} \max_D V_{InfoGAN}(D, G, Q) = V_{GAN}(D, G) - \lambda L_I(G, Q). \tag{2.12}$$

The Variational Bound on Mutual Information: Let $\mathbf{c}$ be a random variable representing the latent information code and $\mathbf{x}_g = G(\mathbf{z}, \mathbf{c})$ be a random variable representing the generated data produced by generator G from the code $\mathbf{c}$ and noise $\mathbf{z}$. we want to find the variational lower bound on mutual information between two RVs $\mathbf{c}$ and $\mathbf{x}_g$, with joint distribution $P(\mathbf{c}, \mathbf{x}_g)$. As shown in [4], this yields a lower bound on mutual information according to the fact that KL divergence is non-negative gives

$$\begin{aligned}
I(\mathbf{c}; \mathbf{x}_g) &= H(\mathbf{c}) - H(\mathbf{c} \mid \mathbf{x}_g) \\
&= H(\mathbf{c}) + \mathbb{E}_{\mathbf{x}_g} \mathbb{E}_{\mathbf{c}|\mathbf{x}_g} [\log P(\mathbf{c} \mid \mathbf{x}_g)] \\
&= H(\mathbf{c}) + \mathbb{E}_{\mathbf{x}_g} \mathbb{E}_{\mathbf{c}|\mathbf{x}_g} \left[\log \frac{P(\mathbf{c} \mid \mathbf{x}_g) Q(\mathbf{c} \mid \mathbf{x}_g)}{Q(\mathbf{c} \mid \mathbf{x}_g)} \right] \\
&= H(\mathbf{c}) + \mathbb{E}_{\mathbf{x}_g} \mathbb{E}_{\mathbf{c}|\mathbf{x}_g} [\log Q(\mathbf{c} \mid \mathbf{x}_g)] + \mathbb{E}_{\mathbf{x}_g} \mathbb{E}_{\mathbf{c}|\mathbf{x}_g} \left[\log \frac{P(\mathbf{c} \mid \mathbf{x}_g)}{Q(\mathbf{c} \mid \mathbf{x}_g)} \right] \\
&= H(\mathbf{c}) + \mathbb{E}_{\mathbf{x}_g, \mathbf{c}} [\log Q(\mathbf{c} \mid \mathbf{x}_g)] + \mathbb{E}_{\mathbf{x}_g} [KL(P(\mathbf{c} \mid \mathbf{x}_g) \| Q(\mathbf{c} \mid \mathbf{x}_g))] \\
&\geq H(\mathbf{c}) + \mathbb{E}_{\mathbf{x}_g, \mathbf{c}} [\log Q(\mathbf{c} \mid \mathbf{x}_g)].
\end{aligned} \tag{2.13}$$

This indicates that we can maximize the mutual information $I(\mathbf{c};\mathbf{x}_g)$ by maximizing $\mathbb{E}_{\mathbf{x}_g,\mathbf{c}}[\log Q(\mathbf{c} \mid \mathbf{x}_g)]$ or by minimizing the negative log likelihood of $Q(\mathbf{c} \mid \mathbf{x}_g)$. We can write the lower bound alternatively in the following way:

$$I(\mathbf{c};\mathbf{x}_g) = \max_{Q} \left\{ H(\mathbf{c}) + \mathbb{E}_{\mathbf{x}_g,\mathbf{c}} \log Q(\mathbf{c} \mid \mathbf{x}_g) \right\}, \tag{2.14}$$

where $Q(\mathbf{c} \mid \mathbf{x}_g)$ is a discriminator approximation of the posterior $P(\mathbf{c} \mid \mathbf{x}_g)$.

The InfoGAN Framework: As shown in Figure 2.3, the generator takes random noise and latent code as input and produces generated data. Discriminator takes samples from both real and generated data and attempts to differentiate between the two by reporting real or false output for either sample. The Q network is a fully connected layer tacked onto the last representation layer of the discriminator, and it is essentially trying to predict what the latent code is.

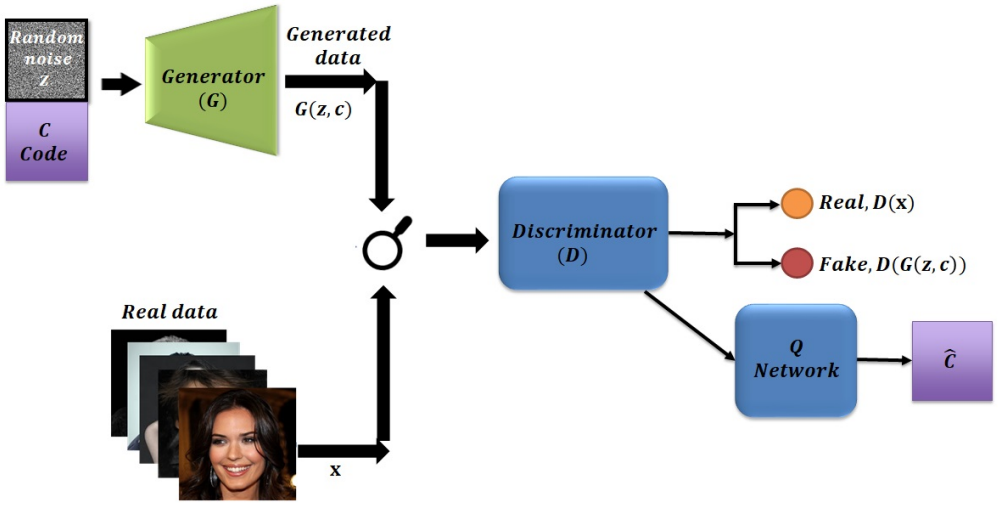

Figure 2.3: High-level description of the InfoGAN.

2.5.2 Variational Autoencoder

The sensitivity of an autoencoder to the input data is the key challenge when designing one. The sensitivity of an autoencoder to the input data is the key challenge when designing one. While an autoencoder should learn a representation that correctly embeds the main data features, it should also encode features that generalize outside the original training set and capture similar features in other datasets [137]. Since the introduction of autoencoders, numerous variations have been suggested. These variants are primarily intended to correct flaws, such as enhanced generalization, avoid overfitting and improve the robustness; such notable examples include denoising autoencoder [147], sparse autoencoder [41], contractive autoencoder [131] and VAE. The idea of VAE is deeply rooted in the methods of variational bayesian and graphical models. VAE learns the input data's underlying distribution parameters rather than just a compressed representation, i.e., making its encoder output two vectors of size n: a vector of means, μ, and another vector of standard deviations, σ. The relationship between the data input $\mathbf{x}$ and the latent encoding vector $\mathbf{z}$ can be defined by $Q_\phi(\mathbf{z}|\mathbf{x})$ as a probabilistic encoder parameterized by ϕ, playing a similar role as $g_\phi(\mathbf{x})$, and $P_\theta(\mathbf{x}|\mathbf{z})$ as a probabilistic decoder parameterized by θ, similar to the decoder $f_\theta(\mathbf{x})$ introduced above. We note that the lower-dimensional space z is stochastic: a standard Gaussian distribution.

Variational Inference: The aim of Bayesian statistics is to find a posterior distribution $P(\mathbf{z}|\mathbf{x})$ of a latent variable $\mathbf{z}$ given some evidence $\mathbf{x}$. However, computing this posterior

distribution is usually difficult since, according to Bayes,

$$P(\mathbf{z} \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid \mathbf{z})P(\mathbf{z})}{P(x)} = \frac{P(\mathbf{x} \mid \mathbf{z})P(\mathbf{z})}{\int_{\mathbf{z}} P(\mathbf{x} \mid \mathbf{z})P(\mathbf{z})d\mathbf{z}} \tag{2.15}$$

it requires computing the integral over the entire latent space $\mathbf{z}$. To get around the intractability problem, one approximates the posterior with another distribution $Q_\phi(\mathbf{x} \mid \mathbf{z})$ in such a way that the similarity measure between the true posterior and the approximation, $Q_\phi(\mathbf{x} \mid \mathbf{z})$, is minimized. A deep neural network, the encoder, is used to model the approximate posterior, $Q_\phi(\mathbf{x} \mid \mathbf{z})$, which generates distribution statistics that are usually Gaussian in the latent space. Our goal is to find the variational parameters ϕ that minimize $\mathbb{KL}\left(Q_\phi(\mathbf{z} \mid \mathbf{x}) \| P_\theta(\mathbf{z} \mid \mathbf{x})\right)$. The optimal approximate posterior is thus:

$$Q^*_\phi(\mathbf{z} \mid \mathbf{x}) = \arg\min_\phi KL\left(Q_\phi(\mathbf{z} \mid \mathbf{x}) \| P_\theta(\mathbf{z} \mid \mathbf{x})\right). \tag{2.16}$$

Consider the following function:

$$
\begin{aligned}
&\mathbb{KL}\left(Q_\phi(\mathbf{z}\mid\mathbf{x})\|P_\theta(\mathbf{z}\mid\mathbf{x})\right)\\
&=\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\log\tfrac{Q_\phi(\mathbf{z}|\mathbf{x})}{P_\theta(\mathbf{z}|\mathbf{x})}\\
&=\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\log\tfrac{Q_\phi(\mathbf{z}|\mathbf{x})P_\theta(\mathbf{x})}{P_\theta(\mathbf{z},\mathbf{x})}\\
&=\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\left(\log P_\theta(\mathbf{x})+\log\tfrac{Q_\phi(\mathbf{z}|\mathbf{x})}{P_\theta(\mathbf{z},\mathbf{x})}\right)\\
&=\log P_\theta(\mathbf{x})+\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\log\tfrac{Q_\phi(\mathbf{x}|\mathbf{x})}{P_\theta(\mathbf{z},\mathbf{x})}\\
&=\log P_\theta(\mathbf{x})+\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\log\tfrac{Q_\phi(\mathbf{z}|\mathbf{x})}{P_\theta(\mathbf{x}|\mathbf{z})P_\theta(\mathbf{z})}\\
&=\log P_\theta(\mathbf{x})+\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\left[\log\tfrac{Q_\phi(\mathbf{z}|\mathbf{x})}{P_\theta(\mathbf{z})}-\log P_\theta(\mathbf{x}\mid\mathbf{z})\right]\\
&=\log P_\theta(\mathbf{x})+KL\left(Q_\phi(\mathbf{z}\mid\mathbf{x})\|P_\theta(\mathbf{z})\right)-\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\log P_\theta(\mathbf{x}\mid\mathbf{z})
\end{aligned}
\tag{2.17}
$$

After rearranging the equation's left and right sides,

$$
\begin{aligned}
\log P_\theta(\mathbf{x})-KL\left(Q_\phi(\mathbf{z}\mid\mathbf{x})\|P_\theta(\mathbf{z}\mid\mathbf{x})\right)=\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\log P_\theta(\mathbf{x}\mid\mathbf{z})-\\
KL\left(Q_\phi(\mathbf{z}\mid\mathbf{x})\|P_\theta(\mathbf{z})\right).
\end{aligned}
\tag{2.18}
$$

The negation of the equation's right-hand side is the variational autoencoder's loss function. The first term is the reconstruction loss or expected negative log-likelihood of the data point, and the second term is a regularizer (this is KL between the encoder's distribution $Q_\phi(\mathbf{z}\mid\mathbf{x})$ and $P_\theta(\mathbf{z})$). Therefore the model is optimized by finding optimal coefficients that minimize lost function as:

$$
\theta^*,\phi^*=\arg\min_{\theta,\phi}-\mathbb{E}_{Q_\phi(\mathbf{z}|\mathbf{x})}\log P_\theta(\mathbf{x}\mid\mathbf{z})+KL\left(Q_\phi(\mathbf{z}\mid\mathbf{x})\|P_\theta(\mathbf{z})\right) \tag{2.19}
$$

Reparameterization Trick: The loss function's expectation term invokes generating samples from $\mathbf{z} \sim Q_\phi(\mathbf{z}|\mathbf{x})$. We can not backpropagate the gradient because sampling is a stochastic operation. The reparameterization trick is used to make it trainable, which suggests that we randomly sample ε from a unit Gaussian, and then shift the randomly sampled ε by the latent distribution's mean μ and scale it by the latent distribution's variance σ as:

$$\mathbf{z} = \mu + \sigma \odot \varepsilon, \text{ where } \varepsilon \sim \mathcal{N}(0, I). \tag{2.20}$$

Now, we can optimize the distribution parameters while still sampling randomly from its distribution. Figure 2.4 illustrates a VAE model with the multivariate Gaussian assumption.

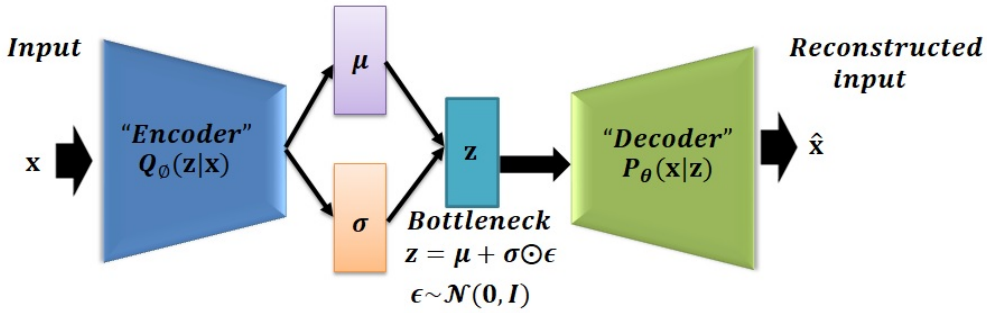

Figure 2.4: Vanila VAE architecture.

2.6 Bayesian Networks

A Bayesian Network $BN{=}<N,A,\Delta>$ is a directed acyclic graph (DAG) $<N,A>$ with a conditional probability distribution for each node, collectively represented by Δ. Each node $n \in N$ represents a RV, and each edge $a \in A$ between nodes represents a probabilis-

tic dependency between the associated nodes [40]. A BN's joint distribution of n variables $P(X_1,\ldots,X_n)$ is equal to the product of a conditional probability $P(node|parents(node))$ for all nodes, as shown below [77]:

$$P(X_1,\ldots,X_n)=\prod_{i=1}^{n}P(X_i \mid X_1,\ldots,X_{i-1})=\prod_{i=1}^{n}P(X_i \mid \text{Parents}(X_i)). \tag{2.21}$$

Inference: Probabilistic inference is the process of calculating the posterior distribution of variables based on existing evidence. Using some statistical tests such as chi-square or mutual information, we can find the conditional independence relationships among the nodes and use these relationships as constraints to construct a BN. In general, a BN can be used to compute the conditional probability of one node, given values assigned to the other nodes; hence, a BN can be used as a classifier that gives the posterior probability distribution of the classification node given the values of other features. We use nodes to represent dataset attributes when learning BNs from datasets [40] [77].

Separation: The idea of separation is that dependencies can be cut by observing intermediate variable(s), i.e., RVs (X,Z,Y) satisfy separation if $X \perp Y \mid Z$ (X and Y are conditionally independent given Z). We can display separation as a graphical model in which X is separated from Y by the target variable Z, as shown if Figure 2.5.

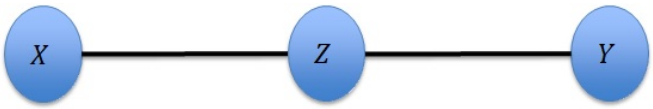

Figure 2.5: Illustration of a graphical separation.

2.7 Federated Learning System

According to ubiquitous data collection, individuals constantly produce diverse swaths of data, including location, health, and financial information. These data streams are often obtained by distributed learning techniques such as the IoT, ubiquitous sensing, edge computing, and many other distributed systems. The distributed learning could fully utilize the low-cost computing resources throughout the network and achieve comparable performance with centralized learning. Nevertheless, the leakage of the data, gradient, and even model during the updating and transmitting process in distributed learning has raised user privacy and security concerns, limiting its applications in some specific fields, such as finance and health. A typical scenario for data release privacy is a trusted server to receive a large volume of data from the clients and then share the aggregated data with other untrusted organizations for different purposes. Several privacy-preserving machine learning methods, such as federated learning (FL), are used to ensure data privacy is not violated during the training or inference process. FL is a new class of distributed machine learning techniques in which the training process is regulated without the data being centralized in cloud data centers. FL is, in other words, a paradigm that enables multiple devices to work together in the training of machine learning models without sharing potentially sensitive data. A standard FL process occurs between centralized internet-enabled servers and the distributed devices and systems connected via the internet. The FL solution protects privacy by enabling connected devices to push model parameters to centralized servers instead of pulling data from large datasets. After that, the server pushes back the updated global model to the

devices. The devices use privacy protection techniques like homomorphic encryption or differential privacy to update their models [109] [145] [158].

System Model: In general, FL entails training a global model (hosted on a central server) with data spread through many remote devices (or nodes), each with restricted bandwidth to connect back to the central server. To make matters even more complicated, data across devices is heterogeneous. The local data is the device-generated data stored and processed on the device, and the local model is the on-device model trained on the local data. To retrain the global model, intermediate changes from the local model are regularly transmitted back to the central server. For instance, assume that we have a wireless multi-user system with one base station and N users (nodes). Each user i has k_i training data samples for training its local model w_i, and K is the total number of samples over all devices. Figure 2.6 depicts the steps involved in training a model using the FL framework for wireless application [118] [82]:

Step 1: The base station distributes the initial model to all users.

Step 2: Data is collected at each node i to generate a local model w_i.

Step 3: Updated local model parameters are communicated back to the base station to generated a global model g and broadcasted it back to all users.

Step 4: Steps 2 and 3 are repeated until finding the optimal models to minimize the loss functions of the users' models.

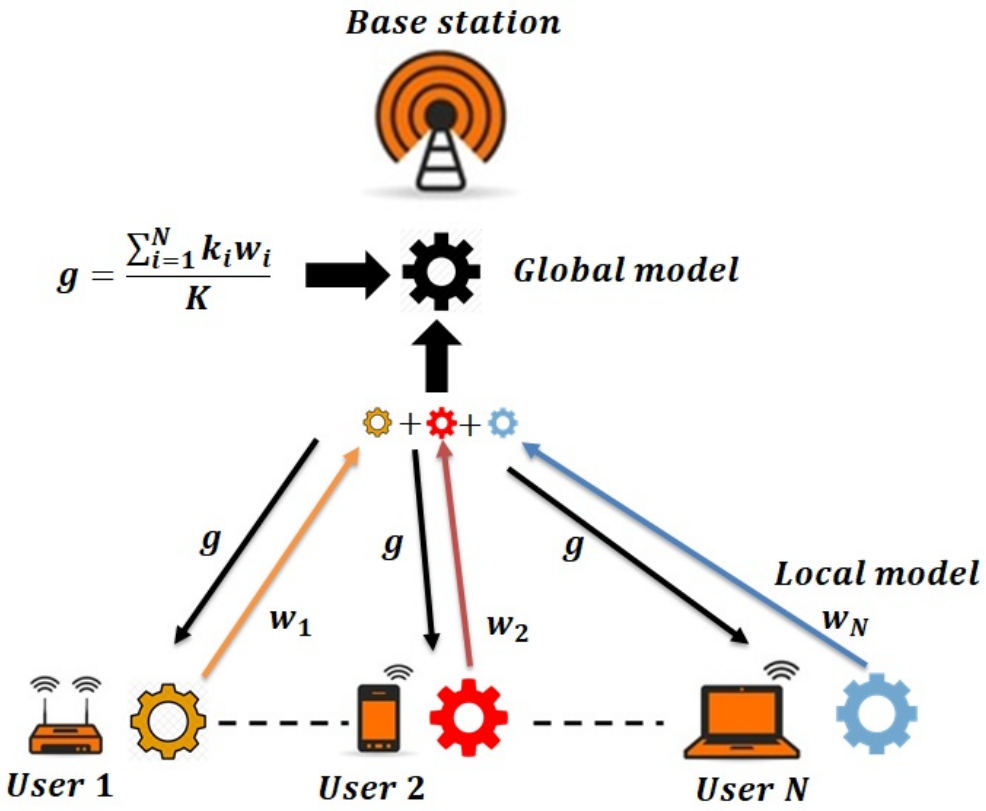

Figure 2.6: Illustration of the application of federated learning for a wireless system.

Chapter 3: Literature Review

Over the past few decades, machine learning methods have seen countless breakthroughs. As these methods are increasingly implemented, data privacy has been a growing concern. Many recent works have focused on how to make machine learning more privacy-preserving and non-discriminatory. This chapter discusses prior work on privacy-preserving data processing, focusing on standard techniques, distributed systems, and fairness in machine learning.

3.1 Common Privacy-preserving Approaches

Several formal privacy definitions, also referred to as privacy models, are provided, after which the anonymized data reveals some formal guarantees. Various privacy concepts, including k-anonymity, differential privacy, and cryptography, are discussed in the literature. We review some well-known formal privacy definitions in this section. Not all these methods are applied to deep learning, but we briefly discuss them for the sake of comprehensiveness.

K-anonymity: Several generic privacy-preserving models have been suggested to preserve data privacy by raising the amount of uncertainty, such as $k-anonymity$ [140]. It is a privacy model widely used to protect data subjects' privacy in data sharing situations and the guarantees that $k-anonymity$ can provide when used to anonymize

data. With $k-anonymity$, an original dataset is suppressed or generalized until each row is identical with at least $k-1$ other rows. At this point, the database is said to be $k-anonymous$ [55]. Any row in a $k-anonymized$ dataset has a maximum probability of $1/k$ of being re-identified [19]. Recently, several authors have recognized that $k-anonymity$ cannot prevent attribute disclosure. The notions of $l-diversity$ [106] and $t-closeness$ [97] have been proposed as a way of countering the faults of $k-anonymity$. In the $l-diversity$ table, there are at least l well-represented values for each sensitive attribute, and the idea of $t-closeness$ is that the distribution of sensitive data in every group is not too far from the distribution in the total population [46]. These approaches are only suitable for low-dimensional data because quasi-identifiers and sensitive attributes cannot be easily defined for high-dimensional data [5].

Differential Privacy: As a state-of-the-art privacy-preserving mechanism, differential privacy (DP) is a more formal way to open-source a database while keeping all individual records private by adding well-designed noise (adding random noise to the ground truth) [53]. A statistical query release is the simplest scenario: a data owner may specify counting questions, such as "how many women are in the database?" and obtain responses that have been tampered with by a small amount of random noise. DP was introduced in 2006 by Dwork, McSherry, Nissim, and Smith as [52]: A randomized algorithm M is ε-differentially private if for all S in the range of M, and for any pair of datasets D and D' differing in only one row,

$$P[M(D) \in S] \leq \exp(\varepsilon) P\left[M\left(D'\right) \in S\right], \tag{3.1}$$

where, $P[M(D) \in S]$ denotes the probability that the algorithm M outputs S. A graphical illustration of DP for privacy-preserving is shown in Figure 3.1. We control the privacy guarantee's strength by tuning the privacy parameter ε, i.e., the quantity $\ln \frac{P[M(D)\in S]}{P[M(D')\in S]}$, which is also called a privacy loss. The lower the value of the parameter, the more indistinguishable the results and, therefore, better privacy protection. The fact that a more strict privacy guarantee always demands more noise added to the data often limits its application scenarios, especially when high accuracy of learning tasks is needed [1] [154]. Implementation of DP in four domains named energy systems, transportation systems,

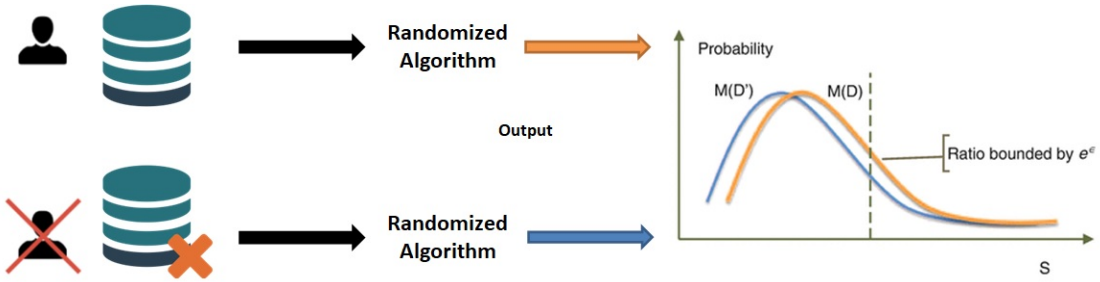

Figure 3.1: Illustration of the deferential privacy principle.

healthcare, and medical systems, and industrial systems is presented in [74]. DP can also be achieved with some distributed learning approaches by having each participant apply differentially private randomization to their data locally before sharing it with the central server [84] [57] [65]. Also, DP offers a mathematically provable guarantee of privacy security against various privacy attacks, including differencing, linkage, and reconstruction attacks [53]. A large body of DP mechanisms has been proposed for many real-world applications [1] [68].

