TEC | Tecnológico de Costa Rica

Maestría en Computación, énfasis en Ciencias de la Computación

Escuela de Ingeniería en Computación

Data Quality Metrics for Unlabelled Datasets in Medical Imaging

Ana Catalina Díaz Villaplana

Supervisor:

Saúl Calderon-Ramirez

San José, CRC

July 2024

Contents

List of Tables

Acronyms

CAD Computer Aided Diagnosis. 4, 5

DeDiM Deep Dataset Dissimilarity Measure. iii, 40

DeDiMs Deep data set Dissimilarity Measures. v, 20, 25–27, 29, 31, 55, 60, 63

RT-PCR Real-time Reverse Transcription Polymerase Chain Reaction. 5

SLM Supervised Learning Model. 8–10, 15

SSDL Semi-supervised Deep Learning. 6, 24, 29, 31

SSLM Semi-supervised Learning Model. 3–6, 10, 14, 19, 59–61

USLM Unsupervised Learning Model. 9, 10

Abstract

Deep learning models typically require large, labeled datasets for optimal performance. However, in real-world applications such as medical imaging, labeled data can be scarce. Semi-supervised deep learning addresses this challenge by leveraging both labeled and unlabeled data to enhance model accuracy. Most semi-supervised methods assume similar distributions between labeled and unlabeled datasets, an assumption that may not hold in practice. To ensure data quality and consistency, we introduce Mahalanobis-based and Frobenius-based distance measures in the embedding space of the deep learning model to evaluate the similarity between labeled and unlabeled datasets. Our findings reveal that the Mahalanobis-based distance correlates strongly with the accuracy of the popular semi-supervised method MixMatch, whereas Frobenius distance results show inconsistent behavior. Moreover, the proposed approach is significantly more efficient than existing methods in the field.

Escuela de Ingeniería en Computación
Unidad de Posgrado

TEC | Tecnológico de Costa Rica

ACTA DE APROBACION DE TESIS

Data Quality Metrics for Unlabelled Datasets in Medical Imaging

Por: DIAZ VILLAPLANA ANA CATALINA

TRIBUNAL EXAMINADOR

Saúl Calderon R

Dr. Saúl Calderón Ramírez
Profesor Asesor

Dr. Luis Alexander Calvo Valverde
Profesor Lector

MSc. Johan Guilllén Meza
Lector Externo

Dra.-Ing. Lilliana Sancho Chavarría
Presidente, Tribunal Evaluador Tesis
Programa Maestría en Computación

SINAES POST-GRADO ACREDITADO COSTA RICA

14 de octubre, 2024

1. Introduction

1.1 Background

The use of machine learning models, especially deep learning, has proven to be valuable in various fields, with healthcare being a particularly interesting area [3]. Computer vision, in particular, has been effective in analyzing medical images and aiding in the diagnosis of major illnesses such as cancer. For example, in a study [4], researchers used mammography images to identify suspicious lesions, and the findings of the model contributed to more accurate diagnoses [4].

The use of deep learning models is relatively recent. However, these models have faced criticism due to the poor quality of data used to train them. For example, in [3], the authors highlight the negative impact of using non-representative datasets in healthcare systems.

In order to make accurate predictions, machine learning models, especially deep learning models, must learn representations from the data provided during training [5]. Therefore, it is critical to emphasize the importance of data quality. Additionally, deep learning models need significant amounts of labeled data [6] and operate under the assumption of Independent and Identically Distributed (IID) data, which may not always be applicable to real-world datasets [7].

In the field of medicine, large datasets may not always be available or labeled as required. Labeling a dataset requires specialized expertise, which can lead to increased costs and make it an expensive task [8]. Because of this, Semi-supervised Learning Model (SSLM) are a solution that helps with this problem. Deep learning models can learn from both labeled and unlabeled data [8].

Using small datasets in deep learning models can be challenging. This can lead to overfitting, resulting in inaccurate predictions. For example, in a dataset with class imbalance, the model may generate false positives, reducing confidence in the predicted results. One way to address this is by leveraging unla-

beled data to train deep learning models. [9]

Recently, there have been various approaches to SSLM that have shown promising results, as demonstrated by the work in [10] and [11]. In the context of Computer Aided Diagnosis (CAD) systems there are multiple unlabeled data sources available. Selecting one or a combination of these sources is a critical decision. Research in [12] and [13] indicates that the use of unlabeled data from diverse sources can significantly affect the performance of SSLM.

Investigations in data quality acknowledged the importance of different quality dimensions in the data like completeness, validity, consistency, currency, interpretability, and relevance [14], among others. All these concepts have been important to measure data quality in datasets of structured data. These concepts have been extrapolated to machine learning models and deep learning. For example, the author in [15] mentioned various concepts related to data quality metrics that have been extended to deep learning and machine learning models.

The same problems found in exploring structured data are also extrapolated to machine learning models, for example, if the dataset has outlier data, it can cause instability in learning, which results in inaccurate predictions [16]. Deep learning models have a better ability to handle raw data than normal systems but are more sensitive to changes in data, hence, it is important to investigate and generate more techniques or data metrics to obtain better results and better unlabeled datasets.

In [17], the author discusses the use of a suitable selection metric for sampling unlabeled data as a main issue in SSLM learning. The author in [7] argues that general-purpose datasets used in investigations may contain built-in bias, which can also be present in datasets created from scratch. The authors emphasize the importance of being able to measure and quantify the quality of unlabeled or labeled datasets.

1.2 COVID-19 detection using Chest X-ray images

Since COVID-19 was discovered, the world has been experimenting how to live in an pandemic [18], where thousands of lives have been lost.

The emergence of fast-spreading variants created new phases and increased the panic during the COVID-19 pandemic [19]. Because of this, research on faster and more cost-effective methods for detecting infected patients remains relevant. [2] noted that Real-time Reverse Transcription Polymerase Chain Reaction (RT-PCR) tests can be expensive and time-consuming. They proposed an alternative approach using medical imaging, specifically X-ray images, to develop deep learning SSLM CAD. This method could serve as a viable solution during pandemic emergencies [2].

The implementation of the deep learning CAD approach requires a large number of labeled datasets, which are scarce in a pandemic scenario [2], and the proposed solution is SSLM. The model can be enhanced by combining labeled and unlabeled datasets [2].

It is important to note that the use of unlabeled datasets in real-world applications comes with a potential risk. This risk arises from not knowing the labels or the distribution, which can lead to inaccurate predictions. Given this risk, it is crucial to prioritize data quality and to accurately measure the impact of noise on the model.

1.3 Problem definition

Unlabeled data sets are large collections of data gathered in various ways. The observations in the dataset may not encompass all possible scenarios or possess a similar distribution. This might affect the outcomes of the deep learning models. Hence, it is crucial to have a method for evaluating the quality of unlabeled datasets.

Different techniques can be used to extract useful information from unlabeled datasets [20], and semantic heuristics are often used to compare labeled and unlabeled datasets but few investigations proposed a numerical value as a metric [1]. As a result, some questions emerge: How can we efficiently measure or estimate the suitability of an unlabeled and labeled datasets for SSLM? How can we accurately assess the qualities of an unlabeled dataset?.

The above questions will drive the work in this thesis. In this work, we strive for one or more efficient metrics to estimate how good an unlabeled dataset is for SSLM. This means that a high correlation of the proposed metrics with the Semi-supervised Deep Learning (SSDL) performance is expected.

1.4 Objectives

Main Objective:

Implement one or more metrics to measure the quality of a dataset for SSLM specifically for COVID-19 detection using chest X-ray images.

Specific objectives:

1. Propose and implement at least one data quality metric for an unlabeled dataset relative to a labeled dataset.
2. Measure the processing time of the proposed metric to evaluate its efficiency compared to other methods in different research studies.
3. Measure the correlation coefficient of the metric to evaluate the metric results against the accuracy of the semi-supervised model.
4. Evaluate the proposed metric in the context of a real-world application: COVID-19 detection.

2. Literature Study

2.1 Conceptual Framework

A learning model in machine learning can be defined as knowledge acquisition, it learns new symbolic information or representations and can apply it in an effective manner. [21].

The author in [21] defined a learning model when a computer program is said to learn with the tasks in T, measured by P, and improve with the experience E.

Definition 1. *A learning model processes input to create an output, mathematically, it is a function of [22],*

$$L : \mathbf{X} \rightarrow \mathbf{Y}^{'} \tag{2.1}$$

A task, denoted as T, refers to how an algorithm processes a task or a sample. The measure, denoted as P, is a quantitative metric used to evaluate the performance of the algorithm. For instance, accuracy, which will be explained later in the document, can be used as a measure. E is defined as the experience that the model has access to during the learning process [21]. In the context of a machine learning algorithm, the experience corresponds to a dataset, denoted as D.

A dataset D is defined as a collection of data points or observations D, where $D = \{\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ..., \mathbf{x_n}\}$, and an observation $\mathbf{x}$ is a vector of features $\mathbf{s} \in \mathbb{R}^n$, where every s_i is a list of labels y.

Definition 2. *D is defined as $((x_i, y_i))^l$*

Given the previous definitions, a learning model L can be defined as the result of a learning model when processing a data set D" [23].

Definition 3. *Learning Model $y = L_{\boldsymbol{\theta}}(\mathbf{x})$*

Definition 4. *The algorithm can process the experience E or dataset D in a supervised or unsupervised manner [21].*

A Supervised Learning Model (SLM) learns to link some input with an output, given a dataset D of inputs x and outputs y [21], it can infer a function to determine the label y' on data point x' not seen before [24].

The outputs are difficult to collect and must be labeled by a supervisor [21]. Normally this type of algorithms are provided with two different datasets, a training set, and a test set, each will be used in a different process of the model.

A training set consist of a collection of data points D_{t}, that the learning algorithm uses to learn the relations between the input x_i and an output y_i. For a SLM consisting of a dataset D with labeled data D_l, according to [24] it is a collection of D_l labeled data points $D_l = ((x_i, y_i))^l{}_{i=1}$. Each pair of data points (x_i, y_i) is integrated of an object $x_i \in D_l$ from an input space S, which has an associated label y_i. [24].

A test set, denoted as D_{ts}, is a collection of data points D that the learning algorithm uses during testing.

According to [24], D_u is a collection of data points where there is no explicit output y_i for each input x_i, and this output is inferred by the learning model.

Selecting the most relevant and informative features of the vector $\mathbf{s}$ is a feature extractor and will help speed up the learning process, improve performance, and understand the data that the model will use [25]. A feature extractor is defined as a technique to find the best features in the vector x_i and generate a new $x_i{}'$.

In the machine learning community, there has been a discussion about the implementation of feature engineering in training datasets. The feature extractor $f(x_i)$ holds significant importance in this context. Feature engineering utilizes a feature extractor to discover transformations of the input data in order to enhance the model's performance or to increase computational efficiency [24].

The SLM approach has the disadvantage that if the collected data do not cover the entire data space and are limited in quantity, the generalization performance may suffer [8].

An alternative to SLM, is Unsupervised Learning Model (USLM), the authors in [21] defined it as the ability to process D_u, where only unlabeled data is considered in a training set, and will *try to infer some underlying structure from the inputs* [24]. This type of algorithm has a "learner" but does not use label information, because of that it must learn to infer the y' from the data that is provided. The USLM will get the "best" representation that obtains the most approximate information from $\mathbf{x}$ and must be the simpler representation of all possibilities [21].

Definition 5. *USLM is defined as the ability to process D_u, where only unlabeled data is considered in a training set, and will try to infer some underlying structure from the inputs [24]*

The learning models can be categorized as nonparametric or parametric, as the name described it, a parametric model will have parameters θ, $L(\mathbf{y}|\mathbf{x};\theta)$, more specifically when a model learns a function described by a parameter vector whose size is finite and fixed before any data are observed. Non-parametric models do not have such limitation [21].

As was mentioned before a learning model L, is a function $L(x)$, with this, every learning algorithm must include a task of minimizing, resulting in an optimization problem [21].

The function that the model has to minimize or maximize is *called the objective function or criterion*, but when it tries to minimize, it can be called the cost function, loss function or error function $\mathcal{L}(x)$. [21]

Definition 6. $x^* = argmin\ L(\boldsymbol{x})$. *[21]*

The function $L(x)$ can be optimized using the gradient descent technique. This technique uses the derivative to minimize the function, *because it tells us how to change x in order to make a small improvement in y* [21].

If the function $f(x- \in sing\ (f^{'}(x)))$ is less than $f(x)$, then, $f(x)$ can be reduced by moving x with the opposite sign of the derivative [21], the figure 2.1 illustrates an example.

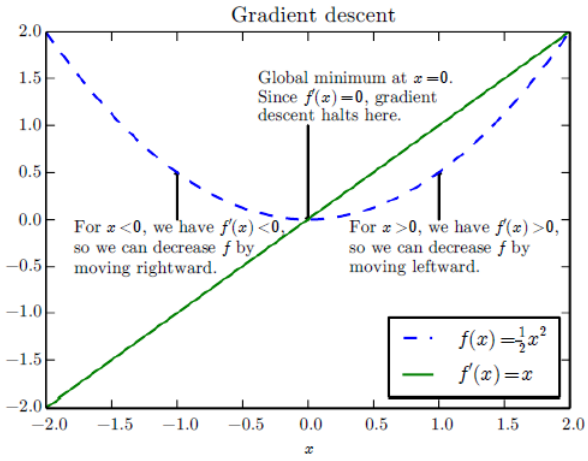

Figure 2.1: Gradient Descent technique representation on how derivatives of a function can be used to follow the function downhill to a minimum [21]

A SSLM is a branch that combines both SLM and USLM, labeled and unlabeled data, during training. D_l and D_u are used to learn the representations and predict a result. This approach can be used in domains where labeled data is scarce, but unlabeled data is available and sufficient, helping to construct a better learning model [24].

Definition 7. *SSLM is defined as a branch that combines both SLM and USLM, labeled and unlabeled data, during training.*

It's important to note that a good learning model should be able to make predictions based on what it has learned from the training set. When the model predicts the correct output, it shows that it can generalize, or in other words, perform well on previously unseen inputs [21].

In order to assess how well the model generalizes, It can be used the test error E_{test}, which represents the expected error on new input data [21]. The test error measures the model's performance on a set of test examples that is separate from the training data. The test error is closely linked to the training error E_{train}, which is the error computed and minimized during the model training process [21].

