

Robust Representation Learning for Privacy-Preserving Machine Learning: A Multi-Objective Autoencoder Approach

Sofiane Ouaari^{1,2}, Ali Burak Ünal^{1,2,3}, Mete Akgün^{1,2,3} and Nico Pfeifer^{1,2}

¹Methods in Medical Informatics, Department of Computer Science, University of Tuebingen, Germany

²Institute for Bioinformatics and Medical Informatics (IBMI), University of Tuebingen, Germany

³Medical Data Privacy and Privacy Preserving Machine Learning, University of Tuebingen, Germany
{sofiane.ouaari, ali-burak.uenal, mete.akguen, nico.pfeifer}@uni-tuebingen.de

¹ **Abstract**—Several domains increasingly rely on machine learning in their applications. The resulting heavy dependence on data has led to the emergence of various laws and regulations around data ethics and privacy and growing awareness of the need for privacy-preserving machine learning (ppML). Current ppML techniques utilize methods that are either purely based on cryptography, such as homomorphic encryption, or that introduce noise into the input, such as differential privacy. The main criticism given to those techniques is the fact that they either are too slow or they trade off a model’s performance for improved confidentiality. To address this performance reduction, we aim to leverage robust representation learning as a way of encoding our data while optimising the privacy-utility trade-off. Our method centers on training autoencoders in a multi-objective manner and then concatenating the latent and learned features from the encoding part as the encoded form of our data. Such a deep learning-powered encoding can then safely be sent to a third party for intensive training and hyperparameter tuning. With our proposed framework, we can share our data and use third party tools without being under the threat of revealing its original form. We empirically validate our results on unimodal and multimodal settings, the latter following a vertical splitting system and show improved performance over state-of-the-art.

I. INTRODUCTION

A wide range of application sectors is drastically integrating machine learning (ML) in diverse products. A successful ML model often requires a huge amount of training data and powerful computational resources. However, the need for such enormous volumes of data to develop performing models raises serious privacy concerns. Such ML models might face multiple types of adversarial attacks depending on the type of access an adversary might have to the model (white or black-box). A membership inference attack (Shokri et al., 2017) allows an attacker to query a trained machine learning model to predict whether a given example is in the model’s training data set. On the other hand, an inversion attack (Fredrikson et al., 2015; K.-C. Wang et al., 2021; Ye et al., 2022) aims to recreate an input data point given a confidence score obtained from a black-box inference of the model. In order to make researchers and engineers take such privacy threats into consideration, many regulations and ethical data policies, such as GDPR,

CCPA, and CPRA (Hijmans & Raab, 2018; Rochel, 2021) were set to raise awareness around this topic and restrict any data violations that might occur in a given ML pipeline. Previous works have been done to reduce the effectiveness of different privacy attacks. Among those studies, differential privacy (DP) is the most commonly used approach which operates by incorporating predetermined randomization into a machine learning algorithm’s computation. The perturbation introduced by DP might be applied on the users’s input, parameters, prediction output and even on loss functions (Abadi et al., 2016; Phan et al., 2016). However, many studies have shown that such noise reduces the performance of the model for the sake of privacy (Truex et al., 2019). Furthermore, Setting up a correct value for ϵ is complex by nature and requires some trial and error process.

Homomorphic encryption (HE) is a cryptographic method applied in the domain of ppML. It is defined as a type of encryption method that allows computations to be performed on encrypted data without first decrypting it with a secret key. Yet HE has some limitations. It was originally designed to allow only algebraic operations such as addition and multiplication which excludes the various non-linear activation functions leveraged in neural networks. Numerous studies have been conducted to approximate such functions using polynomials (Hesamifard et al., 2017; J.-W. Lee et al., 2022; J. Lee et al., 2021), however such approximations result in high computational burden and reduce the ability to apply various methods by extending the depth of deep learning models, since performing all that in a HE fashion significantly increases computation time.

The aim behind this paper is to create a deep learning-oriented encoding strategy by training a supervised residual autoencoder and concatenating the features learned from the encoder part as the representation to be shared with third parties for further training and an extensive hyperparameters search. A framework, which trains an autoencoder and shares its latent space embedding with other parties for data sharing purposes was presented by María Quintero-Ossa et al., 2022. We consider such a framework as a baseline in our experiments and empirically demonstrate (see Section V-B) that our suggested architecture considerably improves the

¹Preprint. Under review

performance. In contrast to María Quintero-Ossa et al., 2022, we also provide a threat analysis to discuss how secure our framework is and the different types of access that an adversary might have, considering different actors directly involved and interacting with our framework. The autoencoder proposed in our framework is trained in a multi-objective fashion by simultaneously considering the data reconstruction and the supervised learning problem, and ensuring an informative and discriminative representation. This ppML encoding is applied to the data part of the machine learning pipeline and we empirically demonstrate the efficiency of this method by first experimenting on unimodal settings using the MNIST, FashionMNIST, Leukemia, and Retinal OCT datasets. In addition, we further explored the capabilities offered by our ppML framework on multimodal data distributed in a vertical setting, where each modality is provided by a given data party. For this purpose, a TCGA multi-omics breast cancer dataset was leveraged. We summarize our main contributions as follows:

- We developed a considerably improved version of the data sharing strategy through latent space embedding proposed by María Quintero-Ossa et al., 2022 by increasing the performance on different prediction tasks and providing a detailed threat analysis.
- We demonstrated the application flow of our proposed encoding framework for both unimodal and multimodal (vertically distributed) settings.
- We empirically validated our approach to be utility-privacy efficient by comparing models trained on the original data against models trained on the generated encoded data and show that both perform equally.

II. BACKGROUND & RELATED WORK

In this section, we present previous works that have been performed in the sphere of autoencoders, representation learning and ppML methods enhanced with deep learning.

