2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) | 979-8-3503-0249-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPRW59228.2023.00067

TopFusion: Using Topological Feature Space for Fusion and Imputation in Multi-Modal Data

Audun Myers[1], Henry Kvinge[1,2], and Tegan Emerson[1,3,4]

[1]Pacific Northwest National Laboratory, Mathematics of Data Science
[2]University of Washington, Department of Mathematics
[3]Colorado State University, Department of Mathematics
[4]University of Texas El Paso, Department of Mathematical Sciences

Abstract

We present a novel multi-modal data fusion technique using topological features. The method, TopFusion, leverages the flexibility of topological data analysis tools (namely persistent homology and persistence images) to map multi-modal datasets into a common feature space by forming a new multi-channel persistence image. Each channel in the image is representative of a view of the data from a modality-dependent filtration. We demonstrate that the topological perspective we take allows for more effective data reconstruction, i.e. imputation. In particular, by performing imputation in topological feature space we are able to outperform the same imputation techniques applied to raw data or alternatively derived features. We show that TopFusion representations can be used as input to downstream deep learning-based computer vision models and doing so achieves comparable performance to other fusion methods for classification on two multi-modal datasets.

1. Introduction

Data comes in many forms and modalities including time series, images, and text. The field of data science has seen continued improvements and new state-of-the-art performance for each of these modalities individually for several tasks. The problem of how to best handle heterogeneous data sets including multiple modalities or collections of related measurements, however, remains open. The goal of data fusion is to fuse multi-modal datasets together for automated or manual decision making [14]. As such, the challenge of data fusion could be framed as trying to represent multiple data sources into a single space that can be analyzed with, for example, machine learning models.

In this work we present a novel data fusion method which uses tools from topological data analysis (TDA) [9, 33] to embed multiple modalities into a common feature space independent of the data type. We leverage an image-based representation of topological features, persistence images [1] (PIs), to take advantage of machine learning's current strength in computer vision problems. There are now a wide variety of persistent homology filtrations, each appropriate to certain modalities. This means that our approach is applicable in a number of scenarios. By utilizing a variety of filtration techniques, the same core homological features (with different interpretations) are extracted from disparate data sources thereby providing a natural map into a common space. When presented with multiple data types or sources, we create a "multi-channel" PI by stacking the collection of PIs derived across the different filtrations. We then apply deep learning-based computer vision models to these PI stacks.

We investigate the performance of our topological fusion technique, TopFusion, for classification and robustness to incomplete data. TopFusion is tested on two multi-modal datasets; Wearable Stress Affect Detection (WESAD) [29] and Audio Visual MNIST (AV-MNIST) [32]. On both datasets, we demonstrate comparable classification accuracies to state-of-the-art approaches when trained and tested on complete data (e.g. all modalities or sensors are present for test data). The strong discriminatory power of the topological features for AV-MNIST is in part supported by a novel image scanning filtration approach presented in Section 2.3.3 based on zigzag persistence. We further evaluate the utility of TopFusion when presented with incomplete data. We consider both the setting where one modality is absent in the validation experiments and where one sensor (in a collection of sensors) fails to collect measurements during validation. We show that by applying existing data imputation techniques [6] in a topological feature space, rather than in the raw data space, one sees stronger performance recovery suggesting that TopFusion captures more strongly discriminative, robust features.

In summary, our contributions in this paper include:

- We propose a new framework, TopFusion, for multi-modal learning which maps a common feature space based on persistent homology features.
- We show that computer vision models trained on the output from TopFusion achieve competitive performance with equivalent models trained with other fusion techniques.
- Our tests of robustness to missing data show that by utilizing a common topological space, models which use TopFusion output are more robust to missing data.

2. Background

2.1. Data Fusion

There are three types of data fusion: Early, Late, and Intermediate [19]. Early data fusion is the process of fusing data sources before doing analysis. For example, [17] proposes two possible approaches for early fusion technique. The first approach is combining data by removing the correlation between two sensors. The second approach is to fuse data at its lower dimensional common space. A major drawback of early data fusion lies in the difficulty of combining multiple data sources with significant differences such as continuous or discrete. Late data fusion, where an individual model is used for each modality and their outputs are analyzed or used for input to downstream machine learning pipelines. A problem with this approach is the unavoidable complexity of first choosing models for each data source and then stitching the outputs of these models with the input for the last model. The third type of fusion, intermediate, is a mix of both early and late data fusion, where some data modalities are fused earlier and fed into a model and others at a later point in the pipeline. This allows for more flexibility and the fusion procedure; however, again the complexity of the pipeline has only increased. In this work we use early data fusion with a preprocessing step that brings the modalities to a common (topological) feature space independent of the data type.

