

---

# EEG TIME SERIES CLASSIFICATION

## THE TOPOLOGICAL APPROACH

---

**Ivan Akinfiev**

Department of Mathematics

University of Arizona

Tucson

akinfiev@arizona.edu

### ABSTRACT

With the development of the Machine Learning framework, as well as the complexification of modern big data structures, the development of new methods is increasingly required. This article is an attempt to use Topological Data Analysis to classify Time Series using the Time Delay Embedding technique. If the classical one-dimensional time series is a homotopy trivial space, then Time Delay, with the desired configuration, will have a non-trivial topology. Later we will calculate persistent entropy, the kernel of which will be applied to the popular classifiers. In most datasets, topological information is not enough, but it is vectorizable, i.e. it can be used as additional information about the dataset. However, for the sake of this article, we will work on raw TDA data.

## 1 Introduction to TDA framework.

The main idea of Topological Data Analysis is that the data has a shape and understanding this shape will help us to describe the underlying process of how the data is generated. The method that attempts to catch the shape is coming from algebraic topology and known as persistence homology groups.

### 1.1 Background: Homological Algebra

Generally, topological space is a very "soft" mathematical object, since it doesn't contain a metric itself. Therefore, we need some algebraic tool that will be able to deal with this softness. The idea is to consider a family of maps of topological space to its versions in different dimensions. Consider the general chain complex,

$$\dots \rightarrow C_{n+1} \xrightarrow{\partial_{n+1}} C_n \xrightarrow{\partial_n} C_{n-1} \rightarrow \dots \rightarrow C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0 \quad (1)$$

where composition of any  $\partial_n \circ \partial_{n-1}$  is always 0.  $C_n$  are called chains and basically abelian groups that encode the topological information of given topological space  $X$  in dimension  $n$ . The crucial points is find their "interaction" between each other in neighbour dimensions. The way to calculate it is to compute the factor groups,

$$\mathbb{H}_n(X) = Ker \partial_n / Im \partial_{n+1} \quad (2)$$

The following construction is called the  $n$ -th homology group of topological space  $X$ . The most pleasurable aspect here is that this family of objects are topological invariants. It doesn't matter how "badly" we deform the original space, the homology groups will remain the same. Geometrically, the homology groups encode the amount of  $n$ -dimensional holes. We will demonstrate that later visually. The drawback is, for our purposes, this tool is very abstract and it seems a problem to apply this instrument in a computer. To deal with this problem, we introduce a more rigid approach.

## 1.2 Simplicial Homology

As we already pointed out, the introduced method is very soft for our purposes, so we need to introduce to more combinatorial approach that is computationally friendly. The point is to realise that some common types of topological spaces can be seen through the triangulation. For instance, torus  $T$ , real projective plane  $\mathbb{R}P^2$  and Klein bottle can be seen in the following way:

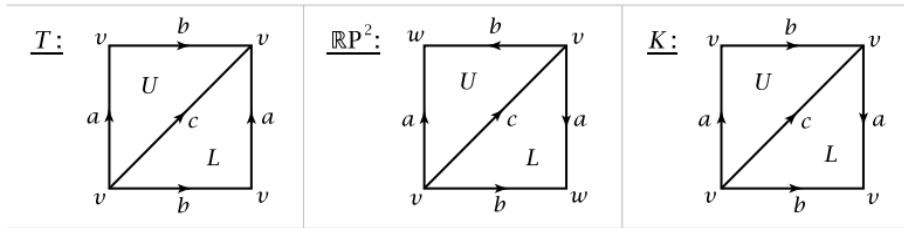


Figure 1: From left to right: the torus, projective plane and Klein bottle.

By identification of the opposite sides, we obtain their usual shape. For example, take the torus above, glue the opposite sides and imagine which shape will you get. Indeed, from the very abstract entities, we came to a very combinatorial approach: let's introduce its formalization. Here, we introduce a standard  $n$ -dimensional simplex

$$\Delta^n = \left\{ (t_0, t_1, \dots, t_n) \in \mathbb{R} \mid \sum_i t_i = 1, t_i \geq 0, \forall i \right\}, \quad (3)$$

which is basically generalization of 2-dimensional triangle.

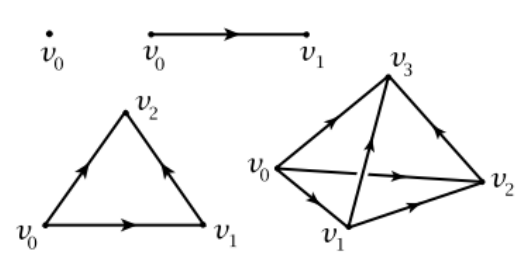


Figure 2: 0,1,2 and 3-dimensional simplex.

