Deep Learning Classification Enhancement and Privacy-Preserving Deep Learning: A Data-Centric Approach

Double-space title (see sample title page in College Guide)

by

Hung Nguyen

There is another PhD student with the same name, so use your middle initial, also: Hung S. Nguyen, here and below.

need space

A dissertationsubmitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Electrical Engineering
College of Engineering
University of South Florida

Need period after 'J.'

Major Professor: Morris J Chang, Ph.D.
Ismail Uysal, Ph.D.
Zhuo Lu, Ph.D.
Lu Lu, Ph.D.
Simon Ou, Ph.D.

Reminder: use the last date signed by major prof or committee member on Cert of Approval.

Date of Approval:
July 7, 2022 (TBD)

Do not use any keyword terms that are already in your title

Keywords: ~~Deep learning~~, machine learning, non-IID data, class imbalnaced data, privacy preservation

IMPORTANT! I do not proofread, but noticed this while doing your format review. All contents should be finalized and thoroughly proofread before uploading to ProQuest/Graduate Studies. No changes can be made after Graduate Studies approval.

Acknowledgments

If your graduate program / research has been funded by an institution, government, etc., this is where you acknowledge them (some students are required to do this).

Format text on this page the same as text in the rest of your manuscript, and use the American spelling of Acknowledgments for the heading.

REMINDER: TOC entries must match the wording and capitalization (all headings must be heading-style capitalization) of the corresponding headings in the text *exactly*.
Check every entry before final submission to Grad Studies/ProQuest!

If entry in TOC runs onto more than one line, indent 2nd line ¼" Fix any instances of this

Capitalization of all headings must be heading-style here and in the text headings (see College Guide sample pages)

Table of Contents

REMINDER: LOT/LOF entries must match the wording and capitalization *of the first sentence* of the corresponding table/figure titles in the text *exactly*.
Sentence = from the first letter to the first non-abbreviation period (full stop).
Check every entry before final submission to Grad Studies/ProQuest!

List of Tables

Use consistent capitalization style for all table titles. Use heading style or sentence style capitalization, but be consistent

Use consistent capitalization style for all figure titles. Use heading style or sentence style capitalization, but be consistent

Familiarize yourself with the format requirements in the College Guide!

Use the *first sentence only* from the table/figure title in the text as the entry in the LOT/LOF. A sentence = from first letter of title to the first non-abbreviation period.

Line spacing must be consistent between all same-level headings and first line below. Use at least double-spacing.

Abstract

Deep Learning and its applications become attractive to a lot of research recently because of its capability to capture important information from large amounts of data. While most of the work are focusing on finding the best model parameters, improving the data itself is still lacking attention. In this work, we propose data processing approaches to enhance the robustness of deep learning classification by improving data quality. For example, our data processing proposals are aimed to alleviate the impacts of class imbalanced data and non-IID data in classification and federated learning scenarios, receptively. In addition, data pre-processing strategies such that dimensionality reduction is also enhanced using deep learning-based techniques for a scenario of data privacy preservation. By conducting several experiments and comparisons, we show that our approaches yield good performance and constantly outperform many state-of-the-art methods.

Chapter 1: Introduction

Deep learning, in the last decades, has become successful in many automatic tasks such as pattern recognition, natural language processing, image and vision computing by extracting and learning from large amounts of data. While most existing works focus on enhancing machine learning models to improve task performance, data quality improvement is still in lack of attention. There are numerous aspects to look for in order to improve data quality such as data augmentation, data dimensionality reduction, data imbalance, missing data, feature extraction. In practice, by improving data quality, we could significantly boost machine learning performance. In this work, we aim to enhance machine learning performance by focusing on data processing techniques to alleviate negative impacts of class imbalance data, data that is not independent and identical distributed (non-IID). Besides, we also propose a deep learning-based dimensionality reduction technique to improve classification accuracy and preserve user privacy at the same time. Proposed techniques are introduced in the following chapters. Specifically, Chapter 2 introduces a technique to tackle the non-IID issue in federated learning which might significantly reduce machine learning performance. Chapter 3 introduce a class balancing technique which generates synthetic data for minor class. This could help deep learning to avoid slow convergence problems and task performance. In Chapter 4, we propose a dimensionality reduction technique based on the state-of-the-art generative models (i.e., AutoEncoder and Generative Adversarial Network) to not only improve classification accuracy but also preserve data privacy.

Chapter 2: Federated Learning for Skewed Distribution Data

2.1 Introduction

Since the demand for massive data in artificial intelligent machines, MacMahan et al. [60] introduced the concept of federated learning (FL) in 2016, which is a collaboratively decentralized learning framework. In contrast to centralized learning approaches in which datasets are sent to an aggregator, FL encourages data holders to contribute without the privacy concern of exposing the raw data. For example, several hospitals holding patient records would participate in a machine learning system to provide better disease predictions to other patients via FL without privacy disclosure. Since then, FL has been seen in various applications in different fields [36, 88, 24, 92]. FL can be categorized into two schemes: cross-silo and cross-device. Cross-device is a setting that is more concerned about participating thousands of devices (e.g., personal phones, sensors, wearable devices) such as an IoT system. Whereas, cross-silo is more concerned about the setting that enables FL to learn from a number of databases, and each of them contain a big trunk of data.

MacMaHan et al. [60] also introduced Federated Averaging (FedAvg) and demonstrated its robustness. The main idea is that clients (data holders) are able to involve a model training process by exchanging local models' weights instead of exchanging raw data. One of the main concerns is FL performance since the data might come from different sources and have different distributions (skewed distribution). Thus, FL performance is significantly reduced because this violates the most common machine learning assumption that data should be independent and identically distributed (IID). In this work, we focus on tackling the non-IID data issue in a federated learning system, in which clients' collected data distributions are skewed. The skewness might be caused by many different reasons. For example,

clients might perform different sampling methods, apply different normalization methods, or sample using different devices. FL over non-IID data has been shown in existing works [97, 73, 43, 56, 79, 81, 52, 94] where its performance deteriorates dramatically than that over the IID data.

Over the past few years, there have been a number of approaches aiming at reducing non-IID data impacts. While many current works focus on the skewed label distribution, there are only limited approaches considering skewed feature distribution data which is very common in various fields, e.g., medical images collected from different x-ray machines. Zhao et al. [95] explained the performance reduction as the problem of weight divergence. They then proposed an alleviation by combining local data with global shared data to train each client. However, it raises the concern of privacy violation with the shared data. Li et al. illustrated in their work (FedBN) [54] that Local Batch Normalization would help to reduce the problem of non-IID data. FedBN suggests clients to not synchronize local batch normalization parameters with the global model. However, we will show in the experiments that by choosing highly skewed data, FedBN performance decreases dramatically. Duan et al. proposed FedDNA [19] which shares statistical parameters of models (means and variances) and aims at finding averaging weights for each client's model to minimize models' weights divergence across clients. However, as this only considers the aggregating weight for each model, the improvement is minor. FedNova [83] suggests to normalized local weights before synchronizing with the aggregator. FedMA [82], AFL [62] and PFNM [91] consider combinations of layer-wise parameters and provide an aggregation of such parameters to alleviate the non-IID issue. However, these suggestions do not directly consider the data distribution skewness at the data level.

To the best of our knowledge, there is still a lack of theoretically comprehensive approach for FL learning that directly tackles skewed feature distribution data at the sample-level. While the problem of distribution skewness in centralized learning (in which the aggregator has access to the entire data) is straightforward to deal with, such issue is more challenging

in federated learning because of the limited access to the raw data. To overcome this, we propose an algorithm that implicitly leverages statistical information of the data over clients to alleviate the distribution skewness issue. Our method only requires clients to exchange additional model weights in a similar way as a typical federated learning system, so that it still preserves a certain level of privacy. The proposed method aims to exchange data statistical information so that it can generate sample weights to adjust distribution skewness at the sample-level. After acquiring the adjusting weights, the machine learning model can be trained under typical federated learning framework. Our contributions are as follows:

1. Provide a theoretical base to deal with skewed distribution data for federated learning. We suggest adjusting sample weights, which is derived from the machine learning empirical risk.

2. Provide a practical solution to mitigate the problem of learning from skewed distribution data for FL framework, but still preserve data privacy. Specifically, a state-of-the-art deep learning method (i.e., MADE) is leveraged to sharing data statistical information without sharing the raw data.

3. Several experiments are conducted in a classification task to evaluate the methods. The results demonstrate that the proposed method outperforms other state-of-the-art methods, and its performance is close to an upper bound performance where privacy is not preserved.

The rest of this chapter is organized as follows. Section 2.2 introduces a neural network-based model is leveraged in our work to carry density information. Section 2.3 introduces our problem in a scenario where clients hold different distribution datasets. Our proposed solution is introduced in Section 2.4. Section 2.6 shows our experimental results and illustrates the proposed method's performance. Section 2.7 summarizes our study and discuss the future work to improve the proposed method.

2.2 Preliminary: Masked Autoencoder for Distribution Estimation (MADE)

The proposed method asks the clients to share additional model weights that carry their local datasets' distribution density information instead of sharing the raw data. We utilize a neural network-based density estimation model, namely, Masked Autoencoder for Distribution Estimation (MADE) [29]. We briefly introduce MADE in this section.

MADE is designed to estimate the probability distribution of input components (e.g., pixels in an image). MADE assumes input components are dependent instead of independent, which is relevant in many applications. MADE decomposes the distribution of an instance $\mathbf{x}$ consisting three components x_1, x_2, x_3 as follows:

$$p(\mathbf{x}) = p(x_2)p(x_3|x_2)p(x_1|x_2, x_3). \tag{2.1}$$

For implementation, MADE poses the constraint on a neural network that each output component in a certain layer only connects to its dependent input components in the previous layer. Masks are created based on such principle, and applied to the weights of the network. Specifically, MADE assigns each unit in a hidden layer an integer m between 1 and $D - 1$, where D is the number of dimensions. Denote $m(k)$ as the maximum number of units in the previous layer to which the k^{th} hidden unit can connect, the weight mask M is then formulated as follows:

$$M_{k,d} = 1_{m(k)\geq d} = \begin{cases} 1 & \text{if } m(k) \geq \text{d} \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

for $d \in \{1, ..., D\}$ and $k \in \{1, ..., K\}$ with K being the number of hidden layer units.

2.3 Scenario

In this section, we introduce and formulate the scenario of federated learning with skewed feature distribution across clients. Our scenario is a learning collaboration between K clients

to build a global classification model that maximizes the global accuracy given arbitrary data. Each client holds a number of individual records that they are not willing to share with others due to privacy concerns. How to prevent the performance of the global network from deteriorating by the distribution skewness issue [53] across clients is the main focus of this work.

We denote the data and associated labels held by client $k \in \{1, ..., K\}$ as $\{(\mathbf{x}_k^i, y_k^i)\}_{i=1}^{N_k}$ where $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{N}$. In stead of learning each model for every client $f(\mathbf{w}_k)$, the objective is to maximize the performance of a global model $g(\mathbf{w})$ that is resilient to data skewness issue.

2.4 Federated Learning for distribution skewed data using sample weights

In this section, we propose a solution to alleviate the negative impact of distribution skewness across clients for federated learning. Mainly, the proposed method aims to find weights for training samples so that it could correct local distributions to global distribution. To achieve this goal, we need to exchange some statistical information between clients and the aggregator in a privacy-preserving manner. The remaining of this section introduces how we design sample weights, how we exchange statistical information without exposing clients' raw data, and how we derive sample weights from achieved information. After achieving weights for the samples, the training process for machine learning tasks is similar to Fedavg [60]. Our framework is illustrated in Figure 2.1. The proposed method, namely FedDisk, requires a 2-step process. First, clients jointly learn a global density estimation model and their local density models. These models are then used to derive sample weights for local training process. Second, the machine learning tasks, e.g., classification, can be leaned in a similar way to conventional FL procedure, with the data skewness issue mitigated by the sample weights in the first step.

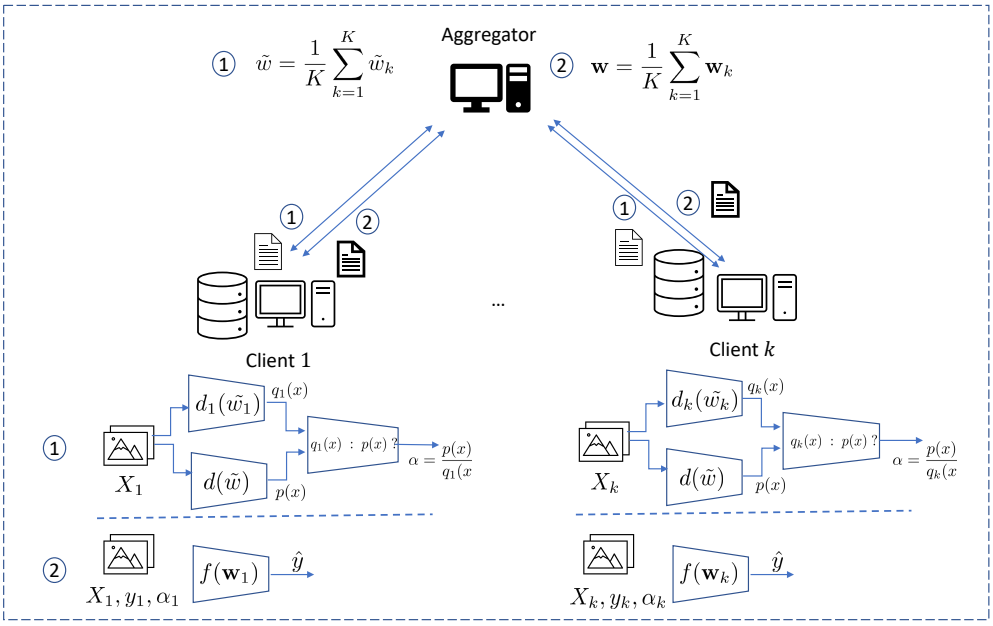


Figure 2.1: FedDisk Framework: The proposed framework has two steps. First, local and global probability densities ($p(x), q(x)$) are estimated via MADE models leveraging federated learning procedures, so that clients do not have to expose their raw data. Then, the sample weights α is computed by approximating density ratio via class probability estimation. Second, the machine learning tasks (e.g., classification) can be performed similarly to a typical federated learning method (e.g., Fedavg) with the sample weights acquired from step 1.

2.4.1 Sample Weights Design

As we do not have sufficient information about the true distribution, we consider the combination of all clients' dataset distribution as our true distribution. Thus, we consider the pdf of the true distribution as

Use consistent line-spacing above all equations and consistent line-spacing below all equations. Use at least double-spacing

$$p(\mathbf{x}) = \sum_{k=1}^{K} q_k(\mathbf{x}), \tag{2.3}$$

ts the pdf of the data distribution of the i^{th} client.

To jointly learn a global model, the system aims at finding the expectation of the loss function $l(g(\mathbf{x}), y)$ with sample $\mathbf{x}$ that drawn from the true distribution. The expected loss is formulated by the associated risk [46] as follows:

$$\mathbb{E}[l(g(\mathbf{x}), y)] = \iint l(g(\mathbf{x}), y)p(\mathbf{x}, y)d\mathbf{x}dy \tag{2.4}$$

$$= \iint l(g(\mathbf{x}), y)p(y|\mathbf{x})p(\mathbf{x})d\mathbf{x}dy, \tag{2.5}$$

where $p(\mathbf{x}, y)$ is the joint pdf of a sample $\mathbf{x}$ and its associated label y, and $p(y|\mathbf{x})$ is the conditional probability of a label y given a sample $\mathbf{x}$. We also assume that for any client $k \in \{1, ..., K\}$ with local data distribution $q_k(\mathbf{x})$, the conditional probability of a label y given a sample $\mathbf{x}$ is equivalent to that of the true distribution, namely

$$q_k(y|\mathbf{x}) = p(y|\mathbf{x}). \tag{2.6}$$

From Equation 2.3, 2.5, 2.6, and by multiplying with factor $\frac{q_k(\mathbf{x})}{q_k(\mathbf{x})} = 1$, the expected loss in Equation 2.5 can be expanded as follows:

$$\mathbb{E}[l(g(\mathbf{x}), y)] = \iint l(g(\mathbf{x}), y)p(y|\mathbf{x})p(x)d\mathbf{x}dy, \tag{2.7}$$

$$= \iint l(g(\mathbf{x}), y)p(y|\mathbf{x})\frac{q_k(\mathbf{x})}{q_k(\mathbf{x})}p(\mathbf{x})d\mathbf{x}dy \tag{2.8}$$

$$= \iint l(g(\mathbf{x}), y)q_k(y|\mathbf{x})\frac{q_k(\mathbf{x})}{q_k(\mathbf{x})}p(\mathbf{x})d\mathbf{x}dy \tag{2.9}$$

$$= \iint l(g(\mathbf{x}), y)q_k(\mathbf{x}, y)\frac{p(\mathbf{x})}{q_k(\mathbf{x})}d\mathbf{x}dy. \tag{2.10}$$

The objective of the global model thus amounts to minimize the empirical risk over all K clients' datasets:

$$\sum_{k=1}^{K}\frac{1}{N_K}\sum_{j=1}^{N_K}\alpha_k^j\, l(g(\mathbf{x}_k^j), y_k^j. \tag{2.11}$$

For simplicity, we assume clients hold the same number of samples $N_1 = ... = N_K$, Equation 2.11 then becomes

$$\boxed{\underset{g}{\text{minimize}}\frac{1}{N}\sum_{k=1}^{K}\sum_{j=1}^{N/K}\alpha_k^j\, l(g(\mathbf{x}_k^j), y_k^j),} \tag{2.12}$$

where $\mathbf{x}_k^j$ and y_k^j are the j^{th} sample and its label in the k^{th} client's dataset. α_k^j is the corresponding sample weight computed as

$$\boxed{\alpha_k^j = \frac{p(\mathbf{x})}{q_k(\mathbf{x})} = \frac{\sum_{i=1}^{K} q_i(\mathbf{x})}{q_k(\mathbf{x})}.}. \tag{2.13}$$

Our problem becomes minimizing the loss function (Equation 2.12) over all samples with corresponding sample weights α_k^j. The sample weights could be estimated as the density ratio between the true distribution (global distribution) and the client distributions (local distributions). For each client, the local distribution can be estimated straightforward. However, the challenge is to achieve the true distribution without having access to data from

other clients. To solve this, we leverage a neural network-based density estimation model to estimate the global density by a federated learning procedure. Thus, clients can implicitly exchange some statistical information, while still preserving the privacy in client data.

2.4.2 Probability Density Approximation

To estimate global density and preserve client privacy at the same time, we propose to leverage neural network-based density estimation so that we can exchange local density information with the aggregator without sharing the raw data. In this work, we leverage a well-known method, namely, Masked Autoencoder for Distribution Estimation (MADE, [29]), which is briefly reviewed in Section 2.2. Elaborately, each client aims to estimate its local probability density $q_k(\mathbf{x})$ using its own dataset, and all K clients jointly estimate the global probability density $p(\mathbf{x}) = q_1(\mathbf{x}) + ... + q_K(\mathbf{x})$. Learned MADE models are used to approximate local probability density functions, and the global MADE model approximates the global probability density. The learning process is described as follows.