2.2. Missing Data Imputation

Missing data is a challenge commonly encountered in the multi-modal setting. For example, the sensor associated with a particular modality could go out leaving us with only a partial measurement. In such a setting we need to estimate the missing data in our feature vector space for input to our machine learning model. One way to do this is to enter estimates (impute) the missing data based on the data from other sensors as well as previous data. There are three main cases of missing data presented in [20]; in this work we use the most general case since we are artificially removing data with no correlation to the other modalities.

There are many methods for data imputation ranging from simply replacing the missing values with a constant to deep learning methods [15]. In this work we use the common supervised learning method of k-Nearest Neighbors (kNN), which has been shown to often outperform even the complex deep neural network methods as well as being relatively computationally efficient [10, 15].

2.3. Topological Data Analysis

The field of Topological Data Analysis (TDA) studies the shape of data to gain understanding of the underlying system. In this work we leverage two tools from TDA: persistent homology and zigzag persistence. Both of these tools are filtrations that study the changing simplicial homology of abstract simplicial complexes as representations of a data source. The output of these filtrations is a persistence diagram D_p that captures the topology of the data in dimension p. A more thorough background on TDA, and persistent homology specifically, can be found in [8, 21, 27]. We leverage several commonly used techniques which are presented conceptually in the body of the paper and refers the reader to appendices (found in the supplemental material) for more technical details. The method presented for applying zigzag persistence to images is novel and therefor we include more details in the main paper.

2.3.1 Sublevel Set Filtration on Functions

In this work one of the filtration frameworks we leverage is called sublevel set filtration. As the name implies, this approach extracts topological features based on the level sets of a function assuming the function satisfies some general and standard conditions [4]. Under these assumptions, the relationships between extrema can be described in terms of the first two homologies, e.g. connected components and holes. The appendix (see supplemental material) provides an example of how we map the sublevel sets of a function of a single variable to a persistence diagram. By considering time series and images as functions of one or two variables, respectively, we can use sublevel set filtration to derive topological features from those modalities. More details and formalization of the sublevel set filtration techniques used can be found in Appendix (see supplemental material).

2.3.2 Vietoris-Rips Filtration on Time Series Data

For time series data where there is a reasonable assumption of an underlying dynamical system, albeit potentially unknown, there is an alternative way of capturing topological features. To employ this alternative filtration, the one-dimensional time series is first converted to a point cloud using a technique referred to as a time-delayed embedding [26] rooted in the rich theory of Takens' theorem [30]. After converting your time series to a point cloud, topological features can be extracted using the Vietoris-Rips (VR) filtration. The VR filtration is a prominent tool in persistent homology where a nested sequence of simplicial complexes is formed based on a similarity parameter [7,33]. In the case of point-cloud data, the similarity parameter that the filtration is based on is typically distance. Again, we provide a more rigorous presentation of the VR filtration in Appendix (see supplemental material).

2.3.3 Zigzag Persistence on Image Data

This section provides an introduction to zigzag persistent homology [3] and how it is a generalization of persistent homology. We also develop a method for applying zigzag persistence to study image data through directional scans.

Persistent homology is limited by its requirement that each Abstract Simplicial Complex (ASC) is a subset of the previous ASC to form the persistence module. This means at each step we can not remove simplices in the sequence of ASCs and can only add new simplices. There are many real-world applications of TDA where we have a parameterized sequence of ASCs with simplices both entering and exiting the ASC throughout the sequence making the use of persistent homology not possible. To solve this issue zigzag persistence [3] was developed. Zigzag persistence allows for an arbitrary subset directions in the ASC sequence where there is no inclusion direction rule:

$$K_0 \leftrightarrow K_1 \leftrightarrow K_2 \leftrightarrow \ldots \leftrightarrow K_\ell, \tag{1}$$

