
CLASSIFICATION BASED ON TOPOLOGICAL DATA ANALYSIS

Rolando Kindelan

Computer Science Department
 Faculty of Mathematical and Physical Sciences
 University of Chile
 851 Beauchef Av. Santiago de Chile, Chile.
 Center of Medical Biophysics,
 Universidad de Oriente, Santiago de Cuba, Cuba.
 rkindela@dcc.uchile.cl

José Frías

Center for Research in Mathematics
 Jalisco S/N, Col. Valenciana
 CP: 36023 Guanajuato, Gto., México
 friaas@matem.unam.mx

Mauricio Cerdá

Integrative Biology Program
 Institute of Biomedical Sciences
 Biomedical Neuroscience Institute
 Center for Medical Informatics and Telemedicine
 Faculty of Medicine
 Universidad de Chile
 1027 Independencia Av., Santiago, Chile.
 mauricio.cerda@uchile.cl

Nancy Hitschfeld

Computer Science Department
 Faculty of Mathematical and Physical Sciences
 University of Chile
 851 Beauchef Av. Santiago de Chile, Chile
 nancy@dcc.uchile.cl

February 9, 2021

ABSTRACT

Topological Data Analysis (TDA) is an emergent field that aims to discover topological information hidden in a dataset. TDA tools have been commonly used to create filters and topological descriptors to improve Machine Learning (ML) methods. This paper proposes an algorithm that applies TDA directly to multi-class classification problems, even imbalanced datasets, without any further ML stage. The proposed algorithm built a filtered simplicial complex on the dataset. Persistent homology is then applied to guide choosing a sub-complex where unlabeled points obtain the label with most votes from labeled neighboring points. To assess the proposed method, 8 datasets were selected with several degrees of class entanglement, variability on the samples per class, and dimensionality. On average, the proposed TDABC method was capable of overcoming baseline classifiers (wk-NN and k-NN) in each of the computed metrics, especially on classifying entangled and minority classes.

1 Introduction

The processing and extraction of information from large and noisy data sets is a challenging problem in Computer Science. The techniques of algebraic topology have gained the attention of scientists for years, giving rise to an emerging research field called Topological Data Analysis (TDA) [8, 14]. TDA is an approach to infer the topology underlying a dataset by using combinatorial algebraic structures known as simplicial complexes. TDA also involves the computation of invariant properties from continuous transformations of these simplicial complexes: a process known as persistent homology [14].

Over several decades, the high dimensionality of datasets coupled with the combinatorial and continuous character of topology have been problematic issues, making computing persistent homology a challenge that has been addressed by several authors. Edelsbrunner et al. [14] present an efficient algorithm and its visualization as a persistence diagram [14, 38]. Carlsson et al. [8] strengthened the mathematical foundations and also proposed another visualization tool called persistence Barcodes [19, 8]. Further developments in the TDA field are derived from those initial works.

As a consequence of their combinatorial nature, the construction and representation of simplicial complexes also represent a challenge. Many works have dealt with efficient construction and representation of filtered simplicial complexes. Data structures and algorithms have been developed [3, 39, 4, 5], and they have mainly focused on the construction of Čech, Rips and other kinds of simplicial complexes such as: Witness, Alpha, Delaunay, Tangent, and Cover complexes. Theoretical and practical results have been organized as TDA libraries: GUDHI [5, 26], Dionysus, Ripser, Diphia, Perseus and JavaPlex. A complete benchmark of those libraries can be found in [28].

Regarding the use of TDA for classification, a TDA-based method was used in [18] for classifying high-resolution diabetic retinopathy images. They used a preprocessing stage for computing persistent homology to detect topological features encoded into persistence diagrams. A support vector machine (SVM) was used to classify the images according to the persistence descriptors which were used to discriminate between diabetic and healthy patients.

Moreover, TDA has been applied to time-series analyses [10]. One common pipeline is to consider the time-series as a dynamic system and compute the attractors or time-variant of the signal, which creates a manifold around the attractors and turns the signal into phase-domain [35, 34]. Persistent Homology or another TDA-tool is applied on these phase-space manifold to create topological descriptors [33], and as a final step, a machine learning method is applied such as k-NN, CNN, or SVM. Recently, TDA has been applied in Deep learning to address the interpretability problem [9], to regularize loss functions [17], and to build a persistence layer to consider topological information during learning [20, 17].

What all those examples of TDA-applications have in common is that TDA has been used as a preprocessing stage of conventional Machine Learning (ML) algorithms. However, during the TDA pipeline execution, multi-scale relationships among data occur and disappear. From the moment when a multi-scale relationship occurs until it is mixed with another one, it is called persistence. The persistence of many of those relationships is captured and represented by persistence diagrams or barcodes. Taking advantage of the entire TDA pipeline and not just the result could help address some of the current challenges of supervised and semi-supervised learning, such as imbalanced data classification, identification, and correction of mislabeled data; missing data analysis; and dimensionality reduction.

In this scenario, this paper proposes a methodology to make a TDA pipeline that is able to classify balanced and imbalanced datasets with no ML further stage. The fundamental idea is to provide neighborhoods on a filtered simplicial complex related to a point set (a simplex) or a point as a special case. Those neighborhoods will be, in fact, a sub-complex of the filtered simplicial complex built on the dataset. Persistent homology is used to guide the detection of an appropriate sub-complex from the entire filtration. A labeling process is then made to propagate labels from labeled points to unlabeled points, taking advantage of the simplicial relationships.

To illustrate, the proposed method is compared with several baseline classifiers. One of the baseline algorithms is the k-NN algorithm, one of the most popular supervised classification methods. The second baseline method is an enhanced version of k-NN, the weighted k-NN (wk-NN) especially suited for imbalanced datasets. This document is organized into several sections. Section 2 presents the mathematical foundations used in this work. Section 3 explains the concepts, algorithms, and methodology of the proposed classification method. Next, Section 4 describes algorithms, datasets, the experimental protocol, evaluation criteria, and the selected metrics to assess the proposed method performance. In Section 5, the results and implementation details of the proposed method are explained. Conclusions are presented in Section 6.

2 Mathematical foundations

In this section, mathematical definitions are introduced (i.e.: simplices, simplicial complex, the Čech and Rips complexes, the star, and link concepts). Concepts such as persistent homology, filtration, sub-complex, and filtration levels are briefly presented. For more detailed definitions, please see [14].

2.1 Simplicial Complexes

Simplicial complexes are combinatorial and algebraic objects which represent a discrete space homotopically equivalent to a data space. Concepts related to simplicial complexes are defined briefly as follows: a q-simplex σ is the convex hull of $q + 1$ affinely independent points $\{s_0, s_1, \dots, s_q\} \subset \mathbb{R}^n$, $q \leq n$. In this case, the set $\mathcal{V}(\sigma) = \{s_0, s_1, \dots, s_q\}$ is called the set of *vertices* of σ and the simplex σ is generated by the set $\mathcal{V}(\sigma)$; this relation will be denoted by $\sigma = [s_0, s_1, \dots, s_q]$. A q-simplex σ has dimension $\dim(\sigma) = q$ and it has $|\mathcal{V}(\sigma)| = q + 1$ vertices. Given a q-simplex σ , a d-simplex τ with $0 \leq d \leq q$ and $\mathcal{V}(\tau) \subseteq \mathcal{V}(\sigma)$ is called a *d-face* of σ , denoted by $\tau \leq \sigma$, and σ is called a *q-coface* of τ , denoted by $\sigma \geq \tau$. Note that the 0-faces of a q-simplex σ are the elements of $\mathcal{V}(\sigma)$, the 1-faces are line segments with endpoints in $\mathcal{V}(\sigma)$ and so forth. A q-simplex has $\binom{q+1}{d+1}$ d-faces and $\sum_{d=0}^q \binom{q+1}{d+1} = 2^{q+1}$ faces in total.

In order to define homology groups of topological spaces, the notion of simplicial complexes is central:

Definition 1. (Simplicial complex): A simplicial complex $\mathcal{K} \subset \mathbb{R}^n$ is a finite collection of simplices such that:

- $\sigma \in \mathcal{K}$ and $\tau \leq \sigma \implies \tau \in \mathcal{K}$.
- $\sigma_1, \sigma_2 \in \mathcal{K} \implies \sigma_1 \cap \sigma_2 \leq \sigma_1$ and $\sigma_1 \cap \sigma_2 \leq \sigma_2$.

The dimension of \mathcal{K} is $\dim(\mathcal{K}) = \max\{\dim(\sigma) \mid \sigma \in \mathcal{K}\}$.

There are many known simplicial complexes, though two of the most popular are the Čech [19, 14] and Vietoris-Rips complexes. In the following definitions the set $B_\varepsilon(x) \subset \mathbb{R}^n$ should denote the open ball of radius ε and centered at x , namely $B_\varepsilon(x) = \{y \in \mathbb{R}^n \mid |x - y| < \varepsilon\}$.

Definition 2. (Čech complex): Let X be a finite subspace of \mathbb{R}^n and fix $\varepsilon > 0$. The Čech complex $\check{C}ech(\varepsilon)$ is a simplicial complex where the vertices or 0-simplices are the elements of X and a set of vertices $\{v_0, v_1, \dots, v_q\}$ define a q -simplex if

$$\bigcap_{i=0}^q B_\varepsilon(v_i) \neq \emptyset$$

Definition 3. (Vietoris-Rips or VR complex): Let X be a finite metric space and fix $\varepsilon > 0$. The Vietoris-Rips complex $VR(\varepsilon)$ is a simplicial complex where the 0-simplices are the elements of X and a set of vertices $\{v_0, v_1, \dots, v_q\}$ define a q -simplex $\sigma = [v_0, v_1, \dots, v_q]$ if $\text{diam}(\sigma) \leq 2\varepsilon$.

From the above definitions it follows that $\check{C}ech(\varepsilon) \subseteq VR(\varepsilon) \subseteq \check{C}ech(2\varepsilon)$, where a proof is given in [14], and this relationship is shown in Figure 1. The Čech complex is intrinsically a high dimensional simplicial complex. From a computational sense, VR complex is more feasible (i.e. lower storage and time complexity) than Čech, even when the VR complex has more simplices in general. Compared to Čech, the VR complex does not need to be stored entirely, as it can be stored as a graph and be reconstituted combinatorially [19]. Even when the results in this paper could be applied to several simplicial complexes with minor changes, this document is focused on Čech and VR complexes.

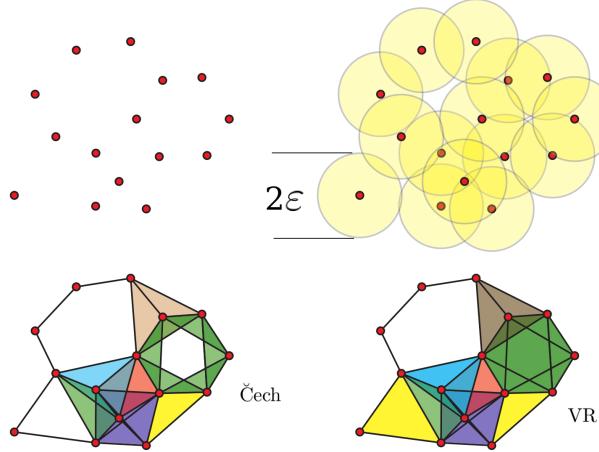


Figure 1: From a point set [upper left] a proximity parameter ε is applied [upper right] and two complexes were built: a Čech complex [lower left] and a VR complex [lower right]. This picture was taken from [19].

Definition 4. (Star, Closure, Closed Star, and Link): Let \mathcal{K} be a simplicial complex, and $\sigma \in \mathcal{K}$ be a q -simplex. The star(σ) of σ is the set of all co-faces of σ in \mathcal{K} [14]:

$$St_{\mathcal{K}}(\sigma) = \{\tau \in \mathcal{K} \mid \sigma \leq \tau\}$$

Let K be a subset of simplices $K \subset \mathcal{K}$. The closure of K is the smallest simplicial complex containing K :

$$Cl_{\mathcal{K}}(K) = \{\mu \in \mathcal{K} \mid \mu \leq \sigma \text{ for some } \sigma \in K\}$$

The $St_{\mathcal{K}}(\sigma)$ is not a simplicial complex because of all the missing faces. The smallest simplicial complex that contains $St_{\mathcal{K}}(\sigma)$ is the closed star (closure of star) of σ :

$$\overline{St}_{\mathcal{K}}(\sigma) = Cl_{\mathcal{K}}(St_{\mathcal{K}}(\sigma))$$

The link(Lk) of σ is a set of simplices in its closed star that does not share any face with σ [14]:

$$Lk_{\mathcal{K}}(\sigma) = \{\tau \in \overline{St}_{\mathcal{K}}(\sigma) \mid \tau \cap \sigma = \emptyset\}$$

The concept of link of a simplex in a simplicial complex will be important along this paper. For this reason we present two equivalent characterizations of this set:

Lemma 1. : Let \mathcal{K} be a simplicial complex and $\sigma \in \mathcal{K}$. Then $Lk_{\mathcal{K}}(\sigma)$ coincide with the sets

$$A = \overline{St}_{\mathcal{K}}(\sigma) \setminus (St_{\mathcal{K}}(\sigma) \cup Cl_{\mathcal{K}}(\sigma)), \text{ and} \quad (1)$$

$$B = \bigcup_{\mu \in St_{\mathcal{K}}(\sigma)} \{[\mathcal{V}(\mu) \setminus \mathcal{V}(\sigma)]\} \quad (2)$$

Proof. Let τ be a simplex in $Lk_{\mathcal{K}}(\sigma)$. In particular, τ does not belong to $St_{\mathcal{K}}(\sigma)$ nor $Cl_{\mathcal{K}}(\sigma)$ since any simplex in one of these two sets necessarily intersects σ , then $Lk_{\mathcal{K}}(\sigma) \subset A$.

If τ is a simplex in A , then there exists $\mu \in St_{\mathcal{K}}(\sigma)$ such that $\tau \leq \mu$ and $(\mathcal{V}(\mu) \setminus \mathcal{V}(\tau)) \subset \mathcal{V}(\sigma)$. It follows that $\tau = [\mathcal{V}(\mu) \setminus \mathcal{V}(\sigma)]$ and $A \subset B$.

Finally, if $\tau \in B$, then $\tau = [\mathcal{V}(\mu) \setminus \mathcal{V}(\sigma)]$ for some $\mu \in St_{\mathcal{K}}(\sigma)$. It follows that $\tau \in \overline{St}_{\mathcal{K}}(\sigma)$, but $\tau \cap \sigma = \emptyset$. Then, $B \subset Lk_{\mathcal{K}}(\sigma)$, and the equivalence of sets is stated. \square

Figure 2 presents an example of St and Lk of point s_4 from a given simplicial complex \mathcal{K} build on a point set S .

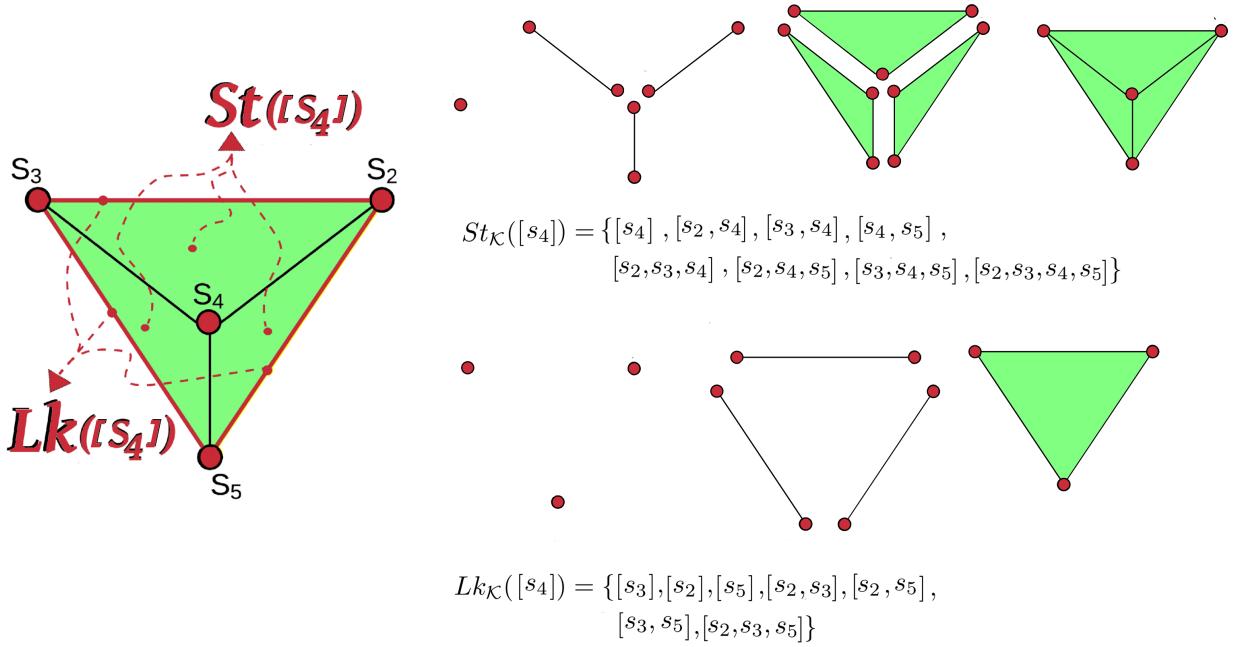


Figure 2: Example of $St_{\mathcal{K}}([s_4])$ and $Lk_{\mathcal{K}}([s_4])$ on a given simplicial complex \mathcal{K} .