To begin with, the system learns a local density estimation model $d_k(\tilde{w}_k)$ for the k^{th} client ($k = 1, ..., K$) and jointly learn a global density estimation model $d(\tilde{w})$ using framework similarly to Fedavg, where $\tilde{w}_k$ and $\tilde{w}$ are local and global density estimation model's parameters. Specifically, each client train a MADE network on its dataset for a certain number of iterations, and then model parameters are sent to an aggregator for aggregation. After aggregating all clients' model parameters, the aggregator shares global model parameters to all clients. The steps are repeated until reaching a desired number of global iterations T_1. The global MADE model aggregation from K clients at iteration t can be described as follows:

$$\tilde{w}^t = \frac{1}{K} \sum_{k=1}^{K} \tilde{w}_k^t \qquad t = 1, ..., T_1. \tag{2.14}$$

2.4.3 Sample Weight Approximation

After achieving local and global density approximations by MADE models (Section 2.4.2), we can estimate sample weights in Equation 2.13. Since MADE models output vectors of conditional probabilities for each element in the d-dimensional input $\mathbf{x}$ (e.g., $p(x_1|x_2, x_3)$, $p(x_2)$, $p(x_3|x_2)$, an intuitive way to compute $p(\mathbf{x})$ is by multiplying all the conditional probabilities (e.g., $p(\mathbf{x}) = p(x_1|x_2, x_3)p(x_2)p(x_3|x_2)$). However, as $p(\mathbf{x})$ vanishes whenever any of the conditional probabilities vanishes, we instead keep the output as a vector of conditional probabilities (same size as input) and approximate the density ratio in Equation 2.13 using a class probability estimation method inspired by [61]. The method aims at training a binary classifier to output a probability which representing the ratio between $p(\mathbf{x})$ and $q(\mathbf{x})$. The solution detail is described in the rest of this subsection.

After updating global MADE models to the clients, each client trains its own local MADE model with its local dataset. The local training data X_k is then fed into global MADE and local MADE to estimate $p(\mathbf{x})$ and $q_k(\mathbf{x})$, respectively. Denote $\mathbf{u}$ as the output vector of density estimation models, and l be the pseudo label indicating whether it is sampled from the global destination ($l = 1$) or the local distribution ($l = 0$). Each client then trains a binary classifier to differentiate whether the output $\mathbf{u}$ comes from $p(\mathbf{x})$ or $q_k(\mathbf{x})$. Outputs of the two MADE models (the sample size of each output is N_k) are concatenated to a new vector dataset including samples $\{(\mathbf{u}_k^i, l_k^i)\}_{i=1}^{2N_k}$, and is used to train the binary classifier. The conditional probabilities of the binary classification model $h(\mathbf{u}, w_h)$ (where $\mathbf{u}$ is the input variable , w_h is the model parameter) can be approximated as following:

$$\mathcal{P}(\mathbf{u}|l = 0) \propto q_k(\mathbf{x}), \quad \mathcal{P}(\mathbf{u}|l = 1) \propto p(\mathbf{x}). \tag{2.15}$$

From Bayes' rule, we have

$$\frac{p(\mathbf{x})}{q(\mathbf{x})} = \frac{\mathcal{P}(\mathbf{u}|I=1)}{\mathcal{P}(\mathbf{u}|I=0)} \tag{2.16}$$

$$= \left(\frac{\mathcal{P}(I=1|\mathbf{u})\mathcal{P}(\mathbf{u})}{\mathcal{P}(I=1)}\right)\left(\frac{\mathcal{P}(I=0)}{\mathcal{P}(I=0|\mathbf{u})\mathcal{P}(\mathbf{u})}\right) \tag{2.17}$$

$$= \frac{\mathcal{P}(I=1|\mathbf{u})\mathcal{P}(I=0)}{\mathcal{P}(I=0|\mathbf{u})\mathcal{P}(I=1)}. \tag{2.18}$$

We approximate the marginal probability ratio by the number of samples from the two distributions N_k where N_k is k^{th} client's training sample size over the concatenated dataset size ($2N_k$). We have

$$\frac{\mathcal{P}(I=0)}{\mathcal{P}(I=1)} = \frac{N_k}{2N_k}\frac{2N_k}{N_k} = 1. \tag{2.19}$$

The density ratio can be estimated as follows:

remove comma and space

$$\frac{p(\mathbf{x})}{q(\mathbf{x})} = \frac{\mathcal{P}(I=1|\mathbf{u})}{\mathcal{P}(I=0|\mathbf{u})} = \frac{\mathcal{P}(I=1|\mathbf{u})}{1-\mathcal{P}(I=1|\mathbf{u})}. \tag{2.20}$$

, where $\mathcal{P}(I=1|\mathbf{u})$ is the classifier's probability-liked output indicating how likely an input vector $\mathbf{u}$ comes from the global probability $p(\mathbf{x})$.

Eventually, j^{th} local training sample of client k, $\mathbf{x}_k^j$, is fed into local MADE model to achieve its corresponding density estimation $\mathbf{u}_k^j$. $\mathbf{u}_k^j$ is then fed into the binary classification function $h(\mathbf{u})$ to achieve the class probability $\mathcal{P}(I=1|\mathbf{u}_k^j)$. This is used to estimate the sample weight α_k^j based on Equation 2.20.

2.4.4 Collaborative Learning With Skewed Distribution Data Across Clients

After acquiring sample weights, each client starts to train the final model with their own dataset and corresponding sample weights for a machine learning task (e.g., classification) as a typical federated learning framework. In this work, we follow the procedure introduced by Fedavg to learn the global model. The aggregator aggregates clients' local models as follows:

$$\mathbf{w}^t = \frac{1}{K}\sum_{k=1}^{K}\mathbf{w}_k^t \qquad t = 1, ..., T_2, \tag{2.21}$$

where T_2 is desired number of global iterations.

2.5 Ablation Study: An Upper Bound Method

In this section, we study a setting that is similar to FedDisk, but the method does not use the density estimation model (MADE) to find sample weights. Instead of feeding MADE outputs ($\mathbf{u}$) to the binary classifier introduced in 2.4.3, this method uses directly the raw data as the binary classifier inputs to estimate the density ratio ($p(\mathbf{x})/q(\mathbf{x})$) and computes sample weights α according to the binary classifier output as introduced in Section 2.4.3.

Instead of sampling from the global distribution $p(\mathbf{x})$ and $q_k(\mathbf{x})$, a client k can use the combination of datasets X (size of $N = N_1 + ... + N_K$) and its own local dataset X_k (with the size of N_k) to estimate $p(\mathbf{x})$ and $q_k(\mathbf{x})$, respectively. However, for the marginal probability ratio in Equation 2.19 holds true, the two datasets should be the same size; we randomly sample N_k samples from X so that the number of samples from $p(\mathbf{x})$ is equivalent to the number of samples coming from $q_k(\mathbf{x})$. The two sets of data are then used to train the binary classifier to estimate the sample weights as similar as the approximation in 2.20. We name this method as UpperBound and consider this method performance as the upper bound to compare since this directly use the raw data to estimate the sample weights (privacy is not preserved).

2.6 Experiments

In this section, we conduct several experiments to evaluate the proposed method on non-IID federated learning scenarios with two image data (digits and chest xray). Our FL system gold is to learn a global classifier leveraging all data from clients. The classification accuracy is used as a metric to evaluate the performance of proposed method. We also compare our method with Fedavg and FedBN. The mutual setting in both scenarios is described as follows.

Density estimation models are constructed with 4 hidden layers, each with 1000 neurons. We use a shallow, fully connected neural network to discriminate the density estimation output vectors coming from which of the two distribution density functions $p(\mathbf{x})$ and $q_k(\mathbf{x})$. The model contains 2 hidden layers and 100 neurons in each layer. For all neural netowrk models, we choose Relu as the activation function for all hidden layers, and Softmax for the output layer. All models apply learning rate of 0.001 and Adam optimization were used in training process. For each result, we repor trials.

For code and variables, you can use a different font type/size per your field, but for regular numbers and text, use the exact same font type/size throughout your dissertation

2.6.1 Digits Data

In this section, we implement our met mage classification datasets, i.e., MNIST [17], USPS [42], Synthetic Dig 6], SVHN [64]. The datasets contain digit images with 10 classes (from olor with different sizes. For consistency, we resize all the images to the size of 28×28 pixels, randomly sample 7437 images for training and 1859 images for testing.

2.6.1.1 Setting

We conduct experiments for 2 FL system settings, one includes 5 clients and another includes 3 clients. Each client holds one distinct dataset. In the 3-client setting, clients hold the most challenging and perhaps highly skewed distribution datasets (the 3 color datasets

with messed background, i.e., Synthetic Digits, MNIST_M, SVHN). They collaboratively learn a 10-classes classification model to classify 10 digits.

For the main federated learning classification task, we use a convolutional neural network [66] with 2 convolutional layers containing 32 and 16 3×3 filters, and one 32-neuron fully connected neural layer. The output layer has 10 neurons corresponding to 10 classes. Local models are trained 1 iteration before synchronized with the aggregator, and the number of global iterations is set to 50.

2.6.1.2 Experiment Result

Table 2.1 shows the classification accuracy of the global model over different test sets. Overall, FedDisk outperforms other methods and close to the upper bound. For example, in the 5-client setting, FedDisk with accuracy of 83.48% outperforms others (Fedavg, FedBN) with an improvement of about 2%, and its performance is very close to the upper bound at 83.77%. Additionally, FedDisk not only improves the average accuracy, but Table 2.1 also yields better performance when testing on individual datasets. For example, accuracies for MNIST, SynthDigits, MNIST_M and SVHN are higher than that of Fedavg and FebBN. In the 3-client setting, which includes high skewed distribution datasets, the pattern of FedDisk performance still holds. Its average accuracy is close to UpperBound's and outperforms others. In this challenge, FedBN performs worse than Fedavg.

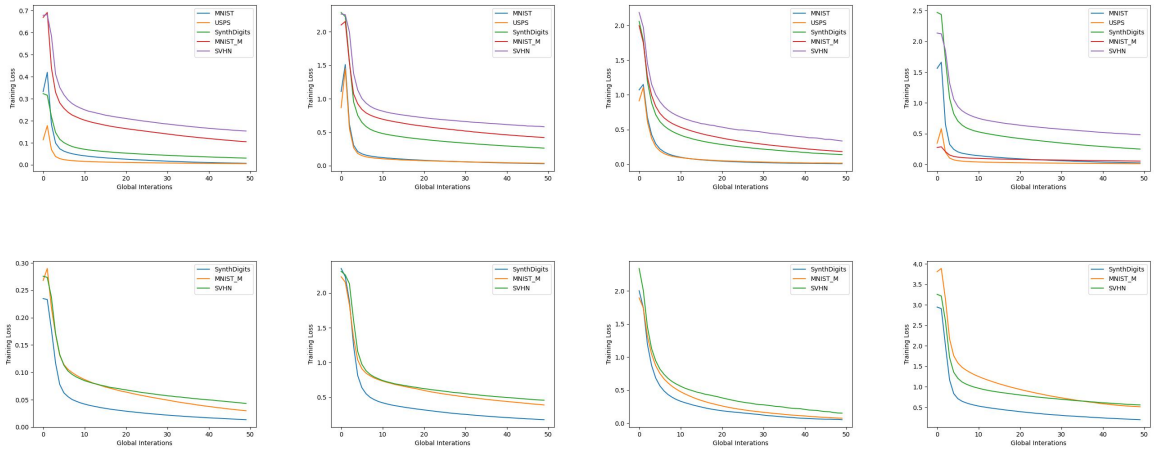
Figure 2.2 illustrates the training loss measured on different datasets. As the Figure

Use at least 3 line-spaces between table titles and text.

ifically, the model in 5-client setting can learn well on synthetic digit data (green lines) so that the loss of synthetic digits is converged faster than that of FedBN and Fedavg.

Table 2.1: Global model accuracy (%) on different testing datasets and average.

Settings	5 Clients						3 Clients			
Method/Data	SynthDigits	MNIST_M	SVHN	MNIST	USPS	**Average**	SynthDigits	MNIST_M	SVHN	**Average**
FedDisk	83.66	73.33	68.26	97.42	94.73	**83.48**	87.78	73.41	76.33	**79.17**
Fedavg	82.15	70.11	63.92	96.72	95.07	**81.59**	87.26	72.72	74.09	**78.02**
FedBN	82.90	70.73	64.06	96.13	94.54	**81.67**	86.72	67.47	74.98	**76.39**
UpperBound	84.41	72.71	68.87	97.29	95.59	**83.77**	88.57	72.94	77.78	**79.76**



(a) FedDisk (b) Fedavg (c) FedBN (d) Upper Bound

Figure 2.2: Training losses measured on two settings. The first row shows the setting including 5 clients where each client holds a dataset, and the second row shows the result for the system with 3 clients. The second setting includes the most challenging datasets (Synthetic Digits, SVHN, MNIST_M) among 5 datasets (Synthetic Digits, SVHN, MNIST_M, USPS, MNIST).

2.6.2 Chest-Xray Data

In this section, we conduct experiment on 3 chest-xray image datasets which contain pneumonia and normal chest xray images, i.e., COVID-19 [14], Shenzhen Hospital [44], and an dataset from University of California San Diego (UCSD) [47]. All the images are converted to grey and resized to 64×64 pixel. The training and testing set are randomly selected from the datasets with the euquivalent sizes at 300 and 100, respectively.

2.6.2.1 Setting

There are 3 clients in our FL system which hold 3 datasets. They collaboratively learn a classification to classify pneumonia and normal iamges. For the classification, we use a convolutional neural network with 2 convolutional layers containing 64 and 32 3×3 filters, and one 200-neuron fully connected neural layer. The output layer has 2 neurons corresponding to 2 classes, pneumonia and normal. The number of local and global iterations are set to 1 and 15, respectively.

2.6.2.2 Experiment Result

Table 2.2 shows the global model accuracy over 3 datasets. Overall, the average accuracy of FedDisk at 78.95% is approximately 7% higher than that of Fedavg and FedBN, and it is close to the upper bound at 79.37%

Table 2.2: Global Model Accuracy (%) on 3 chest-xray image datasets

Method/Data	Shenzhen	COVID19	UCSD	Average
FedDisk	80.49	81.25	75.11	**78.95**
Fedavg	82.15	70.11	63.92	**72.06**
FedBN	74.30	73.06	69.61	**72.32**
UpperBound	80.12	80.30	77.70	**79.37**

2.7 Conclusion

In this work, we have proposed a federated learning method to tackle the issue of distribution skewed data. The method utilizes federated learning framework and a neural network-based density estimation model to derive training sample weights. Thus, this helps to correct the distribution of local clients without revealing their raw data. However, the method works better for the cross-silo setting but have not been investigated for the cross-device setting. We will continue to study this problem in the cross-silo setting and improve our method to tackle the issue in both federated learning categories.

Chapter 3: Synthetic Information Towards Maximum Posterior Ratio for Deep Learning on Class Imbalanced Data (SIMPOR)

3.1 Introduction

Data imbalance is a common phenomenon; it could be caused by sampling procedures or simply the nature of data. For example, it is difficult to sample some of the rare diseases in the medical field, so collected data for these are usually significantly less than that for other diseases. This leads to the problem of class imbalance in machine learning. The chance of rare samples appearing in model training processes is much smaller than that of common samples. Thus, machine learning models will be dominated by the majority class; this results in a higher prediction error rate. Existing work also observed that imbalanced data cause a slow convergence in the training process because of the domination of gradient vectors coming from the majority class [87, 57]. In the last decades, a number of techniques have been proposed to soften the negative effects of data imbalance on conventional machine learning algorithms by analytically studying particular algorithms and developing corresponding strategies. However, the problem for heuristic algorithms such as deep learning is often more difficult to tackle. In this work, we address data imbalance for deep learning models by providing a solution that utilizes both deep active learning techniques and statistical derivations of Bayes' Theorem.

We categorize existing solutions into model-centric and data-centric approaches in which the first approach aims at modifying machine algorithms, and the latter looks for data balancing techniques, respectively. Perhaps data-centric methods are more commonly used because they do not tie to a specific model. In this category, a simple data balancing technique is to duplicate minority instances to balance the sample quantity between classes.

This can preserve the best data structure and reduce the negative impact of data imbalance to some degree. However, this puts too much weight on a very few minority samples; as a result, it causes over-fitting problems in deep learning when the imbalance ratio becomes higher.

Another widely-used method in this category is Synthetic Minority Oversampling Technique (SMOTE) [11], which randomly generates synthetic data on the connections (in Euclidean space) between minority samples. However, this easily breaks data topology, especially in high-dimensional space, because it can accidentally connect instances that are not supposed to be connected. In addition, if there are minority samples located in the majority class, the method will generate sample lines across the decision boundary, which leads to distorted decision boundaries and misclassification. To improve SMOTE, Hui Han, *et al.* [35] proposed a SMOTE-based method (Borderline SMOTE), in which they only apply SMOTE on the near-boundary samples determined by the labels of their neighbors. For example, if a sample Euclidean space-based group includes samples from other classes, they can be considered as samples near the border. Since this method is entirely based on Euclidean distance from determining neighbors to generating synthetic data, it performs poorly in high dimensional space. Similar to SMOTE, if there is any mis-generated sample near the boundary, it will worsen the problem due to synthetic samples bridges across the bounder. Leveraging the same way as SMOTE generates synthetic samples, another widely-used technique, ADASYN [37], controls the number of generated samples by the number of samples in different classes within small groups. Again, this technique still suffers distortion of the decision boundary in the case the boundary region is imbalanced.

To alleviate the negative effects of data imbalance and avoid the drawbacks of existing techniques, we propose a minority oversampling technique that focuses on balancing at the informative region that provides the most important information to the deep learning models. Besides, the technique enhances the chance that synthetic data fall into the minority class

so that it will not cause more errors to the model. By carefully generating synthetic data near minority samples, our proposed technique also preserves the best data topology.

To find informative samples, we leverage an entropy-based deep active learning technique that is able to select samples yielding high entropy to deep learning models. We denote where the informative samples are located as the informative region. We then balance this region first and the remaining data are balanced later so that it would reduce the decision distortion mentioned earlier. For each minority sample in this region, we safely generate its synthetic neighbors so that the global data topology is still preserved. However, generating synthetic samples in this region is critical because it can easily fall across the decision boundary. Therefore, we design a direction to generate synthetic samples that maximize their posterior probability of belonging to the minority class based on Bayes's Theorem. However, maximizing the posterior probability is facing infeasible computation in the denominator. To overcome this, we maximize the posterior ratio instead, so that the denominator will disappear. This also ensures that the synthetic samples are not only close to the minority class but also far from the majority class. The remaining data are eventually balanced by randomly generating neighbors for each sample.

The proposed technique results in a balanced dataset that improves the training performance and alleviates the class imbalance problem. Our experiments indicate that we can achieve better classification results over widely-used techniques in all experimental cases by applying the proposed strategy.

Our work has the following main contributions:

1. Exploring the impact of class imbalance on deep learning.

2. proposing a minority oversampling technique, namely Synthetic Information towards Maximum Posterior Ratio, to balance data classes and alleviate data imbalance impacts. Our technique is enhanced by following key points.

(a) Leveraging an entropy-based active learning technique to prioritize the region that needs to be balanced. It is the informative region where samples provide high information entropy to the model.

(b) Leveraging Maximum Posterior Ratio and Bayes's theorem to determine the direction to generate synthetic minority samples to ensure the synthetic data fall into the minority class and not fall across the decision boundary.

(c) Approximating the likelihood in the posterior ratio using kernel density estimation, which can approximate a complicated statistical model. Thus, the proposed technique is able to work with large, distributively complex data.

(d) Carefully generating synthetic samples surrounding minority samples so that the global data topology is still preserved.