Cryptographic Approaches: Another way for data privacy protection is to use cryp-

tographic operations to encrypt the dataset [66]. Encryption is based on complex algorithms called ciphers. The primary purpose of encryption methods is to keep sensitive information secret from others by processing readable data into a long series of random or pseudo-random ciphers. Homomorphic encryption, garbled circuits, secret sharing, and secure processors are the most widely used cryptographic techniques [123] [6]. A typical approach is to use secure multi-party computation (SMC) [160], where each party uses a set of cryptographic methods and the oblivious transfer scheme to jointly compute a function using their private data [100]. IoT devices currently use encryption protocols to protect the privacy of their dada [128] [129], e.g., the health-care industry [121] and smart home use cases [2]. In [113] and [98], deep learning algorithms are used with encryption to enhance the privacy-preserving by keeping high utility gain and maintaining low leakage of sensitive information.

Privacy-preserving Data Mining: Within the private preserving framework, privacy-preserving data mining (PPDM) techniques exist in the database community [110] [112] [93] whose goal is to prevent association of any instance in a database to a person. In addition to PPDM, many privacy-preserving machine learning (PPML) techniques [125] [169] [133] [132] [36] [1] have been proposed to deal with data beyond those in the traditional databases. Most existing PPML literature ensures that private information cannot be mined and make no assumption about the non-private information. On the other hand, our work assumes pre-specified sets of private and non-private information. Such a formulation makes the proposed data sanitization more effective and provides a flexible tradeoff between privacy and the ability to mine non-private information from the sanitized data.

Feature Selection: Several studies have investigated feature selection as a tool for obtaining privacy for sensitive data [16] [80]. The idea is not to release the complete information in the data but only selected features. Unlike these studies, the work in [157] propose a privacy mechanism to minimize the exposure of confidential information that the client may wish to keep private by zeroing out feature components in the approximate null space. The framework proposed in [56] transforms data so that the covariance between data and desired information is increased, while the covariance between data and confidential information is decreased.

Information-Theoretic Privacy: Information theorists have studied privacy-preserving notions under the rubric of information-theoretic privacy [18] [152] [107] [48] [143]. Information-theoretic privacy has predominantly been quantified by mutual information, which models how well an adversary can refine its belief about the data's private features with access to the released data. Other works use mutual information minimization between certain latent variables in various ways. The work in [43] [92] proposed a VAE based generative model which uses mutual information minimization between the latent space of a VAE and the feature labels. In [103] independence between latent variables is enforced by an additional penalty term based on the "Maximum Mean Discrepancy." Other works such as [54] [70] [64] have utilized adversarial methods in learning latent representations which are not as directly comparable to ours.

3.2 Information Bottleneck

Extracting the relevant data features were previously addressed through the IB method [143] by formalizing the ideas as an information tradeoff between accuracy and complexity and showing how to compress data while preserving its concerning target information. Given the raw data variable X and utility variable U, IB operates to get a compressed version of X while preserving U. Paper [135] shows that the IB can provide succinct representations with good generalization using smaller sample sizes than are needed to estimate the underlying distribution by proving several finite sample bounds. Gaussian IB and its applications have been studies in [37], [163]. The Gaussian lower bound to the IB curve has been found in [120]; also, they find an embedding of the data, which maximizes its "Gaussian part." The authors in [9] propose to use the variational method to optimize information bottleneck by first calculated a lower bound of the original target and then maximize the lower bound to push the results closer to the actual optimization problem's optimal solution. In [8], the authors consider a similar set-up to that of [9] and study how to improve the network's classification calibration and the ability to detect out-of-distribution data. The IB approach has found remarkable applications in supervised, unsupervised, and generative adversarial learning problems [148], deep learning [69], clustering [139], and prediction [7].

The privacy funnel (PF) estimates the privacy-utility tradeoff against adversaries when the log-loss is included in the privacy metric as well as the utility metric [107]. It assumes that the original data X is turned into sanitized data Z before disclosure. We can model the problem as finding a mapping $X \rightarrow Z$ that maximizes the mutual information

between X and Z, as long as the mutual information between Z and sensitive information S is smaller than a predefined threshold. In [115], proposes a variational approach that does not rely on adversarial training and considers the setting of continuous and high-dimensional disclosed data.

3.3 Federated Learning

Researchers have recently proposed distributed learning architectures that allow multiple users to share their data to train machine learning models. Training data is generally composed of a set of instances, each storing values for multiple attributes. Distributed datasets can result in two types of fragmentation: horizontal fragmentation, in which subsets of instances are stored in separate locations, and vertical fragmentation, in which subsets of attributes of instances are stored in different locations [145] [122]. The sensitive information leakage among parties should be considered due to information distribution and cooperation among users [166]. Privacy and confidentiality concerns limit this approach's application, preventing specific organizations such as medical institutions from fully benefiting from distributed deep learning [20] [81]. Researchers have proposed several schemes to protect data privacy under distributed machine learning architecture to overcome this challenge [15]. FL has recently risen as a promising solution under the traditional centralized approach of training artificial intelligence models. The standard formulation of FL enables multiple parties to collaboratively train a shared global model on their collective data without exposing their private training data [159] [27]. FL has found numerous practical applications where data is distributed,

and privacy is essential. For example, it has exhibited exemplary performance and robustness for healthcare systems [156] and wireless networks [38]. Alternate approaches to learning a model privately from multiple data sources based on DP has been as proposed in [127] [71] [89] [79] [96].

3.4 Fairness in Machine Learning

In fair machine learning, the central concern is to ensure that the machine learning model does not discriminate against individuals based on particular attributes (e.g., race or gender). Various machine learning techniques commonly exercise intuitively unfair behaviors, typically due to bias already encoded in the data or minimizing average error to fit majority populations. Nave approaches would require the algorithm to ignore all protected attributes such as race, color, religion, gender, or disability. This approach of fairness, which is called *fairness through unawareness*, is not achievable as there are redundant encodings, ways of predicting protected attributes from other features. Another approach is *fairness through awareness* this means we should include the sensitive attribute as a feature in the training data [73] [35] [146].

Mathematical Notation: In a supervised classification problem, we are given a labeled dataset $\mathscr{D} = \{X, S, Y\} \in \mathscr{X} \times \mathscr{S} \times \mathscr{Y}$ of n instances (also called samples or individuals): X is the set of classifier input features, $S \in \{0,1\}$ denotes a protected feature and $Y \in \{0,1\}$ represents the true label. This data is used to construct a clasefier $C : X \to [0,1]$ predicts a score $\hat{Y} = C(X)$. In this model, $S = 1$ is the protected group (favored population), while $S = 0$ is the unprotected group (disfavored population). In

a similar manner, $\hat{Y} = 1$ will indicate the preferred outcome, assuming that it represents the more desirable outcome between the two possible results [83] [170] [75].

3.4.1 Fairness Notions

It is necessary to define a metric for measuring fairness before developing a fair predictor. As a result, what is perceived as fair differs from use case to use case. Generally, fairness is treated at two different levels: group fairness and individual fairness. An individual's fairness can only be expected if we assume a non-discriminatory world and do not care about addressing discrimination, while group fairness does just the opposite [146] [146].

3.4.1.1 Group Fairness

In recent years, group fairness has received considerable attention resulting in various fairness criteria that machine learning systems should satisfy. According to group fairness measures, different protected groups must be treated similarly on average [73] [146]. Overall, there are three significant measures of group fairness:

Demographic Parity: It is one of the most well-known criteria for fairness, also called *Independence*. Demographic parity aims to sign a favorable outcome to each subgroup of a sensitive class at equal rates. Statistically, fairness definitions satisfy demographic parity if the sensitive attribute S is independent of the prediction $\hat{Y}$, i.e., the output be

independent of the sensitive attribute:

$$P(\hat{Y} \mid S) = P(\hat{Y}), \tag{3.2}$$

which for binary settings, this equals:

$$P(\hat{Y} = 1 \mid S = 0) \mid = P(\hat{Y} = 1 \mid S = 1). \tag{3.3}$$

The demographic parity, also known as *Statistical Parity Difference*, at times refers to binary classification problems with a binary sensitive attribute and considers the difference between the protected groups and unprotected groups of favorable classifications:

$$P(\hat{Y} = 1 \mid S = 0) - P(\hat{Y} = 1 \mid S = 1) \in [-1, 1]. \tag{3.4}$$

These metrics lie in the range [-1, 1], where 0 is optimal fairness. Signs of measurement demonstrate whether a group is protected or unprotected. Similarly, *Disparate Impact* is the ratio of favorable classifications for the protected group to those for the unprotected group:

$$\frac{P(\hat{Y} = 1 \mid S = 0)}{P(\hat{Y} = 1 \mid S = 1)} \in [0, \infty). \tag{3.5}$$

If the ratio is approximate to 1, it implies fairness [83] [61] [73].

Equalized Odds: According to this definition, equalized odds is met if the prediction $\hat{Y}$ is conditionally independent of the protected attribute S, given the actual value Y. This is equivalent to saying [73] [170]:

$$P(\hat{Y} \mid Y,S) = P(\hat{Y} \mid Y), \tag{3.6}$$

which for the binary setting is equivalent to:

$$P(\hat{Y} = 1 \mid Y = y, S = 0) = P(\hat{Y} = 1 \mid Y = y, S = 1) \quad \text{for } y \in \{0,1\}. \tag{3.7}$$

Equal Opportunity: Equal opportunity uses the same mathematical formulation as equalized odds but focuses on one label, $Y = 1$. In this way, we are able to [73] [170]:

$$P(\hat{Y} = 1 \mid Y = 1, S = 0) = P(\hat{Y} = 1 \mid Y = 1, S = 1). \tag{3.8}$$

3.4.1.2 Individual Fairness

As the name suggests, individual fairness is based on the individual, unlike the last three measures. Individual fairness assumes a distance metric that captures the similarity between different people and applies a Lipschitz condition to that metric to quantify this. The Lipschitz condition states that any two individuals at a certain distance $\mathrm{d}(\mathrm{x}_1, \mathrm{x}_2) \in [0,1]$ map to respective distributions $M(\mathrm{x}_1)$ and $M(\mathrm{x}_2)$ such that their statistical distances are $\mathrm{d}(\mathrm{x}_1, \mathrm{x}_2)$ at most. In other words, two individuals $M(\mathrm{x}_1)$ and $M(\mathrm{x}_2)$ who are separated by $\mathrm{d}(\mathrm{x}_1, \mathrm{x}_2) \in [0,1]$ should have indistinguishable outcomes (similar classification) [50] [146].

3.4.2 Algorithmic Interventions for Achieving Fairness

This section discusses techniques that aim to improve fairness (remove bias) from classification outcomes. These techniques can be divided into three different categories: pre-processing, in-processing, and post-processing [47] [162] [111]. Below, we discuss each of these categories separately.

Pre-processing: By pre-processing the data, the bias from the training data can be removed, so the classifier does not have to account for discrimination. As a result, the training process shows only fair examples, resulting in a fair classifier [33] [22]. Based on this pre-processed data, we expect predictions not to contain illegal or unexplainable discrimination. In [86], the uniform sampling and preferential sampling methods result from undersampling and oversampling instances of the four groups based on a binary sensitive attribute and binary labels. These methods are differentiated by the criteria used to select duplicated or discarded instances. In an adversarial approach, the objective is for minimizing the capability of an adversary task to predict the sensitive attribute from the representation [25] [13] [10] [54]. In the proposed method [165], an adversary tries to model a sensitive attribute solely based on predictions rather than representations. The goal of [3] is to achieve demographic parity; the data can be pre-processed so that the sensitive attribute is independent of utility features.

In-processing: The second strategy consists of modifying the training procedure of the classifier. The analysis of these methods constrains the behavior of learning algorithms to make sensitive features independent of target labels [32] [87]. Typically, this involves taking into account one or more fairness constraints when optimizing the clas-

sifier [23] [24] [155] [161]. Contrary to pre-processing methods, these methods are less general but can, at least in theory, result in higher utility because the optimizer can directly account for fairness. The authors of [50] address the problem by building a predictive model capable of achieving statistical parity and individual equity, which is to say that similar individuals should be treated equally. In [164], pre-processing and in-processing are combined by jointly learning a fair representation of the data and the classifier parameters.

Post-processing: Techniques used for post-processing work by taking a trained classifier might be biased and adjusting based on the protected attributes. [45]. A black-box algorithm cannot modify the training data or learn the algorithm, so post-processing is required in which the labels assigned by the black-box model are reselected after the training phase and then modified by a function [22] [23]. In order to remove unfairness, a particular decision threshold must be learned for a given score function. Due to the fact that these strategies rely on sensitive feature data at the decision time, they cannot be used if sensitive feature data is unavailable [162]. Combine the in-processing and post-processing methods in [51] by first training the classifiers (each having a different acceptance rate) for each group. Second, selecting classifiers based on group conditions that minimize loss functions. The loss function is built from the accuracy loss plus a penalty term representing the deviation from fairness.

Chapter 4: Gaussian Privacy Protector

This chapter aims to design a continuous high-dimensional data release mechanism called the GPP. Thus, GPP can prevent adversaries from mining private information from publicly released data while accurately revealing as much information about the utility as possible. We utilize variational lower bounds of mutual information approximation implemented as supervised learning using an adversarial training algorithm. Furthermore, we demonstrate that a centralized platform suited to this framework can be designed over distributed datasets.

4.1 Privacy for the Internet of Things Devices

An IoT system may generate or collect sensitive data, such as personal data, patients' privacy data in healthcare, and business data, usually transmitted and stored on cloud servers. The increasing use of intelligent devices is connected by the IoT, making data security and privacy concerns arise. Many existing solutions encrypt data before sending it to a cloud server. However, they often struggle to deal with complex attacks both during data conversion and after cipher transmission. As a result, privacy is a critical component of any IoT ecosystem and significant concern that prevents widespread adoption. This work aims to design a novel data sanitization framework called GPP, preventing adversaries from private mining information from IoT devices while ensuring

that public information can be detected using the sanitized data, as shown in Figure 4.1. Consider a scenario where users want to use their data with an untrusted application. Once provided, data may be used without the users' consent for purposes other than those initially intended, leading to serious privacy issues and loss of data sovereignty at worst. The users need to remove sensitive and application irrelevant features from their data while keeping the utility features.

Figure 4.1: High-level representation of our privacy-preserving approach for the IoT devices.

Our framework allows one to customize the privacy-preserving by focusing on what the target application considers private and utility. To illustrate our idea, we consider the following scenarios.

Security Cameras: It is possible to use crowd-sourced videos to help find exciting targets (e.g., crime suspect, lost vehicle) on demand. The requester (e.g., the police) receives images that include the target (utility information), while all other captured images (private information) of the onlookers are not disclosed.

Health Systems: We introduce the privacy-preserving issues through the practical scenario shown in Figure 4.2. In a remote health monitoring system, patients are continuously monitored by robots and IoT devices in their residential space. The system's

objective is to detect indicators or symptoms of medical conditions based on sensor measurements. The IoT devices collect the data and send them to specialists through the internet. While the sensors provide information about patients' medical conditions, they may also convey sensitive information they do not want to share. For example, motion sensor data might disclose the weight or gender or enable their re-identification. Robots also provide a video recording and chat interface with a mobile base so the remote operator can look around and even drive from place to place. Seeing the patients in a private setting is a privacy concern. The proposed GPP can be potentially applied to the robots and IoT devices sensors to filter out private data while guaranteeing that the disclosed data can be used to detect medical events with high accuracy.

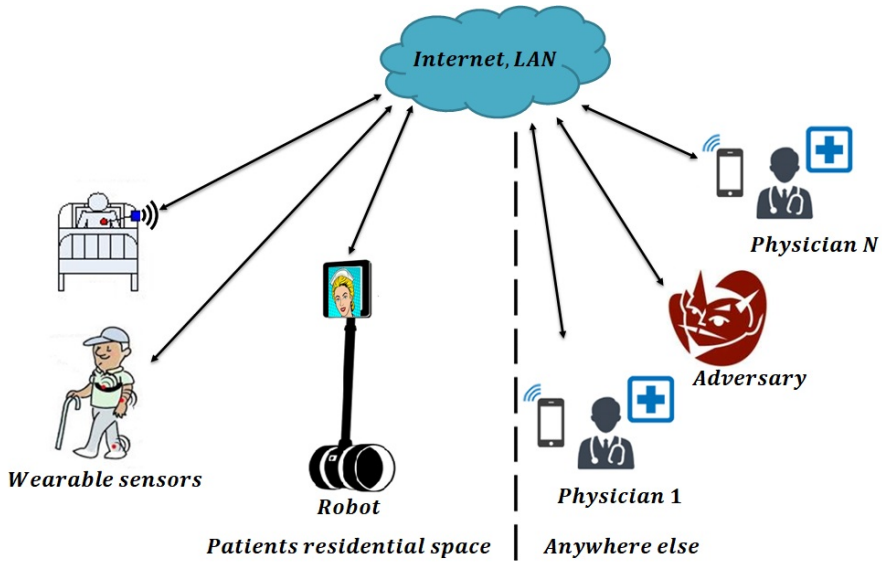

Figure 4.2: Model of a remote health monitoring system under attack.

4.2 Privacy with Compressed Data

Let $\mathbf{x}_1, \ldots, \mathbf{x}_n$ be a random sample independent and identically distributed (iid) of size n from a distribution P. We call $X = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ a dataset where $X \in \mathscr{X}$. We wish to find parameter set value $\theta \in \Theta$ achieving good average performance under a loss function $\mathcal{L}(X, \theta)$. We measure the expected performance of $\theta \in \Theta$ via the risk function

$$R(\theta) := \mathbb{E}_P[\mathcal{L}(X; \theta)], \tag{4.1}$$

where the expectation is taken over some unknown distribution P over the space $\mathscr{X}$. In the standard formulation of statistical risk minimization, a machine learning model is given n samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$, each drawn independently from P, and its goal to output an estimate $\widehat{\theta}$ that approximately minimizes the risk function R. In this dissertation, instead of providing the machine learning model with access to the samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$, however, we study the effect of giving only some compressed view $\mathbf{z}_i$ (the output of the privacy-preserving model) of each datum $\mathbf{x}_i$. With $\widehat{\theta}$ now denoting an estimator based on the sanitized samples $\mathbf{z}_i$, we explicitly quantify the rate of convergence of $R\left(\widehat{\theta}\right)$ to $\inf_{\theta \in \Theta} R(\theta)$ as a function of the number of samples n and the amount of privacy provided by $\mathbf{z}_i$.

4.3 Preliminaries

4.3.1 Problem Formulation

We consider the problem of a private data owner who aims to prepare his data for public release to maintain crucial insensitive information while keeping the privacy risk for sensitive data low. Let X, Z, U, and S be RVs distributed on finite alphabets $\mathcal{X}, \mathcal{Z}, \mathcal{U}$, and $\mathcal{S}$ respectively. Let X denote continuous high-dimensional raw data, U the utility attributes that the user is willing to reveal, S the private attributes that the user wants to hide (e.g., race and age), and Z the released data. We consider $\mathbf{x} \in \mathbb{R}^{d_x}$, $\mathbf{z} \in \mathbb{R}^{d_z}$, $\mathbf{u} \in \mathbb{R}^{d_u}$, and $\mathbf{s} \in \mathbb{R}^{d_s}$, where $d_z \ll d_x$, as instances vectors for X, Z, U, and S, respectively. The $\mathbf{s}$ and $\mathbf{u}$ can be discrete, continuous, and/or high-dimensional vectors. The goal is to design a stochastic mapping $P(Z|X)$ takes X as input and generates output Z to provide information about the utility variable U but provides relatively little knowledge of S. For instance, $\mathbf{x}^i = [x_1^i, x_2^i, ..., x_{d_x}^i]^T$ might be a face of image i, with d_x pixels, the model uses for making the prediction, $\mathbf{u}^i = [u_1^i, u_2^i, ..., u_{d_u}^i]^T$ represents labels of target features (e.g. facial expressions), $\mathbf{s}^i = [s_1^i, s_2^i, ..., s_{d_s}^i]^T$ represents sensitive features (e.g. gender, race and disability), and $\mathbf{z}^i = [z_1^i, z_2^i, ..., z_{d_z}^i]^T$ a released data that loses any information about sensitive features $\mathbf{s}$ while keeping as much other information as possible about $\mathbf{u}$.

4.3.2 Bayesian Network

From the chain rule, we can write the joint probability of the four RVs X, Z, U, S as:

$$P(X,Z,U,S) = P(X)P(Z|X)P(U|Z,X)P(S|Z,X,U). \tag{4.2}$$

If privacy is to be preserved, S and U should be independent given Z. Thus, equation (4.2) could be written as:

$$P(X,Z,U,S) = P(X)P(Z|X)P(U|Z)P(S|Z), \tag{4.3}$$

where $P(X)$ is the raw data distribution, $P(Z|X)$ is a GPP inference, $P(S|Z)$ is an adversary inference, and $P(U|Z)$ is a utility inference. Figure 4.3 illustrates a Bayesian belief network for equation (4.3).

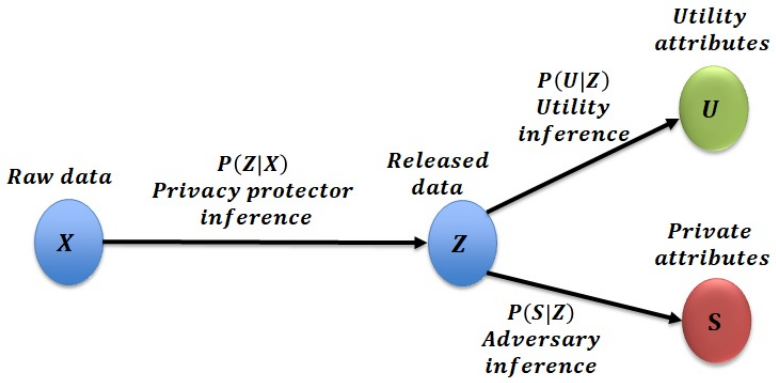

Figure 4.3: Bayesian network modeling of the GPP framework.

4.4 Proposed Approach

4.4.1 Basic Framework

Our goal is to find an optimal probabilistic mapping, $P(Z|X)$, in a way that the transformed data Z are such that an inference of sensitive information $P(S|Z)$ fails to reveal private information, whereas an inference of non-sensitive information $P(U|Z)$ generates inference that is as accurate a $P(U|X)$. This tradeoff can be formally stated as:

$$\begin{aligned} P(Z|X)^* = & \underset{P(Z|X)\in\mathbb{P}}{\arg\min}\, I(Z;S) \\ & \text{s.t. } I(Z;U) \geq \gamma, \end{aligned} \tag{4.4}$$

where γ is the minimum utility level, and the $\mathbb{P}$ is the set of all possible probabilistic mappings for GPP. The constraint in (4.4) can be written as $H(U) - H(U|Z) \geq \gamma$. As a result, it can be rewritten as:

$$\begin{aligned} P(Z|X)^* = & \underset{P(Z|X)\in\mathbb{P}}{\arg\min}\, I(Z;S) \\ & \text{s.t. } H(U|Z) \leq \acute{\gamma}, \end{aligned} \tag{4.5}$$

where $\acute{\gamma} = H(U) - \gamma$. The optimization problem in (4.5) has been studied in the context of optimal design of the privacy-preserving data release mechanism [18] [152].

As we cannot practically search over all possible probabilistic mappings $\mathbb{P}$, we consider a transformation $T_\theta(X) : X \rightarrow Z$ as a type of ANN to find the required $P(Z|X)^*$, and θ is the parameter set. The network optimizer finds the optimal parameter set θ^* by

searching the space of all the possible parameter set, Θ, as:

$$\begin{aligned} \theta^* = \arg\min_{\theta\in\Theta} & I(T_\theta(X);S) \\ \text{s.t. } & H(U|T_\theta(X)) \leq \acute{\gamma}. \end{aligned} \tag{4.6}$$

In order to solve (4.6), we have to reformulate it as an unconstrained optimization problem with a Lagrange multiplier $\beta > 0$. Therefore, we can rewrite (4.6) as follows:

$$\begin{aligned} \theta^* &= \arg\min_{\theta\in\Theta} \mathscr{L}(\theta) \\ &= \arg\min_{\theta\in\Theta} (I(T_\theta(X);S) + \beta H(U|T_\theta(X))), \end{aligned} \tag{4.7}$$

where $\mathscr{L}(\theta)$ is the objective function.