2.2 Persistent Homology

Persistent homology is a tool to find topological features in a metric space [14, 8]. As a general rule, the objective of persistent homology is to track how topological features on a topological space appear and disappear when a scale value (usually a radius) varies incrementally, in a process known as filtration [15, 38].

Definition 5. (Sub-complex): Let \mathcal{K} be a simplicial complex. \mathcal{K}' is a sub-complex of \mathcal{K} if $\mathcal{K}' \subseteq \mathcal{K}$ and \mathcal{K}' is also a simplicial complex.

Definition 6. (Filtration): Let \mathcal{K} be a simplicial complex. A filtration \mathcal{F} on \mathcal{K} is a succession of increasing sub-complexes of \mathcal{K} :

$$\emptyset \subseteq \mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \cdots \subseteq \mathcal{K}_n = \mathcal{K}$$

In this case, \mathcal{K} is called a filtered simplicial complex.

In most of simplicial complexes where the simplices are determined by proximity under a distance function (as in the case of Čech or VR complexes), a filtration on a simplicial complex \mathcal{K} is obtained by taking a sequence of positive values $0 < \varepsilon_0 \leq \varepsilon_1 \leq \cdots \leq \varepsilon_n$, and the complex \mathcal{K}_i corresponds to the value ε_i .

Definition 7 (Filtration level function, ψ): Let \mathcal{K} be a finite simplicial complex and \mathcal{F} a filtration on \mathcal{K} : $\emptyset \subseteq \mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \mathcal{K}_3 \subseteq \cdots \subseteq \mathcal{K}_n = \mathcal{K}$. The filtration level function $\psi_{\mathcal{F}}$ is defined on $\{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_n\}$ by $\psi_{\mathcal{F}}(\varepsilon_i) = \mathcal{K}_i$.

A filtration could be understood as a method to build the whole simplicial complex \mathcal{K} from a “family” of sub-complexes incrementally sorted according to some criteria, where each level i corresponds to the “birth” or “death” of a topological feature as described in Definition 10.

This process is illustrated in Figure 3.

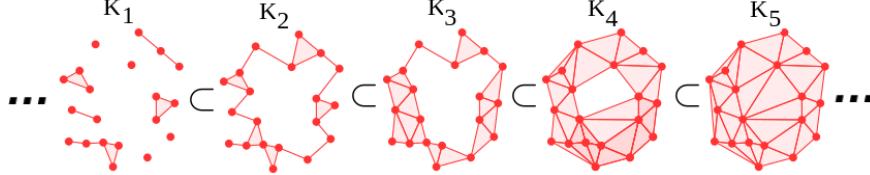


Figure 3: A fragment of a simplicial complex filtration.

Definition 8 (Filtration value collection $\mathcal{E}_{\mathcal{K}}$): Let \mathcal{K} be a filtered simplicial complex and \mathcal{F} a filtration on \mathcal{K} . Let $\mathcal{E}_{\mathcal{K}} = \{\varepsilon_0, \varepsilon_1, \dots, \varepsilon_n\} \subset \mathbb{R}$ be a set of non-negative numbers such that $0 \leq \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon_n$, where $\varepsilon_i \in \mathcal{E}_{\mathcal{K}}$ is a filtration value (radius) applied to build the sub-complex $\mathcal{K}_i \subseteq \mathcal{K}$ in the filtration \mathcal{F} . The set $\mathcal{E}_{\mathcal{K}}$ is called the filtration value collection associated to \mathcal{F} .

Definition 9 (Filtration value of a q-simplex $\xi_{\mathcal{K}}(\sigma)$): Let \mathcal{K} be a filtered simplicial complex and $\mathcal{E}_{\mathcal{K}}$ its filtration value collection. Let $\sigma \in \mathcal{K}$ be a q -simplex. If $\sigma \in \mathcal{K}_j$ but $\sigma \notin \mathcal{K}_i, \forall i < j$, then $\xi_{\mathcal{K}}(\sigma) = \varepsilon_j$ is the filtration value of σ .

Note that $\tau \leq \sigma \implies \xi_{\mathcal{K}}(\tau) \leq \xi_{\mathcal{K}}(\sigma)$, which means that in a filtered simplicial complex \mathcal{K} , every simplex $\tau \in \mathcal{K}$ appears before all its co-faces $\sigma \in \mathcal{K}$.

Definition 10 (Birth and Death). Birth is a concept to describe the filtration level when a new topological feature appears. Similarly, death refers to the filtration level when a topological features disappears. Thus, a persistence interval (birth, death) is the “lifetime” of a given topological feature [14].

3 Proposed Classification Method

Let $P \subset \mathbb{R}^n$ be a finite point set, where \mathbb{R}^n is a feature space. Suppose P is divided in two subspaces $P = S \cup X$, where S is the training set, and X is the test set. Let $L = \{l_1, l_2, \dots, l_N\}$ be the label set, and let $\mathbb{T} = \{(p, l) : p \in P, l \in L\}$ be the association space, which relates every point $p \in P$ with a unique label $l \in L$. Let T_S and T_X be the two disjoint association sets corresponding to S and X , respectively, where $\mathbb{T} = T_S \cup T_X$. In this setting, the real label list, $Y = \{l_i \mid (x_i, l_i) \in T_X\}$, is the list of labels assigned to each element of X in the association set T_X . Thus, the classification problem can be defined as how to predict a suitable label $l \in L$ for every $x \in X$ by assuming the association set T_X as unknown. Consequently, the predicted label list, $\hat{Y} \subset L^{|T_X|}$, will be the resulting collection of labels after classifying each $x \in X$. Since $|Y| = |\hat{Y}|$, it is common to use Y to evaluate the quality of \hat{Y} . Depending on the size of X and S the problem is known as supervised classification ($|X| \leq |S|$), or semi-supervised classification ($|S| < |X|$).

A classification method based on TDA is presented in this section. Overall, a filtered simplicial complex \mathcal{K} is built over $S \cup X$ to generate data relationships. Only a few of those relationships will be relevant relationships between data

points. In this context, a relationship between points is real if this relationship is part of the data's hidden structure. Thus, persistent homology is applied to capture the real structure of the dataset. This information is helpful to detect a subset of relationships likely to be real. The proposed method is based on the supposition that on the filtration a sub-complex $\mathcal{K}_i \subset \mathcal{K}$ exists, whose simplices represent real data relationships. For every q-simplex $\sigma \in \mathcal{K}_i$, the set of vertices of σ , $\mathcal{V}(\sigma)$, will be split into labeled and unlabeled points, where any of these subsets could be empty. The fact that a point set $\{v_0, v_1, \dots, v_q\} \subset S \cup X$ belongs to a q-simplex $\sigma \in \mathcal{K}$ implies a similarity or dissimilarity relationship between points v_0, v_1, \dots, v_q . This implicit relationship among data is applied to propagate labels from labeled points to unlabeled points. Thus, a link-based labeling propagation method is developed to make a suitable label prediction for each point $x \in X$.

3.1 Link-based label propagation function

On a filtered simplicial complex \mathcal{K} , the neighborhood relationships of a q-simplex $\sigma \in \mathcal{K}$ could be recovered by using the link, star, and closed star concepts (Definition 4). A key component of the proposed method is the label propagation over a filtered simplicial complex. Given a simplicial complex \mathcal{K} , a separation between useful simplices, and not-useful simplices (see Definition 11) needs to be considered. This simplicial classification is helpful because useful-simplices contribute to discriminate more labeling information during the propagation and label assignment process. In this way, those sub-complexes $\mathcal{K}_i \in \mathcal{K}$ with an appropriate distribution of useful-simplices will be preferred.

Definition 11. (useful-simplex, and non-useful-simplex)

Let \mathcal{K} be a simplicial complex built on $S \cup X$, and $\sigma \in \mathcal{K}$ be a q-simplex. We say σ is a useful-simplex if it contains more elements from S than elements from X . In another case, then σ is a non-useful-simplex.

3.1.1 The labeling function

Let \mathcal{K} be a finite simplicial complex built on $S \cup X$, and \mathcal{F} be a filtration on \mathcal{K} : $\emptyset \subseteq \mathcal{K}_0 \subseteq \mathcal{K}_1 \subseteq \mathcal{K}_2 \subseteq \dots \subseteq \mathcal{K}_n = \mathcal{K}$. Suppose a preferred subcomplex \mathcal{K}_i in the filtration \mathcal{F} has been selected. Let A be the \mathbb{R} -module with generators $\hat{l}_1, \hat{l}_2, \dots, \hat{l}_N$. The generator \hat{l}_j will be associated to the label l_j according to the following definition:

Definition 12 (Association function). *Let $\Phi_i : \mathcal{K}_i \rightarrow A$ be the association function defined on a 0-simplex $v \in \mathcal{K}_i$ as $\Phi_i(v) = \hat{l}_j$ if $(v, l_j) \in T_S$ and $\Phi_i(v) = 0 \in A$ in any other case. The association function can be extended to a q-simplex σ as $\Phi_i(\sigma) = \sum_{v \in \mathcal{V}(\sigma)} \Phi_i(v)$.*

As an intermediate step to propagate labels from labeled points in S to the unlabeled points in X by means of the link operation in simplicial complexes, define the extension function in X as follows:

Definition 13 (Extension function). *Let $\Psi_i : X \rightarrow A$ be the function defined on a point $v \in X$ by*

$$\Psi_i(v) = \sum_{\sigma \in Lk_{\mathcal{K}_i}([v])} \frac{1}{\xi_{\mathcal{K}}(\sigma)} \cdot \Phi_i(\sigma) \quad (3)$$

In Equation 3, for every q-simplex $\sigma \in Lk_{\mathcal{K}_i}([v])$, the filtration value $\xi_{\mathcal{K}}(\sigma)$ is applied to prioritize the influence of σ to label v . Let $\alpha, \beta \in Lk_{\mathcal{K}_i}([v])$ be two simplices, such that $\xi_{\mathcal{K}}(\alpha) < \xi_{\mathcal{K}}(\beta)$. This condition implies that the vertices of α cluster around v earlier than the vertices of β do since they were added first to the filtration. In consequence, α contributions should be more important than β contributions.

According to the previous definitions, given a point $v \in X$, the evaluation of the extension function at v would be $\Psi_i(v) = \sum_{j=1}^N a_j \cdot \hat{l}_j$, where $a_j \in \mathbb{R}^+ \cup \{0\}, j = 1, \dots, N$.

Definition 14 (Labeling function). *Let v be a point in X such that $\Psi_i(v) = \sum_{j=1}^N a_j \cdot \hat{l}_j$. If \tilde{a} is the maximum value in $\{a_j\}_{j=1}^N$, define the labeling function Υ_i at v as $\Upsilon_i(v) = l_k$ where k is uniformly selected from the set $\{j \mid a_j = \tilde{a}\}$.*

If there exists a unique maximum in the set $\{a_j\}_{j=1}^N$ from the previous definition, then the labeling function is uniquely defined at v . In most of datasets where the proposed TDA classification method was tested, the label assignment of each point in X was uniquely defined. Figure 4 shows the labeling process on a previously selected sub-complex.

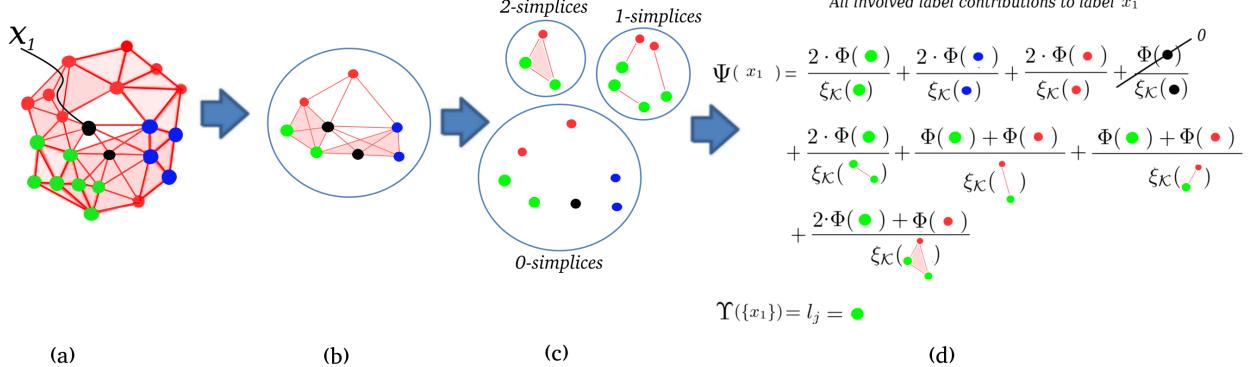


Figure 4: Example of execution of the labeling function on unlabeled points x_1 . In (a), a sub-complex $\mathcal{K}_i \subseteq \mathcal{K}$ is presented where the labeling will be executed. The neighborhood of x_1 is drawn in (b). In (c), $Lk([x_1])$ is shown divided into 0-simplices, 1-simplices, and 2-simplices. In (d), the extension function $\Psi(x_1)$ is executed, and finally the labeling function $\Upsilon(x_1)$ assigns a green label to x_1 . The $\xi_{\mathcal{K}}$ function disambiguates label contributions, which at the beginning seems to be a tie between labels.

3.2 Classification by using simplicial complexes and persistent homology

The proposed method computes the predicted label list \hat{Y} corresponding to X . The entire process is summarized in Algorithm 1. Following subsections explain each step in detail.

Algorithm 1 TDABC: TDA-Based Classification Algorithm

Require: A training set $S \neq \emptyset$.

A test set $X \neq \emptyset$ to be classified.

The incomplete association set T' .

Ensure: A prediction list \hat{Y} of X by using T' .

1: A filtered simplicial complex \mathcal{K} is constructed by using the Algorithm 2.

2: Obtain the prediction list $\hat{Y} = (l_1, l_2, \dots, l_{|X|})$, where each $l_i \in \hat{Y}$ is the most reliable label corresponding to $x_i \in X$, with $1 \leq i \leq |X|$ by using \mathcal{K} (Algorithm 4)

3: **return** the prediction list \hat{Y} .

3.2.1 Building the filtered simplicial complex.

Let S and X be two point sets, where S is the labeled set, and X is the unlabeled set. The filtered simplicial complex \mathcal{K} is built on $P = S \cup X$. A maximal dimension $2 \leq q \ll |P|$ is given to control the simplicial complex combinatorial growing. Algorithm 2 illustrates this process.

Algorithm 2 Construction of a filtered simplicial complex \mathcal{K}

Require: A non-empty training set S . A non-empty test set X . An $0 < \epsilon \leq 1$ increment value to change the filtration level. Let n be the maximum desired level of the filtration on \mathcal{K} .