where $\leftrightarrow$ denotes one of the two inclusion maps: $\hookrightarrow$ or $\hookleftarrow$. A special case of Eq. (1) is when the left and right inclusions alternate or zigzag. We artificially construction this special zigzag case by interweaving the original ASCs with either unions or intersections of adjacent ASCs, which forms the ASCs

$$K_0 \hookrightarrow K_{0,1} \hookleftarrow K_1 \hookrightarrow \ldots \hookrightarrow K_{\ell-1,\ell} \hookleftarrow K_\ell. \tag{2}$$

for unions with $K_{i,i+1} = K_i \cup K_{i+1}$ or

$$K_0 \hookleftarrow K_{0,1} \hookrightarrow K_1 \hookleftarrow \ldots \hookleftarrow K_{\ell-1,\ell} \hookrightarrow K_\ell \tag{3}$$

for intersections with $K_{i,i+1} = K_i \cap K_{i+1}$.

The inclusion maps between simplicial complexes are extended to linear maps between homology groups resulting in the zigzag persistence module tracking the changing homology of Eq. (2) or (3) just as was the case for standard persistent homology. For the case of the union in Eq. (2), the zigzag persistent homology module is

$$H_p(K_0) \hookrightarrow H_p(K_{0,1}) \hookleftarrow \ldots \hookrightarrow H_p(K_{\ell-1,\ell}) \hookleftarrow H_p(K_\ell). \tag{4}$$

By leveraging the same algebra as is used in standard persistence homology we track persistence pairs where (when) homology features are born and die based on the zigzag persistence module with some loss on the intuition as our parameterization function has changed. Specifically, we again track the persistent homology using a persistence diagram D_p consisting of half-open intervals (persistence pairs) $[b,d)$; however, we use the indices of the ASCs as the birth and death times in substitution of the filtration parameter. For example, if there is one-dimensional homology (i.e., a loop) that appears at K_2 and persists until it disappears at K_3, we represent this as the persistence pair [2,3). In the case of a class appearing or disappearing at the union (or intersection) complex $K_{i,i+1}$, we use the half index pair $i, i+1$. If a topological feature persists in the last ASC in the zigzag persistence module we set its death past the last index with the pair $\ell, \ell+1$, where ℓ is the number of ASCs (without interwoven unions or intersections).

In many works zigzag persistence is used for studying temporal changes in data [23, 24, 31]; however, here we leverage zigzag persistence for "scanning" single-channel images. For this method we slide a window of width w along the image in a direction.

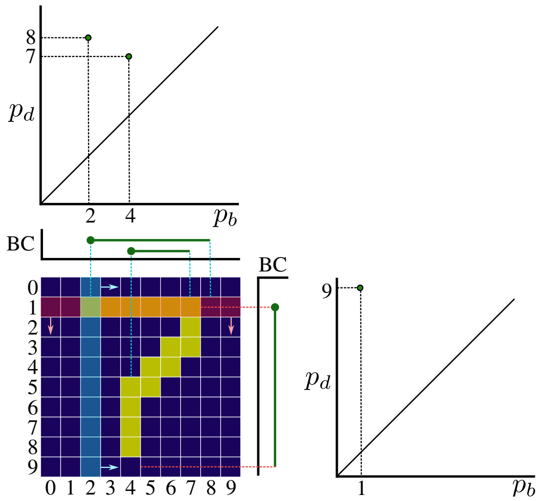


Figure 1. Zigzag persistence example on a 10x10 image of the number 7 in both the vertical and horizontal direction.

For each window W_i we can create an ASC K_i by taking a node set P as the pixel indices that have a value greater than a threshold s. For the toy example the image in binary and as such we set the threshold as $s = 0.5$. For images with multiple color channels one could apply these scans for each color separately or look at the grey scale image. For each pair of pixels $p_j, p_k \in P$ we add 1-simplices to the ASC K_i if $k - j = 1$ (i.e., if the two pixels are adjacent). We do not add any higher dimensional simplices as we are only interested zero-dimensional homology. Applying this procedure to the windows as they are slid across the image results in a sequence of ASCs which we apply unions to to create the zigzag persistence module. Calculating the persistence pairs results in persistence diagrams D_0 for each direction which summarize the “shape" of the image.

Here we provide an example demonstrating how to perform zigzag persistence to scan single image data. Specifically, the example we use is a 10×10 pixel image of the number 7 shown in Fig. 1.