Privacy Loss: We need to discover the variational lower bound of mutual information between $T_\theta(X)$ and S

$$\begin{aligned} I(T_\theta(X);S) &= H(S) - H(S|T_\theta(X)) \\ &= H(S) + \mathbb{E}_{T_\theta(X)}\mathbb{E}_{S|T_\theta(X)}\left[\log P(S|T_\theta(X))\right]. \end{aligned} \tag{4.8}$$

In practice, the mutual information term $I(T_\theta(X);S)$ is hard to minimize directly as it requires access to the posterior $P(S|T_\theta(X)) = \frac{P(S,T_\theta(X))}{\int_S P(S,T_\theta(X))ds}$. The integration over S to calculate $P(T_\theta(X))$ in the denominator is typically intractable because this integral is unavailable in closed form. Fortunately, we can obtain a lower bound of it by defining an auxiliary posterior distribution $Q_\phi(S|T_\theta(X))$ to approximate $P(S|T_\theta(X))$. We define

$Q_\phi(S|T_\theta(X))$ as an ANN with weights and biases ϕ.

$$\begin{aligned}
&I(T_\theta(X);S)\\
&= H(S) + \mathbb{E}_{T_\theta(X)}\mathbb{E}_{S|T_\theta(X)}\left[\log\frac{Q_\phi(S|T_\theta(X))P(S|T_\theta(X))}{Q_\phi(S|T_\theta(X))}\right]\\
&= H(S) + \mathbb{E}_{T_\theta(X)}\mathbb{E}_{S|T_\theta(X)}[\log Q_\phi(S|T_\theta(X))] + \mathbb{E}_{T_\theta(X)}\mathbb{E}_{S|T_\theta(X)}\left[\log\frac{P(S|T_\theta(X))}{Q_\phi(S|T_\theta(X))}\right]\\
&= H(S) + \mathbb{E}_{S,T_\theta(X)}[\log Q_\phi(S|T_\theta(X))] + \mathbb{E}_{T_\theta(X)}KL[P(S|T_\theta(X))||Q_\phi(S|T_\theta(X))].
\end{aligned} \tag{4.9}$$

It must be a probability distribution for the KL divergence to be non-negative, therefore for the bound to hold

$$I(T_\theta(X);S) \geq H(S) + \mathbb{E}_{S,T_\theta(X)}[\log Q_\phi(S|T_\theta(X))]. \tag{4.10}$$

The bound is tight if P is exactly the same as the conditional distribution Q_ϕ. Therefore, with the constant $H(S)$ term dropped, we can write this lower bound alternatively in the following way:

$$I(T_\theta(X);S) = \max_{\phi\in\Phi}\mathbb{E}_{S,T_\theta(X)}[\log Q_\phi(S|T_\theta(X))]. \tag{4.11}$$

Utility Loss: The conditional entropy of U given $T_\theta(X)$ can be written as:

$$H(U|T_\theta(X)) = \max_{\psi\in\Psi}\mathbb{E}_{U,T_\theta(X)}[-\log Q_\psi(U|T_\theta(X))]. \tag{4.12}$$

Substituting (4.11) and (4.12) into (4.7) we get

$$\theta^* = \arg\min_{\theta\in\Theta}\max_{\phi\in\Phi,\psi\in\Psi}\mathbb{E}_{S,T_\theta(X)}[\log Q_\phi(S|T_\theta(X))] + \beta\mathbb{E}_{U,T_\theta(X)}[-\log Q_\psi(U|T_\theta(X))]. \tag{4.13}$$

We can obtain θ^* using backpropagation with stochastic gradient descent (SGD) for the multi-objective loss function. We also can determine β as a tradeoff between utility and privacy through cross-validation over the training dataset.

The minimax formulation in (4.13) is similar to the GAN objective function. It may be interpreted that GPP is trying to minimize utility loss and maximize privacy loss, whereas the adversary is trying to minimize privacy loss. This optimization problem can be practically addressed via the training of three neural networks: GPP $T_\theta(X)$ as an encoder, an adversary $Q_\phi(S|T_\theta(X))$ and a utility $Q_\psi(U|T_\theta(X))$ as classifiers. For notational simplicity, we define utility classifier $Q_\psi(U|T_\theta(X))$ as $Q_\psi(Z)$ and adversary classifier $Q_\phi(S|T_\theta(X))$ as $Q_\phi(Z)$.

The CE loss function indicates the distance between what the model believes the output distribution should be and the original distribution. It is defined as, $CE(p,q) = \mathbb{E}_p[-\log(q)]$ where p is the true distribution, and q is the estimated distribution. As observed in [48], when CE is the loss, (4.13) can be written as:

$$\min_{\theta} \left(\beta \sum_{i=1}^{d_u} \min_{\psi_i} CE(u_i, Q_{\psi_i}(\mathbf{z})) - \sum_{i=1}^{d_s} \min_{\phi_i} CE(s_i, Q_{\phi_i}(\mathbf{z}))\right), \tag{4.14}$$

which is the objective function of our approach. The objective function is close to an adversary task for small $\beta \ll 1$, and large $\beta \gg 1$ is close to a utility task.

4.4.2 From BN Structure to Deep Neural Networks

Generally, neural networks are not used to model complete probability densities. They can be interpreted as fitting a probability density function if proper activation functions are chosen and certain conditions are respected. For instance, when trained with the CE loss, it represents a conditional distribution of the label given the input [130]. In this case, for binary label variables, the adversary inference $P(S|Z)$ and utility inference $P(U|Z)$ will be considered as a Bernoulli distribution. For GPP inference $P(Z|X)$, the Z is typically referred to as a 'bottleneck' because GPP must learn an efficient compression of the data into this lower-dimensional space with Gaussian distribution. We could design Z to be the multivariate Gaussian distribution $Z \sim \mathcal{N}(\mu, \Sigma)$ by employing the reparameterization trick of [90], which first: the mean μ and the covariance $\Sigma = LL^T$ are calculated by a neural network $T_\theta(X)$. Then, generate $Z = \mu + L \odot \varepsilon$ where ε a vector of independent standard normal variables $\varepsilon \sim \mathcal{N}(o, I)$ and $\odot$ represents element-wise product. If the encoder outputs representations Z that are different from those from a standard normal distribution, it will receive a penalty in the loss. The penalty term is the KL divergence between $\mathcal{N}(\mu, L)$ and ε. So the full objective function for the GPP framework with penalty term is:

$$\min_{\theta} \left(\beta \sum_{i=1}^{d_u} \min_{\psi_i} CE(u_i, Q_{\psi_i}(Z)) - \sum_{i=1}^{d_s} \min_{\phi_i} CE(s_i, Q_{\phi_i}(Z)) + KL(\mathcal{N}(\mu, L)||\varepsilon)\right). \tag{4.15}$$

According to (4.15), Figure 4.4 illustrates the architecture of the GPP framework.

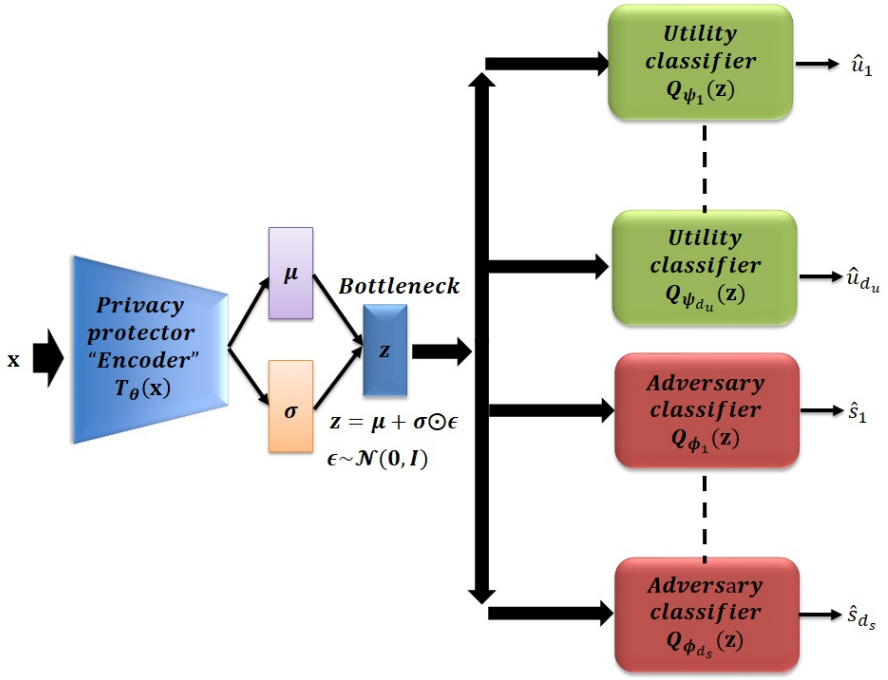

Figure 4.4: Diagram of the Gaussian privacy protector framework.

4.4.3 Training

From an optimization perspective, we aim to minimize the objective function $\mathscr{L}$, as:

$$\theta^* = \arg\min_{\theta \in \Theta} \mathscr{L}(\theta). \tag{4.16}$$

In general, the gradient-based learning is used to minimize $\mathscr{L}$, seeking to iteratively reduce the loss as:

$$\theta_{t+1} = \theta_t - \alpha \frac{\partial \mathscr{L}}{\partial \theta}. \tag{4.17}$$

where α is the learning rate [117].

Our goal is to train an encoder to produce an output leading to high inference ac-

curacy when used for utility information and randomly guessing private information. GPP maps X into an identity-obscuring low dimensional latent representation Z and updates its objective function from a pre-trained utility and adversary classifiers. Instead of just training on a single batch, as usually done in deep learning training, the utility and adversary classifiers should be trained for several batches, k, on the entire dataset to synchronize the training's convergence speed model. Algorithm 1 summarizes the training steps of the GPP.

4.5 Distributed Dataset Framework

4.5.1 Addressing Versus Challenge

While the advantages of distributed learning are well understood, individuals and organizations are still reluctant to disclose their data, such as health records, financial information, and research information, in a distributed environment [168]. For example, consider the traditional federated learning case in which t clients (e.g., mobile devices or data centers) store local datasets of private information on their respective devices and would like to cooperate to build a common learning objective. Generally, each client calculates certain abstract summary information (e.g., neural network parameters) locally and transmits it to an aggregator (central computing server). Then, the aggregator aggregates these parameters for model updating. The resulting model is then distributed to all clients, resulting in a joint representative model without directly exchanging data [159] [144]. However, although the clients only expose their abstract

norelsize 2 GPP training algorithm.

Require: b, the batch size; k, steps are used for updating $\psi_{(1,..,d_u)}$ and $\phi_{(1,..,d_s)}$ in each iteration; β, Lagrange multiplier; $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{u}^i, \mathbf{s}^i), i = 1, \ldots, n\}$, training data.

$\mathrm{T}_\theta(.), Q_{\psi_{(1,..,d_u)}}(.), Q_{\phi_{(1,..,d_s)}}(.) \leftarrow$ Random initialization
while θ has not converged **do**
  for k steps **do**
    Sample $\left\{\mathbf{x}^i, \mathbf{u}^i, \mathbf{s}^i\right\}_{i=1}^b$ a batch from the training data.
    $\left\{\mathbf{z}^i\right\}_{i=1}^b \leftarrow T_\theta(\left\{\mathbf{x}^i\right\}_{i=1}^b)$
    Perform SGD-updates for $\psi_{(1,..,d_u)}$ and $\phi_{(1,..,d_s)}$
    for $j = 1 : d_u$ **do**
      $g_{\psi_j} \leftarrow \nabla_{\psi_j} \frac{1}{b} \sum_{i=1}^{b} CE(u_j^i, Q_{\psi_j}(\mathbf{z}^i))$
      $\psi_j \leftarrow \psi_j$ - α . AdamOptimizer(ψ_j, g_{ψ_j})
    end for
    for $j = 1 : d_s$ **do**
      $g_{\phi_j} \leftarrow \nabla_{\phi_j} \frac{1}{b} \sum_{i=1}^{b} CE(s_j^i, Q_{\phi_j}(\mathbf{z}^i))$
      $\phi_j \leftarrow \phi_j$ - α . AdamOptimizer(ϕ_j, g_{ϕ_j})
    end for
  end for
  Sample $\left\{\mathbf{x}^i, \mathbf{u}^i, \mathbf{s}^i\right\}_{i=1}^b$ a batch from the training data.
  $\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b \leftarrow T_\theta(\left\{\mathbf{x}^i\right\}_{i=1}^b)$
  $\varepsilon \sim \mathcal{N}(0, I)$
  $\left\{\mathbf{z}^i\right\}_{i=1}^b \leftarrow \mu(\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b) + L(\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b) \odot \varepsilon$
  Perform SGD-updates for θ

$$g_\theta \leftarrow \nabla_\theta \frac{1}{b} \sum_{i=1}^{b} \left\{ \beta \sum_{j=1}^{d_u} CE(u_j^i, Q_{\psi_j}(\mathbf{z}^i)) - \sum_{j=1}^{d_s} CE(s_j^i, Q_{\phi_j}(\mathbf{z}^i)) + KL\left[\mathcal{N}\left(\mu(\check{\mathbf{z}}^i), L(\check{\mathbf{z}}^i)\right) || \varepsilon\right] \right\}$$

  $\theta \leftarrow \theta$ - α . AdamOptimizer(θ, g_θ)
end while

network parameters to others, adversaries can still cause privacy leakage [104] [105], as depicted in Figure 4.5. In this work, distributed GPP provides a suitable solution for protecting privacy-preserving and secure decentralized machine learning systems by training sensitive data locally and shares utility data in a distributed learning process.

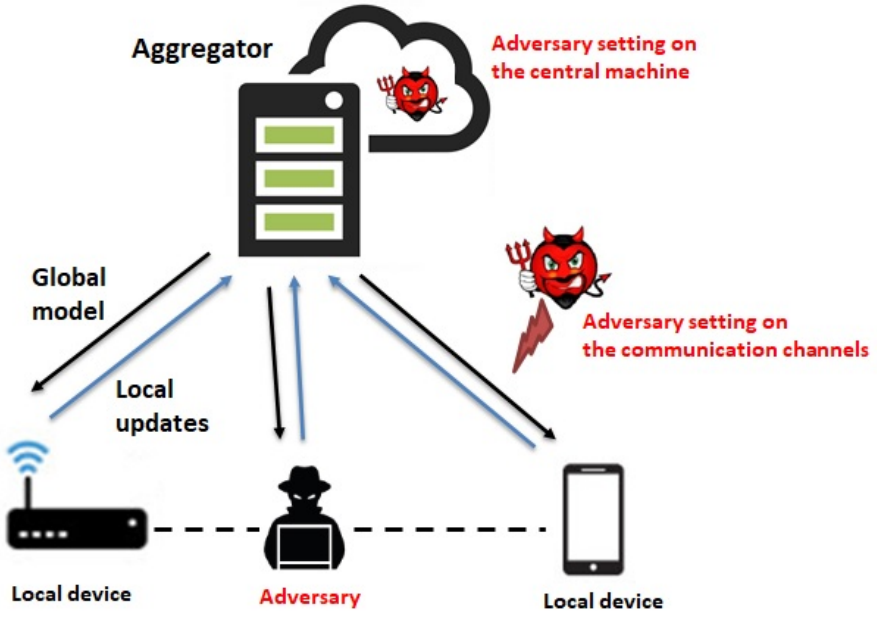

Figure 4.5: Federated learning system under attack.

4.5.2 Distributed GPP Problem Statement

For distributed GPP computation problem, let D^j denote the original dataset denoted by $D^j = \{\mathbf{x}_i^j, \mathbf{u}_i^j, \mathbf{s}_i^j\}_{i=1}^{n_j} (j = 1, \ldots, t)$ where n_j is the size of the dataset associated with GPP j and t denotes the number of GPPs. The data $\mathbf{x}$ consists of records where each record stores several attributes' values. Data of this type has a distributed nature, leading to

two common types of data partitioning: horizontal and vertical. In a horizontal partition, GPPs hold values for some records in the dataset. GPPs are vertically partitioned, containing specific attributes of a dataset. The horizontal method is explored since it is the most natural and appropriate method for most applications.

4.5.3 Distributed Learning Algorithm

The proposed distributed system consists of three kinds of deep learning networks: distributed GPP, aggregator (utility classifiers), and adversary classifiers. We formulate this problem as a learning game among three parties: (1) users using GPP to sanitize data samples, (2) a cooperative data aggregator learning a utility task using the sanitized data, and (3) an adversary learning to identify contributors using the same sanitized data. As shown in Figure 4.6, the proposed distributed system consists of t users. User i represents a GPP framework that has access to its private labels $\mathbf{s}^i$, public labels $\mathbf{u}^i$, raw data $\mathbf{x}^i$ and corresponding transfer function $T_{\theta_i}(\mathbf{x}^i)$. Each user learns an individual $T_{\theta_i}(\mathbf{x}^i)$ and private label $\mathbf{s}^i$ and sharing the common public label $\mathbf{u}^i$. In other words, this method concerned with enabling multiple GPPs to evaluate utility information jointly and returns the result to all GPPs, without sharing any information about their private inputs, i.e., each GPP has its privacy information $\mathbf{s}$ and shares the utility information $\mathbf{u}$ with other GPPs. The training of the distributed GPP is described in algorithm 3.

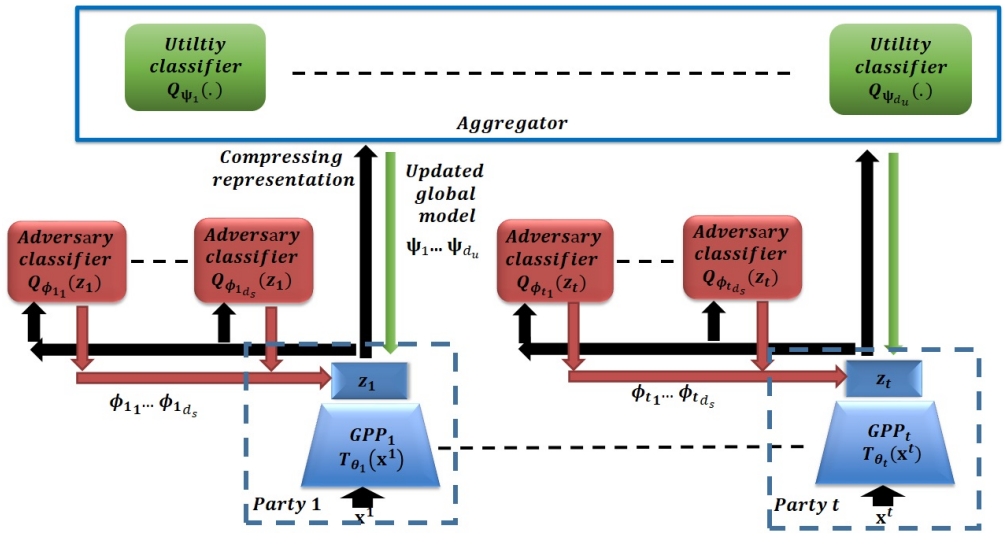

Figure 4.6: Distributed GPP diagram.

norelsize 3 Distributed GPP learning algorithm.

Require: t, number of GPPs; b, the batch size; $k_1, ..., k_t$,; hyperparameters to be used for updating $\phi_{(1,..,d_s)}$ and $\psi_{(1,..,d_u)}$ in each iteration; $\beta_1, ..., \beta_t$, Lagrange multipliers.

1: $T_{(1,..,t)}\theta_{(1,..,t)}(.), Q_{\phi_{(1,..,d_s)}}(.), Q_{\psi_{(1,..,d_u)}}(.) \leftarrow$ Random initialization
2: **while** $\theta_{(1,..,t)}$ has not converged **do**
3: **for** $itr \leftarrow 1$ to t **do**
4: Use algorithm 2 from step 3 to 15 to update the utility and adversary classifiers.
5: **end for**
6: **for** $itr \leftarrow 1$ to t **do**
7: Use algorithm 2 from step 16 to 23 to update the GPPs.
8: **end for**
9: **end while**

Chapter 5: Frameworks for Information Bottleneck Family

This chapter extends the ideas to obtain compressed representations that preserve relevant information for continuous high-dimensional data, tackled in chapter four. Our work adapts the IB principle based on private data as the core of the data to be classified; then, we consider the PF problem inspired by utility data as the central part of data to be revealed. The representation should ensure reliable reconstruction of the desired features while still preserving the data's sensitive parts. Then, we design and evaluate a distributed learning framework, a system that allows a group of IoT devices to share data securely to build a utility model.

5.1 Why Information Bottleneck

In recent years, information has become increasingly important to the point where information can be seen as an asset, just like stocks or patents. This has driven businesses to gather more information than ever before, especially with machine learning that allows businesses to use their aggregate data sets. Therefore, many practical machine learning applications have emerged that require training on sensitive data, such as financial fraud detection [153], and medical imaging [101]. The majority of privacy-preserving methods [53] [26] render unusable schemes because of filtering sensitive data. Therefore, when the often content of data is sensitive information, and the system's task is to

preserve privacy, the problem is reduced to learning public representations of the data; i.e., representations are informative of the utility data but not informative of the private information. If the level of utility information is measured with the mutual information, generating public representations is known as the IB principle [143].

Extracting the relevant data features were previously addressed through the IB method, which has become an essential element in the information-theoretic analysis deep models. Given the raw data variable X and utility variable U, IB operates to get a compressed version of X while preserving U. This section investigates the tradeoff between utility and privacy in terms of mutual information. More specifically, we aim at maintaining a certain level of utility information about the data output while minimizing all the other sensitive information. We tackle the optimization problem with a variational bound on mutual information approach [10], [12].

5.2 Principle of Data Reduction

Definition 1: (Sufficient Statistics) Let $U \in \mathcal{U}$ be an unknown parameter and $X \in \mathcal{X}$ be a random variable with conditional probability distribution function $P(X \mid U)$. Given a function $T : \mathcal{X} \to \mathcal{Z}$, the random variable $Z = T(X)$ is called a sufficient statistic for U if $\forall \mathbf{x} \in \mathcal{X}, \mathbf{u} \in \mathcal{U}$,

$$P(X = \mathrm{x} \mid U = \mathbf{u}, Z = \mathbf{z}) = P(X = \mathbf{x} \mid Z = \mathbf{z}), \tag{5.1}$$

which can be written as:

$$P(U = \mathbf{u} \mid X = \mathrm{x}) = P(U = \mathbf{u} \mid Z = \mathbf{z}). \tag{5.2}$$

Theorem 1: Let Z be a probabilistic function of X. Then, Z is a sufficient statistic for U if and only if

$$I(Z;U) = I(X;U). \tag{5.3}$$

Definition 2: (Minimal Sufficient Statistics) A sufficient statistic $T(X)$ is minimal if $T(X) = g(S(X))$ for all sufficient statistics $S(X)$.

Theorem 2: If $T(X)$ is minimal sufficient statistic, then

$$\begin{aligned} T(X) &\in \arg\min_{S} I(X, S(X)) \\ \text{s.t.} \quad & I(U, S(X)) = I(U, X). \end{aligned} \tag{5.4}$$

In other words, it is a statistic that has the smallest mutual information with X while having the most considerable mutual information with U [135], [58].

The sufficiency is related to the concept of data reduction. Suppose that $\mathbf{x}$ takes values in $\mathbb{R}^{d_x}$. If we can find a sufficient statistic $\mathbf{z}$ that takes values in $\mathbb{R}^{d_z}$, then we can reduce the original data vector $\mathbf{x}$ (whose dimension d_x is usually large) to the vector of statistics $\mathbf{z}$ (whose dimension d_z is usually much smaller) with no loss of information about the parameter $\mathbf{u}$.

5.3 Preliminaries

5.3.1 Problem Formulation

Let X, Z, and U be RVs distributed on finite alphabets $\mathcal{X}, \mathcal{Z}$, and $\mathcal{U}$ respectively. Let X denote continuous high-dimensional raw data, U the public attributes that the user is willing to reveal, and Z the released data. We consider $\mathbf{x} \in \mathbb{R}^{d_x}$, $\mathbf{z} \in \mathbb{R}^{d_z}$, and $\mathbf{u} \in \mathbb{R}^{d_u}$, where $d_z \ll d_x$, as instances vectors for X, Z, and U respectively. The $\mathbf{u}$ can be discrete, continuous, and/or high-dimensional vector. The goal is to design a stochastic mapping $P(Z|X)$ takes X as input and generates output Z to provide as much information about the utility variable U. For instance, $\mathbf{x}^i = [x_1^i, x_2^i, ..., x_{d_x}^i]^T$ might be a face of image i, with d_x pixels, the model uses for making the prediction, $\mathbf{u}^i = [u_1^i, u_2^i, ..., u_{d_u}^i]^T$ represents labels of public features (e.g. facial expressions) and $\mathbf{z}^i = [z_1^i, z_2^i, ..., z_{d_z}^i]^T$ a released data that keeping as much information as possible about $\mathbf{u}$.

5.3.2 Bayesian Model for Information Bottleneck

Consider a joint distribution over three random variables X, Z, and U such that:

$$P(X,Z,U) = P(X)P(Z|X)P(U|Z,X). \tag{5.5}$$

BNs satisfy the local Markov property, which states that a node is conditionally independent of its non-descendants given its parents. So that (5.5) can be written as:

$$P(X,Z,U) = P(X)P(Z|X)P(U|Z), \tag{5.6}$$

which represent Bayesian model that uses Bayesian inference for IB computations. Thus, we obtain the BN structure shown in Figure 5.1.

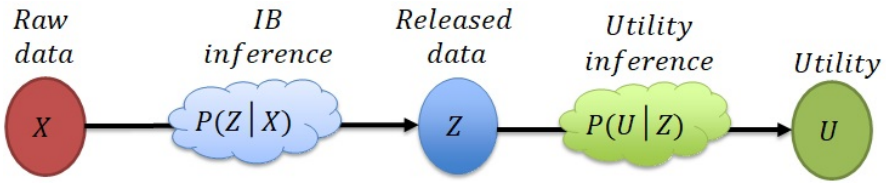

Figure 5.1: Structure of the Bayesian network for the IB framework.