Ensure: a filtered simplicial complex \mathcal{K} .

```

1: Unify  $S$  and  $X$  by  $P \leftarrow S \cup X$ 
2:  $i \leftarrow 0$ 
3:  $\varepsilon_i \leftarrow 0$ 
4:  $\mathcal{K} \leftarrow \{\}$ 
5: while  $i \leq n$  do
6:   Build a simplicial complex  $\mathcal{K}_i$  from  $P$  by using an  $\varepsilon_i$  value, with  $\dim(\mathcal{K}_i) \leq q$ , according to [14, 3, 5, 4]
7:   if  $\mathcal{K} \neq \mathcal{K}_i$  then
8:      $\mathcal{K} \leftarrow \mathcal{K} \cup \mathcal{K}_i$ 
9:      $i \leftarrow i + 1$ 
10:     $\varepsilon_i \leftarrow \varepsilon_{i-1} + \epsilon$ 
11:   else
12:      $i \leftarrow n$ 
13:   end if
14: end while
15: return  $\mathcal{K}$ 
```

3.2.2 Obtaining the most reliable label

Once a filtered simplicial complex \mathcal{K} is obtained, all the elements of X need to be labeled. Because of the full connectivity on \mathcal{K} , it is highly likely that all the vertices are interconnected. Then $\dim(\mathcal{K}) = N - 1$, where $N = |P|$, and the number of simplices in \mathcal{K} would be 2^N .

Therefore, if functions $\Psi_n(x)$ and $\Upsilon_n(x)$ from Definition 13 and 14 are computed using $\mathcal{K}_n = \mathcal{K}$, it could occur that a given point $x \in X$ has contributions of all possible labels, due to the high number of co-faces of the analyzed simplices. In this context, it is worth understanding which simplices of \mathcal{K} containing some $x \in X$ are reliable or which are noise to perform the label propagation and labeling. Then, the purpose is to choose a sub-complex \mathcal{K}_i from the filtered simplicial complex \mathcal{K} , such that: i) \mathcal{K}_i constitutes a good approximation to the real structure of the dataset, and ii) there are enough useful simplices in \mathcal{K}_i to label each point $x \in X$. In this vein, persistent homology is used to guide the process of selecting \mathcal{K}_i , reduce the classification space, and guarantee the useful simplices inside the selected sub-complex.

With persistence homology, multi-dimensional topological features are detected. For a filtered simplicial complex \mathcal{K} of dimension q , persistent homology will compute up to q -dimensional homology groups. Each topological feature represented by an element in a homology group of a given dimension will be represented by a persistence interval $(\text{birth}, \text{death}) \in \mathbb{R}^2$ (see Definition 10).

The collection of simplices that shapes one topological feature belongs to a homology class. However, it is no trivial to recover information about every simplex belonging to that homology class, except by the simplex generator of that topological feature. For example, a 0-simplex connects to another 0-simplex and creates a 1-simplex. The 1-simplex, which joins together two 1-simplices, creates a hole. A 2-simplex is attached to the other three 2-simplices and becomes a 3-simplex, leaving a void inside. Precisely, the moment when this happens is the birth of a topological feature and the death of another topological feature of a lower dimension. As it can be noticed, the relation between simplices and topological features is injective, and it is only related to the generator. Thus, only the generator of a topological feature might be recovered from a persistence interval at its birth.

Based on the assumption that any simplex is related to only one persistence interval, this relation will persist from its birth to its death. Therefore, all persistence intervals which have intersected their lifetimes will have their corresponding related simplices coexisting during the intersection time. Eventually, when a homology class dies, their connection with simplices also dies, and they are no longer considered, at least not directly. A conventional way to get a well-defined membership relation from simplices to a homology class is to look at the simplices associated with the birth of a homology class, which are the only simplices that can reliably be associated with the homology class.

The challenge is to find appropriate homology classes to ask for their associated simplices. It is known that long life invariants ($\text{high } \text{death} - \text{birth}$) represent topological features, while short life invariants are commonly considered noise. However, short life persistence intervals could also mean local topological features or high dimensional topological features, which could be more profitable in useful-simplices. On this scenario, persistence intervals will be considered from higher homology groups ($q - 1$) downwards (see Algorithm 3).

Persistent homology is then considered to recover topological features which represent meaningful data relationships. Some of those topological features hopefully will represent (in their birth) a level of filtration \mathcal{K}_i that maximizes the number of useful-simplices associated with each $x \in X$. As a result, it is highly likely that we obtain a reliable label for every $x \in X$.

Persistent homology is computed according to [14, 15, 11, 12], and a collection $\mathbb{D} = \{D^0, D^1, \dots, D^q\}$ is obtained with D^i the persistence interval set of the i^{th} -dimensional homology group. Algorithm 3 computes a persistence interval set $D \in \mathbb{D}$, which is the non-empty homology group with higher dimension.

Algorithm 3 GetPersistenceIntervalSet: Computing the persistence interval set D

Require: A filtered simplicial complex \mathcal{K} .

Ensure: A persistence interval set D , where:

- 1: $D \leftarrow \{d_i \mid d_i = (\text{birth}, \text{death})\}$.
 - 2: $\mathbb{D} \leftarrow \text{ComputePersistentHomology}(\mathcal{K})$ [11, 15] with $\mathbb{D} = \{D^0, D^1, \dots, D^q\}$, D^i the persistence interval set of the i^{th} -dimensional homology groups.
 - 3: $D \leftarrow \{\}$
 - 4: **while** $D == \emptyset$ and $i \geq 1$ **do**
 - 5: $D \leftarrow D^i$ with $D^i \in \mathbb{D}$
 - 6: $i \leftarrow i - 1$
 - 7: **end while**
 - 8: **return** D
-

Let $d \in D$ be a persistence interval. Then $\text{life}(d) = d[\text{death}] - d[\text{birth}]$. We notice that $\text{life}(d)$ becomes undefined for immortal topological feature (i.e. infinite death time). To overcome this issue, it is enough to change the death time from infinite to the maximum of the filtration value collection $\mathcal{E}_{\mathcal{K}}$. We call this a ϑ -transformation (see Equation 4). Thus, a new function $\text{int}(d) = \text{life}(\vartheta(d))$ is defined to apply the ϑ -transformation before $\text{life}(d)$ is called.

$$\vartheta(d) = \begin{cases} d & \text{iff } d[\text{death}] \neq \infty, \\ (d[\text{birth}], \max(\mathcal{E}_{\mathcal{K}})) & \text{iff } d[\text{death}] = \infty. \end{cases} \quad (4)$$

A desired persistence interval $d \in D$ is selected by using the (naive) functions defined on Equation 5, Equation 6, and Equation 7:

(a) The maximum persistence interval:

$$d_m = \text{MaxInt}(D) = \arg \max_{d \in D} (\text{int}(d)). \quad (5)$$

(b) A persistence interval selected in a random way:

$$d_r = \text{RandInt}(D) = \vartheta(\text{random}(D)). \quad (6)$$

(c) The closest interval to the persistence intervals average:

$$d_a = \text{AvgInt}(D) = \arg \min_{d \in D} |\text{int}(d) - \text{avg}(D)|, \quad (7)$$

where $\text{avg}(D) = \frac{1}{|D|} \cdot \sum_{d_i \in D} \text{int}(d_i)$.

Although homology groups are different, the birth and death of persistence intervals values are absolutes. Even when those persistence intervals that contain high dimensional invariants were selected, low dimensional topological features are not necessarily excluded. In Algorithm 4, a persistence interval d is selected from a filtration, to recover a sub-complex $\mathcal{K}_i \subset \mathcal{K}$ and classify all $x \in X$.

Because of the injectivity between simplices and birth time of persistence intervals, the sub-complex $\mathcal{K}_{d[\text{birth}]} \subset \mathcal{K}$ might be selected. Nevertheless, the sub-complexes $\mathcal{K}_{\frac{d[\text{birth}]+d[\text{death}]}{2}} \subset \mathcal{K}$ and $\mathcal{K}_{d[\text{death}]} \subset \mathcal{K}$ could be selected as well. In these cases, the middle time ($\frac{d[\text{birth}]+d[\text{death}]}{2}$) and death time of persistence interval d will capture all those simplices which are generators of topological features still alive (or born) on the middle and death times. The choice between birth, middle, or death time to select the most appropriate sub-complex seems related to the homology group

dimension. When the selected homology group has a high dimension, $\mathcal{K}_{d[birth]}$ gives good precision on classification. On the other hand, if a 0-dimensional homology group is selected, the sub-complex $\mathcal{K}_{d[death]}$ should be the best choice. The sub-complex $\mathcal{K}_{\frac{d[birth]+d[death]}{2}}$ could be also selected on 1-dimensional homology groups. As a result, the birth time was chosen (see Algorithm 4) to select the sub-complex because it is always guaranteed to be present.

Figure 5 shows the selection of a sub-complex $\mathcal{K}_i \subseteq \mathcal{K}$, where \mathcal{K} is a filtered simplicial complex built on the Circles dataset (with noise = 10), one of the artificial datasets used to evaluate the proposed method in Section 4. The selection of \mathcal{K}_i is guided by the persistent homology information and the application of $MaxInt(\cdot)$, and $RandInt(\cdot)$ selection functions (see Equation 5, and Equation 6) to select an appropriate persistence interval according to each criterion. Note that $RandInt(\cdot)$ was coincident with $AvgInt(\cdot)$, and the results of only one persistence interval were shown in Figure 5.

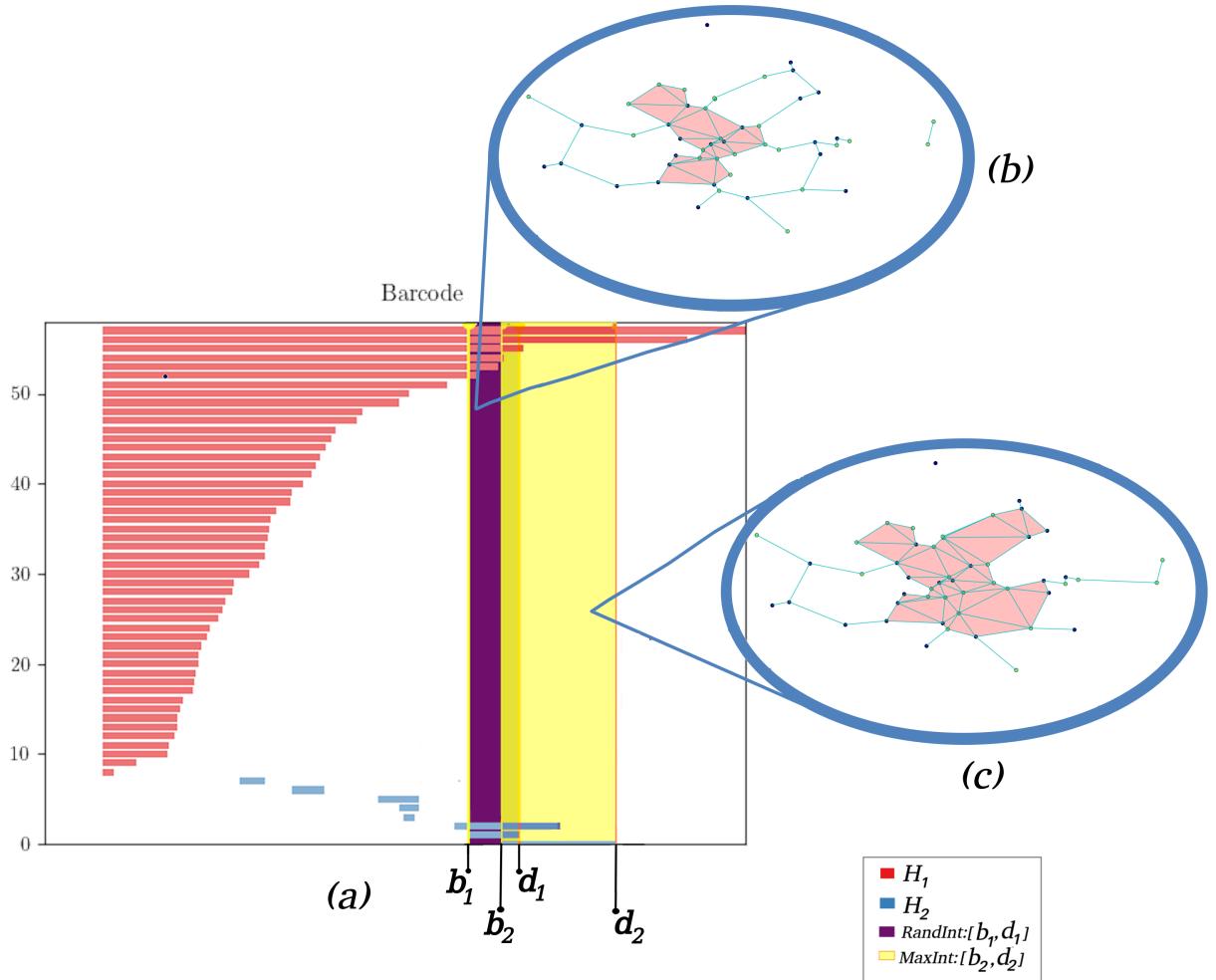


Figure 5: A filtered simplicial complex \mathcal{K} was built on the Circles dataset, detailed in Section 4.2.1. Persistent homology was computed to guide the sub-complex selection. A barcode representation of persistent homology is shown in (a), with two selected persistence intervals $[b_1, d_1]$ (purple), and $[b_2, d_2]$ (yellow) corresponding to the $RandInt(\cdot)$; and $MaxInt(\cdot)$ selection functions respectively. In (b), the sub-complex $\mathcal{K}_{d_1} \subseteq \mathcal{K}$, and $\mathcal{K}_{d_2} \subseteq \mathcal{K}$ is shown in (c).

Algorithm 4 Labeling: Labeling a test set X .**Require:** A filtered simplicial complex \mathcal{K} by a filtration \mathcal{F} .A non-empty test set X .**Ensure:** A predicted labels list \hat{Y} of X .

- 1: $D \leftarrow \text{GetPersistenceIntervalSet}(\mathcal{K})$ where:
 $D = \{d_i \mid d_i = (\text{birth}, \text{death})\}$
- 2: Get a desired persistence interval d where:
 $d \in \{\text{MaxInt}(D), \text{RandInt}(D), \text{AvgInt}(D)\}$
- 3: $\varepsilon_i \leftarrow d[\text{birth}]$
- 4: $\mathcal{K}_i \leftarrow \psi_{\mathcal{F}}(\varepsilon_i)$ see Definition 7
- 5: $\hat{Y} \leftarrow \{\}$
- 6: **while** $X \neq \emptyset$ **do**
- 7: $x \in X$
- 8: $l \leftarrow \Upsilon_i(x)$ see Definition 13
- 9: $\hat{Y} \leftarrow \hat{Y} \cup \{l\}$
- 10: $X \leftarrow X \setminus \{x\}$
- 11: **end while**
- 12: **return** \hat{Y}

An overview of the proposed method is presented in Figure 6. To classify two black points $x_1, x_2 \in X$, a 4-steps process is executed.