A. Autoencoders

Autoencoders are a type of neural network originally implemented to solve the unsupervised task of data reconstruction. Formally, an autoencoder is defined with three main parts, which are an encoder $E(\cdot)$, a latent space representation s and a decoder $D(\cdot)$. Given $x, \hat{x} \in \mathbb{R}^d$ and $s \in \mathbb{R}^m$ we have $s = E(x)$ and $\hat{x} = D(s)$ with $m \ll d$ and $\hat{x}$ defined as the reconstructed output of the original input x . Beyond its original purpose, the use of autoencoders were extended to other applications such as data generation with variational autoencoders (Kingma & Welling, 2013), anomaly detection (Sakurada & Yairi, 2014) and recommendation systems (Ferreira et al., 2020; Pan et al., 2020). They were also leveraged for supervised learning purposes, Le et al., 2018 implemented a supervised autoencoder (SAE) where the latent space s is linked to a classifier f_c trained in parallel with the original data reconstruction problem and the overall loss function defined as follows:

$$L(x, y, \theta_e, \theta_d, \theta_c) = \frac{1}{t} \sum_{i=1}^t L_r(x, D(E(x, \theta_e), \theta_d)) + L_c(y, f_c(E(x, \theta_e), \theta_c)) \quad (1)$$

With θ_e , θ_d and θ_c being the parameters of the encoder, decoder and classifier respectively and $L_r(\cdot, \cdot)$, $L_c(\cdot, \cdot)$ defined as reconstruction and categorical cross entropy losses.

B. Representation learning

A good encoding demands an informative representation of the original data by reducing the dimension of the input without lowering the inter-dependencies and the important relations needed for a given ML model to perform efficiently in a given task. A reasonable-sized learnt representation might encompass a vast array of potential input configurations, because good representations are expressive. Bengio et al. (2013) surveyed what makes the essence of a good representation which we took into consideration while developing our framework.

- *Smoothness*: Given $x, y \in \mathbb{R}^d$ and a representation function $f(\cdot)$ defined as: $f : \mathbb{R}^d \rightarrow \mathbb{R}^m$, where $m < d$, a smooth representation implies that if $x \approx y$ then $f(x) \approx f(y)$. In a geometric point of view using distances this translates to: if $\text{dist}(x_1, x_2) < \text{dist}(x_1, x_3)$ then $\text{dist}(f(x_1), f(x_2)) < \text{dist}(f(x_1), f(x_3))$. This is a reason why we introduced a center loss in our framework which is explained in Section III-C1 to keep a similar distance semantics when encoding the data and mapping it to a space with reduced dimension.
- *Sparsity & Invariance*: for any given observation in x , only a small fraction of the possible factors are relevant. In terms of representation, this could be represented by features that are often zero or by the fact that most of the extracted features are insensitive to small variations of x . For our case, this can be achieved through sparse autoencoders (Rangamani et al., 2018) which is a sort of autoencoder that uses sparsity to create a bottleneck in the flow of information. In particular, the loss function is designed to punish the activation generated by the layer. L1 regularization is usually used to apply the sparsity constraint.

C. Privacy enhancement with deep learning

Linking deep learning training concepts for the purpose of ppML has been addressed before. Adversarial learning is the most frequent way to handle privacy-utility trade-off where the privacy variable is explicitly introduced in the adversarial objective. Mandal et al. (2022) presented UAE-PUPET where an autoencoder takes an input x and generates $\hat{x}$ being a distorted version of x that contains minimum information of a given private attribute x_p while always keeping the most important information of the targeted utility variable x_u . UAE-PUPET works by linking 2 classifiers to the autoencoder γ_u , γ_p responsible on predicting the utility and private attributes respectively, then adversarially minimizing L_u and maximizing L_p while added to the original reconstruction loss. However, such a method always requires choosing only one privacy variable at a time which is not always the case, in addition the authors did not run a clear threat analysis through analyzing how a harmful attack as model inversion might be applied against the generated $\hat{x}$.

T. Xiao et al. (2020) proposed an adversarial reconstruction

learning framework that prevents the latent representations to decode into the original input data. In other words, this time the reconstruction loss is directly maximized while minimizing the utility prediction error. Even though such a method allowed to empirically reduce the effectiveness of a model inversion attack, it is theoretically still possible and not inevitable. In addition the authors also noted that adversaries may still be able to exploit private meta information, such as determining the ethnicity of an individual based on skin color presented in reconstructed images, despite the fact that the images are not identical to the input.

Training a model $\mathcal{M}$ with generated synthetic data is another ppML with deep learning approach. Previous works were performed to train Generative Adversarial Neural networks (GANs) using differential privacy DP-GAN (Cao et al., 2021; Harder et al., 2021; Y. Wang et al., 2021). Yet, DP leads to the injection of noise in the generated data, which might reduce the model’s performance especially in complex data like genomics where the data suffers from the curse of dimensionality and where one single gene can decide the main task’s outcome, DP might result in a further performance loss (Chen et al., 2020). Dong et al. (2022) were the first to introduce dataset condensation technique into the domain of ppML. Data condensation (Zhao et al., 2021) works by generating synthetic data through condensing the larger original data points into a smaller subset. It allows a data-efficient learning since the synthetic data are created by first randomly initializing them, afterwards two neural networks are trained where one is linked to the original data $\mathcal{T}$ and the other to the synthetic data $\mathcal{S}$ and then the latter is updated iteratively by back-propagating the loss in a way that both networks share the same weights. Dong et al. (2022) theoretically proved that dataset condensation is similar to differential privacy from the perspective that one sample has limited effect over the parameter distribution of a network. Then they empirically showed that in addition to keeping good performance a model trained with the synthetic data was also more robust towards membership inference attacks. However, such a method does present some limitations. First, the membership inference risk was still present even though it was decreased, in fact the authors even found out that in the case of FashionMNIST the membership attack was more effective on the model trained with synthetic data $\mathcal{S}$ than on its original counterpart due to the grey-scale nature of the image, assuming that the synthetic data might contain more features prone to be memorized. Furthermore, dataset condensation initially requires a large amount of data, which excludes databases with few samples.