Computing homology groups, we basically "killing" the  $n$ -dimensional cycles with  $n+1$  dimensional solid cells corresponding to them. This is clearly seen through simplicial approach. Let's define the boundaries on  $n$ -dimensional simplices:

$$\partial[v_1 - v_0] = [v_1] - [v_0] \quad (4)$$

$$\partial[v_0, v_1, v_2] = [v_1, v_2] - [v_0, v_2] - [v_0, v_1] \quad (5)$$

$$\partial[v_0, v_1, v_2, v_3] = [v_1, v_2, v_3] - [v_0, v_2, v_3] + [v_0, v_1, v_3] - [v_0, v_1, v_2] \quad (6)$$

#### ADD DEFINITION OF SIMP COMPLEX

The signs are set according to the counterclockwise orientation of simplexes. Using this geometric intuition, we can define a boundary homomorphism  $\partial_n : \Delta_n(X) \rightarrow \Delta_{n-1}(X)$ . Set its values in basis of elements, we get the explicit formula,

$$\partial_n(\sigma_\alpha) = \sum_i (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_n] \quad (7)$$

It is easy to prove that,  $\Delta_n(X) \xrightarrow{\partial_n} \Delta_{n-1}(X) \xrightarrow{\partial_{n-1}} \Delta_{n-2}(X)$  is a null map. Therefore, we can apply tool of homological algebra and find homology groups on simplicial complexes and do it in a pretty combinatorial way. Indeed, we just gently shifted the problem to a field where computers can deal with it. To clarify the theory, let's consider an easy example:  $n$ -dimensional sphere  $\mathbb{S}^n$ . In order not to confuse a reader, let's not talk about  $H_0(\mathbb{S}^n)$  and discuss other dimensions.  $n$ -dimensional sphere doesn't have any holes except one with dimension  $n$ . Therefore, for  $\mathbb{S}^n$ ,  $H_{n \pm j}(\mathbb{S}^n) = 0$  for all  $j \neq n$ . However, since it has one  $n$ -dimensional home (empty inside),  $H_n(\mathbb{S}^n) = \mathbb{Z}$ . The dimension of the resulted homological group  $\dim H_n(X)$  will define the **amount** of holes in corresponding dimension and called Betti numbers. We will need them later. Now, as we got acquainted with topological tools, let's incorporate it into the world of data.

### 1.3 Data as a Simplicial Complex

In the field of Machine Learning generally a widely applicable model for the data is a metric space of finite dimension  $(X, \delta_X)$  with finite amount of points, equipped with Euclidian distance  $\delta_X$ . **The core idea is to see points in a dataset as a vertices of corresponding simplicial complex with edges drawn whenever open ball intersect each other** So, let's introduce it.

### 1.3.1 Definition

Let  $X \subset \mathbb{R}^n$  be a subspace with fixed  $\epsilon > 0$ . The *Cech Complex*  $C_\epsilon(X, \delta_X)$  is an abstract simplicial complex with points as vertices and  $k$ -simplex  $[v_0, v_1, \dots, v_n]$  whenever points  $\{v_0, v_1, \dots, v_n\} \subset X$  satisfies

$$\bigcap_i B_\epsilon(v_i) = \emptyset \quad (8)$$

Basically, the structure Cech Complex is purely related to the cover of space  $X$ . Using the cover of a space we can define the nerve of  $\mathbf{X}$ .

### 1.3.2 Definition

The *Nerve*  $N(\{U_i\})$  If  $X$  is a simplicial complex with vertices as corresponding sets of  $\{U_i\}$  and  $k$ -simplex  $[l_0, l_1, \dots, l_k]$  when the intersection of covering is non-empty.

Given these definitions we are ready to introduce the most important theorem in TDA.

### 1.3.3 Nerve Theorem

Let  $\{U_i\}$  be an open cover of topological space  $\mathbf{X}$ , such us all non-empty intersections

$$U_{i_1} \cap U_{i_2} \cap \dots \cap U_{i_k} \quad (9)$$

are contractible. Then, the Nerve is equivalent to  $\mathbf{X}$ .

The second theorem define a strong relationship between Cech Complexes and our dataset.

### 1.3.4 Theorem

Since  $\mathbf{X}$  is homotopy equivalent to  $N(\{U_i\})$  and obviously  $N(\{U_i\}) \cong C_\epsilon(\mathbf{X}, \partial_X)$ ,  $\mathbf{X}$  is homotopy equivalent to  $C_\epsilon(\mathbf{X}, \partial_X)$ .

Since we established a strong connection between data and Cech complex, there is a clear reason to explore the given complex directly. Note: there are also another types of complexes like Vietoris-Rips complex or Alpha complex. They are very similar except for the dimension flexibility. By triangulating  $X$  the Vietoris-Rips complex can establish simplices with dimension even more than  $n$ .

The whole idea is pretty simple: let's take two points in  $\mathbb{R}^2$  and draw a circles with centers in these points. If the balls intersect, we draw an edge between them. All these balls generates big simplicial complex that could be explored within the topological framework.

## 1.4 Persistent Homology

We already familiar with a tool that helped us to find  $n$ -dimensional holes in topological space. Now, we need to set this tool to serve us for the computational purposes.

However, if we want to triangulate the dataset with balls with radius  $r$ , how do we choose  $r$ ? And this question is almost impossible to answer directly. If we live in 100 dimensions, there is almost impossible to choose the right  $r$ . However, the people comes up with the following idea:

#### 1.4.1 Filtration

Generally, filtration defines an order on the subsets of objects of some mathematical structure. For example, let's take the set  $\{1, 2, 3\}$  and give it index 3. The set  $\{1, 2\}$  will be a subset of the original set, and, therefore, will have the lower index 2. The set  $\{1\}$  will have index 1. As we can see, we defined the order on the subsets of  $\{1, 2, 3\}$ . We will apply this idea for the sake of TDA.