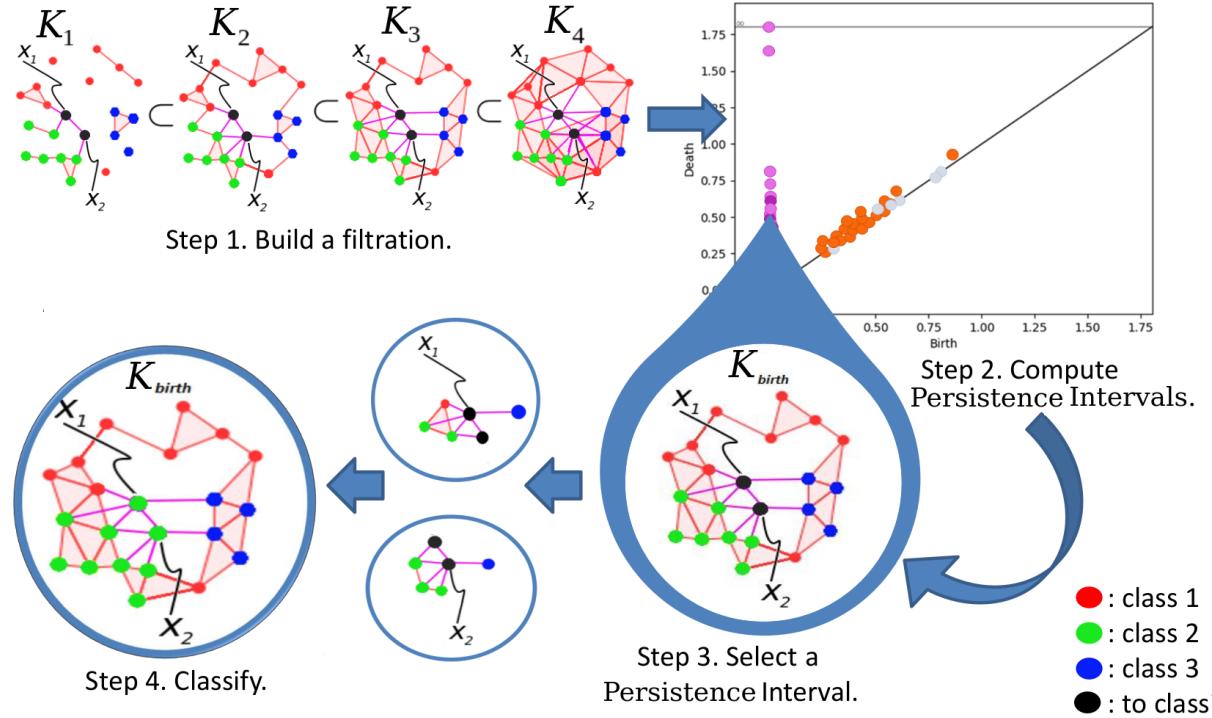


Figure 6: Overall TDABC algorithm (see Algorithm 1). The first step is to build a filtered simplicial complex \mathcal{K} , following Algorithm 2. On the second step, persistent homology is computed to recover topological features according to [11] and [12]. The third step consists of applying Algorithm 3 to compute a persistence interval by using Equation 5, 6, and 7. A sub-complex \mathcal{K}_i is then recovered from the filtration by using the birth of the selected interval. Fourth and last, the classification step uses a label propagation approach and takes the most voted label (Algorithm 4 and Definition 14).

3.3 Implementation

The proposed TDA-based classifier (TDABC) was implemented on top of the GUDHI library [26, 4, 3], which is one of the most complete libraries for building simplicial complexes [25] and computing homology groups [4, 28, 3, 5].

3.3.1 Simplicial complex construction and persistence computation

The first step of the proposed TDABC method (see Figure 6) is building a filtered simplicial complex \mathcal{K} on $S \cup X$. For datasets with high dimensions or datasets with too many samples, the implementation of the Algorithm 2 in GUDHI could be impractical due to combinatorial complexity. Consequently, the combinatorial complexity of the simplicial complex must be reduced. To address this problem, the approach followed in this paper is using the edge collapse [32] method on the GUDHI library. The collapse of edges in GUDHI has to be performed on the 1-skeleton of the simplicial complex and then expand to build all high dimensional simplices up to a maximal dimension $q \ll |S \cup X|$. Algorithm 5 computes a simplex tree by using the edge collapsing method. A collapsing coefficient depends on the maximal dimension q , but it could be enhanced by invoking *collapse_edges* repeatedly until the simplex tree does not change any more.

Algorithm 5 Build a simplex tree with edge collapsing

Require: A non-empty point set $P = S \cup X$.
Ensure: An updated simplex tree \mathcal{S} .

- 1: $\mathcal{K} \leftarrow \text{buildSC}(P)$, create a simplicial complex on P up to dimension 1.
- 2: $\mathcal{S} \leftarrow \text{create_simplex_tree}(\mathcal{K}, \text{max_dim} = 1)$
- 3: $c \leftarrow \lceil \frac{q}{3} \rceil$ { a simple coefficient of edge collapsing }
- 4: $\mathcal{S} \leftarrow \text{collapse_edges}(\mathcal{S}, c)$
- 5: $\mathcal{S} \leftarrow \text{expansion}(\mathcal{S}, \text{max_dim} = q)$
- 6: **return** \mathcal{S}

3.3.2 Persistence computation and persistence interval selection

In GUDHI, instead of persistent homology, persistent cohomology is computed using the algorithm [11] and [12] and the Compressed Annotation Matrix data structure implementation presented in [2]. Due to homology and cohomology's duality both methods compute the same homological information, but cohomology provides richer topological information [11]. Algorithm 3's implementation in GUDHI is direct, using the *persistence*(\cdot) method of the simplex tree data structure.

3.3.3 Label propagation implementation

The extension function Ψ from Definition 13 depends on the $Lk_{\mathcal{K}}$ operation from Definition 4. Up to now the python interface of GUDHI Library (v.3.3.0) [31] does not have an implementation of the simplex link operation; however, it provides implementations of star and co-face operators. As a result, a link operation function was implemented based on Equation 2 from Lemma 1.

According to Lemma 1, Ψ function from Definition 13 could be implemented based on $St_{\mathcal{K}}$. In addition, two ways of removing the σ contributions are possible: a strict way and a belated one. The first method is computing strictly $Lk_{\mathcal{K}}$ using Lemma 1. The advantage of this way is reducing the quantity of invocations of the association function $\Phi(\sigma)$ in Definition 12. In the second way, the $St_{\mathcal{K}}(\sigma)$ function is used as a whole, and the σ contributions are then ignored during function $\Phi(\sigma)$ execution because it would be 0 for unknown 0-simplices. Lemma 1 and Definition 13 show that both approaches are equivalent.

In GUDHI, each q -simplex $\sigma \in \mathcal{K}$ represented by a simplex tree \mathcal{S} is related to its filtration value $\xi_{\mathcal{K}}(\alpha)$. Thus, the $star(\mathcal{S}, \sigma)$ is a function in \mathcal{S} which returns a 2-tuple set $\{(\mu, \xi_{\mathcal{K}}(\mu)) \mid \mu \in St_{\mathcal{K}}(\sigma)\}$. This facilitates the implementation of function Ψ and the recovery of the ε -values to impose a priority over simplices and minimize a tie.

4 Results

The proposed TDA-based classifier (TDABC) is sensible to the chosen selection function $RandInt(\cdot)$, $MaxInt(\cdot)$, and $AvgInt(\cdot)$. Those selection functions can detect a specific sub-complex in the filtered simplicial complex built on the dataset. Due to this dependency, the proposed method's behavior needs to

explored by using those functions. Consequently, three versions of TDABC methods are configured to assess the proposed solutions:

- (i) The TDABC-R classification method using the $\text{RandInt}(\cdot)$ selection function.
- (ii) The TDABC-M, because of the utilization of the $\text{MaxInt}(\cdot)$ selection function.
- (iii) The TDABC-A, which uses the $\text{AvgInt}(\cdot)$ selection function.

4.1 Selected baseline classifiers

Three baseline methods were selected to compare the proposed methods:

- (i) The k-Nearest Neighbors (k-NN) implementation from Scikit-Learn [29] was chosen.
- (ii) The distance-based weighted k-NN from Scikit-Learn was also selected to assess the proposed methods.

4.2 Datasets

Several data sets were chosen to evaluate the proposed methods and compare them to the baseline classifiers. Table 1 shows the datasets with some of their characteristics.

Name	Dimensions	Classes	Size	Samples per class	Noise	Mean	Stdev
Circles	2	2	50	[25,25]	3	-	-
Moon	2	2	200	[100,100]	10	-	-
Swissroll	3	6	300	[50,50,50,50,50,50,]	10	-	-
Normdist	350	5	300	[60,10,50,100,80]	-	[0,0.3,0.18,0.67,0]	0.486
Sphere	3	5	653	[500,100,25,16,12]	-	0.3	0.147
Iris	4	3	150	[50,50,50]	-	-	-
Wine	13	3	178	[59, 71, 48]	-	-	-
Breast Cancer	30	2	569	[212, 357]	-	-	-

Table 1: Selected datasets to evaluate proposed and baseline classifiers.

Each dataset will be explained in this section, and a graphical representation is presented in Figure 8 and Figure 9. There are datasets with more than 3 dimensions. In case datasets involves more than 3 dimensions, a Principal Component Analysis (PCA) was applied to reduce the dimensionality for visualization purposes only. Then, resulting datasets were plotted taking pairwise variables $\binom{3}{2}$ to provide several two-dimensional points of view with the axis XY, XZ, and YZ, respectively.

4.2.1 Artificial datasets

A group of datasets was artificially generated: The Circles, Swissroll, Moon, Normdist, and Sphere datasets (see Figure 8). In this section, details regarding each one of those datasets will be provided.

The **Circles dataset** is an artificial and simple dataset that consists of a large circle with a small circle inside. Both circles are Gaussian data with a spherical decision boundary for binary classification. A Gaussian noise factor of 3 was added to the data making the circular boundary more diffused. This dataset was proposed to assess the ability to disentangle or to deal with overlapped data regions. The label set will be $L = \{0, 1\}$ denoting both circles. The point set $P \in \mathbb{R}^2$ will be all samples points from both circles. Figure 7 shows the Circle dataset without noise. Figure 8 presents this dataset with a noise factor of 3. The noisy Circles dataset was selected for experiments.

The **Moon dataset** is a simple dataset generated by making two interleaving half circles. A noise factor of 3 was added to data to make it difficult to separate both half circles. The label set $L = \{0, 1\}$ denoting both classes. The point set $P \in \mathbb{R}^2$ is composed of all generated samples of the dataset. Figure 7 shows the Moon dataset samples distribution without noise. Figure 8 shows this dataset with used noise.

The **Swissroll dataset** is an \mathbb{R}^2 point set mapped to \mathbb{R}^3 with a rolled shape. In this paper, a Swissroll dataset was generated using 300 samples from 5 different classes. Besides, a noise factor of 10 was added to the data, which dissolves the rolled shape almost totally. The label set $L = \{0, 1, 2, 3, 4\}$ will be composed by enumerating all classes.

The generated samples will be directly used to build the point set $P \in \mathbb{R}^3$. Figure 7 shows the Swissroll dataset without noise, and Figure 8 shows it with noise.

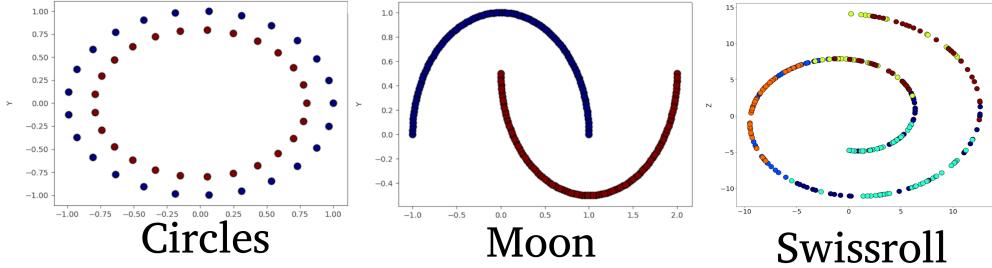


Figure 7: The Circles, Moon, and Swissroll artificial datasets without noise.

An Artificial Dataset Generator framework was implemented for developing datasets distributions. This framework is flexible enough to simulate several complex situations. It is possible to define the desired number of objects (classes) with samples number per class, and global mean and standard deviation or per class.

The ***Normal Distribution based Dataset*** is generated by defining a several per class and overall parameters such as: dataset size, samples dimension, mean per class, standard deviation per class, number of samples per class, number of objects. The number of objects determine the number of classes or labels to be part of the Dataset. The dimension of the Dataset is solved by generating a normal distribution in each component.

An artificial dataset based on mixtures of normal distribution (*Norm_dist*) was generated using the dataset generation framework. This dataset is a high dimensional point set $P \subset \mathbb{R}^{350}$, with a total size of 300 samples. The label set will be $L = \{0, 1, 2, 3, 4\}$. The point set P is composed by generating a normal distribution across each component. The sample number list is [60, 10, 50, 100, 80]. To guide the dispersion and density of the point cloud, we used a mean values collection $\text{mean} = [0, 0.3, 0.18, 0.67, 0]$ and a *standard deviation* (0.486) per label. Figure 8 shows the samples' distribution after the PCA process was applied for visualization.

Generating a ***Sphere-based dataset*** is similar to generating a normal distribution-based one. Although these datasets are always in three dimensions, they are oriented to capture problems associated with data shape, and entanglement between different class samples and diverse class sample distributions and sizes. Figure 8 shows a sphere-based dataset $P \subset \mathbb{R}^3$. This data set has a total size of 653 samples, with a label set $L = \{0, 1, 2, 3, 4\}$. The label distribution is also imbalanced with [500, 100, 25, 16, 12]. The *mean* (0.397) and the *standard deviation* (0.147) are equal per label samples subset.

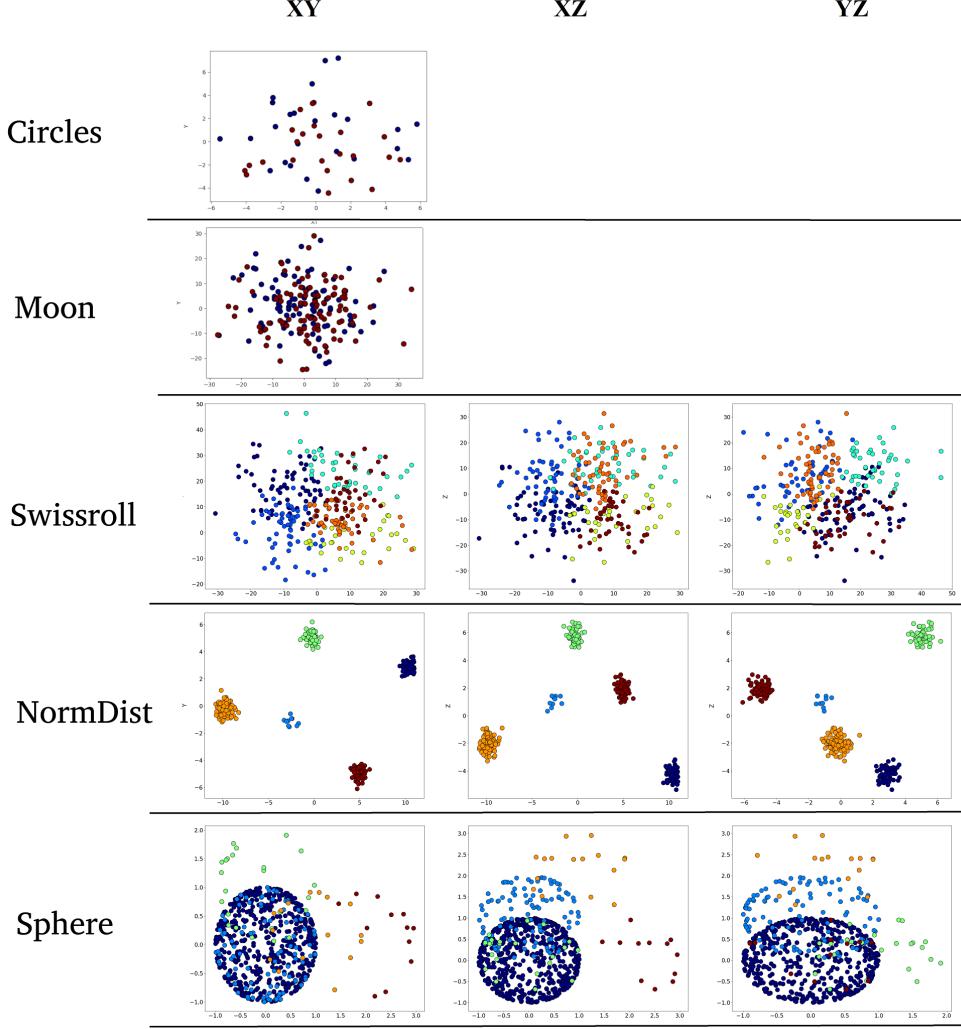


Figure 8: The Circles, Moon, Swissroll, Normdist, and Sphere artificial datasets.

4.2.2 Real datasets

The Iris, Wine, and Breast Cancer datasets were selected as real datasets to compare the proposed classifiers and the baseline ones. In this section, each real dataset will be explained and several of their characteristics will be described.