When looking at E_{test} and E_{train}, a model can either overfit or underfit a given dataset. Overfitting occurs when there is a large gap between the training error and test error [21], while underfitting occurs when the model is unable to achieve a sufficiently low error value on the training set [21].

Definition 8. *Regularization is defined as any modification made to a learning algorithm to reduce its generalization error without reducing its training error [21].*

According to the **no free lunch** theorem, there is no single best learning algorithm, and therefore, there is no best regularization method. The most suitable regularization method depends on the specific model [21].

A common method of regularization involves adding a penalty, known as a regularizer $\Omega(\mathbf{w})$, to the cost function of a model that learns $L(\mathbf{x};\theta)$ [21].

Data augmentation is another type of regularization that is useful for reducing both the test error (E_{test}) and the training error (E_{train}). This technique is considered one of the best methods for preventing overfitting. By creating augmented data, a more comprehensive set of possible data points is represented, which minimizes the distance between the training and validation sets [26].

Definition 9. *Data augmentation can be defined as methods for constructing iterative optimization or sampling algorithms introducing unobserved data or latent variables [27],*

Based on the definitions provided above, there are various types of learning models, and deep learning is one of them. Deep learning is founded on neural networks, which are a type of parametric model. In the study by [22], a neural network is defined as a group of interconnected nodes or neurons, where each connection between neurons is characterized by a numerical value known as a weight θ [28].

The perceptron is the earliest and simplest neural network. It was first proposed by McCulloch and Pitts in [29], but was implemented and described by Fran Rosenblatt in [30]. The author based the design of the perceptron on the "connectionist" empiricist theories and the research of Rashevsky, McCulloch and Pitts, Culbertson, Kleene, and Minsky.

The structure of the Rosenblatt perceptron is *Sensory units (S-points), association cells (A-units), projection area (A_I) and association area (A_{II})* [30], as illustrated in 2.2.

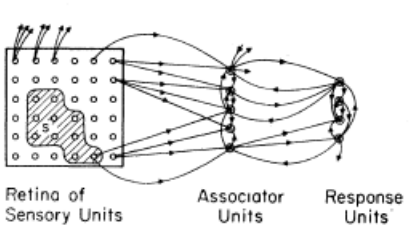

Figure 2.2: Rosenblatt representation of the Perceptron [31]

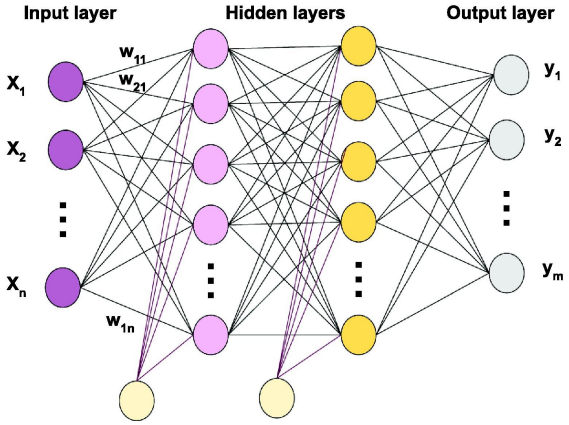

Figure 2.3: Multilayer perceptron representation [33]

The perceptron can be described mathematically as input **x** and resulting in an output y, as follows,

$$y = g(\sum_{j=1}^{d}(w_j * x_j) + b), \tag{2.2}$$

g is the activation function that allows the neural network to activate or deactivate a neuron using a non-linear transformation function. w_j and b are the parameters that the model should learn from D_{tr} [22].

A perceptron can be connected to other perceptrons; they can be fully connected and have three types of layers: the input layer, output layer, and hidden layer, generating what is called a multilayer perceptron network [32], as it is described in the figure 2.3. The multilayer perceptron was achieved thanks to the research and implementation of the mathematic technique backpropagation.

A specific type of neural network is the convolutional network. The name comes from the use of the convolution method, which is a specialized linear method as defined in [21].

$$s(t) = (\mathbf{x}w)(t), \tag{2.3}$$

where $\mathbf{x}$ represents the input of the model, w is the kernel, which should be a valid probability density, typically defined as a hyperparameter in the network, and the output $s(t)$ is referred to as the feature map.

The Cambridge dictionary defines "metric" as *a system for measuring something*. In this work context, it is necessary to measure the performance of the model L or to assess the quality of a dataset D.

To evaluate a model's performance, the standard metric used is accuracy, which is defined as the proportion of examples for which the model generates the correct output [21].

Definition 10. *Accuracy* $A = \frac{CorrectOutputs}{TotalOutputs}$

The authors in [34] discussed how a learned distance metric can be used in machine learning models such as K-means and nearest-neighbors classifier. They asked the question: *Can we automatically learn a distance metric on* $\mathbb{R}^{\mathbb{N}}$ *that respects these relationships, one that assigns small distances between similar pairs?* [34]. Based on these two questions, the authors demonstrated that the use of distance metrics improved the clustering performance, leading to better performance of the machine learning model.

Distance metrics are valuable for evaluating the results of machine learning models based on this research and the research in [35]. The author in [36] explains that distances have been a key tool in real-world applications and image classification.

To assess the quality of the dataset, this work will propose using two commonly used distances to compare populations: the Mahalanobis and Frobenius distances that are defined below next to some other distances that are commonly used.

The authors in [2] demonstrated that it is faster and computationally more efficient to compare matrices of two populations. By using norms, a scalar can be derived to represent both populations, which can be referred to as a metric.

Mahalanobis distance was proposed by P. C. Mahalanobis in [37], in the search for racial likeness [38], and since it has been a useful tool to *measure of divergence or distance between groups in terms of multiple characteristics.* [38]

The authors in [36] define the Mahalanobis distance as a measure between two data points in a feature space. Following the same definition of the authors, the distance counts for *unequal variances as well as correlations between features*.

In [38], Mahalanobis distance is defined is a metric to calculate the differences between two populations. It is assumed that a vector X has the same variation about its mean within either group, the difference is calculated based on the difference between the vectors X for each group, as represented in the formula below:

Definition 11. *Mahalanobis distance* = $d_M(D_1, D_2) = (D_{1m} - D_2)^T \Sigma^{-1} (D_{1m} - D_2)$

The author indicates that D_{1m} is the mean of the dataset D. The index T is a transpose matrix, Σ denoted the co-variance matrix for each of the population group, D_1 and D_2. Since the covariance matrix Σ is nonsingular, it can be assumed that the values are positive and hence it can be called a metric [38].

The square version of Mahalanobis is used in classification problems, pattern recognition, or discriminant analysis.

Definition 12. *Mahalanobis distance* = $\Delta^2 = (\mu_1 - \mu_2)^T \Sigma^{-1} (\mu_1 - \mu_2)$ *[38]*

The authors in [2] describes that the Mahalanobis distance can generate good results comparing two data sets in a SSLM.

Frobenius distance can be defined as the Frobenius norm. This term is used in the matrix norm documentation [39].

The authors in [40] investigated how the Frobenius distance or Frobenius norm can be used to calculate the distance between covariance structures. The authors concluded that the method can be extended to other co-variance models because it will only need co-variance information. The Frobenius distance is called the Euclidean norm [39] and the formula is as follows:

Definition 13. *Frobenius Distance* = $d_F(D) = \|D_1\| = \sqrt{trace(D_1^t D_1)}$

In a recent study [41], the authors proposed using the Frobenius norm to minimize the covariance matrix to *avoid collapse and reduce mean square error*, thereby enhancing invariance. They applied this norm during the training process of the SLM, which simplified the computation of eigenvalues and improved the overall efficiency of the model.

Another distance that is commonly used, and implemented by the authors in [2] is the Manhattan distance or L_1. The authors in [42] defined the distance as:

Definition 14. *Manhattan Distance* $= D_{MH}(a,b) = |x_1 - x_2| + |y_1 - y_2| + ... + |x_n - y_n| = \sum_{i=1}^{n} |x_i - y_i|$

In [43], the authors defined Manhattan distance as a function that calculates the distance between two data points by following a grid-like path. It is named the City Block distance because it represents the distance between points during a city road grid [43].

The authors in [44] defined the Manhattan distance as the distance that calculates the absolute differences between the coordinates of a pair of objects.

Euclidean distance is the most common distance [43]. This distance computes the *root of the square difference between the coordinates of a pair of objects* [44]. The authors in [44] defined the distance as below:

Definition 15. *Euclidean Distance* = $D_{ED}() = \sqrt{\sum_{k=1}^{m}(X_{ik} - X_{jk})}$

Another important distance is Minkowski. This distance is a generalization of the Euclidean distance and Manhattan distance [43]. The distance is defined as below:

Definition 16. *Minkowski distance* = $D_{MI}(x,y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$

In [45], a method was proposed to compare covariance matrices. The method uses the generalized eigenvalues from the covariance matrices. In [46], the distance was calculated, and the authors noted that computing the generalized eigenvalues can be done with $O(d^3)$ arithmetic operations, and the distance computation takes logarithmic time in d. This is much faster than comparing two histograms, which grow exponentially in computer processing. The formula for this distance is shown below:

Definition 17. *Eigenvalues distance* $= D_{EI}(d_1, d_2) = \sqrt{\sum_{i=1}^{n} ln^2 \lambda_i(D_1, D_2)}$

Where $\lambda_i(d_1, d2)$ are the generalisation of the eigenvalues, computed from,

$$\lambda_i d_1 x_i - d_2 x_i = 0 i = 1...d \tag{2.4}$$

In a recent investigation [47], various measures from information theory were explored, such as conditional expectation gain, mutual information, and conditional mutual information. These measures were applied to machine learning research, and their definitions are provided below.

The authors in [48] demonstrated that conditional expectation is:

Definition 18. $E[f(x)E(y \mid x)] = E(fy)$ *whenever is* $E(fy)$ *is finite, and that* $\sigma 2E(y \mid x) \leq\leq \sigma 2y$, *where* $E(y \mid x)$ *is the conditional expectation of* y *with respect to* x.

In [49], conditional expectation is defined as the sum of all conditional expectations of the different variables of y with respect to x, as below.

Definition 19. *Conditional Expectation* $= E(Y) = \sum_{i=0}^{\infty} y_i * P(Y = y_i)$

In [50], mutual information is defined as a measure of the amount of information one random variable contains about another random variable. It quantifies the reduction in uncertainty of one variable based on the knowledge of another variable. The formula can be denoted as follows:

Definition 20. *Mutual information* $= I(X;Y) = H(X) - H(X \mid Y)$

In [51], mutual information is used alongside a classifier with active learning to improve the model's results. It helped to learn more about the unlabeled data and improve the prediction of labels.

In [50], conditional mutual information is defined similarly to mutual information. However, it measures the reduction in uncertainty of a random variable based on the knowledge of another random variable, given a third random variable (the condition). The formula for this measure is:

Definition 21. *Conditional mutual information* $= I(X;Y \mid Z) = H(X \mid Z) - H(X \mid Y, Z)$

The variable mentioned in the above measures is known as the labels.

It is essential to validate an idea by testing it against real data or values, a process known as experiments [52]. In these experiments, variables are proposed and the results are analyzed. The design of experiments is the study of how to create and implement these tests. In [53], an experiment is designed to test the effects of some intervention on one or more variables. Below, we will define various important concepts for designing experiments and conducting investigations:

Definition 22. *Experimental unit can be a person, an animal, a factory, or any other entity that is analyzed during an experiment, according to [53].*

Definition 23. *A treatment refers to how the experimental unit is used. [54]*

An example of a treatment is if the experimental unit is a corn field, the treatments could be the different types of fertilizers used [54].

Definition 24. *Replication involves repeating the experiment with different experimental units [54].*

Definition 25. *Explanatory variables are the variables that are manipulated and variables that are not manipulated but potentially affect the outcome. [53]*

Definition 26. *Quantitative variables are defined as variables that can quantify values or recorded numbers. [53]*

In [53], if the variable is not quantitative, it is a categorical variable.

Definition 27. *The latent space is the compressed representation of the data used to find patterns [55].*

The data points in a dataset can be incomplete or inconsistent. Because of this, there is a research field known as data quality. In [56], data quality is defined as "fitness for use" because data is the most important asset that a company has.

Quality is defined in [56] as

Definition 28. *Quality refers to all the characteristics of an entity that impact its ability to meet stated and implied needs.*

Data quality dimension is defined by the authors in [56] as:

Definition 29. *Data quality dimension is a characteristic or part of information for classifying data and information requirements. It provides a way to measure and manage data quality.*

Dimensions consist of various concepts. For example, consistency refers to data that are presented in the same format and compatible with previous data [56]. Accuracy involves ensuring that the data values stored in the database correspond to real world values [56]. Completeness refers to having all the required parts of an entity's information present [56]. Duplication is a measure of redundant data points [56]. The author explained additional relevant concepts in the same paper.

2.2 State of the art

2.2.1 Semi supervised deep learning models

The semi-supervised models utilize various approaches that are relevant to this investigation. One such approach is called self-training, which involves using a labeled dataset and a pseudo-labeled dataset that was labeled in a previous step of the model [24].

The process begins by training the model with the labeled dataset and then generating predictions using the unlabeled data. The most confident predictions are added to the labeled dataset. The next step involves retraining the model with the new labeled dataset, and this process iterates until there is no more unlabeled data [24]. In a paper [57], the authors used the self-training approach in a model called Speed.

Co-training follows a similar process to self-learning but involves using multiple supervised classifiers. The main difference is that the newly labeled data will be added to the other classifiers for each iteration [24]. This approach aims to take advantage of any disagreements between the multiple learners.

In the co-training approach, there is multi-view co-training, which is the basic algorithm involving two learners. Additionally, there is single-view co-training, in which the approach attempts to automatically split the feature set in each iteration to address the predefined disjoint feature set [24].

In [58], a co-training method was proposed to train multiple deep neural networks to obtain different views. Adversarial regularization was used to exploit the different views and prevent them from collapsing on each other.

Another semi-supervised approach is co-regularization, which proposes a regularization framework where the ensemble quality and the disagreement among the learners are optimized simultaneously [24]. The key to this approach is a criterion function that uses two terms: penalizing incorrect predictions in the ensemble and penalizing the predictions from the classifiers [24].

Boosting is a different approach in which each base learner depends on the previous learner. The learner is given the full dataset, but the weights are determined based on the performance of the previous base learner. The final prediction is obtained using a linear function combination [24].