5.4 Privacy-preserving Under Information Bottleneck

5.4.1 Proposed Approach

This problem was shown in [143]: what is the compressing representation of the variable X relevant for predicting another variable U?. This general problem was shown to have a natural information-theoretic formulation: Find a compressed representation of the variable X, denoted by Z, such that the mutual information between X and Z, $I(X;Z)$, is as low as possible while keeping the mutual information between Z and U, $I(Z;U)$, as high as possible. In other word, for each value $\mathbf{x} \in X$ we seek a possibly stochastic mapping (transformation) to a representative $\mathbf{z} \in Z$, characterized by a con-

ditional distribution $P(\mathbf{z} \mid \mathbf{x})$, which is the simplest representation of the data such that it can still be useful according to the measure of utility $\mathbf{u} \in U$. IB solves the following optimization problem:

$$P_{\mathbf{z}|\mathbf{x}} = \underset{P_{\mathbf{z}|\mathbf{x}} \in \mathbb{P}}{\arg\min} \quad I(\mathbf{x};\mathbf{z}) \;\text{ s.t. }\; I(\mathbf{u};\mathbf{z}) \geq \gamma, \tag{5.7}$$

where γ is the utility level, and $\mathbb{P}$ is the set of all possible probabilistic mapping for $P_{\mathbf{z}|\mathbf{x}}$. The constraint in (5.7) can be written as $H(\mathbf{u}) - H(\mathbf{u}|\mathbf{z}) \geq \gamma$. So that (5.7) can be rewritten as:

$$P_{\mathbf{z}|\mathbf{x}} = \underset{P_{\mathbf{z}|\mathbf{x}} \in \mathbb{P}}{\arg\min} \quad I(\mathbf{x};\mathbf{z}) \;\text{ s.t. }\; H(\mathbf{u}|\mathbf{z}) \leq \hat{\gamma}, \tag{5.8}$$

where $\hat{\gamma} = H(\mathbf{u}) - \gamma$. By introducing a Lagrange multiplier $\beta > 0$, we can express (5.8) as the variational minimization problem of finding

$$P_{\mathbf{z}|\mathbf{x}} = \underset{P_{\mathbf{z}|\mathbf{x}}}{\arg\min} \quad (I(\mathbf{x};\mathbf{z}) + \beta H(\mathbf{u}|\mathbf{z})). \tag{5.9}$$

As we cannot practically search over all possible probabilistic mapping $\mathbb{P}$, we consider a transform $T_\theta(\mathbf{x}) : \mathbf{x} \to \mathbf{z}$, where θ is the parameter set, is a type of ANN to approximate the required $P_{\mathbf{z}|\mathbf{x}}$ and look for the optimal parameter set through training. The network optimizer finds the optimal parameter set θ^* by searching the space of all the possible parameter set, Θ, as:

$$\theta^* = \underset{\theta \in \Theta}{\arg\min} \Big[I(\mathbf{x};\mathbf{z}) + \beta H(\mathbf{u}|\mathbf{z}) \Big]. \tag{5.10}$$

Adversary Part: In our case, we want to find a variational lower bound of mutual information between $\mathbf{x}$ and $\mathbf{z}$

$$\begin{aligned} I(\mathbf{x};\mathbf{z}) &= H(\mathbf{x}) - H(\mathbf{x}|\mathbf{z}) \\ &= H(\mathbf{x}) + \mathbb{E}_{\mathbf{z}}\mathbb{E}_{\mathbf{x}|\mathbf{z}}[\log P(\mathbf{x}|\mathbf{z})]. \end{aligned} \tag{5.11}$$

In practice, the mutual information term $I(\mathbf{x};\mathbf{z})$ is hard to minimize directly as it requires access to the posterior $P(\mathbf{x}|\mathbf{z}) = \frac{P(\mathbf{x},\mathbf{z})}{\int_x P(\mathbf{x},\mathbf{z})d\mathbf{x}}$. The marginalization over $\mathbf{x}$ to calculate $P(\mathbf{x})$ in the denominator is typically intractable because this integral is unavailable in closed form. Fortunately, we can obtain a lower bound of $I(\mathbf{x};\mathbf{z})$ by defining a parametric probability distribution $Q_\phi(\mathbf{x}|\mathbf{z})$ to approximate $P(\mathbf{x}|\mathbf{z})$. We define $Q_\phi(\mathbf{x}|\mathbf{z})$ as an ANN having weights and biases both are represented by ϕ.

$$\begin{aligned} I(\mathbf{x};\mathbf{z}) &= H(\mathbf{x}) + \mathbb{E}_{\mathbf{z}}\mathbb{E}_{\mathbf{x}|\mathbf{z}}\left[\log \frac{Q_\phi(\mathbf{x}|\mathbf{z})P(\mathbf{x}|\mathbf{z})}{Q_\phi(\mathbf{x}|\mathbf{z})}\right] \\ &= H(\mathbf{x}) + \mathbb{E}_{\mathbf{z}}\mathbb{E}_{\mathbf{x}|\mathbf{z}}[\log Q_\phi(\mathbf{x}|\mathbf{z})] + \mathbb{E}_{\mathbf{z}}\mathbb{E}_{\mathbf{x}|\mathbf{z}}\left[\log \frac{P(\mathbf{x}|\mathbf{z})}{Q_\phi(\mathbf{x}|\mathbf{z})}\right] \\ &= H(\mathbf{x}) + \mathbb{E}_{\mathbf{z},\mathbf{x}}[\log Q_\phi(\mathbf{x}|\mathbf{z})] + \mathbb{E}_{\mathbf{z},\mathbf{x}} KL[P(\mathbf{x}|\mathbf{z})||Q_\phi(\mathbf{x}|\mathbf{z})]. \end{aligned} \tag{5.12}$$

The KL divergence is a non-negative value that indicates how close two probability distributions are, therefore the lower bound to hold is:

$$I(\mathbf{x};\mathbf{z}) \geq H(\mathbf{x}) + \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log Q_\phi(\mathbf{x}|\mathbf{z})]. \tag{5.13}$$

If $P(\mathbf{x}|\mathbf{z}) = Q_\phi(\mathbf{x}|\mathbf{z})$, the KL divergence is zero and the bound is tight. So, with the constant $H(\mathbf{x})$ term dropped, we can write this lower bound alternatively in the following

way:

$$I(\mathbf{x};\mathbf{z}) = \max_{\phi \in \Phi} \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log Q_\phi(\mathbf{x}|\mathbf{z})]. \tag{5.14}$$

The max problem in equation (5.14) is the objective function of the adversary.

Utility Part: The conditional entropy of $\mathbf{u}$ given $\mathbf{z}$ can be written as:

$$H(\mathbf{u}|\mathbf{z}) = \max_{\psi \in \Psi} \mathbb{E}_{\mathbf{u},\mathbf{z}}[-\log Q_\psi(\mathbf{u}|\mathbf{z})], \tag{5.15}$$

sub (5.14) and (5.15) in (5.10) we can find the multi-objective loss function of our approach as:

$$\theta^* = \arg\min_{\theta \in \Theta} \max_{\phi \in \Phi, \psi \in \Psi} \Big[ ]\mathbb{E}_{\mathbf{x},\mathbf{z}}[\log Q_\phi(\mathbf{x}|\mathbf{z})] + \beta \mathbb{E}_{\mathbf{u},\mathbf{z}}[-\log Q_\psi(\mathbf{u}|\mathbf{z})] \Big]. \tag{5.16}$$

We obtain θ^* using backpropagation with SGD and the multi-objective loss function. This optimization problem can be practically addressed via the training of three neural networks: IB model $T_\theta(\mathbf{x})$ as an encoder, an adversary $Q_\phi(\mathbf{x}|T_\theta(\mathbf{x}))$ as a decoder, and utility $Q_\psi(\mathbf{u}|T_\theta(\mathbf{x}))$ as classifiers. To make the notations simple, we define decoder as $Q_\phi(\mathbf{z})$ and classifier as $Q_\psi(\mathbf{z})$. The equation (5.16) can be rewritten with the help of the CE loss function as:

$$\min_\theta \Big[\beta \textstyle\sum_{i=1}^{d_u} \min_{\psi_i} CE(u_i, Q_{\psi_i}(\mathbf{z})) - \min_\phi CE(\mathbf{x}, Q_\phi(\mathbf{z})) \Big], \tag{5.17}$$

which is the objective function of our approach. The objective function is close to adversary tasks for a small $\beta \ll 1$, and for a large $\beta \gg 1$ is close to utility tasks.

5.4.2 Gaussian Information Bottleneck

The Bayesian networks require prior estimate of the conditional probability distribution. The variables Z is discrete, in this case, $P(U|Z)$ can be represented as a Bernoulli distribution. The random variable Z is the released data in the IB framework, we required we require that its conditional distribution be of the form

$$P(Z \mid X) \sim \mathcal{N}(\mu(X), \sigma(X)), \tag{5.18}$$

where $\mathcal{N}(\mu, \sigma)$ is the multivariate Gaussian distribution with mean vector μ and covariance vector σ. Using the reparameterization trick in [90], that instead of mapping the latent variable Z into a fixed vector, we map it into multivariate Gaussian distribution $\mathrm{Z} = \mu + \varepsilon\sigma$ where $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. So, the final BI objective function that we need to optimize is:

$$\min_{\theta} \left[\beta \sum_{i=1}^{d_u} \min_{\psi_i} CE(u_i, Q_{\psi_i}(\mathbf{z})) - \min_{\phi} CE(\mathbf{x}, Q_{\phi}(\mathbf{z})) + KL(\mathcal{N}(\mu, \sigma) || \varepsilon) \right]. \tag{5.19}$$

The architecture of the BI framework is illustrated in Figure 5.2. The algorithmic approach that we use to solve the optimization in (5.19) are detailed in algorithm 4.

5.5 Distributed Information Bottleneck Framework

The proposed approach can be used in the FL setting, where the secure aggregation is needed. Based on an FL system model, we propose a distributed IB framework, where

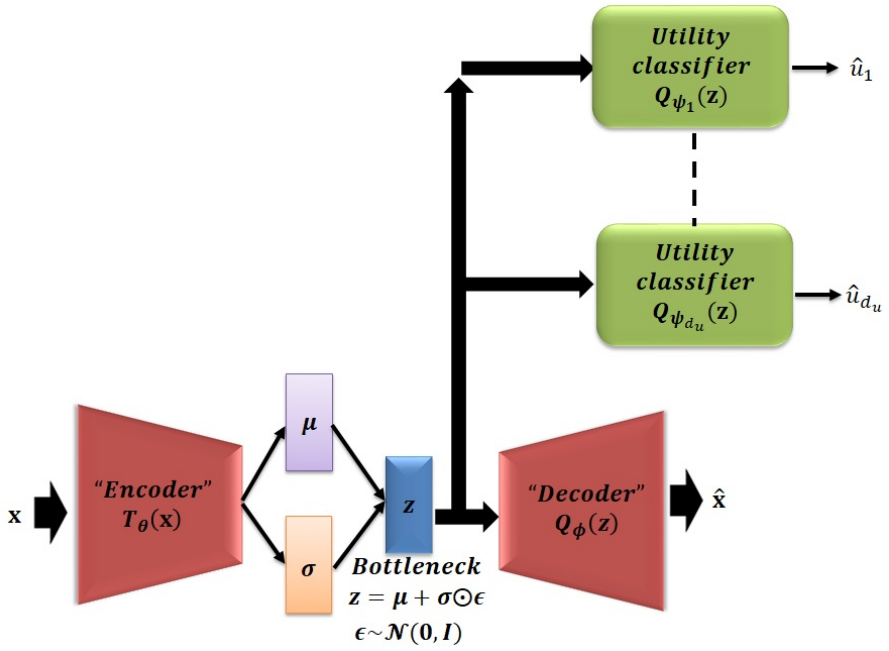

Figure 5.2: Diagram of the Gaussian IB framework.

the IoT devices communicate with each other via an aggregator, which is public classifiers, and the sensitive data of data owners are kept locally. Figure 5.3 illustrates the implementation of the distributed IB scheme, and algorithm 5 represents the distributed datasets algorithm.

norelsize 4 IB training algorithm.

Require: b, the batch size; k, steps are used for updating $\psi_{(1,..,d_u)}$ and ϕ in each iteration; β, Lagrange multiplier; $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{u}^i), i = 1, \ldots, n\}$, training data.

$\mathrm{T}_\theta(.), Q_{\psi_{(1,..,d_u)}}(.), Q_\phi \leftarrow$ Random initialization
while θ has not converged **do**
  for k steps **do**
    Sample $\left\{\mathbf{x}^i, \mathbf{u}^i\right\}_{i=1}^b$ a batch from the training data.
    $\left\{\mathbf{z}^i\right\}_{i=1}^b \leftarrow T_\theta(\left\{\mathbf{x}^i\right\}_{i=1}^b)$
    Perform SGD-updates for $\psi_{(1,..,d_u)}$ and ϕ
    for $j = 1 : d_u$ **do**
      $g_{\psi_j} \leftarrow \nabla_{\psi_j} \frac{1}{b} \sum_{i=1}^{b} CE(u_j^i, Q_{\psi_j}(\mathbf{z}^i))$
      $\psi_j \leftarrow \psi_j$ - α . AdamOptimizer(ψ_j, g_{ψ_j})
    end for
    $g_\phi \leftarrow \nabla_\phi \frac{1}{b} \sum_{i=1}^{b} CE(\mathbf{x}^i, Q_\phi(\mathbf{z}^i))$
    $\phi \leftarrow \phi$ - α . AdamOptimizer(ϕ, g_ϕ)
  end for
  Sample $\left\{\mathbf{x}^i, \mathbf{u}^i\right\}_{i=1}^b$ a batch from the training data.
  $\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b \leftarrow T_\theta(\left\{\mathbf{x}^i\right\}_{i=1}^b)$
  $\varepsilon \sim \mathcal{N}(0, I)$
  $\left\{\mathbf{z}^i\right\}_{i=1}^b \leftarrow \mu(\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b) + \sigma(\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b) \odot \varepsilon$
  Perform SGD-updates for θ

$$g_\theta \leftarrow \nabla_\theta \frac{1}{b} \sum_{i=1}^{b} \left\{ \beta \sum_{j=1}^{d_u} CE(u_j^i, Q_{\psi_j}(\mathbf{z}^i)) - CE(\mathbf{x}^i, Q_\phi(\mathbf{z}^i)) + KL\left[\mathcal{N}\left(\mu(\check{\mathbf{z}}^i), \sigma(\check{\mathbf{z}}^i)\right) || \varepsilon\right] \right\}$$

  $\theta \leftarrow \theta$ - α . AdamOptimizer(θ, g_θ)
end while

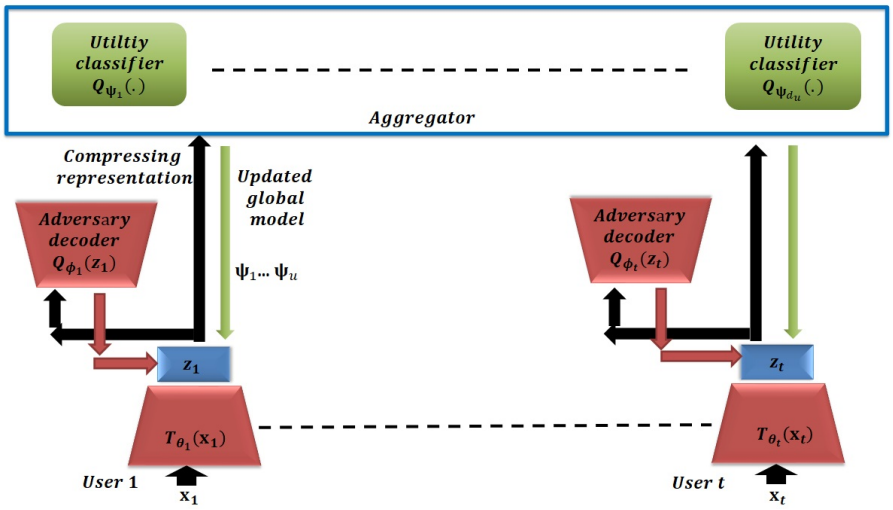

Figure 5.3: Architecture of distributed IB framework.

norelsize 5 Distributed IB learning algorithm.

Require: $D^m = \{\mathbf{x}_m^i, \mathbf{u}_m^i\}_{i=1}^n (m = 1, \ldots, t)$, raining data, where n is the size of the dataset associated with IB framework m, and t denotes the number of IB frameworks; b, the batch size; $k_1, ..., k_t$,; hyperparameters to be used for updating $\phi_{(1,..,t)}$ and $\psi_{(1,..,d_u)}$ in each iteration; $\beta_1, ..., \beta_t$, Lagrange multipliers.

1: $T_{\theta_{(1,..,t)}}(.), Q_{\phi_{(1,..,t)}}(.), Q_{\psi_{(1,..,d_u)}}(.) \leftarrow$ Random initialization
2: **while** $\theta_{(1,..,t)}$ has not converged **do**
3: **for** $itr \leftarrow 1$ to t **do**
4: Use algorithm 1 from step 3 to 13 to update the utility classifiers and adversary decoders.
5: **end for**
6: **for** $itr \leftarrow 1$ to t **do**
7: Use algorithm 5 from step 14 to 20 to update the IB frameworks.
8: **end for**
9: **end while**

5.6 Privacy Funnel

5.6.1 Problem Formulation

Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ denote the set of the raw data. Similarly, we adopt $S = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\}$ to denote the set of private labels that the adversary classifier aims to infer, and $\mathbf{s}_i = \{s_1, \ldots, s_{d_s}\}$ denotes the corresponding labels of each private class. The private label can be a discrete, continuous, and/or high-dimensional vector. Let $P_{Z|X}$, which is a probabilistic privacy mapping converting X into Z, a disclosed data. In a privacy-preserving data release, the goal is to find a probabilistic mapping $P_{Z|X}$ such that releasing Z will not violate the privacy of individuals. Without privacy in mind, we could think of this as a feature transformation. This framework is specified by joint probability function $P_{X,Z,S} = P_X P_{Z|X} P_{S|Z,X}$. For privacy-preserving we want S to be independent of X for a given Z. So that the joint probability of our approach can be factorized into $P_{X,Z,S} = P_X P_{Z|X} P_{S|Z}$, where P_X is a raw data, $P_{Z|X}$ is PF inference, and $P_{S|Z}$ is an adversary inference, which forms a BN as shown in Figure 5.4.

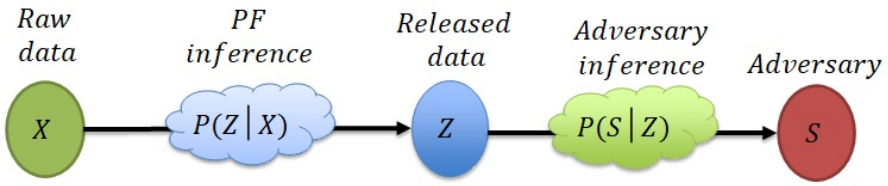

Figure 5.4: Structure of the Bayesian network for the PF framework

5.6.2 Proposed Approach

The PF considers there is a tradeoff between the information that the user shares about $\mathbf{x}$ and the information that the user keeps private about $\mathbf{s}$. Let us consider we pass $\mathbf{x}$ through a probabilistic mapping $P_{\mathbf{z}|\mathbf{x}}$ to reveal $\mathbf{z}$ to the public. The purpose of this mapping is to make $\mathbf{z}$ informative about $\mathbf{x}$ and uninformative about $\mathbf{s}$. In other words, we want to design $P_{\mathbf{z}|\mathbf{x}}$ to maximize the amount of information $I(\mathbf{x};\mathbf{z})$ that the user discloses about the public information, $\mathbf{x}$, while minimizing the collateral information about the private variable $\mathbf{s}$ measured by $I(\mathbf{s};\mathbf{z})$. The tradeoff between disclosure and privacy in the design of PF is represented by the following optimization:

$$\begin{aligned} P_{\mathbf{z}|\mathbf{x}} = \underset{P_{\mathbf{z}|\mathbf{x}}\in\mathbb{P}}{\arg\min} \quad & I(\mathbf{s};\mathbf{z}) \\ \text{s.t.} \quad & I(\mathbf{x};\mathbf{z}) \geq \gamma, \end{aligned} \tag{5.20}$$

where γ is the disclosure level, and $\mathbb{P}$ is the set of all possible probabilistic mapping for PF. The constraint in (5.20) can be written as $H(\mathbf{z}) - H(\mathbf{z}|\mathbf{x}) \geq \gamma$. So that (5.20) can be rewritten as:

$$\begin{aligned} P_{\mathbf{z}|\mathbf{x}} = \underset{P_{\mathbf{z}|\mathbf{x}}\in\mathbb{P}}{\arg\min} \quad & I(\mathbf{s};\mathbf{z}) \\ \text{s.t.} \quad & H(\mathbf{z}|\mathbf{x}) \leq \grave{\gamma}, \end{aligned} \tag{5.21}$$

where $\grave{\gamma} = H(\mathbf{z}) - \gamma$. By introducing a Lagrange multiplier $\beta > 0$, we can express (5.21) as the variational minimization problem of finding

$$P_{\mathbf{z}|\mathbf{x}} = \underset{P_{\mathbf{z}|\mathbf{x}}\in\mathbb{P}}{\arg\min} \quad \Big[I(\mathbf{s};\mathbf{z}) + \beta H(\mathbf{z}|\mathbf{x}) \Big]. \tag{5.22}$$

As we cannot practically search over all possible probabilistic mapping $\mathbb{P}$, we consider a transform $T_\theta(\mathbf{x}) : \mathbf{x} \to \mathbf{z}$, where θ is the parameter set, is a type of ANN to approximate the required $P_{\mathbf{z}|\mathbf{x}}$ and look for the optimal parameter set through training. The network optimizer finds the optimal parameter set θ^* by searching the space of all the possible parameter set, Θ, as:

$$\theta^* = \arg\min_{\theta \in \Theta} \Big[I(\mathbf{s};\mathbf{z}) + \beta H(\mathbf{z}|\mathbf{x}) \Big]. \tag{5.23}$$

First term of (5.23): We will determine a variational lower bound for mutual information between $\mathbf{z}$ and $\mathbf{s}$

$$\begin{aligned} I(\mathbf{s};\mathbf{z}) &= H(\mathbf{s}) - H(\mathbf{s}|\mathbf{z}) \\ &= H(\mathbf{s}) + \mathbb{E}_{\mathbf{z}}\mathbb{E}_{\mathbf{s}|\mathbf{z}}[\log P(\mathbf{s}|\mathbf{z})]. \end{aligned} \tag{5.24}$$

In practice, the mutual information term $I(\mathbf{s};\mathbf{z})$ is hard to minimize directly as it requires access to the posterior $P(\mathbf{s}|\mathbf{z}) = \frac{P(\mathbf{s},\mathbf{z}}{\int_{\mathbf{s}} P(\mathbf{s},\mathbf{z}) d\mathbf{s}}$. The marginalization over $\mathbf{s}$ to calculate $P(\mathbf{z})$ in the denominator is typically intractable because this integral is unavailable in closed form. Fortunately, we can obtain a lower bound of $I(\mathbf{s};\mathbf{z})$ by defining a parametric probability distribution $Q_\phi(\mathbf{s}|\mathbf{z})$ to approximate $P(\mathbf{s}|\mathbf{z})$. We define $Q_\phi(\mathbf{s}|\mathbf{z})$ as an ANN having weights and biases both are represented by ϕ.

$$\begin{aligned} I(\mathbf{s};\mathbf{z}) &= H(\mathbf{s}) + \mathbb{E}_{\mathbf{z}}\mathbb{E}_{\mathbf{s}|\mathbf{z}}\left[\log \frac{Q_\phi(\mathbf{s}|\mathbf{z})P(\mathbf{s}|\mathbf{z})}{Q_\phi(\mathbf{s}|\mathbf{z})}\right] \\ &= H(\mathbf{s}) + \mathbb{E}_{\mathbf{z}}\mathbb{E}_{\mathbf{s}|\mathbf{z}}[\log Q_\phi(\mathbf{s}|\mathbf{z})] + \mathbb{E}_{\mathbf{z}}\mathbb{E}_{\mathbf{s}|\mathbf{z}}\left[\log \frac{P(\mathbf{s}|\mathbf{z})}{Q_\phi(\mathbf{s}|\mathbf{z})}\right] \\ &= H(\mathbf{s}) + \mathbb{E}_{\mathbf{s},\mathbf{z}}[\log Q_\phi(\mathbf{s}|\mathbf{z})] + \mathbb{E}_{\mathbf{z}} KL[P(\mathbf{s}|\mathbf{z})||Q_\phi(\mathbf{s}|\mathbf{z})]. \end{aligned} \tag{5.25}$$

The KL divergence is a non-negative value that indicates how close two probability distributions are, therefore the lower bound to hold is:

$$I(\mathbf{s};\mathbf{z}) \geq H(\mathbf{s}) + \mathbb{E}_{\mathbf{s},\mathbf{z}}[\log Q_\phi(\mathbf{s}|\mathbf{z})]. \tag{5.26}$$

If $P(\mathbf{s}|\mathbf{z}) = Q_\phi(\mathbf{s}|\mathbf{z})$, the KL divergence is zero and the bound is tight. So, with the constant $H(\mathbf{s})$ term dropped, we can write this lower bound alternatively in the following way:

$$I(\mathbf{s};\mathbf{z}) = \max_{\phi\in\Phi} \mathbb{E}_{\mathbf{s},\mathbf{z}}[\log Q_\phi(\mathbf{s}|\mathbf{z})]. \tag{5.27}$$

The max problem in equation (5.27) is the objective function of the adversary.