The **Iris dataset** [16] contains 3 classes of 50 instances each, where each class refers to a type of Iris plant. One class is linearly separable from the other two, which are not linearly separable from each other (see Figure 9). The label set in Iris dataset is $L = \{0, 1, 2\}$ and their corresponding names are “Setosa”, “Versicolor”, and “Virginica”, respectively. Each sample in the Iris dataset is a 5-tuple, defined by (sepal_length, sepal_width, petal_length, petal_width, label). The class of Iris plant will be the predicted attribute. The point set P is built using the first four components of each sample.

The **Wine dataset** [13] is the result of a chemical analysis of wines grown in the same region in Italy by three different growers. There are thirteen different measurements taken for different components found in the three types of wine. The label set $L = \{0, 1, 2\}$ to enumerate the three wine types will be taken from the first component of each sample. The point set $P \subset \mathbb{R}^{13}$ will be completed using the remaining 13 components of each sample. Figure 9 shows the Wine dataset samples distribution after applying PCA to reduce dimensions to 3, and it was plotted combining two dimensions.

The **Breast Cancer dataset** [13] features were computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe the characteristics of the cell nuclei present in the image. The label set $L = \{0, 1\}$ will denote Malignant (0) tumors and Benign (1) tumors. The point set will be $P \subset \mathbb{R}^{30}$ where each sample represents

the cell nuclei information of one image. Figure 9 shows this dataset after applying a PCA process to visualize it from several 2-dimensional perspectives.

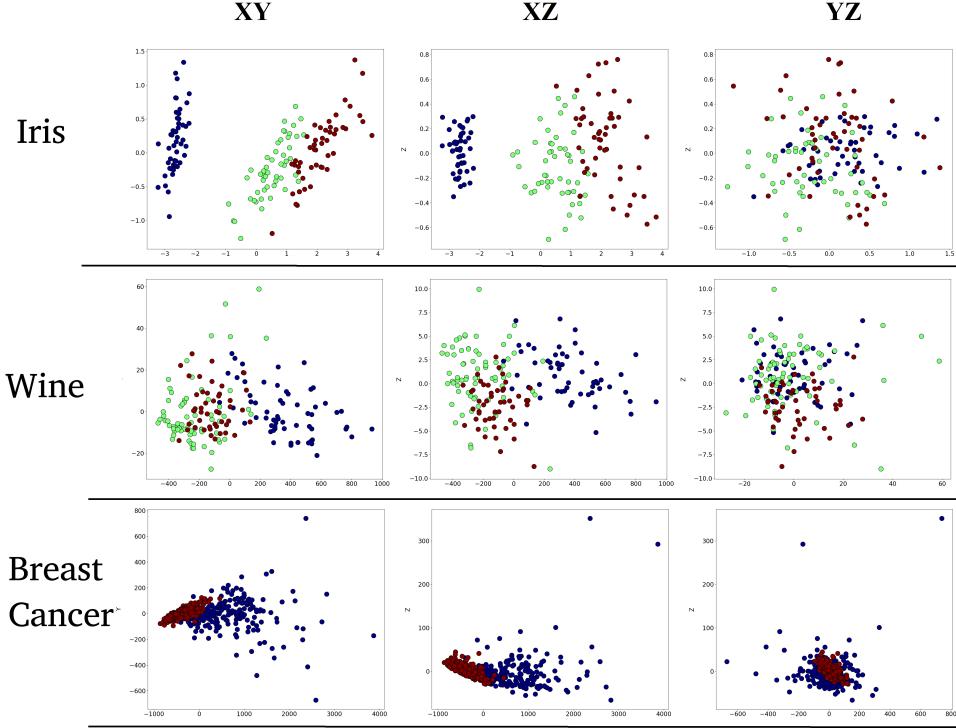


Figure 9: The Iris, Wine, and Breast Cancer datasets, chosen as real datasets.

4.3 Classifier Evaluation

The classifier evaluation over all datasets were conducted using a Repeated cross-validation process (see A for details). The aim is to avoid biased results because the training and test sets are each time from the same dataset.

Let R be the fold size in the Repeated Cross-Validation approach to avoid confusion with the use of k in k -NN. The R-FOLD Cross Validation is then repeated 5 times ($N=5$), and R will be the 10% of the selected dataset. For any value of R , $(\lceil \frac{P}{R} \rceil - 1)$ R -folds will be selected to be the training set S , and the remaining R -fold will be the test data X in each iteration. When $2 \cdot R \geq |P|$ the problem is considered to be semi-supervised, where there are more unknown samples to classify than there are labeled samples.

It is common in ML algorithms to use parameters whose values are changed before the learning process begins. Those parameters are called hyper-parameters [29, 21, 27]. For k -NN and wk-NN algorithms, a $k=15$ was considered a good number of neighbors; we obtained it using the hyper-parameter estimators from scikit-learn [29]. For the three TDABC algorithms, the maximal simplex dimension needs to be fixed to a q value to control the VR-complex construction process. Experiments were conducted with $3 \leq q \leq 10$. In B, a detailed explanation of selected metrics is given. Results presented were obtained on two computers: 8 GB RAM, Intel Core i7-6500U CPU 2.50GHz x 4, and 16 GB RAM, AMD RyzenTM 7 3700U 2.30 GHz x 4.

4.4 Comparison

Each classifier is executed a total number of $W = N \cdot \lceil \frac{|P|}{R} \rceil$ times because of the repeated cross validation process. This process results in a total number of W predicted collections $\hat{Y}_1, \hat{Y}_2, \hat{Y}_3, \dots, \hat{Y}_W$ for each classifier. Similarly, a total number of real label collections Y_1, Y_2, \dots, Y_W results for each classifier. Those two lists of collections, $\{\hat{Y}_i\}_{1 \leq i \leq W}$, and $\{Y_i\}_{1 \leq i \leq W}$ are concatenated by putting the collection $i+1$ at the end of the previous collection, which results in two big collections of predicted, and real labels.

$$\hat{Y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{|P|}, \hat{y}_{|P|+1}, \dots, \hat{y}_{2 \cdot |P|}, \dots, \hat{y}_{N \cdot |P|}), \quad (8)$$

$$Y = (y_1, y_2, \dots, y_{|P|}, y_{|P|+1}, \dots, y_{2 \cdot |P|}, \dots, y_{N \cdot |P|}), \quad (9)$$

Where $\hat{Y}_i = (\hat{y}_j)_{(i-1) \cdot |P| + 1 \leq j \leq i \cdot |P|}$ is the predicted labels list and $Y_i = (y_j)_{(i-1) \cdot |P| + 1 \leq j \leq i \cdot |P|}$ the real labels list, both resulting from i^{th} execution of Repeated R-Fold. As the correspondence between each components of both predicted and real labels is maintained, it is easy to generalize all metrics' computation by considering $n = |\hat{Y}| = |Y|$.

4.4.1 General metrics computation result

In Table 2 and Table 3, all metrics results are shown. Each metric was computed across all datasets. For each metric, columns from 2 to 10 represent the datasets, and rows represent each classifier results. The two last columns show the arithmetic mean and standard deviation of the corresponding metrics across all datasets. More details about the metrics' computations are presented in B.

The experiments were conducted per dataset for each fixed simplicial complex dimension on the interval $q \in [3, 9]$. Nevertheless, in this section, results are shown for one value of q in each dataset. For example, from the Iris, Circles, and Sphere datasets, a fixed simplicial complex dimension $q = 8$ was selected. As another example, for the Moon-dataset, results were selected for experiments conducted using a simplicial complex dimension $q = 3$. In contrast, metrics results on the Swissroll and Wine datasets were selected for simplicial complex dimension $q = 6$. On the other hand, the presented results were obtained for the Breast Cancer dataset using a fixed simplicial complex dimension $q = 4$. Meanwhile, from the Normdist dataset, results were selected using a fixed simplicial complex dimension $q = 7$.

Table 4 summarizes the classifiers' average performance. It was built using the two last columns of Tables 2 and Table 3, which represent the mean and the standard deviation of each metric across all datasets.

4.4.2 Selected confusion matrices

For a graphical visualization of the evaluated classifiers' performance, 40 confusion matrices were created, each one corresponding to a classifier in a dataset (5 classifiers, 8 datasets). Nonetheless, only matrices for Iris, Circles, Moon and Sphere datasets are shown in this section. All confusion matrices can be seen in C.

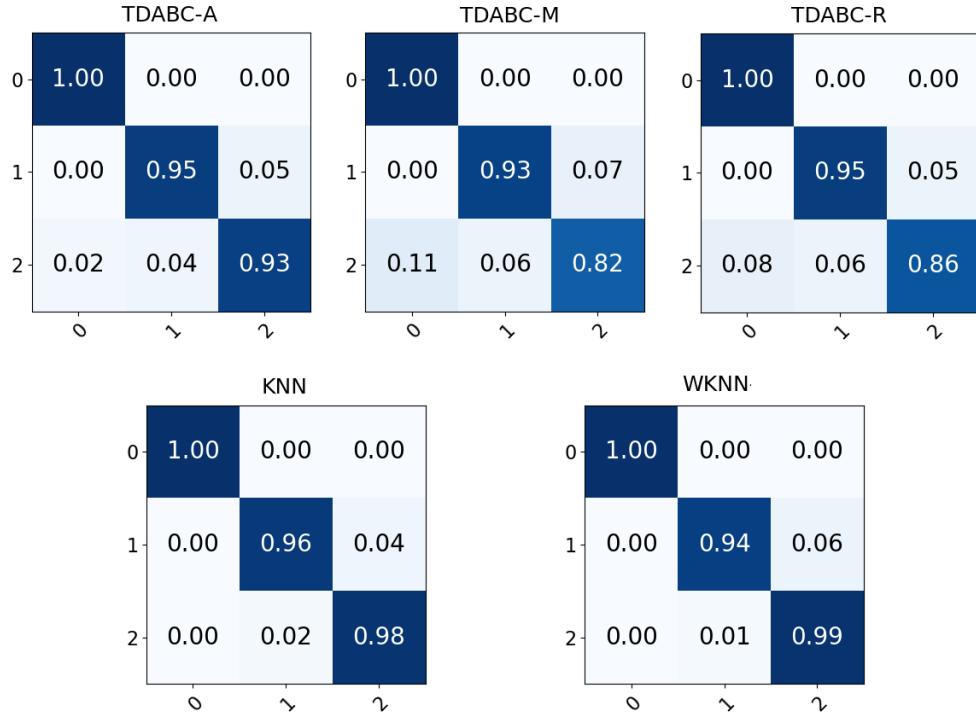


Figure 10: Confusion matrices from classifiers' results on the Iris dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 8$.

Accuracy (ACC)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,668	0,516	0,919	0,993	0,918	0,961	0,769	0,91	0,832	0,167
TDABC-M	0,700	0,524	0,911	0,983	0,938	0,920	0,767	0,92	0,832	0,157
TDABC-R	0,676	0,530	0,913	0,990	0,925	0,936	0,768	0,92	0,832	0,159
wk-NN	0,468	0,456	0,943	0,990	0,901	0,977	0,739	0,93	0,801	0,223
k-NN	0,508	0,475	0,918	0,987	0,887	0,980	0,708	0,93	0,799	0,209
Precision (PR)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,668	0,516	0,920	0,979	0,718	0,960	0,765	0,91	0,805	0,165
TDABC-M	0,700	0,524	0,911	0,971	0,747	0,917	0,763	0,92	0,806	0,151
TDABC-R	0,676	0,530	0,913	0,976	0,742	0,934	0,764	0,92	0,807	0,155
wk-NN	0,468	0,456	0,940	0,950	0,579	0,976	0,742	0,92	0,754	0,224
k-NN	0,508	0,475	0,914	0,933	0,549	0,979	0,701	0,92	0,747	0,213
Recall (RE)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,668	0,516	0,692	0,964	0,454	0,927	0,622	0,90	0,718	0,193
TDABC-M	0,700	0,524	0,667	0,901	0,496	0,859	0,619	0,91	0,709	0,164
TDABC-R	0,676	0,530	0,673	0,941	0,472	0,885	0,621	0,91	0,713	0,178
wk-NN	0,463	0,456	0,764	0,966	0,400	0,957	0,590	0,93	0,691	0,243
k-NN	0,509	0,475	0,688	0,955	0,379	0,962	0,547	0,93	0,681	0,239
True Negative Rate (TNR)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,668	0,516	0,983	0,999	0,981	0,981	0,873	0,90	0,863	0,178
TDABC-M	0,700	0,524	0,981	0,996	0,985	0,961	0,871	0,91	0,866	0,169
TDABC-R	0,676	0,530	0,981	0,998	0,982	0,969	0,872	0,91	0,864	0,171
wk-NN	0,463	0,456	0,988	0,998	0,977	0,989	0,859	0,93	0,833	0,235
k-NN	0,509	0,475	0,983	0,997	0,974	0,990	0,841	0,93	0,838	0,220
False Positive Rate (FPR)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,332	0,484	0,080	0,021	0,282	0,040	0,235	0,09	0,195	0,165
TDABC-M	0,300	0,476	0,089	0,029	0,253	0,083	0,237	0,08	0,194	0,151
TDABC-R	0,324	0,470	0,087	0,024	0,258	0,066	0,236	0,08	0,193	0,155
wk-NN	0,532	0,544	0,060	0,050	0,421	0,024	0,258	0,08	0,246	0,224
k-NN	0,492	0,525	0,086	0,067	0,452	0,021	0,295	0,08	0,252	0,213

Table 2: The Acc, Pr, Re, TNR, and FPR metric results per classifier across all datasets.

F1-Measure (F1)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,668	0,516	0,789	0,971	0,504	0,943	0,683	0,91	0,748	0,185
TDABC-M	0,700	0,524	0,770	0,934	0,560	0,883	0,680	0,91	0,745	0,157
TDABC-R	0,676	0,530	0,774	0,958	0,527	0,906	0,682	0,91	0,745	0,170
wk-NN	0,450	0,455	0,842	0,954	0,433	0,966	0,648	0,93	0,709	0,240
k-NN	0,494	0,475	0,784	0,937	0,524	0,970	0,608	0,93	0,715	0,213
Matthews Correlation Coefficient (MCC)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,336	0,032	0,752	0,967	0,478	0,915	0,514	0,82	0,601	0,321
TDABC-M	0,400	0,048	0,729	0,925	0,541	0,828	0,510	0,82	0,600	0,288
TDABC-R	0,352	0,060	0,735	0,952	0,506	0,861	0,512	0,82	0,600	0,300
wk-NN	-0,069	-0,088	0,815	0,950	0,384	0,950	0,465	0,86	0,533	0,432
k-NN	0,017	-0,050	0,747	0,932	0,346	0,955	0,400	0,85	0,525	0,404
Geometric Mean (GMean)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,668	0,516	0,824	0,981	0,614	0,954	0,735	0,90	0,774	0,169
TDABC-M	0,700	0,524	0,809	0,946	0,656	0,908	0,733	0,90	0,773	0,146
TDABC-R	0,676	0,530	0,812	0,969	0,632	0,925	0,734	0,90	0,773	0,156
wk-NN	0,463	0,456	0,869	0,982	0,522	0,973	0,709	0,93	0,738	0,231
k-NN	0,509	0,475	0,822	0,976	0,500	0,976	0,676	0,93	0,733	0,221
Classification Error (CErr)										
Method	Circles	Moon	Swissroll	Norm dist	Sphere	IRIS	Wine	Breast cancer	Mean	Stdev
TDABC-A	0,332	0,484	0,081	0,007	0,083	0,039	0,231	0,09	0,168	0,167
TDABC-M	0,300	0,476	0,089	0,017	0,062	0,080	0,233	0,08	0,168	0,157
TDABC-R	0,324	0,470	0,081	0,010	0,075	0,064	0,232	0,08	0,168	0,160
wk-NN	0,532	0,544	0,057	0,010	0,099	0,023	0,261	0,07	0,199	0,223
k-NN	0,492	0,525	0,082	0,013	0,113	0,020	0,292	0,07	0,201	0,209

Table 3: The F1, MCC, GMean, and CErr metric results per classifier across all datasets.

