

# Importance Attribution in Neural Networks by Means of Persistence Landscapes of Time Series

Aina Ferrà<sup>1\*</sup>, Carles Casacuberta<sup>1</sup> and Oriol Pujol<sup>1</sup>

<sup>1</sup>Departament de Matemàtiques i Informàtica, Universitat de  
Barcelona, Gran Via de les Corts Catalanes 585, 08007,  
Barcelona, Spain.

\*Corresponding author(s). E-mail(s): [aina.ferra.marcus@ub.edu](mailto:aina.ferra.marcus@ub.edu);  
Contributing authors: [carles.casacuberta@ub.edu](mailto:carles.casacuberta@ub.edu);  
[oriol\\_pujol@ub.edu](mailto:oriol_pujol@ub.edu);

## Abstract

This article describes a method to analyze time series with a neural network using a matrix of area-normalized persistence landscapes obtained with topological data analysis. The network's architecture includes a gating layer that is able to identify the most relevant landscape levels for a classification task, thus working as an importance attribution system. Next, a matching is performed between the selected landscape levels and the corresponding critical points of the original time series. This matching enables reconstruction of a simplified shape of the time series that gives insight into the grounds of the classification decision. As a use case, this technique is tested in the article with input data from a dataset of electrocardiographic signals. The classification accuracy obtained using only a selection of landscape levels from data was **94.00%  $\pm$  0.13** averaged after five runs of a neural network, while the original signals achieved **98.41%  $\pm$  0.09** and landscape-reduced signals yielded **97.04%  $\pm$  0.14**.

**Keywords:** Persistent homology · Persistence landscape · Neural network · Importance attribution

**MSC Classification:** 68T07 68T09 55N31 62R40 62R07

## 2 *Importance attribution in neural networks*

**Acknowledgments.** The authors were supported by the Agencia Estatal de Investigación (MCIN/AEI) under grant PRE2020-094372 (A. Ferrà) and projects PID2019-105093GB-I00 (A. Ferrà, O. Pujol) and PID2020-117971GB-C22 (C. Casacuberta).

# 1 Introduction

In this article we use topological data analysis (TDA) for the purpose of interpretability of classification results in deep learning. More precisely, we use persistence landscapes to retrieve information about features from data on which a neural network focuses to perform a classification task.

While the use of topological methods to enhance the performance of neural networks is widespread, this is the first study, to our knowledge, in which TDA-based algorithms have been implemented for importance attribution.

**Related work.** A number of articles have used TDA in connection with neural networks since 2018. Tracking changes in the topology of a dataset as it passes through the layers of a trained neural network is the subject of [1], while the topology of neuron activations is analyzed in [2]. Assessment of the generalization gap by means of persistence descriptors without the need of a testing set is discussed in [3, 4]. None of these articles, however, addresses attribution of importance based on classification outcomes.

The use of landscapes as persistence descriptors was initiated by Bubenik in [5]. Landscapes were used in connection with deep learning in [6] with the goal of improving learnability by adding information on topological features of input data into subsequent layers, but not for explainability purposes either. Activation landscapes have also been used as topological summaries of performance of neural networks in [7].

Many articles address the study of time series by means of neural networks without using topology. For example, in [8], a pre-training method using auto-encoders was designed for time series prediction, and in [9] a multilayer feedforward perceptron neural network was used to assess its capability of accurately predicting stock market short-term trends.

A survey of topological methods for time-series analysis in deep learning using Betti numbers is offered in [10]. Persistent homology is used in [11] to detect and quantify topological patterns in time series of financial crashes, and for personalized arrhythmia classification in [12]. In different directions, methods from topological data analysis have also been used to provide versatile vectorizations [13], or to achieve a higher prediction accuracy or classification accuracy [14], or to regularize learning algorithms by feeding topological information extracted from data [15–17]. Topology has also been used to reduce the size of datasets without much loss in training accuracy [18].

In contrast with most of the aforementioned articles, the purpose of the present paper is neither to achieve an increased classification accuracy nor to investigate any aspects of the structure of a neural network, but rather to link classification outcomes with specific topological characteristics of the dataset.

**Problem statement.** While the reasons for a classification outcome from a neural network often remain unknown, it is feasible to determine which features of data were especially relevant after training a network. The purpose of this article is twofold: First, to design a mechanism for importance attribution using persistence descriptors, and second, to ascertain whether such descriptors (or a skeleton of data focusing on selected descriptors) achieve a similar classification accuracy through the same architecture.

**Research approach and methods.** The hierarchical structure of persistence landscapes allows us to design a method for finding the most informative levels. For this, we preprocess data so that the network is fed with a persistence landscape extracted from data instead of the original signals. Furthermore, we introduce an additional layer to a chosen architecture, whose mission is to assign weights to landscape levels. Then we run again the network using only those levels with the highest weights. The results show that the set of selected landscape levels (normally 2 to 4) yield similar classification accuracies as the whole landscape.

Selecting the most relevant landscape levels for a deep learning classification task opens the possibility of reconstructing the given data using only the chosen landscape functions. The resulting simplified version of the given data sheds light on which parts of data signals were most relevant for the network's classification task. Our reconstruction method is described more precisely in a companion article [19], which addresses some mathematical questions related with the present paper and is related with the inverse problem in TDA, namely recovering certain types of data from persistence summaries [20–22].

In the context of a heartbeat analysis (Section 4.2) we checked that our neural network obtains similar accuracies when fed with reconstructions of signals from selected landscape levels in comparison with those obtained with raw data. This enhances confidence in the classification results by providing evidence that the network is not focusing on artifactual details during the learning process.

**Outline.** Basic facts about persistence landscapes are collected in Section 2, and our attribution algorithm for landscape levels is described in Section 3. In Subsection 4.1 we validate our technique with nine datasets from the UCR Time Series Classification Archive [23], and use it in Subsection 4.2 to test the accuracy of classification of electrocardiographic signals from the MIT-BIH Arrhythmia Database [24]. In Subsection 4.3, the effect of shifting signals on classification accuracy is analyzed.