Second term of (5.23): The conditional entropy of $\mathbf{z}$ given $\mathbf{x}$ can be written as:

$$H(\mathbf{z}|\mathbf{x}) = \max_{\psi\in\Psi} \mathbb{E}_{\mathbf{z},\mathbf{x}}[-\log Q_\psi(\mathbf{z}|\mathbf{x})], \tag{5.28}$$

sub (5.27) and (5.28) in (5.23) we can find the multi-objective loss function of our approach as:

$$\theta^* = \arg\min_{\theta\in\Theta} \max_{\phi\in\Phi,\psi\in\Psi} \Big[\mathbb{E}_{\mathbf{s},\mathbf{z}}[\log Q_\phi(\mathbf{s}|\mathbf{z})] + \beta\mathbb{E}_{\mathbf{z},\mathbf{x}}[-\log Q_\psi(\mathbf{z}|\mathbf{x})]\Big]. \tag{5.29}$$

We obtain θ^* using backpropagation with SGD and the multi-objective loss function. Our minimax formulation in (5.29) is similar to a GAN objective function. It can be interpreted as PF wants to maximize the privacy loss, while the adversary is trying to minimize privacy loss. This optimization problem can be practically addressed via the

training of three neural networks: encoder $T_\theta(\mathbf{x})$ as PF, decoder $Q_\psi(\mathbf{z}|\mathbf{x})$ as utility, and an adversary $Q_\phi(\mathbf{s}|\hat{\mathbf{x}})$ as classifiers. The equation (5.29) can be rewritten with help of the CE loss function as:

$$\min_{\theta} \left[\beta \min_{\psi} CE(\mathbf{x}, \hat{\mathbf{x}})) - \sum_{i=1}^{d_s} \min_{\phi_i} CE(s_i, Q_\phi(\mathbf{z})) \right], \tag{5.30}$$

which is the objective function of our approach. The objective function is close to adversary tasks for a small $\beta \ll 1$, and for a large $\beta \gg 1$ is close to utility tasks. In the same manner, as in subsection 5.4.2, we can obtain the Gaussian PF as:

$$\min_{\theta} \left[\beta \min_{\psi} CE(\mathbf{x}, \hat{\mathbf{x}})) - \sum_{i=1}^{d_s} \min_{\phi_i} CE(s_i, Q_\phi(\mathbf{z})) + KL(\mathcal{N}(\mu, \sigma)||\varepsilon) \right]. \tag{5.31}$$

The solution to (5.31) will defined as an optimal PF for privacy-utility tradeoffs in terms of autoencoder as PF framework and adversary classifiers. The architecture of the PF framework is illustrated in Figure 5.5. The algorithmic approach that we use to solve the optimization in (5.31) are detailed in algorithm 6.

norelsize 6 PF training algorithm.

Require: Require: b, the batch size. k, a hyperparameter to be used for updating $\phi_{(1,..,m)}$ in each iteration. β, Lagrange multiplier. $\mathcal{D} = \{(\mathbf{x}^i, \mathbf{s}^i), i = 1, \ldots, n\}$, training data.

$\mathrm{T}_\theta(.), Q_{\phi_{(1,..,d_s)}}(.), Q_\psi \leftarrow$ Random initialization
while θ has not converged **do**
  for k steps **do**
    Sample $\left\{\mathbf{x}^i, \mathbf{s}^i\right\}_{i=1}^b$ a batch from the training data.
    $\left\{\mathbf{z}^i\right\}_{i=1}^b \leftarrow T_\theta(\left\{\mathbf{x}^i\right\}_{i=1}^b)$
    Perform SGD-updates for $\phi_{(1,..,d_s)}$ and ψ
    for $j = 1 : d_s$ **do**
      $g_{\phi_j} \leftarrow \nabla_{\phi_j} \frac{1}{b} \sum_{i=1}^{b} CE(s_j^i, Q_{\phi_j}(\mathbf{z}^i))$
      $\phi_j \leftarrow \phi_j$ - α . AdamOptimizer(ϕ_j, g_{ϕ_j})
    end for
    $g_\psi \leftarrow \nabla_\psi \frac{1}{b} \sum_{i=1}^{b} CE(\mathbf{x}^i, Q_\psi(\mathbf{z}^i))$
    $\psi \leftarrow \psi$ - α . AdamOptimizer(ψ, g_ψ)
  end for
  Sample $\left\{\mathbf{x}^i, \mathbf{s}^i\right\}_{i=1}^b$ a batch from the training data.
  $\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b \leftarrow T_\theta(\left\{\mathbf{x}^i\right\}_{i=1}^b)$
  $\varepsilon \sim \mathcal{N}(0, I)$
  $\left\{\mathbf{z}^i\right\}_{i=1}^b \leftarrow \mu(\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b) + \sigma(\left\{\check{\mathbf{z}}^i\right\}_{i=1}^b) \odot \varepsilon$
  Perform SGD-updates for θ

$$g_\theta \leftarrow \nabla_\theta \frac{1}{b} \sum_{i=1}^{b} \left\{ \beta CE(\mathbf{x}^i, Q_\psi(\mathbf{z}^i)) - \sum_{j=1}^{d_s} CE(s_j^i, Q_{\phi_j}(\mathbf{z}^i)) + KL\left[\mathcal{N}\left(\mu(\check{\mathbf{z}}^i), \sigma(\check{\mathbf{z}}^i)\right) || \varepsilon\right] \right\}$$

  $\theta \leftarrow \theta$ - α . AdamOptimizer(θ, g_θ)
end while

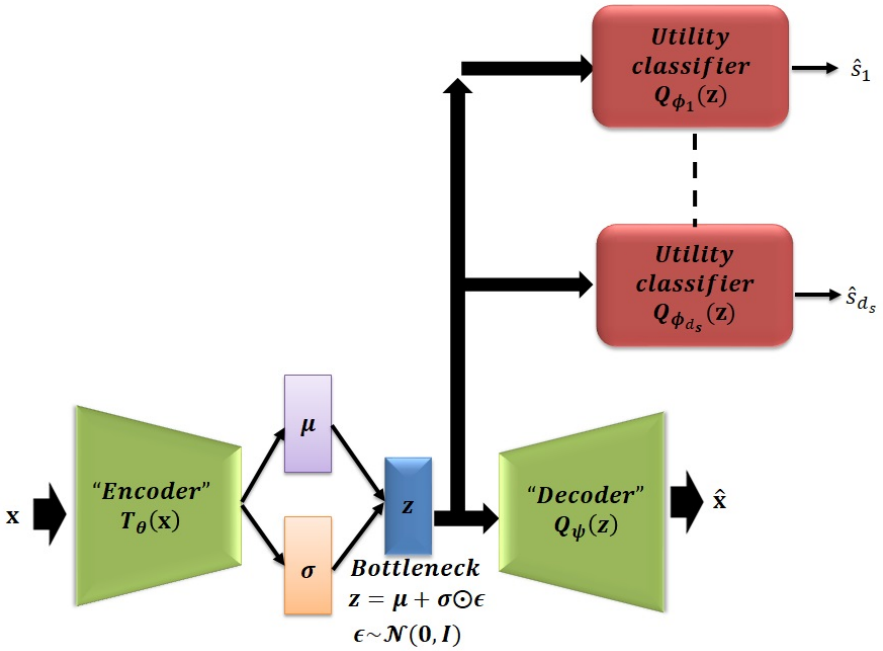

Figure 5.5: Diagram of the Gaussian PF framework.

Chapter 6: Fairness-Aware Machine Learning

Problems arising from the presence of sensitive information are not necessarily privacy-related. Most classification tasks face the challenges of achieving utility through classification while also preventing discrimination. We identify fairness in classification as a challenging concern in this chapter and conduct a formal study. Ideally, fairness should prevent discrimination against protected group members in classification systems. In this chapter, we focus on fairness on a group level and fairness through awareness. We develop a minimax adversarial framework, called the features protector (FP) framework, to achieve the information-theoretical tradeoff between minimizing distortion of target data and ensuring that sensitive features have similar distributions.

6.1 Sources of Unfairness

Classification aims to develop a reasonable value for an unknown variable U based on an observed variable X. For instance, we might use various characteristics such as credit history and salary to predict whether a loan applicant would pay back the loan. Although it has shown promise in terms of enhanced decision accuracy, its results have also been shown to be discriminatory to people from certain social classes in certain situations (e.g., women, blacks). Researchers and practitioners from various backgrounds have emphasized the ethical and legal issues raised by the use of machine-learned mod-

els and the potential for such systems to discriminate against particular demographic groups due to algorithmic decision-making biases [47] [162]. For instance, facial recognition performs extremely poorly for women with darker skin [28], a recruitment tool for STEM jobs assumes men are more skilled and biased towards women [91], and pedestrian detection accuracy in self-driving cars is very low a subgroup of people [150]. In order to address the above challenges, features protector (FP) framework is covering recent progress to tackle algorithmic fairness problems of deep learning from the decision-making perspective.

6.2 Preliminaries

6.2.1 Problem Formulation

Let $X = \left[\mathbf{x}^1, \mathbf{x}^2 ..., \mathbf{x}^n\right] \in \mathbb{R}^{d_x \times n}$ a raw data matrix, (i.e., is a collection of n data vectors as columns, each with d_x features), $\hat{X} = \left[\hat{\mathbf{x}}^1, \hat{\mathbf{x}}^2 ..., \hat{\mathbf{x}}^n\right] \in \mathbb{R}^{d_x \times n}$ a released data matrix, $U = \left[\mathbf{u}^1, \mathbf{u}^2 ..., \mathbf{u}^n\right] \in \mathbb{R}^{d_u \times n}$ a matrix of target (non-sensitive) features (labels) we want to predict, and $S = \left[\mathbf{s}^1, \mathbf{s}^2 ..., \mathbf{s}^n\right] \in \mathbb{R}^{d_S \times n}$ a matrix of sensitive demographic features. We will index observed individuals by superscript, e.g., $\left(\mathbf{x}^i, \mathbf{s}^i, \mathbf{u}^i\right)$ is the i_{th} individual in a training dataset. For instance, $\mathbf{x}^i = [x_1^i, x_2^i, ..., x_{d_x}^i]^T$ might be a face image, with d_x pixels, the model uses for making the prediction, $\mathbf{u}^i = [u_1^i, u_2^i, ..., u_{d_u}^i]^T$ represents labels of target features (e.g. facial expressions), $\mathbf{s}^i = [s_1^i, s_2^i, ..., s_{d_s}^i]^T$ represents sensitive features (e.g. gender, race and disability), and $\hat{\mathbf{x}}^i = [\hat{x}_1^i, \hat{x}_2^i, ..., \hat{x}_{d_x}^i]^T$ a released image that loses any information about sensitive features $\mathbf{s}$ while keeping as much other information

as possible about $\mathbf{y}$. The goal of our approach is to find a features mapping $g_\theta(\mathbf{x})$ (e.g., a neural network with parameters θ that minimizes a loss function $\mathscr{L}(\theta)$ such as the CE), computation performed as: $\hat{\mathbf{x}} = g_\theta(\mathbf{x})$, such that the protected features vector $\hat{\mathbf{x}}$ can be used to accurately predict $\mathbf{u}$ (equal opportunity), but will typically random guess if used to predict $\mathbf{s}$ (demographic parity). The target labels $\mathbf{u}$ and sensitive features $\mathbf{s}$ could be discrete, continuous, and/or high-dimensional data.

6.2.2 Fairness Notions

Technically, association-based notions measure the association between the sensitive feature and the utility feature and are widely used to assess the strength of discrimination and the fairness of fairness judgments. In the context of fairness in classification, a predictor $Q(\hat{\mathbf{x}}) = \hat{u}$ can be consider fair if:

(1) *Demographic Parity* which requires prediction $\hat{u}$ do not depend on the sensitive features s [31]

$$P(\hat{u} = 1|s = 1) = P(\hat{u} = 1|s = 0). \tag{6.1}$$

The goal here is, the released data $\hat{\mathbf{x}}$ would be uncorrelated with s. Thereby, the advantaged outcome $u = 1$ is independent of sensitive features s. However, the demographic parity has drawback if $P(u = 1|s = 1) \neq P(u = 1|s = 0)$. As a result of that, we can present an equivalent standard depend on the true label u.

(2) *Equal Opportunity* which requires predictions $\hat{u}$ to be conditional independence of

the sensitive feature s given $u = 1$ [161]

$$P(\hat{u} = 1|s = 1, u = 1) = P(\hat{u} = 1|s = 0, u = 1), \tag{6.2}$$

and for given $u = 0$

$$P(\hat{u} = 0|s = 1, u = 0) = P(\hat{u} = 0|s = 0, u = 0). \tag{6.3}$$

If we achieve these, the data released $\hat{\mathbf{x}}$ should be uncorrelated with sensitive feature s when the predicted label $\hat{u}$ equals to the true label u.

6.3 Proposed Approach

6.3.1 Formulating the Goal

Our perspective model from viewing the fairness involves three entities: FP, adversary tasks, and target tasks. In order to satisfy the demographic parity, the adversary tasks attempts to predict sensitive information denoted by $\mathbf{s}$ using the outcome data $\hat{\mathbf{x}}$ from FP as an input. Meanwhile, FP preventing the adversary tasks from predicting the sensitive features $\mathbf{s}$ accurately. To achieve the equality opportunity, the target tasks attempts to predict desired information denoted by $\mathbf{u}$ from $\hat{\mathbf{x}}$ with high accuracy.

Achieving Fairness: Using the information-theoretic fairness model, we formulate the problem of finding an optimal FP as follows. Let $\mathbf{x}$ a RV denoting the feature vector consisting of raw data, $\mathbf{u}$, and $\mathbf{s}$ two RVs of target and sensitive events, respectively. The

goal of our approach is to find a transform function $g_\theta(\mathbf{x})$ that:

$$\textit{Demographic Parity}, \quad I(g_\theta(\mathbf{x});\mathbf{s}) \approx 0 \tag{6.4}$$

$$\textit{Equality Opportunity}, \quad I(\mathbf{x},\mathbf{u}) \approx I(g_\theta(\mathbf{x});\mathbf{u}). \tag{6.5}$$

Equation (6.4) ensures that any inference algorithm on the sensitive event using the transformed data is similar to a random guess. In contrast, equation (6.5) allows a target classifier to accurately detect the target event using the transformed data. It is important to note that the performance of the classifier using a feature vector $g_\theta(\mathbf{x})$ to identify events $\mathbf{u}$ and $\mathbf{s}$ depend fundamentally on their mutual information $I[g_\theta(\mathbf{x}),\mathbf{u}]$ and $I[g_\theta(\mathbf{x}),\mathbf{s}]$ respectively. Large $I[g_\theta(\mathbf{x}),\mathbf{u}]$ implies more information is shared between $g_\theta(\mathbf{x})$ and $\mathbf{u}$. Therefore, a good classifier should have higher classification accuracy with larger $I[g_\theta(\mathbf{x}),\mathbf{u}]$. In contrast, when $I[g_\theta(\mathbf{x}),\mathbf{s}] \approx 0$, no classifier will be better than a random guess [78] [21]. Therefore, sub $g_\theta(\mathbf{x})$ by $\hat{\mathbf{x}}$, we can rewrite (6.4) and (6.5) as:

$$\textit{Demographic Parity}, \quad \min_\theta I(\hat{\mathbf{x}};\mathbf{s}) \tag{6.6}$$

$$\textit{Equality Opportunity}, \quad \max_\theta I(\hat{\mathbf{x}};\mathbf{u}). \tag{6.7}$$

Framework Objective: The overall objective function of the proposed framework, stated as follows:

$$\min_\theta \mathscr{L}_{FP}(\theta) = \min_\theta \Big[I(\hat{\mathbf{x}};\mathbf{s}) - \lambda I(\hat{\mathbf{x}};\mathbf{u}) \Big], \tag{6.8}$$

where $\lambda > 0$ determines the relative importance of target versus adversary tasks.

6.3.2 Features Protector Framework

A naive approach to solve the problem in equation (6.8) would be first estimate the joint probability mass function $P(\mathbf{x},\mathbf{u},\mathbf{s})$. Based on the estimated $P(\mathbf{x},\mathbf{u},\mathbf{s})$ analytically write down the mutual information quantities that allow for various optimization algorithms to find a good solution. For low-dimensional data, this naive approach might work. However, in many applications, we have a high-dimensional $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{s}$ and the training data size is not large enough; in such cases, estimating $P(\mathbf{x},\mathbf{u},\mathbf{s})$ is very difficult. In the following, we describe our solution approach to tackle this challenge. Variational methods have recently become popular in the context of inference problems. Variational mutual information is a particular variational method which aims to find a lower bound for a mutual information [17].

Sensitive Loss: Let us find a lower bound the mutual information between two random variables $\hat{\mathbf{x}}$ and $\mathbf{s}$, with joint distribution distribution $P(\hat{\mathbf{x}},\mathbf{s})$

$$\begin{aligned} I(\hat{\mathbf{x}};\mathbf{s}) &= H(\mathbf{s}) - H(\mathbf{s}|\hat{\mathbf{x}}) \\ &= H(\mathbf{s}) + \mathbb{E}_{\hat{\mathbf{x}}}\mathbb{E}_{\mathbf{s}|\hat{\mathbf{x}}}[\log P(\mathbf{s}|\hat{\mathbf{x}})]. \end{aligned} \tag{6.9}$$

In practice, the mutual information term $I(\hat{\mathbf{x}};\mathbf{s})$ is hard to minimize directly as it requires access to $P(\mathbf{s}|\hat{\mathbf{x}})$. Fortunately, we can obtain a lower bound of it by defining a posterior distribution $Q_\phi(\mathbf{s}|\hat{\mathbf{x}})$ to approximate $P(\mathbf{s}|\hat{\mathbf{x}})$. We define $Q_\phi(\mathbf{s}|\hat{\mathbf{x}})$ as an ANN has

parameters ϕ.

$$\begin{aligned} I(\hat{\mathbf{x}};\mathbf{s}) &= H(\mathbf{s}) + \mathbb{E}_{\hat{\mathbf{x}}}\mathbb{E}_{\mathbf{s}|\hat{\mathbf{x}}}\left[\log\frac{Q_\phi(\mathbf{s}|\hat{\mathbf{x}})P(\mathbf{s}|\hat{\mathbf{x}})}{Q_\phi(\mathbf{s}|\hat{\mathbf{x}})}\right] \\ &= H(\mathbf{s}) + \mathbb{E}_{\hat{\mathbf{x}}}\mathbb{E}_{\mathbf{s}|\hat{\mathbf{x}}}[\log Q_\phi(\mathbf{s}|\hat{\mathbf{x}})] + \mathbb{E}_{\hat{\mathbf{x}}}\mathbb{E}_{\mathbf{s}|\hat{\mathbf{x}}}\left[\log\frac{P(\mathbf{s}|\hat{\mathbf{x}})}{Q_\phi(\mathbf{s}|\hat{\mathbf{x}})}\right] \\ &= H(\mathbf{s}) + \mathbb{E}_{\hat{\mathbf{x}}}\mathbb{E}_{\mathbf{s}|\hat{\mathbf{x}}}[\log Q_\phi(\mathbf{s}|\hat{\mathbf{x}})] + \mathbb{E}_{\hat{\mathbf{x}}}KL[P(\mathbf{s}|\hat{\mathbf{x}})||Q_\phi(\mathbf{s}|\hat{\mathbf{x}})]. \end{aligned} \tag{6.10}$$

It has to be a probability distribution for the KL divergence to be non-negative therefore for the bound to hold. Also, the bound is tight if P is exactly the same as the conditional distribution Q_ϕ, so that we can rewrite (6.10) as:

$$I(\hat{\mathbf{x}};\mathbf{s}) \geq H(\mathbf{s}) + \mathbb{E}_{\mathbf{s},\hat{\mathbf{x}}}[\log Q_\phi(\mathbf{s}|\hat{\mathbf{x}})] \tag{6.11}$$

With the constant $H(\mathbf{s})$ term dropped, we can write this lower bound alternatively in the following way:

$$I(\hat{\mathbf{x}};\mathbf{s}) = \max_\phi \mathbb{E}_{\mathbf{s},\hat{\mathbf{x}}}[\log Q_\phi(\mathbf{s}|\hat{\mathbf{x}})] \tag{6.12}$$

Target Loss: Using previous steps, the mutual information between $\mathbf{u}$ and $\hat{\mathbf{x}}$ can be written as:

$$I(\hat{\mathbf{x}};\mathbf{u}) = \max_\psi \mathbb{E}_{\mathbf{u},\hat{\mathbf{x}}}[\log Q_\psi(\mathbf{u}|\hat{\mathbf{x}})] \tag{6.13}$$

Sub sensitive loss (6.12) and target loss (6.13) in (6.8) we get:

$$\begin{aligned} \mathscr{L}_{PF}(\theta) &= \min_\theta \left[\max_\phi \mathbb{E}_{\mathbf{s},\hat{\mathbf{x}}}[\log Q_\phi(\mathbf{s}|\hat{\mathbf{x}})] - \lambda \max_\psi \mathbb{E}_{\mathbf{u},\hat{\mathbf{x}}}[\log Q_\psi(\mathbf{u}|\hat{\mathbf{x}})]\right] \\ &= \min_\theta \left[\lambda \min_\psi \mathbb{E}_{\mathbf{u},\hat{\mathbf{x}}}[\log Q_\psi(\mathbf{u}|\hat{\mathbf{x}})] - \min_\phi \mathbb{E}_{\mathbf{s},\hat{\mathbf{x}}}[\log Q_\phi(\mathbf{s}|\hat{\mathbf{x}})]\right] \end{aligned} \tag{6.14}$$

Our minimax formulation in (6.14) is similar to GAN objective function. It can be interpreted as FP wants to minimize both the target and adversary loss terms, while the adversary is trying to maximize sensitive loss. This optimization problem can be practically addressed via the training of three neural networks: FP $g_\theta(\mathbf{x})$ as an autoencoder, an adversary $Q_\phi(\mathbf{s}|\hat{\mathbf{x}})$ and a target $Q_\psi(\mathbf{u}|\hat{\mathbf{x}})$ as classifiers. To make the notations simple, we define target classifier(s) as $Q_\psi(\hat{\mathbf{x}})$ and adversary classifier(s) as $Q_\phi(\hat{\mathbf{x}})$.

Training Objective Function: Using the FP as an autoencoder, we will add the reconstruction function to (6.14). Practically, for binary vectors $\mathbf{s}$ and $\mathbf{u}$, we can write the objective function using the CE loss as fellow:

$$\min_{\theta} \Bigg[\lambda_1 \sum_{i=1}^{d_u} \min_{\psi_i} CE(u_i, Q_{\psi_i}(\hat{\mathbf{x}})) + \lambda_2 CE(\mathbf{x}, \hat{\mathbf{x}}) - (1-\lambda_1-\lambda_2)\lambda_3 \sum_{i=1}^{d_s} \min_{\phi_i} CE(s_i, Q_{\phi_i}(\hat{\mathbf{x}})) \Bigg]. \tag{6.15}$$

The solution to (6.15) will refer to as FP and is define as optimal protector for sensitive-target tradeoff in term of autoencoder as FP and target and adversary classifiers. Figure 6.1 shows our proposed adversarial framework.