Name	TDABC-A	TDABC-M	TDABC-R	wk-NN	k-NN
Acc	0,832 ± 0,17	0,832 ± 0,16	0,832 ± 0,16	0,801 ± 0,22	0,799 ± 0,21
Pr	0,805 ± 0,17	0,806 ± 0,15	0,807 ± 0,15	0,754 ± 0,22	0,747 ± 0,21
Re	0,718 ± 0,19	0,709 ± 0,16	0,713 ± 0,18	0,691 ± 0,24	0,681 ± 0,24
TNR	0,863 ± 0,18	0,866 ± 0,17	0,864 ± 0,17	0,833 ± 0,23	0,837 ± 0,22
FPR	0,195 ± 0,17	0,194 ± 0,15	0,193 ± 0,15	0,246 ± 0,22	0,252 ± 0,21
F1	0,75 ± 0,18	0,75 ± 0,16	0,75 ± 0,17	0,71 ± 0,24	0,71 ± 0,21
MCC	0,601 ± 0,32	0,600 ± 0,29	0,600 ± 0,30	0,533 ± 0,43	0,525 ± 0,40
GMEAN	0,774 ± 0,17	0,773 ± 0,15	0,773 ± 0,16	0,738 ± 0,23	0,733 ± 0,22
CErr	0,168 ± 0,17	0,168 ± 0,16	0,168 ± 0,16	0,199 ± 0,22	0,201 ± 0,21

Table 4: Summary table with the arithmetic mean of the classifiers across all analyzed data sets. Each mean result and standard deviation is shown for the performance in this metric across all datasets.

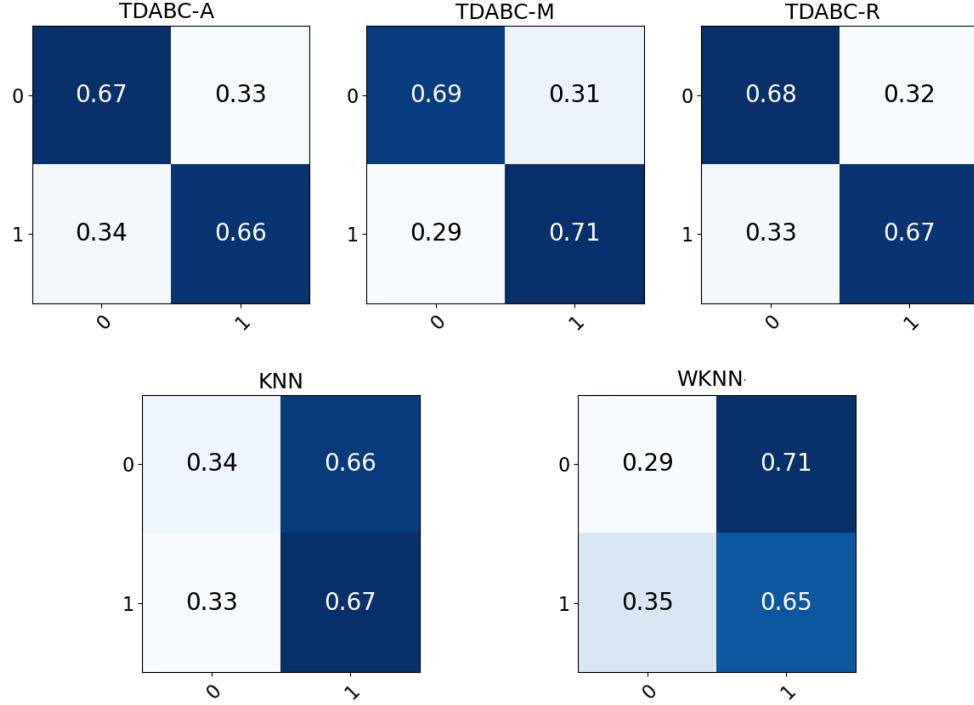


Figure 11: Confusion matrices from classifiers’ results on the Circles dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 8$.

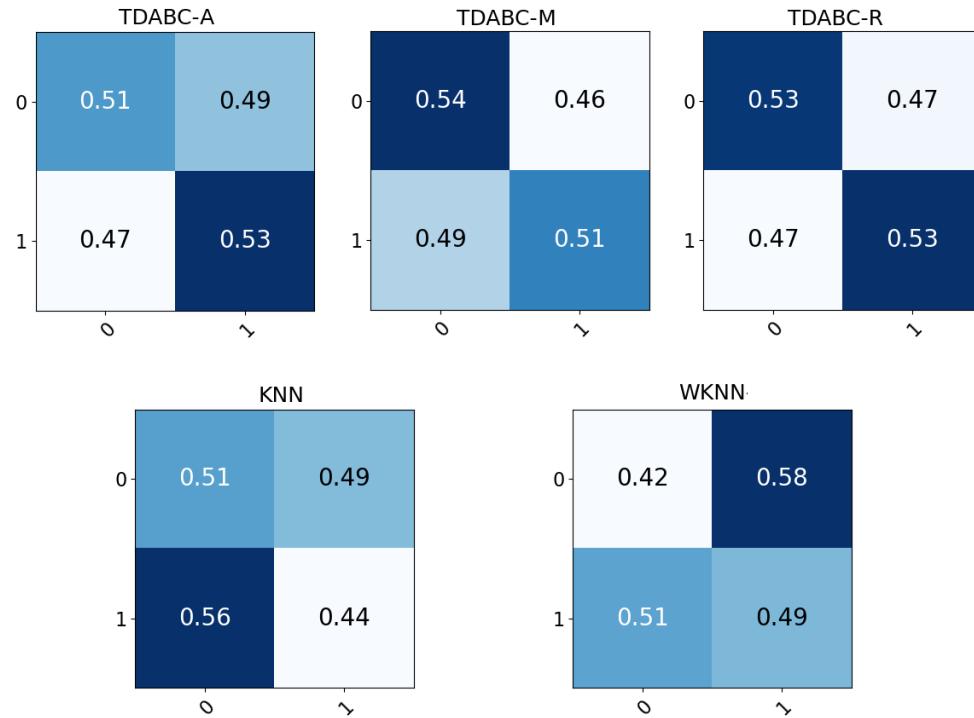


Figure 12: Confusion matrices from classifiers’ results on the Moon dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 3$.

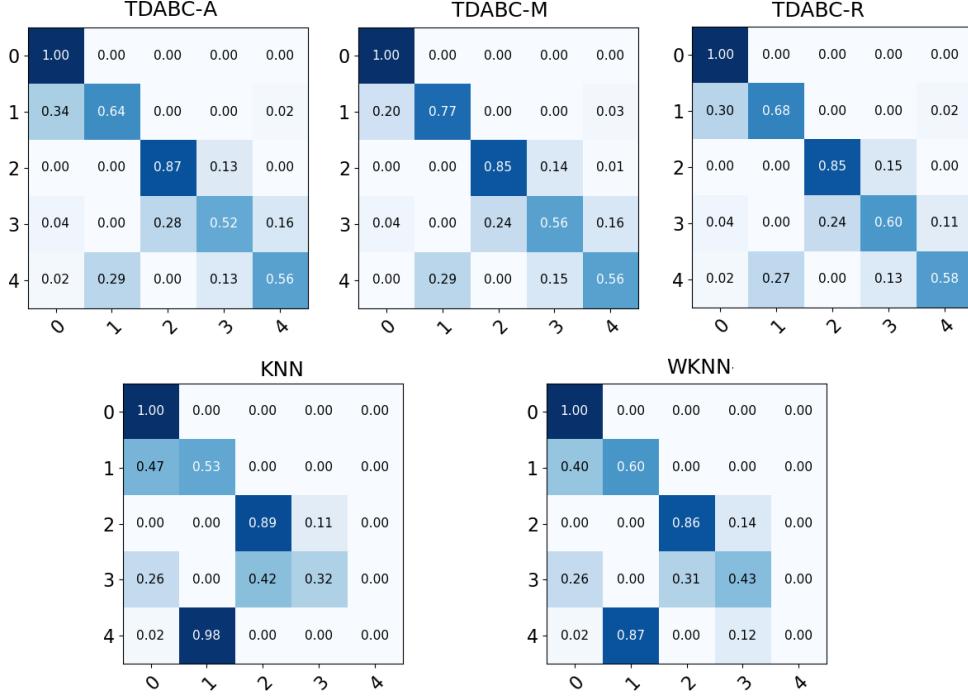


Figure 13: Confusion matrices from classifiers' results on the Sphere dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 8$.

5 Discussion

The discussion section is organized into three subsections. First, the analysis of results, followed by a highlight of the proposed method's most relevant characteristics and a discussion of related works.

5.1 Results

Average performance was computed with the arithmetic mean, geometric mean, and harmonic mean in all datasets. Algorithm ranking remained across means' computations; thus, only arithmetic mean is shown in Table 4.

By analyzing results per dataset independently, it could be noted that the TDABC approaches were superior to wk-NN and k-NN in 5 of the 8 evaluated datasets, specifically on the Circles, Moon, Normdist, Sphere, and Wine datasets (see Table 2 and Table 3). On the other hand, baseline methods were slightly better on the three remaining datasets.

The Circles, Moon, Normdist, Sphere, and Wine datasets have different challenging features such as high dimensionality, the imbalanced distribution of labels, and highly entangled classes. Despite these challenges, TDABC approaches overcome baseline methods in every computed metric.

The Circles and Moon datasets are balanced and have very entangled classes due to the noise factor, making the classification a challenge. In these datasets, wk-NN and k-NN behave poorly, as observed through the negative values obtained after applying the MCC measure (see Table 5). This behavior is related to the fixed value of k and to the assumption that each data point is equally relevant. Even though wk-NN imposes a local data point weighting based on distances, it is not enough with highly entangled classes, as our results show. The TDABC methods are capable of dealing with the entanglement challenge through a disambiguation factor based on filtration values (ξ_K).

In the case of the Normdist and Sphere datasets, there is a high imbalanced ratio of the classes, with [60, 10, 50, 100, 80] and [500, 100, 25, 16, 12] samples per class, respectively. In this situation, it is important to have dynamic neighborhoods. The proposed method generates dynamic-sized “neighborhoods” for each point, in contrast to k-NN and wk-NN classifiers. In the high imbalance case, the disambiguation factor (ξ_K) also provides a multi-scale local weighting to TDABC methods.

The Wine and Normdist datasets are highly dimensional (13 and 350 dimensions, respectively). TDABC methods behave better than baseline approaches for those datasets in all metrics. k-NN and wk-NN use Euclidean distance directly to detect their k neighbors. Even though TDABC methods also use Euclidean distance, they are still able to unravel multi-scale, multi-dimensional relationships among data, which better handle high dimensionality.

Swissroll and Iris are balanced datasets, with 50 samples per class. In contrast, the Breast Cancer dataset has [212, 357] samples per class. In the Swissroll and Iris datasets, weighted k-NN and k-NN were better, respectively, than proposed classifiers in all metrics. Interestingly, for the Breast Cancer dataset, which is slightly imbalanced, TDABC was equally performant in 2 out of 9 metrics.

5.2 Key aspects of the proposed method

Regarding the proposed TDA-based classification methodology, two key aspects are discussed: persistent homology and voting system.

The first aspect is related to persistent homology's key role in selecting the desired sub-complex from a filtered simplicial complex built on the dataset. Algorithm 3 reduces the search space in the filtered simplicial complex by taking advantage of topological features encoded inside selected persistence intervals. Although, selecting the right sub-complex is a very challenging problem [7], the simple criterion we propose (death time of persistence intervals resulting from MaxInt, RandInt, and AvgInt) is sufficient to achieve good classification results.

Despite the birth time's theoretical guarantees to selecting the sub-complex, the middle and death times might be useful depending on the dataset structure and complexity. Experimentally, promising results were obtained using both middle and death time. However, death time was better experimentally because it can reach more stable topological features and minimize the presence of isolated points. This process is summarized in Figure 5.

The second aspect is the proposed voting system (see Definition 13), which gives richer information than the one used in the classification. During the voting system execution, a fundamental stage is the label propagation performed by the labeling function from Definition 14. The result of the labeling function could also be represented by a contribution vector $\Upsilon(\sigma) \equiv v \in \mathbb{R}^{|L|}$, with each component i being the contribution of each label $l_i \in L$. By normalizing v , the probabilities of σ to belong to each component's class is obtained. Thus, the voting system provides the probability of each class, allowing, for instance, the use of ensemble techniques.

5.3 Related methods

In other related approaches, the authors of [37] proposes the Rare-class Nearest Neighbour (KRNN), a k-NN variant to deal with the sparsity of the positive samples on an imbalanced dataset. The KRNN uses dynamic local query neighborhoods that contain at least k positive nearest neighbors (a member of minority classes). In [36], a different approach is proposed to deal with imbalanced datasets, focusing on negative samples (from majority class) in contrast to [37]. They experimentally prove that negative samples on the overlapping region cause the most inaccuracies on classification. Thus, a neighbor-based algorithm is proposed in [36] that removes negative samples from the overlapped area.

Both [37] and [36] were built to deal with two-classes imbalanced classification problems successfully. However, when applied to multi-class imbalanced problems, several issues arose in both methods, mostly related to the ambiguity of determining if an instance is a positive or a negative one. In multi-classes imbalanced problems, the same class l_i could play both roles simultaneously because it could be a minority class concerning a class l_j , but it could be majority respect to a class l_k . Closely related testing scenarios were the Normdist and Sphere datasets, where the proposed TDABC method was experimentally evaluated. The proposed method obtains good classification rates on minority classes, and it was also able to deal with the overlapping area because of its disentanglement properties.

Recent TDA works still consider TDA as a complement of ML tasks. Works as [1, 30] focus on discovering better ways to transform persistent homology representations into topological features for deep learning pipelines or sophisticated ML methods. In [1], the stability of persistent entropy is provided, justifying its application as a useful statistic in topological data analysis. In [30], TDA is applied to bioinformatics by proposing a novel algorithm based on another major TDA tool called the Mapper algorithm, used to visualize and interpret low and high volume of data (see [8]), and built a ML classifier on top of the mapper generated graphs. In [24], Self-Organized Maps were combined with TDA tools to cluster and classify time series in the financial domain with competitive results. In this context, our work is an example of a fully TDA-based approach applied to supervised learning, with a preliminary version shown as two technical reports in [22, 23].

6 Conclusions

In this work, TDA was applied directly in a classification problem and evaluated in 8 datasets, including imbalanced and high dimensionality ones, with good results compared to baseline methods. Overall, we show that Topological Data Analysis alone can classify without any ML method. To our knowledge, this is the first study that proposes this approach for classification.

The proposed TDA-based classification method propagates labels from labeled points to unlabeled ones over the built filtered simplicial complex. The filtration values were interpreted as indirect distance indicators to provide a natural disambiguation method to label-contributions.

The use of persistent homology was key to reduce the search space's complexity by providing the topological features needed to select a sub-complex close enough to the data topology and use it for classification.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research work was supported by the National Agency for Research and Development of Chile (ANID), with grants ANID 2018/Beca DOCTORADO NACIONAL-21181978, FONDECYT 1181506, ICN09_015, and PIA ACT192015. Beca postdoctoral CONACYT (Mexico) also supports this work. The first author would like to thank professor José Carlos Gómez-Larrañaga from CIMAT, Mexico, due to his support and collaboration.

A Repeated cross-validation process

The performance evaluation of any classifier in a multi-class classification problem is a difficult task. One of the most significant issues is ensuring that the assessment does not make any assumption about data distributions or the classifier. Another problem to address is guaranteeing the testing robustness against bias, overfitting, and underfitting. A well-known approach is to use a cross-validation method [6, 29, 21, 27]. Cross-validation aims to divide the data set P into equal pieces or folds of size R , one of those pieces is selected to be the test set X , and the $(\frac{|P|}{R} - 1)$ remaining folds are considered the training set S . This process continues until the last fold is selected to be X . However, since all folds are taken from the same dataset, sometimes a fold is a test set and, at other times, it is part of the training set. This process makes Cross Validation biased. One way to overcome this issue is by making a repeated cross validation process. This means repeating the R -Fold cross validation process N times. This method is called Repeated Cross-Validation (see Figure 14).