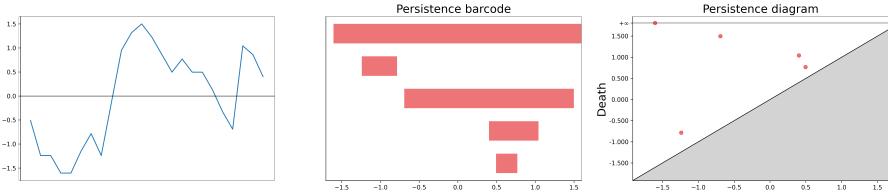
## 2 Persistence landscapes for sublevel sets

Time-series arrays can be viewed as one-dimensional continuous piecewise linear functions where persistent homology can be applied to study the evolution of sublevel sets. Thus we consider a sliding parameter  $t$  along the  $y$ -axis, and for each function  $f$  defined on an interval  $[a, b]$  and each value of  $t$  we compute the number of connected components of the corresponding sublevel set

#### 4 Importance attribution in neural networks

$L_t(f) = \{x \in [a, b] \mid f(x) \leq t\}$ . This coincides with the number of connected components of the part of the graph of  $f$  which lies at or below height  $t$ . The collection of all sublevel sets for a given function yields a persistence module whose value at  $t$  is the vector space  $H_0(L_t(f); \mathbb{R})$ , where  $H_0$  denotes zero-dimensional homology and coefficients in the field  $\mathbb{R}$  of reals are used.

For background about persistence modules and their associated barcodes and persistence diagrams, see [25]. Barcodes were first considered in a topological context in [26]. A *barcode* depicts the lifetime of each connected component of a sublevel set, from the height  $t = b$  (birth) where it appears until the height  $t = d$  (death) in which it merges with some other connected component. The corresponding *persistence diagram* contains a point  $(b, d)$  for each barcode line starting at  $b$  and ending at  $d$ . The infinite ray depicting the essential homology class that survives to infinity is discarded for practical purposes.



**Fig. 1** From left to right, a piecewise linear function, its barcode of zero-dimensional homology of sublevel sets and the corresponding persistence diagram.

Persistence diagrams are not optimal for their use in deep learning. Neural networks perform best with array-shaped data. Therefore, in this article we use *landscapes* as persistence summaries. Persistence landscapes were defined in [5] and, in our case, they express the evolution of connected components of sublevel sets of signals by means of a finite sequence of continuous piecewise linear functions with compact support. Computationally, each landscape function can be expressed as an array of discretized values, which makes it suitable to be introduced into a deep learning system.

The sequence of landscape functions associated with a persistence diagram is defined as follows. For each point  $(b, d)$  in the persistence diagram, one considers the corresponding *tent function*

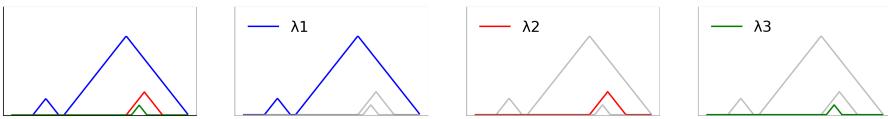
$$\Lambda_{(b,d)}(t) = \max\{0, \min\{t - b, d - t\}\}.$$

Next, a piecewise linear function  $\lambda_k : \mathbb{R} \rightarrow \mathbb{R}$  is defined for each  $k \geq 1$  as

$$\lambda_k(t) = \text{kmax}\{\Lambda_{(b,d)}(t)\},$$

where `kmax` returns the  $k$ -th largest value of a given set of real numbers whose elements are counted with multiplicities, or zero if there is no  $k$ -th largest value. Therefore, since the number of points in a persistence diagram is finite,  $\lambda_k = 0$  for all sufficiently large values of  $k$ . The first landscape levels  $\lambda_1, \lambda_2 \dots$

depict the most persistent topological features, while the last ones correspond to less persistent phenomena.



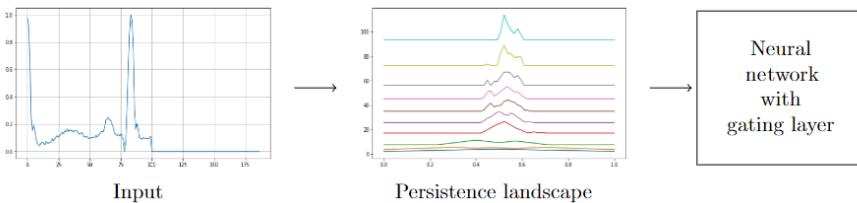
**Fig. 2** Sequence of nonzero levels  $\lambda_k$  of a persistence landscape (left).

### 3 Attribution of importance

The fact that persistence landscapes can be stratified into a hierarchical sequence of levels makes it possible to design a mechanism for importance attribution ranking landscape levels of a given sample of signals. In [19] a deterministic procedure is described to reconstruct signals from directional persistence landscapes in a number of chosen directions. It is also shown in [19] how to partially reconstruct the given signals using only a subset of selected landscape levels, which is the focus of interest in the present article. By combining this procedure with a machine learning assignment of a sequence of weights to landscapes, we achieve a substantial reduction of the number of critical points of the given data functions without losing much classification accuracy.

To do this, we stack landscape functions from persistence of sublevel sets of the given signals in a matrix that will be fed into a neural network. Landscapes provide a convenient representation, since each landscape level corresponds to a different region of the oscillation of the input signal.

Since our objective is to feed a deep learning model, we decided to normalize the area under each landscape function in order to force the network to focus on their morphology instead of their actual values. This process is illustrated in Fig. 3.



**Fig. 3** Extracting information through persistence landscapes to feed a neural network.

The existence of different levels of information naturally leads to the study of which levels are more important than others for the classification task.

## 6 *Importance attribution in neural networks*

In order to implement this idea, we propose the use of a *gating layer*: we maintain the matrix shape throughout the architecture and, before applying the fully connected layers, each landscape level  $\lambda_k$  is multiplied by a positive less-than-one learnable weight  $w_k$ . Thus we obtain a set of weights that indicate how influential is each landscape level for the classification task. Typically, a network should regard the first landscape levels as more important than the last ones, given that the first levels contain information about the most persistent topological features.

By building a ranking of landscape levels, we are able to decide at which threshold of information the network stops learning. This is helpful in two main ways: first, we are able to reduce the information that we use to train our system by reducing the number of landscape functions that we pass to our network; and second, we can attribute importance to the parts of the original data that are producing the most relevant landscape levels.

## 4 Experimental setting and results

In this section, we present the results of our experiments using a neural network with a fixed architecture and different input signals. Our main aims are to assess the changes in classification accuracy by using only a set of selected landscape levels in comparison with the full landscape and with the original data, while determining which are the most relevant landscape features in each database. Robustness of our method is estimated by applying it to nine databases of very different nature.

**Data.** We applied our methodology to a collection of datasets taken from the UCR Time Series Classification Archive [23]. The criteria for choosing a dataset were the following: the dataset should have at most five different classes and the total number of samples divided by the number of classes should be greater than or equal to 500. These criteria were adopted in order to avoid dealing with data scarcity problems and difficulties caused by imbalanced classes or by an excessive number of classes. Table 1 contains a summary of the characteristics of each dataset.