#### 1.4.2 Definition

Filtration of a simplicial complex  $C(\mathbf{X})$  a nested collection of subcomplexes  $C_r(\mathbf{X})_{r \in T}$ , where  $T \subset \mathbb{R}$ , such that for any  $r, r' \in T$  if  $r \leq r'$ , then  $C_r(\mathbf{X}) \leq C_{r'}(\mathbf{X})$ . Overall,  $C(\mathbf{X}) = \cup_{r \in T} C_r(\mathbf{X})$

Filtering will consist in the fact that we will gradually increase the radius of our balls. The balls, as the radius increases, will cross more and more of their neighbors, thereby determining the filtration of the complex. To build intuition, I highly recommend to associate radius with time. This is clearly demonstrated in the picture below.

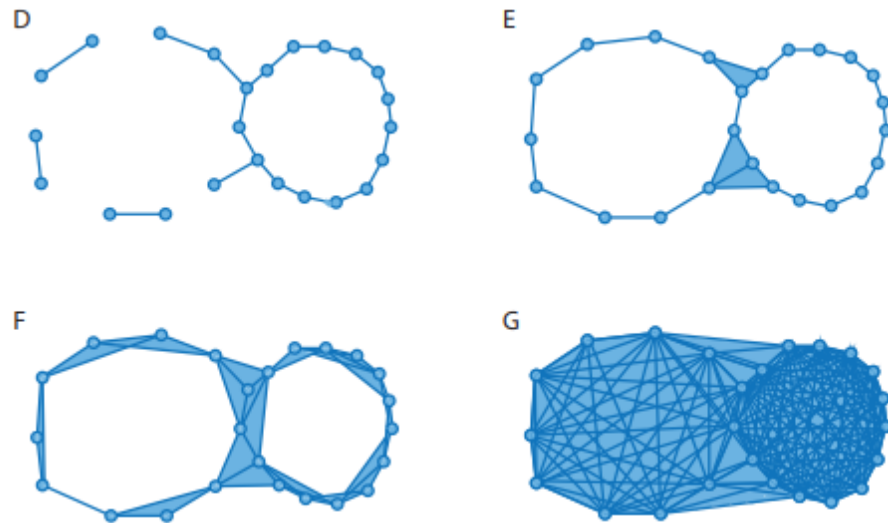


Figure 3: Filtration, as we increase radius  $r$ . On the pictures (E) and (R), we clearly see two 1-dimensional holes.

As we defined filtration, our computer will try to compute persistent homology of  $C(\mathbf{X})$ .

## 1.5 Persistent Modules

*Remark:* To avoid a sufficiently large mathematical description of the structure of modules, consider that the module as just a vector space over an arbitrary ring.

After computation of persistent homologies, we need a way to encode and visualize our topological information. It will rely on the fact that as we commit filtration, some homologies "die" immediately, and some live for a sufficient amount of time, as the radius increases. The amount of time will be served as an information that we have found an emptiness in our space. To do this, we need to introduce the concept of a persistent module.

### 1.5.1 Definition

*Persistent Module*  $\mathbb{V}$  over  $T \in \mathbb{R}$  is an indexed family of vector spaces  $(V_r | r \in T)$  and a doubly indexed family of linear maps  $(v_s^r : V_r \rightarrow V_s | r \leq s)$  in which the following composition is  $v_t^s \circ v_s^r = v_t^r$  fulfilled whenever  $r \leq s \leq t$  and where  $v_r^r$  is an identity map on  $V_r$ .

For instance, during the filtration, we have computed some persistent homology groups and got the following sequence

$$\dots \rightarrow 0 \rightarrow \dots, \rightarrow \mathbb{Z} \rightarrow \dots, \rightarrow \mathbb{Z} \rightarrow \mathbb{Z} \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow \dots \rightarrow 0 \rightarrow \mathbb{Z} \rightarrow 0 \rightarrow \dots, \quad (10)$$

Denoting  $q$  as an infimum of the interval of non-zero maps, we can collect all the  $q_i$  from the sequence. The following set will form a persistence barcode of  $\mathbb{V}$ . The illustration is below.

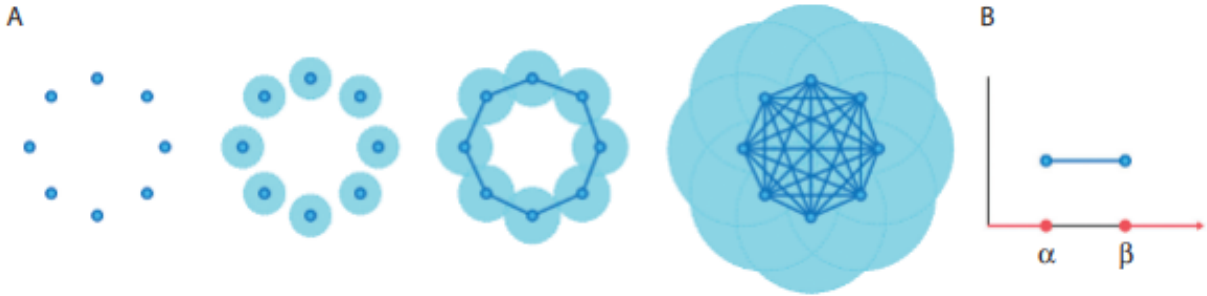


Figure 4: As algorithm detected the 1-dimensional homology, it proceeds further while memorising time when the hole existed. The number  $\beta - \alpha$  will be an element of barcode of  $\mathbb{V}$

## 2 Example of Topological Pipeline

Before we start work with the data, we will how TDA works in practice. We will calculate homology 3-genus surface and show other techniques of topological inference. Let's first use analytical approach.