As a state-of-the-art technique, MixMatch is a semi-supervised algorithm that predicts low-entropy labels for unlabeled examples and combines labeled and unlabeled data using [11].

In 2019, Google Research introduced MixMatch. The process involves using data augmentation to predict labels for an unlabeled dataset. These guessed labels are then adjusted using a shaping function to minimize the entropy of the label distribution. The authors mentioned that lowering the temperature encourages the model to reduce the entropy predictions. Lastly, the MixUp step combines all labeled and guessed labels into one dataset, which serves as the data source for SSLM [11].

After MixMatch was released, another approach called FixMatch was published with a simpler method [10]. One of the differences is that in FixMatch, the guessed labels are kept only if the model assigns a high probability to one of the possible classes [10].

2.2.2 Semi Supervised Deep Learning with distribution mismatch

In [2], the impact of distribution mismatch in semi-supervised models in real-world applications was discussed. The authors found a strong correlation between the DeDiMs measures and MixMatch accuracy, suggesting a significant influence of feature distribution mismatch.

During the experiment, the authors discovered that training the model with datasets contaminated with the unlabeled Costa Rican data set resulted in the lowest accuracy. Furthermore, using a data set with a higher degree of contamination led to decreased accuracy [2].

The authors concluded that according to the results, further investigation in the creation of data quality metrics for semi-supervised deep learning models can narrow the gap between the investigation and real-world implementations [2].

In a recent approach mentioned in [17], the author pointed out that one of the main challenges in semi-supervised learning is the selection of a proper metric for sampling from the unlabeled data in order to extract informative points. To address this, the author proposed a new metric based on neighbor construction, which involves selecting peak data points for sampling using the Apollonius circle. The proposed algorithm aims to achieve agreement between the classifier and neighbor construction to assign labels to the unlabeled data during the training process [17]. This is a recent study investigating how to measure the impact of the unlabeled dataset in a Semi-Supervised Learning Model (SSLM) and proposing a metric.

2.2.3 Data quality metrics for unstructured data

In [56], the authors conducted a survey of the various important data quality dimensions discussed in the data quality literature. They concluded that achieving higher data quality requires considering multiple dimensions. The authors recommended selecting the appropriate dimensions that have a strong correlation with each other. The dimensions that the authors, in [56], recollected are:

1. **Belieaviability**: Defined as the extent to which data are accepted or regarded as true, real, and credible, the authors in [59] proposed a method to assess the provenance of the data, or the source from which the data originates. While their approach is primarily intended for structured data, it could also be applied to deep learning models.

2. **Interpretability**: is defined as the clarity and ease with which data can be understood or interpreted. In [60], the authors proposed a metric to measure the trust and interpretability of a model predictions. They calculate the factor of the mutual information between human decisions and the model's predictions, as well as the mutual information between human decisions and the true labels.

3. **Relevancy**: is defined as the extent to which data is applicable or helpful for the intended purposes. In [61], the authors proposed a framework using the Graph Convolution Network (GCN)-AutoEncoder Hash (GAH) algorithm to recognize data (illuminate dark data). They used Hamming distance and the Cluster PageRank (CPR) algorithm to calculate the importance score for each node (image), thereby obtaining a relevance representation based on semantic propagation.

4. **Accuracy**: is a measure of the proximity of a data value, v, to another value, $v^{'}$, that is considered correct.

5. **Consistency**, is defined as the extent to which data is presented in the same format and is compatible with previous data. This dimension could be useful to implement in deep learning models. However, during the research, no information was found to establish a correlation between this dimension and its use in deep learning models.

6. **Duplication**, is defined as a measure of unwanted duplicate records. In deep learning models, a library, such as ImageHash, can be used to eliminate duplicate records, but it is not a measure itself.

7. **Objectively (Objectivity)**, is defined as the extent to which information is unbiased, unprejudiced, and impartial. There is no research on objectivity

in deep learning models or in data quality for unstructured data.

8. **Data decay** is defined as the extent of negative change in the data. In [62], the authors proposed a method for learning using drift, a similar concept, and they developed a method for detecting changes in the probability distribution of examples. The method calculates the impact of changes in data after the model is trained and receives new information.

9. **Data Coverage**, is defined as a measure of the availability and comprehensiveness of data compared to the total data universe or population of interest. It is an interesting dimension; in deep learning, it could be possible to measure how much data will be needed and how much will be available.

10. **Appropriate amount of data**, equal to the name, this dimension is defined as how much data is appropriate for a task.

The data quality dimensions explained above were originally intended for structured data. However, it has been shown that few investigations have been conducted for unstructured data or for datasets (labeled and unlabeled) used in deep learning models. Accordingly, the authors in [7] draw attention to the fact that their paper represents the first and important discussion about dataset quality. They emphasize that as machine learning models are increasingly applied in real-world applications, data quality becomes crucial.

The authors in [63] define a high-quality dataset as one that accurately represents real-world scenarios, is comprehensive, and is free from biases. According to the authors, the accuracy and effectiveness of machine learning models can be significantly impacted if the dataset lacks the expected quality.

The authors [63] described more concepts in data quality that are important in this research:

1. **Completeness**: is defined as the degree to which subject data associated with an entity has values for all expected attributes and associated strength values in a given environment.

2. **Self-consistency**: is defined as the level of no contradiction in the semantics of the data in a given context.

3. **Unbiasedness**: is defined as the extent to which the distribution of data categories or features in a dataset aligns with the distribution in a given environment.

The authors in [64] detailed a different approach to data quality. They detailed four topics that are important in a data-centric deep learning model:

1. Data Collection: this topic is divided data acquisition that represents how is discovered, augmented or generated a new dataset. Data labeling is how to add informative observations that are useful for the model to learn.

2. Data Validation, Cleaning, and Integration: To validate data, it is recommended to use data visualization tools and schema generation techniques. Cleaning can be used to fix data, but it does not always result in improved model accuracy. To analyze all data together, the authors recommend alignment and co-learning techniques.

3. Robust Model Training: The authors concluded that data quality could still be an issue even after cleaning the data. Real-world datasets can still be dirty despite the data cleaning process. Based on this, the authors identified different flaws in the datasets, such as noisy features, missing features, noisy labels, and noisy features. They recommend using different model training approaches to decrease the potential for "data poisoning." Some of the techniques they recommend include adversarial training, knowledge distillation (Defensive distillation), feature squeezing, adversary detection network (MagNet), informative missingness patterns (GRU-D), loss correction (Bootstrap, F-correction, ActiveBias), sample selection (de Decouple, MentorNet, Coteaching), SELFIE, prestopping (MORPH), representative techniques and augmentation techniques (Mean-Teacher, MixMatch).

4. Fair Model Training: The author's purpose is to address bias and they recommend using fairness measures to discuss how to assess and mitigate unfairness. Fairness mitigation can be achieved using the following

metrics: demographic parity, equalized odds, predictive parity, individual fairness, and causality fairness. The authors proposed three techniques to mitigate unfairness: pre-processing mitigation (removing data bias before training), in-processing mitigation (modifying the objective function by adding fairness constraints, competing/modifying with a fairness discriminator, reweighting input samples for fairness), and post-processing mitigation (fixing predictions for fairness and combining models using randomization to obtain desired fairness).

The authors in [64] concluded that during this data-centric era, collecting data and improving its quality are becoming increasingly critical for deep learning. The authors aim for the deep learning community to start incorporating data collection, data cleaning, validation, and integration together.

In [7], the authors conducted cross-dataset generalization validation on various general-purpose datasets, revealing limited generalization capabilities even among datasets collected from the same source (the internet). They concluded that these datasets exhibited inherent bias, suggesting that incorporating data augmentation could mitigate this issue. Additionally, introducing negative set bias into the datasets to assess model behavior might also prove beneficial.

Building on the second step recommended in [7], the authors in [2] employed various dataset partitions and contaminated partitions to assess the accuracy of the SSDL model

In [65], the authors elaborate on the deficiency of machine learning tools in processes aimed at improving data quality, highlighting that it is often an ad-hoc process during model implementation. They introduce a framework designed to identify noisy samples within datasets and to identify informative samples.

In [65], the authors proposed two model-based metrics: a confidence-certainty metric and a certainty metric. The first metric aims to identify noisy or mislabeled samples within a dataset. The second metric, based on the confident learning approach, learns a class-conditional joint distribution Q between the provided dataset labels (assumed noisy) and the latent labels (assumed uncorrupted) to identify noisy samples.

In [47], a new framework called PRISM was proposed, utilizing metrics such as conditional expectation gain, mutual information, and conditional mutual information, defined in the previous section.

These measures require a conditional or a variable for calculation. In our research, the variable is the labels. However, in unlabeled datasets, the labels are unknown, thus making these metrics unsuitable as proposed measures for this research.

The authors in [13], [1] and [2] adopted an alternate methodology, introducing DeDiMs. They employed samples from two datasets to perform comparisons of the populations, with the objective of calculating dissimilarity measures within the feature space subsequent to the datasets being processed by a Wide-ResNet model pretrained on ImageNet.

The dissimilarity measures were assessed across two distinct subsample datasets employing both Euclidean and Manhattan distances. Furthermore, two non-parametric density metrics, specifically Jensen-Shannon divergence and cosine similarity, were evaluated.

The steps to measure DeDiMs in [1] are:

1. The researchers initiated the dataset sampling procedure by constructing subsample datasets for D_a and D_b, referred to as D_{sa} and D_{sb} respectively, where s represents the sample size. They authors determined that D_a corresponds to the labelled dataset D_l and D_b corresponds to unlabelled dataset D_u.

2. The subsequent phase entailed the transformation of each observation x_j from the subsample dataset D_{si}, where each $x_j \in \mathbb{R}^n$, and n denotes the dimensionality of the output space. This output space is derived using a feature extractor f, whereby each observation x_j is processed through the function $f(x_j)$, resulting in the generation of h_j, as delineated below:

$$h_j = f(x_j) \tag{2.5}$$

3. The authors used an ImageNet pretrained Wide-ResNet feature extractor, employing the last convolutional layer which returns 512 features.

4. The feature vector $h_j \in \mathbb{R}^{n'}$, where $n' < n$ and $n' = 512$. Each feature vector from the sampling datasets D_{si} created a new dataset denoted as H_{sa} and H_{sb}.

The authors in [1] designated the computed distances as $d_{l2}(D_{sa}, D_{sb}, s, C)$ and $d_{l1}(D_{sa}, D_{sb}, s, C\mathcal{C})$, employing H_{sa} and H_{sb} as inputs, $\mathcal{C}$ as the total number of samples to compute the mean sampled dissimilarity measure, and s as the sample size,

1. The distances d_{lp}, p = 1 or p =2 are Manhattan and Euclidean distances for each feature vector H_{sa} and H_{sb}, where $h_j \in H_{sa}$ find the closest feature vector in $h_k \in H_{sb}$, $\hat{d}_j = min_k \left\| h_j - h_k \right\|_p$, creating a list of distance calculations as, $d_{lp}(D_{sa}, D_{sb}, s, \mathcal{C}) = \left\{ \hat{d}_1, \hat{d}_2, \hat{d}_3, ... \hat{d}_C \right\}$.

2. To create a inter-data distance reference, the authors computed the distances D_{sa} against itself $d_{lp}(D_{sa}, D_{sa}, s, \mathcal{C})$, thereby generating a list of distances given by $\left\{ \check{d}_1, \check{d}_2, \check{d}_3, ..., \check{d}_C \right\}$. In the research, D_{sa} was equivalent to D_l, the labelled dataset.

3. To ensure the absolute differences between the reference and inter-data $d_c = \left| \hat{d}_c - \check{d}_c \right|$ distances are significant, the authors computed the p-value.

4. In the final step, when the distance $d_{lp}(D_{sa}, D_{sb}, s, \mathcal{C})$ is calculated for the two datasets, the average reference-subtracted distance $\bar{d}$ and the corresponding p-value are determined.

The DeDiMs distances proposed to measure the density followed a similar approach [1],

1. For each dimension in H_{sa} and H_{sb}, the normalized histograms were computed to approximate the density functions c_{ra} and c_{rb} respectively.

2. The distances Jensen-Shannon is denoted as d_{js} and cosine distance as d_c

3. The dissimilarity between c_{ra} and c_{rb} for sample j was computed as $\hat{d}_j = \sum_{r=1}^{n'} \delta_g(c_{ra}, c_{rb})$, where $g = JS$ and $g = C$, representing Jensen-Shannon and cosine distances, respectively.

4. The sum was computed for $\mathcal{C}$ samples, resulting a list as $\left\{\hat{d}_1, \hat{d}_2, \hat{d}_3, ..., \hat{d}_C\right\}$
5. Similar to the Manhattan and Euclidean distances, intra-data distances were calculated using the labeled dataset D_l, resulting in a list as follows: $\left\{\check{d}_1, \check{d}_2, \check{d}_3, ..., \check{d}_C\right\}$
6. To validate that $d_c = \left|\hat{d}_c - \check{d}_c\right|$ is statistically significant, the p-value was calculated and computed.

The authors found that the above approach DeDiMs represented an important advancement in their line of investigation. However, the proposed dissimilarity measures do not fully meet the criteria to be classified as mathematical pseudo-metrics due to non-zero distance between an object and itself and lack of symmetry properties. Nevertheless, these density-based dissimilarity measures serve as effective filters for estimating significant accuracy improvements [1]. The computational process associated with the approach does not currently optimize speed or efficiency.

In a similar approach using the feature space, [2] proposed two methods to assess the suitability of an unlabeled dataset for a model. One method involves using a feature extractor and histograms, while the other calculates the Mahalanobis distance from a covariance matrix. The latter approach yielded superior results, improving model accuracy, and proved to be computationally less expensive and faster than calculating histograms. Based on these findings, the method proposed in this research will adopt a similar approach.

3. Scientific Proposal

3.1 Research questions

Following the state of art, below are the questions that aroused for this investigation:

1. Using distance as a measure, can we find a correlation between the measure results and the accuracy of the SSDL model?

2. Can we devise a faster algorithm to compare two datasets than the DeDiMs proposed in [2] and [1]?

3. Among the proposed data quality metrics, which performs better in terms of computational efficiency and correlation?

3.2 Proposed method

The proposed method involves creating one or more metrics for detecting COVID-19 using chest X-ray images in unlabeled datasets. It uses a similar approach in preprocessing to DeDiMs, as both require the same feature space for comparison.