In Fig. 1 we do scans in both the vertical and horizontal directions and use a window width of 1 pixel. However, we could do many more window widths and directions by rotating the image.

In our example in Fig. 1 in the vertical direction (from top down) at pixel $p_b = 1$ a single component is born. This component persists until reaching the bottom of the image at $p_d = 9$. We summarize this as the persistence pair $[1,9)$.

In the horizontal direction from left to right we begin at pixel $p_b = 2$ at the top of the seven where our first component is born. The second separate component is born at $p'_b = 4$ at the bottom of the seven. These two components combine at $p'_d = 7$ resulting in the persistence pair $[4,7)$ following the Elder Rule. At the next pixel $p_d = 8$ the only component left also dies resulting in the persistence pair $[2,8)$.

This example demonstrates how zigzag persistence captures information about then shape of the level set of the image that is not captured by other persistent homology filtrations. For example, sublevel set persistence would only show a single component in dimension 0.

2.3.4 Persistence Images

Persistence images [1] are a stable method for vectorizing the topological features summarized in persistence diagram for applications. The vectorization of topological features has been an active area of research to accommodate the integration of these features into machine learning frameworks. While there are many alternative representations, we have chosen PIs because they naturally integrate into conventional neural network architectures. At high level, PIs are formed by overlaying distributions centered on each point in a PD, summing those distributions to form a surface, and then discretizing the surface to form an image. Greater detail, and the parameter selection paradigm used, are provided in Appendix (see supplemental material).

3. Topological Data Fusion Using TopFusion

Our method, *TopFusion*, involves moving all the data sources of a multi-modal dataset to a universal input space. This is the space of persistence diagrams, which can be easily vectorized using persistence images to matrices of the same size. These can then be stacked together for input to a machine learning model.

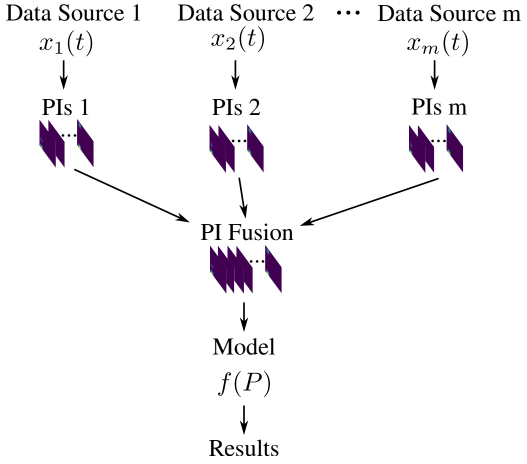


Figure 2. General method pipeline for studying multi-modal data using persistence image fusion.

We outline this procedure in Fig. 2. We begin with m data sources, $x_1(t) \ldots x_m(t)$, and extract the persistent homology using a method determined by their particular datatype. For example, time series are typically studied using zero-dimensional sublevel set persistence and Vietoris-Rips filtrations of their time-delay embeddings [21, 22, 25, 28], while image data is typically studied using one-dimensional sublevel set persistence [2, 13]. The output of this process is a set of PD capturing the persistent topological features found within the data. These PDs are vectorized into PIs and then stacked together into a tensor. In this work we apply this framework for multi-modal data sets that can have multiple types of filtrations resulting in several PIs for each data source. The PIs for each data source can then be stacked together in the fusion step which results in a 3-tensor data structure which can be used for downstream applications.

A benefit of this representation is that it allows for many standard neural network architectures (e.g. convolution neural networks) that are used to learn on multi-channel images to be natively applied to the stacked PI tensor. The only requirement for applying this early topological fusion framework is an appropriate method of calculating persistent homology given the data type. However, as the field of TDA matures this becomes less of an issue as more advanced topological methods for studying data types continue to be developed.

4. Experiments

Here we describe the experiments on two datasets we ran using TopFusion. The first is the Wearable Stress and Affection Dataset (WESAD), which is composed of 10 biological signals recorded using both a wrist and chest wearable sensor. The second dataset is the Audio Visual-Modified National Institute of Standards and Technology (AV-MNIST) as a multi-modal version of the popular MNIST dataset.