**Methodology.** In order to avoid discrepancies in the accuracy of the method due to the different ranges of values among datasets, input functions have been standardized to have values between 0 and 1. Moreover, when the topological preprocessing is applied, landscapes have been normalized so that the area under each landscape function is equal to 1. In doing so, we force the neural network to study the shape of the landscape, rather than only taking into account its actual values.

The main objective of our study is to compare the ability of landscape levels to capture information against a baseline of the raw data with the only pre-processing of standardization. Furthermore, to assess if the selected landscape levels are sufficient to classify, the results of feeding a neural network with the full landscape and the results of using only the selected levels are compared.

**Table 1** A summary of the characteristics of each dataset. For each dataset we indicate the total number of samples, the length of each sample, the number of classes and whether the dataset is imbalanced or not.

Dataset	Samples	Length	Classes	Imbalanced
<b>ECG5000</b>	5000	140	5	Yes
<b>FreezerRegularTrain</b>	3000	301	2	No
<b>HandOutlines</b>	1370	2709	2	Yes
<b>ItalyPowerDemand</b>	1096	24	2	No
<b>MoteStrain</b>	1272	84	2	No
<b>PhalangesOutlinesCorrect</b>	2658	80	2	Yes
<b>StarLightCurves</b>	9236	1024	3	Yes
<b>Wafer</b>	7164	152	2	Yes
<b>Yoga</b>	3300	426	2	Yes

The architecture of the neural network is as follows: three convolutional layers combined with row-preserving max pooling layers followed by two dense layers (Fig. 4). Our gating layer is used for selection and attribution purposes and it is only present when landscapes levels are used as input. In such case, the gating layer is placed between the last max pooling layer and the first dense layer. The experiments are conducted using a 5-fold cross-validation. Training sets amount to 80% of each dataset. The neural network is trained during 240 epochs, with a starting learning rate of 0.01 that is divided by 5 every 100 epochs. This architecture has been chosen to be rather generic, without attempting to achieve the highest possible accuracy, neither with the original data nor by means of landscapes. Our purpose was to assess the validity our method while avoiding possible particularities due to a tailored choice of an optimal architecture.

As for performance metrics, only accuracy is taken into account in the present article.



**Fig. 4** Architecture of the neural network designed for this study. The gating layer is placed immediately before the first dense layer (pink) when landscapes are used as input.

## 4.1 Validation of the method

### 4.1.1 Performance results

We carried out the same experiment for 9 different datasets from [23] to verify the stability of the results (Table 2). For each dataset, we ran a neural network (Fig. 4) with three different inputs: the original data, a sequence of persistence landscape levels, and a selected subset of levels. Since the length of the full sequence of nonzero landscape levels is variable, we chose the first 10 levels  $\lambda_1, \dots, \lambda_{10}$  as in most cases the 10th level was already zero, and fixing a larger number of landscape levels caused memory difficulties during the training process without a significant increase in accuracy.

Subsequently, the selection of a smaller number of principal landscape levels was made by choosing the highest weights provided by the gating layer. The number of selected levels ranged from 2 to 5 depending on the dataset (Fig. 5). Further details about the selection of an appropriate subset of landscape levels are given in Subsection 4.1.2.

Table 2 shows the average accuracy and standard deviation of each experiment using 5-fold cross-validation. The table contains average accuracy results using raw data, unnormalized landscapes, normalized landscapes, and a selected subset of normalized landscape levels. The results show that landscapes achieve sufficiently high classification accuracies, especially when they are normalized (third and fourth columns). In that respect, landscape accuracies are statistically comparable up to one standard deviation to using raw data in four out of the nine datasets.

In Table 2, the results obtained by TDA-based strategies that are statistically comparable among them—including the method that achieved maximum accuracy—are highlighted in bold font. Unnormalized landscapes consistently miss relevant information in most cases, and this is translated into a significant reduction in accuracy. It is also remarkable that the selected landscape levels achieve similar performances as whole (normalized) landscapes. This reinforces the hypothesis that most of the information contained in data is captured by a small subset of landscape levels.

In the PhalangesOC dataset, normalized landscapes perform even better than the original data. As pointed out in the Discussion, this could be due to the inherent elastic deformation invariance provided by the landscape representation.

### 4.1.2 Ranking of landscape levels

The keystone of our process is to be able to identify which landscape levels carry the highest amount of information for classification outcomes. The gating layer multiplies each landscape level  $\lambda_k$  (with  $k = 1, \dots, 10$ ) by a learnable weight  $w_k$  with  $0 \leq w_k \leq 1$ . After the full training process of the neural network, the resulting weights are used to attribute importance to each landscape level.

**Table 2** Average accuracies given as percentages and standard deviations on test sets from five runs of a neural network (Fig. 4) for nine signal datasets. Accuracies obtained from original data (first column) are compared with those obtained from the first 10 landscape levels without area normalization (second column) and with area normalization (third column), and from the most informative landscape levels (fourth column). The last column indicates how many landscape levels were selected in each case. Statistically comparable accuracies among TDA-based strategies appear in boldface.