6.3.3 Learning Algorithm

Our goal is for target classifiers $Q_{\psi_{(1,..,d_u)}}(\hat{\mathbf{x}})$ to predict $\mathbf{u}$ and for adversary classifiers $Q_{\phi_{(1,..,d_s)}}(\hat{\mathbf{x}})$ to predict $\mathbf{s}$ as well as possible, but for $g_\theta(\mathbf{x})$ is to make it hard for adversary classifiers to predict $\mathbf{s}$. The training procedure includes three steps of training as follows: **1)** Select a batch b training data and freeze the autoencoder to train adversary and target loss functions by solving the optimization problems (6.15). In other words, we train

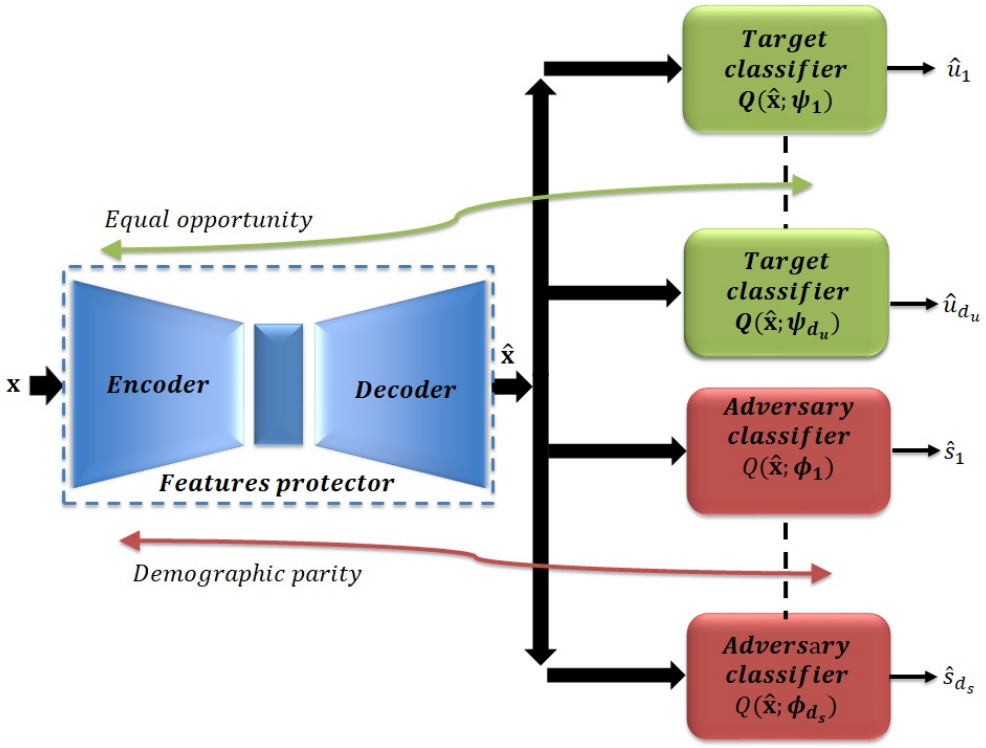

Figure 6.1: Architecture for the FP framework.

classifiers on the frozen autoencoder. Run the gradient ascent algorithm k iterations to get good an adversary and a target as classifiers.

2) Freeze adversary and target classifiers, use a new batch b training data to find $\hat{\mathbf{x}}$, $Q_\phi(\hat{\mathbf{x}})$ and $Q_\psi(\hat{\mathbf{x}})$.

3) Solve the optimization problem (6.15) to train the autoencoder.

Our approach for training the autoencoder as FP and the target and adversary classifiers as players in a game is detailed in learning algorithm 7.

norelsize 7 FP training algorithm.

Require: b, the batch size. k, steps are used for updating $\psi_{(1,..,d_u)}$ and $\phi_{(1,..,d_s)}$ in each iteration. λ_1 and λ_2 tradeoff factors.

$g_\theta(.), Q_{\psi_{(1,..,d_u)}}(.), Q_{\phi_{(1,..,d_s)}}(.) \leftarrow$ Random initialization
while θ has not converged **do**
  for k steps **do**
    Sample $\left\{\mathbf{x}^i, \mathbf{u}^i, \mathbf{s}^i\right\}_{i=1}^b$ a batch from training data.
    $\left\{\hat{\mathbf{x}}^i\right\}_{i=1}^b \leftarrow g_\theta(\left\{\mathbf{x}^i\right\}_{i=1}^b)$
    Perform SGD-updates for $\psi_{(1,..,d_u)}$ and $\phi_{(1,..,d_s)}$
    for $j = 1 : d_u$ **do**
      $m_{\psi_j} \leftarrow \nabla_{\psi_j} \frac{1}{b} \sum_{i=1}^{b} CE(u_j^i, Q_{\psi_j}(\hat{\mathbf{x}}^i))$
      $\psi_j \leftarrow \psi_j$ - α . AdamOptimizer(ψ_j, m_{ψ_j})
    end for
    for $j = 1 : d_s$ **do**
      $m_{\phi_j} \leftarrow \nabla_{\phi_j} \frac{1}{b} \sum_{i=1}^{b} CE(s_j^i, Q_{\phi_j}(\hat{\mathbf{x}}^i))$
      $\phi_j \leftarrow \phi_j$ - α . AdamOptimizer(ϕ_j, m_{ϕ_j})
    end for
  end for
  Sample $\left\{\mathbf{x}^i, \mathbf{u}^i, \mathbf{s}^i\right\}_{i=1}^b$ a batch from the training data.
  $\left\{\hat{\mathbf{x}}^i\right\}_{i=1}^b \leftarrow g_\theta(\left\{\mathbf{x}^i\right\}_{i=1}^b)$
  Perform SGD-updates for θ

$$m_\theta \leftarrow \nabla_\theta \frac{1}{b} \sum_{i=1}^{b} \left\{ \lambda_1 \sum_{j=1}^{d_u} CE(u_j^i, Q_{\psi_j}(\hat{\mathbf{x}}^i)) + \lambda_2 CE(\mathbf{x}, \hat{\mathbf{x}}) - (1 - \lambda_1 - \lambda_2) \sum_{j=1}^{d_s} CE(s_j^i, Q_{\phi_j}(\hat{\mathbf{x}}^i)) \right\}$$

  $\theta \leftarrow \theta$ - α . AdamOptimizer(θ, m_θ)
end while

Chapter 7: Experimental Analysis

In this chapter, we will evaluate the privacy-preserving algorithms described in chapters 4 and 5 and the algorithm for fair representations described in chapter 6. Also, we compare algorithm 2 with the existing literature solutions and present them graphically.

7.1 Datasets

Synthetic MNIST Dataset: The modified national standards and technology (MNIST) dataset is a handwritten digit dataset consisting of 60,000 training examples and 10,000 testing examples. Each sample is a 28×28 grayscale image [95]. We concatenating two digits to create one digit, the first set is between 00 and 19, and the second set is between 70 and 89; then, we create three synthetic datasets, one grayscale and two colored for each set. To keep the image resolution consistent across all experiments, we resize all the images in synthetic MNIST datasets to 64×64 pixels. We use 25,000 synthetic images for training, and 5,000 were used for testing.

CelebA Dataset: Large-scale celebrity faces attributes (CelebA) is a dataset with more than 200,000 celebrity images, each with 40 binary attribute annotations such as age (old or young), gender (male or female), whether the person is wearing glasses, and whether they are smiling. The images are each 218x178 pixels [102]. We prepare three datasets, each containing 20,000 training images and 2,500 testing images. Each image

has been cropped and resized to 64×64 pixels.

HAPT-Recognition Dataset: The human activities and postural transitions' recognition using smartphone data (HAPT-Recognition) is a dataset based on recordings of 30 participants performing six activities (Walking, Walking Upstairs, Walking Downstairs, Laying, Sitting) of daily living. Each participant was wearing a mobile phone (Samsung Galaxy S II) around their waist. A 50Hz constant rate of acceleration and angular velocity was recorded by using its embedded accelerometer and gyroscope. In addition, the experiments were video-recorded so that the data could be labeled manually. The dataset includes 10929 instances and 561 features [14]. By selecting 15 participants at random from a sample of 30 participants, we generate two datasets. We randomly split each dataset into training instances (70%) and test instances (30%).

7.2 Performance Metric

The receiver operating characteristic (ROC) analysis is one of the most important methods of measuring performance, as it provides a visual and numerical summary of the area under the receiver operating characteristic curve (AUC) of the behavior of a classifier. The ROC curve represents the probability, and the AUC represents the degree of separability. In this sense, it is a measure of how well a model can distinguish between classes. The Figure 7.1 shows that the diagonal line represents random classifiers (AUC $= 0.5$), suitable for sensitive information classification, which providing random answers regardless of the input. As long as the AUC is high (approximately 1), the classifier is more likely to predict 0 classes as 0 and 1 classes as 1, which is optimal for

utility information classification [126] [72].

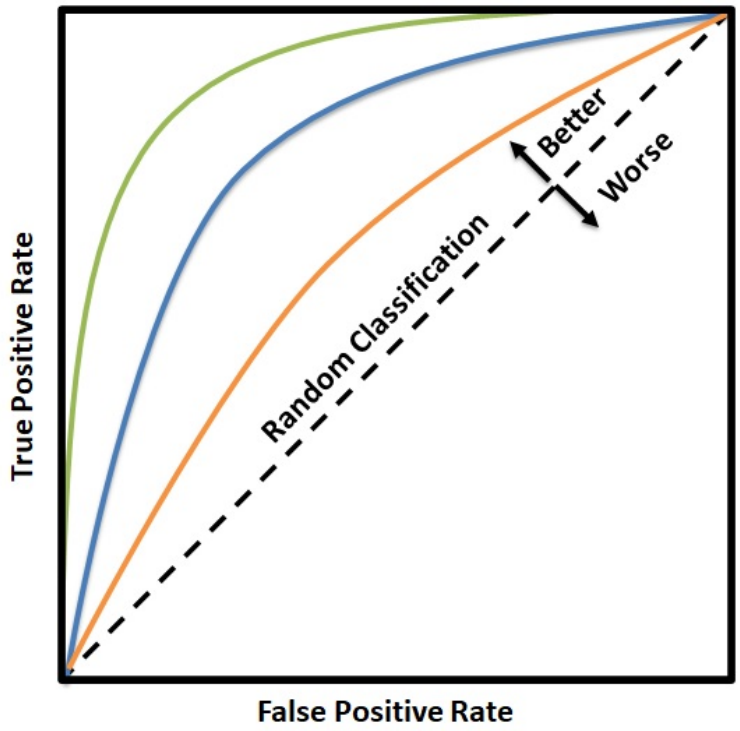

Figure 7.1: ROC curves.

7.3 Evaluation of the Proposed Algorithms

7.3.1 Algorithm 2

In this section, we empirically evaluate our proposed GPP model on real-world benchmark datasets and synthetic datasets. Three datasets are used: the MNIST dataset, the CelebA dataset, and the HAPT-Recognition dataset. The networks were trained on Py-

torch deep learning platform using Adam optimizer with a learning rate of 0.0001. We set β equal to 1, 0.7 and 1 for MNIST, CelebA, and HAPT-Recognition respectively and $b = 64$ for all datasets. We use $k = 2$ for MNIST and HAPT-Recognition while $k = 4$ for CelebA. To evaluate trained GPP, we implement utility and adversary classifiers as deep neural networks that are trained separately using the sanitized training instances $\{(T(\mathbf{x}^i), \mathbf{u}^i, \mathbf{s}^i), i = 1, \ldots, n\}$. Presumably, these classifiers act as ideal classifiers for classifying utility and private data.

Baseline Methods: It is essential to establish baseline performance to our GPP framework. In this work, we compare GPP to three other baseline machine learning algorithms:

- Privacy partial least squares (PPLS): using algorithm 1 from [56].
- Cleaning the null space (CNS): using algorithm 1 from [157].
- Non-negative matrix factorization (NMF): using algorithm 1 from [11].

7.3.1.1 Performance of Bottleneck Dimensions

Bottleneck "latent space" consists of a compressed representation of sanitized high-dimensional data, which is all the information that utility and private classifiers can use to detect public and private events, respectively. It is crucial that GPP design with a z dimension that includes the most relevant features. As a result, the bottleneck layer serves as another indicator of the privacy-utility tradeoff.

MNIST-(00-19) Dataset: Any number greater than or equal to 10 is considered a pri-

vate piece of information. Numbers are odd according to the target's information. GPP's performance was tested with values of $z = 40, 60, 60, 80, 100, 120$. The comparison is performed using three methods: PPLS, CNS and NMF. The accuracy results for utilities and adversaries are depicted in Figure 7.2. Our method achieves the highest accuracy for the utility classification (number is odd) and randomly guess, which implies vital privacy preservation for the adversary (number is ≥ 10) at $z = 120$. GPP achieves the best compromise of all baseline methods across all bottleneck dimensions z. When it comes to privacy, NMF achieves the right privacy-preserving level (0.54) at $z = 120$, but PPLS and CNS do less well for reducing the privacy risk.

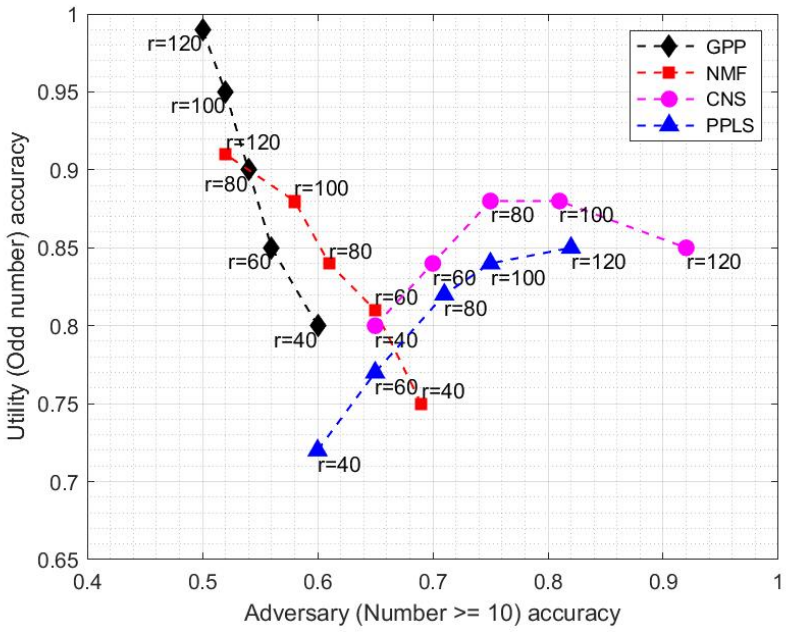

Figure 7.2: MNIST-(00-19) Dataset: number greater or equal to 10 vs odd number.

CelebA-Gender Dataset: Gender is considered a private piece of information. Wear-

ing glasses and smiling can provide useful information. In order to illustrate the tradeoff between privacy and utility accuracy, we tune the bottleneck layer, which controls the dimensions of the compressing representation, with dimensions $z = 100, 300, 400, 500, 600$. Figure 7.3 presents the utility accuracy of different compression techniques concerning adversary detection. At $z = 600$, our method achieves the highest accuracy for utility classification (smiling and wearing glasses) and random guess, which implies maintaining crucial privacy for the adversary (gender). With the increasing bottleneck dimension from 400 to 600, the utility accuracy rises from 90% to 97%. Accordingly, the bottleneck dimension adjusts the privacy with utility tradeoff.

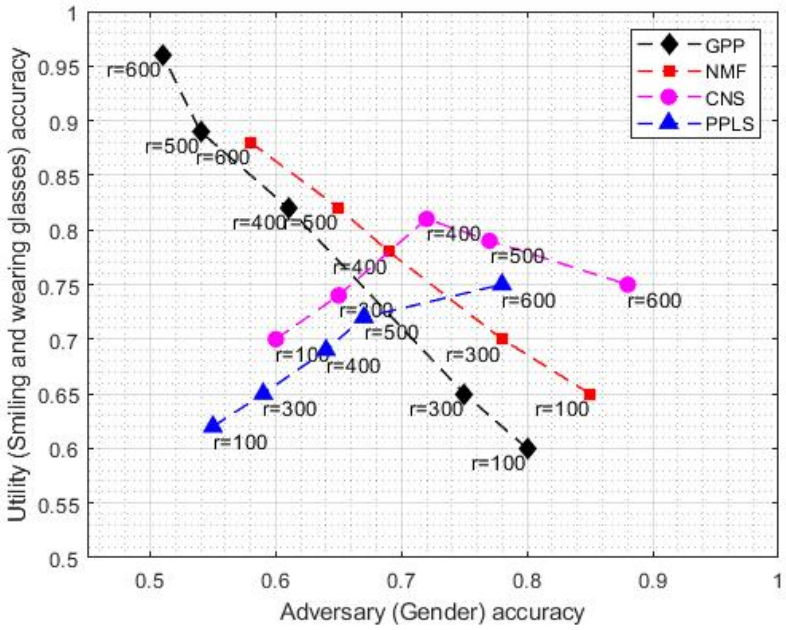

Figure 7.3: CelebA-Gender Dataset: Gender vs smiling and wearing glasses.

HAPT-Recognition Dataset: Public information is defined as the identity of a group of

individuals, while private information is defined as their activities. Figure 7.4 displays the accuracy results for utility and adversary classifiers. For all $z = 20, 40, 60, 80, 100$ values, GPP achieves the highest accuracy as a compromise with all baseline methods, as demonstrated in previous experiments. Therefore, we cannot distinguish individuals, but we can detect their activity accurately. For both GPP and NMF, utility accuracy increases as z increases, and adversary accuracy decreases. With increasing z, the CNS is better than PPLS in utility tasks; however, the PPLS is better in adversary tasks.

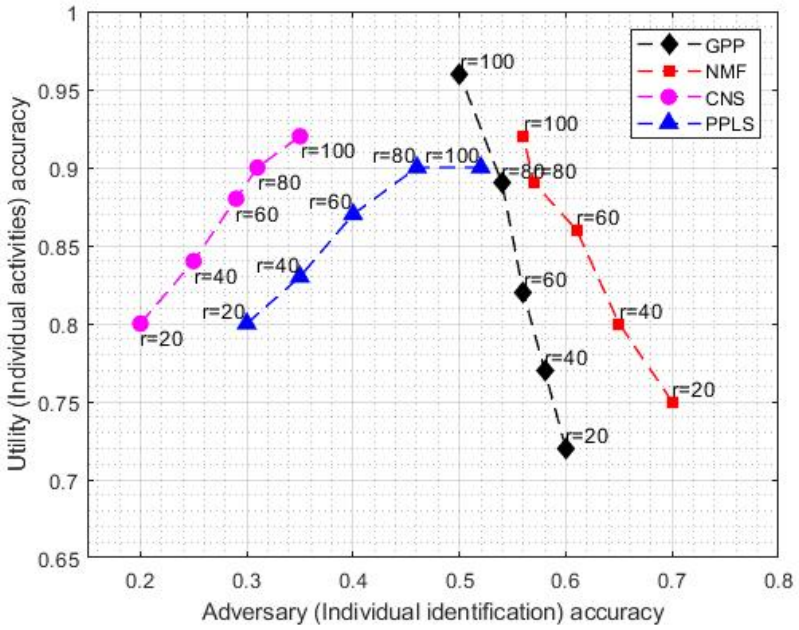

Figure 7.4: HAPT-Recognition Dataset: Individual identification vs individual activities.

7.3.2 Algorithm 3

Experimental analyses of algorithm 3 have been conducted using three datasets: MNIST-(00-19), CelebA-Smiling, and HAPT-Recognition. To carry out the experiment, we set β_1 and β_2 equal to 0.7 for CelebA-Smiling and equivalent to 1 for MNIST-(70-89) and HAPT-Recognition. The k value is 4 for CelebA-Smiling and 2 for MNIST-(00-19) and HAPT-Recognition. Adam optimizer's learning rate is 0.0001. All datasets have b equal to 64.

MNIST-(00-19) Datasets: In this procedure, two datasets are used, gray and colored. Forty thousand images were divided equally among the two datasets for training and 2,500 samples for each GPP as a testing dataset. The utility is defined as an odd number, and an adversary is a number greater than or equal to 10. The GPP maps a 64×64 input image to latent space $z \in R^{120}$. The evaluation of the methodology focuses on its accuracy. The ROC curves of utility and adversary classifiers are shown in Figure 7.5 based on sanitized training data. AUC approaches 1 for utility classifiers and 0.5 for adversary classifiers (the average test accuracy over the two datasets). From 0.99 to 1, we observe a significant improvement in the computation benefits of utility events. In summary, GPPs produce sanitized features that allow utility data to be mined effectively, while private data cannot be inferred, i.e., the adversary classifier is like a random guess.

CelebA Datasets: The experiment considers gender as an adversary, smiling as a utility for the first dataset, wearing glasses as an adversary, and smiling as a utility for the second dataset. The number of training images for each dataset is 20,000, and the number

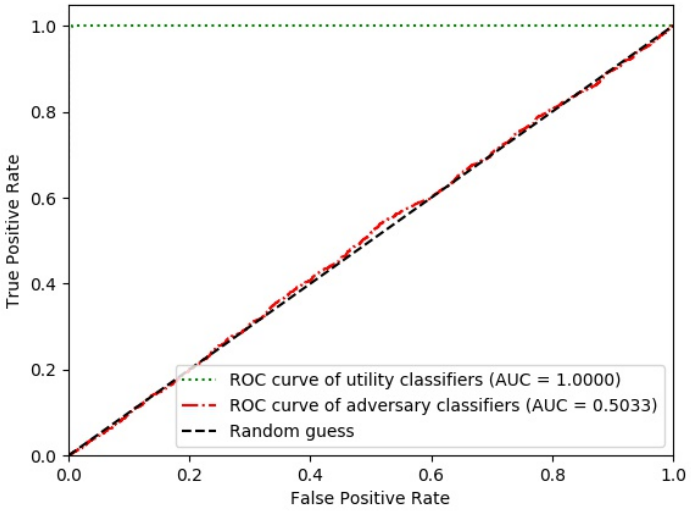

Figure 7.5: MNIST Datasets: ROC curves for utility and adversary classifiers.

of testing images is 2,500. GPP maps a 64×64 input image to latent space $z \in R^{600}$. Figure 7.6 shows that the ROC for the utility classifier (smiling) is quite good, whereas the ROC for the adversary classifiers (gender and wearing glasses) is an almost random estimate.

HAPT-Recognition Datasets: Two groups of HAPT-Recognition dataset with 15 participants for each group are used in the experiment. The training portion is split randomly into 75 percent, and the testing portion is 25 percent. This GPP has the dimension $z \in R^{100}$. The Users' identities are sensitive information, while activity recognition is the utility part. Figure 7.7 shows that the utility classifier's ROC is quite good. However, ROC for the adversary classifier (the average accuracy of both datasets for the adversary in a test case) is almost random.

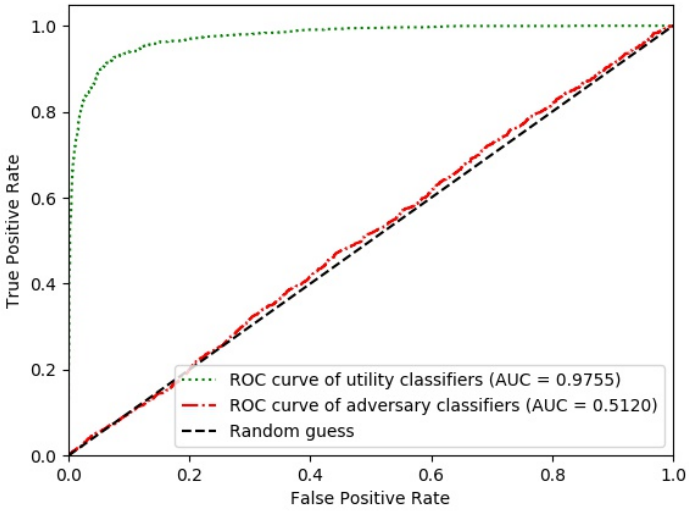

Figure 7.6: CelebA Datasets: ROC curves for utility and adversary classifiers,

7.3.3 Algorithm 4

We evaluated the question: How can we ensure data privacy in a scenario where sensitive information is so often present? We conducted experiments that empirically addressed the answer. We define utility information as the two-digit number in a synthetic image. This type of information is what we want to expose. For all datasets, we set $\beta = 1$, $b = 64$, and $k = 2$. To evaluate a trained IB framework $T_\theta(\cdot)$, we implement utility classifier and adversary decoder that are trained separately using the sanitized training instances $\{(T(\mathbf{x}^i), \mathbf{u}^i), i = 1, \ldots, n\}$.

To evaluate our proposed method, we conducted experiments on image classification. We use synthetic datasets to evaluate our proposed IB framework. Based on MNIST-(00-19) and MNIST-(70-89) datasets, Figures 7.8 and 7.9 illustrate the optimal

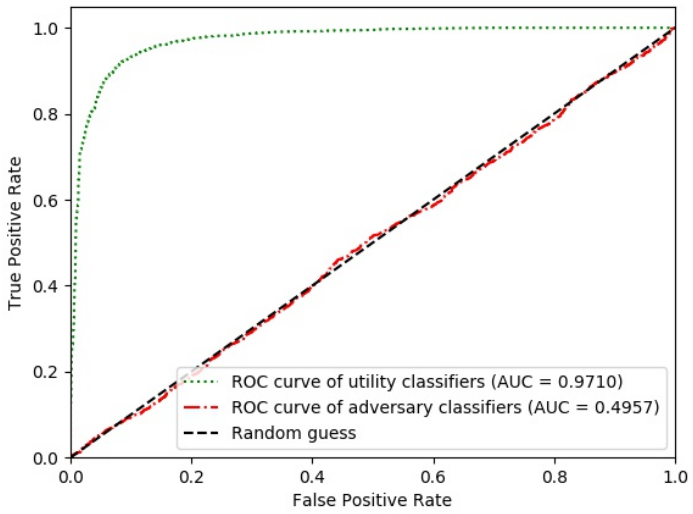

Figure 7.7: HAPT-Recognition Datasets: ROC curves for utility and adversary classifiers.

IB framework's performance. IB Framework achieves superior accuracy by maintaining utility information, two-digit number, which make it easy to classify. In contrast, the IB framework preserves sensitive features, background image colors, and digits' colors.

7.3.4 Algorithm 5

For the IB principle, this experiment explores the privacy-preserving in the IoT domain. In IB distributed training, two IoT devices are integrated with one aggregator, i.e., utility classifiers, which allow each device to update the sensitive parameters locally and send their sanitized data to the aggregator. The first framework uses the gray MNIST dataset, while the second framework uses the colored MNIST dataset.