III. FRAMEWORK ARCHITECTURE & METHODOLOGY

A. Residual Autoencoder

Ensuring the training stability in a neural network raises several questions, especially on how to correctly set the number of hidden layers to be used. Wickramasinghe et al. (2021) showed that residual autoencoders (RAE) performed well and in a stable manner while ranging the number of repeated layers from 2 to 90 on MNIST, FashionMNIST and CIFAR. For this stability purpose, we decided to use the residual autoencoder as

the main backbone of our encoding framework. By extension, we used a Convolution-RAE (C-RAE) when encoding images (Fig 1). Then, every layer in the encoder part including the latent space is forwarded in parallel to a classifier. Formally,

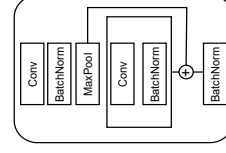


Figure 1: Building block of the Convolution Residual Autoencoder (C-RAE) in the encoder part $E(\cdot)$.

each hidden layer of the encoder $X_{enc}^{(i)} \in \mathbb{R}^{e_i}$, with $e_l < e_{i+1} < e_i < d$ and d, e_l being the respective dimensions of the input data and the latent space ($e_l < d$), is separately fed into a neural network classifier $f_i(\cdot)$ to perform the original classification task assigned to it $\hat{y}_i = f_i(X_{enc}^{(i)}, \theta_i)$, $\hat{y}_i \in C$ and i is the index of layers included in the encoder $E(\cdot)$ with the latent space taking part as well.

B. Application Flow

In this section, we explain the application pipeline of our encoding framework in both unimodal and multimodal schemes.

1) *Unimodal Setting*: After training the supervised residual autoencoder (SRAE) of our framework on a dataset $\{x_j, y_j\}$, we use the encoder part $E(\cdot)$ of it to compute different representation of a given sample x_j until the layer producing the latent space as defined in the previous section. The final encoding Ψ to be shared is defined simply as the concatenation of those representations (Eq. 2).

$$\Psi = (X_{enc}^1, X_{enc}^2, \dots, X_{enc}^l) \quad (2)$$

with $X_{enc}^{(i)} \in \mathbb{R}^{e_i}$ and $\Psi \in \mathbb{R}^{\sum_{i=1}^l e_i}$. In Fig 2, we illustrate how the encoding pipeline is generated by our proposed framework for a unimodal scenario.

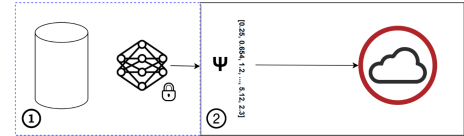


Figure 2: Execution flow of our ppML encoding framework on unimodal data, in ① our supervised residual autoencoder is trained in-house on the original data, then in ② the generated concatenated encoding Ψ will be sent and shared with another party (for example: cloud) for the purpose of performing a heavy hyper-parameters tuning and further computationally expensive experiments.

We would like to highlight and point out that in the unimodal setting our encoding framework is helpful when further computational power from external resources is required, for instance to evaluate different hyperparameters settings and combinations and/or the use of complex ensemble and mixture of experts models. In other words, the trained residual autoencoder is

kept in-house and only the encoded data is shared with the server providing the compute resources allowing us to benefit from cloud services without being under the threat of revealing our original data x . Our framework can also be leveraged to permit other institutions, when requested, to benefit from the inference of our model by sharing with them both the encoder $E(\cdot)$ and the final model trained on the cloud $\mathcal{M}_\Psi$.

2) *Multimodal Setting*: In this section of the paper we explain how to extend the utility of our framework in a multimodal use case. Let us consider a multimodal dataset consisting of m modalities $\{x_j^{(m)}, y_j\}$ and distributed in a vertical setting over m data providers where basically each modality is stored in a data supplier. For this specific scenario,

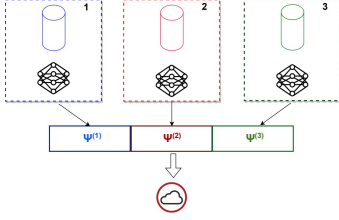


Figure 3: Execution flow when applying our ppML encoding framework on a multimodal use case with vertically distributed data

the adopted strategy works by training a supervised residual autoencoder on every modality separately at the data provider. Each one then sends in parallel its final encoding $\Psi^{(m)}$ to a third party system to perform a cooperative training (Fig 3). Such a workflow permits mutual work between various providers by sending good representations of the input thus not hurting the performance and in the same time guaranteeing total privacy of the data since no information about the original data format or properties are revealed including the original shape, the distribution, the type of modality (image, sound, tabular...) and if the data-type is homogeneous (only numerical, only categorical) or heterogeneous.

For the sake of a better understanding of how the proposed encoding framework can be applied on a multimodal scenario, we consider the following illustrative example. In the healthcare domain, 3 clinics are collaboratively working together to diagnose a given patient P_j with a certain pathology H_k . Each clinic is responsible for delivering specific information about the patient P_j . For instance one presents the X-ray image, the second delivers the IRM image and the last one shares the electronic health record (EHR). Our framework allows the 3 institutions to collectively train a model responsible for detecting a pathology H_k and to infer if patient P_j suffers from it, all that without revealing the true format of the data to other parties in a way that clinic 2 and 3 are not able to know the original content of the X-ray image stored in clinic 1.

C. Multi-Objective Paradigm

Our model, as previously mentioned, is trained to solve multiple tasks simultaneously. In addition to considering both the data reconstruction and classification problems in parallel,

two other tasks within the sphere of representation learning are taken into account by the model.