The rest of this chapter is organized as follows. Section 3.2 introduces related concepts that will be used in this work, i.e., Imbalance Ratio, Macro F1-score, and Entropy-based active learning. Section 3.3 will provide more detail on the problem of learning from an imbalanced dataset. Our proposed solution to balance dataset, Synthetic Information towards Maximum Posterior Ratio, will be explained comprehensively in Section 3.5. Section 3.6 discusses the technique's implementation and complexity. We will show experiments on different datasets, including artificial and real datasets in Section 3.7. We also discuss experimental results in the same section. In Section 3.4, we briefly review other existing works. Section 3.8 concludes the work and discusses future work.

3.2 Preliminaries

In this section, we introduce related concepts that will be utilized in our work.

3.2.1 Imbalance Ratio (IR)

For binary classification, we use imbalance ratio (IR) to depict the data imbalance as it has been widely used. IR is the ratio of the majority class samples to the minority class's samples. For example, if a dataset contains 1000 class-A samples and 100 class-B samples, the Imbalance Ratio is 10:1.

3.2.2 F1 Score

In this work, we evaluate balancing data techniques by the classification results on balanced data. To measure the accuracy of classification, we use Macro-averaging F1-Score, in which we compute F1 scores per class and average with the same weight regardless of how often they appear in the dataset. The F1 score is computed based on two factors Recall and Precision as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

$$F1 - score = \frac{2 * Recall * Precision}{Recall + Precision}, \tag{3.3}$$

where T and F stand for True and False; P and N stand for Positive and Negative.

3.2.3 Entropy-based Active Learning

To find informative samples, we leverage entropy-based active learning. The method gradually selects batch-by-batch samples that provide high information to the model based on information entropy theory [77]. The information entropy is quantified based on the "surprise" to the model in terms of class prediction probability. Take a binary classification for example, if a sample is predicted to be 50% belonging to class A and 50% belonging to

class B, this sample has high entropy and is informative to the model. In contrast, if it is predicted to be 100% belonging to class A, it is certain and gives zero information to the model. The class entropy E for each sample can be computed as follows.

$$E(x, \theta) = -\sum_{i}^{n} P_\theta(y = c_i|x) \log_n P_\theta(y = c_i|x) \tag{3.4}$$

where $P_\theta(y = c_i|x)$ is the probability of data x belonging to the ith class of n classes with current model parameter θ.

In this work, we consider a dataset containing N pairs of samples X and corresponding labels y, and a deep neural network with parameter θ. At the first step $t^{(0)}$, we train the classifier with parameter $\theta^{(0)}$ on a random batch of k labeled samples and use the $\theta^{(0)}$ to predict the labels for the rest of the data (we assume their labels are unknown). We then compute the prediction entropy of each sample based on Equation 3.4. We are now able to collect the first batch of informative samples by selecting k samples based on the top k highest entropy. We query labels for this batch and concatenate to existing labeled data to train the classifier parameter $\theta^{(1)}$ in the next step $t^{(1)}$. Steps are repeated until all sample's entropy are less than a threshold e.g., $Threshold = 0.7$.

3.3 The Problem of Learning From Imbalanced Datasets

In this section, we review the problem of learning from imbalanced datasets. Although the problem may apply to different machine learning methods, we focus on deep learning in this work.

Figure 3.1 illustrates our problem on a binary classification. The imbalance in the informative region (light blue eclipse) could lead to separation errors. The dashed green line depicts the expected boundary, while the solid blue line is the model's boundary. Since the minority class is lacking data in this region, the majority class will dominate the model even with a few noisy samples, and this leads to a shift of the model's boundary. In contrast to

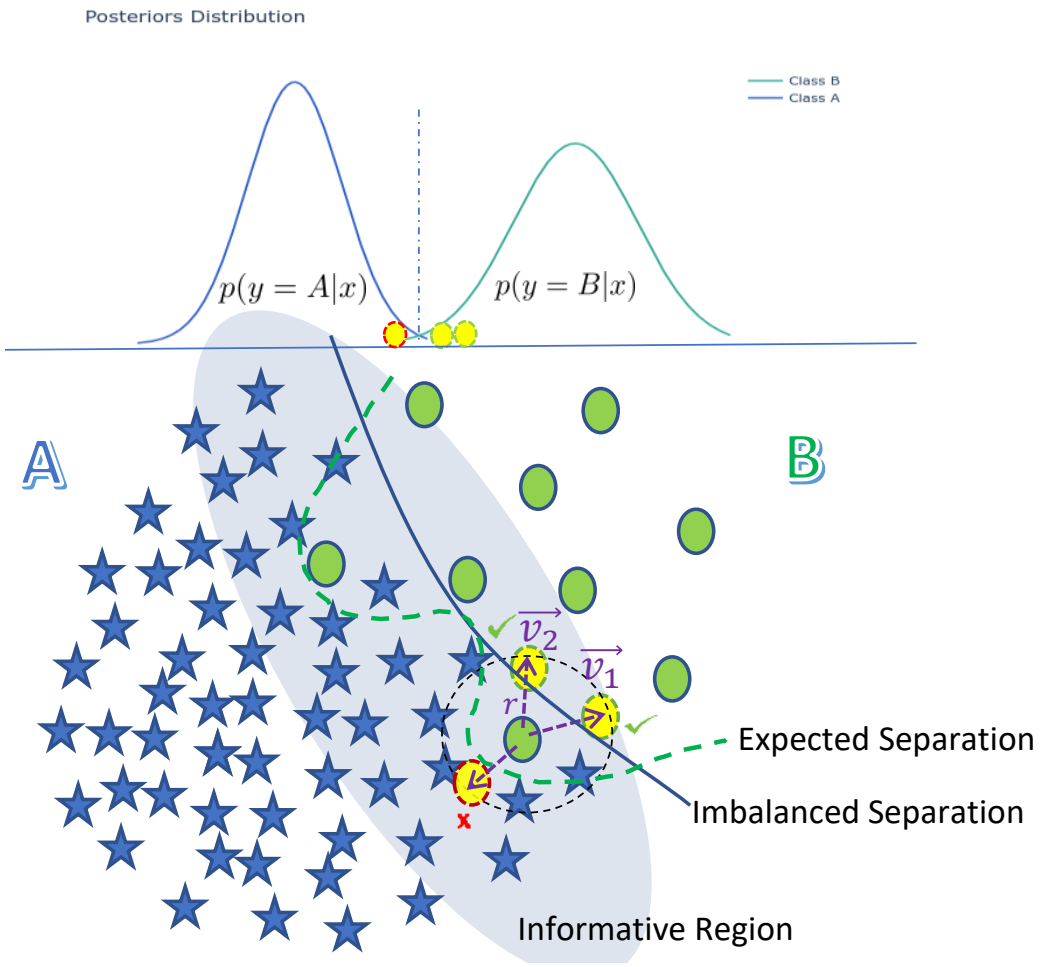


Figure 3.1: Learning from imbalanced datasets.

the study by Ertekin *et al.* [22] which assumes the informative region is more balanced by nature and proposes a solution that only classifies over the informative samples, our assumption is different. We consider the case that the informative region contains high imbalanced data, which we believe happens in most of the real scenarios. In a more complex setting such as high dimensional and topologically complex data, the problem could be more severe. Therefore, we proposed a technique to tackle the problem of data imbalance by oversampling the minority class in an informative manner. The detail of our proposed technique will be described in Section 3.5.

3.4 Related Work

In the last few decades, there have been a number of solutions proposed to alleviate the negative impacts of data imbalance in machine learning. However, many of them are not efficient when it comes to high-dimensional data and deep learning. In this section, we review algorithms that aim at deep learning and strategies inherited from conventional machine learning methods. These techniques can mainly be categorized into two different categories, i.e., data-centric and model-centric approaches.

Model-centric approaches usually require modifications of algorithms on the cost functions in order to balance the weight of each class. Specifically, such cost-sensitive approaches put higher penalties on majority classes and less on minority classes to balance their contribution to the final cost. For example, [15] provided their designed formula $(1-\beta^n)/(1-\beta)$ to compute the weight of each class based on the effective number of samples n and a hyper-parameter β which is then applied to re-balance the loss of a convolutional neural network model. [41],[70], [59] assign classes' weights inversely proportional to sample frequency appearing in each class.

Compared to model-centric-based manners, data-centric approaches have been attracting more research attention as it is independent of machine learning algorithms. In this category, we divide into two main approaches, i.e. sampling-based and generative approaches. Sampling-based methods [78], [27], [34], [51], [23] mainly generate a balanced dataset by either over-sampling minority classes or down-sampling majority classes. Some methods are not designed for deep learning, but we still consider them since they are independent of the machine learning model architecture. In a widely used method SMOTE [11], Chawla *et al.* attempt to oversampling minority class samples by connecting an sample to its neighbors in feature space and arbitrarily drawing synthetic samples along the connections. However, one of the drawbacks of SMOTE is that if there are samples in the minority class located in the majority class, it creates synthetic sample bridges towards the majority class [32]. This renders difficulties in differentiation between the two classes. Another SMOTE-based work

namely Borderline-SMOTE [35] was proposed in which its method aims to do SMOTE with only samples near the border between classes. The samples near the border are determined by the labels of its k distance-based neighbors (if more than half of neighbors belong to the other class, the sample is considered to be on the border). This "border" idea is similar to ours to some degree. However, finding a good k is critical, and it is usually highly data-dependent. In addition, Borderline-SMOTE again faces the problems of SMOTE.

Under the down-sampling category, other works [22], [4] leverage active learning techniques to find informative samples which authors believe the imbalance ratio in these areas is much smaller than that in the entire dataset. They then classify this small pool of samples to improve the performance and expedite the training process for the SVM-based method. however, this method was only designed for SVM-based methods which mainly depend on the support vectors. Also, this potentially discards important information of the entire dataset because only a small pool of data is used.

Generative approaches which generate synthetic samples in minor classes by sampling from data distribution are becoming more attractive as they are outperforming other methods in high dimensional data [55]. When it comes to images, a number of deep learning generative-based methods have been proposed as deep learning is capable of capturing good image representations. [71] [16] [63] utilized Variational Autoencoder as a generative model to arbitrarily generate images from learned distributions. However, most of them assumed simple prior distributions such as Gaussian for minor classes, they tend to simplify data distribution and might not succeed in sophisticated distributions. Our solution also falls into this category; however, we leverage the idea of a mixture model to tackle this issue for image data.

3.5 Synthetic Information towards Maximum Posterior Ratio

To alleviate the negative effects of data imbalance, we propose a comprehensive approach, Synthetic Information towards Maximum Posterior Ratio (SIMPOR), which aims to generate

synthetic samples for minority classes. We first find the informative region where informative samples are located and balance this region by creating synthetic surrounding neighbors for minority samples. The remaining region is then fully balanced by arbitrarily generating minority samples' neighbors. We elaborate on how our strategy is developed in the rest of this section.

3.5.1 Methodology Motivation

As Chazal and Michel mentioned in their work [50], the natural way to highlight the global topological structure of the data is to connect data points' neighbors; our proposed method preserves their observation but in the reverse procedure, generating surrounding synthetic neighbors for minority samples. Thus, our method not only generates more data for minority class but also preserve the underlying topological structure of the entire data.

Agreeing with the mutual idea in [22] and [4], we believe that informative samples play the most important role in the prediction success of both traditional machine learning models (e.g., SVM, Naive Bayes) and modern deep learning approaches (e.g., neural network). Thus, our method finds these informative samples and focuses on augmenting minority data in this region. In this work, we apply an entropy-based active learning strategy mentioned in 3.2.3 to find the samples that maximize entropy to the model. This strategy is perhaps the most popular active learning technique and over-performs many other techniques on several datasets [30], [69] [75].

3.5.2 Generating minority synthetic data

A synthetic neighbor x' and its label y' can be created surrounding a minority sample x by adding a small random vector v to the sample, $x' = x + v$. This lays on the d-sphere surface centered by x, and the d-sphere's radius is set by the length of vector $\vec{v}$, $|\vec{v}|$. It is, however, critical to generate synthetic data in the informative region because synthetic samples can unexpectedly jump across the decision boundary. This can be harmful to the model as this

might create outliers and reduce the model's performance. Therefore, we safely find vector $\vec{v}$ towards the minority class such as $\vec{v}_0$ and $\vec{v}_1$ depicted in Figure 3.1. Our technique is described via a binary classification scenario as follows.

Let consider a binary classification problem between majority class A and minority class B. From the Bayes' theorem, the posterior probabilities $p(y' = A|x')$ or $p(y' = B|x')$ can be used to present the probabilities that a synthetic sample x' belongs to class A or class B, respectively. Let the two posterior probabilities be f_0 and f_1; they can be expressed as follows.

$$p(y' = A|x') = \frac{p(x'|y' = A)\, p(A)}{p(x')} = f_0 \tag{3.5}$$

$$p(y' = B|x') = \frac{p(x'|y' = B)\, p(B)}{p(x')} = f_1 \tag{3.6}$$

As mentioned earlier, we would like to generate each synthetic data x' that maximizes the probability of x' belonging to the minority class B and minimizes the chance x' falling into the majority class A. Thus, we propose a method that maximizes the fractional posterior f,

$$f = f_1/f_0 \tag{3.7}$$

$$= \frac{p(x'|y' = B)\, p(B)}{p(x'|y' = A)\, p(A)}. \tag{3.8}$$

Approximation of likelihoods in Equation 3.8: We use non-parametric kernel density estimates (KDE) to approximate the likelihoods $p(x'|y' = A)$ and $p(x'|y' = B)$

This is an unnumbered heading/incomplete sentence, which you cannot have, so you must reformat. Here are your options (the first 2 options should be fairly easy to do):
~format according to your chosen bullet or numbered list style (do *not* use bold or underlining),
~write out in a full sentence/paragraph (do *not* use bold or underlining) and remember to format consistently as far as indenting, line-spacing and alignment, or
~number and format appropriately as heading, list in TOC .

Reformat any like this.

(Note: this does not apply to standard math terms (e.g., Lemma, Proof, Theorem, Corollary, and Definition) These are okay.)

sliding through the data. Among different commonly used kernels for KDE, we choose Gaussian Kernel as it is a powerful continuous kernel that would also eases the derivative computations for finding optima.

Approximation of priors in Equation 3.8: Additionally, we assume prior probabilities of observing samples in class A ($p(A)$) and class B ($p(B)$) (in Equation 3.8) are constant. Hence, these probabilities do not affect our algorithm in terms of generating synthetic neighbors for minority samples because we only determine the relative direction between the minority and the majority class. Thus, they can be canceled out at the end of the equation reduction.

Equation 3.8 reduction: Let X_A and X_B be the subsets of dataset X which contain samples of class A and class B, $X_A = \{x : y = A\}$ and $X_B = \{x : y = B\}$. N_A and N_B are the numbers of samples in X_A and X_B. d is the number of data dimensions. h presents the width parameter of the Gaussian kernel. The posterior ratio for each synthetic sample x' then can be estimated as follows:

$$f = \frac{p(x'|y' = B)\, p(B)}{p(x'|y' = A)\, p(A)} \tag{3.9}$$

$$\propto \frac{\frac{1}{N_B h^d} \sum_{i=1}^{N_B} (2\pi)^{-\frac{d}{2}}\, e^{\frac{1}{2}(\frac{x' - x_{B_i}}{h})^2}\, p(B)}{\frac{1}{N_A h^d} \sum_{j=1}^{N_A} (2\pi)^{-\frac{d}{2}}\, e^{\frac{1}{2}(\frac{x - x_{A_j}}{h})^2}\, p(A)} \tag{3.10}$$

$$\propto \frac{N_A}{N_B} \frac{\sum_{i=1}^{N_B} e^{\frac{1}{2}(\frac{x' - x_{B_i}}{h})^2}\, p(B)}{\sum_{j=1}^{N_A} e^{\frac{1}{2}(\frac{x' - x_{A_j}}{h})^2}\, p(A)} \tag{3.11}$$

$$\propto \frac{\sum_{i=1}^{N_B} e^{\frac{1}{2}(\frac{x' - x_{B_i}}{h})^2}}{\sum_{j=1}^{N_A} e^{\frac{1}{2}(\frac{x' - x_{A_j}}{h})^2}}. \tag{3.12}$$

Finding synthetic samples surrounding a minority sample: Because we want to generate neighbors for each minority sample that maximizes Function f in Equation 3.12, we examine points lying on the sphere centered at the minority sample with a small radius

r. As a result, we can find a vector $\vec{v}$ so that it can be added to the sample to generate a new sample. The relationship between a synthetic sample x' and a minority sample can be described as follows,

$$\vec{x'} = \vec{x} + \vec{v}, \tag{3.13}$$

where $|\vec{v}| = r$, and r is sampled from a uniform distribution $r \sim U(0, R)$, $0 < r < R$. The range parameter R is relatively small and computed as the average distance of k-nearest neighbors of the minority sample x to itself. This will ensure that the generated sample will be surrounding the minority sample. Consider a minority sample x and its k-nearest neighbors in the Euclidean space, R can be computed as follows:

$$R = \frac{1}{k} \sum_{1}^{k} ||x - x_j||, \tag{3.14}$$

where $||x - x_j||$ is the Euclidean distance between a minority sample x and its jth neighbor. k is a parameter indicating number of the neighbors. In practice, we prefer to tune k from 5 to 20.

Figure 3.2 depicts a demonstration of finding 3 synthetic samples from 3 minority samples. In fact, one minority can be re-sampled to generate more than one synthetic sample. For a minority sample x_0, we find a synthetic sample x_0' by maximizing the objective function $f(x_0'), x_0' \in X$ with a constraint that the Euclidean length of $\vec{v_0}$ equals to a radius r_0, $||\vec{v_0}|| = r_0$ or $||\vec{x_0'} - \vec{x_0}|| = r_0$ (derived from Equation 3.13).

The problem can be described as a constrained optimization problem. For each minority sample x, we find a synthetic sample $x' \in \mathbb{R}^d$ lying on the d-sphere centered at x with radius r and maximizing function in Equation 3.12,

$$\max_{x'} f(x') \quad \text{s.t. } ||\vec{x'} - \vec{x}|| = r, \tag{3.15}$$

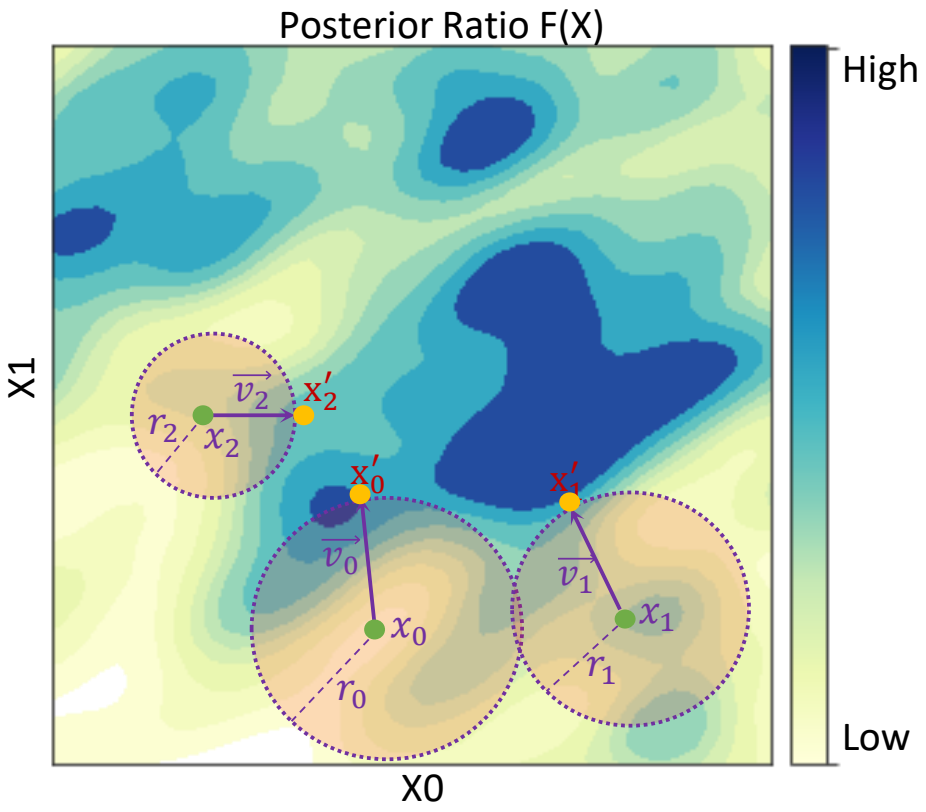


Figure 3.2: Demonstration on how SIMPOR generates three synthetic samples x'_0, x'_1, x'_2, from three minority samples x_0, x_1, x_2, by maximizing the Posterior Ratio.

where $r \sim U(0, R)$.