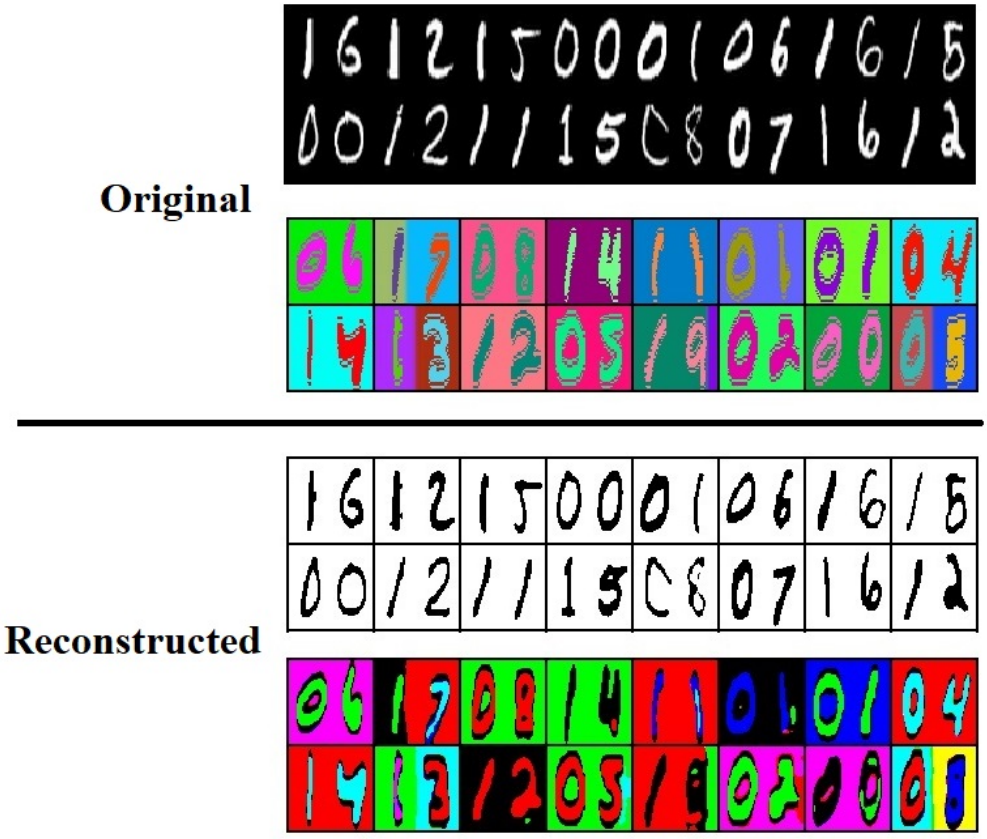

Figure 7.8: MNIST-(00-19) Dataset: Input digits of the encoder (top) and output digits of the decoder (bottom).

In table 7.1, the AUCs for the utility classifiers are shown for the MNIST-(00-19) and MNIST-(70-89) datasets. Using the table, we can see how algorithm 5 compares well to algorithm 4 in terms of utility event classification on sanitized data, similar to raw data performance.

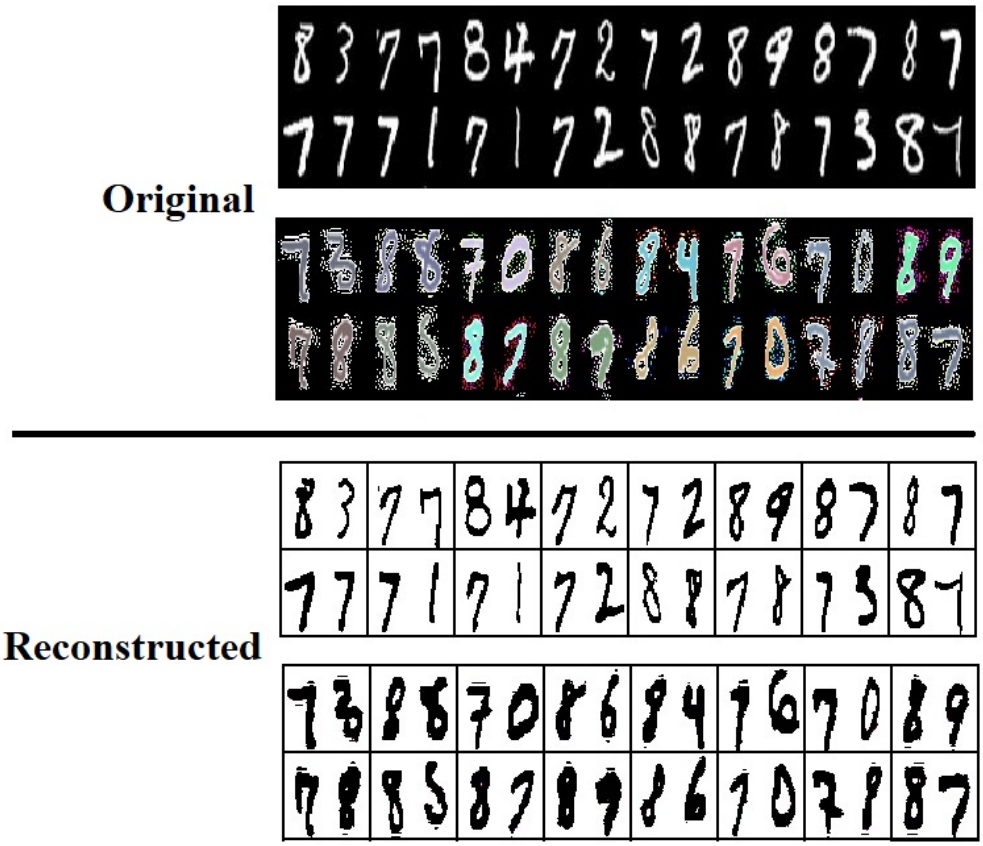

Figure 7.9: MNIST-(70-89) dataset: Input digits of the encoder (top) and output digits of the decoder (bottom).

Table 7.1: Results of algorithms four and five.

Datasets	**AUC Raw Data**	**AUC Algorithm 4**	**AUC Algorithm 5**
MNIST-(00-19)	1	0.98	0.99
MNIST-(70-89)	0.97	0.956	0.96

7.3.5 Algorithm 6

In a scenario where utility information is regularly present, how can we ensure data privacy? We conducted experiments to answer the question empirically. To evaluate

our work, we use two datasets: gray MNIST-(70-89) dataset and HAPT-Recognition dataset. The networks were trained using the Adam optimizer with a learning rate of 0.0001. We set β equal to 1 for MNIST and HAPT-Recognition and $b = 64$ for both datasets. We use $k = 2$ for MNIST while $k = 3$ for HAPT-Recognition.

MNIST-(70-89): We define private data as the two-digit number in the synthetic image greater than or equal to 80. We want to hide the first digit to ensure that an adversary cannot guess whether it is eight or seven. In the testing phase, the utility information is defined as whether the two-digit number in the image is odd.

Figure 7.10 shows the outputs from our learned PF for MNIST-(70-89) images. There are original images on the top and reconstructed images on the bottom. The first digit represents a private piece of information that we wish to keep hidden. An adversary classifier cannot determine whether it is seven or eight due to the perturbation of the first digit. Figure 7.11 provides ROC curves for utility and adversary classifiers trained using trained PF and used to evaluate the efficiency of the proposed PF to retain the data features needed for accurate classification. As can be seen, the AUC is close to 0.96 for the utility classifier and near 0.53 for the adversary classifier. As a result, PF produces sanitized features that make it possible to mine utility data effectively. Comparatively, private data cannot be inferred, i.e., the adversary classifier performs like a random guess.

HAPT-Recognition Dataset: In the utility part, the activity recognition is featured, while users' identities are presented in the sensitive part. According to Figure 7.12, the ROC for utility classifiers is quite good, while the ROC for adversary classifiers is a random guess.

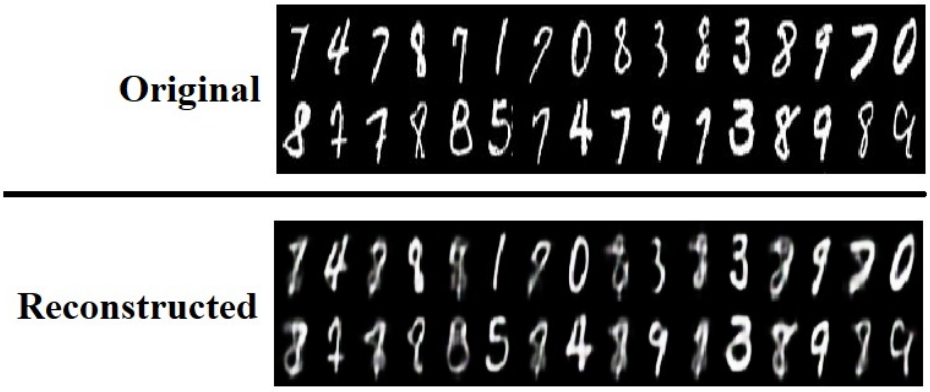

Figure 7.10: MNIST(70-89) Dataset: The top two rows show the original images, and the remaining rows visualize outputs for original images from our learned PF.

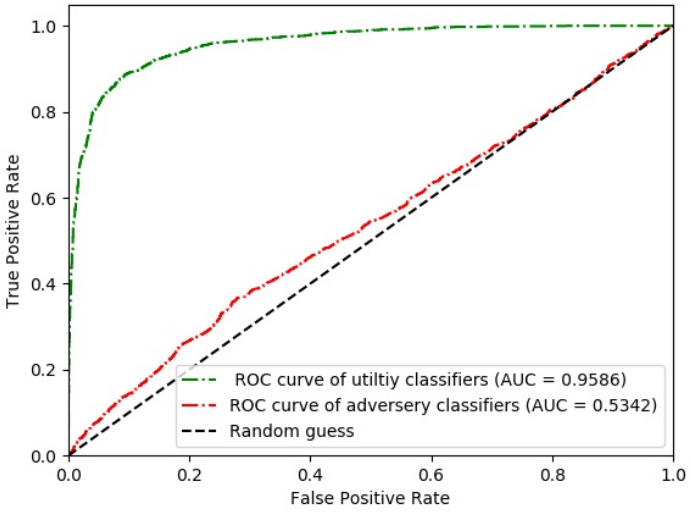

Figure 7.11: MNIST-(70-89) Dataset: ROC curves for utility and adversary classifiers.

7.3.6 Algorithm 7

This section presents numerical experiments with the proposed method on two real-world datasets: MNIST datasets and CelebA datasets. The networks were trained using

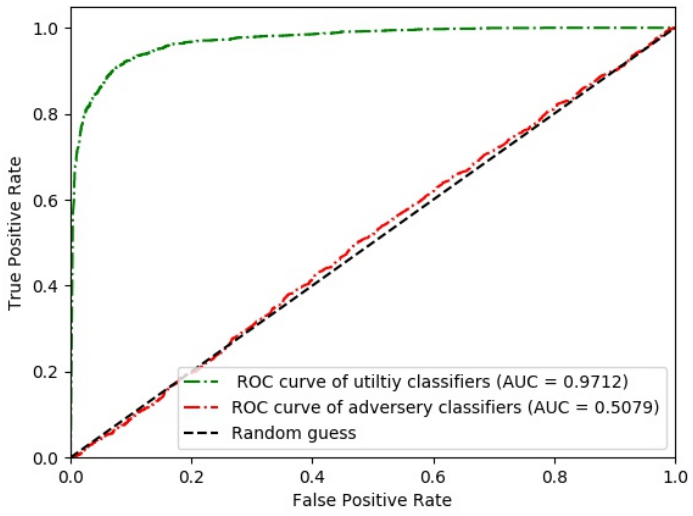

Figure 7.12: HAPT-Recognition Dataset: ROC curves for utility and adversary classifiers.

the Pytorch deep learning platform using Adam optimizer with a learning rate of 0.0001. λ_1 and λ_2 are equal to 0.1 for MNIST and 0.2 for CelebA, and b is equal to 128 for both datasets. The k value for MNIST is 2 whereas the k value for CelebA is 4.

MNIST Datasets: A private piece of information is defined as any number greater than or equal to 10 based on MNIST-(00-19) and greater or equal to 80 based on MNIST-(70-89). The target's information is that the number is odd.

Figures 7.13 and 7.14 show three sets of images; the top row shows the original images, and the bottom row shows the reconstructed images based on FP output. Reconstructed sets indicate that FP successfully removes sensitive features, like digits greater than 10 for MNIST-(00-19) and digits greater than 80 for MNIST-(70-89) while keeping

target features, such as digits odd and clear to recognize. In other words, we can never know the actual value of the first digit found in MNIST-(00-19) if it is one or zero, and in MNIST-(70-89) if it is seven or eight. In addition, the target data is reconstructed very closely to the ground truth images.

CelebA Datasets: Three datasets are created for CelebA. In the first dataset, we refer to smile, i.e., whether or not the individuals in the images smile, hair color, and oval face as utility information, while gender and straight hair are considered private information. According to the second dataset, eyeglasses and straight hair are sensitive information, while the smile, gender, and hair color are not. We consider setting the gender and hair color as public information while smile and oval face are private information in the third dataset.

Figures 7.15, 7.16, and 7.17 show the top row as input to FP and the bottom row as output from FP. The reconstructed examples illustrate the method's ability to hide sensitive features while preserving target features in all images.

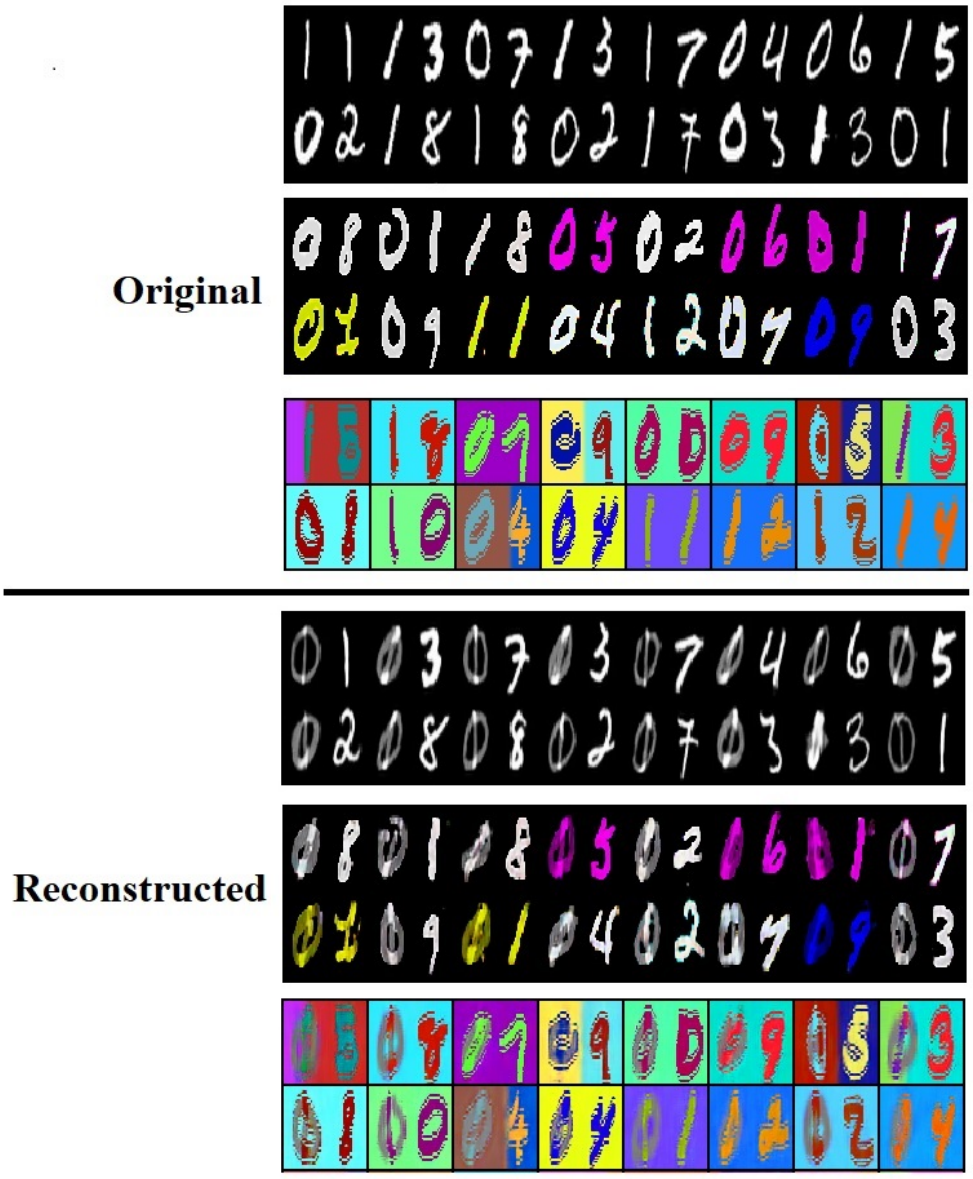

Figure 7.13: MNIST-(00-19) Dataset: For utility information, the two-digit number is odd, whereas for private information, the two-digit number is ≥ 10.

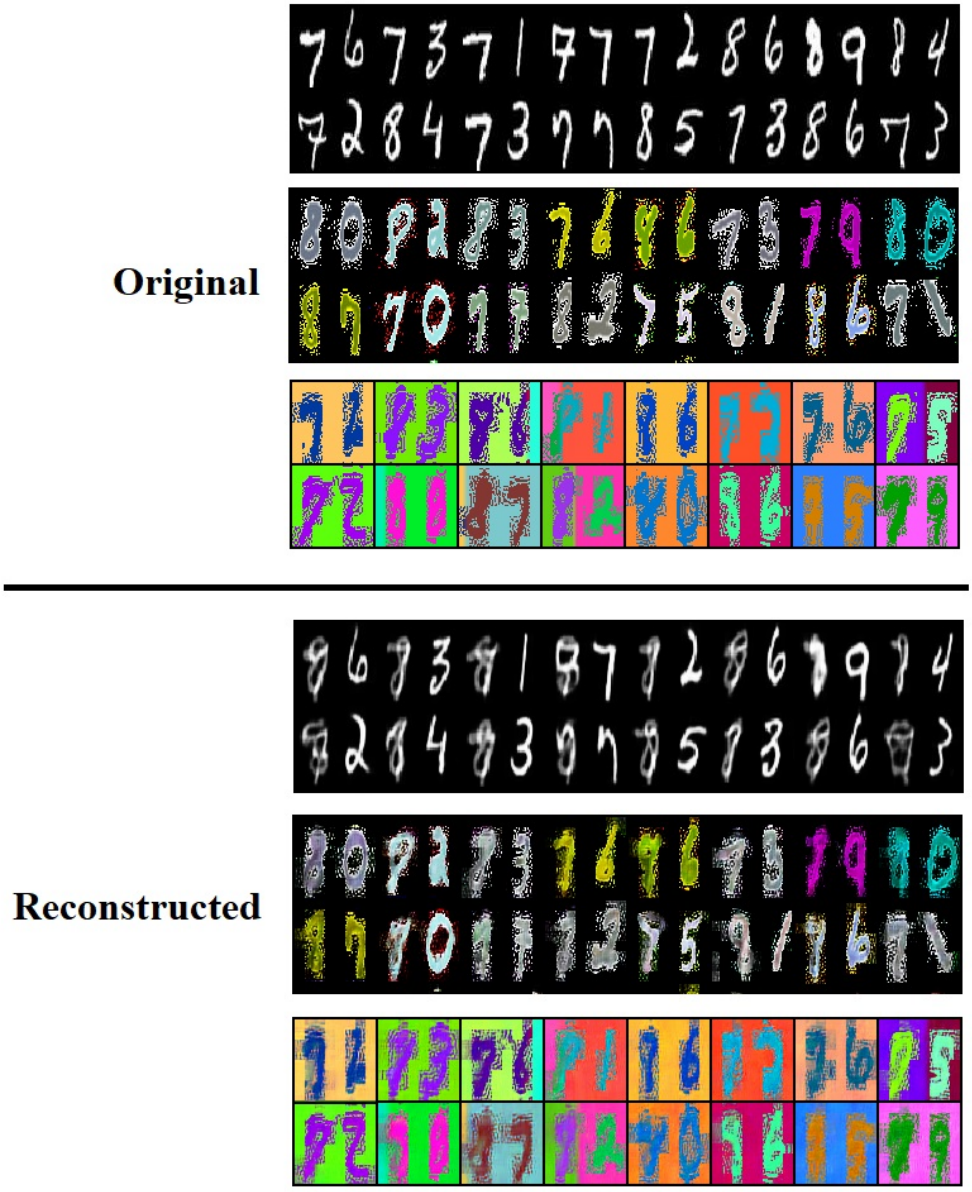

Figure 7.14: MNIST-(00-19) Dataset: For utility information, the two-digit number is odd, whereas for private information, the two-digit number is ≥ 80.

Figure 7.15: CelebA Dataset: Smile, hair color, and oval face as utility information, while gender and straight hair as private information.

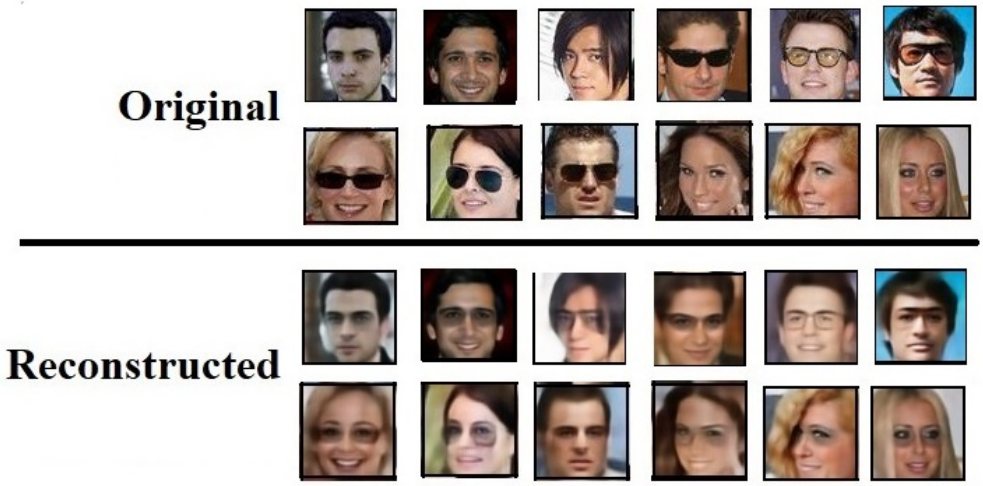

Figure 7.16: CelebA Dataset: Smile, gender, and hair color as utility information, while eyeglasses and straight hair as private information.

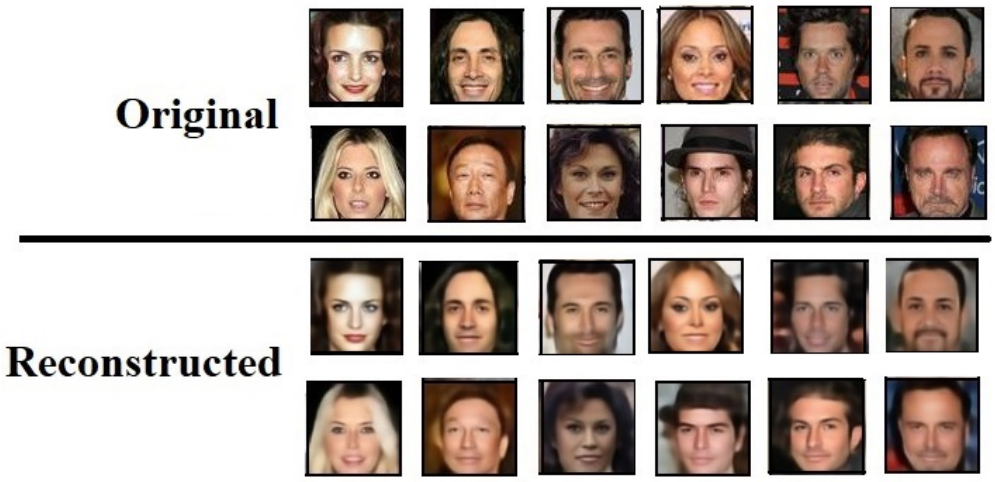

Figure 7.17: CelebA Dataset: Gender and hair color as utility information while smile and oval face as private information.

Chapter 8: Conclusion

8.1 Chapter Four

This chapter proposes GPP, which optimizes privacy-preserving data release mechanisms to minimize distortion of public data while concealing sensitive information. In this setting, there are two types of highly correlated data: private and useful. GPP is evaluated specifically in data sanitization, which is simply removing private information from the data while keeping the relevant information used to improve the inference accuracy of non-private information. Specifically, we use adversarially-trained neural networks to compute a variational approximation of mutual information privacy. Also, a distributed learning algorithm is demonstrated on a real dataset for the GPP framework. The experimental results on three datasets MNIST, CelebA, and HAPT-Recognition, show that the GPP framework is highly effective and achieves the highest classification accuracy.

