UNIVERSITAT DE BARCELONA
Facultat de Matemàtiques i Informàtica

Integrating topological features to enhance cardiac disease diagnosis from 3D CMR images

Author: Marina Anguas Escobar

Mathematics director: Dr. Carles Casacuberta
Computer science director: Dr. Polyxeni Gkontra
Performed at: Mathematics and Computer Science Department

Barcelona, June 2023

Abstract

Persistent homology is a technique from the field of algebraic topology for the analysis and characterization of the shape and structure of datasets in multiple dimensions. Its use is based on the identification and quantification of topological patterns in the dataset across various scales. In this thesis, persistent homology is applied with the objective of extracting topological descriptors from three-dimensional cardiovascular magnetic resonance (CMR) imaging. Thereafter, topological descriptors are used for the detection of cardiovascular diseases by means of Machine Learning (ML) techniques.

Radiomics has been one of the recently proposed approaches for disease diagnosis. This method involves the extraction and subsequent analysis of a significant number of quantitative descriptors from medical images. These descriptors offer a characterization of the spatial distribution, texture, and intensity of the structures present in the images.

This study demonstrates that radiomics and topological descriptors achieve comparable results, providing complementary insights into the underlying structures and characteristics of anatomical tissues. Moreover, the combination of these two methods leads to a further improvement of the performance of ML models, thereby enhancing medical diagnosis.

2020 Mathematics Subject Classification. 00A69

Resum

L'homologia persistent és una tècnica del camp de la topologia algebraica que permet l'anàlisi i la descripció de la forma i l'estructura de conjunts de dades en dimensions arbitràries. El seu ús es fonamenta en la identificació i la quantificació de patrons topològics que persisteixen en els conjunts de dades al llarg de diferents escales.

En aquest estudi s'aplica l'homologia persistent amb l'objectiu d'extreure descriptors topològics de ressonàncies magnètiques cardiovasculars en dimensió 3. Aquests descriptors topològics seran utilitzats per a detectar malalties cardiovasculars mitjançant tècniques de *Machine Learning*.

L'ús dels *radiomics* és un dels procediments proposats en els darrers anys per resoldre aquest problema. Aquest mètode consisteix en l'extracció i posterior anàlisi d'un nombre elevat de descriptors quantitatius d'imatges mèdiques. Aquests descriptors ofereixen una caracterització de la distribució espacial, la textura i la intensitat de les estructures presents en les imatges.

En aquest estudi, es demostra que els *radiomics* i els descriptors topològics proporcionen resultats comparables, tot i que fan referència a característiques diferents dels teixits i de les estructures anatòmiques. Mentre que els *radiomics* se centren en la quantificació de la forma, la topologia fa referència a la seva textura. També es demostra que la combinació d'aquests dos mètodes aconsegueix augmentar les seves mètriques d'avaluació individuals i, per tant, millorar el diagnòstic mèdic.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my supervisors, Polyxeni Gkontra and Carles Casacuberta, for their guidance and support throughout the research process. Their expertise and insights were invaluable in shaping this study and helping me to overcome challenges.

I am especially indebted to Aina Ferrà, who serves as an inspiration to me not only in the fields of mathematics and computer science but also as an exceptional woman. I am immensely appreciative of your invaluable help in directing my thesis and the significant support you have offered me in the realm of programming.

I cannot fail to mention Rubén Ballester's willingness to assist me and his evident enthusiasm for topology.

To my friends, thank you for all the adventures we have experienced together and for all the little things we have shared. You are like family to me.

I cannot forget to mention my friend Víctor. It is such a pleasure to end this stage of life next to you.

Pol, I cannot thank you enough for staying by my side since we sat together during the first week of classes; it would not have been the same without you.

I would also like to express my gratitude to my grandparents, for whom I feel a deep admiration.

Finally, I would like to thank my parents and my sister, whose love and guidance are with me in whatever I pursue.

Contents

1 Introduction

1.1 Motivation

The range of diseases that affect the heart and blood vessels, which are referred to as cardiovascular diseases, have been categorized as the most common cause of morbidity and mortality worldwide [13]. To overcome the challenges posed by these diseases, early and accurate diagnosis is essential [14].

In recent years, there has been a notable surge in the adoption of Artificial Intelligence (AI) in the medical field. One prominent area of application is cardiovascular imaging [29], including cardiovascular magnetic resonance (CMR) imaging, which is considered the reference modality for evaluating heart structure and function. This is primarily due to the large amount of data generated by modern imaging systems that makes manual assessment of CMR a laborious, time-consuming and expensive process. At the same time, Machine Learning (ML), a subfield of AI, holds tremendous potential for automated CMR-based diagnosis by leveraging past observations to uncover hidden and complex patterns that may otherwise go unnoticed. Thereby, efficiency, accuracy, and cost-effectiveness of diagnostic procedures are improved [12], [19], [30].

Topology is a branch of mathematics that concerns the study of the properties and characteristics of geometric spaces which are preserved under continuous transformations. In applied mathematics, topological data analysis (TDA) is an analytical approach that applies concepts from topology to explore discrete data. Dealing with high-dimensional, incomplete and noisy data can be difficult, as traditional methods may struggle to extract meaningful information [2], [16]. TDA offers a versatile framework to analyze such data, overcoming the limitations of specific metrics and offering advantages like dimensionality reduction and robustness against noise [1]. Overall, TDA allows a more comprehensive understanding of the underlying structures. Furthermore, it is assumed that it could provide medical research with information about the functional features of the systems which are being studied [1].

Traditionally, CMR data can be represented as a collection of coordinates which form a discrete set of points in a 4-dimensional stereotactic space due to the fact that time is included. However, in this study time does not represent an additional dimension because a 3-dimensional image is considered for each of the two different time-points which are end-diastole and end-systole. TDA-based descriptors cannot be extracted directly from this type of structure because it requires a non-discrete space. Therefore, topological signatures are used in order to detect and represent shape features such as connectivity, loops, cavities, flares, or clusters [2].

In contrast with other analytical methods, a main branch of TDA named persistence homology, exhibits invariance to small perturbations, which provides a distinct advantage when considering anatomical applications. Additionally, an important benefit of using TDA is its ability to recover structure in higher dimensions without requiring dimensionality reduction [6].

1.2 Objectives

The aim of this thesis is to improve diagnosis of cardiovascular diseases by leveraging Topological Data Analysis and Machine Learning. More precisely, the main objectives are the following:

- Examine and analyze the structural characteristics derived from images using topological methods, and particularly persistent homology, leading to the development of novel descriptors for assessing cardiac health from CMR images.
- Explore and comprehend the most common machine learning approaches used in image-based diagnosis.
- Combine the aforementioned topology-based CMR descriptors with state-of-the-art machine learning algorithms to improve diagnosis of cardiac diseases using CMR images.
- Enhance the classification outcomes achieved through radiomics by incorporating topological descriptors.

1.3 Contributions

In this work, topological features were extracted from 3-dimensional CMR images for subsequent diagnosis of cardiac diseases by means of Machine Learning, resulting in a novel pipeline for diagnosis. This study also distinguishes itself by the use of persistence descriptors in homological dimension 2 and their significantly better contributions to classification than features in homological dimensions 0 and 1.

The proposed model, combining radiomis and topological features, is tested on a dataset of CMR images from the MICCAI 2017 Automated Cardiac Diagnosis Challenge (ACDC), a challenge which was organized for benchmarking CMR segmentation and classification algorithms [4]. The individual performance of radiomics and TDA-based features is shown to be comparable. Nevertheless, the integration of these methods yields superior accuracy, thereby enhancing medical diagnoses. The results yielded by this research demonstrate encouraging outcomes in the implementation of topological data analysis within the medical domain and suggests prospective avenues for further investigation.

2 Mathematical background

Topological data analysis is a mathematical field which combines techniques from algebraic topology and computational geometry in order to analyze and understand complex datasets. With this aim, topological features that provide insights into shape are extracted. The aforementioned shape features encompass connectivity, loops, cavities, flares or clusters.

For this intention, qualitative and quantitative methods have been developed. However, in this study only quantitative methods by means of persistent homology will be carried out.

The references [6], [20], [21] and [28] greatly contributed to the development of this section.

2.1 Persistent homology

Persistent homology is a feature extraction method which consists of assigning multiscale topological descriptors to sublevel sets $X_\alpha = \{x \in X : f(x) \leq \alpha\}$ of a given real-valued function $f\colon X \to \mathbb{R}$ on a set X.

The function $f : X \to \mathbb{R}$ introduced in the preceding definition is commonly referred to as the *filter function*. The aforementioned sublevel sets are naturally nested, which implies that $X_\alpha \subset X_\beta$ if $\alpha \leq \beta$.

Moreover, this collection of subsets form a *filtration* of X, and persistence records the evolution of the topology of this filtration as a function of α. Further details on filtrations and calculations of their descriptors will be given in subsequent sections.

2.2 Simplicial complexes

Definition 2.1. Given a set of points $S = \{u_0, u_1, \ldots, u_n\}$ in $\mathbb{R}^d$, $\sum_{i=0}^n \lambda_i u_i$ is referred to as an *affine combination* of the set S if $\sum_{i=0}^n \lambda_i = 1$.

The *affine hull* of a given set S is defined as the set of all affine combinations, which will be an n-plane if the $n+1$ points are affinely independent.

A set $S = \{u_0, u_1, \ldots, u_n\}$ is *affinely independent* if for any two identical affine combinations $\sum_{i=0}^n \lambda_i u_i = \sum_{i=0}^n \mu_i u_i$ it follows that $\lambda_i = \mu_i$ for all i.

An affine combination such that $\lambda_i > 0$ for all i is called a *convex combination*. The *convex hull* of a set is the collection of all its convex combinations.

Definition 2.2. A *k-simplex* σ is defined as a convex hull of a set S of $k+1$ affinely independent points.

If σ is the convex hull of $\{u_0, u_1, \ldots u_k\}$, then the points u_i will be referred to as *vertices* of σ, while *edges* are convex hulls of pairs of vertices. The convex hull of a subset $H \subseteq S$ with $H \neq \emptyset$ is called a *face* of the k-simplex σ and it is denoted as $\tau \leq \sigma$. A *proper* face is a face τ satisfying the condition that $H \not\subseteq S$.

Therefore, simplicial complexes generalize the notion of graphs, which store the relationships between data points encoded as 1-dimensional structures. A collection of simplices is called a *simplicial complex* if all faces of any simplex in the collection are also in the collection.

In the following definition, this concept is presented in a more formal manner.

Definition 2.3. A *simplicial complex* K is defined as a finite collection of simplices such that for any simplex $\sigma \in K$, if $\tau \leq \sigma$ then $\tau \in K$. Furthermore, the following condition must also be satisfied: for all $\sigma, \sigma' \in K$, their intersection is a face of both σ and σ' or it is empty.

Simplicial complexes offer a straightforward way to construct filtrations. Each simplex σ in a complex can be associated with a real value $f(\sigma)$ that represents the parameter value at which it enters the filtration. The only requirement for the filter function f is that it satisfies the following consistency condition: If σ is a face of a higher-dimensional simplex τ (such as an edge on the boundary of a triangle), then $f(\sigma) \leq f(\tau)$.

This ensures that simplices appear in the filtration in a logical order. As the sublevel sets of a filtered simplicial complex evolve, the introduction of specific edges or higher-dimensional simplices can alter the topological structure of the underlying space. Homology provides a precise measure of topology by quantifying the number of connected components (0-dimensional homology), cycles (1-dimensional homology) or cavities (2-dimensional homology) present in the space. Consequently, changes in homology occur when connected components merge or new cycles are formed. These topological changes are attributed to *critical simplices*. Persistent homology captures the parameter values at which critical simplices arise, identifies the dimension in which homology changes occur, and pairs critical values by matching the appearance and disappearance of homological features.

Definition 2.4. The *underlying topological space* of a geometrical simplicial complex X is the space

$$|X| = \bigcup_{\sigma \in X} \sigma$$

with the topology induced by the Euclidean topology in $\mathbb{R}^N$.

Definition 2.5. An *abstract simplicial complex* with vertex set $V = \{v_i\}_{i \in I}$ is a set K of finite subsets of V such that elements of V belong to K and every subset $\sigma \subset K$ also belongs to K.

Definition 2.6. A *point cloud* is a finite set of points $X = \{x_i\}_{i \in I}$ in $\mathbb{R}^N$ for some $N > 1$.

Every point cloud X is a metric space with the Euclidean distance restricted to X.

In previous sections, homology was defined as a mathematical tool to be applied to simplicial complexes. In the present study, training data will be stored as point clouds. Therefore, two methods for building simplicial complexes from a given dataset will be described below.

Definition 2.7. (*Čech complex*) Given a point cloud X in $\mathbb{R}^N$ and $\epsilon \in \mathbb{R}^+$, $C_\epsilon(x)$ is the abstract simplicial complex with vertex set X whose k-faces are collections $\{x_{i_0}, \ldots, x_{i_k}\}$ such that the intersection of the closed balls $\bar{B}_{\frac{\epsilon}{2}}(x_{i_0}) \cap \cdots \cap \bar{B}_{\frac{\epsilon}{2}}(x_{i_k})$ contains at least one point.

Definition 2.8. (*Vietoris-Rips complex*) Given a point cloud X in $\mathbb{R}^N$ and $\epsilon \in \mathbb{R}^+$, $R_\epsilon(x)$ is the abstract simplicial complex with vertex set X whose k-faces are collections $\{x_{i_0}, \ldots, x_{i_k}\}$ of diameter at most ϵ, that is, $d(x_{i_r}, x_{i_s}) \leq \epsilon$ for all r, s.

2.3 Cubical complexes

Firstly, the necessary concepts will be presented for the subsequent definition of a cubical complex. See [16] and [17] for more detailed information.

Definition 2.9. *Elementary intervals* are divided into *non-degenerate* and *degenerate*. The first ones are intervals of a form $[n, n+1]$ for $n \in \mathbb{N}$ and their boundary is a chain $\partial[n, n+1] = [n+1, n+1] - [n, n]$. Degenerate intervals are of the form $[n, n]$ for $n \in \mathbb{N}$ and their boundary is $\partial[n, n] = 0$.