Solving optimization problem in Equation 3.15: Interestingly, the problem in Equation3.15 is solvable. Function $f(x)$ in Equation 3.12 is defined and continuous for $x' \in (-\infty, +\infty)$ because all of the exponential components (Gaussian kernels) are continuous and greater than zero. In addition, the constraint, $||\vec{x'} - \vec{x}|| = r$, which contains all points on the sphere centered at x with radius r is a closed set ([2]). Thus, a maximum exits as proved in [1]. To enhance the diversity of synthetic data, we accept either the global maximum or any local maximum, so that the synthetic samples will not simply go the same direction.

We solve the problem in Equation 3.15 by using the Projected Gradient Ascent approach in which we iteratively update the parameter to go up the gradient of the objective function. A local maximum is found if the objective value cannot be increased by any local update. For simplification, we rewrite the problem in Equation 3.15 by shifting the origin to the considered minority sample. The problem becomes finding the maximum of function $f(x')$, $x' \in \mathbb{R}^d$, constrained on a d-sphere, i.e., $||x'|| = r$. Our solution can be described in Algorithm 3.1. After shifting the coordinates system, we start with sampling a random point on the constraint sphere (line $1-2$). The gradient of objective function at time t, $g_t(x'_t)$, is computed and projected onto sphere tangent plane as p_t (line $4-5$). It is then normalized and used for update a new x'_{t+1} by rotating a small angle $lr * \theta$ (line $6-7$). The algorithm stops when the value of $f(x')$ is not increased by any update of x'. We finally shift to the original coordinates and return the latest x'_t.

Algorithm 3.1 Sphere-Constrained Gradient Ascent for Finding Maximum

Input: A minority sample x_0, objective function $f(x, X)$
Parameter:
r : The radius of the sphere centered at x_0
θ : Sample space $\theta \in [0, 2\pi]$
lr : Gradient ascent learning rate
Output: An local maximum x'

1: Shift the Origin to x_0
2: Randomly initiate x'_t on the sphere with radius r
3: **while** converge condition **do**
4: Compute the gradient at x'_t
$g_t(x'_t) = \nabla f(x'_t)$
5: Project the gradient onto the sphere tangent plane
$p_t = g_t - (g_t \cdot x'_t)x_t$
6: Normalize projected vector
$p_t = p_t / ||p_t||$
7: Update x' on the constrained sphere
$x'_{t+1} = x'_t cos(lr * \theta) + p_t sin(lr * \theta)$
8: **end while**
9: Shift back to the Origin
10: **return** x'_t

3.5.3 Algorithm

Our strategy can be described in Algorithm 3.2. Our algorithm takes an imbalanced dataset as its input and results in a balanced dataset which is a combination of the original dataset and synthetic samples. We first choose an active learning method $AL(\cdot)$ and find a subset of informative samples S, which is shown in lines $1-2$ in Algorithm 3.2. In this work, we choose entropy-based active learning for our experiments. We then generate synthetic data to balance S. For each random sample x_i^c in S and belonging to minority class c, we randomly sample a small radius r and find a synthetic sample that lies on the sphere centered at x_i^c and maximizes the posterior ratio in Equation 3.12 (lines $3-11$). The process is repeated until the informative set S is balanced. Similarly, the remaining region is balanced which can be described in the pseudo-code from line 12 to line 20. The final output of the algorithms is a balanced dataset D'.

3.6 Algorithm Implementation and Complexity

Our proposed method is straightforward in implementation. We first train a neural network model with initial samples and start querying next batches data based on the entropy scores from previous model to find informative samples. The model is then updated with new batches of data until the entropy scores reach a certain threshold. All the informative samples are then balanced first, and the remaining data are balanced later. Each synthetic data point is generated by finding a local maxima in Equation 3.12.

Perhaps the costly part of SIMPOR is that each synthetic sample requires to compute a kernel density estimation of the entire dataset. Elaborately, let n be the number of samples of the dataset. In the worst case, the number of samples of minority and majority class are $N_B = 1$ and $N_A = n - 2$ respectively. We need to generate $n - 1$ synthetic samples to completely balance the dataset. Since each generated sample must loop through the entire dataset of size n to estimate the density, the complexity is $O(n^2)$.

Algorithm 3.2 SIMPOR

Input: Original Imbalance Dataset D including data X and labels y.
Parameter: MA is the majority class, MI is a set of other classes.
k: Number of neighbors of the considered sample which determines the maximum range of the sample to its synthetic samples.
$Count(c, P)$: A function to count class c sample number in population P.
$G(x_0, f, r)$: Algorithm 3.1, which returns a synthetic sample on sphere centered at x_0 with radius r and maximize Equation 3.12.
Output: Balanced Dataset D' including $\{X', y'\}$

Select an Active Learning Algorithm $AL()$
Query a subset of informative samples $S \in D$ using AL: $s \leftarrow AL(D)$
{Balance the informative region}
for $c \in MI$ **do**
  while $Count(c, S) \leq Count(MA, S)$ **do**
    Select a random $x_i^c \in S$
    Compute maximum range R based on k
    Randomly sample a radius $r \sim U(0, R)$
    Generate a synthetic neighbor x' from x_i^c: $x' = G(x_i^c, f, r)$
    Append x' to D'
  end while
end for
{Balance the remaining region}
for c in MI **do**
  while $Count(c, D') \leq Count(MA, D')$ **do**
    Select a random $x_j^c \in \{X - S\}$
    Compute maximum range R based on k
    Randomly sample a radius $r \sim U(0, R)$
    Generate a synthetic neighbor x' of x_j^c
    Append x' to D'
  end while
end for
return

Although generating synthetic data is only a one-time process, and this does not affect the classification performance in the testing phase, we still alleviate its weakness by providing parallelized implementations. We provide two suggestions, multiple CPU thread-based and GPU-based implementations. While the former simply computes each synthetic data sample in a separated CPU thread, the later computes each exponential component in 3.12 parallelly in GPU's threads. More specifically, Equation 3.12 can be rewritten as N_B components of

$e^{\frac{1}{2}(\frac{x-X_{B_i}}{h})^2}$ and N_A components of $e^{\frac{1}{2}(\frac{x-X_{A_i}}{h})^2}$. Fortunately, they are all independent and can be parallelly processed in GPUs. The latter is then implemented using Python Numba and Cupy libraries which utilize CUDA toolkit from NVIDIA [65]. The consumption time for kernel density estimation for each synthetic data point is then reduced by $N_A + N_B = n$ times, which significantly simplifies the complexity to $O(n)$. Our source code can be found on following Github link https://github.com/nsh135/SIMPOR.

3.7 Experiments and Discussion

In this section, we experiment on binary classification for both artificial dataset (i.e., Moon) for demonstration and real-world datasets (i.e., Breast Cancer, Credit Card Fraud). Samples in artificial Moon have two dimensions while samples in Breast Cancer and Credit Card Fraud are both 30-dimension numerical data. Compare to the original Breast Cancer dataset size (569 samples), the other original Credit Card dataset contains a much larger amount of data, 284,907 samples. The implementation steps to balance datasets are following Algorithm 3.2. To evaluate our proposed balancing technique, we compare the classification performance to different widely-used techniques. More specifically, We compare SIMPOR to SMOTE [11], Borderline-SMOTE [35], ADASYN [37], Random Oversampling, and Raw data which does not apply any balancing technique. To evaluate classifications performance for skewed datasets, we measure some powerful and widely-used metrics such as Recall, Precision, F1-score, Area Under The Curve (AUC).

3.7.1 Sharing Settings

This subsection describes settings sharing for all datasets. In order to find the informative subset, we leverage entropy-based active learning as mentioned in Section 3.2.3. Classifier for active learning is a fully connected neural network model containing 3 hidden layers with *relu* activation functions and 100 neurons each layer. The output layer uses softmax activation

Table 3.1: Classification models' setting for each dataset.

Model Setting	Moon	Breast Cancer	Credit Card
Hidden Layers	3	3	3
Neurons/Layer	100	50	200
Hidden Activation	ReLu	ReLu	ReLu
Output Activation	Softmax	Softmax	Softmax
Epochs	200	150	200
Batch size	32	32	64
Optimizer	Adam	Adam	Adam
Learning Rate	0.01	0.01	0.01

function. The models are trained in a maximum of 300 epochs with the early stop option when the loss does not change after updating weights.

In addition, we randomly split the data into two parts, 80% for training and 20% for testing. Reported testing results for each dataset are the averages of 5 experimental trials. For SIMPOR to find optima of the function in Equation 3.15, we use a gradient ascent rate of 0.00001 and the maximum iteration of 300. The architecture detail of the evaluation model for each dataset is described in Table 3.1.

3.7.2 SIMPOR on artificial Moon dataset

We implement our technique on an artificial 2-dimension numeric dataset as a demonstration of our proposed method. Figure 3.3 captures the classification F1 results for different techniques. We also visualize model decision boundaries to provide additional information on how the classification models are affected. To classify the data, we use a fully connected neural network which is described in Table 3.1.

3.7.2.1 Dataset

We first generate a balanced dataset using python library *sklearn.datasets.make_moons* including 3000 two-dimensional samples labeled in two classes, A and B. We then create

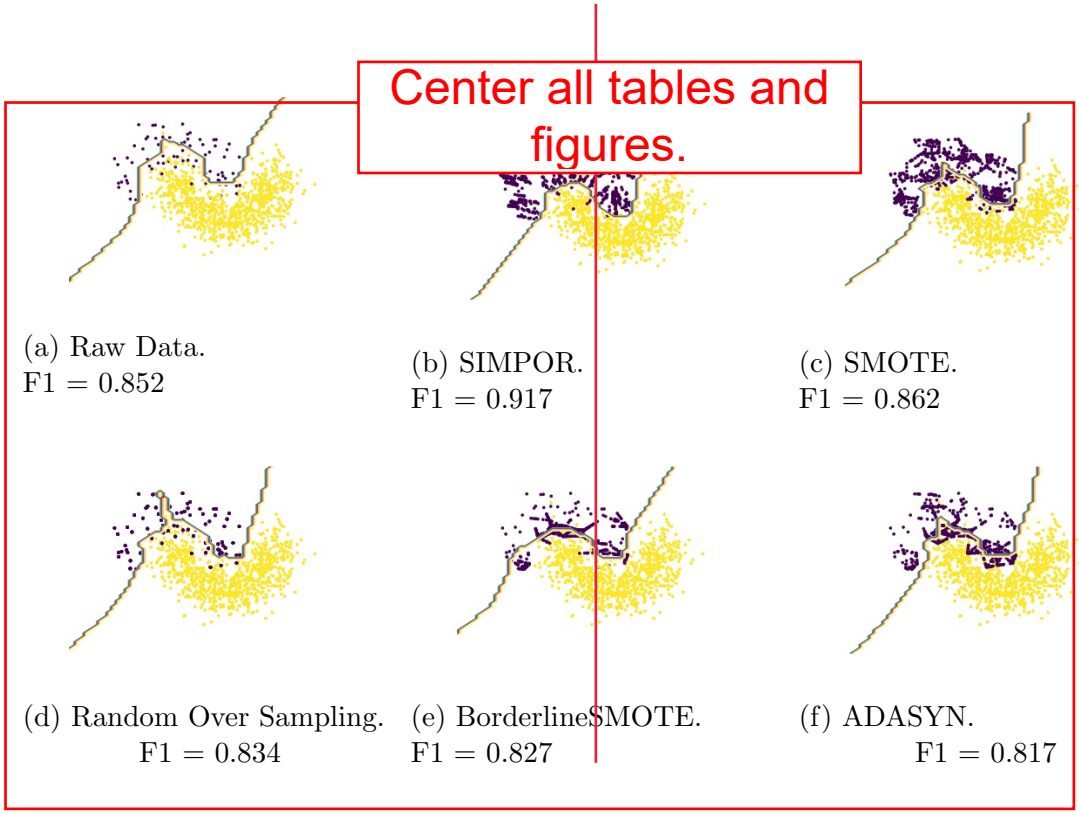


Figure 3.3: Balanced training data plot, model's decision boundary plot and testing F1 score for Moon Dataset.

an imbalanced dataset with an Imbalance Ratio of 3:1 by randomly removing 1000 samples from class B. As a result, the training dataset becomes imbalanced as visualized in Figure 3.3a, which contains 400 samples of class B and 1200 samples of class A. The testing data contains 100 samples of class B and 300 samples of class A.

3.7.2.2 Results and Discussion

From the results shown in Figure 3.3, it is clear that SIMPOR performs better than others at the F1 score of 0.9170 (Figure 3.3b). Without any balancing techniques, it is not surprising that the classification result on the raw imbalanced data achieves the lower F1 Score, at 0.852 (Figure 3.3a). SMOTE technique does help to improve the classification F1-score to 0.862; however, it has not reached SIMPOR's performance. Other techniques perform poorly in this case; they could not achieve an F1-score as high as the raw data can achieve.

Additionally, SIMPOR, from our observation, results in a smooth and robust model decision boundary. We can see that the Random Over Sampling which randomly duplicates minority samples might cause overfitting where samples are duplicated many times and significantly increase their weights. SMOTE does better than Random Over Sampling. However, due to the fact that SMOTE does not take the informative region into account, unbalanced data in this area lead to a severe error in decision boundary. In Figures 3.3f and 3.3e, BorderlineSMOTE and ADASYN focus on the area near the model's decision boundary, but they inherit a drawback from SMOTE; any noise or mislabeled samples can create very dense bridges crossing the expected border so that it leads to decision errors. In contrast, by generating neighbors of samples in the direction towards the minority class and balancing the informative region, SIMPOR (Figure 3.3b) helps the classifier to make a better decision with a solid smooth decision line. It is also worth noticing that the classifier decision boundary lines of all other techniques are rougher than that of SIMPOR. This is because they randomly generate synthetic samples, and it might cause data imbalance in the informative region.

3.7.3 SIMPOR on Breast Cancer dataset

3.7.3.1 Dataset

Breast Cancer is a real-world dataset containing information on cancer patients collected by Dua and Graff [18]. This has two classes including 357 negatives and 212 positives; each record includes 30 features extracted from a digitized image of a fine needle aspirate of a breast mass e.g., radius, area, and perimeter. Some of the records in the training dataset are randomly removed to create different imbalance ratios.

3.7.3.2 Results and Discussions

Table 3.2 shows the classification results of different data balancing techniques on the Breast Cancer dataset over the imbalance ratios of 3:1 and 7:1. Overall, SIMPOR outper-

Table 3.2: Classification results of different data balancing techniques on Breast Cancer Dataset.

Breast Cancer							
Majority:Minority (IR)	Metric	**SIMPOR**	**SMOTE**	**Borderline SMOTE**	**Over-Sampling**	**ADASYN**	**RawData**
357:120 (3:1)	F1	**0.953**	0.931	0.891	0.935	0.835	0.944
	AUC	**0.974**	**0.974**	0.964	**0.974**	**0.974**	0.971
	Precision	0.970	0.924	0.868	0.935	0.790	**0.975**
	Recall	**0.938**	**0.938**	0.922	0.936	0.928	0.919
357:50 (7:1)	F1	**0.943**	0.903	0.865	0.939	0.821	0.879
	AUC	**0.968**	**0.968**	0.967	0.967	**0.968**	0.966
	Precision	**0.952**	0.899	0.846	0.948	0.805	0.873
	Recall	**0.936**	0.918	0.903	0.932	0.879	0.907

forms other techniques on both imbalance ratios at an F1-score of 0.947 and 0.935 for IR of 3:1 and 7:1, respectively. SIMPOR also achieves the highest AUC results at 0.977 and 0.986 for both imbalance ratios. Both tests over the raw data (without any balancing technique) received the lowest F1-score as expected.

3.7.4 SIMPOR on Credit Card Fraud

3.7.4.1 Dataset

In this section, we experiment with our technique on Credit Card Fraud dataset [33]. The original dataset contains 492 fraud records out of 284,807 transactions; each record includes 30 features. During the experiments, we fix the size of fraud records as it becomes the minority class and randomly select a part of the remaining normal transactions to generate new datasets with different imbalance ratios. The classification model for the experiments under this dataset can be found in Table 3.1.

3.7.4.2 Results and Discussions

Table 3.3 shows the classification results on Credit Card Fraud dataset over different balancing techniques. SIMPOR constantly overperforms other techniques over all the settings and achieves the highest F1 and AUC score. While SIMPOR, SMOTE improve classifica-

Table 3.3: Classification results of different data balancing techniques on Credit Card Fraud Dataset.

Credit Card Fraud							
Majority:Minority (IR)	Metric	**SIMPOR**	**SMOTE**	**Borderline SMOTE**	**Over-Sampling**	**ADASYN**	**RawData**
1470:490 (3:1)	F1	**0.943**	0.903	0.865	0.939	0.821	0.879
	AUC	**0.968**	**0.968**	0.967	0.967	**0.968**	0.966
	Precision	**0.952**	0.899	0.846	0.948	0.805	0.873
	Recall	**0.936**	0.918	0.903	0.932	0.879	0.907
3430:490 (7:1)	F1	**0.953**	0.931	0.891	0.935	0.835	0.944
	AUC	**0.974**	**0.974**	0.964	**0.974**	**0.974**	0.971
	Precision	**0.975**	0.924	0.868	0.935	0.790	**0.975**
	Recall	**0.938**	**0.938**	0.922	0.936	0.928	0.919
4900:490 (10:1)	F1	**0.952**	0.901	0.863	0.658	0.906	0.930
	AUC	**0.972**	0.957	0.967	0.964	0.969	0.975
	Precision	**0.979**	0.891	0.855	0.662	0.885	0.931
	Recall	0.929	0.925	0.918	0.845	0.933	**0.936**
7350:490 (15:1)	F1	**0.952**	0.909	0.885	0.917	0.883	0.944
	AUC	**0.968**	0.963	0.955	0.963	0.965	0.960
	Precision	0.966	0.892	0.850	0.909	0.849	**0.984**
	Recall	**0.940**	0.931	0.931	0.935	0.931	0.911

tion performance in most of the settings, ADASYN, BorderlineSMOTE fail to create good synthetic samples and reduce the classification performance.