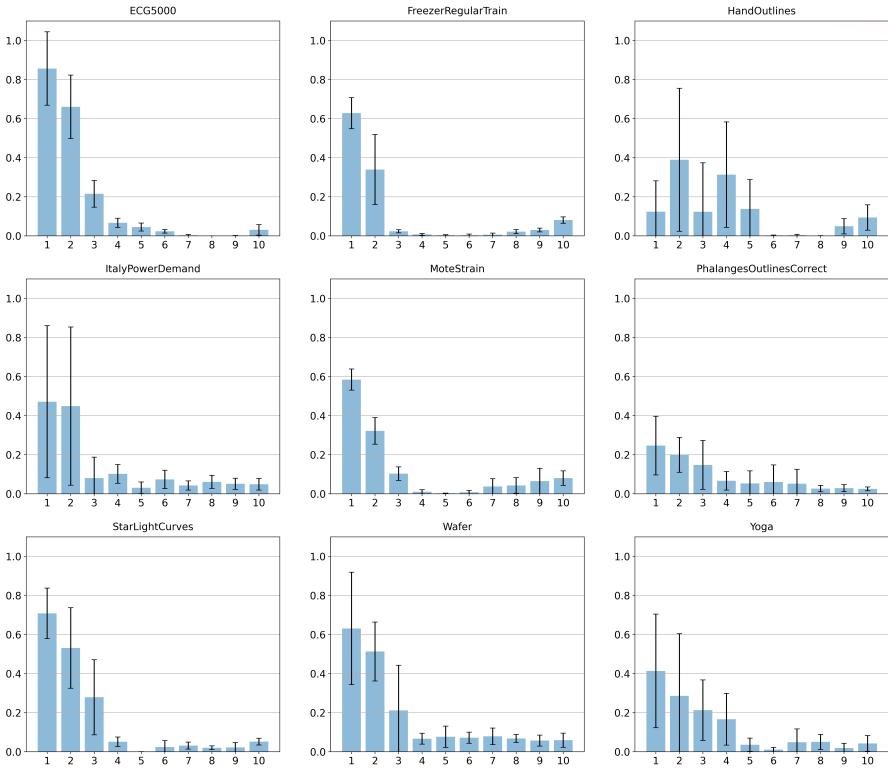
Dataset	Raw data	Unnormalized	Normalized	Selected	No.
<b>ECG5000</b>	$94.72 \pm 0.7$	<b><math>92.96 \pm 0.5</math></b>	<b><math>93.12 \pm 0.4</math></b>	<b><math>92.88 \pm 0.3</math></b>	3
<b>FreezerRT</b>	$99.53 \pm 0.4$	$63.70 \pm 4.2$	<b><math>88.97 \pm 0.3</math></b>	<b><math>88.70 \pm 1.6</math></b>	2
<b>HandOutlines</b>	$89.20 \pm 1.7$	$75.77 \pm 4.6$	<b><math>85.26 \pm 2.0</math></b>	$81.90 \pm 2.0$	5
<b>ItalyPowerD</b>	$97.73 \pm 1.1$	$87.18 \pm 3.4$	<b><math>89.45 \pm 1.7</math></b>	<b><math>90.36 \pm 1.3</math></b>	2
<b>MoteStrain</b>	$90.98 \pm 1.1$	$71.84 \pm 1.8$	<b><math>77.33 \pm 2.8</math></b>	<b><math>76.31 \pm 1.7</math></b>	3
<b>PhalangesOC</b>	$64.02 \pm 2.1$	$63.95 \pm 4.2$	<b><math>68.95 \pm 0.9</math></b>	<b><math>69.21 \pm 0.8</math></b>	4
<b>StarlightC</b>	$95.70 \pm 0.4$	$89.82 \pm 2.3$	<b><math>94.92 \pm 0.1</math></b>	<b><math>95.22 \pm 0.5</math></b>	3
<b>Wafer</b>	$99.65 \pm 0.2$	$90.86 \pm 0.8$	<b><math>98.63 \pm 0.3</math></b>	<b><math>98.79 \pm 0.4</math></b>	3
<b>Yoga</b>	$82.48 \pm 1.6$	$64.21 \pm 4.1$	$75.36 \pm 1.1$	<b><math>78.33 \pm 1.0</math></b>	4

To ensure significance, we performed the experiment five times and recorded the mean weight value and standard deviation for each landscape level, as seen in Fig. 5. Although there is no obvious numerical method to determine the number of landscape levels that should be considered important in view of their weights, we used the following criterion. If  $w_k < \frac{1}{2}w_{k-1}$  for some  $k$ , we call  $k$  a *significant drop*. If  $k$  is the largest significant drop with  $w_{k-1} > 0.1$ , then we select  $\lambda_1, \dots, \lambda_{k-1}$  as most important landscape levels. If there is no significant drop with  $w_{k-1} > 0.1$ , then we pick the smallest  $k$  such that  $w_1 + \dots + w_{k-1} > w_k + \dots + w_{10}$  and also select  $\lambda_1, \dots, \lambda_{k-1}$ .

With very few exceptions, the network regards the first landscape levels as more important. These contain information of the most persistent topological features of each signal (connected components of sublevel sets). The first 10 levels were used in all the experiments. In some cases —namely, ItalyPowerD and PhalangesOC— landscape levels  $\lambda_k$  with  $k > 6$  were zero for all samples in the dataset. In these cases, the gating layer assigned small but not necessarily zero weights to the null levels.

It is remarkable that the terminal landscape level (i.e., the 10th in our study) tends to be consistently more relevant than the immediately precedent ones, except in those cases where it is zero for the whole dataset. This suggests that the terminal landscape level may convey discriminant information, deserving further study.

Fig. 5 shows that for certain datasets all weights are below 0.4, specifically HandOutlines and PhalangesOC, and marginally also Yoga. Looking at Table 2, we find that these datasets are precisely the ones that yield accuracies below 90% on test sets after the neural network had been trained with the original data. The datasets where the original data achieved a higher classification accuracy coincide with those with a smallest number of important landscape

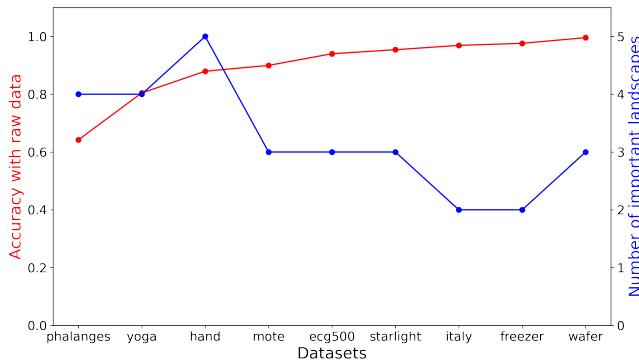


**Fig. 5** Average weights and standard deviations of the first ten landscape levels for nine datasets after five runs of a neural network (Fig. 4) equipped with a gating layer.

levels. Indeed, Fig. 6 shows an inverse relationship between accuracies and the number of selected landscape levels.

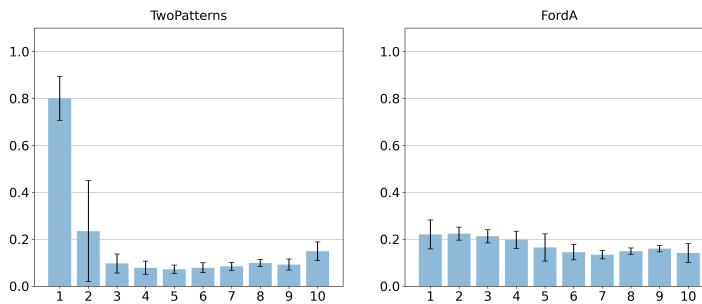
As examples of unfavorable cases, we now discuss results obtained with the datasets FordA and TwoPatterns from [23]. These datasets share a common property, namely they consist of wave-like signals with a varying wavelength and the key information to classify them is the  $x$ -coordinate where the changes in the waves are happening. In one of them (FordA), the original data are difficult to classify, while in the other one (TwoPatterns) the original data are easily classifiable. In both cases, replacing the data by persistence landscapes erases the relevant information for a neural network classifier —since landscapes are invariant under wavelength changes if amplitude is preserved— and thus we obtain low accuracy and considerable overfitting if landscapes are used instead of raw data.