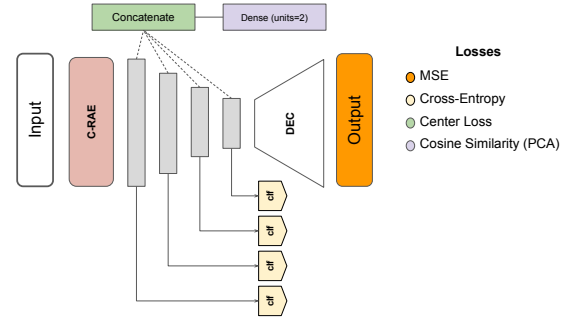
1) *Center Loss*: In order to ensure that the layers' concatenation which will be shared by our framework has a widely class-separated structure to maximize the distance from decision boundaries we introduced a center loss on the concatenation layer. Center loss was presented by Wen et al. (2016) and defined in Eq. 3, where $c_{y_i} \in \mathbb{R}^d$ denotes the y_i -th class center. It aims to minimize the intra-class distance.

$$L_c = \sum_{i=1}^n \|x_i - c_{y_i}\|_2^2 \quad (3)$$

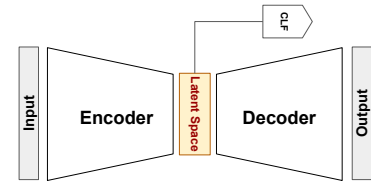
2) *Cosine Similarity with PCA*: Explicitly introducing an interpretation mechanism in our representation is of crucial importance. Since the "black-box" barrier is always present when training neural network models such as autoencoders, we wanted to explicitly make the learning of the representation aligned with the PCA of the original data, as PCA is a well accepted technique for dimensionality reduction. For this aim, we decided to use a cosine similarity loss function. In the same network, we connect the concatenation layer, where the center loss is already applied (Section III-C1), to a dense layer of 2 units and minimize the following function:

$$L_{pca} = 1 - \frac{f_2(\Psi) \cdot x_{pca}}{\|f_2(\Psi)\| \|x_{pca}\|} \quad (4)$$

and $f_2(\cdot)$ defined as $f_2 : \mathbb{R}^{\sum_{i=1}^l e_i} \rightarrow \mathbb{R}^2$ and x_{pca} being the 2-d PCA of the original input x . All parts of our framework are summarized in Fig 4.



(a) Summary architecture of our proposed encoding framework



(b) Baseline architecture proposed by María Quintero-Ossa et al., 2022

Figure 4: Suggested Framework vs Baseline

IV. DATASETS

We will now define the list of datasets we experimented on to evaluate the efficiency of our encoding framework in terms of optimizing the performance-privacy trade-off. We went beyond using only standard benchmark datasets by also including data characterized with real life constraints as data

imbalance and limited number of data points with features exceeding the number of samples.

A. Unimodal Dataset

1) *Image Data*: **MNIST** (LeCun et al., 1998) is considered a benchmark for image classification and includes grey scale images of digits. **Fashion MNIST** (H. Xiao et al., 2017) is a benchmark dataset for machine learning similar to MNIST, but is by nature more complex since the task is to identify 10 types of Zalando articles. **Retinal Optical Coherence Tomography (OCT)** is an imaging technique used to capture high-resolution cross sections of the retinas. The dataset presented by Kermay et al. (2018) is comprised of 84 495 retina images with 4 classes: Normal, Diabetic Macular Edema, Drusen and Choroidal Neovascularization.

2) *Tabular Data*: **Leukemia**: this tabular dataset comes from the Curated Microarray Database (Feldes et al., 2019) which is a repository containing 78 handpicked cancer microarray datasets, extensively curated from 30.000 studies from the Gene Expression Omnibus (GEO), solely for machine learning. For our study, we experimented on the leukemia dataset which contains 281 samples with 22284 gene expression values with the task of classifying 7 types of leukemic cancer.

B. Multimodal Dataset

TCGA Breast Information Core is a multi-omics dataset for breast cancer. Different types of high-throughput sequencing methods used parameters of DNA genome sequence, RNA expression and DNA methylation. Each datatype is labeled with the term “omics” (genomics, transcriptomics and methylomics respectively for our case). This dataset was leveraged by Rappoport and Shamir (2018), Röder et al. (2019) and Cantini et al. (2021) to perform and experiment on multi-omics and multi-view clustering methods. However, since the current version of our representation-oriented encoding framework is specific to supervised learning use cases, we applied it to predict the survival status of patients.

V. EXPERIMENTS & EVALUATION

In this section, we present experiments conducted to compare the performance of training models on the encoded data against models directly trained on the original data or trained only on the latent space embedding (María Quintero-Ossa et al., 2022).

A. Model Parameters

Details about the type, number, and hyperparameters of the layers composing the residual autoencoder in our framework are presented in Appendix A. Hardware specifications are presented in Appendix B.

B. Experimental Setup

For every dataset, we trained the residual autoencoder of our encoding framework on the train set, then obtained the concatenated encoding Ψ^{train} and Ψ^{test} for the train and test sets respectively. We then trained a set of machine

learning models (KNN, SVM, Decision Trees, Random Forests, Multi-Layer Perceptron) in a randomized grid search manner (Appendix C) on the original data x^{train} , the latent space embedding only (baseline) and on our encoded data Ψ^{train} and compared the performance metrics on the respective test points. However, for MNIST, FashionMNIST, and Retinal OCT since they are of type image we trained their original versions using a ResNet-50. From table I we can empirically observe that the average macro F_1 -score of models trained on the encoded data is most of the time better than the other approaches. We further explored the impact of the introduced center loss by comparing the silhouette score of the concatenated encoding Ψ with and without applying center loss. Prior to calculating the silhouette score we first reduce the concatenated encoding to 2 dimensions using t-SNE. As a reminder, silhouette score is a quantitative metric which measures how good clusters are grouped together.

Table II summarizes the silhouette score comparison between two versions of our framework (with/without center loss). We observe an increase of the silhouette score by 478%, 43%, and 57% for FashionMNIST, Leukemia, and TCGA, respectively after adding the center loss and a small decrease of 3% and 0.5% for MNIST and OCT. We clearly notice the positive impact of the center loss in making the encoded representation well grouped.

We also applied an ablation study to check the impact of the introduced cosine similarity loss (Eq. 4). Even though the main reason of introducing this loss was to properly guide the alignment of the input and the shared representation, we also evaluated its impact on the overall classification performance. For this case, we only took the best type of ML model per dataset while using the encoded data in Table I. The results of this ablation study are presented in Table III.