Definition 2.10. An *elementary cube* $\sigma \subset \mathbb{R}^n$, which can be referred to as an *n-cube*, is defined as a product of elementary intervals

$$\sigma = I_1 \times \cdots \times I_n,$$

where n represents the number of intervals, degenerate or not, and is called *embedding dimension*. The number of non-degenerate elementary intervals in the definition is named the *dimension* of the cube σ.

Accordingly, 3-cubes are also known as *voxels*, 2-cubes as *squares*, 1-cubes as *edges* and 0-cubes as *vertices*.

Definition 2.11. The *boundary* of a n-cube σ is a chain obtained as follows:

$$\partial\sigma = (\partial I_1 \times \cdots \times I_n) + (I_1 \times \partial I_2 \times \cdots \times I_n) + \cdots + (I_1 \times I_2 \times \cdots \times \partial I_n).$$

A cubical complex is a combination of cubes closed under the operation of taking boundary, which means that the boundary of every cube from the collection is in the collection.

More formally, a cubical complex can be defined as follows:

Definition 2.12. A *cubical complex* C is a collection of n-cubes satisfying that if $c \in C$ and $c' \subseteq c$ then $c' \in C$.

If filtration values are assigned to cubes, a *filtered cubical complex* is obtained.

It is worth mentioning that cubical complexes can be converted into simplicial complexes by cutting up cubes into simplicial pieces.

2.4 Filtrations

Definition 2.13. A *filtration* of a simplicial complex K is defined as a nested family of subcomplexes $(K_r)_{r\in T}$ where $T \subseteq \mathbb{R}$ satisfying that for all $r, r' \in T$ if $r \leq r'$ then $K_r \subseteq K_{r'}$, and $K = \bigcup_{r\in T} K_r$.

More generally, a *filtration* of a topological space M is described as a nested family of subspaces $(M_r)_{r\in T}$ where $T \subseteq \mathbb{R}$ such that for all $r, r' \in T$ and $r \leq r'$ then $M_r \subseteq M_{r'}$, and $M = \bigcup_{r\in T} M_r$.

2.4.1 Filtrations built on top of data

Definition 2.14. Given a subset X of a compact metric space (M, ρ), theVietoris-Rips complex $(R_\epsilon(x))_{r\in\mathbb{R}}$ and the Čech complex $(C_\epsilon(x))_{r\in\mathbb{R}}$ are filtrations. The parameter ϵ is defined as the *resolution* of the given dataset X.

2.4.2 Sublevel set filtrations

As pointed out in [6], there are two different ways of computing a cubical complex from a given d-dimensional image:

- *Lower-star filtration*: Voxels are viewed as vertices and higher dimensional cubes as coming from voxel adjacencies. In that way, an edge would be formed by pairs of adjacent pixels and squares would come from adjacent voxels, and so on.

  This can be extended to the entire complex as follows:

  Definition 2.15. Given a cube τ and a filter function f, $f(\tau)$ is defined as $\max_\sigma f(\sigma)$, where the maximum is taken over all vertices $\sigma < \tau$.

  Therefore, not square can emerge until all the constituent vertices appear.

- *Upper-star filtration*: Voxels are viewed as d-dimensional cubes and the lower-dimensional cubes as the faces of these voxels.

  This can be extended as before to the entire complex:

  Definition 2.16. Given a cube σ and a filter function f, $f(\tau)$ is defined as $\min_\tau f(\tau)$ where the minimum is taken over all voxels σ contained in τ.

  In that way, not until at least one of the voxels which form the cube appears, can a cube come into view.

2.5 Persistence diagrams

When applying persistent homology, a collection of intervals called a *barcode* is obtained. These intervals represent the lifetime of topological features (such as connected components, cycles or cavities) as the threshold parameter changes, from a birth parameter value b until a death parameter value d.

Definition 2.17. A *persistence diagram* is a topological signature which encodes a barcode as a collection of points by mapping each interval (b, d) to their corresponding point $(b, d) \in (\mathbb{R} \cup \{-\infty\} \cup \{\infty\})^2$.

Persistence diagrams can be transformed into vectors with a vectorization process. The coordinates (b, d) of each point in a persistence diagram correspond to the birth and death of a homology generator.

The points that exhibit proximity to the diagonal correspond to instances where death is close to birth. Consequently, the persistence of the topological feature is minimal and is generally viewed as noise or insignificant fluctuations in the data.

It is important to note that the presence of a point at infinity in homological dimension 0 is an inherent characteristic of persistence diagrams, owing to the fact that there is always a connected component that cannot disappear or merge with other components.

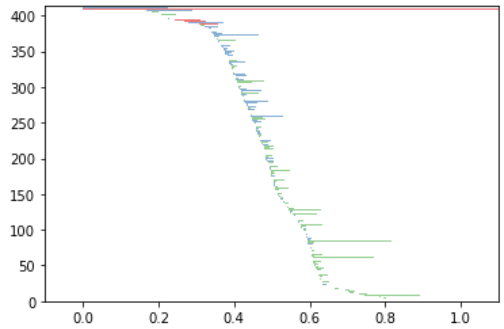

Figure 1: Barcode

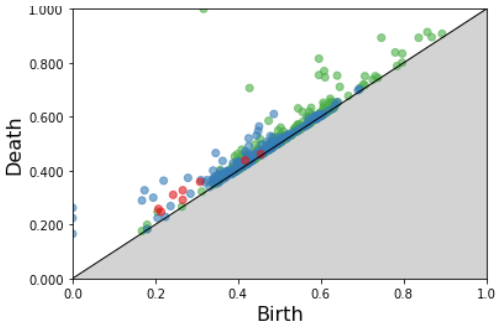

Figure 2: Persistence diagram

On the one hand, in Figure 1, a barcode of a specific 3-dimensional image is shown. Red colour represents topological features from dimension 0 which present the birth and death of connected components; green colour stands for 1-dimensional features as cycles, and, finally, blue colour corresponds to 2-dimensional features like cavities.

On the other hand, in Figure 2, its corresponding persistence diagram is exhibited. It can be perceived that the topological features which have the shortest persistence in the barcode become the closest points to the diagonal in the persistence diagram. It is important to note that the point at the infinity in dimension 0 is not displayed due to the chosen axis limits when plotting.

2.6 Properties of functional persistence

Some of the properties of functional persistence that support its use are described below:

- Stability to small perturbations of the input data, although it is not stable to outliers. This is caused by the fact that distance between the barcodes for $f, g \in X$ is bounded by $\| f - g \|_\infty$.
- Functional persistence is said to be flexible because there is a single persistence diagram associated to each pair (X, f). The lack of additional parameters makes possible its straightforward application but no data-dependent featurizations.

Nevertheless, it is important to mention that, due to the fact that there are many (X, f) producing identical barcodes, there is a difficulty in distinguishing images.

2.7 Persistence descriptors

Persistence descriptors are defined as numerical or vectorized summaries of the information obtained from persistence diagrams. They are especially significant for classification tasks, owing to the fact that they characterize different regions in the data and provide robust and interpretable results. Furthermore, their lower dimensions, compared to persistence diagrams, significantly reduces the computational costs of classification procedures.

2.7.1 Total persistence

Total persistence can be defined as a quantitative measure derived from persistent homology which analyzes the evolution of topological features, such as connected components, loops, or voids.

Total persistence measures the sum of the horizontal lengths of all the intervals in a barcode. Each interval corresponds to the existence of a topological feature for a certain range of threshold values. It is worth noting that total persistence is computed for each of the dimensions in the persistence diagram. Therefore, for 3-dimensional images, total persistence in homological dimensions 0, 1 and 2 is computed.

Total persistence of a persistence diagram D is computed as follows:

$$T(D) := \sum_{(b,d)\in D} |d - b|.$$

By considering total persistence, TDA enables the characterization of the global topological structure and the identification of meaningful features that persist across different scales. It provides valuable insights into the robustness and stability of the topological properties of the dataset.

2.7.2 Entropy

Entropy is another numerical measure obtained from a persistent diagram. In this context, it quantifies the degree of dispersion or scattering of points in the diagram. It provides insight into the complexity and uncertainty associated with the topological structure of the dataset. The entropy of a given persistence diagram D is defined as follows:

$$E(D) := -\sum_{i=1}^{n} \frac{d_i - b_i}{L} \log_2 \Big(\frac{d_i - b_i}{L}\Big),$$

where $L = \sum_{i=1}^{n}(d_i - b_i)$, and $(b_i, d_i) \in D$ for all $i \in I$.

3 Dataset

In the present work, the methodology was implemented using the Automated cardiac diagnosis challenge (ACDC) dataset. It was created for the homonymous challenge, and it is composed of real clinical exams conducted at the University Hospital of Dijon (France).

The ACDC dataset contains information from 150 patients, evenly divided into five groups, including four disease categories and one group of normal patients [15]. The classes were equally distributed in both the *Train* and *Test* datasets provided by the challenge organizers, which are formed by 100 and 50 patients respectively. Therefore, it is said to be a balanced dataset. Data is stored in *NIfTI* format.

The study includes five distinct groups, each characterized as follows:

- 30 normal subjects (NOR)
- 30 patients with previous myocardial infarction, exhibiting an ejection fraction of the left ventricle lower than 40% and several myocardial segments with abnormal contraction (MINF)
- 30 patients with dilated cardiomyopathy, demonstrating a diastolic left ventricular volume >100 mL/m2 and an ejection fraction of the left ventricle lower than 40% (DCM)
- 30 patients with hypertrophic cardiomyopathy, presenting a left ventricular cardiac mass higher than 110 g/m2, several myocardial segments with a thickness higher than 15 mm in diastole, and a normal ejection fraction (HCM)
- 30 patients with abnormal right ventricle, characterized by a volume of the right ventricular cavity higher than 110 mL/m2 or an ejection fraction of the right ventricle lower than 40% (RV)

The dataset is separated into training and hold-out testing set by the challenge organizers. The training set includes 100 patients, i.e. 20 patients for each group, while the testing set includes 50 patients, i.e. 10 patients per group.

As it can be observed, each group has been defined based on different physiological parameters.

For each patient, the following data are provided:

- A 4-dimensional CMR image, with the fourth dimension representing the phase of the cardiac cycle. From this image, 3-dimensional CMR images were extracted to capture the states of end-systole (ES) and end-diastole (ED).
- 3-dimensional segmentations of the heart corresponding to the two cardiac phase of interest, ie. ED and ES. It is divided into three regions of interest: left ventricle (LV), myocardium (MYO) and right ventricle (RV).

- A file with information regarding the patient, such as the frame of the end-systole or end-diastole.

The objective of utilizing segmentations is to partition the image into distinct anatomical structures, allowing for their individual characterization and subsequent analysis.

Formally, the process of segmentation involves utilizing masks, which are arrays containing distinct numbers that represent different regions of interest (ROIs) within an image. These masks serve as multi-class maps, where each number corresponds to a specific region.

To obtain the texture of the ROIs, the Hadamard product, also referred to as element-wise multiplication, is performed between the original grayscale CMR and the segmentation mask. The Hadamard product operates on corresponding elements of two arrays and produces a new array where each element is the product of the corresponding elements in the original image and the segmentation mask. It can be defined as follows:

Definition 3.1. Let us denote the original image as $I(x, y, z)$ where (x, y, z) represents the spatial coordinates of a voxel, and the segmentation mask as $S(x, y, z)$ where each value represents a specific structure or region. The *segmented image*, denoted as $SI(x, y, z)$, is obtained through the Hadamard product as follows:

$$SI(x, y, z) = I(x, y, z) * S(x, y, z)$$

By performing the Hadamard product, the values of the original image are selectively multiplied by the corresponding values in the segmentation mask. This process suppresses the information outside the regions of interest, facilitating subsequent analysis, visualization, and interpretation.

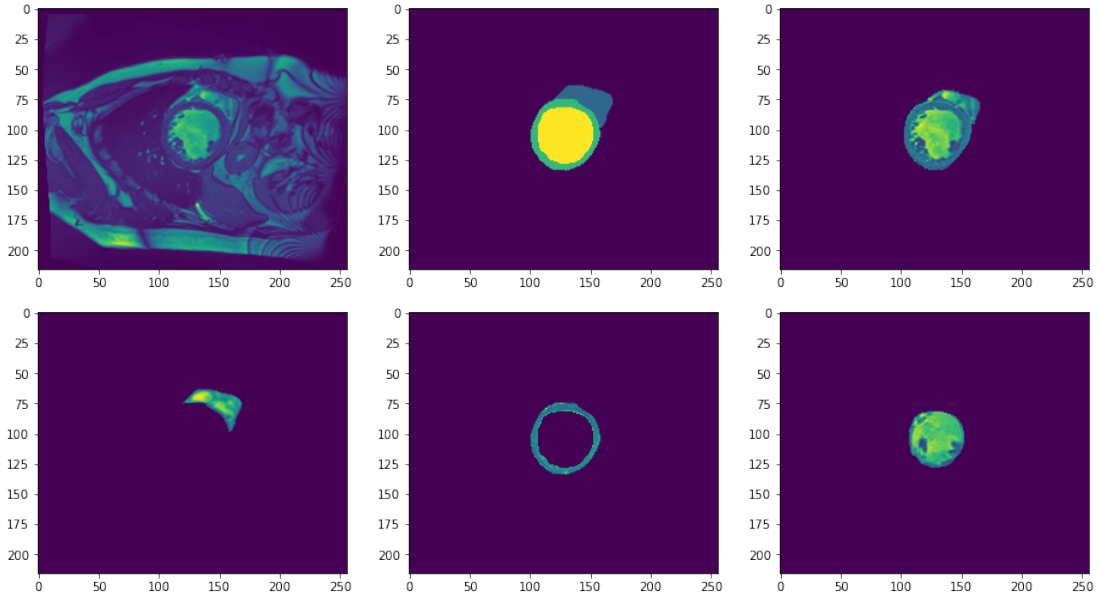

Figure 3: Cardiac cine-MRI, segmentation mask and segmented images

The first image located in the top-left corner corresponds to the cardiac cine-MRI. The adjacent image corresponds to the segmentation, which, when applied, results in the image in the upper right corner where the isolated heart can be observed.

The three images at the bottom of Figure 3 exhibit the segmented images obtained according to anatomical and functional characteristics of the heart. The different structures shown are the right ventricle, the myocardium and the left ventricle from left to right.