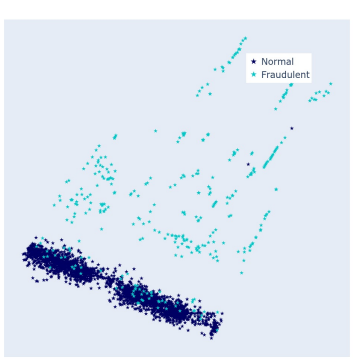
To better understand how the techniques perform, we visualize the generated data by projecting them onto lower dimensions (i.e., one and two dimensions) space using the Principle Component Analysis technique (PCA) [25]. Data's 2-Dimension (2D) plots and 1-Dimension histograms are presented in Figure 3.4. A hard-to-differentiate ratio (HDR) is defined as the ratio of intersection between 2 classes in the 1D histogram to the total Fraudulent samples ($HDR = \frac{No.\ Intersection\ samples}{No.\ Fraud\ samples}$). This ratio is expected to be as small as 0% if the two classes are well separated; in contrast, 100% indicates that the two classes are unable to be distinguished. Other than HDR, the bottom tables in Figure 3.4 also show the absolute numbers of Fraudulent, Normal, and Intersection samples for each technique. From the plots, we can observe how the data distribute in 2D space and quantify hard-to-differentiate samples in the histograms.

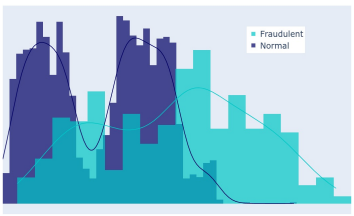
While some techniques help to reduce the hard-to-differentiate ratios, others increase this ratio and worsen the data distribution. For example, in the 1D histogram of the ADASYN

technique (Figure 3.4d), there are 2115 samples in the histogram intersection between 2 classes; they account for 81.82% of the total 2585 Fraudulent samples, which is worse than HDR of the Rawdata case (69.75%). In addition, the 2-Dimension plot illustrates that many synthetic samples (Fraudulent class) cross the majority (Normal class). Similarly, Figure 3.4e shows that BorderlineSMOTE also poorly generates synthetic minority data in this setting (CreditCard Fraudulent dataset with IR of 7:1) with an HDR of 75.98%. Among all techniques, SIMPOR achieves the best HDR of 11.14% and the synthetic data are far away from the majority class, which helps to improve the classification results as shown in Table 3.3.

3.8 Conclusion

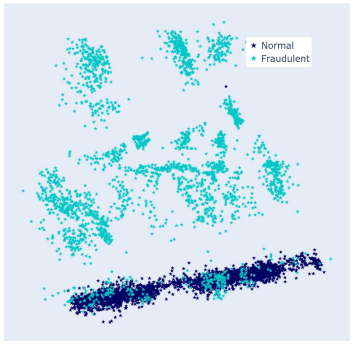
We propose a data balancing technique by generating synthetic data for minority samples, which maximizes the posterior ratio to embrace the chance they fall into the minority class and do not fall across the expected decision boundary. While maximizing the posterior ratio, we use kernel density estimation to estimate the likelihood so that it is able to work with complex distribution data without requiring data distribution assumptions. In addition, our technique leverage entropy-based active learning to find and balance the most informative samples. This is important to improve model performance as we have shown in our experiments. In future work, we would like to investigate imbalanced image datasets and enhance our technique to adapt to image data.

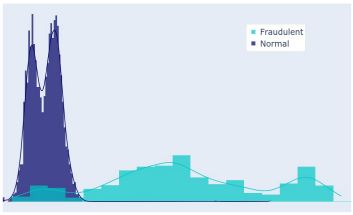




Fraud	Normal	Inter.	HDR
367	2585	256	69.75%

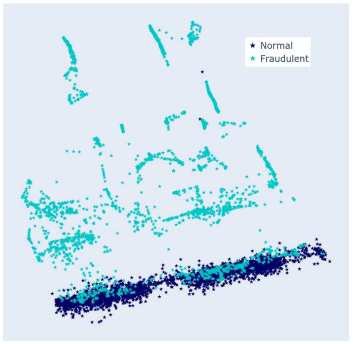
(a) Raw Data.

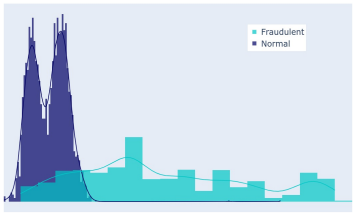




Fraud	Normal	Inter.	HDR
2585	2585	288	**11.14%**

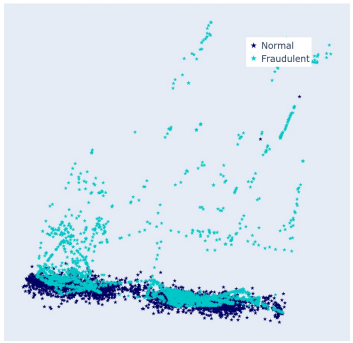
(b) Generated data by SIMPOR.

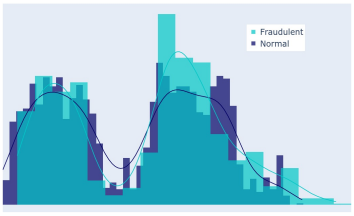




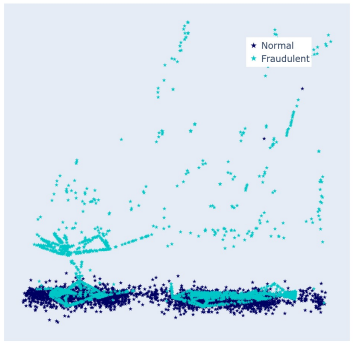
Fraud	Normal	Inter.	HDR
2585	2585	534	20.66%

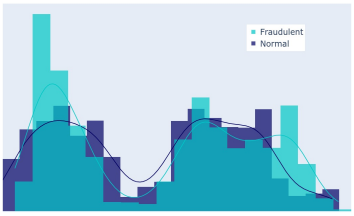
(c) Generated data by SMOTE.



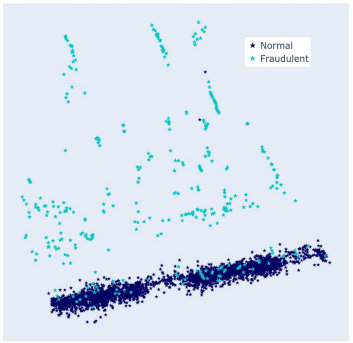


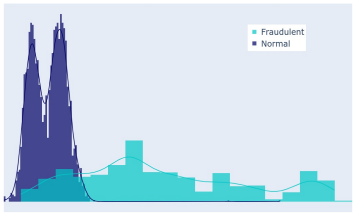
Fraud	Normal	Inter.	HDR
2585	2585	2115	81.82%





Fraud	Normal	Inter.	HDR
2585	2585	1964	75.98%





Fraud	Normal	Inter.	HDR
2585	2585	544	21.4%

enerated data by
sampling.

pace and their histograms
sion reduction technique.

The explanation tables illustrate the number of samples in each class (Fraudulent and Normal), 1-Dimension histogram intersection between 2 classes, and the hard-to-differentiate ratio ($HDR = \frac{Inter.}{Fraud}100\%$).

Do not let any text or other items run into margins (page numbers are okay to be within the bottom margin)

Chapter 4: AutoGAN-Based Dimension Reduction for Privacy Preservation

4.1 Introduction

Machine Learning (ML) is an important aspect of modern applications that rely on big data analytics (e.g., an on-line system collecting data from multiple data owners). However, these applications are progressively raising many different privacy issues as they collect different types of data on a daily basis. For example, many types of data are being collected in smart cities such as patient records, salary information, biological characteristics, Internet access history, personal images and so on. These types of data then can be widely used in daily recommendation systems, business data analysis, or disease prediction systems which in turn affect the privacy of individuals who contributed their sensitive data. Considering a multi-level access control system of a company using biometric recognition (e.g., face recognition, fingerprint) for granting permission to access data resources, the company staff members may concern their biological information being vulnerable to adversaries. Even though the utility of these biometric features can be effectively used in machine learning tasks for authentication purpose, leaking this information might lead to privacy breaches. For example, an adversary could utilize them to determine the members' identities.

Several tools and methods have been developed to preserve the privacy in machine learning applications, such as homomorphic encryption [8, 21, 38], secure multi-party computing [90, 76], differential privacy (DP) [9, 93, 68, 3, 85], compressive privacy [98, 13, 12, 96, 48, 49, 86] and so on. Typically, differential privacy-based methods aim at preventing leaking individual information caused by queries. However, they are not designed to serve large number of queries since they require adding huge amount of noise to preserve privacy, thus significantly decreasing the ability to learn meaningful information from data. On the other

hand, homomorphic encryption-based methods can be used to privately evaluate a function over encrypted data by a third party without accessing to plain-text data, hence the privacy of data owners can be protected. However, due to the high computational cost and time consumption, they may not work with a very large dataset, normally required in ML applications.

In this study, we consider an access control system collecting dimension-reduced face images of staff members to perform authentication task and to provide permission for members who would like to access company's data resources (Figure 4.1). We propose a non-linear dimension reduction framework to decrease data dimension for the authentication purpose mentioned above and to protect against an adversary from reconstructing member images. Firstly, we introduce ϵ-DR Privacy as a theoretical tool for dimension reduction privacy evaluation. It evaluates the reconstruction distance between original data and reconstructed data of a dimension reduction (DR) mechanism. This approach encourages a DR mechanism to enlarge the distance as high distance yields high level of privacy. While other methods such as differential privacy-based methods rely on inference uncertainty to protect sensitive data, ϵ-DR Privacy is built on reconstruction error to evaluate privacy. Therefore, unlike differential privacy methods, ϵ-DR Privacy is not negatively impacted by the number of queries. Secondly, as detailed in Section 4.3, we recommend a privacy-preserving framework Autoencoder Generative Adversarial Nets-based Dimension Reduction Privacy (AutoGAN-DRP) for enhancing data owner privacy and preserving data utility. The *utility* herein is evaluated via machine learning task performance (e.g., classification accuracy).

Our dimension reduction (DR) framework can be applied to different types of data and used in several practical applications without heavy computation of encryption and impact of query number. The proposed framework can be applied directly to the access control system mentioned above. More elaboratively, face images are locally collected, nonlinearly compressed to achieve DR, and sent to the authentication center. The server then performs classification tasks on the dimension-reduced data. We assume the authentication server is

semi-honest, that is to say it does not deviate from authenticating protocols while being curious about a specific member's identity. Our DR framework is designed to resist against reconstruction attacks from a strong adversary who obtains the training dataset and the transformation model.

During the stage of experiments, we implemented our framework to evaluate dimension-reduced data in terms of accuracy of the classification tasks, and we attempted to reconstruct original images to examine the capacity of adversaries. We performed several experiments on three facial image datasets in both gray-scale and color, i.e., *the Extended Yale Face Database B* [28], *AT&T* [74], and *CelebFaces Attributes Dataset (CelebA)* [58]. The experiment results illustrate that with only seven reduced dimensions our method can achieve accuracies of 93%, 90%, and 80% for AT&T, YaleB, and CelebA respectively. Further, our experiments show that at the accuracies of 79%, 80% and 73% respectively, the reconstructed images could not be recognized by human eyes. In addition, the comparisons shown in Section 4.6 also illustrate that AutoGAN-DRP is more resilient to reconstruction attacks compared to related works. Our work has two main contributions:

1. To analytically support privacy guarantee, we introduce ϵ-DR Privacy as a theoretical approach to evaluate privacy preserving mechanism.

2. We propose a non-linear dimension reduction framework for privacy preservation motivated by Generative Adversarial Nets [31] and Auto-encoder Nets [6].

The rest of this chapter is organized as follows. Section 4.2 summarizes state-of-the-art privacy preservation machine learning (PPML) techniques and reviews knowledge of deep learning methods including generative adversarial neural nets and Auto-encoder. Section 4.3 describes the privacy problem through a scenario of a facial recognition access control system, introduces the definition of ϵ-DR Privacy to evaluate DR-based privacy preserving mechanisms, and presents our framework AutoGAN-based Dimension Reduction for Privacy Preservation. Section 4.4 presents and discusses our experiment results over three different

face image datasets. Section 4.5 compares AutoGAN-DRP to a similar work GAP in terms of reconstruction error and classification accuracy. Section 4.6 demonstrates reconstructed images over AutoGAN-DRP and other privacy preservation techniques (i.e., Differential Privacy and Principle Component Analysis). Finally, the conclusion and future work are mentioned in Section 4.7.

4.2 Related Work

4.2.1 Literature Review

Cryptographic approach: This approach usually applies to the scenarios where the data owners do not wish to expose their plain-text sensitive data while asking for machine learning services from a third-party. The most common tool used in this approach is fully homomorphic encryption that supports multiplication and addition operations over encrypted data, which enabling the ability to perform a more complex function. However, the high cost of the multiplicative homomorphic operations renders it difficult to be applied on machine learning tasks. In order to avoid multiplicative homomorphic operations, additive homomorphic encryption schemes are more widely used in privacy preserving machine learning (PPML). However, the limitation of the computational capacity in additive homomorphic schemes narrows the ability to apply on particular ML techniques. Thus, such additive homomorphic encryption-based methods in [8, 21, 7, 39] are only applicable to simple machine learning algorithms such as decision tree and naive bayes. In Hesamifard's work [38],the fully homomorphic encryption is applied to perform deep neural networks over encrypted data, where the non-linear activation functions are approximated by polynomials.

In secure multi-party computing (SMC), multiple parties collaborate to compute functions without revealing plain-text to other parties. A widely-used tool in SMC is garbled circuit [90], a cryptographic protocol carefully designed for two-party computation, in which they can jointly evaluate a function over their sensitive data without the trust of each other. In [5], Mohammad introduced a SMC protocol for principle component analysis (PCA)

which is a hybrid system utilizing additive homomorphic and garbled circuit. In secret sharing techniques [76], a secret **s** is distributed over multiple pieces **n** also called *shares*, where the secret can only be recovered by a sufficient amount of **t** *shares*. A good review of secret sharing-based techniques and encryption-based techniques for PPML is given in [67]. Although these encryption-based techniques can protect the privacy in particular scenarios, their computational cost is a significant concern. Furthermore, as [67] elaborated, the high communication cost also poses a big concern for both techniques.

Non-Cryptographic approach: Differential Privacy (DP) [20] aims to prevent membership inference attacks. DP considers a scenario that an adversary infers a member's information based on the difference of outputs of a ML mechanism before and after the member join a database. The database with the member's information and without the member's information can be considered as two neighbor databases which differ by at most one element. DP adds noise to the outputs of the ML mechanism to result in similar outputs from the two neighbor databases. Thus, adversaries cannot differentiate the difference between the two databases. A mechanism M satisfies ϵ-differential privacy if for any two neighbor databases D and D', and any subset S of the output space of M satisfies $Pr[M(D) \in S] \leq e^{\epsilon} Pr[M(D') \in S]$. The similarity of query outputs protects a member information from such membership inference attacks. The *similarity* is guaranteed by the parameter ϵ in a mechanism in which the smaller ϵ provides a better level of privacy preservation. [9, 93, 10, 84, 89] propose methods to guarantee ϵ-differential privacy by adding noise to outcome of the weights $w^* = w + \eta$, where η drawn from Laplacian distribution and adding noise to the objective function of logistic regression or linear regression models. [68, 3] satisfy differential privacy by adding noise to the objective function while training a deep neural network using stochastic gradient descent as the optimization algorithm.

In addition, there are existing works proposing differential privacy dimension reduction. One can guarantee ϵ-differential privacy by perturbing dimension reduction outcome. Principal component analysis (PCA) whose output is a set of eigenvectors is a popular method

in dimension reduction. The original data is then represented by its projection on those eigenvectors, which keeps the largest variance of the data. One can reduce the data dimension by eliminating insignificant eigenvectors which contain less variance, and apply noise on the outcome to achieve differential privacy[85]. However, the downside of these methods is that they are designed for specific mechanisms and datasets and not working well with the others. For example, record-level differential privacy is not effectively used with image dataset as shown in [40]. Also, the amount of added noise is accumulative based on the number of queries so that this approach usually leads to low accuracy results with a high number of queries.

Similar to our work, Generative Adversarial Privacy (GAP) [13] is a perturbation method utilizing the minimax algorithm of Generative Adversarial Nets to preserve privacy and to keep utility of image datasets. GAP perturbs data within a specific l_2 distance constraint between original and perturbed data to distort private class labels and at the same time preserve non-private class labels. However, it does not protect the images themselves, and an adversary can visually infer private label (e.g., identity) from images. In contrast, our method protects an image by compressing it into a few dimension vector and then transferring without clearly exposing the original image.

4.2.2 Preliminaries

To enhance the distance between original and reconstructed data in our DR system, we utilize the structure of Generative Adversarial Network (GAN) [31] for data perturbation and deep Auto-encoder [6] for data reconstruction. The following sections briefly review Auto-encoder and GAN.

4.2.2.1 Auto-encoder

Auto-encoder is aimed at learning lower dimension representations of unsupervised data. Auto-encoder can be used for denoising and reducing data dimension. It can be implemented

by two neural network components: *encoder* and *decoder*. The *encoder* and *decoder* perform reverse operations. The input of the *encoder* is the original data while the output of the *decoder* is expected to be similar to the input data. The middle layer extracts latent representation of original data that could be used for dimension reduction. An Auto-encoder training process can be described as a minimization problem of the auto-encoder's loss function $\mathcal{L}(\cdot)$:

$$\mathcal{L}(x, g(f(x))) \tag{4.1}$$

where x is input data, f(·) is an encoding function, and g(·) is a decoding function.

4.2.2.2 GAN

Generative Adversarial Nets is aimed at approximating distribution p_d of a dataset via a generative model. GAN simultaneously trains two components *generator* G and *discriminator* D, and the input of G is sampled from a prior distribution $p_z(z)$ through which G generates fake samples similar to the real samples. At the same time, D is trained to differentiate between fake samples and real samples, and send feedback to G for improvement. GAN can be formed as a two-player minimax game with value function V(G,D):

$$\begin{aligned}\min_G \max_D V(G, D) =& E_{x\sim p_d}[log(D(x))]+ \\ & E_{z\sim p_z}[log(1 - D(G(z)))]\end{aligned} \tag{4.2}$$

The two components, *Generator* and *Discriminator* can be built from neural networks (e.g., fully connected neural network, convolutional neural network). The goal of G is to reduce the accuracy of D. Meanwhile, the goal of D is to differentiate fake samples from real samples. These two components are trained until the discriminator cannot distinguish between generated samples and real samples.

4.3 Methodology

In this section, we first describe the problem and threat model, then we introduce a definition of DR-Privacy and our dimensionality reduction method (AutoGAN-DRP).