VI. THREAT ANALYSIS

In our solution, we consider four different actors: data owner(s), cloud server, users having access to inference services and institutions where encoder $E(\cdot)$ and final model $\mathcal{M}_\Psi$ are shared. We assume that the cloud server, users, and institutions are honest but curious, meaning they are expected to follow the protocol. The main goal of the adversary corrupting these actors is to attempt to infer sensitive information about the original training samples by using their observations of the protocol execution.

The view of the adversary $\mathcal{A}$ corrupting the cloud consists of the encoded samples of the data owner(s) and the trained model on these samples. Since the adversary $\mathcal{A}$ has no knowledge about the utilized encoder by the data owner(s), it cannot return back to the original samples from the encoded samples. For instance, the adversary $\mathcal{A}$ knows neither the dimensionality nor the type of the original data. In addition to the protection of the privacy of the samples, our framework preserves the privacy of the model implicitly as well. Although the adversary $\mathcal{A}$ has access to the trained classifier model in plaintext, it has no use unless the adversary has access to the encoder. Therefore, we can conclude that our proposed framework securely allows the outsourcing of the computation to a third party or enables

Table I: Macro F1-Score performance with Randomized Grid Search training in a 10-fold stratified cross-validation between using the original data (Org) vs Data shared as the latent space only, being the baseline, (BASE, María Quintero-Ossa et al., 2022) vs the encoded data outputted by our suggested ppML framework (Enc)

	Leukemia			TCGA		
	Org	Base	Enc	Org	Base	Enc
KNN	61.84 $\pm$ 5.18	77.71 $\pm$ 7.57	82.57 $\pm$ 5.01	60.11 $\pm$ 5.65	59.10 $\pm$ 4.56	60.77 $\pm$ 5.97
SVM	83.79 $\pm$ 5.66	78.81 $\pm$ 5.17	80.52 $\pm$ 5.85	33.96 $\pm$ 3.20	59.62 $\pm$ 5.12	62.62 $\pm$ 3.61
DT	66.24 $\pm$ 3.11	70.11 $\pm$ 4.35	74.97 $\pm$ 6.81	56.97 $\pm$ 5.42	59.73 $\pm$ 5.23	57.95 $\pm$ 5.15
RF	78.70 $\pm$ 2.80	78.59 $\pm$ 5.91	83.69 $\pm$ 4.30	61.96 $\pm$ 6.47	59.64 $\pm$ 4.38	63.96 $\pm$ 4.16
MLP	82.06 $\pm$ 4.34	80.36 $\pm$ 5.51	83.96 $\pm$ 3.17	59.74 $\pm$ 4.64	60.27 $\pm$ 5.96	62.79 $\pm$ 3.33

Dataset	KNN-BASE	KNN-ENC	SVM-BASE	SVM-ENC	DT-BASE	DT-ENC	RF-BASE	RF-ENC	MLP-BASE	MLP-ENC	ResNet50-ORG
MNIST	99.44 $\pm$ 0.05	99.40 $\pm$ 0.02	99.47 $\pm$ 0.06	99.44 $\pm$ 0.02	99.16 $\pm$ 0.10	99.29 $\pm$ 0.03	99.44 $\pm$ 0.06	99.43 $\pm$ 0.03	99.44 $\pm$ 0.05	99.41 $\pm$ 0.06	99.31 $\pm$ 0.08
Fashion	92.34 $\pm$ 0.17	92.29 $\pm$ 0.22	92.25 $\pm$ 0.16	92.58 $\pm$ 0.22	90.95 $\pm$ 0.43	91.67 $\pm$ 0.41	92.34 $\pm$ 0.16	92.44 $\pm$ 0.21	92.40 $\pm$ 0.16	92.13 $\pm$ 0.18	91.21 $\pm$ 0.33
OCT	98.33 $\pm$ 1.45	98.43 $\pm$ 0.26	97.68 $\pm$ 1.91	98.56 $\pm$ 0.22	97.68 $\pm$ 1.91	98.33 $\pm$ 0.28	99.02 $\pm$ 0.45	99.19 $\pm$ 0.19	98.20 $\pm$ 1.42	98.58 $\pm$ 0.14	97.01 $\pm$ 1.28

Table II: Comparison between the silhouette score applied on the final concatenated encoding of the test set Ψ^{test} for 10-fold stratified cross-validation to evaluate the effect of adding center loss

L_c	MNIST	Fashion	Leukemia	OCT	TCGA
Without	0.597	0.078	0.095	0.734	0.048
With	0.594	0.451	0.136	0.712	0.076

Table III: For each dataset, we took the best model found in Table I based on our encoding framework and evaluated the effect of the PCA cosine similarity loss.

L_{pca}	MNIST _{SVM}	Fashion _{SVM}	Leukemia _{MLP}	OCT _{RF}	TCGA _{RF}
Without	99.46 $\pm$ 0.03	92.15 $\pm$ 0.34	82.23 $\pm$ 6.54	99.11 $\pm$ 0.21	63.50 $\pm$ 2.84
With	99.44 $\pm$ 0.02	92.58 $\pm$ 0.22	83.96 $\pm$ 3.17	99.19 $\pm$ 0.19	63.96 $\pm$ 4.16

the third party to benefit from the output of the model without compromising the privacy of the data or the model.

The adversary $\mathcal{A}$ corrupting at least one user can perform predictions on the model $\mathcal{M}_\Psi$ for the encodings of given data. Through using the prediction service as an API, $\mathcal{A}$ has access only to the final predictions y_{new} . Without knowledge of the encodings of the training samples and the encoder $E(\cdot)$, it is not possible to extract the training samples. $\mathcal{A}$ might still perform membership inference attacks which is a common problem for all machine learning models trained without differential privacy. Our solution does not aim to address these weaknesses. The adversary, $\mathcal{A}$, who has compromised at least one institution, can access both the encoder $E(\cdot)$ and the model $\mathcal{M}_\Psi$. $\mathcal{A}$ can use $E(\cdot)$ to encode new data, and then use the encodings to make predictions using $\mathcal{M}_\Psi$, resulting in (Ψ, y_Ψ) . Without the encodings of the original training samples, it is impossible to extract the training data. $\mathcal{A}$ can also perform attacks like model inversion and membership inference. However, as we mentioned, we do not present a solution to counteract these

types of attacks in this article.