1. The entries of the proposed method are two datasets $D_{l\tau}$ and $D_{u\tau}$ obtained after a random sampling of D_l and D_u. Labelled dataset $D_{l\tau} = \mathbf{x}_1, ..., \mathbf{x}_\tau$ and unlabelled dataset $D_{u\tau} = \mathbf{x}_1, ..., \mathbf{x}_\tau$, where τ is the sampling size of the dataset.

2. For each $x_j \in \mathbb{R}^n$, from $D_{l\tau}$ and $D_{u\tau}$, where n is the dimensionality of the feature space, a transformation or feature extractor f will be applied to generate a feature array $s_j = f(x_j)$. The proposed feature extractor f to be used is the ImageNet pretrained AlexNet architecture, which outputs a feature dimensionality of 256.

3. Each feature vector $s_j \in \mathbb{R}^{n'}$ where $n' < n$ and $n' = 256$ features, each feature vector created, derive new datasets $H_{l\tau}$ and $H_{u\tau}$

4. The last step is to calculate the covariance matrix Σ_l and Σ_u from $H_{l\tau}$ and $H_{u\tau}$ respectively.

Alexnet is chosen due to hardware limitation in the processing of bigger neural network architectures.

The proposed method will use Frobenius due to the reason the trace fuction can

After the pre-processing, the next step is to calculate the Frobenius distance $d_F(D_{l\tau}, D_{u\tau})$ and then calculate the Mahalanobis distance $d_M(D_{l\tau}, D_{u\tau})$.

1. Frobenius distance d_F, will be computed using covariance matrices Σ_l and Σ_u obtained from $H_{l\tau}$ and $H_{u\tau}$, as follows,

$$d_F(D_l, D_u) = \|\Sigma_l - \Sigma_u\| = \sqrt{trace((\Sigma_l - \Sigma_u)^t(\Sigma_l - \Sigma_u))} \tag{3.1}$$

   The Frobenius distance is selected for its effectiveness in calculating matrix norms, which is essential for identifying outliers in this investigation. This matrix norm leverages the trace to compute eigenvalues, which are key for validating matrices. Additionally, the Frobenius distance is computationally efficient, adding minimal overhead to the method while providing accurate, expected results.

2. Mahalanobis distance d_M will be calculated using the mean of labelled dataset $H_{l\tau}$ denoted as μ_l, and the mean of the unlabeled dataset, $H_{u\tau}$, denoted as μ_u, as follows,

$$d_M(D_l, D_u) = (\mu_l - \mu_u)^T \Sigma_l^{-1} (\mu_l - \mu_u) \tag{3.2}$$

   The Mahalanobis distance is chosen for its robust outlier detection capabilities. This distance metric assumes that the data follows a normal (Gaussian) distribution, allowing it to account for correlations between variables. As a result, the Mahalanobis distance provides a more informative distance measure that reflects both the variance and the correlations within the dataset, making it particularly effective for identifying outliers.

The distances results and MixMatch results reported in [1] are compared using the Pearson correlation technique.

The new proposed method for comparing two populations will be more computationally efficient than the density measures calculated in the DeDiMs method. This method will use covariance matrices to compare the populations directly, without the need for a feature extractor, thereby avoiding additional computational burden.

3.3 Hypothesis

1. The proposed measures are 2 times faster than the previous DeDiMs method (with statistical significance).

2. The proposed measures yield a high correlation accuracy (more than 70%) with state-of-the-art SSDL methods like MixMatch.

4. Experimental design and result

4.1 Experimental design

In this section, two different types of experiments are formulated to apply the proposed method. First, control experiments will be conducted to analyze the performance of the proposed metric. The results of these control experiments represent the initial application of this method, providing insight into expected metric values.

The second experiment is divided into two sections: general-purpose experiments and COVID-19 experiments. These experiments will demonstrate how metrics or measurements behave for specific types of images or unlabeled datasets with different distributions. After obtaining the results, conclusions can be drawn comparing the outcomes from these different types of experiments.

4.1.1 Control Experiments

The purpose of the control experiments is to ensure that the proposed method works as expected and to assess its computational performance. The design will determine the path forward and the expected results.

The general-purpose datasets used in the experiment include CIFAR-10, Cats and Dogs, MNIST, and SVHN. Random partitions with labeled and unlabeled data were created for the experiment. Table 4.2 presents the proposed divisions of the general-purpose datasets used in the experiment.

The experiment aims to compare two datasets that are similar as well as datasets that are completely different from each other. For example, in one of the divisions, detailed in 4.2, the experiment will compare the Cats and Dogs dataset with SVHN. The Cats and Dogs dataset consists of images of dogs or cats, while SVHN contains images of street house numbers. The expectation is to obtain distance results that represent this difference.

Dataset	Description	Num. Classes	Num. Observations	Resolution
MNIST	Collection of images with handwritten digits	10	60000	28×28
SVHN	Images of printed digits (i.e. house numbers)	10	600000	32×32
TinyImageNet	Smaller version of ImageNet dataset for generic object recognition	200	100000	64×64
Gausssian Noise	Images of wear articles in grayscale	10	60000	28×28
Salt and Pepper	Images of wear articles in grayscale	10	60000	28×28

Table 4.1: Description of general purposes datasets used in this work.

D_l		D_u	$\%\_uOOD$
MNIST	**Dif**	SVHN	100
CIFAR-10	**Sim**	CIFAR-10	100
Cats and dogs	**Dif**	SVHN	100
	Sim	Cats and Dogs	100

Table 4.2: Division of datasets used in the control experiment will involve a sample of several general-purpose datasets. The division and comparison between datasets will be based on their similarity or dissimilarity.

The experiment will generate distance values calculated using the proposed distances d_F and d_M. The goal of this experiment is to identify patterns in the results. When the datasets are similar or identical, the distance value is expected to be small, ideally between 0 and 2 points. Conversely, when the datasets are completely different, the distance value is expected to be higher than these values.

The experiment aims to visualize patterns and validate the computational resources consumed by the distances. The goal is to ensure that the computation time does not exceed that required by the distances proposed in [1].

4.1.2 General purposes datasets and Covid-19 experiments

The second phase of experimentation is divided in two, the first subsection is to compute the proposed distances using various contaminated datasets and different contamination levels detailed in 4.3. This design follows the approach outlined in the article [1]

General purposed and contamination experiments

This first experiment aims to find a correlation with the experiments executed in [1] and detail in 4.4 It will compare the values generated by the proposed method with the distance values and accuracy generated by the authors in [1].

The experiment will be conducted ten times to generate accurate percentages for comparison with values from [1]. Each run will involve ten random batches with 80 observations sampled from the dataset. The time taken to compute the distance results will be recorded each time the distances are calculated.

The expectation is to observe the same trend as in the control experiments: higher distance results correspond to higher contamination levels, while lower distance results indicate lower contamination levels.

The experiment will calculate the correlation coefficient using the Pearson method. It will compare the distance values obtained using the same contaminated datasets and then correlate them with the accuracy values reported in [1]. The expected outcome is to find a strong correlation between the distance and accuracy results.

The datasets used in the experiment are CIFAR-10, MNIST, and Fashion-MNIST, with each dataset compared against a contaminated dataset detailed in 4.3. The contamination and different data sets are the same datasets and the contamination set in [1].

The experiment will compare the processing time of the proposed distances against the processing time obtained by the authors in [1]. The expectation is that the time decreases two times the time reported by the authors in the article.

D_l	T_{OOD}	D_u	%_uOOD	n_l		
MNIST	**Dif**	TI	50	60	100	150
			100	60	100	150
		GN	50	60	100	150
			100	60	100	150
		SAP	50	60	100	150
			100	60	100	150
CIFAR-10	**Dif**	SVHN	50	60	100	150
			100	60	100	150
		GN	50	60	100	150
			100	60	100	150
		SAP	50	60	100	150
			100	60	100	150
FashionMNIST	**Dif**	TinyImage	50	60	100	150
			100	60	100	150
		GN	50	60	100	150
			100	60	100	150
		SAP	50	60	100	150
			100	60	100	150

Table 4.3: General-purpose contaminations and divisions used in the second phase of experimentation are detailed in the table. The table illustrates the level of contamination alongside the number of unlabeled observations in each dataset batch. This division is for comparison against the accuracy reported in [1].

S_l	S_u	$\%_{uOOD}$	d_{ℓ_2}	d_{ℓ_1}	d_{JS}	d_C	row #
MNIST	OH	50	*0.011 ± 0.006*	*0.459 ± 0.28*	*0.266 ± 0.221*	*0.811 ± 0.512*	0
		100	*0.014 ± 0.019*	*0.38 ± 0.507*	1.001 ± 0.725	1.263 ± 0.665	1
	SVHN	50	*0.09 ± 0.017*	1.569 ± 0.504	6.789 ± 0.924	12.021 ± 1.757	2
		100	0.25 ± 0.053	4.702 ± 1.04	52.349 ± 3.292	42.026 ± 4.311	3
	TI	50	0.008 ± 0.023	1.519 ± 0.223	3.663 ± 0.772	5.512 ± 0.767	4
		100	0.217 ± 0.04	4.3 ± 0.636	10.305 ± 1.667	15.18 ± 2.698	5
	GN	50	0.11 ± 0.0219	1.958 ± 0.534	14.785 ± 1.052	23.59 ± 1.859	6
		100	0.357 ± 0.081	5.987 ± 1.091	52.349 ± 4.253	86.21 ± 3.471	7
	SAPN	50	0.089 ± 0.0311	2.479 ± 0.7433	15.116 ± 1.475	20.151 ± 1.619	8
		100	0.323 ± 0.07	6.308 ± 1.366	53.397 ± 4.253	77.456 ± 4.474	9
CIFAR-10	OH	50	*0.056 ± 0.023*	*0.915 ± 0.934*	*0.338 ± 0.325*	0.892 ± 0.402	10
		100	*0.061 ± 0.04*	0.769 ± 0.461	*0.451 ± 0.41*	0.648 ± 0.407	11
	TI	50	0.082 ± 0.037	*0.928 ± 0.815*	*0.388 ± 0.243*	*0.423 ± 0.362*	12
		100	*0.056 ± 0.048*	*0.992 ± 0.517*	*0.469 ± 0.426*	0.415 ± 0.232	13
	SVHN	50	*0.055 ± 0.032*	*0.948 ± 0.699*	*0.665 ± 0.565*	*0.414 ± 0.357*	14
		100	*0.075 ± 0.036*	1.291 ± 0.925	0.736 ± 0.658	0.581 ± 0.343	15
	GN	50	*0.107 ± 0.083*	1.344 ± 1.156	1.708 ± 0.421	3.001 ± 0.696	16
		100	0.127 ± 0.087	1.531 ± 0.767	5.855 ± 0.552	8.703 ± 0.926	17
	SAPN	50	0.1146 ± 0.044	1.854 ± 0.894	2.299 ± 0.691	2.561 ± 0.762	18
		100	0.208 ± 0.05	5.502 ± 1.156	8.225 ± 0.866	9.554 ± 0.489	19
FashionMNIST	OH	50	*0.02 ± 0.012*	*0.34 ± 0.162*	*0.669 ± 0.566*	*0.575 ± 0.423*	20
		100	0.059 ± 0.032	0.801 ± 0.402	0.305 ± 0.237	0.774 ± 0.343	21
	FP	50	0.105 ± 0.0526	2.168 ± 0.774	7.263 ± 0.622	5.305 ± 0.405	22
		100	0.195 ± 0.0457	4.819 ± 1.077	9.056 ± 0.462	11.286 ± 0.751	23
	TI	50	*0.04 ± 0.03*	*0.798 ± 0.542*	*0.897 ± 0.516*	0.897 ± 0.516	24
		100	0.065 ± 0.03	1.66 ± 0.45	1.4 ± 0.488	1.912 ± 0.683	25
	GN	50	*0.047 ± 0.03*	*0.533 ± 0.347*	2.819 ± 0.703	3.843 ± 0.704	26
		100	0.074 ± 0.041	1.325 ± 0.631	9.042 ± 0.699	15.511 ± 0.445	27
	SAPN	50	0.036 ± 0.022	0.52 ± 0.303	2.799 ± 0.497	2.799 ± 0.497	28
		100	0.076 ± 0.044	1.411 ± 0.548	8.464 ± 0.553	8.464 ± 0.553	29

Table 4.4: Distance measures reported in [1]. The distances are calculated using labelled and unlabelled datasets S_l and S_u (mean)

	Costa Rica	China	ChestX-ray8(NIS)	Indiana
No. of patients	105	5856	65240	4000
Patient's age	7-86	children	0-94	adults
No. of obs.	105	5236	224316	8121
Hospital/clinic	Clínica Chavarría	No info.	Stanford Hospital	Indiana Network for Patient Care
Resolution	1907 X 1791	1300 X 600	1024 x 1024	1400 X 1400

Table 4.5: Specification of the COVID-19 Datasets

COVID-19 datasets experiments

In the second experiment of the second phase of the experimental design is to perform the proposed method with specific COVID-19 datasets.

The datasets were collected from various sources. COVID-19 positive cases were obtained from the open repository of Dr. Cohen [66], while negative cases were collected from other pathologies such as MERS, ARDS, and SARS from different sources. Detailed information about the datasets is provided in table 4.5 from [2].

The experiment aims to test whether the proposed method behaves similarly in datasets with fewer observations and minimal differences in the images. The expectation is to observe a similar pattern and comparable processing time as in previous experiments.

To validate the computational performance and the measurement results. The experiment will follow the contamination detailed in table 4.6. Different partitions will be created from four different datasets. The experiment will be executed 10 times, using 10 random batches of 40 observations.

The compute time for each run will be recorded from the start of the distance pre-processing to the completion of the process. The expectation is to observe the same trend as in previous experiments, but with smaller differences between higher and lower contaminations. This is due to the subtle differences between the datasets, which are not easily visible.

D_l	D_u	T_{OOD}	D_{uOOD}	%_$uOOD$	n_l	
Indiana	China	O_s	Costa Rica	65	20	40
				35	20	40
	Indiana	**Dif**	Costa Rica	65	20	40
				35	20	40
	NIS	**Dif**	Costa Rica	65	20	40
				35	20	40

Table 4.6: Covid-19 contamination and division experiments. The datasets are obtained from different parts of the world using different machines to generate the X-Ray image.

The acceptance of the hypothesis of this work will dependent on the results of this experiment.