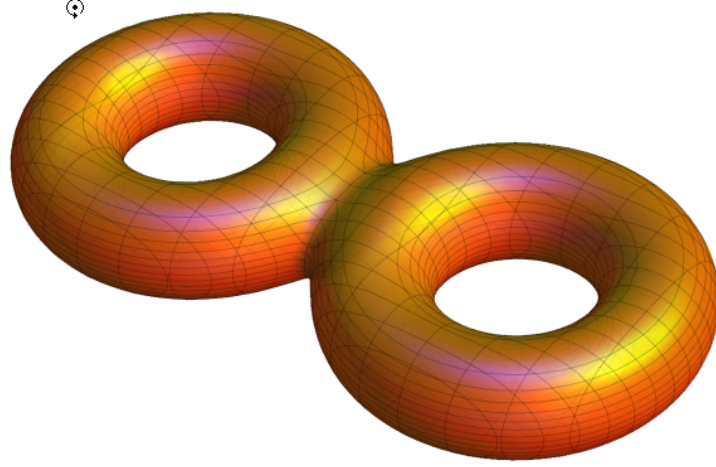


Figure 5: Genus-2 surface

It is obvious that  $H_2(X) = \mathbb{Z}$  and  $H_0(X) = \mathbb{Z}$ . What about  $H_1(X)$ ? Let's first compute the first homology of  $\mathbb{T}^2$  torus. To begin with, we notice that torus itself is defined as a product of two one-dimensional spheres  $\mathbb{S}^1 \times \mathbb{S}^1$ . We know that  $H_1(\mathbb{S}^1) = \mathbb{Z}$ , so how we calculate a product? A fancy way is to use Kunneth formula:

$$H_n(X \times Y) = \bigoplus_{s+k=n} H_s(X) \otimes H_k(Y) \quad (11)$$

$$H_1(\mathbb{S}^1) = \mathbb{Z}$$

$$H_0(\mathbb{S}^2) = \mathbb{Z}$$

Using Kunneth formula,

$$H_0(\mathbb{S}^2) \otimes H_1(\mathbb{S}^2) = \mathbb{Z} \otimes_{\mathbb{Z}} \mathbb{Z} = \mathbb{Z}$$

$$H_1(\mathbb{S}^2) \otimes H_0(\mathbb{S}^2) = \mathbb{Z} \otimes_{\mathbb{Z}} \mathbb{Z} = \mathbb{Z}$$

So,

$$H_n(\mathbb{S}^1 \times \mathbb{S}^1) = \mathbb{Z} \oplus \mathbb{Z} = \mathbb{Z}^2$$

(12)

This is precisely the  $H_1(\mathbb{T}^2)$ . After, we realise that genus-2 surface  $S_2$  is just a connected sum of two toruses.

$$S_2 = \mathbb{T} \# \mathbb{T} \quad (13)$$

Notice that, for any  $X$  and  $Y$ ,  $X \# Y$  are  $X \vee Y$  is we collapse  $\mathbb{S}^{n-1}$  into a point. By Mayer-Vietoris argument, we know that  $H_n(X \vee Y) \cong H_n(X) \oplus H_n(Y)$  We can construct exact sequence

$$\cdots \rightarrow H_i(S^{n-1}) \rightarrow H_i(M \# N) \rightarrow H_i(M \vee N) \rightarrow \cdots$$

. Since  $k$ -dimensional sphere doesn't have any non-trivial homologies except  $H_k$ , we get that  $H_i(M \# N) \cong H_i(M \vee N) \cong H_i(M) \oplus H_i(N)$  for  $i \neq n-1, n$ . Let's now consider a sequence,

$$0 \rightarrow H_n(M \# N) \rightarrow H_n(M \vee N) \rightarrow H_{n-1}(S^{n-1}) \rightarrow H_{n-1}(M \# N) \rightarrow H_{n-1}(M \vee N) \rightarrow 0$$

We have,

$$0 \rightarrow \mathbb{Z} \rightarrow \mathbb{Z} \oplus \mathbb{Z} \rightarrow \mathbb{Z} \rightarrow H_{n-1}(M \# N) \rightarrow H_{n-1}(M \vee N) \rightarrow 0$$

.

We clearly see, that in this case  $H_i(M \# N) \cong H_i(M \vee N) \cong H_i(M) \oplus H_i(N)$ . Therefore, the  $H_1(S_2) = H_1(\mathbb{T} \# \mathbb{T} \# \mathbb{T}) = H_1(\mathbb{T}^2) \oplus H_1(\mathbb{T}^2) = \mathbb{Z}^4$  The corresponding betti numbers are  $\beta_0 = 1, \beta_1 = 6, \beta_2 = 1$ . We finally found the holes analytically, let's apply methods of TDA and try to resolve it computationally.