The adversary $\mathcal{A}$, who has compromised at least one institution and the cloud server, can access the encoder $E(\cdot)$ and the encodings of the original training samples. In this scenario, $\mathcal{A}$ can train the decoder $D_{inv}(\cdot)$ by using the data of compromised institution and the encoder $E(\cdot)$ to reconstruct the original training samples. Therefore, in our security model, we assume that the cloud server and institutions do not collude. This assumption is practical in real-world scenarios. Data owners can conceal their identities from the cloud server, making it difficult to determine with whom the model trained by the cloud server is shared. The 4 scenarios described in our threat analysis are illustrated in Fig 5.

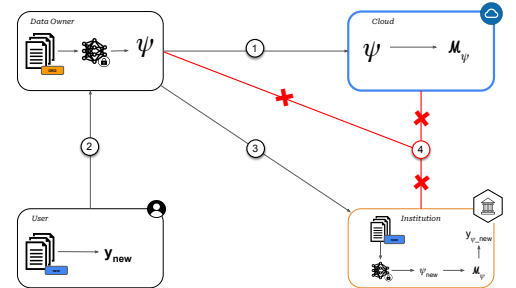


Figure 5: This figure summarizes the threat analysis schema with the four actors taking part in it. In ① the adversary corrupts the cloud, in ② the adversary corrupts a user, in ③ the adversary will have an access to one of the institutions and in ④ the adversary will corrupt both the cloud and one of the institutions. It is only in the latter scenario where the risk of inverting the original data might occur since the adversary will have access to the encoder $E(\cdot)$ and the encoding of the original training samples Ψ thus a decoder $D_{inv}(\cdot)$ can be trained on the pair of the institute data to perform an inversion attack on Ψ .

VII. LIMITATIONS OF THE PROPOSED FRAMEWORK

Even though our encoding framework showed great performance-privacy results, it is for now still constrained to specific scenarios. Our framework is task-oriented which means that it requires the presence of a supervision task that guides our autoencoder for presenting a meaningful mapping. However, it is known within the ML community that most datasets do lack annotation (Humbert-Droz et al., 2022; Nguyen & Le, 2021; Xu et al., 2020) which currently excludes our proposed ppML encoding technique to be leveraged for unsupervised use cases. In addition, our model is not generic towards all types of data training distributions, in fact, our current framework is applicable only if the data is vertically split among input parties where they share the same sample ID but different feature spaces, like in the multimodal example used in our experiments, which excludes (for now) horizontal federated learning (Yang et al., 2019).

VIII. CONCLUSION & FUTURE WORKS

In this study, we introduced an encoding strategy powered by representation learning leveraged for privacy purposes. The main goal and motivation behind implementing such a framework are to take advantage of discriminative representations learned in the hidden layers in the encoder part and take their concatenation as the encoding to be shared with other parties. To achieve this, we implemented a supervised residual autoencoder trained to consider both data reconstruction and an assigned classification task. To ensure a good representation we strengthened the training with two introduced losses one being the center loss applied on the concatenated encoding and a cosine similarity loss used to force the concatenation in having the same direction as the original input. We further presented the application workflow of our framework in a unimodal and multimodal settings. Our framework allows us to benefit from external computational resources to perform training on the encoded data in a secure fashion since no information about the data is being revealed. As future work, we look forward to expanding the domain application of our encoding framework by including unlabeled datasets and horizontally distributed data for federated learning.

REFERENCES

- Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., & Zhang, L. (2016). Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 308–318.
- Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798–1828.
- Cantini, L., Zakeri, P., Hernandez, C., Naldi, A., Thieffry, D., Remy, E., & Baudot, A. (2021). Benchmarking joint multi-omics dimensionality reduction approaches for the study of cancer. *Nature communications*, 12(1), 1–12.
- Cao, T., Bie, A., Vahdat, A., Fidler, S., & Kreis, K. (2021). Don’t generate me: Training differentially private generative models with sinkhorn divergence. *Advances in Neural Information Processing Systems*, 34, 12480–12492.
- Chen, J., Wang, W. H., & Shi, X. (2020). Differential privacy protection against membership inference attack on machine learning for genomic data. *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*, 26–37.
- Dong, T., Zhao, B., & Lyu, L. (2022). Privacy for free: How does dataset condensation help privacy? *arXiv preprint arXiv:2206.00240*.
- Feltes, B. C., Chandelier, E. B., Grisci, B. I., & Dorn, M. (2019). Cumida: An extensively curated microarray database for benchmarking and testing of machine learning approaches in cancer research [PMID: 30789283]. *Journal of Computational Biology*, 26(4), 376–386. <https://doi.org/10.1089/cmb.2018.0238>
- Ferreira, D., Silva, S., Abelha, A., & Machado, J. (2020). Recommendation system using autoencoders. *Applied Sciences*, 10(16), 5510.
- Fredrikson, M., Jha, S., & Ristenpart, T. (2015). Model inversion attacks that exploit confidence information and basic countermeasures. *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 1322–1333.
- Harder, F., Adamczewski, K., & Park, M. (2021). Dp-merf: Differentially private mean embeddings with random-features for practical privacy-preserving data generation. *International conference on artificial intelligence and statistics*, 1819–1827.
- Hesamifard, E., Takabi, H., & Ghasemi, M. (2017). Cryptodl: Deep neural networks over encrypted data. *arXiv preprint arXiv:1711.05189*.
- Hijmans, H., & Raab, C. D. (2018). Ethical dimensions of the gdpr. *Commentary on the General Data Protection Regulation*, Cheltenham: Edward Elgar (2018, Forthcoming).
- Humbert-Droz, M., Mukherjee, P., Gevaert, O., et al. (2022). Strategies to address the lack of labeled data for supervised machine learning training with electronic health records: Case study for the extraction of symptoms from clinical notes. *JMIR medical informatics*, 10(3), e32903.
- Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., et al. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5), 1122–1131.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Le, L., Patterson, A., & White, M. (2018). Supervised autoencoders: Improving generalization performance with unsupervised regularizers. *Advances in neural information processing systems*, 31.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