The left ventricle is one of the four chambers of the heart which is located in the lower left portion. It distinguished itself due to its thicker and more muscular walls compared to the other chambers. On the other hand, the right ventricle is one of the other chambers of the heart situated in the lower right portion of the heart. Lastly, the myocardium is a muscular tissue and represents the middle layer of the heart wall, and it is responsible for the contraction of the heart [31].

Note that, due to their three-dimensional nature, only the middle slice is shown. However, topological analysis is performed on the 3D image volumes respectively.

4 Methods

4.1 Topological features extraction

When computing persistence diagrams, filtration is given at the maximal cubes, and it is then extended by the lower-star filtration to all cubes. So as to invert this process and compute the persistence diagrams applying the upper-star filtration, complementary images were computed. These images are calculated substracting the maximum voxel value of the original image from each voxel value and taking the absolute value of the difference. Further details on the disparity between the persistence diagrams obtained and therefore the topological descriptors will be detailed thereafter.

The shape of the initial image is preserved. However, it represents the inverse of the original image in terms of intensity. The brighter areas in the source image correspond to the higher voxel values as opposed to the darker areas. On the contrary, in the complementary image, the higher voxel values come from darker areas in the source image.

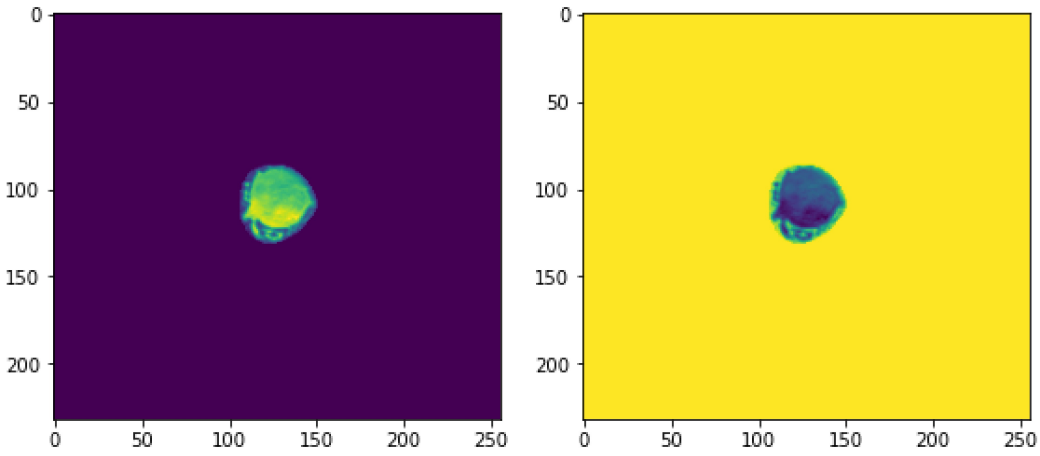

Figure 4: Original and complementary images
corresponding to the middle slice
of the left ventricle at end-diastole CMR

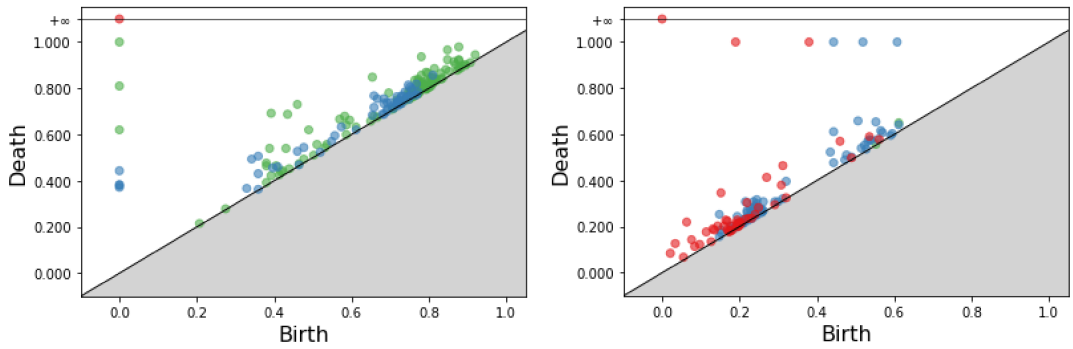

Figure 5: Persistence diagrams of original and complementary image

On the left, the persistence diagram of a 3-dimensional image of the right ventricle at end-systole is displayed. On the right, the persistence diagram corresponding to the complementary image is shown.

Note that the red points represent topological features of homological dimension 0, while the green points stand for 1-dimensional features. Lastly, the blue points represent topological descriptors of homological dimension 2.

As the persistence diagrams state clear dissimilarities from source and complementary image, the persistence descriptors extracted from them will also be different as shown in Tables 1 - 2.

	Lower-star filtration	Upper-star filtration
Total persistence dimension 0	0	3.388
Total persistence dimension 1	3.552	3.608
Total persistence dimension 2	7.173	0.042

Table 1: Total persistences extracted from persistence diagrams in Figure 5

	Lower-star filtration	Upper-star filtration
Entropy dimension 0	0	2.898
Entropy dimension 1	3.310	3.379
Entropy dimension 2	3.806	0.325

Table 2: Entropies extracted from persistence diagrams in Figure 5

In Figures 6 and 7 the following function

$$f(x,y) = \frac{1}{x^2+y^2+\frac{1}{2}} + \frac{3}{(x-3)^2+(y-3)^2+1} + \frac{5}{(x-6)^2+(y-6)^2+2}$$

is plotted as well as the plane $z = 2$. Function f is a combination of three Gaussian functions that simulates the topology of the space. Note that this is a 3-dimensional simplification due to the fact that in the present work, one higher dimension is dealt with so as to compute persistence homology in dimension 2 but could not be graphically represented.

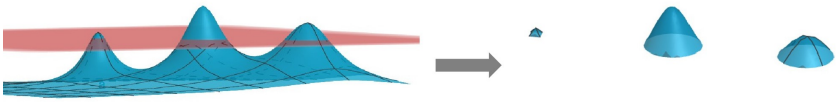

Figure 6: Upper-star filtration computation

On the one hand, in Figure 6, the upper-star filtration is applied. This type of filtration starts with the entire space as the first subset, treating the voxels as vertices and gradually including smaller subsets as the parameter decreases or the scale increases. The subsets are added in a way that preserves the inclusion relationship, meaning that each subsequent subset contains the previous subset. Each subset in the filtration corresponds to a specific simplicial complex that captures the topology of the data at a particular parameter value.

When H_0 is computed, 3 connected components are taken into account. Nonetheless, there are no cycles to be counted by H_1.

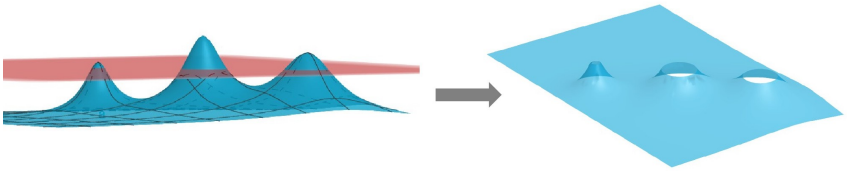

Figure 7: Lower-star filtration computation

On the other hand, in Figure 7, the lower-star filtration is applied. Unlike the upper-star filtration, it starts with the smallest possible subset and progressively adds larger subsets as the parameter increases or the scale decreases. Similarly to the upper-star filtration, the subsets are added in a way that maintains the inclusion relationship, where each subsequent subset contains the previous subset.

Each subset in the lower-star filtration corresponds to a specific simplicial complex that captures the topology of the data at a particular parameter value. As the parameter increases or the scale decreases, simplices are gradually added to the complex, resulting in a sequence of simplicial complexes that represent the evolving topological features of the data.

When computing the persistent homology in Figure 7 in dimension 0 (H_0), the result is only one connected component. However, there will be 3 cycles counted for persistent homology in dimension 1 (H_1).

It is important to note that while the upper-star filtration builds subsets from top to bottom (larger to smaller), the lower-starfiltration builds subsets from bottom to top (smaller to larger). This alternative perspective provides a different lens through which to analyze the data's topological properties, potentially revealing distinct structural insights depending on the chosen filtration approach.

4.2 Machine learning

Machine Learning can be defined as the automated detection of meaningful patterns in data. It involves the development of computational algorithms and system programs that have the ability to acquire knowledge and adapt their behaviour so as to improve their performance over time [8].

It plays an important role in tasks that a human programmer would not be able to provide specification of how they should be executed. Moreover, machine learning can also handle large and complex datasets.

Overall, the main aim of machine learning is to progress from individual examples by means of inductive inference in order to obtain a broader generalization.

The input for learning algorithms is training data, which can be referred to as experience while their output is expertise. Usually the aformentioned expertise is in the form of a computer program which can perform some task.

Machine learning algorithms can be divided into two main categories:

- *Supervised machine learning*: This type of machine learning is based on a scenario in which the inputs provided contain relevant information that is missing in the testing set in which the output will be evaluated. In other words, the model is trained on data that includes relevant information, known as labels, that may not be available during the evaluation or testing phase.

  Therefore, the output is to generate predictions or forecasts for the missing information in the test dataset. In this context we can conceptualize the environment as an instructor that guides the learner by providing the information regrading the labels.

- *Unsupervised machine learning*: It presents no distinction between training and testing data. More precisely, input data is processed so as to capture the underlying patterns, structures, or relationships within the data, without the need for explicit labels or predefined outputs as in the supervised case.

In the present work, only supervised machine learning algorithms were used due to the advantages that they present and the nature of the problem. These types of algorithms are trained to make highly accurate predictions as they have a clear aim which is to minimize the difference between predicted and actual values. In addition, due to the explicit feedback that they receive, an iterative process of training, evaluation and refinement can be carried out.

However, it is of significance to acknowledge that unsupervised machine learning has a great impact on fields in which labelled data is scarce or unavailable.

Subsequent definitions will be required for a better understanding of boosting and bagging methods adopted in this work.

Definition 4.1. In machine learning, the *bias* of a predictor is defined as the difference between the predicted mean value by the model and its actual mean.

High bias values indicate that underfitting process is taking place. Note that underfitting refers to the incompetence of the model to capture underlying patterns and relationships in the training data. It arises when a model is overly simplistic or lacks the required complexity.

Definition 4.2. *Variance* in machine learning assesses the susceptibility of the model to fluctuations in the input. Therefore, it determines the extend to which its predictions may vary.

On the contrast, high variance values denote that the predictor is highly sensitive to specific characteristics of the training dataset which leads to overfitting.

Some machine learning algorithms rely on the principle of ensembles which leverages the combination of multiple models so as to improve accuracy, robustness and generalization.

This ensemble techniques can be categorized into two different groups:

- Bagging: This method entails training multiple models separately on diverse subsets of the training data, often employing techniques like bootstrap sampling. The ultimate prediction is typically derived by aggregating the individual model predictions through averaging or voting, mitigating the influence of individual model variances.
- Boosting: It is an iterative procedure in which models are trained sequentially. Each of the derived models is specifically designed to correct the errors made by the previous models. The ultimate prediction is obtained by aggregating the weighted estimations of all the models in the ensemble. This iterative nature of boosting allows the models to collectively improve its performance over time, leading to more accurate and robust predictions. In addition, it focuses not only in variance but also in bias.

4.2.1 Empirical risk minimization

As mentioned before, the machine learning algorithm receives a training set S as an input which is sampled from an unknown distribution D and labeled by some target function denoted f. The algorithm aims to produce a predictor $h_S : X \to Y$. Note that the dependence of h on S arises from its definition.

The objective of the algorithm is to discover a predictor h_S that minimizes the discrepancy between the predictions made on unseen data according to the distribution D and the true labels defined by the target function f.

Definition 4.3. The error in which the classifier incurs over the training sample is called *empirical error* and is defined as follows:

$$L_S(h) = \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m},$$

where $[m] = \{1, \ldots, m\}$ and m is the number of samples.

The empirical risk minimization process refers to the process of the algorithm to provide a predictor h which reduces $L_S(h)$.

4.2.2 Gradient Boosting

Gradient Boosting is a general ensemble machine learning method which combines a set of weak learners so as to obtain a strong predictive model [26], [27].

At an iteration t, the model outcomes are weighted based on the results of the previous iteration $t - 1$. The outcomes which are predicted correctly are given a lower weight while the incorrectly labeled have higher weights.

One of its advantages is that it is robust to overfitting and it is based on the optimization of an arbitrary differentiable loss function. The aforementioned loss function chosen by default in *Python* is the one which refers to binomial and multinomial deviance which is appropriate for classification with probabilistic outputs.

The parameters used by this method that were optimized by means of *Grid Search* are explained below and this optimization process will be discussed in detail in subsequent sections:

- *learning_ rate*: It determines the influence of each tree on the final outcome. In gradient boosting, the process begins with an initial estimate, which is then refined using the output of each subsequent tree. The learning parameter controls the extent to which these estimates are adjusted.

  Lower values of the learning parameter are typically preferred as they promote model robustness, enabling better generalization. However, opting for lower values implies a larger number of trees to effectively capture all the relationships in the data, which can lead to an increase in computational complexity.

- *n_ estimators*: It refers to the quantity of consecutive trees to be constructed.

- *max_ depth*: It indicates the maximum depth of a generated tree. It is used to control overfitting due to the fact that a higher depth will result in the learning of the specific patterns in training data by the predictor.

It should be taken into consideration that the parameter *random_ state* was initialized to 0 in the coding process so as to ensure that the model produces the same results or behavior when the code is run multiple times, given the same dataset and hyperparameters.

4.2.3 XG-Boost classifier

XG-Boost, otherwise referenced as Extreme Gradient Boosting, represents an enhanced version of a gradient boosting algorithm [24], [25] .

One of its main advantages is that regularization is applied so that the complexity of the objective function is controlled adding a penalty and pushing some of its coefficients to 0.

In contrast with Gradient Boosting which stops when a negative loss in the split is reached, *XG*-Boost makes splits until the maximum depth specified and afterwards it performs backward tree pruning, eliminating splits that do not provide any positive gain.

The parameters which have been optimized with the use of *Grid Search* are detailed below:

- *n_ estimators*: As in Gradient Boosting, this parameter stipulates the number of successive trees that will be generated.
- *max_ leaves*: It refers to the maximum number of nodes to be added.