In Fig. 7 we see that, for the FordA datasets (where the neural network has trouble classifying even with the original data) the weights of persistence landscape levels are all similar and with a low relevance. In contrast, in the TwoPatterns case we see a clear ranking of the first landscape levels. Hence, landscape selection yields meaningful information about the dataset



**Fig. 6** Inverse relationship between the accuracy of our neural network (red) trained with the original raw data and the number of landscape levels (blue) that were selected as important. Datasets in the horizontal axis are ordered by increasing accuracy.

even in disadvantageous situations, since there is a consistent inverse relationship between the ability of the neural network to correctly classify the original data and the number of important landscape levels found through our method. In conclusion, Fig. 6 and Fig. 7 provide evidence that the outcome of landscape level selection can be related to how well a neural network can perform.



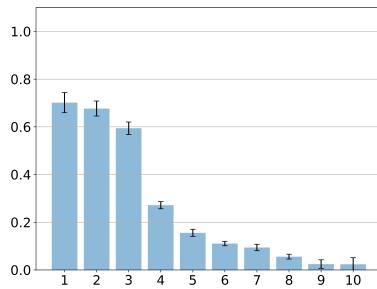
**Fig. 7** Average weights and standard deviations of the first ten landscape levels for two datasets after five runs of a neural network (Fig. 4) equipped with a gating layer.

## 4.2 A use case: Results of a heartbeat analysis

As an application case, we used our algorithm for a classification of electrocardiogram signals (ECG) from the MIT-BIH Arrhythmia Database [24] for evaluation of arrhythmia detectors. The dataset can be retrieved from [27] and it includes 48 half-hour excerpts of 24-hour ECG recordings obtained from 47 subjects (25 men aged 32 to 89 years and 22 women aged 23 to 89 years) studied between 1975 and 1979. Our data sample includes 87,554 heartbeats of five

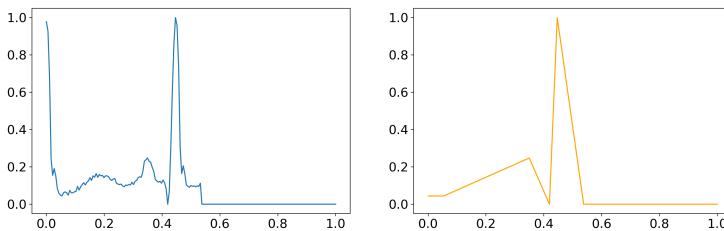
classes: one corresponding to normal beats (82.77%); three classes corresponding to different arrhythmia types, namely supraventricular premature beats (2.54%), premature ventricular contraction (6.61%), and fusion of ventricular and normal beats (0.73%); and one class for unidentifiable heartbeats (7.35%).

Table 3 shows average accuracy after a 5-fold cross-validation. The classification accuracy of our neural network (Fig. 4) fed with the original unprocessed signals (98.41%) is compared with the accuracy of the same architecture using a 10-level landscape (94.55%) and using only the three most important landscape levels (94.00%). Landscapes were area-normalized since Table 2 evidenced an advantage of normalized landscapes versus unnormalized ones. The choice of three levels was based on weights assigned by the network, as shown in Fig. 8, where  $k = 4$  is the largest significant drop.



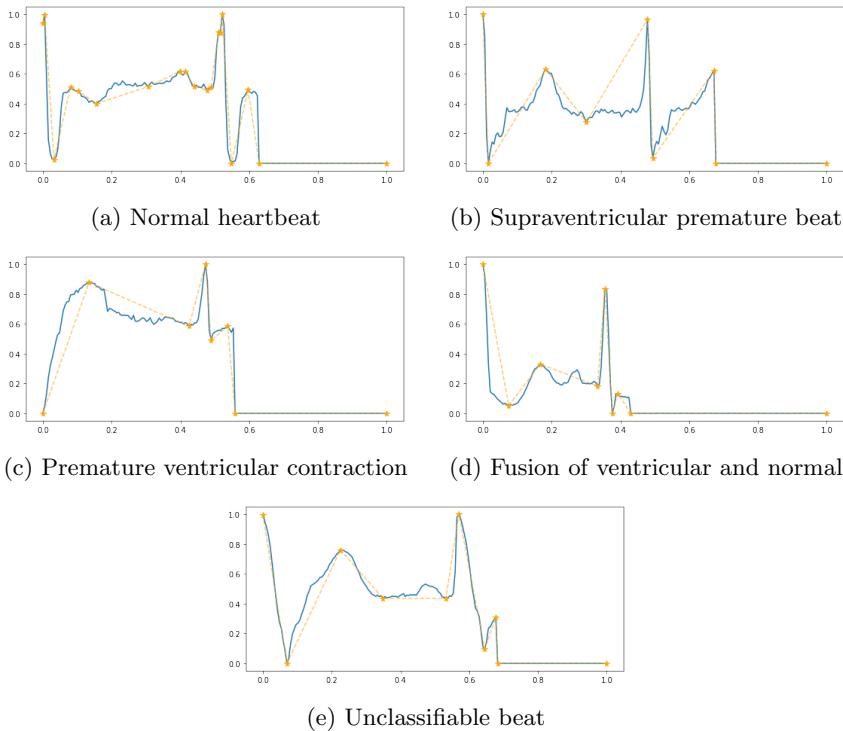
**Fig. 8** Average weights and standard deviations of the first ten landscape levels for a sample of the MIT-BIH Arrhythmia Database after five runs of a neural network (Fig. 4).

Next we used the partial reconstruction technique described in detail in [19, Section 3] in four examples, corresponding to the classes of (a) normal heartbeats, (b) supraventricular premature beats, (c) premature ventricular contraction, and (d) fusion of ventricular and normal beats. Three landscape levels were used for approximation in each case. Results are shown in Fig. 10.



**Fig. 9** An ECG signal function (left) and its approximate reconstruction from a set of selected landscape levels (right).

Each landscape function  $\lambda_k$  was paired with a list of  $y$ -values of critical points of the given signal  $f$  as specified in [19, Proposition 3.1]. Hence we



**Fig. 10** Partial reconstruction of ECG graphs using the three most important landscape levels for each of four types of heartbeats.

obtained a list of  $y$ -values of critical points of  $f$  associated with the subset of selected landscape levels. The values in this list were compared with the list of all critical points of  $f$  in order to obtain the matching  $x$ -values, and a new graph was drawn by joining the resulting critical points of  $f$  in the order of their  $x$ -coordinates, as in Fig. 9. The procedure is detailed below in Algorithms 1, 2 and 3. The resulting simplified graphs (Fig. 10) mark the points of interest, according to the neural network used in our experiment, for the classification of ECG samples. Thus they encode the most relevant information on which the network focused for its task.