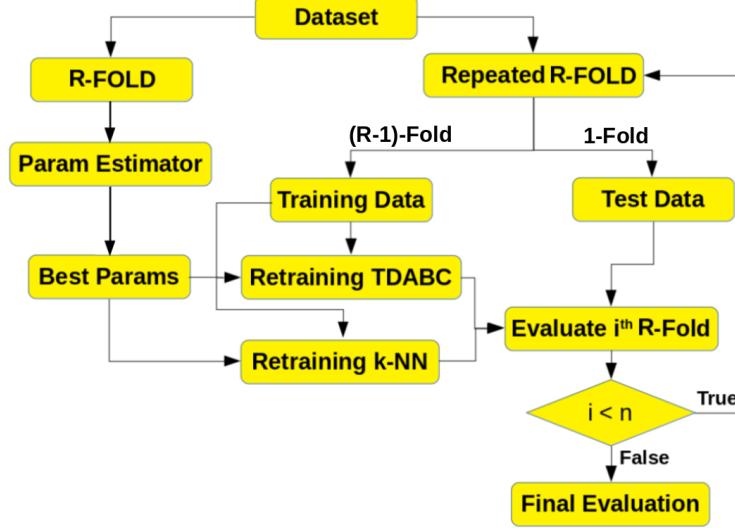


Figure 14: Repeated Cross Validation overall process to compare TDABC proposed variants and the baseline classifiers.

B Metrics for Classifiers Evaluation

Several metrics need to be considered to evaluate the proposed and baseline classifiers' performance. The classification metrics are computed as functions of the True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)'s values. Once those primitive values are computed, it is possible to compute several classification metrics such as Accuracy (Acc), Precision (Pr), Recall (Re), False Positive Rate (FPR), False Negative Rates (FNR), F1-measure ($F1$ or Harmonic Measure of Pr and Re), Matthews Correlation Coefficient (MCC), Geometric Mean (GMEAN), and Classification Error (MSE).

On the other hand, real and predicted label collections definitions are needed to compute the metrics. Let Y be the real label list $\forall x \in X$. Let \hat{Y} be the predicted label list $\forall x \in X$ computed by Algorithm 1, where $n = |Y| = |\hat{Y}| = |X|$. The following sections explain the metrics' computation.

B.1 True positives, True negatives, False Positives, and False Negatives

A True positive sample is a sample successfully classified as belonging to the True label (the critical or most important one). A True Negative value is a sample successfully classified as to be labeled with a negative label. A False positive value is a mislabeled sample with a true label, and a False negative is a mislabeled sample with a negative label.

In a multi-class classification problem (more than two classes), it is more difficult to determine true and negative classes. In this paper, each class is considered a true class, the remaining classes will be the negative ones, and this process is repeated to cover all classes as true classes. In this process, TP_l , TN_l , FP_l , and FN_l are computed for each label $l \in L$ with L the label set, from Equation 10 to Equation 13

$$TP_l = \sum_{i=1}^n \mathcal{I}(l = \hat{y}_i) \cdot \mathcal{I}(\hat{y}_i = y_i), \quad (10)$$

$$FP_l = \sum_{i=1}^n \mathcal{I}(l = \hat{y}_i) \cdot \mathcal{I}(\hat{y}_i \neq y_i), \quad (11)$$

$$TN_l = \sum_{i=1}^n \mathcal{I}(l \neq \hat{y}_i) \cdot \mathcal{I}(\hat{y}_i = y_i), \quad (12)$$

$$FN_l = \sum_{i=1}^n \mathcal{I}(l \neq \hat{y}_i) \cdot \mathcal{I}(\hat{y}_i \neq y_i). \quad (13)$$

Metric Name	Equation	Defined Interval	Worst	Better
Acc	$\frac{1}{ L } \cdot \sum_{l \in L} \frac{TP_l + TN_l}{TP_l + TN_l + FP_l + FN_l}$	[0,1]	0	1
Pr	$\frac{1}{ L } \cdot \sum_{l \in L} \frac{TP_l}{TP_l + FP_l}$	[0,1]	0	1
Re	$\frac{1}{ L } \cdot \sum_{l \in L} \frac{TP_l}{TP_l + FN_l}$	[0,1]	0	1
TNR	$\frac{1}{ L } \cdot \sum_{l \in L} \frac{TN_l}{TN_l + FP_l}$	[0,1]	0	1
FPR	$\frac{1}{ L } \cdot \sum_{l \in L} \frac{FP_l}{TN_l + FP_l}$	[0,1]	1	0
F1	$\frac{1}{ L } \cdot \sum_{l \in L} \frac{2 \cdot TP_l}{2 \cdot TP_l + FP_l + FN_l}$	[0,1]	0	1
MCC	$\frac{1}{ L } \cdot \sum_{l \in L} \frac{TP_l \cdot TN_l - FP_l \cdot FN_l}{\sqrt{(TP_l + FP_l) \cdot (TP_l + FN_l) \cdot (TN_l + FP_l) \cdot (TN_l + FN_l)}}$	[-1,1]	-1	1
GMEAN	$\frac{1}{ L } \cdot \sum_{l \in L} \sqrt{\frac{TN_l \cdot TP_l}{(TN_l + FP_l) \cdot (TP_l + FN_l)}}$	[0,1]	0	1
CErr	$\frac{1}{ n } \cdot \sum_{i=1}^n \mathcal{I}(\hat{y}_i \neq y_i)$	[0,1]	1	0

Table 5: Classifier evaluation metrics information. Macro-averaging is performed in each metric to generalize it to a multi-class classification problem. This approach is valid even for the case of binary classification.

B.2 Metrics computation for binary and multi-class classification

Binary classification is the setting where only two classes are taken into consideration. In this scenario, popular metrics are:

- **Accuracy (Acc):** Percentage of correct predictions over the total samples.
- **Precision:** Number of items correctly identified as positive over the total of positive items.
- **Recall, Sensitivity or True Positive Rate (Re):** Number of items correctly identified as positive out of total true positives.
- **True Negative Rate or Specificity (TNR):** Number of items correctly identified as negative out of total negatives.
- **False Positive Rate or Type I Error (FPR):** Number of items wrongly identified as positive out of total true negatives.
- **False Negative Rate or Type II Error (FNR):** Number of items wrongly identified as negative out of total true positives.
- **F1-Measure:** This measure summarizes Pr and Re in a single metric. It is known to be the harmonic mean from both. It mitigates the impact of the high rate but also accentuates the lower rates' impact.
- **Matthews Correlation Coefficient (MCC):** A measure unaffected by the imbalanced datasets issue. MCC is a contingency matrix method obtained from calculating the Pearson correlation coefficient between real and predicted values.
- **Geometric Mean (GMean):** The geometric mean corresponds to the square root of the product of the Recall and True Negative Rate. It is commonly used to understand the classifier behavior with imbalanced datasets.
- **Classification Error (CErr):** Percentage of misclassification over the total samples.

On the other hand, for more than two classes the problem is named multi-class classification. A common way to address the classifiers' assessment in this setting is to consider a One-vs-All configuration. It consists of taking one class as the positive class, considering the remaining classes as negative ones, and repeating this process for every class. In this case, it is necessary to make metric generalizations to a multi-class environment. A popular method is averaging the metrics through micro-averaging or macro-averaging. Micro-averaging considers all elements TP_l, TN_l, FP_l and FN_l to extend the two-class metric equations. In contrast, macro-averaging considers the per-class metrics' performance. In this paper, a macro-averaging is performed for each metric. Table 5 summarizes this computation.

Those metrics are computed for every iteration of the repeated R-Fold strategy (A).

C Confusion Matrices

A confusion matrix is a specific table layout that allows the visualization of the performance of a supervised learning algorithm. Each row of the matrix represents the instances in a real class, while each column represents the instances in a predicted class (or vice versa). The vectors \hat{Y}, \hat{Y} are used to construct each confusion matrix.

For each dataset, 5 confusion matrices were created, one matrix for each classifier. The three corresponding matrices of TDABC-R, TDABC-M, and TDABC-A proposed methods will be placed on the first row. The remaining two matrices will correspond to the baseline algorithms k-NN and wk-NN, and they will be arranged as a second row.

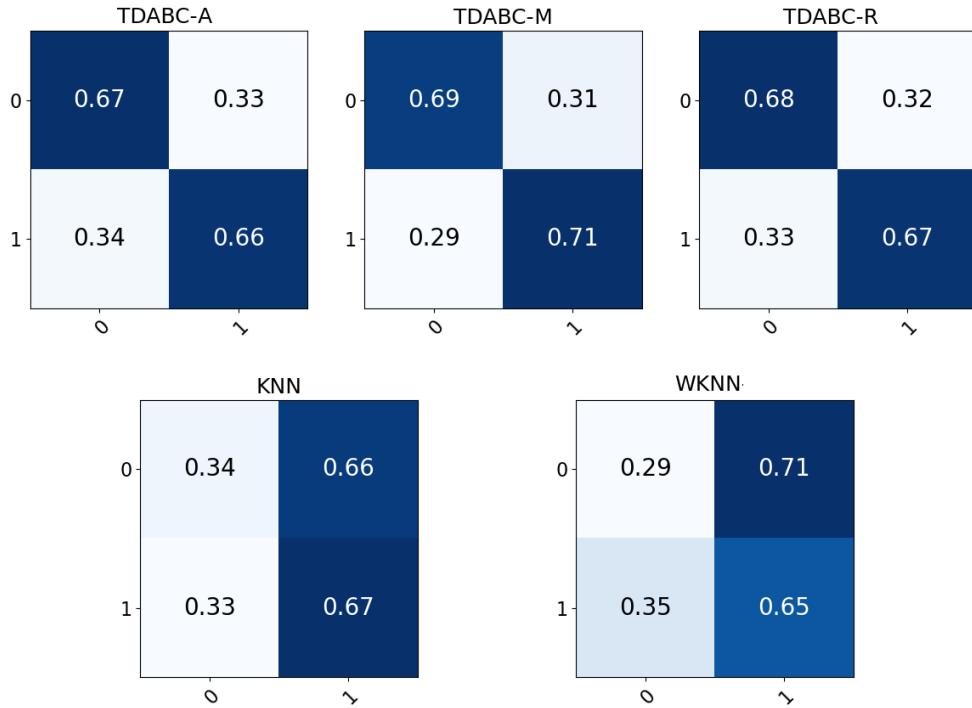


Figure 15: Confusion matrices from classifiers' results on the Circles dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 8$.

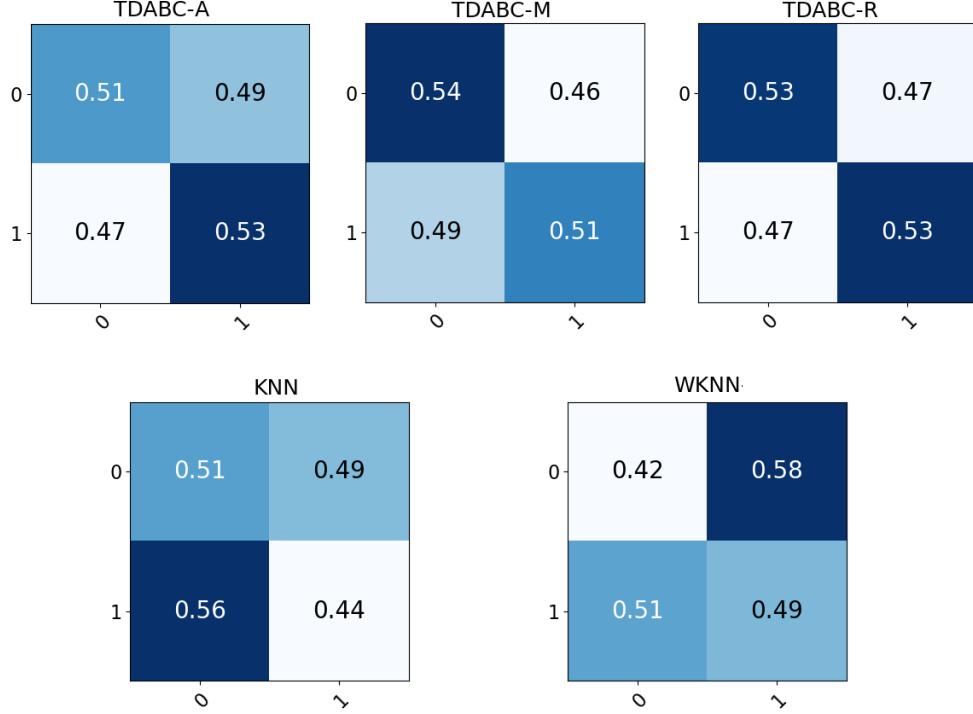


Figure 16: Confusion matrices from classifiers' results on the Moon dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 3$.

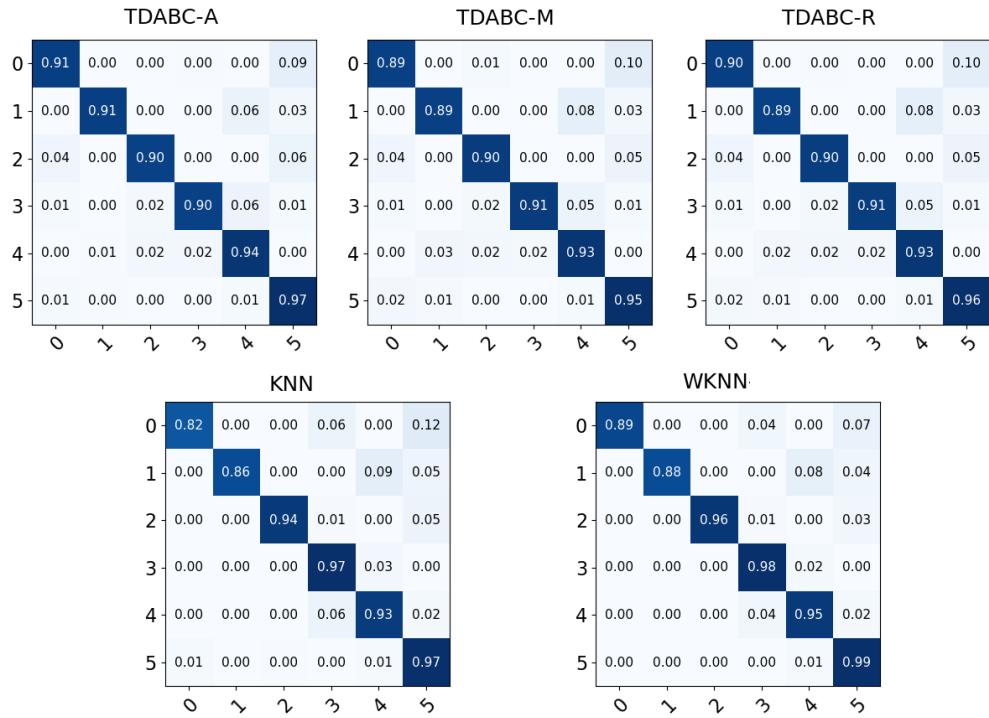


Figure 17: Confusion matrices from classifiers' results on the Swissroll dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 6$.

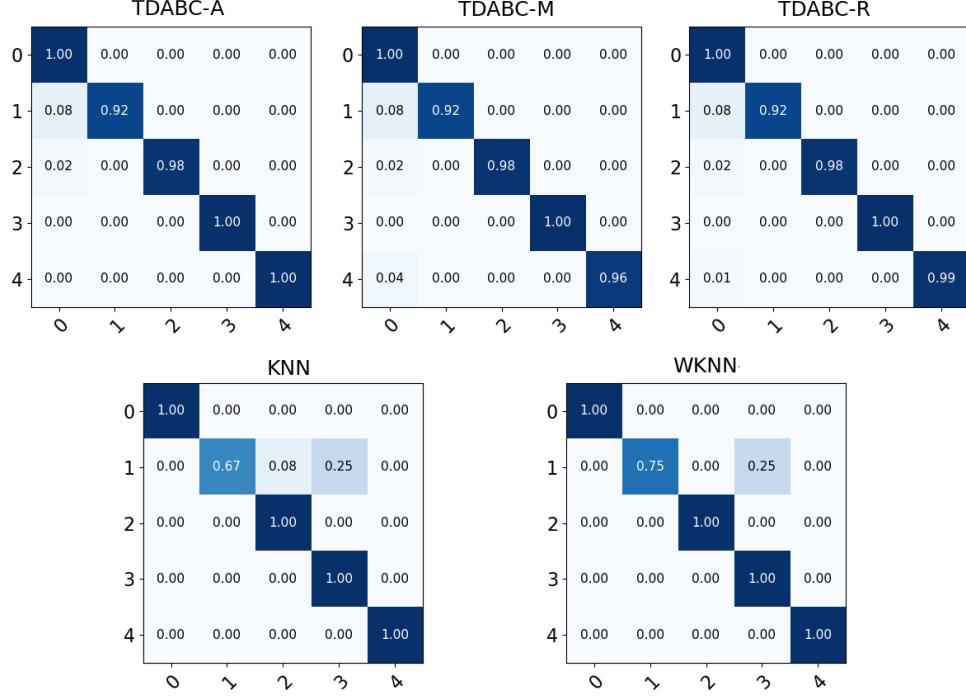


Figure 18: Confusion matrices from classifiers' results on the Normdist dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 7$.