4.2.4 K-nearest neighbors

K-nearest neighbors is a non-parametric and supervised learning method used for classification and regression. When used for classification tasks, as in the present work, the output is a class label. The algorithm consists of storing feature vectors with each corresponding class labels. Subsequently, the unlabeled vectors are assigned the most frequent target among the k training samples which are the nearest. This method relies on computing the distance between the multidimensional vectors, which is usually the Euclidean distance. In the library *Scikit-learn* from *Python* used for the computations, the Minkwoski metric is applied.

Definition 4.4. Given two multidimensional vectors $X = (x_1, \ldots, x_n)$ and $Y = (y_1, \ldots, y_n)$ in $\mathbb{R}^n$, its *Minkowski distance* of order $p \in \mathbb{Z}$ is estimated as follows:

$$D(X, Y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}.$$

Particulary, given an instance domain X endowed with a metric function ρ described as $\rho : X \times X \to \mathbb{R}$ which computes the distance as defined previously between two elements of X.

Let $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ indicate a sequence of training examples where $(y_1, \ldots, y_m)$ represent their corresponding labels.

Definition 4.5. For each $\mathbf{x} \in X$, $\pi_1(\mathbf{x}), \ldots, \pi_m(\mathbf{x})$ denotes a reordering according to the distance to $\mathbf{x}$ so that the condition below is satisfied:

$$\rho(\mathbf{x}, \mathbf{x}_{\pi_i(\mathbf{x})}) \leq \rho(\mathbf{x}, \mathbf{x}_{\pi_{i+1}(\mathbf{x})}), \quad \text{for all } i < m.$$

The k-NN rule for binary classification states that the output for every $\mathbf{x} \in X$ is the majority label among $\{y_{\pi_i(\mathbf{x})} : i \leq k\}$.

In the coding implementation, the algorithm by default is the one which chooses which one of the following algorithms outperforms the others: ball-tree, *kd*-tree and brute-force search.

- *Ball-Tree*: It organizes points in a hierarchical structure where each node represents a ball containing multiple points. The tree recursively partitions the dataset based on the bounding spheres. Because of the narrowing of the search space, fast nearest neighbor queries are achieved.
- *kd-tree*: It is a k-dimensional tree which represents a binary tree structure organizing points in a k-dimensional space. In this case, the space is partitioned into hyperplanes perpendicular to the coordinate axes. By adopting this approach, the points are divided into two subsets at each node. It represents an advantage in high-dimensional spaces as it enables efficient search for nearest neighbors.
- *Brute-force search*: It is characterized by its straightforward approach which involves computing the distance between a query point and all the other points in the dataset and its subsequent comparison. Overall, this method does not rely on any specific structure or optimization techniques although its computational cost is elevated.

The number of K nearest neighbors is an integer which can be defined by the user and its selection depends on the data. Larger values of K reduce the effect of noise although it can reduce the boundaries between classes. However, this parameter can be estimated with hyperparameter optimization, which will be discussed in more detail below.

4.2.5 Support vector machine

Definition 4.6. Given a training sample $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{\pm 1\}$. We say that S is *linearly separable* if there exists a halfspace $(\mathbf{w}, b)$ such that $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$, for all $i \in [m]$.

Definition 4.7. Those $(\mathbf{w}, b)$ satisfying the previous condition are called *ERM hypotheses*.

It is noteworthy to emphasize that for any linearly separable S many ERM halfspaces can be computed.

Definition 4.8. The *margin of a hyperplane*, in relation to a training set, is defined as the minimum distance between a point in the training set and the hyperplane. When a hyperplane has a large margin, it remains capable of separating the training set even when instances are slightly perturbed.

The Hard-SVM learning rule is characterized as the algorithm that produces an ERM hyperplane with the maximum achievable margin, ensuring complete separation of the training set.

Proposition 4.9. *The distance between a point* $\mathbf{x}$ *and the hyperplane* $(\mathbf{w}, b)$ *where* $\|\mathbf{w}\| = 1$ *is* $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b|$.

From this proposition, it arises the definition of the closest point in the training set to the separating hyperplane which is as follows:

$$\min_{i \in [m]} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b|.$$

Definition 4.10. *Hard-SVM* can be formally reformulated as:

$$arg\max_{(\mathbf{w},b):\|\mathbf{w}\|=1} \min_{i \in [m]} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b|$$

such that $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$ for all i.

If the linearly separable case is being considered, we can reformulate the problem as follows:

$$arg\max_{(\mathbf{w},b):\|\mathbf{w}\|=1} \min_{i \in [m]} y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \tag{4.1}$$

The Hard-SVM rule can be reformulated as a quadratic optimization problem as its objective is to minimize convex quadratic function and the constraints are linear inequalities. Its input is the set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$ and the aim of the algorithm will be the computation of

$$(\mathbf{w}_0, b_0) = arg\min_{\mathbf{w}_0, b_0)} \|\mathbf{w}\|^2 \tag{4.2}$$

such that $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$ for all i.

Its final outputs will be $\hat{\mathbf{w}} = \frac{\mathbf{w}_0}{\|\mathbf{w}_0\|}$ and $\hat{b} = \frac{b_0}{\|\mathbf{w}_0\|}$.

By means of the following lemma, the output of hard-SVM will be proved to be the separating hyperplane with the largest margin.

Lemma 4.11. *The output of Hard-SVM is a solution of Equation* (4.1).

Proof. Let $(\mathbf{w}^\star, b^\star)$ be a solution of equation (4.1) and $\gamma^\star = \min_{i \in [m]} y_i(\langle \mathbf{w}^\star, \mathbf{x}_i \rangle + b^\star)$ the margin achieved.

It follows that for all i,

$$y_i(\langle \mathbf{w}^\star, \mathbf{x}_i \rangle + b^\star) \geq \gamma^\star$$

which can be reformulated as

$$y_i(\langle \frac{\mathbf{w}^\star}{\gamma^\star}, \mathbf{x}_i \rangle) + \frac{b^\star}{\gamma^\star} \geq 1.$$

Therefore, $(\frac{\mathbf{w}^\star}{\gamma^\star}, \frac{b^\star}{\gamma^\star})$ satisfies the condition of quadratic optimization in (4.2).

Hence, $\|\mathbf{w}_0\| \leq \|\frac{\mathbf{w}}{\gamma^\star}\| = \frac{1}{\gamma^\star}$ which implies that for all i,

$$y_i(\langle \hat{\mathbf{w}}, \mathbf{x}_i \rangle + \hat{b}) = \frac{1}{\|\mathbf{w}_0\|} y_i(\langle \mathbf{w}_0, \mathbf{x}_i \rangle + b_0) \geq \frac{1}{\|\mathbf{w}_0\rangle} \geq \gamma^\star.$$

As $\|\hat{\mathbf{w}}\| = 1$, $(\hat{\mathbf{w}}, \hat{b})$ is proved to be an optimal solution for (4.1). □

So as to use Hard-SVM, the strong assumption that the training set is linearly separable has to be made. However, there is a relaxation named Soft-SVM of this algorithm in which this suposition is not necessary.

Equation (4.2) imposes the precise restriction $y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$, $\forall i$. This constraint can be breached for some examples in the training set. For this reason, the variables $\xi_1, \cdots, \xi_m > 0$ are introduced and the previous condition is replaced as follows:

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \forall i.$$

The goals of the Soft-SVM algorithm are to minimize the margin $\|\mathbf{w}\|$ and reduce ξ_i which quantifies the violation of the original constraint. The balance between these two expressions is controlled by the parameter λ.

Definition 4.12. The input of the *Soft-SVM* algorithm is the training set $S = (\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_m, y_m)$ which is not required to be linearly separable.

The problem which is aimed to be solved is:

$$\min_{\mathbf{w}, b, \xi} (\lambda \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{i=1}^{m} \xi_i)$$

such that $\forall i, y_i(\|\mathbf{w}, \mathbf{x}_i\| + b) \geq 1 - \xi_i, \xi_i \geq 0$.

Its final output will be $\mathbf{w}, b$.

Kernel based learning

Subsequent definitions will be required for complete understanding of Kernels.

Definition 4.13. Given a vector space V, $u, v, w \in V$ and $\alpha \in \mathbb{R}$, the *inner product* is defined as follows:

$$\langle \cdot, \cdot \rangle : V \times X \to \mathbb{R}$$

with these constraints:

- Positive-definite and non-degenarate conditions:
  $\langle v, v \rangle > 0$ and $\langle v, v \rangle = 0$ if and only if $v \equiv 0$.
- Symmetric: $\langle v, w \rangle = \langle w, v \rangle$.

- $\langle \alpha\ v, w \rangle = \alpha\ \langle v, w \rangle$
- $\langle u + v, w \rangle = \langle u, w \rangle + \langle v, w \rangle$.

Definition 4.14. A *complete space* is a vector space in which all Cauchy sequences converge.

Definition 4.15. A *Hilbert space* is a vector space with inner product which is also complete.

Kernels can be defined as inner products in the feature space and represents a type of a similarity measure between instances. Its more relevant particularity is that they can be viewed as inner products in some Hilbert spaces or Euclidean spaces of high dimensions to which the instance space is embedded.

Definition 4.16. Given an embedding ϕ of some domain X into some Hilbert space, the *Kernel function* is defined as:

$$K(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle.$$

K specifies the similarity between instances and the embedding ϕ as mapping the domain X into a space where the aforementioned similarities are realized as inner products. Kernel-based learning algorithms enable the development of classifiers for halfspaces by utilizing the values of the Kernel function calculated for pairs of input data points. The key advantage of Kernel-based learning lies in its ability to construct linear separators in high-dimensional feature spaces without explicitly specifying the points within that space.

The implementation of SVM used in this work follows the Soft-SVM approach to allow certain degree of misclassifications with the objective of achieving better generality. Moreover, different Kernel functions were provided as hyperparameters to be defined through grid-search such as linear, polynomial, Gaussian and radial basis.

In addition, apart from optimizing the Kernel function used, there is also a parameter C which must be tuned. This parameter determines the degree of importance placed on correctly classifying each training example. When C is set to a high value, the optimization prioritizes achieving accurate classification by selecting a hyperplane with a smaller margin. On contrast, a low value of C prompts the algorithm to seek a larger-margin hyperplane, even if it means misclassifying more points.

4.2.6 Grid-search

Grid-search is a hyperparameter optimization approach which involves the exhaustive searching over a predefined set of parameters for an estimator. In this work, these aforementioned parameters are optimized by cross-validated grid-search over a parameter grid. All the parameters that were optimized following this procedure were described in previous sections.

4.2.7 Cross-validation

When training a model it occurs that it learns the parameters of prediction function so if we tested it with the same data that was used for training it would have perfect accuracy. However, it would fail to predict unseen data due to the overfitting process. This situation is called overfitting. To avoid it, it is common practice when developing a (supervised) machine learning model to hold out part of the available data as a test set.

During hyperparameter optimization, overfitting remains a concern as the parameters can be selected to maximize the model's performance. To mitigate this, we could split further split the training data into training and validation using cross-validation. In that way, the model´s performance would be evaluated on the validation set before the final evaluation on the hold out testing set.

Druring cross-validation, for each fold, the model is trained using the remaining $k-1$ folds as training data and the k-th fold would be used as a validation set to assess the model performance.

The performance of the model is measured by the average of the metrics computed for each of the k folds generated.

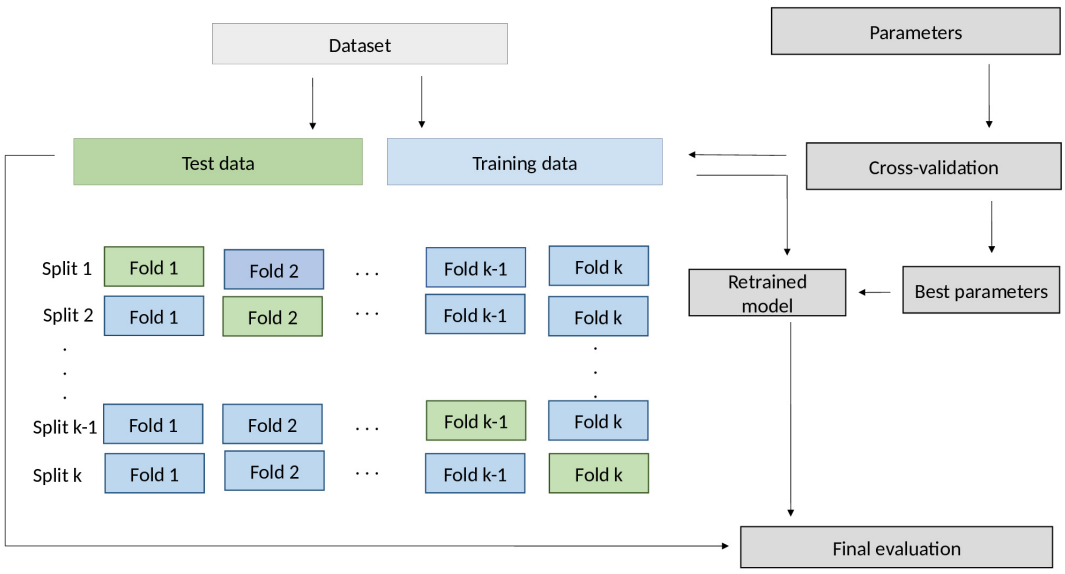

Figure 8: Cross-validation workflow and k-fold generation

In Figure 8, it is outlined the sequential steps involved in the cross-validation procedure as well as the generation of the k folds. Furthermore, the grid search process is incorporated to determine the optimal parameters for the model.

It is noteworthy that, for each iteration, the testing set corresponds to the fold marked in green (designated as the i-th split), while the model is trained on the remaining folds marked in blue.

4.3 Experiment set up

Within this section, the workflow followed in this study will be examined.