The experiment follows a configuration similar to that established by the authors in [2]. It aims to compare and establish a correlation between the results of both experiments. The expectation is to find a strong correlation between the distance results and accuracy using Pearson's method.

The experiment will compare the distances results against the following results from the table 4.7 reported in [2].

The table **??** detailed the accuracy values obtained by the authors in [2]. The experiment compares these values using the Pearson method to get the correlation value.

4.2 Experiments Results

The first experiments showed the results detailed in the table 4.11. The Mahalanobis distance method demonstrated a strong trend in the values. However, the experiments performed using the Frobenius distance did not show the expected consistency. Although the results generally followed a pattern, one experiment deviated from it. Therefore, the Frobenius method will proceed to the second phase of experimentation to validate the results with additional experiments.

Dataset	$d(S_l, S_u)$
China	2.06 ± 0.11
Costa Rica	30.9 ± 0.4
ChestX-ray8	1.04 ± 0.27
ChestX-ray8 65% - Costa Rica 35%	3.95 ± 0.94
ChestX-ray8 35% - Costa Rica 65%	11.84 ± 0.94
China 65% - Costa Rica 35%	5.74 ± 0.79
China 35% - Costa Rica 65%	14.85 ± 0.0
Indiana 65% - Costa Rica 35%	6.33 ± 0.3
Indiana 35% - Costa Rica 65%	16.61 ± 0.3

Table 4.7: Reported in [2], Cosine DeDiM distance, using 10 different batches of 80 observations, between the labelled and unlabelled datasets, S_l and S_u. The S_l used to calculate the distance value was the Indiana dataset. Using Alexnet, to keep computing cost low.

Dataset	$n_l = 40$	$n_l = 20$
Supervised	0.785 ± 0.038	0.809 ± 0.085
Indiana (with COVID-19$^+$ [**?**])	0.782 ± 0.039	0.75 ± 0.06
China	0.648 ± 0.0247	0.659 ± 0.033
Costa Rica	0.501 ± 0.001	0.5 ± 0.001
ChestX-ray8	0.72 ± 0.076	0.71 ± 0.074
ChestX-ray8 65% - Costa Rica 35%	0.711 ± 0.083	0.66 ± 0.11
ChestX-ray8 35% - Costa Rica 65%	0.516 ± 0.022	0.511 ± 0.016
China 65% - Costa Rica 35%	0.701 ± 0.055	0.688 ± 0.084
China 35% - Costa Rica 65%	0.53 ± 0.023	0.528 ± 0.019
Indiana 65% - Costa Rica 35%	0.532 ± 0.024	0.559 ± 0.059
Indiana 35% - Costa Rica 65%	0.501 ± 0.001	0.503 ± 0.009

Table 4.8: [2] Accuracy of a Alexnet model trained with MixMatch with different D_u^s datasets. The S_u unlabelled datasets Chest-Xray8, Costa Rican and Chinese datasets include only COVID-19$^-$ observations. The S_l used is the Indiana dataset.

S_I	T_{OOD}	S_{uOOD}	$\%\_uOOD$	D_F **results**
MNIST	**Dif**	SVHN	100	0,001060
CIFAR-10	**Sim**	CIFAR-10	100	**0,000227**
Cats and dogs	**Dif**	SVHN	100	0,000555
	Sim	Cats and Dogs	100	0,000289

Table 4.9: Frobenius 1rst Experiment Results. The results are distance values. The table illustrates that the expected results are not consistent across all the experiments performed in this first experimental phase, therefore, it cannot be assumed a trend in the results.

S_{IOD}	T_{OOD}	S_{uOOD}	D_F **Time (s)**	D_M **Time (s)**
MNIST	Sim	SVHN	2,8150	1,3424
	Dif	GausianNoise	2,4276	1,3282
CIFAR-10	Sim	CIFAR-10	2,6548	1,3994
Cats and dogs	Dif	SVHN	1,7278	1,3468
	Sim	Cats and Dogs	1,9562	1,3673

Table 4.10: Control Experiments Time results in seconds. This time represents when the Distance pre-processing time started after the distance is calculated. Distance D_F resulted in higher times than D_M distance

In Table 4.9, the Frobenius MNIST-SVHN experiment resulted is a higher value, higher than the results of similar experiments. In the MNIST vs. SVHN experiment, categorized as different experiment, the distance result is higher than the other results. The concern with these results is that the Cats and Dogs vs. SVHN experiment reported smaller values than the MNIST vs. SVHN results, as shown in the table. Therefore, further experimentation will be required. The expected result is a higher value, as observed in control experiments. The question now is whether the discrepancies in results are due to dataset characteristics or contamination levels. Therefore, more investigation and analysis will be required for the Frobenius distance.

S_{IOD}	T_{OOD}	S_{uOOD}	%_$uOOD$	D_F distance results	D_M distance results
MNIST	Sim	SVHN	100	**0,001060**	**72,503**
	Dif	GausianNoise	100	0,000532	201,292
CIFAR-10	Sim	CIFAR-10	100	0,000227	1,932
Cats and dogs	Dif	SVHN	100	0,000555	70,174
	Sim	Cats and Dogs	100	0,000289	0,9438

Table 4.11: Control Experiment Results. The unit of the results are distance values. The results illustrate that the experiment generated the expected results. The two best values are the experiment of MNIST vs SVHN when the contamination is 100%, because D_F ans D_M returned an expected result when the datasets are similar but not equal.

The preliminary findings of the Mahalanobis distance metric align with the objectives of the proposed method.

The experiments reported an average of 2,316 seconds in Frobenius distance and 1,356 seconds in Mahalanobis distance. As an initial conclusion, the processing time in d_M is less than the reporting time in d_F.

In the second phase of experiments, it was observed that the Mahalanobis distance results remained consistent with those from the first experimental phase. Table 4.12 shows that when the contamination is 50%, the results are smaller than when the contamination is 100%. This trend is consistent across different experiments involving general-purpose datasets.

In Table 4.12, Mahalanobis distance experiments CF-10 vs $CF-10_{TN}$ 100% n_l $60^{'}$ and 'CF-10 vs $CF-10_{TN}$ 50% n_l $60^{'}$ demonstrate that the distance value for 100% contamination is $4,081$, while for 50% contamination, the reported value is $2,207$. An interesting observation is that the distance value for the 100% contamination experiment, despite being labeled 'low,' is double that of the 50% contamination value.

The experiments detailed in 4.12 demonstrate that the trend is consistent when the contamination is 50%, with smaller distance results compared to when the contamination is 100%.

The results in Table 4.12 indicate that the distance value for the MNIST vs MNIST contaminated with TinyImage 50% n_l 60 experiment is 12 times smaller than that of the MNIST vs MNIST contaminated with TinyImage 100% n_l 60 experiment. This trend consistently shows a larger difference when using the Gaussian Noise dataset for contamination.

The Pearson coefficient reported in Table 4.12 indicates a strong negative correlation between the Mahalanobis distance values and the accuracy reported in [1]. This means that when one variable increases, the other decreases. For example, when the contamination is 50%, the distance value decreases while the accuracy increases, and when the contamination is 100%, the accuracy value decreases.

The table 4.12 illustrates when the distance value is a higher value the corresponding accuracy is lower, and when the distance result is lower, the corresponding accuracy is higher. This conclusion is proved based on the Pearson values reported in the table 4.12. This conclusion is consistent in all the MNIST, CIFAR-10, and FashionMNIST experiments.

The time reported in table 4.13 is the average time of 1,352 seconds, or 1,335 seconds and 1,321 seconds of the MNIST, CIFAR-10, and FashionMNIST experiments, respectively.

The plot 4.2 represents the values gathered from the three different dataset experiments. The plot shows that the majority of the inputs indicate a positive relationship between the Mahalanobis distance results (X) and the accuracy results (Y).

In the plot 4.3 illustrates that the Mahalanobis distance results obtained in the CIFAR-10 experiments are consistent with the expected trend, as represented in Table 4.12. The plot demonstrates a positive relationship between variables X and Y.

S_l	T_{OOD}	S_u	$\%_{OOD}$	n_l				Pearson Correlation
				60		100		
				d_M	Accuracy	d_M	Accuracy	
MNIST	Dif	TI	50	2,428	0.642 ± 0.094	2,501	0.739 ± 0.074	
			100	14,458	0.637 ± 0.097	14,356	0.732 ± 0.074	
		GN	50	3,335	0.606 ± 0.0989	3,301	0.713 ± 0.087	
			100	212,986	0.442 ± 0.099	219,282	0.461 ± 0.073	
		SAPN	50	2,560	0.631 ± 0.102	2,492	0.735 ± 0.082	
			100	60,548	0.48 ± 0.0951	62,713	0.524 ± 0.09	
								-0,619
CIFAR-10	Sim	TI	50	2,207	0.435 ± 0.054	2,344	0.473 ± 0.039	
			100	4,081	0.417 ± 0.020	4,179	$\mathbf{0.480 \pm 0.039}$	
	Dif	SVHN	50	3,382	0.419 ± 0.027	3,425	0.464 ± 0.044	
			100	15,399	0.385 ± 0.034	16,179	0.418 ± 0.035	
		GN	50	7,766	0.409 ± 0.047	4,716	0.454 ± 0.048	
			100	100,169	0.297 ± 0.029	104,390	0.306 ± 0.034	
		SAPN	50	2,621	0.438 ± 0.029	2,599	0.455 ± 0.037	
			100	53,157	0.236 ± 0.031	57,181	0.246 ± 0.032	
								-0,823
FashionMNIST	Dif	TI	50	2,159	0.690 ± 0.065	2,284	$\mathbf{0.745 \pm 0.093}$	
			100	**8,459**	0.690 ± 0.073	**8,526**	0.728 ± 0.066	
		GN	50	3,127	0.644 ± 0.061	3,289	0.689 ± 0.075	
			100	152,02	0.352 ± 0.025	167,579	0.366 ± 0.065	
		SAPN	50	2,376	0.671 ± 0.072	2,412	0.708 ± 0.095	
			100	51,693	0.276 ± 0.069	57,289	0.297 ± 0.046	
								-0,767

Table 4.12: Second Experiment results. Mahalanobis distance results compared against accuracy values reported in [1]. The table highlights the best accuracy values alongside the corresponding distance results.

S_l	T_{OOD}	S_u	$\%_{ood}$	n_l	
				Time (s)	
				60	**100**
MNIST	Dif	TI	50	1,367	1,323
			100	1,381	1,359
		GN	50	1,355	1,324
			100	1,324	1,333
		SAPN	50	1,304	1,524
			100	1,307	1,322
				1,352	
CIFAR-10	Sim	TI	50	1,338	1,329
			100	1,354	1,312
	Dif	SVHN	50	1,354	1,296
			100	1,385	1,277
		GN	50	1,370	1,282
			100	1,396	1,308
		SAPN	50	1,348	1,307
			100	1,387	1,310
				1,335	
FashionMNIST	Dif	TI	50	1,221	1,285
			100	1,233	1,263
		GN	50	1,296	1,303
			100	1,309	1,307
		SAPN	50	1,334	1,372
			100	1,333	1,379
				1,321	

Table 4.13: Second Experiment, Mahalanobis Distance processing time. This table demonstrates the time in seconds consumed by the distance to finish the process.

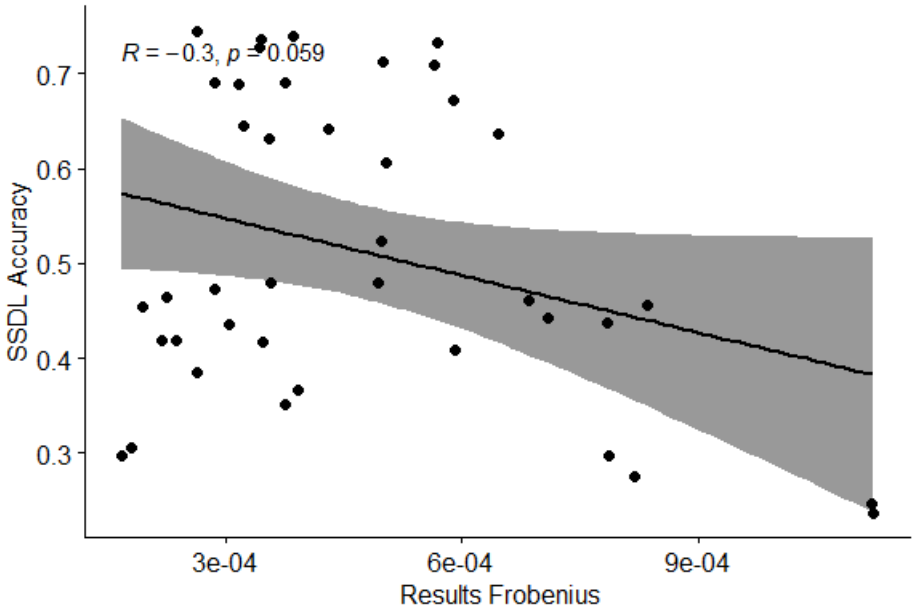

Figure 4.1: Second experiment, Frobenius results vs Accuracy results. The plot illustrates the results obtained from the different experiments conducted in this phase. The data points are spread across the plot without following a discernible pattern, indicating that the results do not demonstrate the expected consistency.

The Frobenius distance results presented in table 4.15 indicate that its consistency is not as strong as that of the Mahalanobis distance. For example, experiments involving CIFAR-10 and Gaussian noise contamination yield results that contradict the expected outcomes, although the expected consistency is maintained across the other experiments.

The Pearson coefficient in table 4.15 shows a strong negative correlation, which follows the same pattern as the Mahalanobis distance results. When the contamination is higher, the accuracy decreases, and when the contamination is lower, the accuracy of the model increases.

Both distances from the proposed method behave similarly, but in the Mahalanobis distance, the expected trend is consistent and persistent, unlike the Frobenius distance results, which show similar correlation results.

Figure 4.1 indicates that the Frobenius results exhibit an inconsistent trend, contrary to the expected experimental design. The data points in the plot do not follow a trend and there is no positive relation between the variables X and Y.