8.2 Chapter Five

Our first objective in this chapter is to propose a new framework to achieve privacy under the IB principle. In general, we consider most of the data that would be exposed as sensitive, i.e., access to only utility data. We utilize Bayesian networks to specify

the system of IB and which information terms should be maintained. Also, we present a novel distributed privacy-preserving framework that implements the IB framework to ensure privacy within IoT devices. Based on the experiments with four databases, the proposed approach appears to provide utility information from sanitized data similar to raw data. In other words, the framework provides utility data while concealing sensitive and personal information.

The second objective of this chapter is to design, implement, and evaluate a novel PF framework resilient against adversarial attacks. As a case study, the PF framework is evaluated in the context of users who wish to reveal data, mainly utility information, to gain utility while maintaining their privacy. Specifically, we study the case of continuous, high-dimensional data with private labels that are high-dimensional vectors. Results on two datasets, MNIST and HAPT-Recognition, show that the PF framework is highly effective and achieves the highest utility classification accuracy and random guess for sensitive information.

8.3 Chapter Six

This chapter presents a framework for creating fair representations for data publishing. A crucial part of this approach is the use of adversarially-trained neural networks. In the first group, the FP functions as an autoencoder and target classifier. In contrast, the second network is adversary classifiers that attempt to retrieve sensitive information from the released data. Our approach offers significant information-theoretically optimal sensitive-target tradeoff, which we demonstrate in experiments using nine datasets.

Experimental results indicate that our approach generates data with improved fairness properties while maintaining classification accuracy. Furthermore, the framework is conceptually simple and can be applied to privacy-preserving data reconstruction.

Bibliography

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318. ACM, 2016.

[2] Mamun Abu-Tair, Soufiene Djahel, Philip Perry, Bryan Scotney, Unsub Zia, Jorge Martinez Carracedo, and Ali Sajjad. Towards secure and privacy-preserving iot enabled smart home: Architecture and experimental study. *Sensors*, 20(21):6131, 2020.

[3] Tameem Adel, Isabel Valera, Zoubin Ghahramani, and Adrian Weller. One-network adversarial fairness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2412–2420, 2019.

[4] David Barber Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in Neural Information Processing Systems*, 16:201, 2004.

[5] Charu C Aggarwal. On k-anonymity and the curse of dimensionality. In *VLDB*, volume 5, pages 901–909, 2005.

[6] Mohammad Al-Rubaie and J Morris Chang. Privacy-preserving machine learning: Threats and solutions. *IEEE Security & Privacy*, 17(2):49–58, 2019.

[7] Alexander A Alemi. Variational predictive information bottleneck. In *Symposium on Advances in Approximate Bayesian Inference*, pages 1–6. PMLR, 2020.

[8] Alexander A Alemi, Ian Fischer, and Joshua V Dillon. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

[9] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

[10] Zahir Alsulaimawi. Gaussian privacy protector for online data communication in a public world. In *2020 IEEE 6th Intl Conference on Big Data Security on*

Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), pages 169–173. IEEE, 2020.

[11] Zahir Alsulaimawi. A non-negative matrix factorization framework for privacy-preserving and federated learning. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2020.

[12] Zahir Alsulaimawi. A privacy filter framework for internet of robotic things applications. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 262–267. IEEE, 2020.

[13] Zahir Alsulaimawi, Jinsub Kim, and Thinh Nguyen. Sequential game network (segane) with application to online data sanitization. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1326–1330. IEEE, 2018.

[14] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. A public domain dataset for human activity recognition using smartphones. In *Esann*, 2013.

[15] Emmanuel Antwi-Boasiako, Shijie Zhou, Yongjian Liao, Qihe Liu, Yuyu Wang, and Kwabena Owusu-Agyemang. Privacy preservation in distributed deep learning: A survey on distributed deep learning, privacy preservation techniques used and interesting research directions. *Journal of Information Security and Applications*, 61:102949, 2021.

[16] Madhushri Banerjee and Sumit Chakravarty. Privacy preserving feature selection for distributed data using virtual dimension. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 2281–2284. ACM, 2011.

[17] David Barber and Felix V Agakov. The im algorithm: a variational approach to information maximization. In *Advances in neural information processing systems*, page None, 2003.

[18] Yuksel Ozan Basciftci, Ye Wang, and Prakash Ishwar. On privacy-utility tradeoffs for constrained data release mechanisms. In *2016 Information Theory and Applications Workshop (ITA)*, pages 1–6. IEEE, 2016.

[19] Roberto J Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *21st International conference on data engineering (ICDE'05)*, pages 217–228. IEEE, 2005.

[20] Brett K Beaulieu-Jones, William Yuan, Samuel G Finlayson, and Zhiwei Steven Wu. Privacy-preserving distributed deep learning for clinical data. *arXiv preprint arXiv:1812.01484*, 2018.

[21] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[22] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.

[23] Richard Berk, Hoda Heidari, Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. A convex framework for fair regression. *arXiv preprint arXiv:1706.02409*, 2017.

[24] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, page 0049124118782533, 2018.

[25] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

[26] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. Machine learning classification over encrypted data. In *NDSS*, volume 4324, page 4325, 2015.

[27] Christopher Briggs, Zhong Fan, and Peter Andras. A review of privacy-preserving federated learning for the internet-of-things. *Federated Learning Systems*, pages 21–50, 2021.

[28] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[29] Kenneth P Burnham and David R Anderson. Practical use of the information-theoretic approach. In *Model selection and inference*, pages 75–117. Springer, 1998.

[30] Kenneth P Burnham and David R Anderson. A practical information-theoretic approach. *Model selection and multimodel inference*, 2, 2002.

[31] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18. IEEE, 2009.

[32] Toon Calders, Asim Karim, Faisal Kamiran, Wasif Ali, and Xiangliang Zhang. Controlling attribute effect in linear regression. In *2013 IEEE 13th international conference on data mining*, pages 71–80. IEEE, 2013.

[33] Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, pages 3992–4001, 2017.

[34] Alessandro Castelnovo, Riccardo Crupi, Giulia Del Gamba, Greta Greco, Aisha Naseer, Daniele Regoli, and Beatriz San Miguel Gonzalez. Befair: Addressing fairness in the banking sector. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3652–3661. IEEE, 2020.

[35] Simon Caton and Christian Haas. Fairness in machine learning: A survey. *arXiv preprint arXiv:2010.04053*, 2020.

[36] Kamalika Chaudhuri, Anand D Sarwate, and Kaushik Sinha. A near-optimal algorithm for differentially-private principal components. *The Journal of Machine Learning Research*, 14(1):2905–2943, 2013.

[37] Gal Chechik, Amir Globerson, Naftali Tishby, Yair Weiss, and Peter Dayan. Information bottleneck for gaussian variables. *Journal of machine learning research*, 6(1), 2005.

[38] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H Vincent Poor, and Shuguang Cui. A joint learning and communications framework for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 2020.

[39] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *arXiv preprint arXiv:1606.03657*, 2016.

[40] Jie Cheng and Russell Greiner. Comparing bayesian network classifiers. *arXiv preprint arXiv:1301.6684*, 2013.

[41] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[42] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[43] Antonia Creswell, Yumnah Mohamied, Biswa Sengupta, and Anil A Bharath. Adversarial information factorization. *arXiv preprint arXiv:1711.05175*, 2017.

[44] Rachel Cummings, Varun Gupta, Dhamma Kimpara, and Jamie Morgenstern. On the compatibility of privacy and fairness. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pages 309–315, 2019.

[45] Brian d'Alessandro, Cathy O'Neil, and Tom LaGatta. Conscientious classification: A data scientist's guide to discrimination-aware classification. *Big data*, 5(2):120–134, 2017.

[46] Richard Dosselmann, Mehdi Sadeqi, and Howard J Hamilton. A tutorial on computing t-closeness. *arXiv preprint arXiv:1911.11212*, 2019.

[47] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. Fairness in deep learning: A computational perspective. *IEEE Intelligent Systems*, 2020.

[48] Flávio du Pin Calmon and Nadia Fawaz. Privacy against statistical inference. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1401–1408. IEEE, 2012.

[49] Ashutosh Dhar Dwivedi, Gautam Srivastava, Shalini Dhar, and Rajani Singh. A decentralized privacy-preserving healthcare blockchain for iot. *Sensors*, 19(2):326, 2019.

[50] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[51] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, pages 119–133, 2018.

[52] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[53] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[54] Harrison Edwards and Amos Storkey. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*, 2015.

[55] Khaled El Emam and Fida Kamal Dankar. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, 15(5):627–637, 2008.

[56] Miro Enev, Jaeyeon Jung, Liefeng Bo, Xiaofeng Ren, and Tadayoshi Kohno. Sensorsift: balancing sensor data privacy and utility in automated face understanding. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 149–158. ACM, 2012.

[57] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

[58] Ronald A Fisher. On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222(594-604):309–368, 1922.

[59] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1322–1333, 2015.

[60] Paul Fremantle, Benjamin Aziz, and Tom Kirkham. Enhancing iot security and privacy with distributed ledgers-a position paper. In *IoTBDS 2017: 2nd Interantional Conference on Internet of Things, Big Data and Security*, pages 344–349. SCITEPRESS–Science and Technology Publications, 2017.

[61] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 329–338, 2019.

[62] Benjamin CM Fung, Ke Wang, Rui Chen, and Philip S Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys (Csur)*, 42(4):1–53, 2010.

[63] Shripad Gade. *Accuracy-aware privacy mechanisms for distributed computation*. PhD thesis, University of Illinois at Urbana-Champaign, 2020.

[64] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

[65] Joseph Geumlek and Kamalika Chaudhuri. Profile-based privacy for locally private computations. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 537–541. IEEE, 2019.

[66] Ran Gilad-Bachrach, Nathan Dowlin, Kim Laine, Kristin Lauter, Michael Naehrig, and John Wernsing. Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy. In *International Conference on Machine Learning*, pages 201–210, 2016.

[67] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[68] Sivakanth Gopi, Pankaj Gulhane, Janardhan Kulkarni, Judy Hanwen Shen, Milad Shokouhi, and Sergey Yekhanin. Differentially private set union. In *International Conference on Machine Learning*, pages 3627–3636. PMLR, 2020.

[69] Hassan Hafez-Kolahi and Shohreh Kasaei. Information bottleneck and its applications in deep learning. *arXiv preprint arXiv:1904.03743*, 2019.

[70] Jihun Hamm. Preserving privacy of continuous high-dimensional data with minimax filters. In *Artificial Intelligence and Statistics*, pages 324–332, 2015.

[71] Jihun Hamm, Yingjun Cao, and Mikhail Belkin. Learning privately from multiparty data. In *International Conference on Machine Learning*, pages 555–563, 2016.

[72] Peng Han, Shuo Shang, Aixin Sun, Peilin Zhao, Kai Zheng, and Panos Kalnis. Auc-mf: point of interest recommendation with auc maximization. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1558–1561. IEEE, 2019.

[73] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in neural information processing systems*, pages 3315–3323, 2016.

[74] Muneeb Ul Hassan, Mubashir Husain Rehmani, and Jinjun Chen. Differential privacy techniques for cyber physical systems: a survey. *IEEE Communications Surveys & Tutorials*, 22(1):746–789, 2019.

[75] Hoda Heidari, Michele Loi, Krishna P Gummadi, and Andreas Krause. A moral framework for understanding fair ml through economic models of equality of opportunity. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 181–190, 2019.

[76] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[77] Dawn E Holmes. *Innovations in Bayesian networks: theory and applications*, volume 156. Springer, 2008.

[78] Bao-Gang Hu. Information theory and its relation to machine learning. In *Proceedings of the 2015 Chinese Intelligent Automation Conference*, pages 1–11. Springer, 2015.

[79] Zonghao Huang, Rui Hu, Yuanxiong Guo, Eric Chan-Tin, and Yanmin Gong. Dp-admm: Admm-based distributed learning with differential privacy. *IEEE Transactions on Information Forensics and Security*, 15:1002–1012, 2019.

[80] Yasser Jafer, Stan Matwin, and Marina Sokolova. A framework for a privacy-aware feature selection evaluation measure. In *2015 13th Annual Conference on Privacy, Security and Trust (PST)*, pages 62–69. IEEE, 2015.

[81] Joohyung Jeon, Junhui Kim, Joongheon Kim, Kwangsoo Kim, Aziz Mohaisen, and Jong-Kook Kim. Privacy-preserving deep learning computation for geo-distributed medical big-data platforms. In *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks–Supplemental Volume (DSN-S)*, pages 3–4. IEEE, 2019.

[82] Yutao Jiao, Ping Wang, Dusit Niyato, Bin Lin, and Dong In Kim. Toward an automated auction framework for wireless federated learning services market. *IEEE Transactions on Mobile Computing*, 2020.

[83] Gareth P Jones, James M Hickey, Pietro G Di Stefano, Charanpal Dhanjal, Laura C Stoddart, and Vlasios Vasileiou. Metrics and methods for a systematic comparison of fairness-aware machine learning algorithms. *arXiv preprint arXiv:2010.03986*, 2020.

[84] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. Extremal mechanisms for local differential privacy. *arXiv preprint arXiv:1407.1338*, 2014.

[85] Faisal Kamiran and Toon Calders. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*, pages 1–6. IEEE, 2009.

[86] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

[87] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 35–50. Springer, 2012.

[88] Muhammad Khan, Simon Foley, and Barry O'Sullivan. From k-anonymity to differential privacy: A brief introduction to formal privacy models. 2021.

[89] Miran Kim, Junghye Lee, Lucila Ohno-Machado, and Xiaoqian Jiang. Secure and differentially private logistic regression for horizontally distributed data. *IEEE Transactions on Information Forensics and Security*, 15:695–710, 2019.

[90] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[91] Svetlana Kiritchenko and Saif M Mohammad. Examining gender and race bias in two hundred sentiment analysis systems. *arXiv preprint arXiv:1805.04508*, 2018.

[92] Jack Klys, Jake Snell, and Richard Zemel. Learning latent subspaces in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 6445–6455, 2018.

[93] Aleksandra Korolova. Privacy violations using microtargeted ads: A case study. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pages 474–482. IEEE, 2010.

[94] Balachander Krishnamurthy and Craig E Wills. Privacy leakage in mobile online social networks. In *Proceedings of the 3rd Wonference on Online social networks*, pages 4–4. 2010.

[95] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[96] Donghe Li, Qingyu Yang, Wei Yu, Dou An, Yang Zhang, and Wei Zhao. Towards differential privacy-based online double auction for smart grid. *IEEE Transactions on Information Forensics and Security*, 15:971–986, 2019.

[97] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115. IEEE, 2007.

[98] Ping Li, Jin Li, Zhengan Huang, Tong Li, Chong-Zhi Gao, Siu-Ming Yiu, and Kai Chen. Multi-key privacy-preserving deep learning in cloud computing. *Future Generation Computer Systems*, 74:76–85, 2017.

[99] Wenqi Li, Fausto Milletarì, Daguang Xu, Nicola Rieke, Jonny Hancox, Wentao Zhu, Maximilian Baust, Yan Cheng, Sébastien Ourselin, M Jorge Cardoso, et al. Privacy-preserving federated brain tumour segmentation. In *International Workshop on Machine Learning in Medical Imaging*, pages 133–141. Springer, 2019.

[100] Yehuda Lindell. Secure multiparty computation (mpc). *IACR Cryptol. ePrint Arch.*, 2020:300, 2020.

[101] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[102] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018.

[103] Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

[104] Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

[105] Chuan Ma, Jun Li, Ming Ding, Howard H Yang, Feng Shu, Tony QS Quek, and H Vincent Poor. On safeguarding privacy and security in the framework of federated learning. *IEEE Network*, 2020.

[106] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24. IEEE, 2006.

[107] Ali Makhdoumi, Salman Salamatian, Nadia Fawaz, and Muriel Médard. From the information bottleneck to the privacy funnel. In *2014 IEEE Information Theory Workshop (ITW 2014)*, pages 501–505. IEEE, 2014.

[108] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. Privacy and utility preserving sensor-data transformations. *Pervasive and Mobile Computing*, 63:101132, 2020.

[109] Yuzhu Mao, Zihao Zhao, Guangfeng Yan, Yang Liu, Tian Lan, Linqi Song, and Wenbo Ding. Communication efficient federated learning with adaptive quantization. *arXiv preprint arXiv:2104.06023*, 2021.

[110] Stan Matwin. Privacy-preserving data mining techniques: survey and challenges. In *Discrimination and Privacy in the Information Society*, pages 209–221. Springer, 2013.

[111] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

[112] Ricardo Mendes and João P Vilela. Privacy-preserving data mining: Methods, metrics, and applications. *IEEE Access*, 5:10562–10582, 2017.

[113] Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pages 198–198. IEEE, 2019.

[114] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

[115] Lihao Nan and Dacheng Tao. Variational approach for privacy funnel optimization on continuous data. *Journal of Parallel and Distributed Computing*, 137:17–25, 2020.

[116] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 111–125. IEEE, 2008.

[117] Michael A Nielsen. *Neural networks and deep learning*, volume 2018. Determination press San Francisco, CA, 2015.

[118] Solmaz Niknam, Harpreet S Dhillon, and Jeffrey H Reed. Federated learning for wireless communications: Motivation, opportunities, and challenges. *IEEE Communications Magazine*, 58(6):46–51, 2020.

[119] Gabriel Orsini, Wolf Posdorfer, and Winfried Lamersdorf. Saving bandwidth and energy of mobile and iot devices with link predictions. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–12, 2020.

[120] Amichai Painsky and Naftali Tishby. Gaussian lower bound for the information bottleneck limit. *The Journal of Machine Learning Research*, 18(1):7908–7936, 2017.

[121] Harsh Kupwade Patil and Ravi Seshadri. Big data security and privacy issues in healthcare. In *2014 IEEE international congress on big data*, pages 762–765. IEEE, 2014.

[122] Diego Peteiro-Barral and Bertha Guijarro-Berdiñas. A survey of methods for distributed machine learning. *Progress in Artificial Intelligence*, 2(1):1–11, 2013.

[123] Benny Pinkas. Cryptographic techniques for privacy-preserving data mining. *ACM Sigkdd Explorations Newsletter*, 4(2):12–19, 2002.

[124] Geong Sen Poh, Prosanta Gope, and Jianting Ning. Privhome: Privacy-preserving authenticated communication in smart home environment. *IEEE Transactions on Dependable and Secure Computing*, 2019.

[125] Ismini Psychoula, Erinc Merdivan, Deepika Singh, Liming Chen, Feng Chen, Sten Hanke, Johannes Kropf, Andreas Holzinger, and Matthieu Geist. A deep learning approach for privacy preservation in assisted living. *arXiv preprint arXiv:1802.09359*, 2018.

[126] Zeng-Chang Qin. Roc analysis for predictions made by probabilistic classifiers. In *2005 International Conference on Machine Learning and Cybernetics*, volume 5, pages 3119–3124. IEEE, 2005.

[127] Arun Rajkumar and Shivani Agarwal. A differentially private stochastic gradient descent algorithm for multiparty classification. In *Artificial Intelligence and Statistics*, pages 933–941, 2012.

[128] Shruthi Ramesh and Manimaran Govindarasu. An efficient framework for privacy-preserving computations on encrypted iot data. *IEEE Internet of Things Journal*, 7(9):8700–8708, 2020.

[129] Wang Ren, Xin Tong, Jing Du, Na Wang, Shan Cang Li, Geyong Min, Zhiwei Zhao, and Ali Kashif Bashir. Privacy-preserving using homomorphic encryption in mobile iot systems. *Computer Communications*, 165:105–111, 2021.

[130] Michael D Richard and Richard P Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483, 1991.

[131] Salah Rifai, Grégoire Mesnil, Pascal Vincent, Xavier Muller, Yoshua Bengio, Yann Dauphin, and Xavier Glorot. Higher order contractive auto-encoder. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 645–660. Springer, 2011.

[132] Benjamin IP Rubinstein, Peter L Bartlett, Ling Huang, and Nina Taft. Learning in a large function space: Privacy-preserving mechanisms for svm learning. *arXiv preprint arXiv:0911.5708*, 2009.

[133] Anand D Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE signal processing magazine*, 30(5):86–94, 2013.

[134] Mohamed Seliem, Khalid Elgazzar, and Kasem Khalil. Towards privacy preserving iot environments: a survey. *Wireless Communications and Mobile Computing*, 2018, 2018.

[135] Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30):2696–2711, 2010.

[136] 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. IEEE, 2017.

[137] Nikola Simidjievski, Cristian Bodnar, Ifrah Tariq, Paul Scherer, Helena Andres Terre, Zohreh Shams, Mateja Jamnik, and Pietro Liò. Variational autoencoders for cancer data integration: design principles and computational practice. *Frontiers in genetics*, 10:1205, 2019.

[138] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pages 587–601, 2017.

[139] DJ Strouse and David J Schwab. The information bottleneck and geometric clustering. *Neural computation*, 31(3):596–612, 2019.

[140] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.

[141] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[142] Sergios Theodoridis. *Machine learning: a Bayesian and optimization perspective*. Academic press, 2015.

[143] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[144] Saeed Vahidian, Mahdi Morafah, and Bill Lin. Personalized federated learning by structured and unstructured pruning under data heterogeneity. *arXiv preprint arXiv:2105.00562*, 2021.

[145] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen, and Jan S Rellermeyer. A survey on distributed machine learning. *ACM Computing Surveys (CSUR)*, 53(2):1–33, 2020.

[146] Sahil Verma and Julia Rubin. Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, pages 1–7. IEEE, 2018.

[147] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[148] Slava Voloshynovskiy, Mouad Kondah, Shideh Rezaeifar, Olga Taran, Taras Holotyak, and Danilo Jimenez Rezende. Information bottleneck through variational glasses. *arXiv preprint arXiv:1912.00830*, 2019.

[149] Isabel Wagner and David Eckhoff. Technical privacy metrics: a systematic survey. *ACM Computing Surveys (CSUR)*, 51(3):57, 2018.

[150] Tianlu Wang, Jieyu Zhao, Mark Yatskar, Kai-Wei Chang, and Vicente Ordonez. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5310–5319, 2019.

[151] Yang Wang, Stephen Adams, Peter Beling, Steven Greenspan, Sridhar Rajagopalan, Maria Velez-Rojas, Serge Mankovski, Steven Boker, and Donald Brown. Privacy preserving distributed deep learning and its application in credit card fraud detection. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1070–1078. IEEE, 2018.

[152] Ye Wang, Yuksel Ozan Basciftci, and Prakash Ishwar. Privacy-utility tradeoffs under constrained data release mechanisms. *arXiv preprint arXiv:1710.09295*, 2017.

[153] Jarrod West and Maumita Bhattacharya. Intelligent financial fraud detection: a comprehensive review. *Computers & security*, 57:47–66, 2016.

[154] Alexandra Wood, Micah Altman, Aaron Bembenek, Mark Bun, Marco Gaboardi, James Honaker, Kobbi Nissim, David R O'Brien, Thomas Steinke, and Salil Vadhan. Differential privacy: A primer for a non-technical audience. *Vand. J. Ent. & Tech. L.*, 21:209, 2018.

[155] Yongkai Wu, Lu Zhang, and Xintao Wu. On convexity and bounds of fairness-aware classification. In *The World Wide Web Conference*, pages 3356–3362, 2019.

[156] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of Healthcare Informatics Research*, pages 1–19, 2020.

[157] Ke Xu, Tongyi Cao, Swair Shah, Crystal Maung, and Haim Schweitzer. Cleaning the null space: A privacy mechanism for predictors. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[158] Ronghua Xu and Yu Chen. Fed-ddm: A federated ledgers based framework for hierarchical decentralized data marketplaces. *arXiv preprint arXiv:2104.05583*, 2021.

[159] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[160] Andrew C Yao. Protocols for secure computations. In *23rd annual symposium on foundations of computer science (sfcs 1982)*, pages 160–164. IEEE, 1982.

[161] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180, 2017.

[162] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *Journal of Machine Learning Research*, 20(75):1–42, 2019.

[163] Abdellatif Zaidi, Iñaki Estella-Aguerri, et al. On the information bottleneck problems: Models, connections, applications and information theoretic views. *Entropy*, 22(2):151, 2020.

[164] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International Conference on Machine Learning*, pages 325–333, 2013.

[165] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

[166] Dayin Zhang, Xiaojun Chen, Dakui Wang, and Jinqiao Shi. A survey on collaborative deep learning and privacy-preserving. In *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pages 652–658. IEEE, 2018.

[167] Quanshi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *arXiv preprint arXiv:1802.00614*, 2018.

[168] Yuan Zhang and Sheng Zhong. A privacy-preserving algorithm for distributed training of neural network ensembles. *Neural Computing and Applications*, 22(1):269–282, 2013.

[169] Jianxin Zhao, Richard Mortier, Jon Crowcroft, and Liang Wang. Privacy-preserving machine learning based data analytics on edge devices. 2018.

[170] Nengfeng Zhou, Zach Zhang, Vijayan N Nair, Harsh Singhal, Jie Chen, and Agus Sudjianto. Bias, fairness, and accountability with ai and ml algorithms. *arXiv preprint arXiv:2105.06558*, 2021.