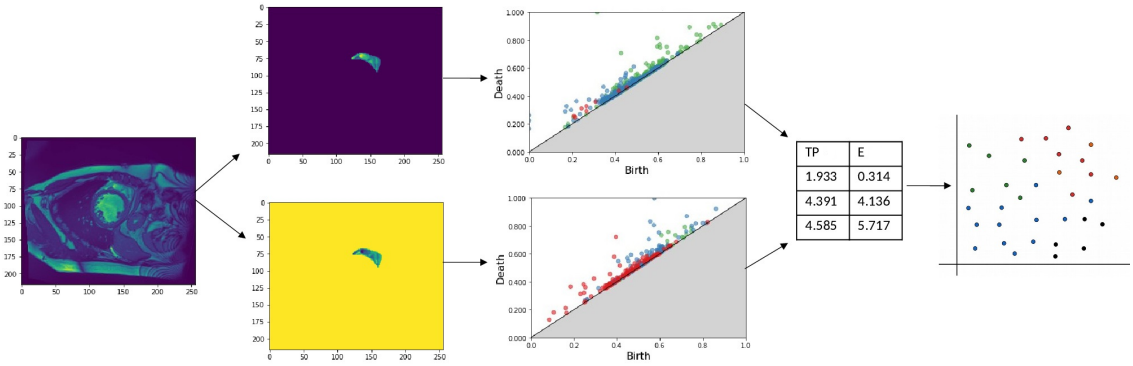

Figure 9: Overview of the proposed pipeline

In Figure 9, the methodology of the present study is displayed. Firstly, the three segmentations are applied to the normalized CMR image and afterwards the complementary image of each of them is computed. Subsequently, their persistence diagrams are calculated and the topological features extracted from them. Lastly, the classification procedure takes place with the normalized features so as to obtain a predictor.

4.4 Image preprocessing

In the subsequent sections, the different pre-processing techniques will be discussed.

4.4.1 Cropping

No cropping was applied in the CMR images due to the fact that removing part of their background would not have a significative impact in the changes of the topological descriptors extracted.

However, it is worth-mentioning that the images did not have the same number of slices in the third dimension due to the different morphological characteristics of each of the patients in the dataset. There was no standardization of the number of slices because when reconstructing the 3 dimensional image so as to obtain the descriptors, it is important not to miss any part of the heart because it could provide important information for diagnostic.

4.4.2 Image normalization

Although there is no direct effect on clinical medical diagnosis by doctors when no normalization is applied, it aids overcome discrepancies in intensities between different patients related to the acquisition process. In this work, the images were first normalized using histogram matching, using as reference one of the studies from the dataset [35].

Subsequently, normalization in the range of $[0, 1]$ was applied. When normalizing a n-dimensional gray-scale image $I\colon \{X \subseteq \mathbb{R}^n\} \to \{\min, \ldots, \max\}$ with voxel intensities in the range $(\min, \max)$ is transformed into a new image $I_N\{X \subseteq \mathbb{R}^n\} \to \{\min', \ldots, \max'\}$ with voxel intensities in the range $(\min', \max')$. Linear normalization is computed according to the following formula:

$$I_N = (I - \min)\,\frac{\max' - \min'}{\max - \min} + \min'.$$

It should be noted than when reducing the range of voxel values in images and scaling the data to a smaller range, the subsequent computations of persistence features were expedited due to the computationally manageable numbers that were obtained.

4.5 Feature normalization

Standard feature normalization is a common technique used in ML and data analysis to transform numerical features in a dataset. It aims to ensure that the features have a similar scale and distribution, which can be beneficial for certain algorithms that are sensitive to the scale of the input data.

It applies a transformation to each feature by subtracting the mean and dividing by the standard deviation. This process centers the data around zero and scales it to have a unit variance.

Mathematically, the transformation for each feature can be represented as:

$$x_{normalized} = \frac{x - \mu}{\sigma},$$

where x is the original feature value and μ and σ refer to the mean and the deviation of the feature values in the training set respectively.

5 Results

5.1 Evaluation metrics

Several metrics were employed to evaluate the performance of the proposed methodology [32], [33], [34]. Before delving into the definitions of these metrics, it is essential to clarify certain abbreviations used in this section. More precisely, the following abbreviations will be utilized:

- TP is the abbreviation for true positive which refers to the positive cases correctly classified.
- FP is an acronym for false positive and it represents the instances in which the negative cases are incorrectly classified as positive.
- TN is analogous to true positive but referring to the negative cases instead.
- FN stands for false negatives and represents the misclassification of positive cases as negative.

Subsequently, we proceed with the introduction of the evaluation metrics used in this work:

- Accuracy: It is defined as the proportion of correctly predicted cases, both positive and negative, relative to the total number of evaluated cases.

  In the context of this work, a high value of accuracy translates to a high ability of the classifier to correctly predict cases of cardiac and non-cardiac diseases.

  Accuracy is computed with the expression below:

$$Acc = \frac{TP + TN}{TP + TN + FP + FP}$$

While accuracy provides an overall measure of correct classifications, there are other metrics that will be exposed which offer additional insights into the classifier's behaviour that will be helpful in the evaluation of the model, specially in the medical field.

- Precision: It is a measure that quantifies the ratio of true positive predictions to the total number of positive predictions made by the classifier. It reflects the accuracy of identifying positive cases correctly.

  A high precision denotes that the classifier has a low rate of false positives which guarantees that the majority of cases classified as positive are indeed

cases of cardiac disease. Furthermore, this is desirable in the medical context as it helps to avoid incorrect diagnoses.

On the contrary, a low precision indicates that the classifier has a high rate of false positives, implying that some of the cases classified as positive may be false or unrelated to cardiac diseases. This can lead to an increase in the number of unnecessary tests or treatments.

This measure is calculated as follows:

$$Prec = \frac{TP}{TP + FP}$$

- Recall or sensitivity: It is the proportion of positive cases that are correctly classified among all the positive cases. It refers to the ability of the classifier to detect all positive cases, minimizing false negatives.

  On the one hand, a high recall ensures that the classifier has a high ability to detect and capture true positive cases, thereby minimizing the chances of missing patients with the disease.

  On the other hand, a low recall indicates that the classifier has a high rate of false negatives, implying that some cases of cardiac diseases are not detected. This can result in important diagnoses being missed.

  Recall is computed with the following formula:

$$Rec = \frac{TP}{TP + FN}$$

- F1 score: It combines precision and recall into a single metric and is defined as their harmonic mean. The F1 score is especially valuable in scenarios where the dataset exhibits class imbalance.

  In addition, a high F1 score implies a steady trade-off between precision and recall signifying the model's proficiency in correctly identifying true positive cases while minimizing both false positives and false negatives.

  F1 score follows the formula below:

$$F1 = 2 \cdot \frac{Prec \cdot Rec}{Prec + Rec}$$

- Receiver operating characteristic (ROC): It is a graphical plot which demonstrates the performance characteristics of a binary classifier as the discrimination threshold is adjusted. It plots the true positive rate (TPR) against the false positive rate (FPR). On the one hand, TPR stands for the fraction of true positives out of positives cases so it is equivalent to the definition of recall. On the other hand, FPR is defined as the fraction of false positives out of negative cases which is computed as follows: $FPR = \frac{FP}{FP+TN}$.

  Note that in the present study, a multi-class classification problem is faced. Therefore, the ROC curves are computed for each of the classes against the others.

- ROC-AUC: It computes the area under the ROC curve so as to summarize the information of the curve in a number. It can be calculated with the algorithm one-vs-rest for multi-class approaches so as to obtain the macro-average ROC-AUC score.

  The ROC-AUC represents the overall discriminative ability of the model in distinguishing between the positive and negative classes. It quantifies the model's ability to correctly rank positive instances higher than negative instances across different threshold settings. In medical applications, a high ROC-AUC implies that the model can effectively differentiate between individuals with a specific disease and those without it.

- Confusion matrix: The confusion matrix, denoted as C, provides a tabular representation of the model's predictions. Each element $C_{i,j}$ within the matrix represents the count of observations that belong to the true class i but have been predicted to be in class j.

  The elements of the diagonal represent the subjects whose classes were correctly predicted. Therefore, higher numbers in the diagonal indicate better predictions.

Note that, as it was considered a multi-class classification, all the aforementioned scores were computed independently for each class. So as to obtain an overall score, the macro-average was calculated as the unweighted mean. This type of average does not take label imbalance into account. However, this does not represent a problem as the ACDC dataset has balanced classes.

5.2 Analysis of the results

Firstly, the classification task was performed separately for each topological descriptor in a specific dimension, taking into account the lower-star filtration at one time and the upper-star filtration at another time. In this way, the results obtained for each type of filtration could be contrasted, choosing the most optimal for each descriptor in every dimension. The tables comprising the obtained evaluation metrics are displayed in Appendix 7.1. It should be noted that the results presented in the Appendix and throughout this section correspond to the testing set consisting of 50 patients.

According to these results, the lower-star filtration performs best for total persistence in dimension 1 and entropy in dimensions 0 and 1. On the other hand, upper-star filtration achieves better results for total persistence in dimensions 0 and 2 and entropy in dimension 2.

5.2.1 Total persistence and entropy in three dimensions

In Tables 3 - 4, the classification metrics for total persistence and entropy for each of the classifiers are shown. The highest accuracy value is highlighted in bold, as well

as the classifier used in that case. Accuracy was the chosen variable to assess the efficiency of the classifier, as it provides an overall measure of correct classifications.

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
TP_0_1_2	**KNN**	**0.68**	0.73	0.69	0.80	0.68
	GB	0.61	0.65	0.62	0.90	0.61
	XG-B	0.68	0.69	0.67	0.92	0.68
	SVM	0.66	0.73	0.68	0.94	0.66

Table 3: Comparative performance of different classifiers based on total persistence in the 3 homological dimensions.

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
ENTR_0_1_2	KNN	0.64	0.65	0.65	0.90	0.64
	GB	0.62	0.63	0.61	0.87	0.62
	XG-B	0.64	0.65	0.64	0.91	0.63
	SVM	**0.74**	0.74	0.73	0.95	0.74

Table 4: Comparative performance of classifiers based on entropy in the 3 homological dimensions.

Based on the results shown, entropy outperforms total persistence when the three dimensions are considered. More metrics are provided in 7.2 such as the confusion matrix and the ROC-curve for the classifier that achieved the highest accuracy.

Moreover, the features that were more relevant for each classifier are presented in a different histogram for each descriptor also displayed in Figure 7.2. It can be observed that dimension 2 is of great interest for predictions in both features.

5.2.2 Radiomics

In the following table, the testing metrics obtained for radiomics classification are shown. The highest accuracy value is 0.76 which is obtained with the use of SVM. This estimate surpasses the ones obtained with topological descriptors. Nevertheless, it is on par with them.

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
RDM	KNN	0.68	0.67	0.68	0.85	0.67
	GB	0.72	0.75	0.73	0.91	0.72
	XG-B	0.74	0.76	0.76	0.91	0.74
	SVM	**0.76**	0.77	0.76	0.95	0.76

Table 5: Comparative performance of classifiers based on radiomics.

Figures 10 - 11 give a visual oversight of the results obtained with SVM.

On one hand, the elements in the diagonal of the confusion matrix are close to 10 which is the number of subjects for each class. This suggests that most of the patients are correctly classified.

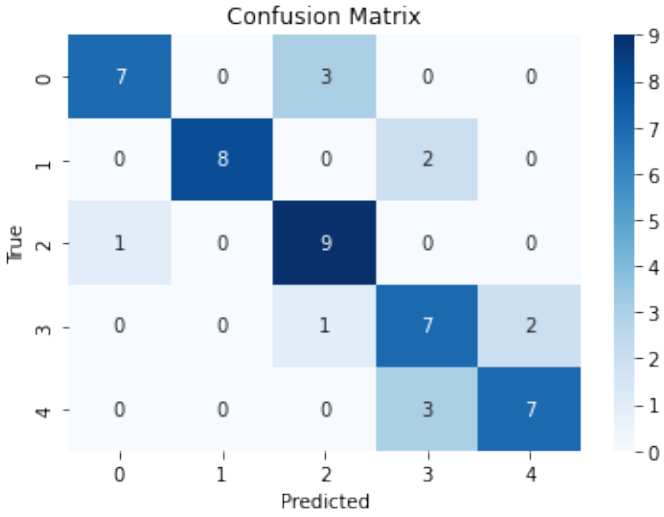

Figure 10: Confusion matrix obtained for SVM classification with radiomics

On the other hand, it is important to mention that the macro-average ROC curve has an AUC close to 1, which indicates that the classifier can notably distinguish between the positive and negative classes with high accuracy.

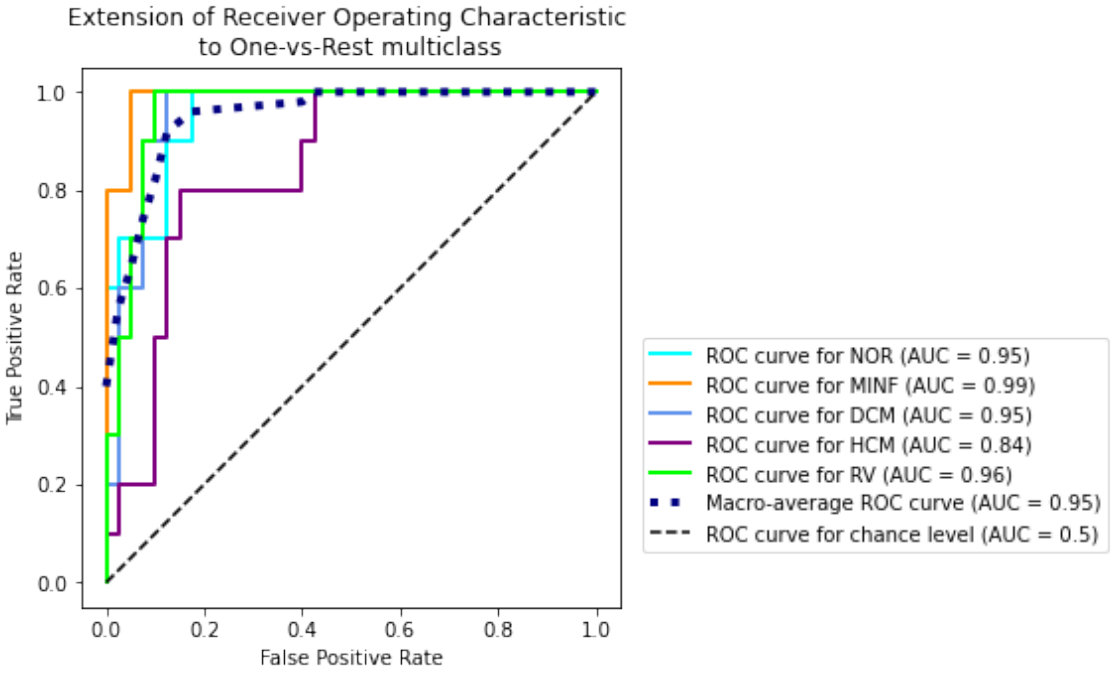

Figure 11: ROC curves obtained for SVM classification with radiomics

When analyzing the top features for the different classifiers in radiomics classification (Figure 12), it can be observed that all of them regard shape.