**Table 3** Average accuracy of classification given in percentages and standard deviation on test sets from five runs of our neural network (Fig. 4). The network was fed with unprocessed data (first column); processed data with ten landscape levels (second column); processed data using the most significant three landscape levels (third column); and data approximately reconstructed by means of the most significant three landscape levels (fourth column).

	Raw data	10 levels	3 levels	Reconstructed
Accuracy	98.41 $\pm$ 0.09	94.55 $\pm$ 0.16	94.00 $\pm$ 0.13	97.04 $\pm$ 0.14

We subsequently introduced the simplified reconstructions of the wave functions (Fig. 9) into the network in order to check if the data features distilled by our reconstruction method were sufficient for the network’s classifications task. The results can be seen in Table 3 and indicate that the simplified signals gave rise to similar accuracies (97.04%) as the original data (98.41%).

### 4.3 Invariance under translations

Persistence summaries are not altered by horizontal shifts of signals and hence the accuracy of a classification task based on landscapes is invariant under such shifts. However, shifts may cause a loss of classification accuracy by a neural network fed with the original data. To demonstrate this effect, we used the same ECG dataset from Subsection 4.2, yet we modified each heartbeat by adding a number of zeros randomly split between the beginning and the end of the beat signal. Thus, while in the original dataset each heartbeat was represented by a vector of length 187, in our experiment we introduced zeros so that the length was increased to 374.

**Table 4** Accuracies (given in percentages) of our neural network fed with unmodified data versus modified data by inserting zero segments at the beginning and end of each signal so as to duplicate the length of the signals (second column).

	Raw data	Double length
Accuracy	98.41 $\pm$ 0.09	96.77 $\pm$ 0.10

Classification of the shifted ECG graphs by means of the same neural network as in Subsection 4.2 with five repetitions resulted in lower accuracy (Table 4) than with the original data. However, shifts do not alter the evolution of connected components of sublevel sets and therefore the landscapes associated with the shifted graphs are the same as those of the original data.

## 5 Discussion

Our results contribute to explainability of classification outcomes by neural networks in the following ways. First, we found that using whole persistence landscapes is not necessary for an accurate classification of signals: once we have identified the subset of landscape levels that is most important for the network, running the experiment with only this subset of levels yields a statistically comparable accuracy (Table 2).

Secondly, our method allows us to partially reconstruct the given signals using the set of selected landscape levels, thus depicting which features of the data are most relevant for classification by means of the chosen architecture. Persistence descriptors are not injective in general and cannot be used to recover data except in some cases where a collection of directional persistence diagrams are considered [19, 20, 22, 28]. However, in our case we need not fully

reconstruct a function with the only knowledge of its persistence diagram, but our reconstruction task consists of matching points in the persistence landscape with corresponding parts of the given signals.

A methodological novelty of this study in the framework of topological data analysis is normalization of landscape level functions so that the area below their graph is constantly equal to one. This was conceived as an attempt to feed the neural network with *shapes* rather than *magnitudes*. As Table 2 shows, the accuracies obtained with normalized landscapes were higher than those obtained prior to normalization. Furthermore, the standard deviation of accuracy is lower after normalization in most cases, suggesting that normalization enhances stability.

**Limitations.** Since the given signals have been discretized, difficulties regarding numerical precision may arise. Thus, when comparing  $y$ -values of critical points obtained from landscape peaks with those of the original functions, a zero difference cannot be expected. Instead, a threshold  $\varepsilon$  has to be used, whose value may depend on the range of functions in the dataset and on the precision with which landscapes are vectorized.

A feature of our method is that persistence diagrams of sublevel sets of signals do not capture information about the distribution of data along the  $x$ -axis, but only along the  $y$ -axis. This can be a disadvantage for the use of persistent homology in cases when, for example, the wavelength of periodic or almost periodic functions is crucial for classification purposes, as illustrated by the datasets FordA and TwoPatterns in Subsection 4.1.2. However, it can be an advantage if expansion or contraction along the  $x$ -axis produces undesired effects, as in the case of bradycardia and tachycardia in [14] or in the experiment made in Subsection 4.3.

**Computational complexity.** The methodology in this article involves three processes, namely calculating persistence landscapes of sublevel sets, training a neural network, and partially reconstructing signals using selected landscapes. Let  $n$  be the vector length of the original function to be analyzed,  $m$  the number of critical points of this function,  $r$  the length of the discretized landscape vector, and  $k$  the number of landscape levels to be computed. Building a persistence diagram of sublevel sets of a function requires  $\mathcal{O}(n)$  to determine the local extrema;  $\mathcal{O}(m \log m)$  to order the  $y$ -values of extrema;  $\mathcal{O}(m)$  to determine birth and merging of connected components; and  $\mathcal{O}(kr)$  to construct a persistence landscape. As a result, data feed into a neural network requires a preprocessing cost of  $\mathcal{O}(kr + n + m \log m)$ . Here  $r \leq n$ , and the larger the value of  $r$  the more precise is expected to be the classification outcome. Thus, the cost is linear in the resolution of the input signal. The training procedure by feeding a neural network with a matrix of  $k$  landscape levels of each function increases its processing time by  $\mathcal{O}(k)$  in comparison with the processing time required to train with the original data, assuming that  $r = n$ . The reconstruction process requires  $\mathcal{O}(\ell r)$  to explore the selected landscapes and  $\mathcal{O}(\ell nm)$  to locate the corresponding critical points in the original function, where  $\ell \leq k$  is the amount of selected landscape levels. Hence, the complexity of this step is

also linear in terms of the resolution of the input signal. In summary, the computational cost of the proposed methodology is linear or sub-linear in terms of the original signal size.

**Future research.** Middle layers in architectures of neural networks such as max pooling or mean pooling layers in a convolutional neural network are mainly used for two purposes: On one hand, they serve to reduce the size of the inner representation of the input signal; on the other hand, they introduce invariance to scale, translation, and rotation. Taking advantage of the invariance inherent to the use of persistent homology, we plan to explore the use of persistence descriptors —such as landscapes— as middle layers in deep neural networks with the aim of testing whether such layers could replace pooling layers and therewith possibly reduce computation time.

From an applied perspective, we plan to explore the use of persistence summaries in domains in which elastic deformations of signals may hinder discrimination. This is the case, for example, in behavior recognition, activity recognition, or action recognition using wearable inertial measurement units, where the speed of actions does not include discriminative information.