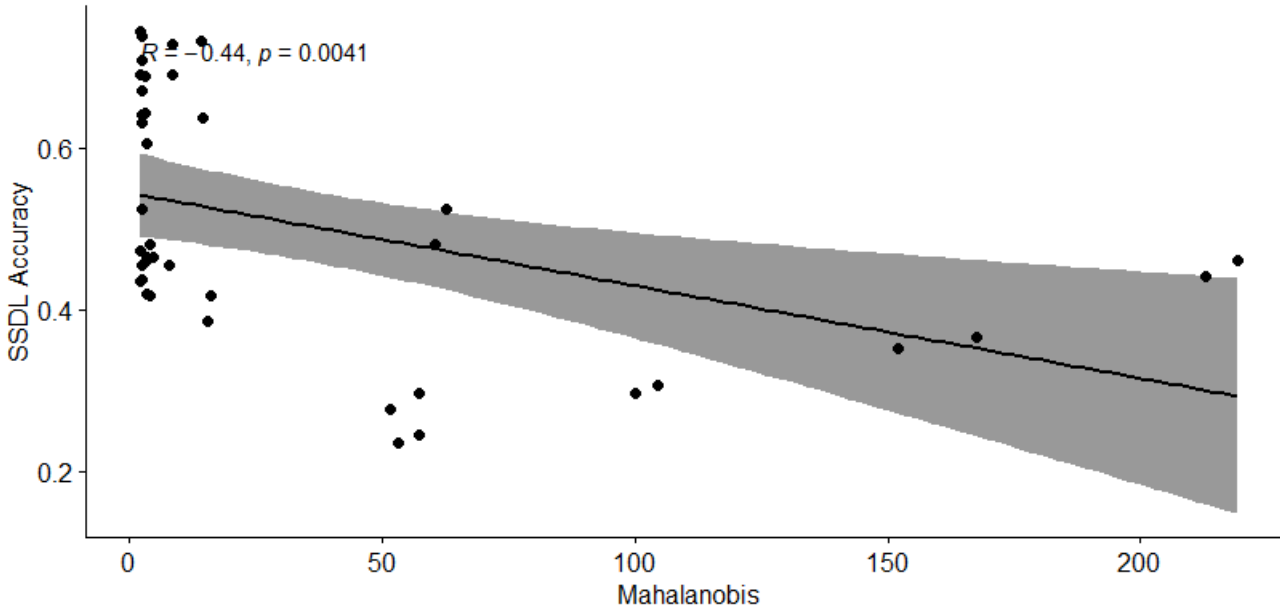

Figure 4.2: Mahalanobis results vs Accuracy results. The plot represents the MNIST, CIFAR-10 and FashionMNIST results versus the accuracy reported in [1]. The plot illustrates a positive relation between variables because it shows a down x pattern.

Table 4.16 compares the Pearson Correlation of the proposed method with the results from the histogram distance d_{js}. In all three scenarios, the histogram distance is consistently greater than the results generated by the proposed method. These findings indicate that when Mahalanobis and Frobenius results are expected but the density distance proposed in [1] serves as a better predictor compared to the proposed method. The authors in [1] explained that this discrepancy is related to the quantitative approximation of the feature distribution mismatch implemented in the distance.

After analyzing the results of the distances, it is necessary to examine the reported times. Before analyzing and reporting the results of a statistical test, it is essential to first test the data.

The figures 4.6 and 4.7 illustrate the variances between the experiments. The Histogram or Distance d_{js} results exhibit the majority of the differences. In contrast, the variances for the Frobenius and Mahalanobis distances are similar, although not identical.

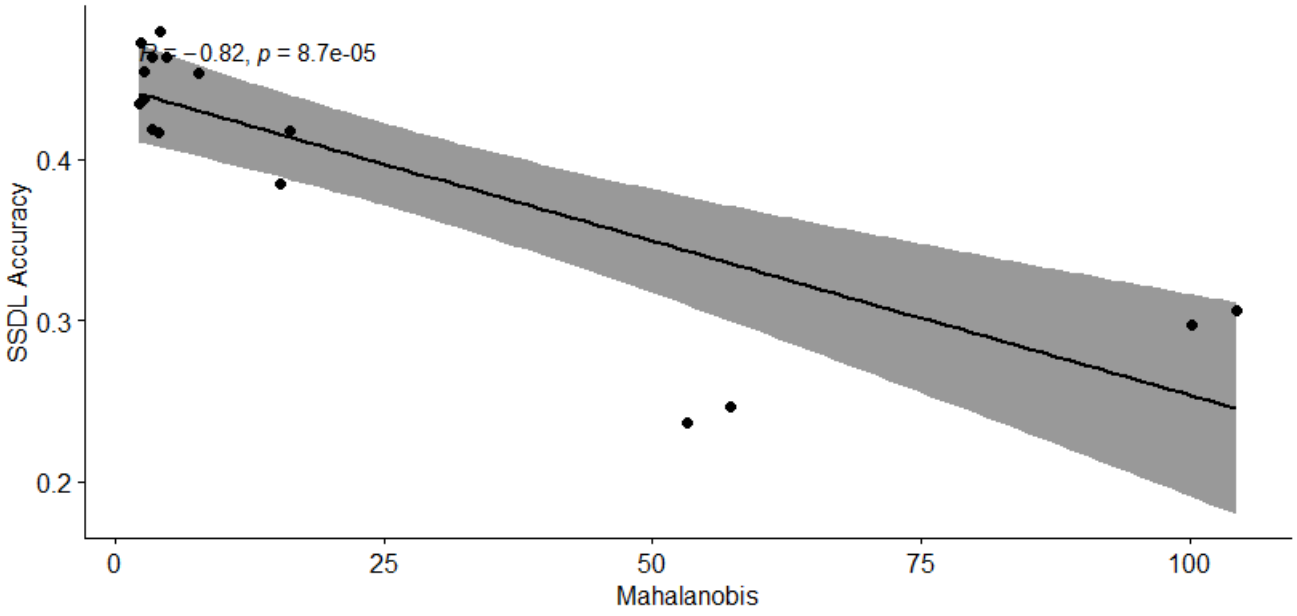

Figure 4.3: CIFAR-10, Mahalanobis results vs Accuracy results. The plot represents the CIFAR-10 Mahalanobis results versus the accuracy reported in [1]. The plot illustrates a positive relation between variables because it shows an uphill pattern. The plot represents the highest Pearson correlation value obtained during the experimentation.

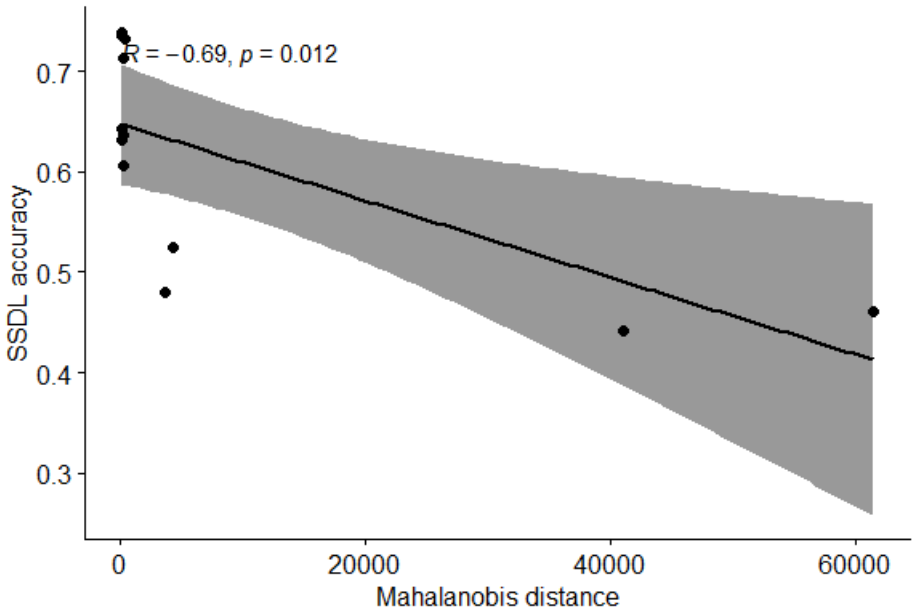

Figure 4.4: The graph depicts the correlation between MNIST Mahalanobis results and the accuracy documented in the reference [1]. The graph demonstrates a direct relationship between the variables as it displays a down x trend.

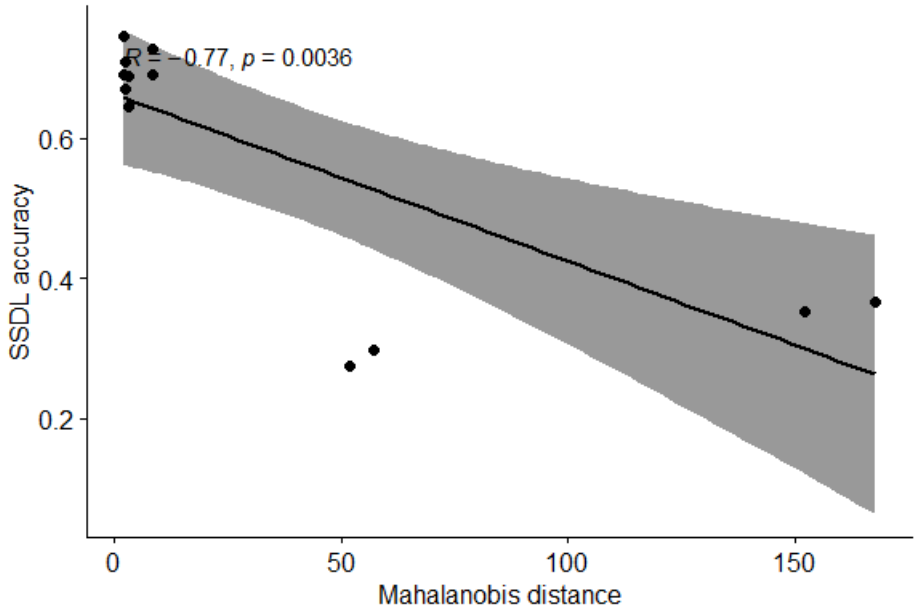

Figure 4.5: The graph presents the correlation between FashionMNIST Mahalanobis results and the accuracy recorded in the reference [1]. The plot illustrates a direct relationship between the variables, demonstrating a consistent downward trend.

As part of the validation of the experiments, a statistical significance test is required. Before selecting the appropriate test, a homogeneity test was performed using Bartlett's K-squared. The results of the test are reported in table 4.14.

The results indicate a significant difference in the variances of the groups compared, with a higher value providing stronger evidence that the variances are not equal.

The results of Bartlett's test indicate a violation of the homogeneity of variances assumption, which is crucial for the analysis of variance (ANOVA). Welch's ANOVA can be used as an alternative, as it does not assume equal variances. Welch's ANOVA is more robust in situations when variances are unequal [67].

Table 4.17 represents the Welsh variance analysis of the processing times between the proposed method and the histogram based on density distance. The goal is to assess if there exists a significant difference in processing times. The null hypothesis H_n posits that both times are similar, while the alternative hypothesis H_i suggests that they are different.

Test of homogeneity
Mahalanobis versus histogram times
Bartlett's K-squared = 1831.3, df = 1, p-value <2.2e-16
Frobenius versus histogram times
Bartlett's K-squared = 2578, df = 1, p-value <2.2e-16

Table 4.14: Test of homogeneity between Mahalanobis, Frobenius and Histogram distance compute times.

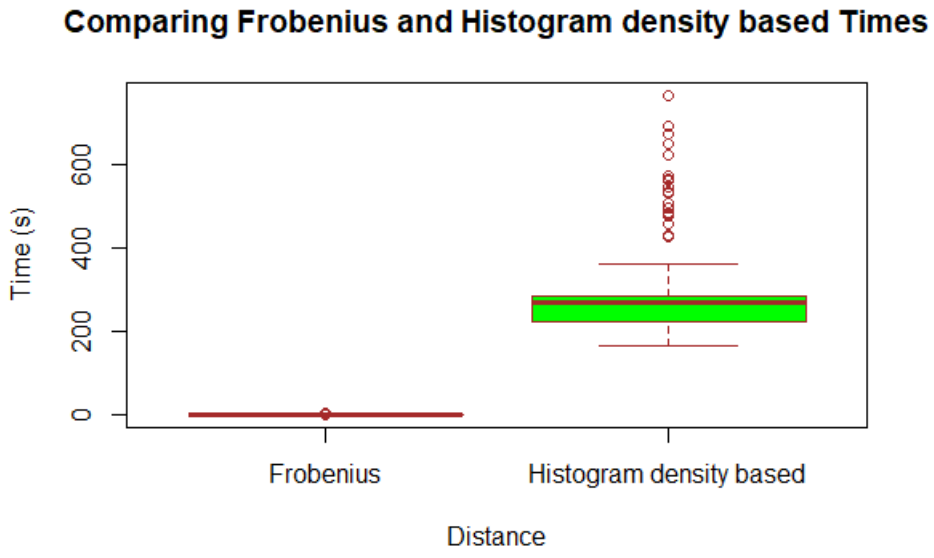

Figure 4.6: Representation of the variances of Frobenius times versus Histogram times. This visualization helps analyze how to demonstrate whether the times are significant or not. The plot illustrates there is significant time difference between both distances.

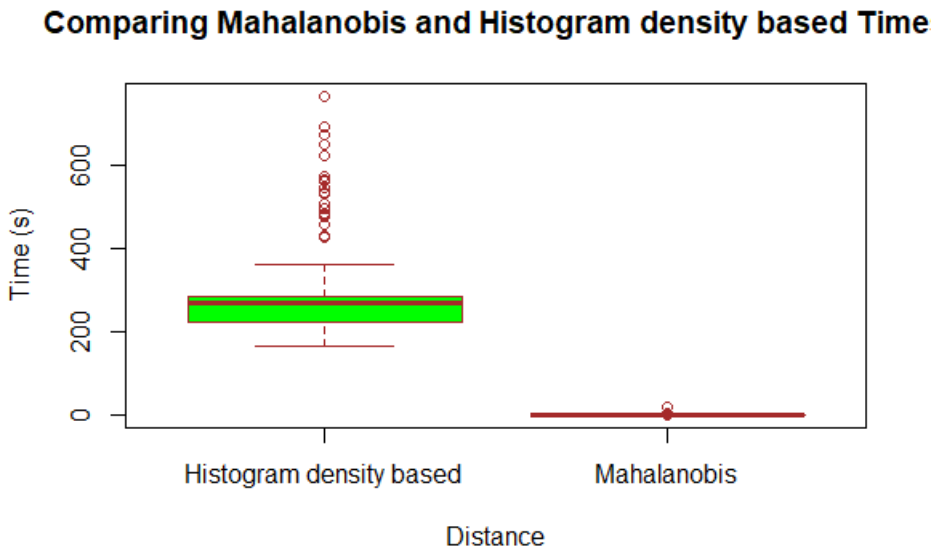

Figure 4.7: Representation of the variances of Mahalanobis versus Histogram times. This visualization helps analyze how to demonstrate whether the times are significant or not.