This analysis of feature importance is carried out by computing the 5 features that contributed best for minimizing the empirical risk for each of the classifiers. Subsequently, a histogram is plotted so as to see the graphical representation of the descriptors that provided the most relevant information.

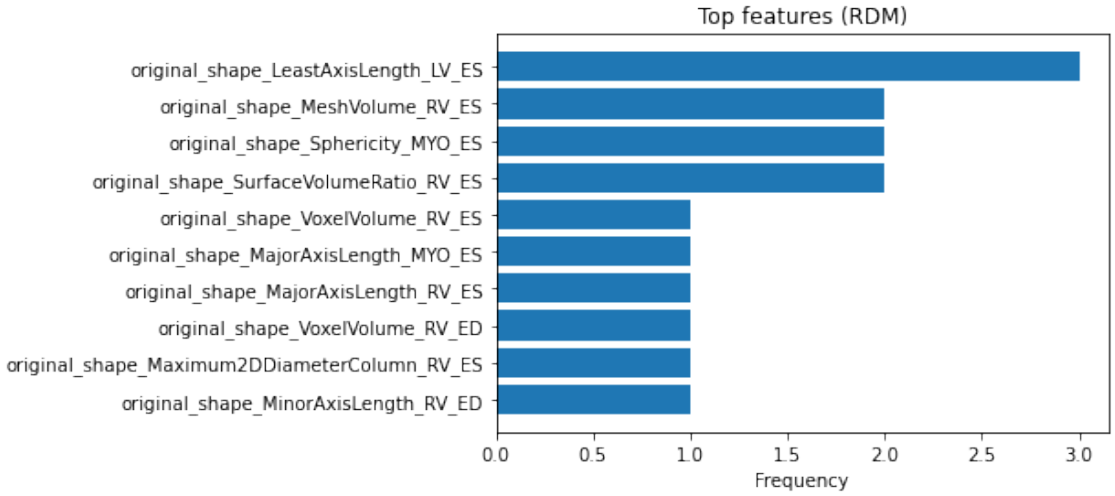

Figure 12: Feature importance for the SVM model using radiomics features.

5.2.3 Radiomics and entropy combination

The following table presents the metrics of classification when combining radiomics with entropy. It is essential to emphasize that the results obtained with Gradient Boosting are comparable to the output of radiomics classification.

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
RDM + ENTR	KNN	0.72	0.73	0.71	0.90	0.72
	GB	**0.76**	0.79	0.77	0.90	0.76
	XG-B	0.74	0.76	0.74	0.92	0.74
	SVM	0.74	0.76	0.75	0.96	0.74

Table 6: Comparative performance of models based on radiomics and entropy features.

When comparing with the confusion matrix obtained with radiomics features, it can be noted that the elements in the diagonal are the same although there are slight changes in the others which leads to a minor divergence between the precision and recall.

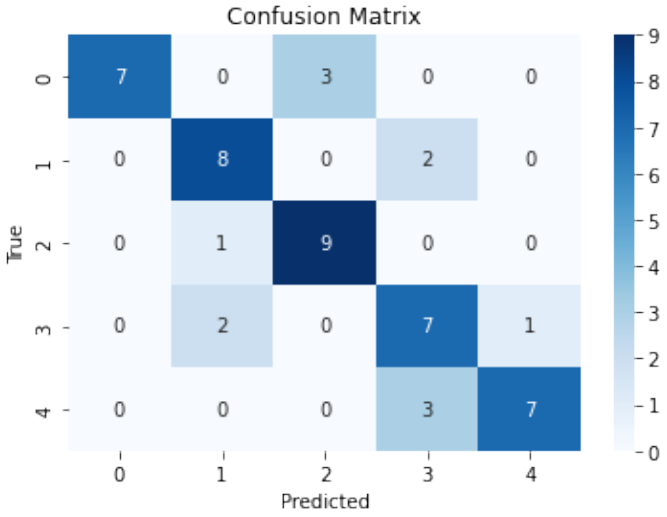

Figure 13: Confusion matrix obtained for GB classification with radiomics and entropy

In terms of the ROC-AUC, it is scarcely lower than the previous value, as can be observed from the fact that the ROC curve exhibits comparatively reduced values.

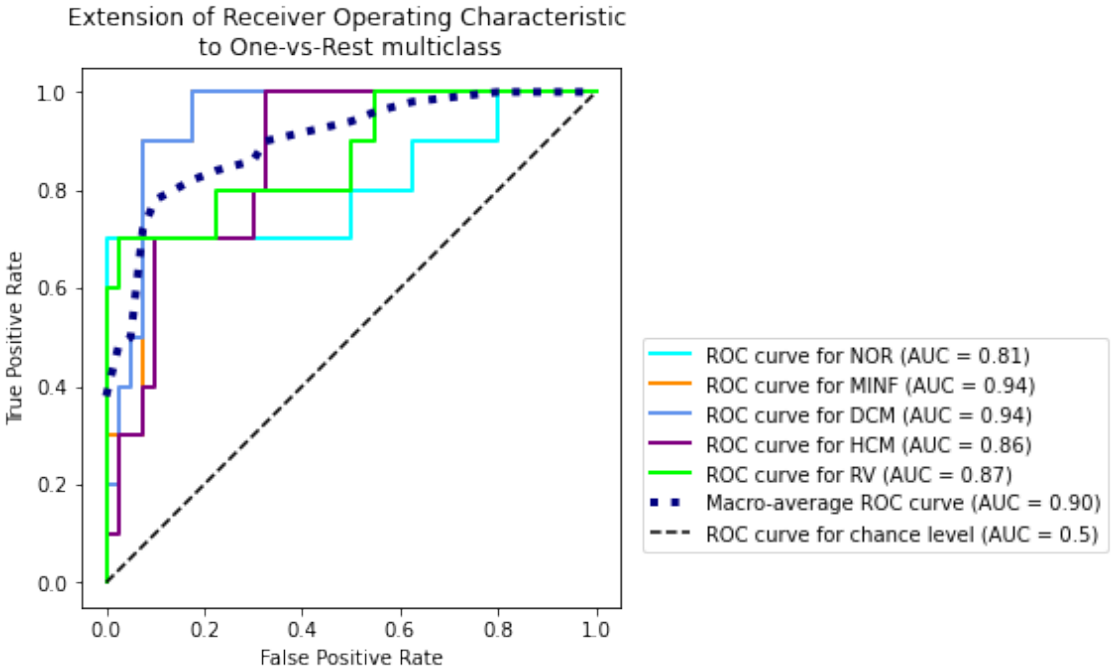

Figure 14: ROC curves obtained for GB classification with radiomics and entropy

15 exhibits the most important features for classification, which comprise mostly radiomics descriptors as well as the entropy for dimensions 0 and 1.

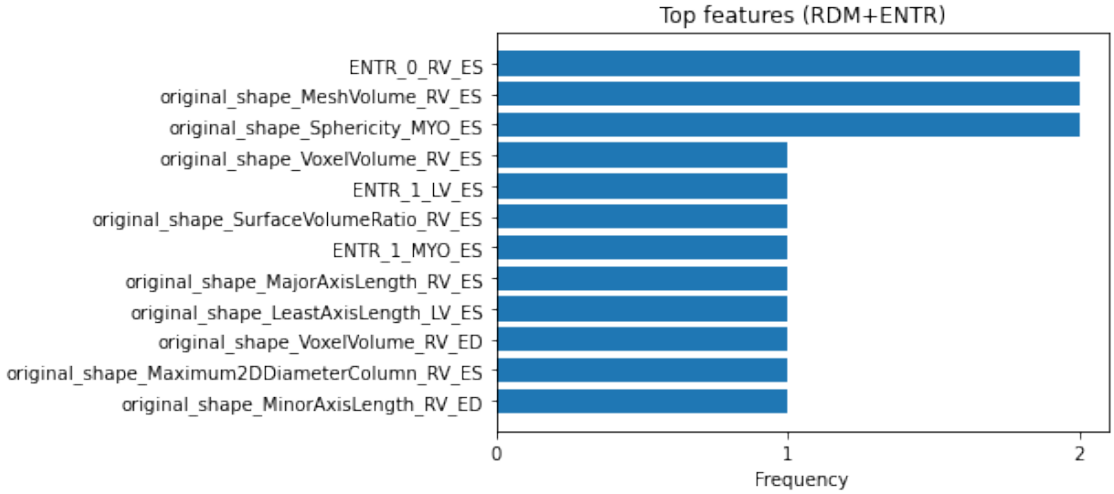

Figure 15: Feature importance for the GB model using radiomics and entropy

5.2.4 Radiomics and total persistence combination

Lastly, the results of combining radiomics with total persistence are presented in Table 7. It is crucial to highlight that with Gradient Boosting an accuracy of 0.79 was achieved. This value is three % greater than the obtained using only radiomics features.

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
RDM + TP	KNN	0.70	0.73	0.71	0.9	0.7
	GB	**0.79**	0.79	0.78	0.91	0.78
	XG-B	0.68	0.71	0.68	0.92	0.68
	SVM	0.76	0.81	0.77	0.96	0.76

Table 7: Comparative performance of models based on radiomics and total persistence

From Figure 16, it can be observed that all patients with previous myocardial infarction (MINF) have been classified correctly. However, the classifier exhibits low performance when classifying subjects with dilated cardiomyopathy (DCM).

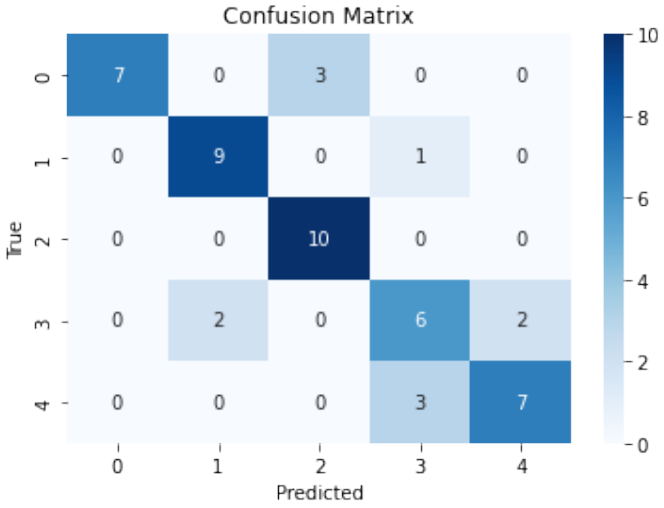

Figure 16: Confusion matrix obtained for GB classification with radiomics and total persistence

The aforementioned disparity in performance between patients with MINF and DCM can also be observed by comparing the two ROC curves.

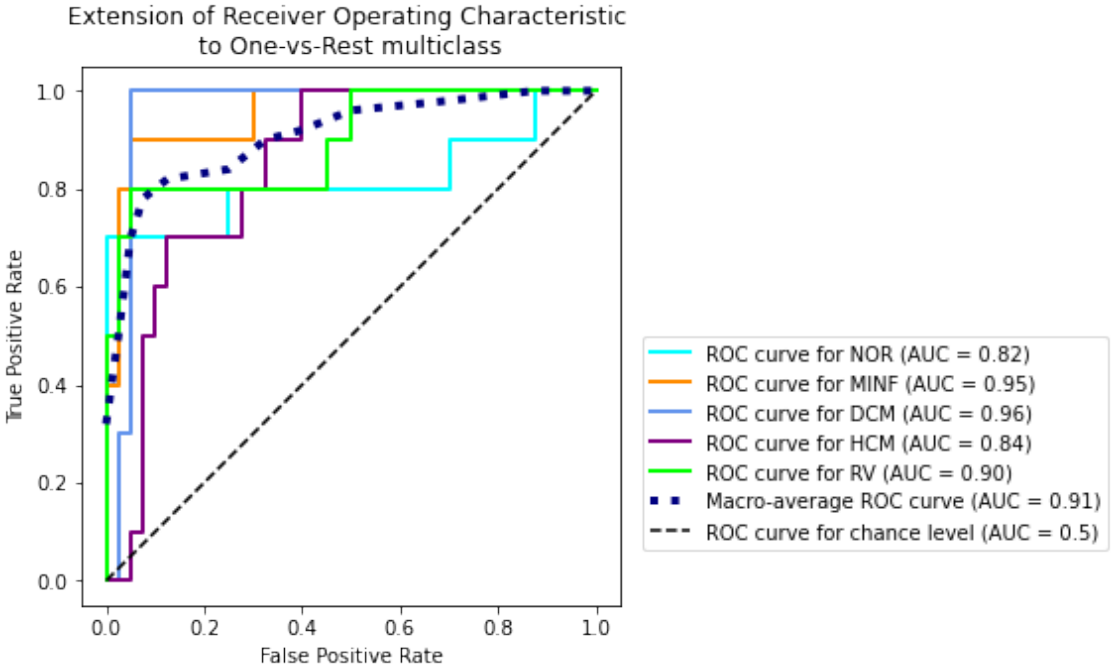

Figure 17: ROC curves obtained for GB classification with radiomics and total persistence

In this particular scenario, it is notable that topological descriptors exhibit greater relevance in the classification process compared to entropy (Figure 17). While topological features primarily capture texture characteristics, the selected radiomics features predominantly describe shape properties. Consequently, the combination of these two types of descriptors offers a full insight into the anatomy of the heart.