- Lee, J.-W., Kang, H., Lee, Y., Choi, W., Eom, J., Deryabin, M., Lee, E., Lee, J., Yoo, D., Kim, Y.-S., et al. (2022). Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10, 30039–30054.
- Lee, J., Lee, E., Lee, J.-W., Kim, Y., Kim, Y.-S., & No, J.-S. (2021). Precise approximation of convolutional neural networks for homomorphically encrypted data. *arXiv preprint arXiv:2105.10879*.
- Mandal, B., Amariuca, G., & Wei, S. (2022). Uncertainty-autoencoder-based privacy and utility preserving data type conscious transformation. *arXiv preprint arXiv:2205.01950*.
- María Quintero-Ossa, A., Solano, J., Jarcía, H., Zarruk, D., Correa Bahnsen, A., & Valencia, C. (2022). Privacy-preserving machine learning for collaborative data sharing via auto-encoder latent space embeddings. *arXiv e-prints*, arXiv–2211.
- Nguyen, N.-Q., & Le, T.-S. (2021). A semi-supervised learning method to remedy the lack of labeled data. *2021 15th International Conference on Advanced Computing and Applications (ACOMP)*, 78–84.
- Pan, Y., He, F., & Yu, H. (2020). Learning social representations with deep autoencoder for recommender system. *World Wide Web*, 23(4), 2259–2279.
- Phan, N., Wang, Y., Wu, X., & Dou, D. (2016). Differential privacy preservation for deep auto-encoders: An application of human behavior prediction. *Thirtieth AAAI Conference on Artificial Intelligence*.
- Rangamani, A., Mukherjee, A., Basu, A., Arora, A., Ganapathi, T., Chin, S., & Tran, T. D. (2018). Sparse coding and autoencoders. *2018 IEEE International Symposium on Information Theory (ISIT)*, 36–40.
- Rappoport, N., & Shamir, R. (2018). Multi-omic and multi-view clustering algorithms: Review and cancer benchmark. *Nucleic acids research*, 46(20), 10546–10562.
- Rochel, J. (2021). Ethics in the gdpr: A blueprint for applied legal theory. *International Data Privacy Law*, 11(2), 209–223.
- Röder, B., Kersten, N., Herr, M., Speicher, N. K., & Pfeifer, N. (2019). Web-rmkl: A web server for dimensionality reduction and sample clustering of multi-view data based on unsupervised multiple kernel learning. *Nucleic acids research*, 47(W1), W605–W609.
- Sakurada, M., & Yairi, T. (2014). Anomaly detection using autoencoders with nonlinear dimensionality reduction. *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, 4–11.
- Shokri, R., Stronati, M., Song, C., & Shmatikov, V. (2017). Membership inference attacks against machine learning models. *2017 IEEE symposium on security and privacy (SP)*, 3–18.
- Truex, S., Baracaldo, N., Anwar, A., Steinke, T., Ludwig, H., Zhang, R., & Zhou, Y. (2019). A hybrid approach to privacy-preserving federated learning. *Proceedings of the 12th ACM workshop on artificial intelligence and security*, 1–11.
- Wang, K.-C., Fu, Y., Li, K., Khisti, A., Zemel, R., & Makhzani, A. (2021). Variational model inversion attacks. *Advances in Neural Information Processing Systems*, 34, 9706–9719.
- Wang, Y., Ding, Z., Xiao, Y., Kifer, D., & Zhang, D. (2021). Dpgen: Automated program synthesis for differential privacy. *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 393–411.
- Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. *European conference on computer vision*, 499–515.
- Wickramasinghe, C. S., Marino, D. L., & Manic, M. (2021). Resnet autoencoders for unsupervised feature learning from high-dimensional data: Deep models resistant to performance degradation. *IEEE Access*, 9, 40511–40520.
- Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arxiv 2017. arXiv preprint arXiv:1708.07747*.
- Xiao, T., Tsai, Y.-H., Sohn, K., Chandraker, M., & Yang, M.-H. (2020). Adversarial learning of privacy-preserving and task-oriented representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07), 12434–12441.
- Xu, J., Zuo, W., Liang, S., & Zuo, X. (2020). A review of dataset and labeling methods for causality extraction. *Proceedings of the 28th International Conference on Computational Linguistics*, 1519–1531.
- Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), 1–19.
- Ye, D., Zhu, T., Zhou, S., Liu, B., & Zhou, W. (2022). Label-only model inversion attack: The attack that requires the least information. *arXiv preprint arXiv:2203.06555*.
- Zhao, B., Mopuri, K. R., & Bilen, H. (2021). Dataset condensation with gradient matching. *ICLR*, 1(2), 3.

APPENDIX A MODEL DETAILS

We present here more details about the type, number and hyperparameters of the layers composing the residual autoencoder in our framework. As previously mentioned in Section 3.1, two versions of supervised residual autoencoders were used in our experiments. If the modality is of type image then convolution layers are included and we name it as C-SRAE. Otherwise we only include dense layers under the name SRAE. The latter is applied for the leukemia and TCGA datasets both being in a tabular format. C-SRAE is also presented in two different formats, a light version for small image input data as MNIST and FashionMNIST, and another for bigger image sizes as OCT with more convolution layers added. Fig 6 presents more details about the three architectural types. However, for more uniformity the final encoding Ψ has the same shape $(N, 480)$ across the three presented versions. We also applied $L1/L2$ kernel regularization on every layer for better generalizability and $L1$ activity regularization on the dense layers taking part in the final concatenated encoding Ψ to ensure a proper invariant representation. We used *Adam* as the optimizer and we monitored the training with an *Early Stopping* callback on the validation loss.