The results in Table 4.17 indicate that the null hypothesis is rejected, suggesting a significant difference between the two processing times. Welsh ANOVA was employed due to variance heterogeneity test. The reported p-values are both less than $< 2.2e - 16$ for Mahalanobis and Frobenius, indicating highly significant differences.

In addition to the significance test, analyzing the time complexity of each function is crucial to determine the most efficient method.

The dimensionality of the data influences the Mahalanobis distance, which in turn affects the time complexity. This distance is calculated by subtracting two mean vectors, which takes time $\mathcal{O}(n)$, where n is the dimension of the vector. The time complexity for inverting the covariance matrix is $\mathcal{O}(n^3)$, and the time for matrix multiplications is $\mathcal{O}(n^2)$. Based on this analysis, we can conclude that the overall time complexity for calculating the Mahalanobis distance is $\mathcal{O}(n^3)$.

The time complexity of the Frobenius distance is scaled by the size of the matrix, specifically $\mathcal{O}(mn)$, where m is the number of rows and n is the number of columns in the matrix. For matrices of reasonable size, the Frobenius distance

calculation is efficient. However, the computation time will increase significantly if the matrices are large.

The time complexity of the authors' method in [2] and [1] is determined by the growth rate of the histogram and the calculation of distances using Jensen-Shannon, and cosine diversity. Each of these functions will be analyzed in detail.

The time complexity of creating a histogram is influenced by both the number of bins and the number of elements in the dataset. It is typically represented as $\mathcal{O}(mn)$, where m represents the number of data elements and n indicates the number of bins. As a result, the time complexity increases as the number of bins n grows.

The computable time for the density distance of the Jensen-Shannon divergence is directly related to the number of elements n in the Kullback-Leibler (KL) divergence. This calculation iterates through each of the n elements for each dimension. The big O notation for this process is $\mathcal{O}(n)$; however, if the number of dimensions increases, the time complexity will also increase.

The calculation for cosine distance is similar to Jensen-Shannon divergence time complexity, as it is also directly related to the number of elements n in the vectors. This involves n multiplications for the product, as well as summing the squares (also n operations) and calculating square roots. In conclusion, the time complexity for computing cosine distance is $\mathcal{O}(n)$.

In conclusion, Jensen-Shannon Divergence and Cosine distances generally offer better time complexity for large datasets or when dealing with high-dimensional data. In contrast, Mahalanobis and Frobenius distances are more efficient for low-dimensional data, especially when a precomputed covariance matrix is available. This makes them suitable for smaller datasets with high dimensionality and a known distribution.

The proposed method increases the time for each batch during the execution of the experiments, but the reported time is still significantly shorter than the time reported for the same number of batches using the density distances.

In the second part of this phase of experimentation, focusing on COVID-19, the observed trend holds true. The Mahalanobis results detailed in table 4.18 illustrate that higher contamination leads to lower accuracy, while lower contamination results in higher accuracy, aligning with expectations. Although the differences between results diminish as contamination decreases, the trend remains consistent.

The table 4.19 echo those of previous experiments, indicating that the cosine distance reported by the authors in [2] serves as a superior predictor compared to the Mahalanobis distance. Notably, datasets with 20 labeled observations exhibit the highest values in both methods, highlighting a consistent trend. The Pearson coefficient value reported in 4.18 is a higher negative value compared to the values reported in the table mentioned earlier. This suggests that the Mahalanobis method can still be considered a reliable predictor.

The results reported in 4.20 demonstrated that the trend about the time is still consistent with the COVID-19 experiments. Mahalanobis results are still significant lower and different as the time reported in [2].

S_{IOD}	T_{OOD}	S_{uOOD}	$\%_{uOOD}$	n_l				Pearson Correlation
				60		100		
				d_F	Accuracy	d_F	Accuracy	
MNIST	Dif	TI	50	0,000431	0.642 ± 0.094	0,000385	0.739 ± 0.074	
			100	0,000646	0.637 ± 0.097	0,000568	0.732 ± 0.074	
		GN	50	0,000505	0.606 ± 0.0989	0,000499	0.713 ± 0.087	
			100	0,000709	0.442 ± 0.099	0,000684	0.461 ± 0.073	
		SAPN	50	0,000355	0.631 ± 0.102	0,000344	0.735 ± 0.082	
			100	0,000493	0.48 ± 0.0951	0,000497	0.524 ± 0.09	
								-0,59
CIFAR-10	Sim	TI	50	0,000303	0.435 ± 0.054	0,000285	0.473 ± 0.039	
			100	0,000348	0.417 ± 0.020	0,000357	$\mathbf{0.480 \pm 0.039}$	
	Dif	SVHN	50	0,000219	0.419 ± 0.027	0,000263	0.464 ± 0.044	
			100	0,000263	0.385 ± 0.034	0,000237	0.418 ± 0.035	
		GN	50	0,000590	0.409 ± 0.047	0,000195	0.454 ± 0.048	
			100	0,000168	0.297 ± 0.029	0,000181	0.306 ± 0.034	
		SAPN	50	0,000785	0.438 ± 0.029	0,000836	0.455 ± 0.037	
			100	0,001122	0.236 ± 0.031	0,001120	0.246 ± 0.032	
								-0,462
FashionMNIST	Dif	TI	50		0.690 ± 0.065		$\mathbf{0.745 \pm 0.093}$	
			100		0.690 ± 0.073		0.728 ± 0.066	
		GN	50		0.644 ± 0.061		0.689 ± 0.075	
			100		0.352 ± 0.025		0.366 ± 0.065	
		SAPN	50		0.671 ± 0.072		0.708 ± 0.095	
			100		0.276 ± 0.069		0.297 ± 0.046	
								-0,767

Table 4.15: Second Experiment Frobenius Distance vs Accuracy. This table reflects the Frobenius results versus the accuracy reported in [1]. It illustrates the pattern is reflected in most of the experiments but not when the contamination is with Gaussian Noise.

$\mathbf{S_l}$	$\mathbf{d_M}$	$\mathbf{d_F}$	$\mathbf{d_{JS}}$
MNIST	-0,6191	-0,5959	**-0,8487**
CIFAR-10	-0,8239	-0,4624	**-0,9414**
FashionMNIST	-0,7674	-0,62417	**-0,9766**

Table 4.16: Pearson coefficient of the results between 4.12, 4.15 and DeDiMs distance d_{js} accuracy results. This table compares the Pearson correlation results obtained from the proposed method with the Pearson correlation results of the accuracy values reported in [1]. The results demonstrate that the density distance executed with a histogram is a better predictor than the method proposed in this investigation.

Histogram density based times vs Mahalanobis and Frobenius times Analysis

Distances		df	F	P-value	denom df
Histogram density distance time (s) d_{js}	Mahalanobis distance times (s)	1.00	1817.6	$< 2.2e-16$	237.05
Histogram density distance time (s) d_{js}	Frobenius distance time (s)	1	1815.3	$< 2.2e-16$	237

Table 4.17: Variance analysis between times calculated by the distances. The Welsh variance analysis does not assume equal variances. The p-values indicate a significant difference between the groups, suggesting that the results are not similar and not due to chance.

	$n_l = 40$		$n_l = 20$	
Dataset	d_M	Acc. FD	d_M	Acc. FD
ChestX-ray8 35% - Costa Rica 65%	2,221	0.709 ± 0.084	2,748	0.682 ± 0.09
ChestX-ray8 65% - Costa Rica 35%	2,367	0.732 ± 0.064	2,396	0.717 ± 0.08
China 35% - Costa Rica 65%	2,767	0.683 ± 0.065	2,929	0.667 ± 0.078
China 65% - Costa Rica 35%	3,005	0.693 ± 0.044	3,168	0.687 ± 0.078
Indiana 35% - Costa Rica 65%	1,876	0.732 ± 0.052	2,058	0.703 ± 0.1
Indiana 65% - Costa Rica 35%	1,582	0.719 ± 0.058	1,595	0.709 ± 0.093
Pearson Correlation	**-0,7553408**			

Table 4.18: Covid-19 Experiments. Mahalanobis distance vs accuracy reported in [2]. The table illustrates the results and Pearson value between the variables. The experiment observed the expected consistency.

Distance	n_l	**Pearson coefficient**
DeDiMs Cosine Distance	20	-0.798
	40	-0.75
Mahalanobis Distance	20	-0,713
	40	-0,839

Table 4.19: COVID-19 Pearson Coefficient correlation between Cosine Distance Alexnet [2] model versus Mahalanobis distance. The results demonstrate that the Mahalanobis is a better predictor than the Cosine distance results.

Distance	**Time (s)**
DeDiMs Cosine Distance reported in	269.7
Histogram Distance d_{js}	189,5
Mahalanobis Distance	2,6

Table 4.20: COVID-19 Compute time in seconds.The table illustrates the mean time in seconds reported in the histogram density distance versus the Mahalanobis distance.

5. Conclusions

5.1 Conclusions

The objective of this investigation was to evaluate the quality of labeled and unlabeled datasets intended for use in SSLM. Our primary focus was to implement the proposed method in a medical context, addressing the challenges of obtaining accurately labeled data for predictive models.

The quality of data is crucial in understanding its impact on SSLM performance. Determining whether the data will enhance or hinder performance is essential, yet there is currently limited research in this area.

Studies on data quality often focus on cleaning datasets, removing noise, generating metrics during the training process, and typically target only labeled datasets. This approach adds extra costs in terms of time and resources during training, which could have been avoided with pre-processing measures. This study aims to introduce the topic and propose a potential solution applicable to medical images that is not time or resource-intensive.

As data sources continue to proliferate, whether closed or open, it becomes more and more crucial to efficiently assess the practicality and effectiveness of utilizing a particular mix of data (both labeled and unlabeled datasets in the realm of SSDL).

As detailed in the experimental results chapter, the proposed method employs two different distances: Mahalanobis and Frobenius. Both distances were tested using various datasets with different levels of contamination, as well as COVID datasets, to evaluate their behavior in medical images.

The experiments demonstrated that both distances produced the expected results, with the Mahalanobis distance consistently yielding more reliable outcomes. This conclusion is drawn from the results presented in Tables 4.12 and 4.18. The findings indicate a consistent trend across all the experiments conducted using the Mahalanobis distance. When contamination was at 50%, the

values were relatively smaller compared to when contamination was at 100%.

The experiment involving COVID X-ray images produced similar and smaller values and minimal differences between the contaminated datasets. The contamination involved datasets containing the identical type of images (X-ray) but from different locations or resources. The aim was to assess whether the results would show a difference when the contamination originated from a similar source. The experiment demonstrated that the trend is consistent: greater distribution mismatch leads to lower SSDL accuracy, resulting in elevated distance values.

The COVID-19 experiment was carried out using the Mahalanobis distance method, as the Frobenius distance did not generate consistent results. The Frobenius results did show a similar trend, but they were not consistent and contradictory in some experiments.

The experimental design compared the proposed method with DeDiMs distances, a demonstrated approach to assessing dataset quality. The correlation results showed a positive correlation between the distances generated by the proposed method and those generated by DeDiMs and the histogram distance. This conclusion is consistent with the trends reported by the authors in [1] and [2].

The correlation between the proposed method and the accuracy values reported in [1] and [2] resulted in a strong negative correlation. This means that as accuracy decreases, the distance result and contamination increase. Conversely, as accuracy increases, the distance result and contamination decrease. These results indicate that when the unlabeled dataset contaminates the labeled dataset, the proposed method will generate a significant distance value, that can be used as a metric to determine whether the unlabeled data set will affect the accuracy of SSLM.

Based on the preliminary conclusions, the proposed method can be utilized as a predictor of how a model SSLM will behave. The proposed distances can be introduced as a process in the Data Cleaning mentioned by the authors in [64].

The proposed Mahalanobis or Frobenius distance can be introduced as a metric to evaluate the relevancy, unbiasedness, and self-consistency dimensions of the datasets. This conclusion aims to integrate the data quality dimensions

and apply them in the context of SSLM.

Both proposed metrics yielded a compute time that is more than two times faster than the time reported by the authors in [1] and [2].

Utilizing the suggested measurement could reduce the computational resources necessary for the model training process. The reduction of energy consumption highlighted in [68] is increasingly important in order to decrease the carbon emissions of computing grids. Concerns about the environmental impact of Artificial Intelligence (AI) models were brought up in [68], emphasizing that model training requires a significant amount of power, resulting in a larger carbon footprint. The authors stressed that integrating these models into everyday life will lead to increased hardware and energy requirements. Assessing dataset compatibility using small samples of labeled and unlabeled datasets can prevent unnecessary model training.

The results demonstrated that the distances proposed by the authors in [1] and [2] behave as similar predictors compared to the proposed method. This conclusion is based on the Pearson results reported in 4.16. However, these distances also consume more resources compared to the proposed metrics. In real-life implementations, resource consumption is critical for feasible use.

The tests showed that the Mahalanobis distance between the embedding densities of the labeled and unlabeled datasets consistently produces reliable results in terms of the reported accuracy in [1], even when faced with various scenarios of distribution mismatch.

For future investigations, it is recommended to extend the implementation of the proposed method beyond sampled data sets to include complete data sets for improved precision in the results. To determine its effectiveness, the proposed method should be expanded to encompass other learning methodologies such as active learning, transfer learning, etc. Although the proposed method was successfully applied to COVID-19 datasets, its applicability should be tested across a wider range of medical image datasets to further validate and generalize its effectiveness.