## 6 Conclusion

This article highlights an instance of the usefulness of topological data analysis in machine learning, specifically towards interpretability of outcomes of neural networks. Our procedure enabled us to distill partial information from the given data sufficiently relevant for classification purposes without a significant loss of accuracy. We used landscapes as persistence descriptors of sublevel sets of signals, exploiting the fact that landscapes come with a hierarchy of levels that enables us to rank the importance of each level by means of weights assigned by a gating layer in a neural network.

Importance attribution in conjunction with a reconstruction algorithm uncovers the most relevant features used by a network during training. Additionally, since topological summaries of data are invariant under affine or elastic temporal deformations, they are particularly suitable when significant recognition ingredients rely on shape properties.

Regardless of the effect on performance metrics of the use of persistence descriptors instead of raw data, we gain insight about key patterns used to classify the given data, which makes the process more trustworthy. Thus, our method not only provides information about the focus of the network’s learning process but it also serves to explore and better understand the dataset.

## Data Availability

The datasets used for the study made in this article were retrieved from the following public databases: The UCR Time Series Classification Archive [https://www.cs.ucr.edu/~eamonn/time\\_series\\_data\\_2018/](https://www.cs.ucr.edu/~eamonn/time_series_data_2018/) and the MIT-BIH Arrhythmia Database <https://doi.org/10.1109/51.932724>.

## Statements and Declarations

The authors declare no conflicts of interest and no competing financial interests directly or indirectly related to this work.

## Appendix A Algorithms

---

### Algorithm 1 Get $y$ -values of critical points

---

**Require:** vector of landscape level data  $level$   
**Ensure:** list of  $y$ -values of critical points  $crits$

```

1:  $crits \leftarrow \{\}$ 
2: for  $point \in level$  do
3:   if  $point$  is a take-off vertex then
4:      $crits \leftarrow crits \cup \{point_x\}$ 
5:   else if  $point$  is a local minimum or a local maximum then
6:      $point \leftarrow crits \cup \{point_x\}$ 
7:      $i \leftarrow \text{length}(crits)$ 
8:      $crits_i \leftarrow 2(crits_i - 0.5 crits_{i-1})$   $\triangleright$  Transform suitable  $x$ -coordinates
       of landscapes into  $y$ -coordinates of critical points of original data
9:   end if
10: end for
11: return  $crits$ 

```

---



---

### Algorithm 2 Get $x$ -values of critical points

---

**Require:** list of  $y$ -values of critical points  $crits$ ; vector of original data  $original$ ; precision  $\varepsilon$   
**Ensure:** list of  $x$ -values of selected critical points of original data  $xcrits$

```

1:  $candcrits \leftarrow \{\}$   $\triangleright$  First, get  $x$ -values of original data that have the  $y$ -value
   equal to some critical point
2: for  $crity \in crits$  do
3:    $candcrits \leftarrow candcrits \cup \{x\text{-values of points of original data } \varepsilon\text{-close to}$ 
      $\text{the value } crity\}$ 
4: end for
5:  $xcrits \leftarrow \{\}$   $\triangleright$  Second, keep only the  $x$ -values that actually correspond to
   critical points
6: for  $xpos \in candcrits$  do
7:   if  $original_{xpos}$  is a local minimum or a local maximum then
8:      $xcrits \leftarrow xcrits \cup \{xpos\}$ 
9:   end if
10: end for
11: return  $xcrits$ 

```

---

---

**Algorithm 3** Set of critical points of a univariate function  $f$  associated with a set of landscape levels of  $f$

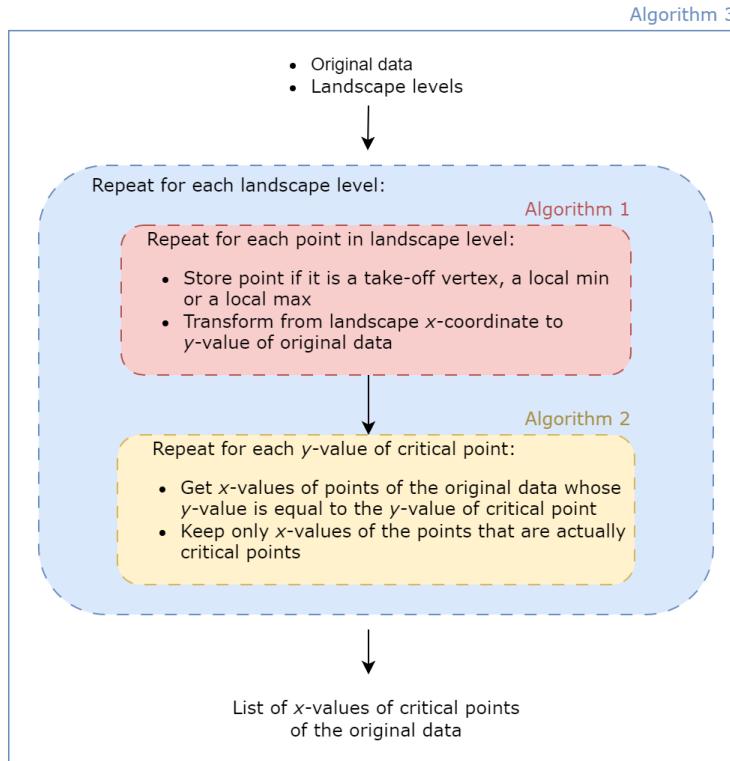
---

**Require:** vector of original data  $original$ ; list of landscape levels  $landscapes$   
**Ensure:** list of  $x$ -values of critical points of the given function  $xcrits$

```

1:  $xcrits \leftarrow \{\}$ 
2: for  $land \in landscapes$  do
3:    $candcrits \leftarrow \{\text{get\_y\_values}(land)\}$ 
4:    $xcrits \leftarrow xcrits \cup \{\text{get\_x\_values}(candcrits, original)\}$ 
5: end for
6:  $xcrits \leftarrow \text{sort}(xcrits)$ 
7: return  $xcrits$ 
```

---



**Fig. A1** Flow diagram of the execution of Algorithms 1–3.

## Appendix B Abbreviations and Notation

ECG	Electrocardiogram
MIT-BIH	Massachusetts Institute of Technology and Beth Israel Hospital
TDA	Topological Data Analysis
UCR	University of California, Riverside
$H_0$	zero-dimensional homology
$k_{\max}$	$k$ -th largest value of a given set of real numbers
$\lambda_k$	$k$ -th persistence landscape level ( $k = 1, 2, \dots$ )
$w_k$	weight of $k$ -th landscape level assigned by a gating layer