The neural network architecture applied to all of these experiments is a simple Convolution Neural Network (CNN). This CNN has two convolutional layers, each with 5×5 filters, and each followed by maximum pooling with filter size 2×2, and one fully connected layer. We trained the CNN for 20 epochs for both experiments.

4.1. Wearable Stress Affect Detection

Our first experiment uses the WESAD dataset [29]. This dataset is composed of 10 total signals collected from two sensors: one chest and one wrist sensor. These sensors measure multiple biological signals. Specifically, the chest sensor measures three-axis acceleration (ACC), Electrocardiogram (ECG), Electromyography (EMG), Electrodermal Activity (EDA), temperature, and respiration data, and the wrist signal collects ACC, Blood Volume Pulse (BVP), EDA, and temperature data. The signals were measured at different sampling rates ranging from 4 to 700 Hz. While the signals were collected, the 15 subjects were in one of three different states of interest: stressed, amused, and baseline. The goal of this dataset is to use the signals to determine which state the subjects are in using a Leave-One-Subject-Out (LOSO) train/test split.

Our pipeline for applying our method (see Fig. 2) to the WESAD dataset is shown in Fig. 3. The first step is to create window snapshots of the data, which will be classified as either stressed, amused, or baseline.

We use a window size of $w = 150\tau$, where we select τ using the mutual information method [11] due to the nonlinear nature of the time series. We also chose to shift the sliding windows by a distance $s = 0.25w_{\max}$ for each consecutive window, where $w_{\max}$ is the largest window size of any of the signal types. These parameters resulted in a total of 1620 windows across all subjects. Reasoning for our window size and shift parameter selection are provided in the Appendix in the supplemental material.

After generating the windows, we extract topological features using both zero-dimensional sublevel set filtration and time-delayed embeddings with VR filtration on the windowed time series. For the VR filtration approach we first need to embed the time series using the time delay-embedding for sampled time series as $v_i = [x_i, x_{i+\tau}, \ldots, x_{i+(n-1)\tau}] \in \mathbb{R}^n$, where each v_i is a vector in the result time-delay embedding point cloud χ. We set the dimension n using the false nearest neighbors [18] approach resulting in dimensions $n \in [3,5]$ depending on the signal. A table of all the dimensions and delays for the time-delay embedding are provided the Appendix (see supplemental material).

The PIs for each of the filtrations were generated using the procedure outlined in Appendix (see supplemental material). The complete collection of PIs coming from both the VR and sublevel set filtrations are fused, i.e. stacked, and then input into a shallow CNN for classification.

4.2. WESAD Results

For the WESAD dataset we used LOSO train/test splits for each of the 15 subjects and then report the average accuracy. We performed this experiment using just the wrist data, just the chest data, and the combination of the wrist and chest data through our PI fusion method as outlined in Fig. 3. These accuracies are reported in Table 1.

As a first mode of comparison we also calculated the summary statistics reported in [29] as a point of compared. We used a support vector machine with an RBF kernel to classify the states from these statistics with accuracies shown in Table 1 for the wrist stats, chest stats, and wrist and chest stats combined.

Table 1. Classification accuracies for WESAD dataset for list data representations and subsets

Data	Accuracy	Uncertainty (Standard Deviation)
Wrist PIs	71.9	7.2
Chest PIs	82.3	6.8
Wrist and Chest PIs	84.3	5.4
Wrist Stats	63.2	12.9
Chest Stats	54.7	6.7
Wrist and Chest Stats	63.4	12.5

From this analysis it is clear we did not replicate the levels of accuracies originally reported in [29] using the statistics, which were 75.2%, 76.5%, and 79.9% for all wrist, all chest, and all physio signals, respectively. We believe this is due to the difficulty of recreating many of the specific statistics that require pre-processing of the signals. However, even compared to the reported accuracies our PI fusion approach is outperforming these accuracies for both the chest PIs and wrist/chest PIs. Furthermore, in [16] a study applying TDA to the WESAD dataset was done using persistence landscapes. In their work they achieved on average 81.35% accuracy for all signals, which is again lower in comparison to our 84.3% accuracy when using all signals.

4.2.1 Imputation on WESAD

We also conducted several imputation experiments on the WESAD dataset to understand how imputation on the fused PI tensors performed in comparison to both imputation on the raw data as well as imputation on the statistics. These imputation experiments are outlined in Fig. 4.