First, let's distribute points uniformly on the surface. For this, we need to find a formula for genus-3. It is computed in the following way - first, take polynomial,

$$f(x) = \prod_{k=1}^3 (x - (k-1))(x - k) = x(x-1)^2(x-2) \quad (14)$$

Then,

$$g(x, y) = f(x) + y^2 \quad (15)$$

$$h(x, y, z) = g(x, y)^2 + z^2 - r^2 = (x(x-1)^2(x-2) + y^2)^2 + z^2 - r^2 \quad (16)$$

Let's compute the surface and generate uniformly distributed points on it.



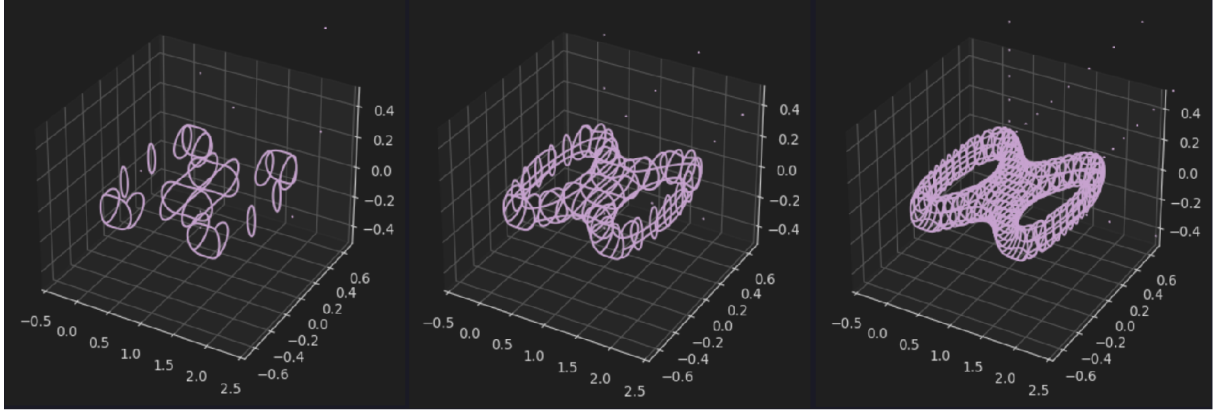


Figure 6: Surface with resolution  $n = 100, 400, 1000$ . From left to right.

As we see, the derived equation is indeed resembles the original genus-2 torus. Let's distribute points on the surface with noise  $n = 0.0001$ .

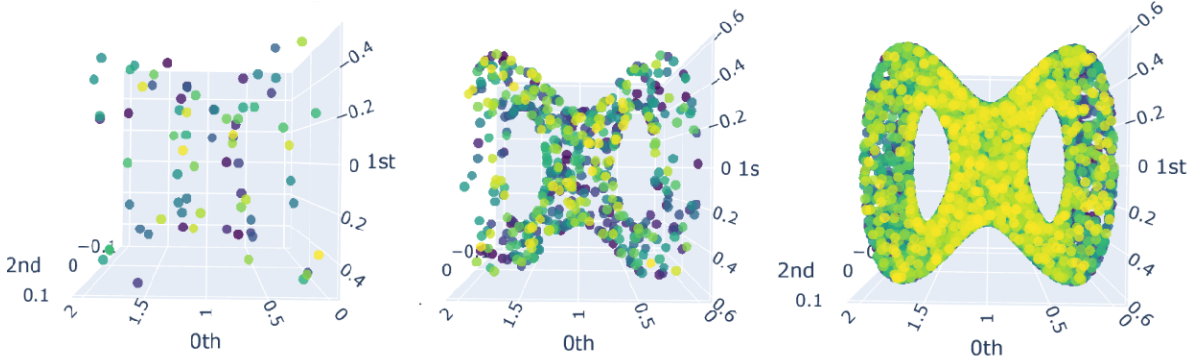


Figure 7: Uniformly distributed points on 2-genus surface.  $n = 100, 400, 2000$ . From left to right.

We will run a standard TDA pipeline using Alpha Complex triangulation. When the filtration done and homologies are computed, we obtain the following persistent diagram and barcode.

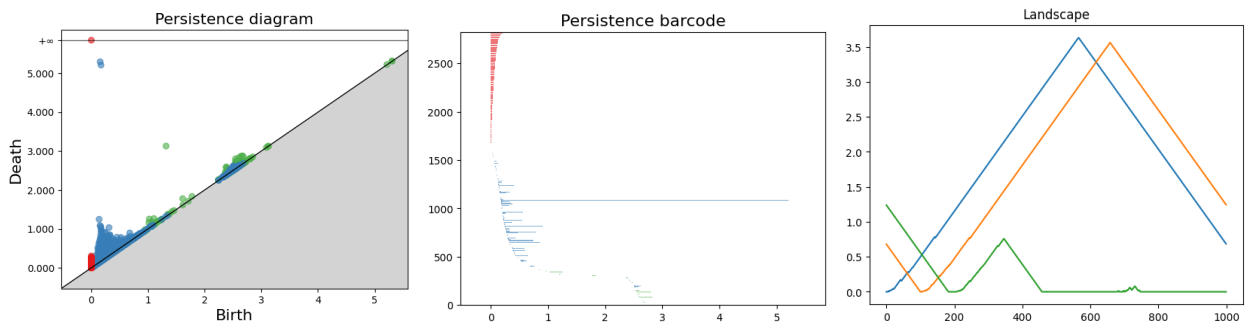


Figure 8: The analysis of genus-2 torus. From left to right. Persistent diagram, barcode and landscape.