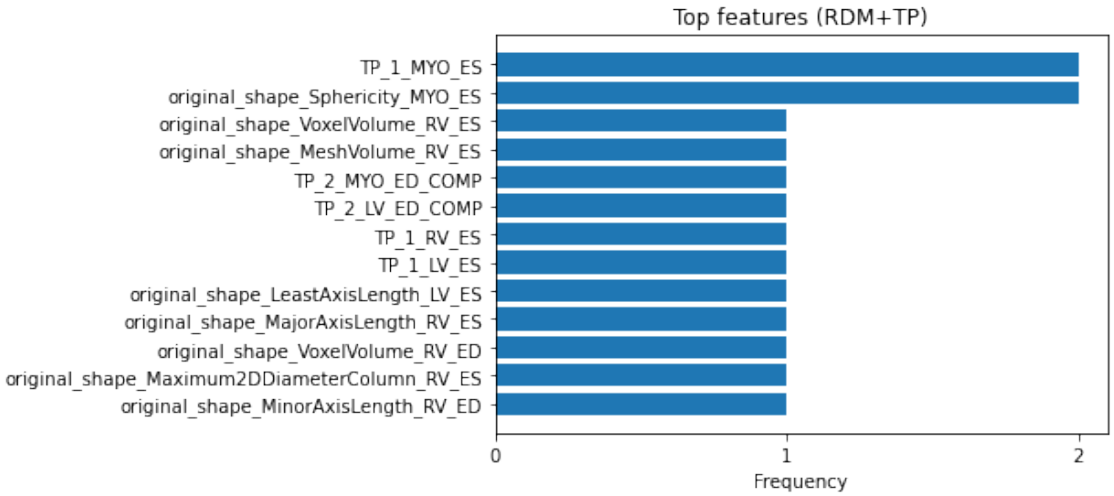

Figure 18: Feature importance for the GB model using radiomics and total persistence

5.2.5 Overall analysis

The table below presents the best classification results obtained for each feature type. The first row corresponds to the radiomics (RDM), which achieve an accuracy of 0.76.

On the other hand, the maximum accuracy achieved with topological descriptors (TDA) is 0.74 and corresponds to the case where entropy was considered in homological dimensions 0, 1, and 2 using the most optimal filtration type for each dimension.

Lastly, we observe that the combination of topological descriptors and radiomics achieves an accuracy of 0.79. This case corresponds to using as features total persistence in all 3 dimensions with their respective optimal filtration type combined with radiomics.

Therefore, it can be concluded that by adding topological descriptors to radiomics, the performance of the classifier is improved. The explanation for this phenomenon is that topological descriptors provide a measure of texture quantification, while the selected radiomics measure the shape of the heart. This is the reason why its combination yields superior results compared to their individual performances.

	Accuracy	Precision	Recall
RDM	0.76	0.77	0.76
TDA	0.74	0.74	0.73
RDM + TDA	**0.79**	**0.79**	**0.78**

Table 8: General outcomes of classification

6 Discussion

6.1 Summary and conclusions

This work focuses on evaluating the potential of topological data analysis for improved cardiac disease diagnosis. The hypothesis of this study is that CMR images contain valuable information that remains hidden even for state-of-the art methods, such as radiomics, which involves acquiring and analyzing a large number of quantitative features from medical images. It was postulated that this information could be extracted in the form of topological features. Consequently, a pipeline was developed for extracting topological descriptors from 3-dimensional CMR images and performing subsequent classification.

The study initiates by establishing a comprehensive theoretical foundation, offering to the reader a profound understanding of the methodologies and tools employed in this work. Within this theoretical framework, state-of-the art topological analysis and machine learning techniques are presented. First of all, the notion of persistent homology, which forms the backbone of the analysis, is introduced. As the study progresses, concepts such as cubical complexes and filtrations are introduced, providing the necessary tools to define persistence diagrams. These persistence diagrams then serve as a crucial basis for introducing topological descriptors that will be utilized for classification purposes.

The practical part starts with an in-depth exposition of the ACDC dataset, which serves as the dataset for conducting the experiments. Subsequently, a detailed explanation is presented on the extraction of topological descriptors, with a specific focus on the observed variations depending on the applied filtration type. Following that, a comprehensive analysis is presented for the classifiers that will be employed in this study. This analysis explores the intricacies of each classifier, providing a thorough examination of their strengths, weaknesses, and performance characteristics. Afterwards, the image preprocessing techniques applied to the CMR images are elaborated upon, ensuring that the input data is optimized for subsequent analysis and classification procedures.

Additionally, a range of metrics that support the analysis is outlined. The rationale behind the selection of these metrics is presented, along with their interpretation and significance in assessing the performance of the classifiers. This comprehensive examination of metrics establishes a solid foundation for evaluating and comparing the classification outcomes.

Finally, the most outstanding results are provided: the addition of topological descriptors to radiomics enhances the performance of the classification task resulting in a three 3% increase in accuracy. This improvement can be attributed to the complementary nature of these two types of descriptors, where topological features

capture texture information and the selected radiomics quantify the shape of the heart. The fusion of these features yields superior results, surpassing the performance attained by using each descriptor type individually.

We hope that this work may have a meaningful impact on cardiac disease diagnosis, as well as on the readers' understanding of topological data analysis. Certainly, we encourage readers to delve into the theory of persistent homology and explore its promising applications in the field of medicine.

6.2 Future work

During the realization of the project, other approaches were considered, but due to time limitations could not be pursued. This section aims to shed light on these alternative approaches and providing directions for future research.

6.2.1 Topological features

In the present study, the topological features extracted are total persistence and entropy. However, other topological descriptors could be tested. An example could be Betti numbers or attributes obtained from landscapes, such as the area below.

6.2.2 Convolution with filters

In [6] a new topological featurization of d-dimensional images is used and proved to extend the capacity of topology to observe patterns in images. Therefore, convolving the CMR images with various filters before extracting the topological features, may improve the classification task.

6.2.3 UKBiobank dataset

The UK Biobank (UKBB) [2] is a significant global health resource comprising data from $500,000$ individuals aged between 40 and 69 years, collected during the period of $2006 - 2010$. This extensive investigation offers a comprehensive understanding of various medical conditions including cancer, cardiovascular diseases, stroke, diabetes, arthritis, osteoporosis, ocular disorders, depression, and diverse forms of dementia. Experts worldwide have utilized this dataset to advance strategies in prevention, diagnosis and treatment.

For a more thorough and comprehensive analysis, the procedures explained in the present work could be implemented with the UKBB dataset. Owing to the extensive magnitude of the dataset, the reliability and robustness of the methodology would be effectively demonstrated. Moreover, the distinct cardiovascular included in the UKBB would also represent an advantage in terms of generizability of the pipeline

followed.

6.2.4 Improvements in the base classifier

Ensembling refers to the combination of multiple classifiers so as to reach a better overall performance than the obtained when using a single model. Its objective is to improve prediction accuracy as well as reducing overfitting and providing more robust results.

In addition, advanced feature selection techniques could be explored as well.

6.2.5 Neural networks

As pointed out in [22] and [23] the performance of deep convolutional neural networks has been proved to be high in determining the presence of diseases by means of medical images. Therefore, another promising approach would be to use deep learning with the topological features extracted.

However, the interpretability of results obtained in deep learning can indeed be more difficult to understand compared to traditional machine learning methods. This drawback arises from several factors.

Deep learning models are characterized by their complex architectures and numerous layers, which can make it challenging to interpret the specific factors or features that contribute to the model's predictions. Additionally, deep learning models often involve a high number of parameters, making it more difficult to analyze the relationships between input variables and output predictions.

In contrast, traditional machine learning algorithms, such as those employed in this work, often have more transparent and interpretable models that allow for a clearer understanding of how inputs are mapped to outputs.

6.3 Programming details and code

The project was developed in *Python* programming language using the version 3.9.12 and the code was executed by means of *Jupyter Notebook*.

So as to compute the persistence diagrams of the CMR images, the library *Giotto-tda* was used. As it is described in its documentation, it is a high performance topological machine learning toolbox built on top of *Scikit-learn* and distributed under the $GNU AGPL v3$ license as a part of the *Giotto* family of open-source projects.

The code used to carry out the practical part of the thesis can be found in the following *Google Colab* page:

Code link.

[2]https://www.ukbiobank.ac.uk/

References

[1] Salch A, et al. (2021) From mathematics to medicine: A practical primer on topological data analysis (TDA) and the development of related analytic tools for the functional discovery of latent structure in fMRI data. PLoS ONE 16(8): e0255859. https://doi.org/10.1371/journal.pone.0255859

[2] Munch, E. (2017). A User's Guide to Topological Data Analysis. Journal of Learning Analytics, 4(2), 47–61. https://doi.org/10.18608/jla.2017.42.6

[3] Yushkevich PA, et al. User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. Neuroimage. 2006 Jul 1;31(3):1116-28. doi: 10.1016/j.neuroimage.2006.01.015. Epub 2006 Mar 20. PMID: 16545965.

[4] Bernard O, et al. Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved? IEEE Trans Med Imaging. 2018 Nov; 37(11):2514-2525. doi: 10.1109/TMI.2018.2837502. Epub 2018 May 17. PMID: 29994302.

[5] Izquierdo C, et al. (2021) Radiomics-Based Classification of Left Ventricular Non-compaction, Hypertrophic Cardiomyopathy, and Dilated Cardiomyopathy in Cardiovascular Magnetic Resonance. Front. Cardiovasc. Med. 8:764312. doi: 10.3389/fcvm.2021.764312.

[6] Solomon, Elchanan and Bendich, Paul. (2022). A Convolutional Persistence Transform. 10.48550/arXiv.2208.02107.

[7] Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems (2nd ed.). O'Reilly.

[8] Shalev-Shwartz, S. & Ben-David, S. (2014). Understanding Machine Learning - From Theory to Algorithms. Cambridge University Press. ISBN: 978-1-10-705713-5

[9] Selesnick, I. (2012). Total variation denoising (an MM algorithm). NYU Polytechnic School of Engineering Lecture Notes, 32.

[10] Igual L. and Seguí Santi. (2017). Introduction to data science : a python approach to concepts techniques and applications. Springer. https://doi.org/10.1007/978-3-319-50017-1

[11] F. Pérez-García, R. Sparks, and S. Ourselin. TorchIO: a Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. Computer Methods and Programs in Biomedicine (June 2021), p. 106236. ISSN: 0169-2607.doi:10.1016/j.cmpb.2021.106236

[12] C. Martín-Isla, et al. Image-based cardiac diagnosis with machine learning: a review. Front. Cardiovasc. Med., 24 January 2020 Sec. Cardiovascular Imaging. https://doi.org/10.3389/fcvm.2020.00001

[13] Wilkins, E., et al. (2017). European Cardiovascular Disease Statistics 2017. European Heart Network. http://www.ehnheart.org/images/CVD-statistics-report-August-2017.pdf

[14] Hannah Ritchie, Fiona Spooner and Max Roser (2018) - "Causes of death". Published online at OurWorldInData.org Retrieved from: https://ourworldindata.org/causes-of-death

[15] O. Bernard, A. Lalande, C. Zotti, F. Cervenansky, et al. "Deep Learning Techniques for Automatic MRI Cardiac Multi-structures Segmentation and Diagnosis: Is the Problem Solved ?" in IEEE Transactions on Medical Imaging, vol. 37, no. 11, pp. 2514-2525, Nov. 2018 doi: 10.1109/TMI.2018.2837502

[16] Kaczynski, T., Mischaikow, K. and Mrozek, M. (2004). Computational homology (Vol. 157). Springer Science and Business Media.

[17] Gudhi Documentation. (n.d.). Cubical Complex. Retrieved from {https://gudhi.inria.fr/doc/latest/group__cubical__complex.html}

[18] Bastian Rieck, Tristan Yates, Christian Bock, Karsten Borgwardt, Guy Wolf, Nicholas Turk-Browne, and Smita Krishnaswamy. 2020. Uncovering the topology of time-varying fMRI data using cubical persistence. In Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20). Curran Associates Inc., Red Hook, NY, USA, Article 579, 6900–6912.

[19] A. Mohammadi et al., "Lung Cancer Radiomics: Highlights from the IEEE Video and Image Processing Cup 2018 Student Competition [SP Competitions]," in IEEE Signal Processing Magazine, vol. 36, no. 1, pp. 164-173, Jan. 2019, doi: 10.1109/MSP.2018.2877123.

[20] Edelsbrunner, Herbert, and John L. Harer. Computational topology: an introduction. American Mathematical Society, 2022.

[21] Leygonie, Jacob, Steve Oudot, and Ulrike Tillmann. "A framework for differential calculus on persistence barcodes." Foundations of Computational Mathematics 22.4 (2022): 1069-1131.

[22] Yadav, S.S., Jadhav, S.M. Deep convolutional neural network based medical image classification for disease diagnosis. J Big Data 6, 113 (2019). https://doi.org/10.1186/s40537-019-0276-2

[23] Chan HP, Samala RK, Hadjiiski LM, Zhou C. Deep Learning in Medical Image Analysis. Adv Exp Med Biol. 2020;1213:3-21. doi: 10.1007/978-3-030-33128-3_1. PMID: 32030660; PMCID: PMC7442218.

[24] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. arXiv preprint arXiv:1603.02754. doi:10.48550/arXiv.1603.02754

[25] Jain, A. (2016, March 1). Mastering XGBoost Parameter Tuning: A Complete Guide with Python Codes. Analytics Vidhya. Retrieved May 19, 2023, from https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xg

[26] Natekin A, Knoll A. Gradient boosting machines, a tutorial. Front Neurorobot. 2013 Dec 4;7:21. doi: 10.3389/fnbot.2013.00021. PMID: 24409142; PMCID: PMC3885826.

[27] Jain, A. (2016, February 21). Complete Machine Learning Guide to Parameter Tuning in Gradient Boosting (GBM) in Python. Retrieved June 15, 2022, from https://www.analyticsvidhya.com/blog/2016/02/complete-guide-parameter-tuning-gr

[28] Chazal, F., & Michel, B. (2021). An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. February 26, 2021.

[29] Romanelli, M. N., Clark, A. K., & Lofaro, O. M. (2021). Potential therapeutic effects of phytocannabinoids in multiple sclerosis: Focus on cannabidiol. Pharmacological Research, 173, 105848. https://doi.org/10.1016/j.phrs.2021.105848

[30] Patel VL, Shortliffe EH, Stefanelli M, et al. The coming of age of artificial intelligence in medicine. Artif Intell Med. 2009 May;46(1):5-17. doi: 10.1016/j.artmed.2008.07.017. Epub 2008 Sep 13. PMID: 18790621; PMCID: PMC2752210.

[31] Tortora, G. J., & Derrickson, B. H. (2018). Principles of Anatomy and Physiology (15th ed.). John Wiley & Sons.

[32] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.

[33] Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.

[34] Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies, 2(1), 37-63.

[35] Gonzalez, R., & Fittes, B. (1975). The 2nd Conference on Remotely Manned Systems (RMS): Technology and Applications. Gray-Level Transformations for Interactive Image Enhancement. Los Angeles, CA (2020) [Conference Proceedings].