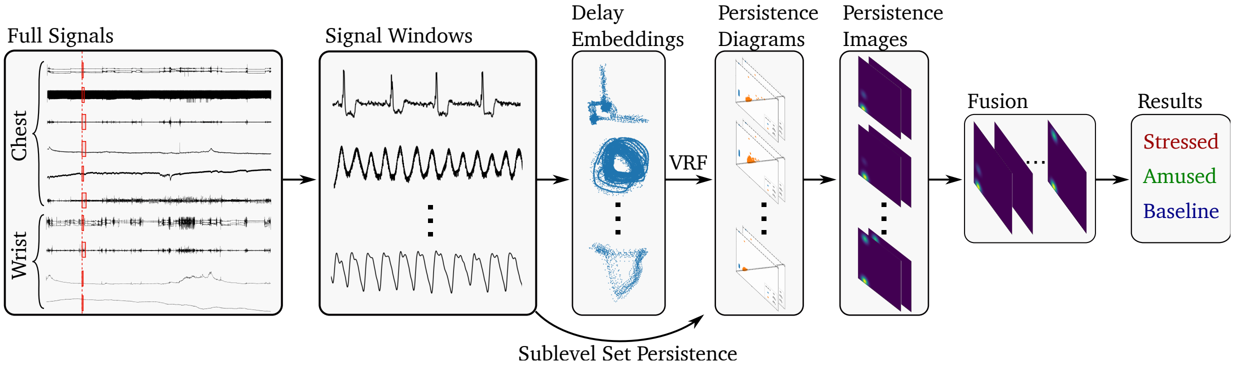
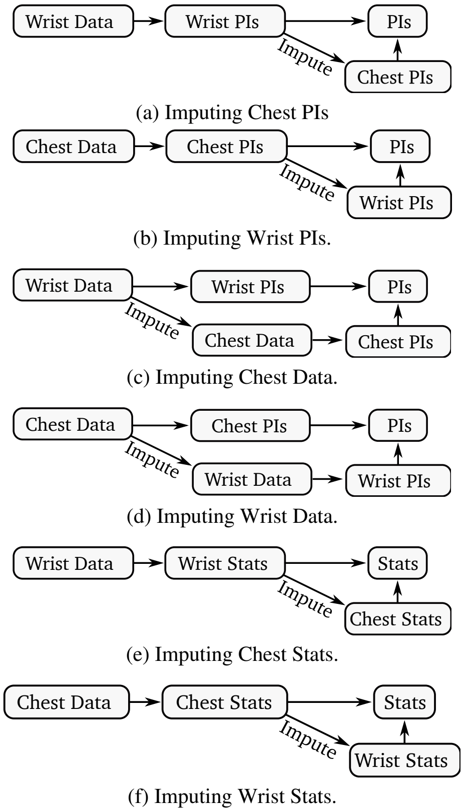


Figure 3. Pipeline for applying persistence image fusion to the WESAD dataset.



(a) Imputing Chest PIs

(b) Imputing Wrist PIs.

(c) Imputing Chest Data.

(d) Imputing Wrist Data.

(e) Imputing Chest Stats.

(f) Imputing Wrist Stats.

Figure 4. Imputation methods for the WESAD dataset.

The imputation approach used and the corresponding accuracies are shown in Table 2. For all of these imputations the *k*-NN approach was used as described in Section 2.2. We have also listed the time needed to perform imputation for each of the approaches in Table 2 as a metric to demonstrate computational demand of the various imputation approaches.

Our first observation is that due to the significant size of the raw time series data, imputation on the raw data is by far the most computationally expensive as shown in Table 2. Imputation on the stats is the fastest, but it is not significantly faster than the imputation on the chest PIs. We found little improvement from the imputed stats for both the chest and wrist. We also see this trend with imputation on the raw data with the classification accuracy decreasing in comparison to the classification on just the wrist and chest PIs separately. This is likely due to the mis-pairing of data modalities compounding the accuracy loss.

In comparison to both the raw data and stats imputation, the imputation on the PI tensors from the chest and wrist data increases performance. Specifically, by imputing the chest PIs we saw a 0.5% accuracy increase and for imputed wrist PIs we saw a 1.5% accuracy increase.