From the figure we clearly see that we caught some invariants. There is a 2-dimensional void, and two 1-dimensional holes. Where are other 2? We see that other two are also seen, but they are very small. This is due the fact that the natural parametrization of genus 2 has almost no space at the end of the figure, while it is easy to catch holes that are outside of the figure. Having this example in mind, let's apply this tool to classify ECG Time Series.

### 3 Classification of ECG Time Series.

The pipeline proposed for consideration can be presented as follows. We will take the data from open access. They will be a collection of multidimensional time series. We will take a one-dimensional version, use Taker Delay Embedding, triangulate the nested points, then calculate homology and output the barcode entropy. The obtained topological two-dimensional data will train the Support Vector Machine and we will be able to classify the series using only topological information about them. The usual pipeline is the following:

$$\left\{ \begin{array}{c} \text{Time} \\ \text{Series} \end{array} \right\} \rightarrow \left\{ \begin{array}{c} \text{Alpha} \\ \text{Complex} \end{array} \right\} \rightarrow \left\{ \begin{array}{c} \text{Persistence} \\ \text{Homologies} \end{array} \right\} \rightarrow \left\{ \begin{array}{c} \text{Persistence} \\ \text{Entropy} \end{array} \right\} \rightarrow \left\{ \begin{array}{c} \text{SVM} \\ \text{with RBK} \end{array} \right\}$$

#### 3.1 Data

This database consists of 45,152 patient ECGs. The number of volts per A/D beats 4.88, and A/D converter hd 32-bit resolution. The amplitude unit was microvolt. The upper limit was 32,767, and the lower limit was 32,768. The institutional review board of Shaoxing People's Hospital and Ningbo First Hospital approved this study, granted the waiver application to obtain informed consent, and allowed the data to be shared publicly after de-identification. The data was encoded into HEA files, which required special processing, in particular the use of the wfdb python library. Each patient had 17 areas where cardio data was taken. This means that the time series of each patient has a dimension of 17. Considering that this would greatly complicate data analysis, as well as the fact that the readings of the series are strongly correlated, it was decided to compress the time series into a one-dimensional one.

#### 3.2 Taken's Time Delay Embedding

In the first approximation, the task of investigating series topologically seems frankly naive. From the view of topology, an ordinary one-dimensional time series is homeomorphic to an open segment, which means it is topologically trivial. However, we will change the "appearance" of the series in the following way: as we go through the series in time, we will take the neighboring points of the series and represent them as the coordinates of the embedding space.

$$TD_{d,\tau}s : \mathbb{R} \rightarrow \mathbb{R}^d, t \rightarrow \begin{bmatrix} s(t) \\ s(t + \tau) \\ s(t + 2\tau) \\ s(t + 3\tau) \\ \dots \\ s(t + (d - 1)\tau) \end{bmatrix} \quad (17)$$

For example, let's consider cosine function and take just one neighbour point.

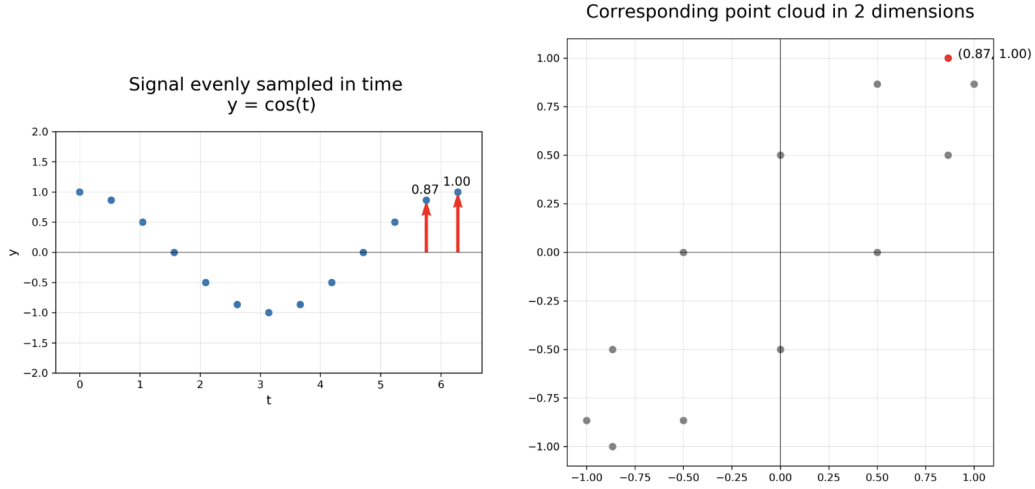


Figure 9: 2-dimensional embedding of the cosine function

The idea is if we have a patterns in a series, Taken's embedding will give us topologically nontrivial space. As we see in the Figure 9, 2-dimensional embedding of cosine function gives us an ellipsis, which has non-trivial one-dimensional homology group.