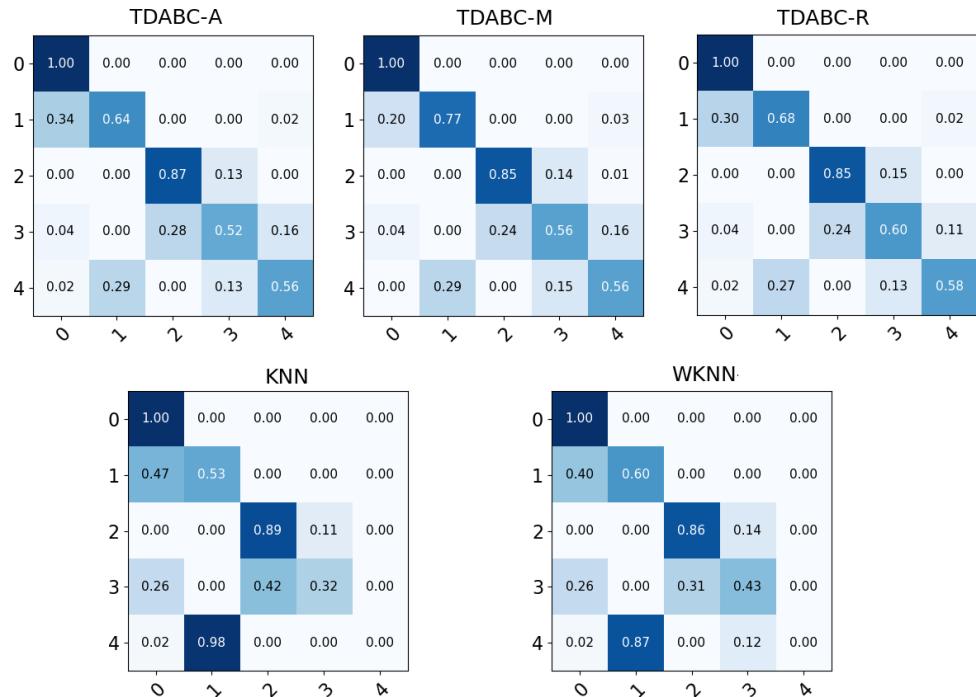


Figure 19: Confusion matrices from classifiers' results on the Sphere dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 8$.

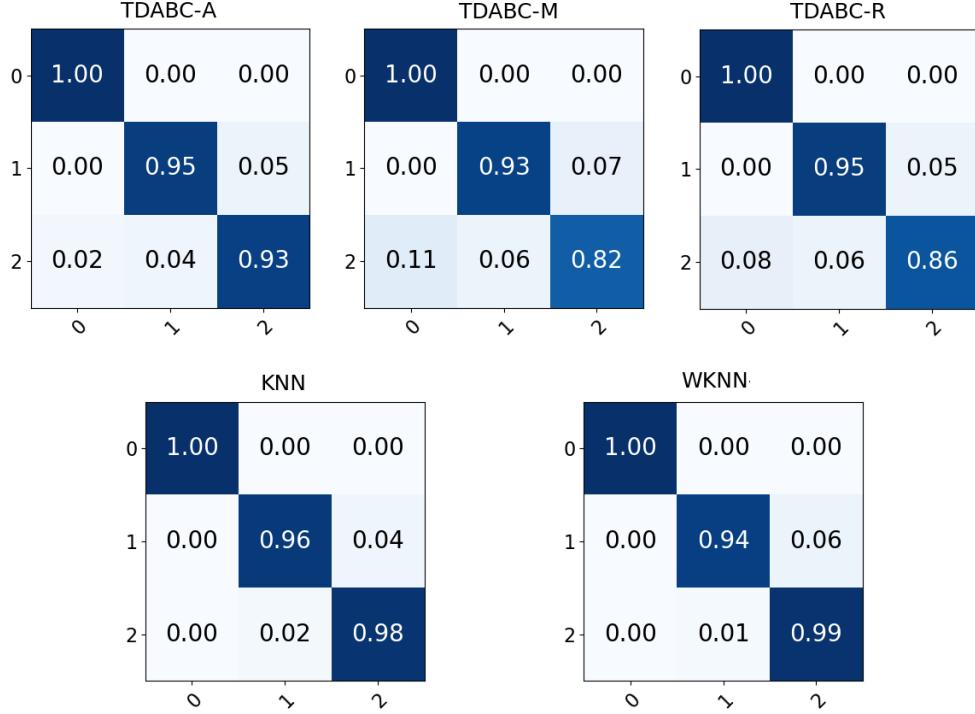


Figure 20: Confusion matrices from classifiers’ results on the Iris dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 8$.

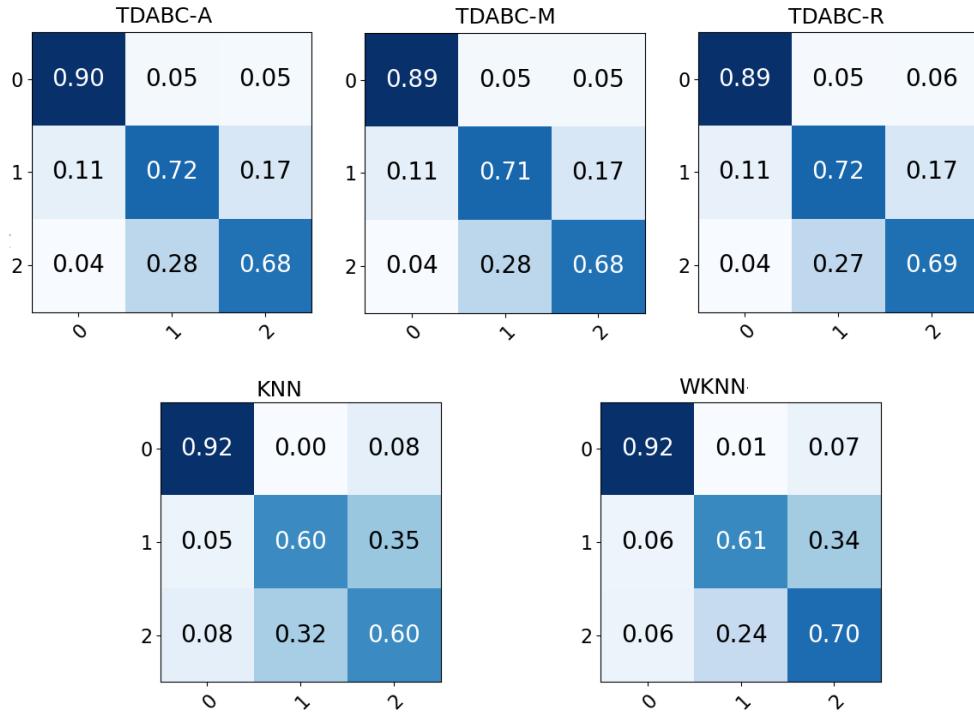


Figure 21: Confusion matrices from classifiers’ results on the Wine dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 6$.

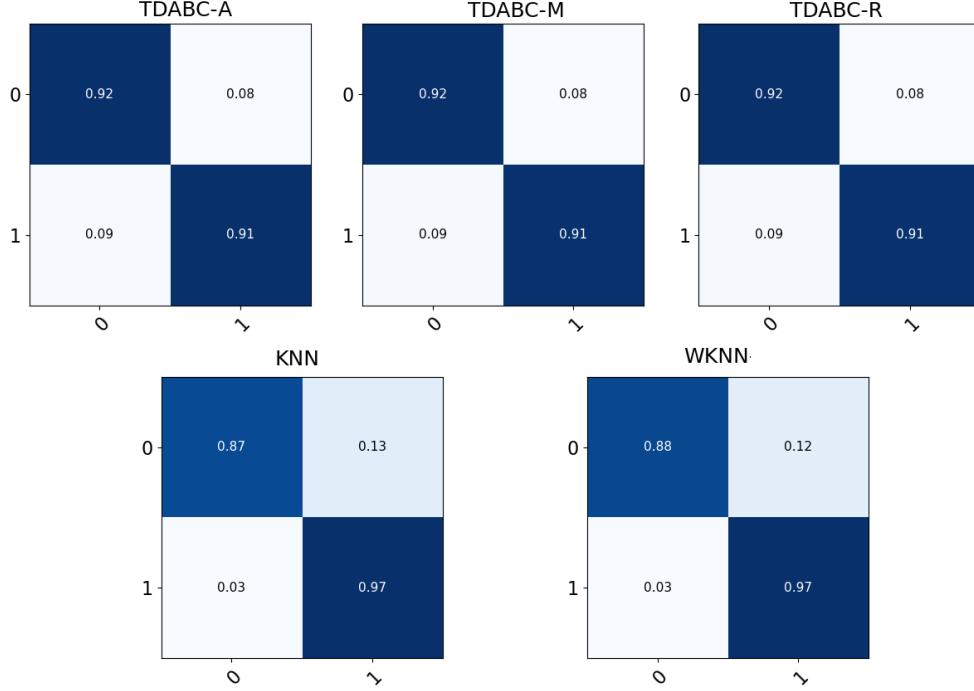


Figure 22: Confusion matrices from classifiers’ results on the Breast Cancer dataset. The filtered simplicial complexes used by TDABC classifiers were built up to a maximal dimension $q = 4$.

References

- [1] Nieves Atienza, Rocio Gonzalez-Díaz, and Manuel Soriano-Trigueros. On the stability of persistent entropy and new summary functions for topological data analysis. *Pattern Recognition*, 107:107509, 2020.
- [2] Jean-Daniel Boissonnat, Tamal K. Dey, and Clément Maria. The compressed annotation matrix: An efficient data structure for computing persistent cohomology. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms – ESA 2013*, pages 695–706, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [3] Jean-Daniel Boissonnat and Karthik C. S. An efficient representation for filtrations of simplicial complexes. *ACM Trans. Algorithms*, 14(4):44:1–44:21, 2018.
- [4] Jean-Daniel Boissonnat, Karthik C. S., and Sébastien Tavenas. Building efficient and compact data structures for simplicial complexes. *Algorithmica*, 79(2):530–567, 2017.
- [5] Jean-Daniel Boissonnat and Clément Maria. The simplex tree: An efficient data structure for general simplicial complexes. *Algorithmica*, 70(3):406–427, 11 2014.
- [6] Michael W. Browne. Cross-validation methods. *Journal of Mathematical Psychology*, 44(1):108 – 132, 2000.
- [7] Claire Caillerie and Bertrand Michel. Model selection for simplicial approximation. *Foundations of Computational Mathematics*, 11(6):707–731, Dec 2011.
- [8] Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 01 2009.
- [9] Gunnar E. Carlsson and Rickard Brüel Gabrielson. Topological approaches to deep learning. *CoRR*, abs/1811.01122, 2018.
- [10] Hamish Carr, Chistoph Garth, and Tino Weinkauf. *Topological Methods in Data Analysis and Visualization IV. Theory, Algorithms, and Applications*. Mathematics and Visualization. Springer International Publishing, Gwerbestrasse, Switzerland, 2017.
- [11] Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete & Computational Geometry*, 45(4):737–759, 06 2011.
- [12] Tamal Dey, Fengtao Fan, and Yusu Wang. Computing topological persistence for simplicial maps. *ArXiv*, abs/1208.5018, 2014.

- [13] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.
- [14] Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction*. American Mathematical Society, Michigan, USA, 2010.
- [15] Herbert Edelsbrunner and Dmitriy Morozov. Persistent homology: theory and practice. *European Congress of Mathematics*, pages 31–50, 01 2014.
- [16] Ronald A. Fisher. *UCI Machine Learning Repository: Iris Data Set*, January 2011.
- [17] Rickard Brüel Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, and Primoz Skraba. A topology layer for machine learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 1553–1563, Online, 08 2020. PMLR.
- [18] Kathryn Garside, Robin Henderson, Irina Makarenko, and Cristina Masoller. Topological data analysis of high resolution diabetic retinopathy images. *PLOS ONE*, 14(5):1–10, 05 2019.
- [19] Robert Ghrist. Barcodes: The persistent topology of data. *BULLETIN (New Series) OF THE AMERICAN MATHEMATICAL SOCIETY*, 45:61–75, 02 2008.
- [20] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS’17, page 1633–1643, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [21] Nathalie Japkowicz and Mohak Shah. *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, New York, NY, USA, 2011.
- [22] Rolando Kindelan, Mauricio Cerdá, and Nancy Hitschfeld. Preliminary results on classification based on topological data analysis. Technical Report TR/DCC-2020-2, University of Chile, https://www.dcc.uchile.cl/TR/2020/TR_DCC-20201209-002.pdf, dec 2020. Submitted to 36th SoCG - Zürich, Switzerland - June 23-26, 2020.
- [23] Rolando Kindelan, Mauricio Cerdá, and Nancy Hitschfeld. Topological data analysis based classification: preliminary results. Technical Report TR/DCC-2020-2, University of Chile, https://www.dcc.uchile.cl/TR/2020/TR_DCC-20201209-002.pdf, dec 2020. Submitted to the Poster session of the Foundations of Computational Mathematics (FoCM) –Vancouver, Canada June 15 -June 24, 2020.
- [24] Sourav Majumdar and Arnab Kumar Laha. Clustering and classification of time series using topological data analysis with applications to finance. *Expert Systems with Applications*, 162:113868, 2020.
- [25] Clément Maria. Filtered complexes. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015.
- [26] Clément Maria, Jean-Daniel Boissonnat, Marc Glisse, and Mariette Yvinec. The gudhi library: Simplicial complexes and persistent homology. In Hoon Hong and Chee Yap, editors, *Mathematical Software – ICMS 2014*, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- [27] Tom M. Mitchell. *Machine learning, International Edition*. McGraw-Hill Series in Computer Science. McGraw-Hill, 1997.
- [28] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, 08 2017.
- [29] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 01 2012.
- [30] Henri Riihimäki, Wojciech Chacholski, Jakob Theorell, Jan Hillert, and Ryan Ramanujam. A topological data analysis based classification method for multiple measurements. *BMC Bioinformatics*, 21(1):336, Jul 2020.
- [31] Vincent Rouvreau. Cython interface. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2016.
- [32] David Salinas. Contraction. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015.
- [33] Lee M. Seversky, Shelvi Davis, and Matthew Berger. On time-series topological data analysis: New data and opportunities. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1014–1022, 06 2016.
- [34] Yuhei Umeda. Time series classification via topological data analysis. *Transactions of the Japanese Society for Artificial Intelligence*, 32:D-G72_1, 05 2017.

- [35] Vinay Venkataraman, Karthikeyan Natesan Ramamurthy, and Pavan K. Turaga. Persistent homology of attractors for action recognition. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*, pages 4150–4154, 2016.
- [36] Pattaramon Vuttipittayamongkol and Eyad Elyan. Neighbourhood-based undersampling approach for handling imbalanced and overlapped data. *Information Sciences*, 509:47–70, Jan 2020.
- [37] Xiuzhen Zhang, Yuxuan Li, Ramamohanarao Kotagiri, Lifang Wu, Zahir Tari, and Mohamed Cheriet. Krnn: k rare-class nearest neighbour classification. *Pattern Recognition*, 62:33–44, Feb 2017.
- [38] Afra Zomorodian. *Topology for Computing*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2009.
- [39] Afra Zomorodian. Fast construction of the vietoris-rips complex. *Computers & Graphics*, 34:263–271, 06 2010.