4.3. AV-MNIST

The second dataset we apply our method to is the Audio-Visual MNIST Dataset (AV-MNIST). This dataset is composed both both spoken audio signals of numbers and image scans of handwritten numbers from 0 to 9. The goal of this dataset is to classify the number using both the image and audio data combined. We apply the TopFusion pipeline to this dataset using both sublevel set persistence and zigzag persistence.

For the image data we used both a sublevel set persistence on the raw image itself as well as a zigzag persistence scan of the image as described in section 2.3.3. The sublevel set persistence results in one persistence diagram with the zigag persistence resulting in two persistence diagram. For the audio signals we first begin by breaking them into 10 sections (5 shown in figure for illustration purposes) and taking the sublevel set persistence for each. By stacking all of the resulting 13 PIS together we get a fused PI of size 13x10x10 for each image-audio pair to feed to our neural network model for digit classification. We use a 70/30 train/test split chosen using a random shuffle repeated 5 times (random seeds 1 to 5). We report our accuracies as the mean accuracy over the five random seeds with reported standard deviation.

Table 2. Classification accuracies for WESAD dataset for list data representations and subsets

Imputation Fig.	Data	Accuracy	Uncertainty (Standard Deviation)	Imputation Time (Sec)
4a	Imputed Chest PIs	72.4	6.2	3.3
4b	Imputed Wrist PIs	83.8	5.8	4.9
4c	PIs from Imputed Chest Data	68.2	6.3	145.2
4d	PIs from Imputed Wrist Data	80.1	6.1	183.3
4e	Imputed Chest Stats	63.4	13.1	0.3
4f	Imputed Wrist Stats	52.5	12.8	0.4

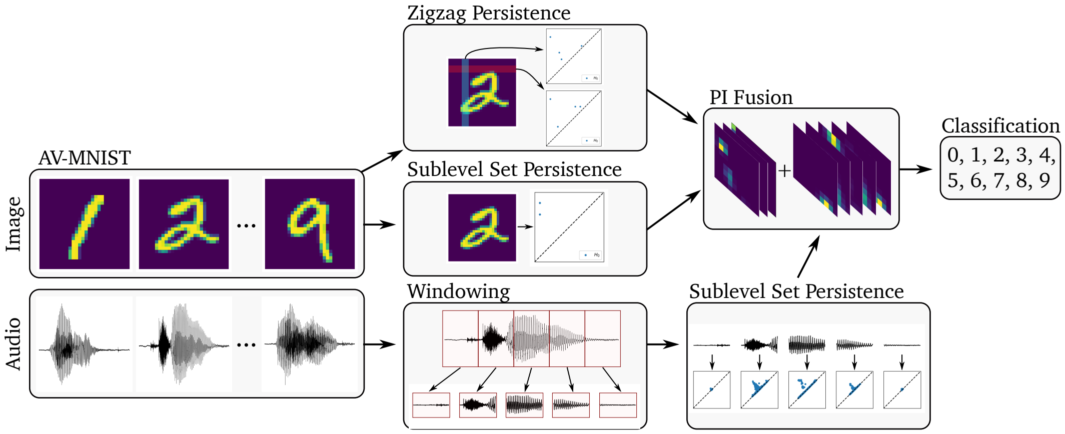


Figure 5. Pipeline for applying persistence image fusion to the AV-MNIST dataset.

4.4. AV-MNIST results

Our first experiment was using the PI fusion pipeline shown in Fig. 5 for the Audio PIs, Image PIs, and both Audio and Image PIs. These results are shown in Fig. 3. Both the audio and image PI stacks performed comparably with 88.1% and 87.7% accuracy, respectively. However, the combined PI stacked saw a significant improvement in performance with 98.4% accuracy. While these accuracies are lower than state-of-the-art neural network architectures designed specifically for this dataset, our architecture is relatively simple only using a stand, simple CNN previously described.