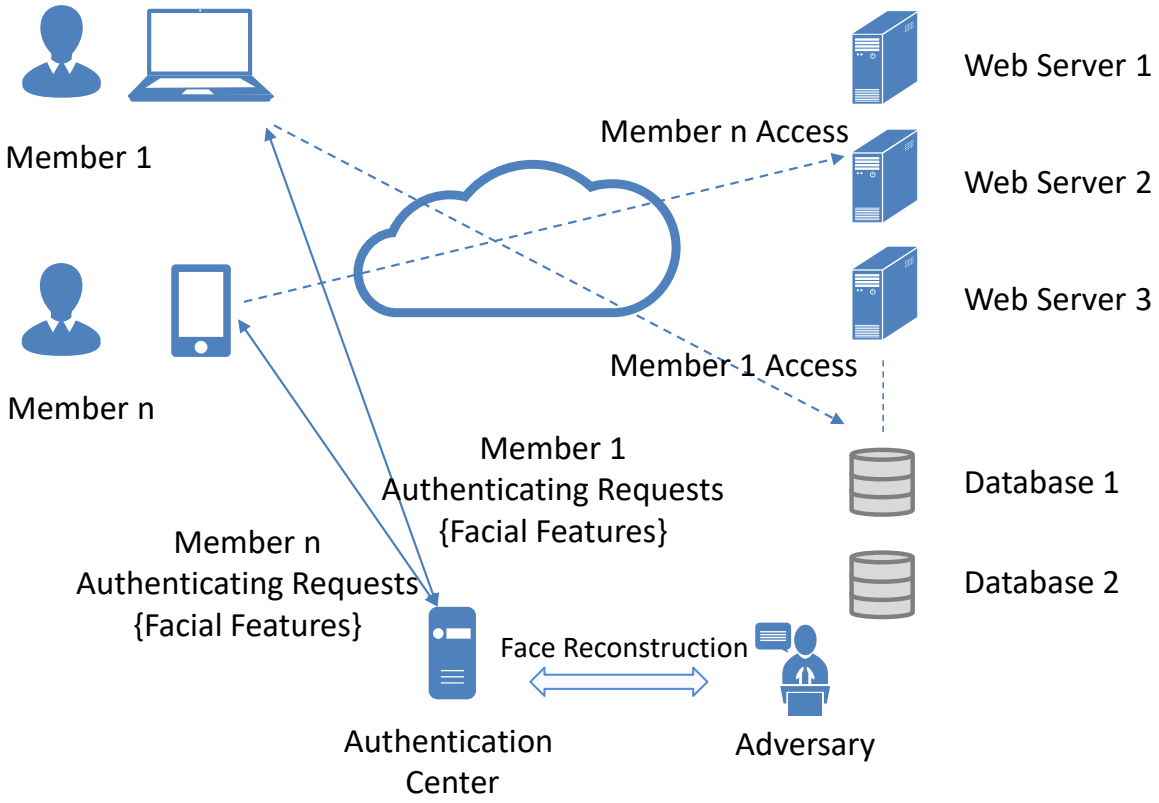


Figure 4.1: Attack Model

4.3.1 Problem statement

We introduce the problem through the practical scenario mentioned in Section 4.1. Figure 4.1 briefly describes the entire system in which staff members (clients) in a company request access to company resources, such as websites and data servers through a face recognition access control system. For example, if member n requests to access web server 2, the local device first takes a facial photo of the member by an attached camera, locally transforms

it into lower dimension data, and sends to an authentication center. The authentication server then obtains the low dimensional data and determines member access eligibility by using a classifier without clear face images of the requesting member. We consider that the system has three levels of privileges (i.e., single level, four-level, eight-level) corresponding to three groups of members. We assume the authentication server is semi-honest (it obeys work procedure but might be used to infer personal information). If the server is compromised, an adversary in the authentication center can reconstruct the face features to achieve plain-text face images and determine members' identity.

4.3.2 Threat Model

In the above scenario, we consider that a strong adversary who has access to the model and training dataset attempts to reconstruct the original face images for inferring a specific member's identity. Our attack model can be represented in Figure 4.1. The adversary utilizes training data and facial features to identify a member identity by reconstructing the original face images using a reconstructor in an auto-encoder. Rather than using fully connected neural network, we implement the auto-encoder by convolutional neural network which more effective for image datasets. Our goal is to design a data dimension reduction method for reducing data dimension and resisting full reconstruction of original data.

4.3.3 ϵ-Dimension Reduction Privacy (ϵ-DR Privacy)

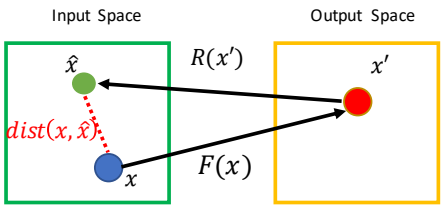


Figure 4.2: DR projection and reconstruction.

We introduce the Dimension Reduction Privacy (DR-Privacy), and define a formal definition of the ϵ-DR Privacy to mathematically quantify/evaluate the mechanisms designed to preserve the DR-Privacy via dimension reduction. The DR-Privacy aims to achieve privacy-preserving via dimension reduction, which refers to transforming the data into a lower dimensional subspace, such that the private information is concealed while the underlying probabilistic characteristics are preserved, which can be utilized for machine learning purposes. To quantify the DR-Privacy and guide us to design such DR functions, we define ϵ-DR Privacy as follows.

Definition 1: (ϵ-DR Privacy) A Dimension Reduction Function $F(\cdot)$ satisfies ϵ-DR Privacy if for each Unbold this part nput sample x drawn from the same distribution D, and for a certain distance measure $dist(\cdot)$, we have

$$\mathbb{E}[dist(x, \hat{x})] \geq \epsilon \tag{4.3}$$

where $\mathbb{E}[\cdot]$ is the expectation, $\epsilon \geq 0$, $x' = F(x)$, $\hat{x} = R(x')$, and $R(\cdot)$ is the Reconstruction Function.

For instance, as shown in Fig. 4.2, given original data x, our framework utilizes certain dimension reduction function $F(x)$ to transform the original data x into the transformed data x'. The adversaries aim to design a corresponding reconstruction function $R(x')$ such that the reconstructed data $\hat{x}$ would be closed/similar to the original data x. DR-Privacy aims to design/develop such dimension reduction functions, that the distance between the original data and its reconstructed data would be large enough to protect the privacy of the data owner.

4.3.4 AutoGAN-based Dimension Reduction for Privacy Preserving (AutoGAN-DRP)

We propose a deep learning framework for transforming face images to low dimensional data which is hard to be fully reconstructed. The framework can be presented in Figure 4.3.

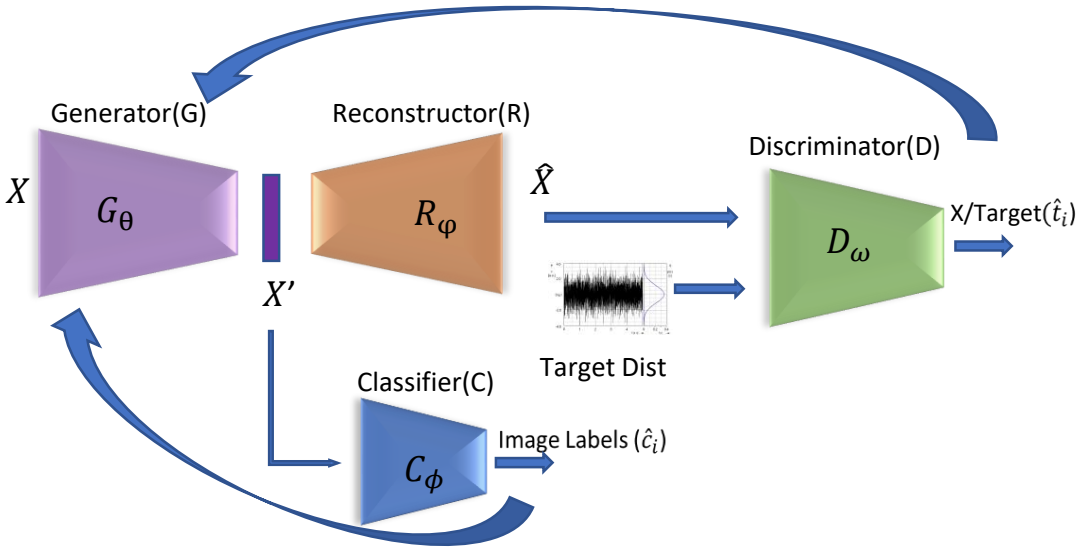


Figure 4.3: AutoGAN-DRP

We leverage the structure of an auto-encoder [6] which contains encoder and decoder (in this work, we called them generator and re-constructor) in order to reduce data dimension. More specifically, the low dimensional representations are extracted from the middle layer of the auto-encoder (the output of the generator). The dimension-reduced data can be sent to the authentication server as an authentication request. We consider an adversary as a re-constructor implemented by a decoder. To resist against fully reconstructing images, the framework utilizes a discriminator in GAN [31] to direct reconstructed data to a designated target distribution with an assumption that the target distribution is different from our data distribution. In this work, the target distribution is sampled from Gaussian distribution and the mean is the average of training data. After projecting data into a lower dimension domain, the re-constructor is only able to partially reconstruct the data. Therefore, the adversary might not be able to recognize an individual's identity. To maintain data utility, we also use feedback from a classifier. The entire framework is designed to enlarge the distance between original data and its reconstruction to preserve individual privacy and retain significant data information. The dimension-reduced transformation model is extracted from the framework and provided to clients for reducing their face image dimensions. The clas-

sification model will be used in an authentication center that classifies whether a member's request is valid to have access (1) or not (0).

We formulate the problem as follows: Let X be the public training dataset. (x_i, y_i) is the ith sample in the dataset in which each sample x_i has d features and a ground truth label y_i. The system is aimed at learning a dimension reduction transformation $F(\cdot)$ which transforms the data from d dimensions to d' dimensions in which $d' \ll d$. Let X' be the dataset in lower dimension domain. The dimension-reduced data should keep significant information to work with different types of machine learning tasks and should resist against the reconstruction or inference from data owner information.

Our proposed framework is designed to learn a DR function $F(\cdot)$ that projects data onto low dimension space and preserves privacy at certain value of ϵ. The larger distance implies higher level of privacy. Figure 4.3 presents our learning system in which the dimension-reduced data X' is given by a generator G. Since X' is expected to be accurately classified by a classifier C, the generator improves by receiving feedback from the classifier via the classifier's loss function $\mathcal{L}_C$. We use a binary classifier for single-level authentication system and multi-class classifiers for multi-level authentication system. The classifier loss function is defined as the cross entropy loss of the ground truth label y and predicted label $\hat{y}$ as follows.

$$\mathcal{L}_C = -\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij}\log(\hat{y}_{ij}) \tag{4.4}$$

where m denotes the number of classes and n denotes the number of samples.

To evaluate data reconstruction and enlarge the reconstruction distance, a re-constructor R is trained as a decoder in an auto-encoder and sends feedback to the generator via its loss function $\mathcal{L}_R$. The re-constructor plays its role as an aggressive adversary attempting to reconstruct original data by using known data. The loss function of R is the mean square

error of original training data (x) and reconstructed data ($\hat{x}$), as displayed in (4.5).

$$\mathcal{L}_R = \sum_{i=1}^{n} (x_i - \hat{x}_i)^2 \tag{4.5}$$

To direct the reconstructed data to a direction that reveals less visual information, the generator is trained with a discriminator D as a minimax game in GAN. The motivation is to direct reconstructed data to a certain target distribution (e.g., normal distribution). To ensure a distance, the target distribution should be different to training data distribution. The discriminator aims to differentiate the reconstructed data from samples of the target distribution. The loss function of D ($\mathcal{L}_D$) can be defined as a cross-entropy loss of ground truth labels (0 or 1) t and prediction labels $\hat{t}$ shown in (4.6).

$$\mathcal{L}_D = -\sum_{i=1}^{n} (t_i \log(\hat{t}_i) + (1 - t_i) \log(1 - \hat{t}_i)) \tag{4.6}$$

The optimal generator parameter θ^* is given by the optimization problem of the generator loss function $\mathcal{L}_G$:

$$\underset{\theta}{\textit{minimize}}\ \mathcal{L}_G(\theta) = \alpha \min_{\phi} \mathcal{L}_C - \beta \min_{\omega} \mathcal{L}_D - \gamma \min_{\varphi} \mathcal{L}_R + \mathcal{C}(\epsilon) \tag{4.7}$$

where θ, ϕ, ω, and φ are the model parameters of the generator, classifier, discriminator, and re-constructor respectively. α, β, and γ are weights of components in the objective function of the generator and can be freely tuned. $\mathcal{C}(\epsilon)$ is a constraint function with respect to hyper-parameter ϵ, as to be elaborated in the following subsection.

4.3.5 Optimization With Constraint

In order to meet a certain level of reconstruction distance, we consider the constrained problem:

$$\begin{aligned} &\underset{\theta}{\mathit{minimize}}\ \mathcal{L}_G(\theta) \\ &s.t \quad \mathbb{E}_{x\sim p_d}[\mathit{dist}(x,\hat{x})] \leq \epsilon \end{aligned} \tag{4.8}$$

The optimization problem above can be approximated as an unconstrained problem [45]:

$$\underset{\theta}{\mathit{minimize}}\ (\mathcal{L}_G(\theta) + \gamma\mathcal{C}(\epsilon)) \tag{4.9}$$

where γ is a penalty parameter and $\mathcal{C}$ is a penalty function

$$\mathcal{C}(\epsilon) = \mathsf{max}(0, \mathbb{E}_{x\sim p_d}[\mathit{dist}(x,\hat{x})] - \epsilon) \tag{4.10}$$

Note that $\mathcal{C}$ is nonnegative, and $\mathcal{C}(\theta) = 0$ iff the constraint in (4.8) is satisfied.

4.3.6 Training Algorithms

Algorithm 4.1 describes the training process of AutoGAN-DRP. The framework contains four components, and they are trained one by one (lines 4-15) within one global training step. After sampling batches from target distribution and data for inputs of the models (lines 2-3), we then train the four components. First, the re-constructor is trained in n_r iterations while other components' parameters are fixed (lines 4-6). Second, the discriminator is trained (lines 7-9). Third, the classifier is trained in n_c iterations (lines 10-12). Fourth, the generator is trained in n_g iterations (lines 13-15). After training each component in their number of local training steps, the above training process is repeated until it reaches the number of global training iterations (lines 1-16). In our setting, the numbers of local training iterations (n_c, n_r, n_d, n_g) are much smaller than the number of global iterations n.

4.4 Experiments and Discussion

In this section, we demonstrate our experiments over three popular supervised face image datasets: *the Extended Yale Face Database B* [28], *AT&T* [74], and *CelebFaces Attributes*

Table 4.1: Implementation information

	VGG16			VGG19			CNN		
	Hidden layers	Units	Parameter	Hidden layers	Units	Parameter	Hidden layers	Units	Parameter
Generator	Conv_block	64×2	16,295,623	Conv_block	64×2	21,605,319	Conv	256	16,451,847
	Max_pooling			Max_pooling			BatchNorm		
	Conv_block	128×2		Conv_block	128×2		Conv	512	
	Max_pooling			Max_pooling			BatchNorm		
	Conv_block	256×3		Conv_block	256×4		Conv	1024	
	Max_pooling			Max_pooling			BatchNorm		
	Conv_block	512×3		Conv_block	512×4		Dense	1024	
	Max_pooling			Max_pooling					
	Conv_block	512×3		Conv_block	512×4				
	Max_pooling			Max_pooling					
	Dense	1024		Dense	1024				
	Dense	1024		Dense	1024				
Reconstructor	Dense	1024	10,184,000	Dense	1024	13,281,472	Dense	1024	18,048,256
	Dense	1024		Dense	1024		BatchNorm		
	Dense	1024		Dense	1024		Reshape		
	Reshape			Reshape			Conv	1024	
	Conv_block-T	512×3		Conv_block-T	512×4		BatchNorm		
	Up_sampling			Up_sampling			Conv	512	
	Conv_block-T	512×3		Conv_block-T	512×4		BatchNorm		
	Up_sampling			Up_sampling			Conv	256	
	Conv_block-T	256×3		Conv_block-T	256×4				
	Up_sampling			Up_sampling					
	Conv_block-T	128×2		Conv_block-T	128×2				
	Up_sampling			Up_sampling					
	Conv_block-T	64×2		Conv_block-T	64×2				
Classifier	Dense	2048	12,636,168	Dense	2048	12,636,168	Dense	2048	12,636,168
	BatchNorm			BatchNorm			BatchNorm		
	Dropout			Dropout			Dropout		
	Dense	2048		Dense	2048		Dense	2048	
	BatchNorm			BatchNorm			BatchNorm		
	Dropout			Dropout			Dropout		
	Dense	2048		Dense	2048		Dense	2048	
	BatchNorm			BatchNorm			BatchNorm		
	Dropout			Dropout			Dropout		
	Dense	2048		Dense	2048		Dense	2048	
	BatchNorm			BatchNorm			BatchNorm		
	Dropout			Dropout			Dropout		
Discriminator	Conv	128	5,084,737	Conv	128	5,084,737	Conv	128	5,084,737
	Dropout			Dropout			Dropout		
	Conv	256		Conv	256		Conv	256	
	Dropout			Dropout			Dropout		
	Flatten			Flatten			Flatten		
	Dense	1024		Dense	1024		Dense	1024	
	Dense	1024		Dense	1024		Dense	1024	

Shared parameters: optimizer Adam, learning rate 0.0001, 7 dimensions
Hardware: GPU Testla T4 16Gb, CPU Xeon Processors @2.3Ghz
Software: Tensorflow 2.0 beta. The number of trainable parameters are reported by model.summary() from Keras library.

Algorithm 4.1 Algorithm for stochastic gradient descent training of ϵ -DR Privacy.

Input: Training dataset X.

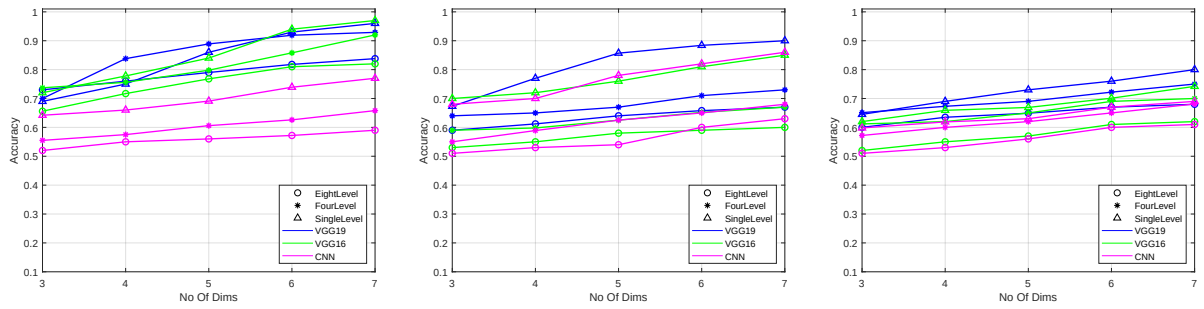
Parameter: learning rate $\alpha_r, \alpha_d, \alpha_c, \alpha_g$, training steps n_r, n_d, n_c, n_g

A constraint for ϵ-DR

Output: Transformation Model

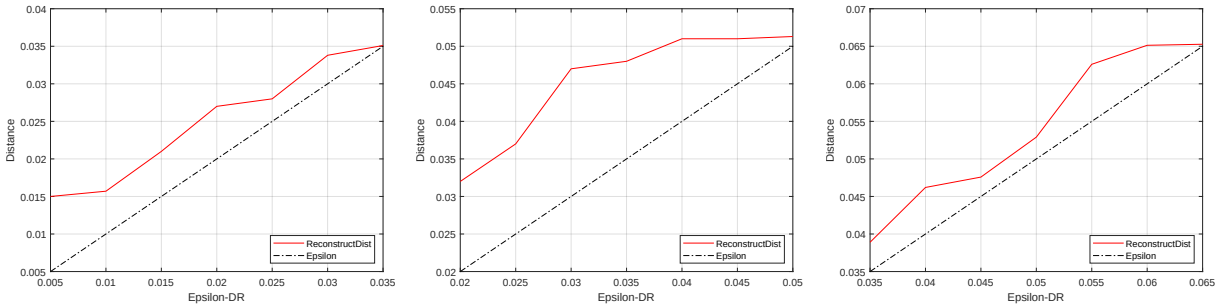
Initialization.

for n global training iterations **do**
  Randomly sample a mini batch from target distribution and label **t**.
  Randomly sample mini batch of data **x** and corresponding label **y**
  for $i = 0$ to n_r iterations **do**
    Update the Reconstruction:
    $\varphi_{i+1} = \varphi_i - \alpha_r \nabla_\varphi \mathcal{L}_R(\varphi_i, \mathbf{x})$
  end for
  for $j = 0$ to n_d iterations **do**
    Update the Discriminator parameter:
    $\omega_{j+1} = \omega_j - \alpha_d \nabla_\omega \mathcal{L}_D(\omega_j, \mathbf{x}, \mathbf{t})$
  end for
  for $k = 0$ to n_c iterations **do**
    Update the Classifier parameter:
    $\phi_{k+1} = \phi_k - \alpha_c \nabla_\phi \mathcal{L}_C(\phi_k, \mathbf{x}, \mathbf{y})$
  end for
  for $l = 0$ to n_g iterations **do**
    Update the Generator parameter:
    $\theta_{l+1} = \theta_l - \alpha_g \nabla_\theta \mathcal{L}_G(\theta_l, \mathbf{x}, \mathbf{t}, \mathbf{y})$
  end for
end for
return



(a) AT&T (b) Yale_B (c) CelebA

Figure 4.4: Accuracy for Different Number of Reduced Dimensions



(a) AT&T (b) Yale_B (c) CelebA

Figure 4.5: Average Distance Measurement Result { 7 dimensions, Single-Level}

Dataset (CelebA) [58]. To comprehensively evaluate our method performance, we also conduct experiments with different generator and re-constructor structures, different types of classifications (binary and multi-class classification), different numbers of reduced dimensions. The effectiveness of the method is then evaluated in terms of utility and privacy.