### 3.3 Persistence Entropy

In order to get topological information for classification, we need to understand which method to choose. The most suitable method for classification needs is persistent entropy, which is simply the codification of persistent barcode. More formally persistent entropy can be defined in the following way. Given persistence diagram  $D = \{(b_i, d_i)\}_{i \in I}$ , the persistence entropy is,

$$E(D) = \sum_{i \in I} p_i \log(p_i) \quad (18)$$

where,

$$p_i = \frac{(b_i - d_i)}{\sum_{i \in I} (b_i - d_i)} \quad (19)$$

The intuition is completely the same as we would take an entropy of some distribution. But instead of this, we take an entropy of barcode. It turned out that this technique can serve us a great encoding tool for machine learning, especially SVM.

### 3.4 Analysis

Let's demonstrate this technique on some time series from the dataset. We will take one series from healthy patient and one with a heart attack.

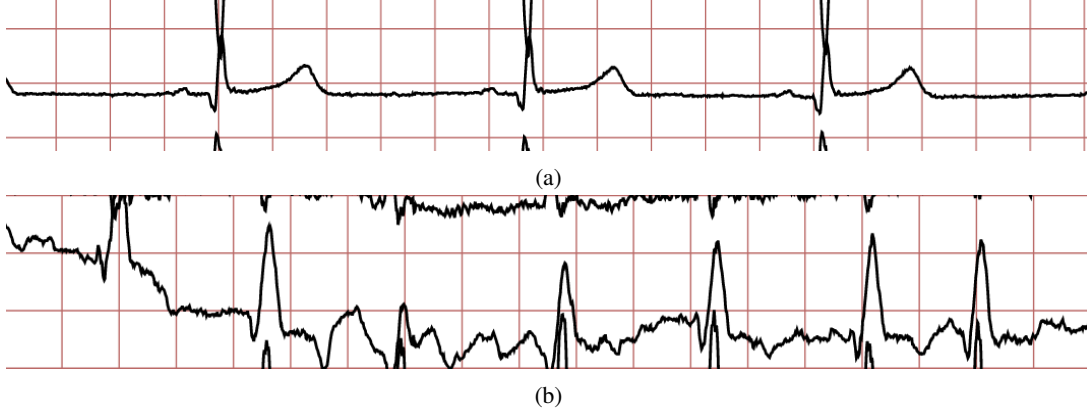


Figure 10: (a) aVL series taken from a healthy patient. (b) The same aVL probe taken from a patient with a heart attack and arrhythmia.

In Illustration 9, we used an embedding in space with dimension 2. However, one neighboring point may not be enough. To increase the accuracy and completeness of the information, we need to capture more neighboring points. However, with each added point, we increase the dimension by 1, thereby blocking the possibilities for effective data visualization. Therefore, after embedding in a space of dimension 10b, we will apply PCA and regain dimension 3. Here is the series described above with the Taken's procedure.

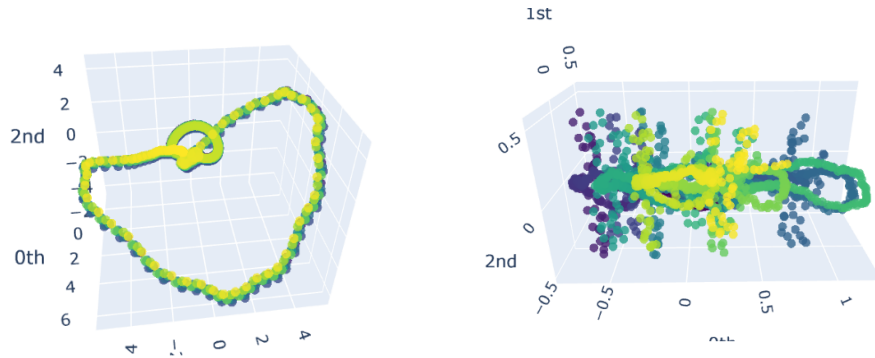


Figure 11: Embedded Time Series. On the left: healthy person. On the right: a person with Myocardial infarction

Next, we compute simplitial homologies and visualise the result using persistent diagrams.