A. Appendix

A.1 Appendix

Dataset	$\mathbf{d_M}$	DeDiM distance
ChestX-ray8 65% - Costa Rica 35%	$2,381$	3.95 ± 0.94
ChestX-ray8 35% - Costa Rica 65%	$2,484$	11.84 ± 0.94
China 65% - Costa Rica 35%	$2,848$	5.74 ± 0.79
China 35% - Costa Rica 65%	$3,087$	14.85 ± 0.0
Indiana 65% - Costa Rica 35%	$1,588$	6.33 ± 0.3
Indiana 35% - Costa Rica 65%	$1,967$	16.61 ± 0.3
Pearson Correlation		**0,1237**

Table A.1: COVID-19 Experiment Mahalanobis versis DeDiMs Cosine Distance

S_{IOD}	T_{OOD}	S_{uOOD}	$\%_{uOOD}$	Distances	
				d_F	**Histograma**
MNIST	Dif	TI	50	0,000408	3,663
			100	0,000568	10,305
		GN	50	0,000502	14,785
			100	0,000697	52,349
		SAPN	50	0,000349	15,116
			100	0,000495	53,397
		Pearson Correlation		**0,570**	
CIFAR-10	Sim	TI	50	0,000294	0,388
			100	0,0003529	0,469
	Dif	SVHN	50	0,0002226	1,296
			100	0,0002506	1,277
		GN	50	**0,0003931**	1,708
			100	**0,0001749**	5,855
		SAPN	50	0,0008112	2,299
			100	0,0011217	8,225
		Pearson Correlation		**0,611**	
FashionMNIST	Dif	TI	50		0,897
			100		1,4
		GN	50		2,819
			100		9,042
		SAPN	50		2,799
			100		8,464
		Pearson Correlation			

Table A.2: Frobenius vrs Distance JS results (Histogram). This table illustrates the distance results of the Frobenius experiments versus the D_{JS} results.

$\mathbf{S_{IOD}}$	$\mathbf{T_{OOD}}$	$\mathbf{S_{uOOD}}$	$\%_{\mathbf{uOOD}}$	Distances	
				d_M	**Histogram Density Distance** d_{js}
MNIST	Dif	TI	50	2,465	3,663
			100	14,407	10,305
		GN	50	3,318	14,785
			100	216,134	52,349
		SAPN	50	2,526	15,116
			100	61,631	53,397
		Pearson Correlation			**0,7903**
CIFAR-10	Sim	TI	50	2,276	0,388
			100	4,130	0,469
	Dif	SVHN	50	1,354	1,296
			100	1,385	1,277
		GN	50	6,241	1,708
			100	102,280	5,855
		SAPN	50	2,610	2,299
			100	55,169	8,225
		Pearson Correlation			**0,805**
FashionMNIST	Dif	TI	50	2,221	0,897
			100	8,492	1,4
		GN	50	3,208	2,819
			100	159,800	9,042
		SAPN	50	2,394	2,799
			100	54,491	8,464
		Pearson Correlation			**0,851**

Table A.3: 2nd Experiment Mahalanobis Distance vrs Histogram Distance. This table illustrates the distance results of the Mahalanobis experiments versus the D_{JS} results. This table reflects that the trend is consistent

References

[1] S. Calderón Ramírez, L. Oala, J. Torrentes-Barrena, D. Elizondo, A. Moemeni, S. Colreavy-Donnelly, W. Samek, M. A. Molina-Cabello, and E. López-Rubio, "Dataset similarity to assess semi-supervised learning under distribution mismatch between the labelled and unlabelled datasets," *IEEE Transactions on Artificial Intelligence*, vol. PP, pp. 1–1, 01 2022.

[2] S. Calderon-Ramirez, S. Yang, D. Elizondo, and A. Moemeni, "Dealing with distribution mismatch in semi-supervised deep learning for covid-19 detection using chest x-ray images: A novel approach using feature densities," *arXiv preprint arXiv:2109.00889*, 2021.

[3] E. Röösli, B. Rice, and T. Hernandez-Boussard, "Bias at warp speed: how AI may contribute to the disparities gap in the time of COVID-19," *Journal of the American Medical Informatics Association*, vol. 28, pp. 190–192, 08 2020.

[4] A. S. Becker, M. Marcon, S. Ghafoor, M. C. Wurnig, T. Frauenfelder, and A. Boss, "Bias at warp speed: how AI may contribute to the disparities gap in the time of COVID-19," *Deep Learning in Mammography, Investigative Radiology*, vol. 52, pp. 434–440, 07 2017.

[5] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data lifecycle challenges in production machine learning: A survey," *SIGMOD Rec.*, vol. 47, p. 17–28, Dec. 2018.

[6] E. Team, "What is Machine Learning? A Definition.," 05 2020.

[7] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR 2011*, pp. 1521–1528, IEEE, 2011.

[8] A. Asma Chebli and Hayet Farida Marouani, "Semi-supervised learning for medical application : A survey," in *2018 International Conference on Applied Smart Systems (ICASS'2018)*, November.

[9] A. Safonova, G. Ghazaryan, S. Stiller, M. Main-Knorn, C. Nendel, and M. Ryo, "Ten deep learning techniques to address small data problems with remote sensing," *International Journal of Applied Earth Observation and Geoinformation*, vol. 125, p. 103569, 2023.

[10] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *arXiv preprint arXiv:2001.07685*, 2020.

[11] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *arXiv preprint arXiv:1905.02249*, 2019.

[12] S. Calderon-Ramirez, S. Yang, A. Moemeni, D. Elizondo, S. Colreavy-Donnelly, L. F. Chavarría-Estrada, and M. A. Molina-Cabello, "Correcting data imbalance for semi-supervised covid-19 detection using x-ray chest images," *Applied Soft Computing*, vol. 111, p. 107692, 2021.

[13] S. Calderón Ramírez, L. Oala, J. Torrents-Barrena, A. Moemeni, W. Samek, and M. A. Molina-Cabello, "Mixmood: A systematic approach to class distribution mismatch in semi-supervised learning using deep dataset dissimilarity measures," 06 2020.

[14] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Communications of the ACM*, vol. 45, no. 4, pp. 211–218, 2002.

[15] A. Z. Ghalwash, N. E. Khameesy, D. A. Magdi, and A. Joshi, *Data Quality Dimensions*. 2019.

[16] V. Gudivada, A. Apon, and J. Ding, "Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations," *International Journal on Advances in Software*, vol. 10, pp. 1–20, 07 2017.

[17] J. Tanha, "A selection metric for semi-supervised learning based on neighborhood construction," *Information Processing and Management*, vol. 58, pp. 2021, 102444, 12 2020.

[18] J. Sun, W.-T. He, L. Wang, A. Lai, X. Ji, X. Zhai, G. Li, M. A. Suchard, J. Tian, J. Zhou, *et al.*, "Covid-19: epidemiology, evolution, and cross-disciplinary perspectives," *Trends in molecular medicine*, vol. 26, no. 5, pp. 483–495, 2020.

[19] W. Zhou and W. Wang, "Fast-spreading sars-cov-2 variants: challenges to and new design strategies of covid-19 vaccines," *Signal Transduction and Targeted Therapy*, vol. 6, no. 1, pp. 1–6, 2021.

[20] M. Méndez, S. Calderon, P. Tyrrell, and S. Calderón Ramírez, "Using cluster analysis to assess the impact of dataset heterogeneity on deep convolutional network accuracy: A first glance," 10 2019.

[21] Y. Ian Goodfellow and Aaron Courville, "Deep learning," in *Deep learning*, October 2017.

[22] T. Qin, *Machine Learning Basics*, pp. 11–23. Singapore: Springer Singapore, 2020.

[23] J. Savoy, *Machine Learning Models*, pp. 109–151. Cham: Springer International Publishing.

[24] Jesper E. van Engelen and Holger H. Hoos, "A survey on semi-supervised learning," in *2018 International Conference on Applied Smart Systems (ICASS'2018)*, November 20190.

[25] I. S. Masoud Nikravesh and Lofti A. Zadeh. Warsaw, Poland: Springer.

[26] Shorten Connor and Khoshgoftaar Taghi M., "A survey on image data augmentation for deep learning," in *Journal of Big Data*, July 2019.

[27] D. A. van Dyk and X.-L. Meng, "The art of data augmentation," *Journal of Computational and Graphical Statistics*, vol. 10, no. 1, pp. 1–50, 2001.

[28] S.-C. Wang, *Artificial Neural Network*, pp. 81–100. Boston, MA: Springer US, 2003.

[29] W. Mcculloch and W. Pitts, "A logical calculus of ideas immanent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 127–147, 1943.

[30] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65 6, pp. 386–408, 1958.

[31] H. D. Block, "The perceptron: A model for brain functioning. i," *Rev. Mod. Phys.*, vol. 34, pp. 123–135, Jan 1962.

[32] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5, pp. 183–197, 1991.

[33] K. Y. Chan, B. Abu-Salih, R. Qaddoura, A. M. Al-Zoubi, V. Palade, D.-S. Pham, J. D. Ser, and K. Muhammad, "Deep neural networks in the cloud: Review, applications, challenges and research directions," *Neurocomputing*, vol. 545, p. 126327, 2023.

[34] E. Xing, M. Jordan, S. J. Russell, and A. Ng, "Distance metric learning with application to clustering with side-information," in *Advances in Neural Information Processing Systems* (S. Becker, S. Thrun, and K. Obermayer, eds.), vol. 15, MIT Press, 2002.

[35] S. Xiang, F. Nie, and C. Zhang, "Learning a mahalanobis distance metric for data clustering and classification," *Pattern Recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.

[36] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," *Michigan State Universiy*, vol. 2, no. 2, p. 4, 2006.

[37] "Reprint of: Mahalanobis, p.c. (1936) "on the generalised distance in statistics."," *Sankhya Ser. A*, vol. 80, pp. 1–7, Dec. 2018.

[38] G. J. McLachlan, "Mahalanobis distance," *Resonance*, vol. 4, no. 6, pp. 20–26, 1999.

[39] I. L. Dryden, A. Koloydenko, and D. Zhou, "Non-euclidean statistics for covariance matrices, with applications to diffusion tensor imaging," *The Annals of Applied Statistics*, vol. 3, no. 3, pp. 1102–1123, 2009.

[40] R. Biscay, L. M. Rodríguez, and E. Díaz-Frances, "Cross-validation of covariance structures using the frobenius matrix distance as a discrepancy function," *Journal of Statistical Computation and Simulation*, vol. 58, no. 3, pp. 195–215, 1997.

[41] O. Skean, A. Dhakal, N. Jacobs, and L. G. S. Giraldo, "Frossl: Frobenius norm minimization for self-supervised learning," *arXiv preprint arXiv:2310.02903*, 2023.

[42] M. D. Malkauthekar, "Analysis of euclidean distance and manhattan distance measure in face recognition," in *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*, pp. 503–507, 2013.

[43] A. A. Thant, S. M. Aye, and M. Mandalay, "Euclidean, manhattan and minkowski distance methods for clustering algorithms," *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 7, no. 3, pp. 553–559, 2020.

[44] A. Singh, A. Yadav, and A. Rana, "K-means with three different distance metrics," *International Journal of Computer Applications*, vol. 67, no. 10, 2013.

[45] W. Förstner and B. Moonen, "A metric for covariance matrices," in *Geodesy-the Challenge of the 3rd Millennium*, pp. 299–309, Springer, 2003.

[46] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *Computer Vision – ECCV 2006* (A. Leonardis, H. Bischof, and A. Pinz, eds.), (Berlin, Heidelberg), pp. 589–600, Springer Berlin Heidelberg, 2006.

[47] V. Kaushal, S. Kothawade, G. Ramakrishnan, J. A. Bilmes, and R. K. Iyer, "PRISM: A unified framework of parameterized submodular information measures for targeted data subset selection and summarization," *CoRR*, vol. abs/2103.00128, 2021.

[48] D. Blackwell, "Conditional expectation and unbiased sequential estimation," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 105–110, 1947.

[49] S. Date, "3 conditionals every data scientist should know," Nov 2020.

[50] *Entropy, Relative Entropy, and Mutual Information*, ch. 2, pp. 13–55. John Wiley and Sons, Ltd, 2005.

[51] Y. Guo and R. Greiner, "Optimistic active-learning using mutual information.," pp. 823–829, 01 2007.

[52] D. C. Montgomery, *Desing and analisys of Experiments*.

[53] H. J. Seltman, *Experimental Design and Analysis*. 2018.

[54] V. J. Easton and J. H, "Experimentation - Statistics Glossary."

[55] E. Tiu, "Understanding Latent Space in Machine Learning," 2020.

[56] F. Sidi, P. H. S. Panahy, L. S. Affendey, M. A. Jabar, H. Ibrahim, and A. Mustapha, "Data quality: A survey of data quality dimensions," in *2012 International Conference on Information Retrieval & Knowledge Management*, pp. 300–304, IEEE, 2012.

[57] S. Cicek, A. Fawzi, and S. Soatto, "Saas: Speed as a supervisor for semi-supervised learning," *CoRR*, vol. abs/1805.00980, 2018.

[58] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *Proceedings of the european conference on computer vision (eccv)*, pp. 135–152, 2018.

[59] N. Prat and S. Madnick, "Measuring data believability: A provenance approach," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008)*, pp. 393–393, 2008.

[60] P. Schmidt and F. Biessmann, "Quantifying interpretability and trust in machine learning systems," 01 2019.

[61] Y. Liu, Y. Wang, L. Gao, C. Guo, Y. Xie, and Z. Xiao, "Deep hash-based relevance-aware data quality assessment for image dark data," *ACM/IMS Trans. Data Sci.*, vol. 2, apr 2021.

[62] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in *Brazilian symposium on artificial intelligence*, pp. 286–295, Springer, 2004.

[63] Y. Gong, G. Liu, Y. Xue, R. Li, and L. Meng, "A survey on dataset quality in machine learning," *Information and Software Technology*, vol. 162, p. 107268, 2023.

[64] S. E. Whang, Y. Roh, H. Song, and J.-G. Lee, "Data collection and quality challenges in deep learning: a data-centric ai perspective," *The VLDB Journal*, vol. 32, p. 791–813, jan 2023.

[65] A. Sanyal, V. Chatterji, N. Vyas, B. Epstein, N. Demir, and A. Corletti, "Fix your models by fixing your datasets," 2021.

[66] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Q. Duong, and M. Ghassemi, "Covid-19 image data collection: Prospective predictions are the future," *arXiv 2006.11988*, 2020.

[67] R. Kohr and P. Games, "Robustness of the analysis of variance, the welch procedure and a box procedure to heterogeneous variances," *The Journal of Experimental Education*, vol. 43, pp. 61–69, 04 2014.

[68] G. Li, X. Deng, Z. Gao, and F. Chen, "Analysis on ethical problems of artificial intelligence technology," in *Proceedings of the 2019 International Conference on Modern Educational Technology*, ICMET 2019, (New York, NY, USA), p. 101–105, Association for Computing Machinery, 2019.