4.4.1 Experiment Setup

The Extended Yale Face Database B (YaleB) contains 2,470 grayscale images of 38 human subjects under different illumination conditions and their identity label. In this dataset, the image size is 168×192 pixels. The AT&T dataset has 400 face images of 40 subjects. For convenience, we resize each image of these two dataset to 64×64 pixels. CelebA is a color facial image dataset containing 202,599 images of 10,177 subjects. 1,709 images of the first 80 subjects are used for our experiment. Each image is resized to 64×64×3 pixels. All pixel values are scaled to the range of [0,1]. We randomly select 10% of each subject's images for validation and 15% for testing dataset.

The generator and re-constructor in Figure 4.3 are implemented by three different structures. Specifically, we follow the architecture of recent powerful models VGG19, VGG16 [80] and a basic convolutional network (CNN). We modify the models to adapt to our data size (64×64). Discriminator and Classifier are built on fully connected neural network and convolutional network respectively. Leaky ReLU is used for activation function in hidden

layers. We use linear activation function for generator's output layers and softmax activation functions for other components' output layers. Each component is trained in 5 local iterations (n_r, n_g, n_d, n_c), and the entire system is trained in 500 global iterations (n). The target distribution is drawn from Gaussian distribution (with the covariance value of 0.5 and the mean is the average of the training data). Table 4.1 provides detail information of neural networks' structures and other implementation information.

To evaluate the reliability, we test our framework with different levels of authentication corresponding to binary classification (single-level) and multi-class classification (multi-level). For the single-level authentication system, we consider half of the subjects in the dataset are valid to access company's resources while the rest are invalid. We randomly divide the dataset into two groups of subjects and labels their images to (1) or (0) depending on their access permission. For the cases of multi-level authentication system, we divide the subjects into four groups and eight groups. Therefore, the authentication server becomes four-class and eight-class classifier respectively.

4.4.2 Utility

We use accuracy metric to evaluate the utility of dimension-reduced data. The testing dataset is tested with the classifier extracted from our framework. Different structures of Generator and re-constructor are applied including VGG19, VGG16, basic CNN on different privilege levels which correspond to multi-class classification. Figure 4.4 illustrates the accuracies for different dimensions from three to seven over the three facial datasets. Overall, the accuracies improve when the number of dimension increases. The accuracies on the two gray image datasets (AT&T and Yale_B) reaches 90% and higher when using VGG with only seven dimensions. This accuracy figure for Celeba is smaller, but it still reaches 80%. In general, VGG19 structure performs better than using VGG16 and basic CNN in terms of utility due to the complexity (table 4.1) and adaptability to image datasets of VGG19. As the dimension number is reduced from 4,096 (64×64) to 7, we can achieve a compression

ratio of 585 yet achieve accuracy of 90% for the two gray datasets and 80% for the color dataset. This implies our method could gain a high compression ratio and maintain a high utility in terms of accuracy. During conducting experiments we also observe that the accuracy could be higher if we keep the original resolution of images. However, for convenience and reducing the complexity of our structure, we resize images to the size of 64×64 pixels.

4.4.3 Privacy

In this study, the Euclidean distance is used to measure the distance between original and reconstructed images: $dist(x, \hat{x}) = ||x - \hat{x}||^2$. Figure 4.5 illustrates the average distances between original images and reconstructed images on testing data with different ϵ constraints (other setting parameters: seven dimensions, single-level authentication, and VGG19 structure). The achieved distances (red lines) are larger than the hyper-parameter ϵ (black dotted lines) where ϵ is less than 0.035 for AT&T, 0.052 for YaleB and 0.067 for CelebA. Thus, our framework can satisfy ϵ-DR with ϵ of above values. Due to the fact that the re-constructor obtained some information (we consider the adversary can reach the model and the training data), we can only set the distance constraint ϵ within a certain range as shown in 4.5. The intersection between the red line and the dotted black line points out the largest distance our framework can achieve. Since the mean of the target distribution is set to be the same as the mean of training dataset, reconstructed images will be close to the mean of training dataset which we believe it will enlarge the distance and expose less individual information. Thus, the range of epsilon can be estimated base on the expectation of the distance between testing samples and the mean of training data. In addition, the first section of Table 4.2 demonstrates some samples and their corresponding reconstructions in single-level authentication and seven dimensions with different achieved accuracies and distances. The reconstructed images could be nearly identical, thus making it visually difficult to recognize the identity of an individual.

4.5 Comparison to GAP[13]

In this section, we compare the proposed framework with GAP, which shares many similarities. At first, we attempt to visualize AutoGAN-DRP and GAP by highlighting their similarities and differences. Then, we exhibit our experiment results of the two methods on the same dataset.

In terms of similarities, AutoGAN-DRP and GAP are utilizing minimax algorithms of Generative Adversarial Nets, applying the state-of-the-art convolution neural nets for image datasets, considering l_2 norm distance (i.e., distortion in GAP, privacy measurement in AutoGAN-DRP) between the original images and reconstructed images. Specifically, both GAP and AutoGAN-DRP consider the reconstruction distance between original and reconstructed images. In GAP this *distortion* refers to the Euclidean between original and privatized images, and AutoGAN-DRP denotes the *distance* as the Euclidean distance between original and reconstructed images. In this context, the distance and distortion refer to the same measurement and have the same meaning. To be consistent, we use the term *distance* to present this measurement in the rest of this section.

However, there are also distinctions between GAP and AutoGAN-DRP. In GAP, the adversary aims to identify a private label (e.g., gender) which should be kept secret while AutoGAN-DRP aims to visually protect the owner's face images by enlarging the reconstruction distance. Thus, instead of considering a private label in loss function of the generator in GAP, AutoGAN-DRP is aimed at driving the reconstructed data into a target distribution using a discriminator.

Figure 4.6 illustrates the visualization of AutoGAN-DRP and GAP. In AutoGAN-DRP, privacy is assessed based on how well an adversary can reconstruct the original data and measured by the distance between original and reconstructed data. The dimension-reduced data is reconstructed using the state-of-the-art neural network (an Auto-encoder). The larger the distance is, the more privacy can be achieved. Further, if the reconstructed images are blurry, privacy can be preserved since it is hard to visually determine an individual identity. The

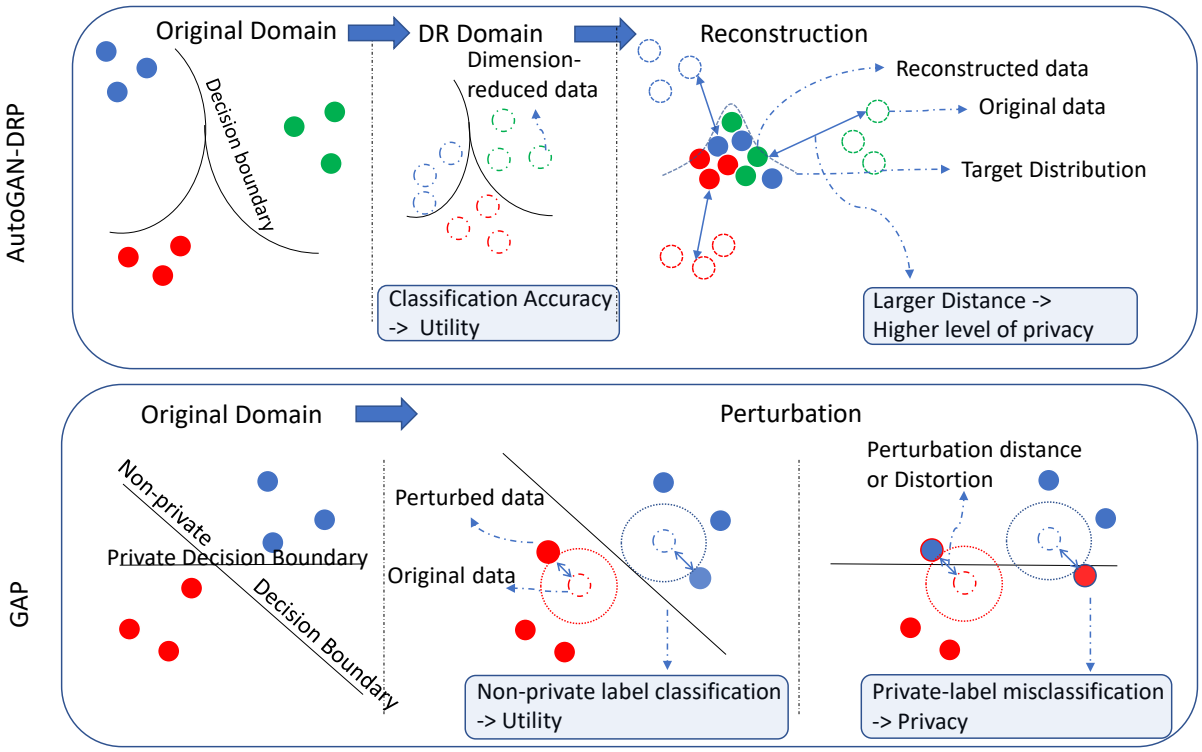


Figure 4.6: AutoGAN-DRP Vs GAP Explanation

data utility is quantified by the accuracy of the classification tasks over dimension-reduced data which captures the most significant data information. Meanwhile, GAP perturbs images with a certain distortion constraint to achieve privacy. It evaluates data utility by the classification accuracy of non-private label and assesses privacy by the classification accuracy of private label. Similar to AutoGAN-DRP, the high distortion is most likely to yield high level of privacy. In GAP, however, high distortion might dramatically reduce the classification accuracy of non-private label. This might be caused by the high correlation between private and non-private labels. This difference enables AutoGAN-DRP to preserve more utility than GAP at the same distortion level, as the experiment result (depicted in Figure 4.7) reveals.

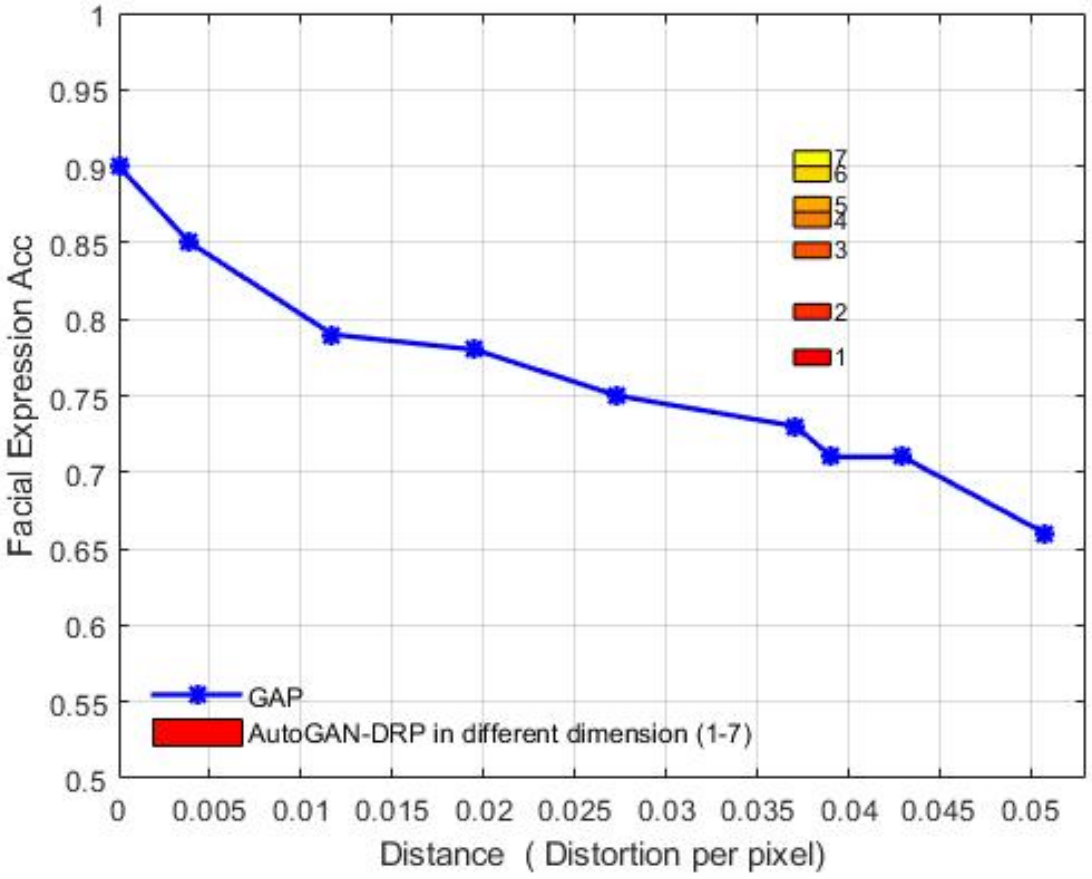


Figure 4.7: GENKI Facial Expression Accuracy Vs Distance using GAP and AutoGAN-DRP

In the experiment, we reproduce a prototype of Transposed Convolutional Neural Nets Privatizer (TCNNP) in GAP using materials and source code provided by [13]. We also modify our framework to make it as similar to TCNNP as possible. Specifically, a combination of two convolutional layers with ReLU activation function and two fully connected neural network layers are used for implementing the Generator similar to TCNNP. Our Classifier is constructed on two convolutional layers and two fully connected hidden layers similar to the Adversary in GAP. We also test our framework on GENKI, the same dataset with GAP. The utility is evaluated by the accuracy of facial expression classification (a binary classification). It should be noted that our framework have been shown to work on different datasets with multi-class classification, which is more challenging and comprehensive. Figure 4.7 shows the

accuracy results of GAP and AutoGAN-DRP for GENKI dataset. AutoGAN-DRP achieves distances ranging from 0.037 to 0.039 for different dimensions from one to seven. At the same range of distance (distortion per pixel), GAP achieves accuracy of only 72% while AutoGAN-DRP gains accuracy rates starting from 77% to 91% for different number of dimensions. It becomes evident that our method can achieve higher accuracy than that of GAP at the same distortion level.

4.6 Visual comparison to privacy preserving techniques using Differential Privacy (DP) [20] and Principle Component Analysis (PCA) [25]

In this section, we compare AutoGAN-DRP with other privacy preserving methods in terms of ability to visually identify client's identities. We choose the widely used tool for privacy preserving Differential Privacy (DP) [20] and another privacy preservation method utilizing dimensionality reduction technique (i.e., Principle Component Analysis [25]).

In these experiments, we implement AutoGAN-DRP following VGG19 structure for the Generator and Re-constructor, and other setting parameters (e.g., number of hidden layers, learning rate, optimization) are shown in Table 4.1. The images are reduced to seven dimensions for different values of ϵ-DR to achieve different distances and accuracies. The datasets are grouped into two groups corresponding to a binary classifier.

For implementing DP, we first generate a classifier on the authentication server by training the datasets with a VGG19 binary classifier (the structure of hidden layers is similar to our Generator in Table 4.1). The testing images are then perturbed using differential privacy method. Specifically, Laplace noise is added to the images with the sensitivity coefficient of 1 (it is computed by the maximum range value of each pixel [0,1]) and different DP epsilon parameters (this DP epsilon is different from our ϵ-DR). The perturbed images are then sent to the authentication server and fed to the classifier. We visually compare the perturbed images of this method with AutoGAN.

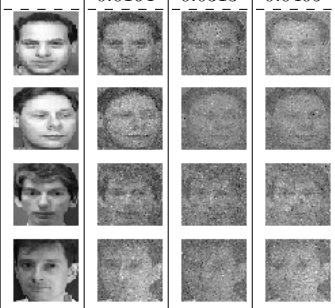
In addition, we follow instruction in FRiPAL [98] in which the clients reduce image dimension using Principle Component Analysis (PCA) and send reduced features to the server. FRiPAL claims that by reducing image dimension, their method can be more resilient to reconstruction attacks. The experiments are conducted with different number of reduced dimension. The images are reconstructed using *Moore–Penrose inverse* method with assumption that an adversary has assess to the model. The classification accuracy is evaluated using a classifier which has similar structure to AutoGAN's classifier.

Table 4.2 shows image samples and results over the three datasets. Overall, AutoGAN-DRP is more resilient to reconstruction attacks compared to the other two techniques. For instance, at the accuracy of 79% on AT&T dataset, 80% on YaleB, and 73% on CelebA, we cannot distinguish entities from the others. For DP method, the accuracy decreases when the DP epsilon decreases (adding more noise), and the perturbed images become harder to recognize. However, at a low accuracy 57%, we are still able to distinguish identities by human eyes. The reason is that DP noise does not focus on the important visual pixels. For PCA, the accuracy also goes down when the number of dimensions decreases and the distances increase. Since PCA transformation is linear and deterministic, the original information can be significantly reconstructed using the inverse transformation deriving from the model or training data. Thus, at the accuracy of 75% on AT&T, 71% on YaleB, and 68% on CelebA, we still can differentiate individuals. Overall, our proposed method shows the advantage in securing the data while retaining high data utility.

4.7 Conclusion

In this work, we introduce a mathematical tool ϵ-DR to evaluate privacy preserving mechanisms. We also propose a non-linear dimension reduction framework. This framework projects data onto lower dimension domain in which it prevents reconstruction attacks and preserves data utility. The dimension-reduced data can be used effectively for the machine learning tasks such as classification. In our future works, we plan to extend the framework

Table 4.2: Sample visualization of AutoGAN, DP, PCA over three datasets

	AT&T				YaleB				CelebA			
Acc		0.93	0.79	0.65		0.90	0.80	0.69		0.73	0.66	0.59
Dist		0.0116	0.0198	0.0245		0.0184	0.0246	0.0585		0.0513	0.0531	0.06618
AutoGAN-DRP												
	Org	(7)	(7)	(7)	Org	(7)	(7)	(7)	Org	(7)	(7)	(7)
Acc		0.69	0.63	0.57		0.68	0.60	0.58		0.62	0.59	0.56
Dist		0.0164	0.0313	0.0405		0.0149	0.0314	0.0407		0.0200	0.0418	0.0509
Differential Privacy												
	Org	(11)	(8)	(7)	Org	(11)	(8)	(7)	Org	(11)	(8)	(7)
Acc		0.90	0.75	0.60		0.87	0.83	0.71		0.71	0.68	0.57
Dist		0.0197	0.0264	0.0348		0.0228	0.0266	0.0287		0.0362	0.0379	0.0511
PCA												
	Org	(15)	(7)	(5)	Org	(15)	(7)	(5)	Org	(15)	(7)	(5)

Acc : Average accuracy on testing data
Dist: Average Euclidean distance between original images and reconstructed/perturbed images
Org : Original images
(.) Experiment parameters: epsilon for DP and number of reduced dimensions for PCA and AutoGAN-DRP

to adapt with different types of data, such as time series and categorical data. We will apply different metrics to compute the distance other than l_2 norm and investigate the framework on several applications in security systems and data collaborative contributed systems.