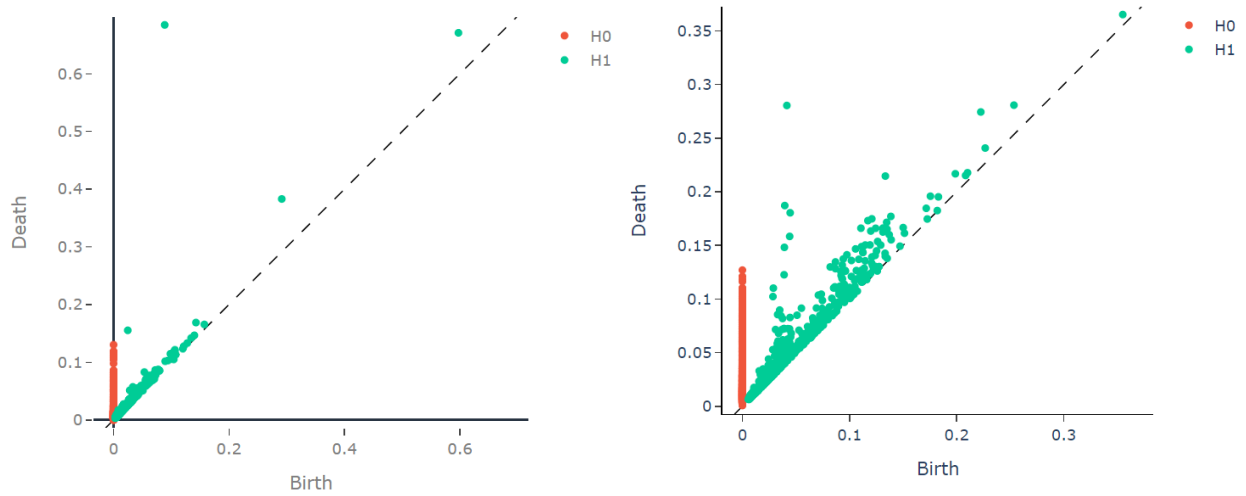


Figure 12: Embedded Time Series. On the left: healthy person. On the right: a person with Myocardial infarction

We see that the first the first persistent diagram has clearly recognised the 1-dimensional hole and a couple of small ones. On the other hand, the second diagram has confirmed the fact of complete mess with the guy with infarction.

### 3.5 Results

After calculating all the topological information we obtain a dataset of persistent entropy. Important to note that persistence entropy is just a 2-dimensional vector. Therefore, the resulting feature space will be also 2-dimensional. Here is a non-normalized, truncated and normalized visualization of the collection of all persistence entropy for all the observations.

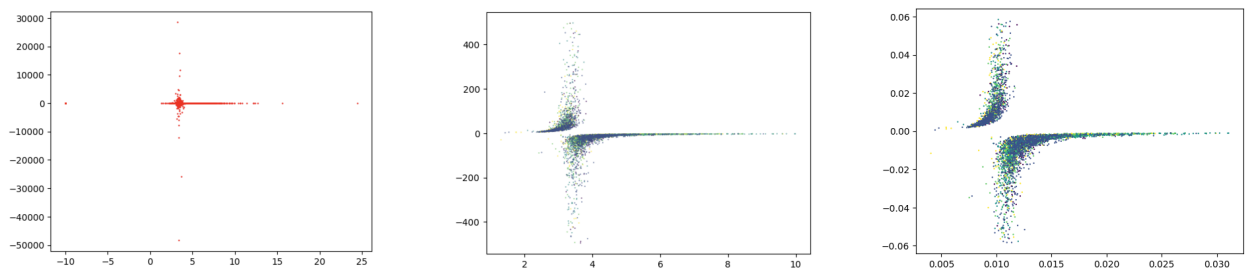


Figure 13: From left to right. Raw topological Features. We can see enormous outliers. In the center we see have truncated outliers and got more recognisable picture. On the right the data is normalized.

From the first perspective it seems that the outliers would affect almost any ML model in a negative way. Indeed, there are a couple of points that are  $>30000$  times bigger than the mean of  $y$ . Here, we tested different models on three structures of data described above. We used a train/test 80/20 and  $CV-k = 5$ .

Accuracy on Test Data			
Model	Raw Data	Truncated Data	Normalized Data
K-means	0.45125	0.44212	0.48395
SVM with Gauss	0.32683	0.34225	0.33248
SVM with RBK	0.69347	0.64122	0.71265
Random Forest	0.43550	0.42094	0.40198
Boosting	0.45917	0.37124	0.51326
Bagging	0.47312	0.35352	0.49646
Feedforward NN 2x8+1x4	0.02030	0.04731	0.26151

The overall picture suggests that, for the most part, SVM with RBK that showed 0.7 accuracy on only two features is capable of learning from topological data, while most of the other models do not even reach 0.5 accuracy, which is not acceptable when analyzing ECG. Taking into account the availability of the amount of data  $n > 6000$ , we used regularization methods that excluded the possibility of overfitting the data. Interestingly, Feedforward neural network with 2 hidden layers with Adam showed almost zero accuracy on raw data. The possible explanation is that generally, most of the neural networks are trained on the normalized data, here, in particular, the outliers blew up the first hidden layer and it blocked any opportunity of network to learn further with possible explosion of gradients.

## 4 Discussion

The best result was achieved using a Support Vector Machine with a Radial Basis Kernel with 0.7 accuracy, which is relatively acceptable for a dataset with two features. However, the effectiveness of this approach should be noted. In the modern world, recurrent and convolutional neural networks are best able to cope with such tasks, which show more effective results with proper training. We, in particular, launched a regular CNN with 3 convolutional layers without the use of topological transformations. It showed 0.847 accuracy, which leaves it desirable to refine the topological approach. One option is to apply persistent entropy transformation along with others and mix the resulting vectorizations, which will probably open the door to improving accuracy readings, as well as other metrics. With the current results, it is strictly forbidden to use our model to predict the electrocardiogram on real tasks.

## 5 Conclusion

In this article, an attempt was made to classify time series using topological data analysis. We started with the theoretical aspects of TDA and introduced theoretical definitions and concepts (e.g., chain complexes, persistent homology, barcodes, persistent diagrams, persistent landscape, etc.). For the application section, we built a classification of time series using Taken's Time Delay embedding. This is where the topological data analysis shows its quality in detection of shifted patterns and crisis of heart functionality. The results on certain types of classifiers, such as SVN, show that this can be a popular directions of research in the intersection of TDA and time series classification.