7 Appendix

7.1 Comparison between lower and upper filtration

In this section, the results of the testing metrics obtained when comparing the lower-star filtration and the upper-star filtration are shown in different tables. Each of the tables represents the comparison between the performance of a topological feature in a specific dimension.

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
TP_0_lower	KNN	0.43	0.53	0.42	0.73	0.45
	GB	0.46	0.52	0.45	0.77	0.46
	XG-B	0.46	0.43	0.44	0.74	0.43
	SVM	**0.48**	0.52	0.49	0.76	0.49
TP_0_upper	KNN	0.48	0.53	0.47	0.73	0.50
	GB	0.36	0.41	0.35	0.72	0.38
	XG-B	0.42	0.46	0.41	0.71	0.42
	SVM	**0.50**	0.53	0.49	0.81	0.50

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
TP_1_lower	KNN	0.38	0.44	0.37	0.71	0.40
	GB	0.46	0.49	0.47	0.74	0.46
	XG-B	**0.52**	0.52	0.51	0.76	0.52
	SVM	0.46	0.51	0.45	0.78	0.47
TP_1_upper	KNN	0.46	0.49	0.45	0.75	0.45
	GB	**0.50**	0.54	0.51	0.78	0.50
	XG-B	0.42	0.44	0.41	0.76	0.42
	SVM	0.42	0.45	0.41	0.80	0.40

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
TP_2_lower	KNN	0.44	0.43	0.44	0.71	0.43
	GB	**0.50**	0.54	0.56	0.80	0.50
	XG-B	0.45	0.48	0.43	0.79	0.45
	SVM	0.39	0.40	0.38	0.83	0.38
TP_2_upper	KNN	0.46	0.44	0.45	0.82	0.43
	GB	0.48	0.53	0.49	0.79	0.49
	XG-B	**0.58**	0.60	0.57	0.86	0.59
	SVM	0.46	0.43	0.46	0.84	0.44

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
ENTR_0_lower	KNN	0.54	0.53	0.56	0.80	0.54
	GB	0.58	0.59	0.57	0.82	0.57
	XG-B	**0.64**	0.67	0.64	0.84	0.64
	SVM	0.52	0.51	0.53	0.83	0.51
ENTR_0_upper	**KNN**	**0.50**	0.53	0.49	0.72	0.50
	GB	0.42	0.42	0.43	0.77	0.42
	XG-B	0.47	0.42	0.43	0.77	0.42
	SVM	0.42	0.43	0.41	0.76	0.42

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
ENTR_1_lower	KNN	0.44	0.45	0.44	0.71	0.43
	GB	**0.56**	0.58	0.55	0.79	0.56
	XG-B	0.44	0.45	0.44	0.75	0.44
	SVM	0.44	0.43	0.43	0.80	0.43
ENTR_1_upper	KNN	0.42	0.44	0.42	0.77	0.41
	GB	**0.50**	0.56	0.51	0.76	0.52
	XG-B	0.46	0.49	0.45	0.77	0.46
	SVM	0.49	0.52	0.51	0.82	0.50

Feature	Model	Acc	Prec	Rec	AUC ROC	F1
ENTR_2_lower	KNN	0.52	0.51	0.50	0.85	0.48
	GB	**0.54**	0.53	0.53	0.81	0.54
	XG-B	0.48	0.47	0.48	0.80	0.47
	SVM	0.53	0.53	0.54	0.88	0.52
ENTR_2_upper	KNN	0.48	0.41	0.48	0.84	0.44
	GB	0.44	0.41	0.43	0.78	0.42
	XG-B	0.46	0.42	0.44	0.79	0.43
	SVM	**0.60**	0.58	0.59	0.88	0.58

7.2 ROC curves, confusion matrices and top features for total persistence and entropy classification

ROC curves and confusion matrices corresponding to the classifier that outperformed the others are displayed below.

In the case of total persistence, the selected classifier is k-nearest neighbors while in the case of entropies the SVM classifier performed best.

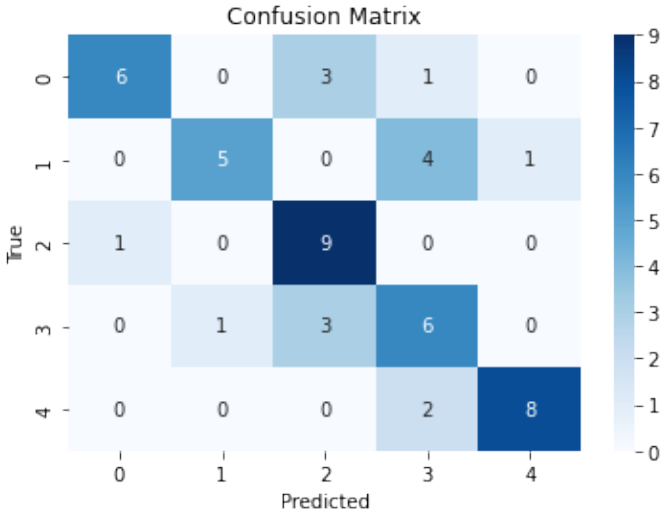

Figure 19: Confusion matrix for classification with KNN and total persistences as features

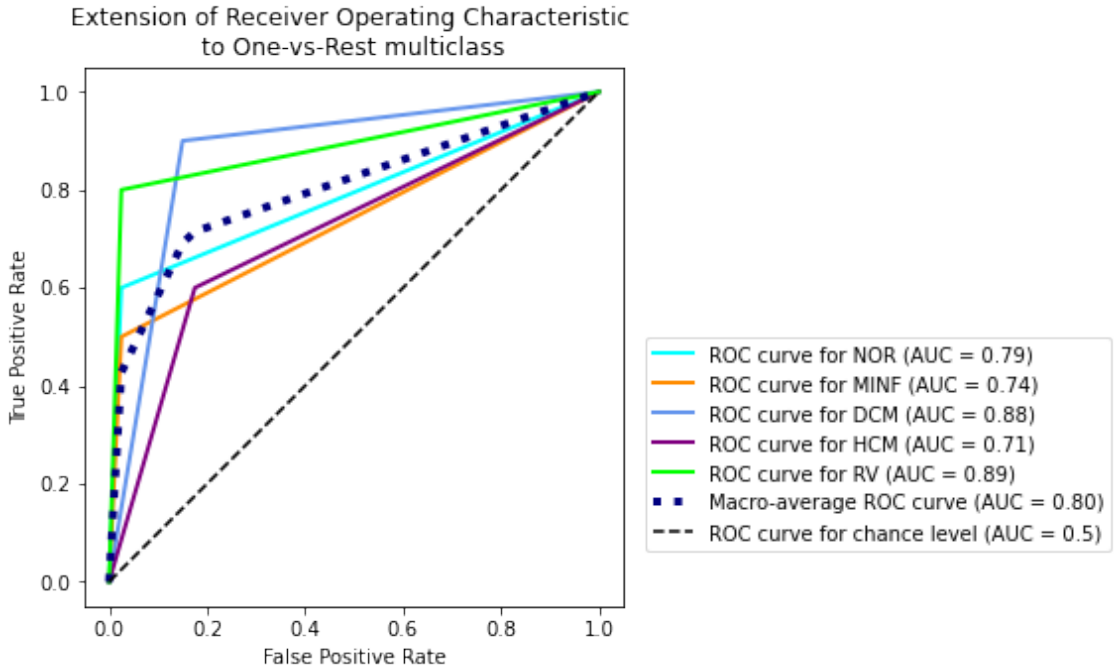

Figure 20: ROC curves for classification with KNN and total persistences as features

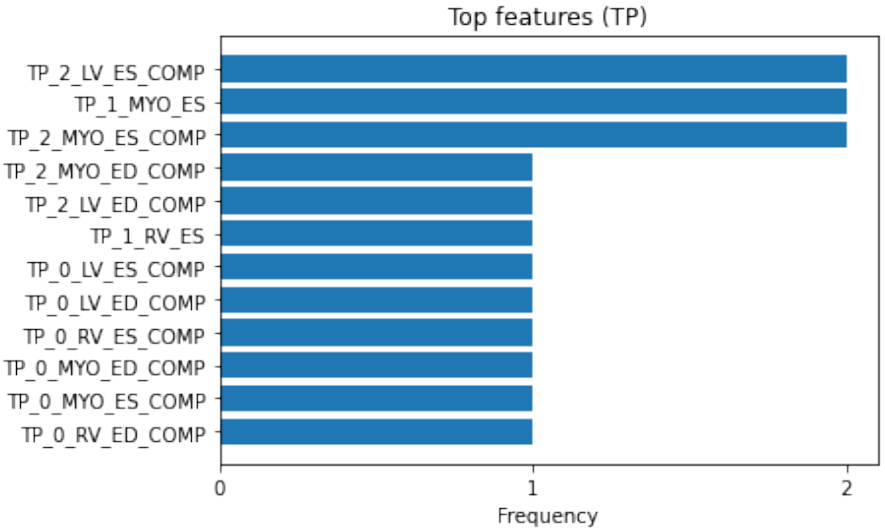

Figure 21: Top features for classification with KNN and total persistence

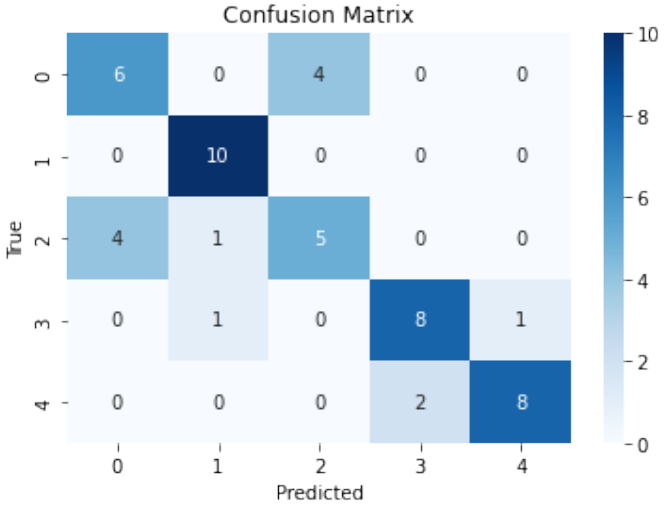

Figure 22: Confusion matrix for classification with SVM and entropies as features

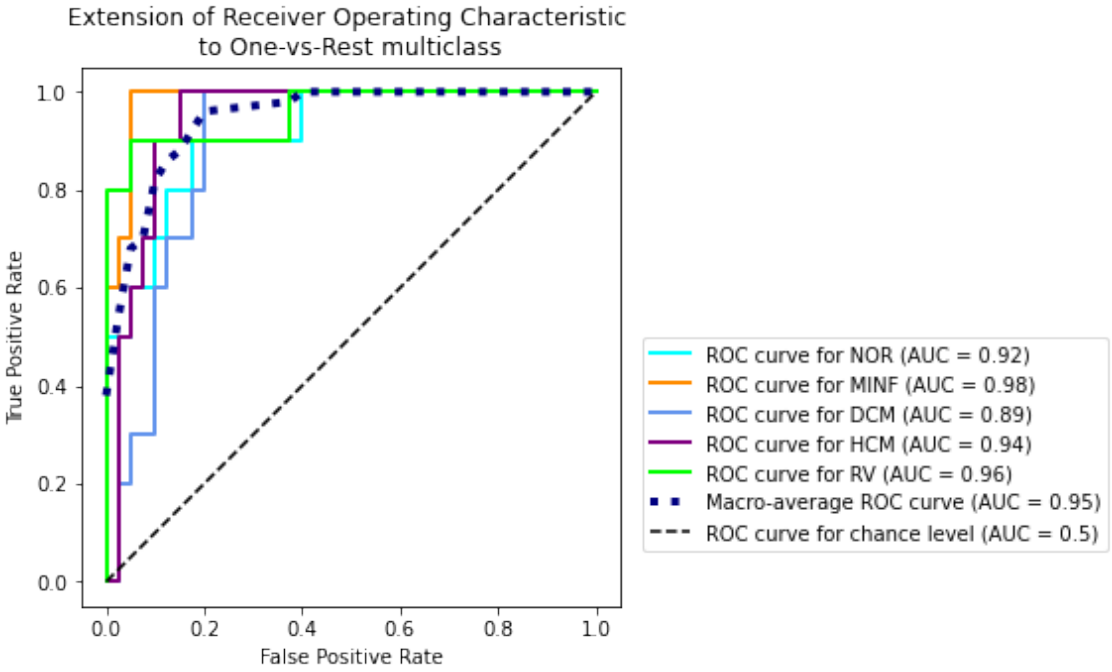

Figure 23: ROC curves for classification with SVM and entropies as features

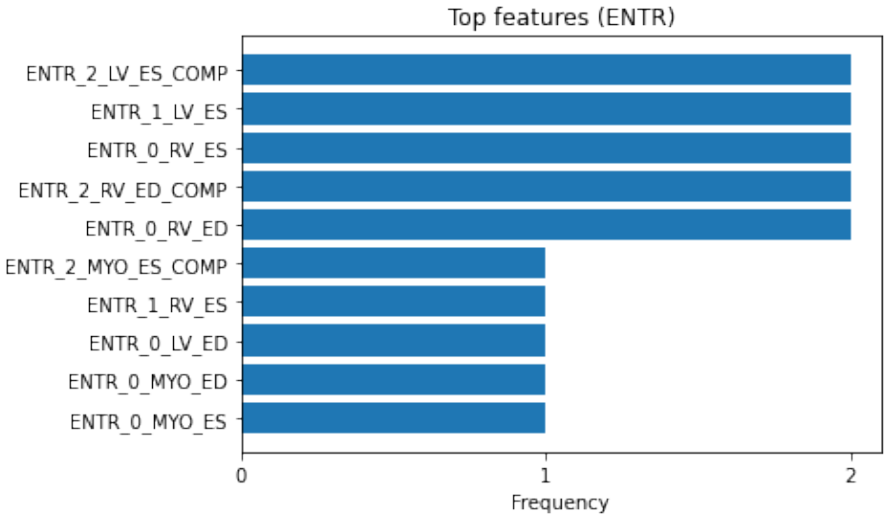

Figure 24: Top features for classification with SVM and entropy