References

[1] Maximal and minimal points of functions theory.

[2] Sphere, Aug 2021.

[3] Martín Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. (Ccs), 2016.

[4] Umang Aggarwal, Adrian Popescu, and Celine Hudelot. Active Learning for Imbalanced Datasets. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1417–1426, Snowmass Village, CO, USA, March 2020. IEEE.

[5] Mohammad Al-rubaie, Pei-yuan Wu, J Morris Chang, and Sun-yuan Kung. Privacy-Preserving PCA on Horizontally-Partitioned Data.

[6] Pierre Baldi. Autoencoders, Unsupervised Learning, and Deep Architectures. *ICML Unsupervised Transf. Learn.*, pages 37–50, 2012.

[7] Raphael Bost, Raluca Ada Popa, Stephen Tu, and Shafi Goldwasser. Machine learning classification over encrypted data. 01 2015.

[8] Raphaël Bost, Ra Popa, Stephen Tu, and S Goldwasser. Machine Learning Classification over Encrypted Data. *Ndss '15*, (February):1–31, 2015.

[9] Kamalika Chaudhuri. Privacy-preserving logistic regression. pages 1–8.

[10] Kamalika Chaudhuri and Claire Monteleoni. Privacy-preserving logistic regression. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 289–296. Curran Associates, Inc., 2009.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, June 2002.

[12] Xiao Chen, Peter Kairouz, and Ram Rajagopal. Understanding compressive adversarial privacy. *CoRR*, abs/1809.08911, 2018.

[13] Xiao Chen Lalitha Sankar Ram Rajagopal Chong Huang, Peter Kairouz. Generative Adversarial Privacy. *Privacy in Machine Learning and Artificial Intelligence Workshop, ICML 2018*, 2018.

[14] Muhammad Enamul Hoque Chowdhury, Tawsifur Rahman, Amith Khandakar, Rashid Mazhar, Muhammad Abdul Kadir, Zaid Bin Mahbub, Khandakar Reajul Islam, Muhammad Salman Khan, Atif Iqbal, Nasser Al-Emadi, and Mamun Bin Ibne Reaz. Can AI help in screening viral and COVID-19 pneumonia? *CoRR*, abs/2003.13145, 2020.

[15] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-Balanced Loss Based on Effective Number of Samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, Long Beach, CA, USA, June 2019. IEEE.

[16] Wangzhi Dai, Kenney Ng, Kristen Severson, Wei Huang, Fred Anderson, and Collin Stultz. Generative Oversampling with a Contrastive Variational Autoencoder. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 101–109, Beijing, China, November 2019. IEEE.

[17] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[18] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[19] Jian-Hui Duan, Wenzhong Li, and Sanglu Lu. *FedDNA: Federated Learning with Decoupled Normalization-Layer Aggregation for Non-IID Data*, pages 722–737. 09 2021.

[20] Cynthia Dwork. Differential privacy. *Proc. 33rd Int. Colloq. Autom. Lang. Program.*, pages 1–12, 2006.

[21] F. Emekci, O. D. Sahin, D. Agrawal, and A. El Abbadi. Privacy preserving decision tree learning over multiple parties. *Data Knowl. Eng.*, 63(2):348–361, 2007.

[22] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07*, page 127, Lisbon, Portugal, 2007. ACM Press.

[23] Seyda Ertekin, Jian Huang, and C. Lee Giles. Active learning for class imbalance problem. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '07*, page 823, Amsterdam, The Netherlands, 2007. ACM Press.

[24] Ines Feki, Sourour Ammar, Yousri Kessentini, and Khan Muhammad. Federated learning for COVID-19 screening from Chest X-ray images. *Applied Soft Computing*, 106:107330, July 2021.

[25] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[26] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks, 2016.

[27] Yonatan Geifman and Ran El-Yaniv. Deep active learning over the long tail. *CoRR*, abs/1711.00941, 2017.

[28] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 23(6):643–660, 2001.

[29] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 881–889, Lille, France, 07–09 Jul 2015. PMLR.

[30] Daniel Gissin and Shai Shalev-Shwartz. Discriminative active learning, 2019.

[31] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. pages 1–9, 2014.

[32] Dr Saptarsi Goswami. Class Imbalance, SMOTE, Borderline SMOTE, ADASYN, November 2020.

[33] Gokhan Goy, Cengiz Gezer, and Vehbi Cagri Gungor. Credit Card Fraud Detection with Machine Learning Methods. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, pages 350–354, Samsun, Turkey, September 2019. IEEE.

[34] Haibo He and E.A. Garcia. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, September 2009.

[35] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In De-Shuang Huang, Xiao-Ping Zhang, and Guang-Bin Huang, editors, *Advances in Intelligent Computing*, pages 878–887, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[36] Andrew Hard, Kanishka Rao, Rajiv Mathews, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *CoRR*, abs/1811.03604, 2018.

[37] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

[38] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. CryptoDL : Deep Neural Networks over Encrypted Data. pages 1–21, 2017.

[39] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. *CoRR*, abs/1711.05189, 2017.

[40] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning. 2017.

[41] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning Deep Representation for Imbalanced Classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5375–5384, Las Vegas, NV, USA, June 2016. IEEE.

[42] J.J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[43] S. Itahara, T. Nishio, Y. Koda, M. Morikura, and K. Yamamoto. Distillation-based semi-supervised federated learning for communication-efficient collaborative training with non-iid private data. *IEEE Transactions on Mobile Computing*, (01):1–1, mar 5555.

[44] Stefan Jaeger, Sema Candemir, Sameer Antani, Yì-Xiáng Wáng, Pu-Xuan Lu, and George Thoma. Two public chest x-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative imaging in medicine and surgery*, 4:475–7, 12 2014.

[45] Paul A. Jensen. Algorithms for constrained optimization. https://www.me.utexas.edu/~jensen/ORMM/supplements/units/nlp_methods/const_opt.pdf.

[46] Chi Jin, Lydia T. Liu, Rong Ge, and Michael I Jordan. On the local minima of the empirical risk. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[47] Daniel S. Kermany, Kang Zhang, and Michael H. Goldbaum. Labeled optical coherence tomography (oct) and chest x-ray images for classification. 2018.

[48] S. Y. Kung. Compressive privacy: From informationestimation theory to machine learning [lecture notes]. *IEEE Signal Processing Magazine*, 34(1):94–112, Jan 2017.

[49] S.Y. Kung. A compressive privacy approach to generalized information bottleneck and privacy funnel problems. *Journal of the Franklin Institute*, 355, 07 2017.

[50] S. Leroueil. Compressibility of Clays: Fundamental and Practical Aspects. *Journal of Geotechnical Engineering*, 122(7):534–543, July 1996.

[51] Lusi Li, Haibo He, and Jie Li. Entropy-based Sampling Approaches for Multi-Class Imbalanced Problems. *IEEE Transactions on Knowledge and Data Engineering*, 32(11):2159–2170, November 2020.

[52] Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. *CoRR*, abs/2102.02079, 2021.

[53] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *MLSys*, 2020.

[54] Xiaoxiao Li, Meirui JIANG, Xiaofei Zhang, Michael Kamp, and Qi Dou. FedBN: Federated learning on non-IID features via local batch normalization. In *International Conference on Learning Representations*, 2021.

[55] Alexander Liu, Joydeep Ghosh, and Cheryl E. Martin. Generative oversampling for mining imbalanced datasets. In Robert Stahlbock, Sven F. Crone, and Stefan Lessmann, editors, *Proceedings of the 2007 International Conference on Data Mining, DMIN 2007, June 25-28, 2007, Las Vegas, Nevada, USA*, pages 66–72. CSREA Press, 2007.

[56] Lumin Liu, Jun Zhang, Shenghui Song, and Khaled Ben Letaief. Edge-assisted hierarchical federated learning with non-iid data. *CoRR*, abs/1905.06641, 2019.

[57] Yang Liu, Xiang Li, Xianbang Chen, Xi Wang, and Huaqiang Li. High-Performance Machine Learning for Large-Scale Data Classification considering Class Imbalance. *Scientific Programming*, 2020:1–16, May 2020.

[58] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[59] Dhruv Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. *CoRR*, abs/1805.00932, 2018.

[60] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282. PMLR, 20–22 Apr 2017.

[61] Aditya Menon and Cheng Soon Ong. Linking losses for density ratio and class-probability estimation. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 304–313, New York, New York, USA, 20–22 Jun 2016. PMLR.

[62] Mehryar Mohri, Gary Sivek, and Ananda Theertha Suresh. Agnostic federated learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4615–4625. PMLR, 09–15 Jun 2019.

[63] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Generative Adversarial Minority Oversampling. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1695–1704, Seoul, Korea (South), October 2019. IEEE.

[64] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[65] NVIDIA, Péter Vingelmann, and Frank H.P. Fitzek. Cuda, release: 10.2.89, 2020.

[66] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.

[67] Tb Pedersen, Y Saygın, and E Savaş. Secret charing vs. encryption-based techniques for privacy preserving data mining. *Fac. Eng. Nat. Sci. Sabanci Univ. Istanbul, TURKEY*, 1:1–11, 2007.

[68] Nhathai Phan, Yue Wang, Xintao Wu, and Dejing Dou. Differential Privacy Preservation for Deep Auto-Encoders: An Application of Human Behavior Prediction. *Aaai*, pages 1309–1316, 2016.

[69] Zhicong Qiu, David J. Miller, and George Kesidis. A maximum entropy framework for semisupervised and active learning with unknown and label-scarce classes. *IEEE Transactions on Neural Networks and Learning Systems*, 28(4):917–933, 2017.

[70] Vivek Kumar Rangarajan Sridhar. Unsupervised Text Normalization Using Distributed Representations of Words and Phrases. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 8–16, Denver, Colorado, 2015. Association for Computational Linguistics.

[71] M.H. Rashid, J.H. Wang, and C. Li. Convergence analysis of a method for variational inclusions. *Applicable Analysis*, 91(10):1943–1956, October 2012.

[72] Prasun Roy, Subhankar Ghosh, Saumik Bhattacharya, and Umapada Pal. Effects of degradations on deep neural network architectures. *arXiv preprint arXiv:1807.10108*, 2018.

[73] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet S. Talwalkar, and Virginia Smith. On the convergence of federated optimization in heterogeneous networks. *ArXiv*, abs/1812.06127, 2018.

[74] F. S. Samaria and A. C. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, pages 138–142, Dec 1994.

[75] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP '08*, page 1070, Honolulu, Hawaii, 2008. Association for Computational Linguistics.

[76] Adi Shamir. How to share a secret. *Commun. ACM*, pages 612–613, 1979.

[77] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, July 1948.

[78] Li Shen, Zhouchen Lin, and Qingming Huang. Learning deep convolutional neural networks for places2 scene recognition. *CoRR*, abs/1512.05830, 2015.

[79] Tao Shen, J. Zhang, Xinkang Jia, Fengda Zhang, Gang Huang, Pan Zhou, Fei Wu, and Chao Wu. Federated mutual learning. *ArXiv*, abs/2006.16765, 2020.

[80] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[81] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, pages 1698–1707, 2020.

[82] Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. In *International Conference on Learning Representations*, 2020.

[83] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in Neural Information Processing Systems*, 2020-December, 2020.

[84] Zhenyu Wu, Zhangyang Wang, Zhaowen Wang, and Hailin Jin. Towards privacy-preserving visual recognition via adversarial training: A pilot study. *CoRR*, abs/1807.08379, 2018.

[85] Lucila Ohno-Machado Xiaoqian Jiang, Zhanglong Ji, Shuang Wang, Noman Mohammed, Samuel Cheng. Differential-Private Data Publishing Through Component Analysis. *Trans. data Priv.*, 6(1):19–34, 2013.

[86] Kun Xie, Xueping Ning, Xin Wang, Jigang Wen, Xiaoxiao Liu, Shiming He, and Daqiang Zhang. An efficient privacy-preserving compressive data gathering scheme in wsns. In Guojun Wang, Albert Zomaya, Gregorio Martinez, and Kenli Li, editors, *Algorithms and Architectures for Parallel Processing*, pages 702–715, Cham, 2015. Springer International Publishing.

[87] Qian Ya-Guan, Ma Jun, Zhang Xi-Min, Pan Jun, Zhou Wu-Jie, Wu Shu-Hui, Yun Ben-Sheng, and Lei Jing-Sheng. EMSGD: An Improved Learning Algorithm of Neural Networks With Imbalanced Data. *IEEE Access*, 8:64086–64098, 2020.

[88] Satya Prakash Yadav, Bhoopesh Singh Bhati, Dharmendra Prasad Mahato, Sachin Kumar, and SpringerLink (Online service). *Federated Learning for IoT Applications*. Springer International Publishing Imprint Springer., Cham, 2022. OCLC: 1295622946.

[89] M. Yang, T. Zhu, B. Liu, Y. Xiang, and W. Zhou. Machine learning differential privacy with multifunctional aggregation in a fog computing architecture. *IEEE Access*, 6:17119–17129, 2018.

[90] Andrew Chi-Chih Yao. How to generate and exchange secrets. *27th Annu. Symp. Found. Comput. Sci. (sfcs 1986)*, (1):162–167, 1986.

[91] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. Bayesian nonparametric federated learning of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7252–7261. PMLR, 09–15 Jun 2019.

[92] Shaolei Zhai, Xin Jin, Ling Wei, Hongxuan Luo, and Min Cao. Dynamic Federated Learning for GMEC With Time-Varying Wireless Link. *IEEE Access*, 9:10400–10412, 2021.

[93] Jun Zhang, Zhenjie Zhang, Xiaokui Xiao, Yin Yang, and Marianne Winslett. Functional Mechanism: Regression Analysis under Differential Privacy. pages 1364–1375, 2012.

[94] Xinwei Zhang, Mingyi Hong, Sairaj V. Dhople, Wotao Yin, and Yang Liu. Fedpd: A federated learning framework with optimal rates and adaptivity to non-iid data. *CoRR*, abs/2005.11418, 2020.

[95] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *ArXiv*, abs/1806.00582, 2018.

[96] Shuheng Zhou, Katrina Ligett, and Larry Wasserman. Differential privacy with compression. *CoRR*, 2009.

[97] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

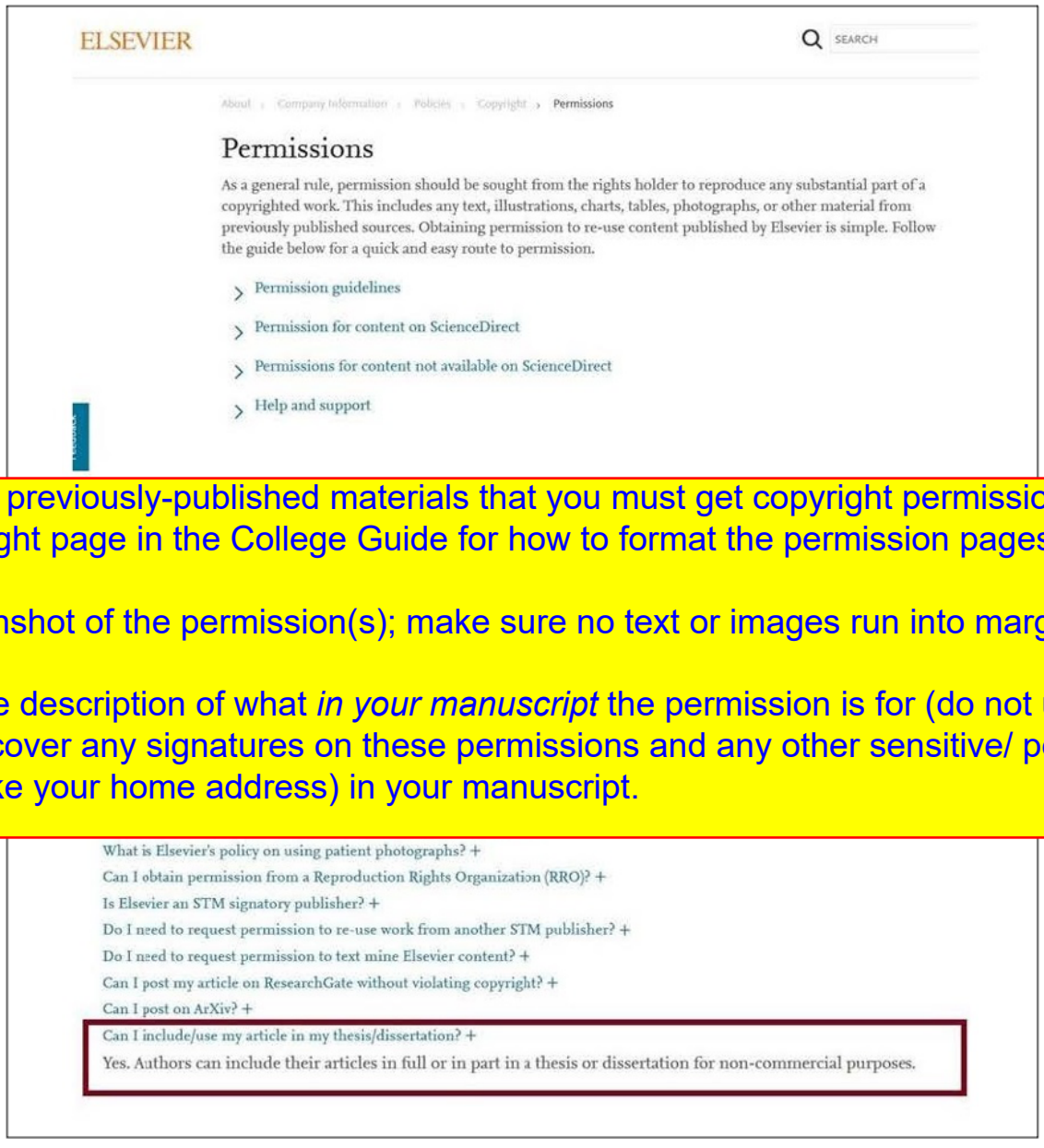
[98] Di Zhuang, Sen Wang, and J Morris Chang. Fripal: Face recognition in privacy abstraction layer. In *2017 IEEE Conference on Dependable and Secure Computing*, pages 441–448. IEEE, 2017.

Appendix A: Appendix

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

Appendix B: Copyright Permissions

The permission below is for the reproduction of material in Chapter ??.



If you *do* have previously-published materials that you must get copyright permissions for, see sample copyright page in the College Guide for how to format the permission pages.

Use jpg screenshot of the permission(s); make sure no text or images run into margins.

Use text as the description of what *in your manuscript* the permission is for (do not use figure titles!). Also, cover any signatures on these permissions and any other sensitive/ personal information (like your home address) in your manuscript.