## References

- [1] Naitzat, G., Zhitnikov, A., Lim, L.-H.: Topology of deep neural networks. *Journal of Machine Learning Research* **21**(184), 1–40 (2020)
- [2] Goldfarb, D.: Understanding deep neural networks using topological data analysis (2018) [arXiv:1811.00852](https://arxiv.org/abs/1811.00852) [cs.LG]
- [3] Ballester, R., Arnal, X., Casacuberta, C., Madadi, M., Corneanu, C.A., Escalera, S.: Predicting the generalization gap in neural networks using topological data analysis (2022) [arXiv:2203.12330](https://arxiv.org/abs/2203.12330) [cs.LG]
- [4] Corneanu, C.A., Escalera, S., Martinez, A.M.: Computing the testing error without a testing set. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2674–2682 (2020). <https://doi.org/10.1109/CVPR42600.2020.00275>
- [5] Bubenik, P.: Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research* **16**, 77–102 (2015)
- [6] Kim, K., Kim, J., Zaheer, M., Kim, J.S., Chazal, F., Wasserman, L.: PPlay: Efficient topological layer based on persistence landscapes. In: 34th Conference on Neural Information Processing Systems (2020)
- [7] Wheeler, M., Bouza, J., Bubenik, P.: Activation landscapes as a topological summary of neural network performance. In: 2021 IEEE International Conference on Big Data (2021). <https://doi.org/10.1109/bigdata52589.2021.9671368>
- [8] Ong, B.T., Sugiura, K., Zettsu, K.: Dynamically pre-trained deep recurrent neural networks using environmental monitoring data for predicting PM<sub>2.5</sub>. *Neural Computing and Applications* **27**, 1553–1566 (2016). <https://doi.org/10.1007/s00521-015-1955-3>

- [9] Namdari, A., Durrani, T.S.: A multilayer feedforward perceptron model in neural networks for predicting stock market short-term trends. *Operations Research Forum* **2**, 38 (2021). <https://doi.org/10.1007/s43069-021-00071-2>
- [10] Umeda, Y., Kaneko, J., Kikuchi, H.: Topological data analysis and its application to time-series data analysis. *Fujitsu Scientific and Technical Journal* **55**, 65–71 (2019)
- [11] Gidea, M., Katz, Y.: Topological data analysis of financial time series: Landscapes of crashes. *Physica A: Statistical Mechanics and its Applications* **491**, 820–834 (2018). <https://doi.org/10.1016/j.physa.2017.09.028>
- [12] Yan, Y., Ivanov, K., Cen, J., Liu, Q.-H., Wang, L.: Persistence landscape based topological data analysis for personalized arrhythmia. *Preprints* (2019). <https://doi.org/10.20944/preprints201908.0320.v1>
- [13] Carrière, M., Chazal, F., Ike, Y., Lacombe, T., Royer, M., Umeda, Y.: Perslay: A neural network layer for persistence diagrams and new graph topological signatures. In: *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 108, pp. 2786–2796. PMLR, Palermo (2020). <https://proceedings.mlr.press/v108/carriere20a.html>
- [14] Dindin, M., Umeda, Y., Chazal, F.: Topological data analysis for arrhythmia detection through modular neural networks. In: *Advances in Artificial Intelligence*, pp. 177–188. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-47358-7\\_17](https://doi.org/10.1007/978-3-030-47358-7_17)
- [15] Chen, C., Ni, X., Bai, Q., Wang, Y.: A topological regularizer for classifiers via persistent homology. In: *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 89, pp. 2573–2582. PMLR, Naha (2019). <https://proceedings.mlr.press/v89/chen19g.html>
- [16] Clough, J., Byrne, N., Oksuz, I., Zimmer, V., Schnabel, J., King, A.: A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **44**, 8766–8778 (2022). <https://doi.org/10.1109/TPAMI.2020.3013679>
- [17] Gabrielsson, R.B., Nelson, B.J., Dwaraknath, A., Skraba, P.: A topology layer for machine learning. In: *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, vol. 108, pp. 1553–1563. PMLR, Palermo (2020). <https://proceedings.mlr.press/v108/gabrielsson20a.html>

- [18] Gonzalez-Diaz, R., Gutiérrez-Naranjo, M.A., Paluzo-Hidalgo, E.: Topology-based representative datasets to reduce neural network training resources. *Neural Computing and Applications* **34**, 14397–14413 (2022). <https://doi.org/10.1007/s00521-022-07252-y>
- [19] Ferrà, A., Casacuberta, C., Pujol, O.: Reconstruction of univariate functions from directional persistence diagrams (2022) [arXiv:2203.01894](https://arxiv.org/abs/2203.01894) [math.AT]
- [20] Belton, R.L., et al.: Reconstructing embedded graphs from persistence diagrams. *Computational Geometry: Theory and Applications* **90**, 101658 (2020). <https://doi.org/10.1016/j.comgeo.2020.101658>
- [21] Fasy, B.T., Micka, S., Millman, D.L., Schenfisch, A., Williams, L.: The first algorithm for reconstructing simplicial complexes of arbitrary dimension from persistence diagrams (2022) [arXiv:1912.12759v4](https://arxiv.org/abs/1912.12759v4) [cs.CG]
- [22] Turner, K., Mukherjee, S., Boyer, D.M.: Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA* **3**(4), 310–344 (2014). <https://doi.org/10.1093/imaiai/iau011>
- [23] Dau, H.A., et al.: The UCR Time Series Classification Archive. [https://www.cs.ucr.edu/~eamonn/time\\_series\\_data\\_2018/](https://www.cs.ucr.edu/~eamonn/time_series_data_2018/) (2018)
- [24] Moody, G.B., Mark, R.G.: The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine* **20**, 3, 45–50 (2001). <https://doi.org/10.1109/51.932724>
- [25] Edelsbrunner, H., Harer, J.: Persistent homology – a survey. *Contemporary Mathematics* **453**, 1–2 (2008). <https://doi.org/10.1090/conm/453>
- [26] Barannikov, S.A.: The framed Morse complex and its invariants. *Advances in Soviet Mathematics* **21**, 93–115 (1994). <https://doi.org/10.1090/advsov/021/03>
- [27] Fazeli, S.: ECG Heartbeat Categorization Dataset. Segmented and Pre-processed ECG Signals for Heartbeat Classification. <https://www.kaggle.com/datasets/shayanfazeli/heartbeat>
- [28] Chazal, F., Oudot, S.: Towards persistence-based reconstruction in Euclidean spaces. *Proceedings of the Annual Symposium on Computational Geometry* (2008). <https://doi.org/10.1145/1377676.1377719>