APPENDIX B SYSTEM COMPONENTS

Here we provide details about the hardware parts constituting our system.

Table IV: Hardware Specifications

Parameter	Technical Specifications
Memory	32 GiB
Processor	11th Gen Intel® Core™ i7-11800H
No CPU Cores	16
CPU Frequency	2.30 GHz
GPU	Nvidia RTX A2000 Mobile

APPENDIX C TRAINING DETAILS

Here we provide more details about the set of hyperparameters related to the machine learning models used in our experiments and that were explored through a randomized grid search. The models and grid search were implemented using Sklearn library.

Table V: Hyperparameters for Randomized Grid Search

ML Model	Hyperparameters
KNN	{"n_neighbors": [i for i in range(2,10)]}
SVC	{"C": [10 ⁱ for i in range(-4, 2)]}
Decision Tree	{"max_depth": [2, 6], "max_depth":[10, 50, 100, 200, 300], "min_samples_leaf": [1,3,5,10], "criterion": ["gini", "entropy"], "class_weight":["balanced"]}
Random Forest	{"n_estimators":[100,200,300], "max_depth":[10, 50, 100, 200, 300] , "min_samples_leaf": [1,3,4,5,6,7], "criterion": ["gini", "entropy"], "class_weight":["balanced"]}
Multilayer Perceptron	{"hidden_layer_sizes": [(10,30,10), (20,), (100,)], "activation": ["tanh", "relu"], "solver": ["sgd", "adam"], "alpha": [0.0001, 0.05], "learning_rate": ["constant","adaptive"] }

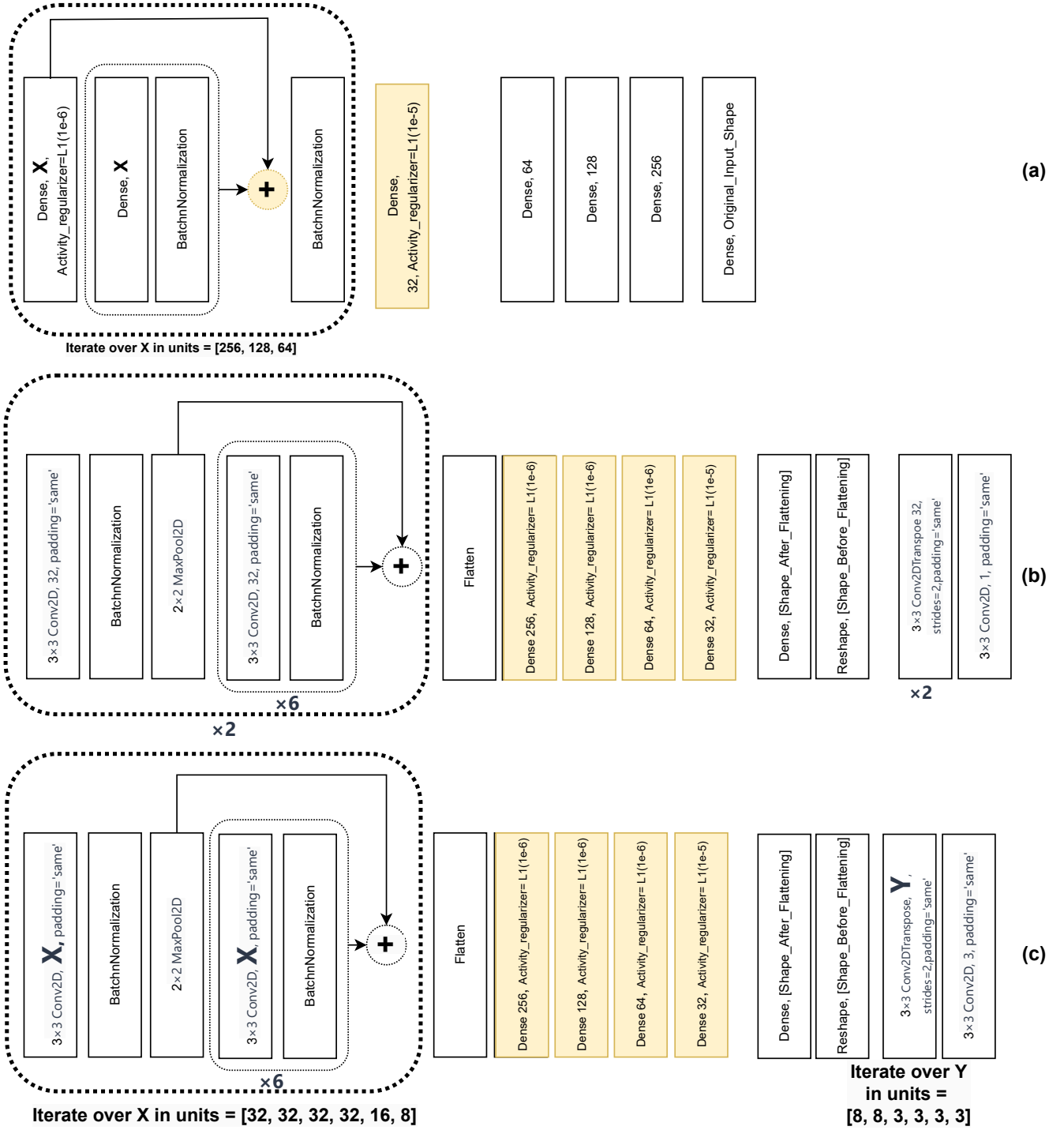


Figure 6: **(a)**, **(b)**, **(c)** are the model's backbone architecture for (Leukemia + TCGA), (MNIST+FashionMNIST) and (OCT) respectively. The layers colored with orange are the dense layers of the encoder part that are connected to a feed-forward neural classifier $f_i(\cdot)$. As mentioned in Section 2.2, for better mapping invariance we applied $L1/L2$ kernel regularizer on all layers and $L1$ activity regularizer on the layer participating in the concatenated encoding (orange-colored). The final encoding Ψ however has the same shape $(N, 480)$ in the three versions.