Table 3. Classification accuracies for AV-MNIST dataset for list data representations and subsets

Data	Accuracy	Uncertainty (Standard Deviation)
Audio PIs	88.1	1.2
Images PIs	87.7	0.9
Audio and Image PIs	98.4	0.3

We were able to get higher accuracies than reported in table 3 by using higher resolution PIs and more filtration directions when using zigzag persistence on images. Specifically, we got 91.8% accuracy using just the audio data with audio file broken into 20 windows instead of 10 with the PIs having 20×20 pixels instead of 10×10. We also got up to 96.5% accuracy using the image data when we used an ensemble of window widths ranging from 1 to 3 and rotated the image by 0, 30, 45, and 60 degree angles, which is comparable in accuracy to state-of-the-art TDA based methods for the MNIST dataset [12]. The combined accuracy for both of these improvements was 99.3% accuracy, which is near in performance to state-of-the-art methods. We believe we can further improve this accuracy with more window widths and more image rotations. One drawback of including more PIs in the stack is the increased computation and training times. As such, we chose to use the limited and low resolution PIs (10×10 pixels) as described in Section 4.3.

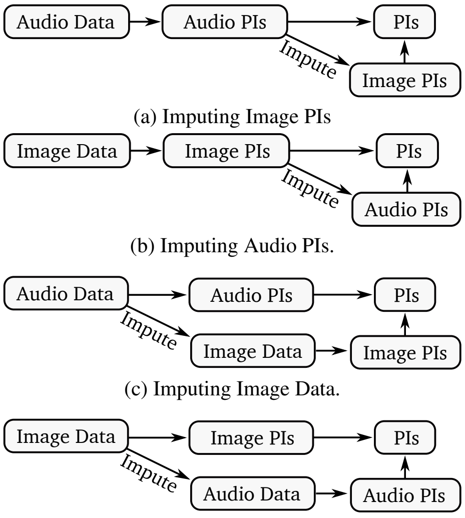
4.4.1 Imputation on AV-MNIST

For the AV-MNIST dataset we looked at four different imputation pipelines illustrated in Fig. 6. Figures 6a and 6b show our imputation on the PIs as topological summaries and Figures 6c and 6d show the pipeline for imputing the raw data and then generating the PIs from the imputed data.

Our first experiment with the WESAD dataset showed that imputation on the topological summaries improved the classification accuracy and imputation on the raw data de-

Table 4. Classification accuracies for AV-MNIST dataset for list data representations and subsets

Imputation Fig.	Data	Accuracy	Uncertainty (Standard Deviation)	Imputation Time (Sec)
6a	Imputed Image PIs	92.1	0.6	48.3
6b	Imputed Audio PIs	92.5	0.5	7.1
6c	PIs from Imputed Image Data	82.2	1.5	1289.2
6d	PIs from Imputed Audio Data	89.4	0.9	114.7



(a) Imputing Image PIs

(b) Imputing Audio PIs.

(c) Imputing Image Data.

(d) Imputing Audio Data.

Figure 6. Imputation methods for the AV-MNIST dataset.

creased it. We again see a similar result here in Table 4 with imputation on the PIs increasing the accuracy by 4.0% and 4.8% when imputing the image and audio data, respectively. In comparison, when computing the raw data, the classification accuracy decreased by 5.9% and increase by just 1.7% for imputation on the raw image and audio data, respectively.

We also show a similar result to our first experiment with imputation on the raw data taking significantly longer to compute on the raw data in comparison to the PI stacks as topological summaries of the data modalities.

There was also the challenge of making the data sizes the same for the raw data imputation. To make the time series equivalent length we chose to use zero padding, which has negative drawbacks of making the comparison between time series less accurate due to the data manipulation.

5. Discussion

Our results on classification accuracy for both the WESAD and AV-MNIST datasets show that the fusion of PIs provides comparable and in some cases better classification accuracies than other more complex approaches. We believe these accuracies can be further improved by including more filtrations on the data. This result means that more experimentation needs to be performed to find better “views" of the data using TDA to improve accuracy to the desired level. This of course has the drawback of longer compute and training times, but we are able to tune the desired accuracy and training time by using more or less fused PIs as input into our model.

We believe that the increased performance when imputing on the fused PI tensors in comparison to raw data and feature vectors is due to the robustness or stability of persistent homology. Specifically, it has been shown that for a small change in signals there is only a small change in the resulting persistence diagram [5], which is translated to the persistence image with an appropriate choice of weighting function [1]. As such, we expect that the slight differences between signals does not have a serious effect on the resulting fused PI tensors and the imputation learning on the topological summaries, which the imputer is learning from making it more robust to the signal differences.

6. Conclusion

In this work we presented a novel method for fusing multi-modal dataset using topological feature called TopFusion. TopFusion is applied to two datasets and tested using complete and incomplete observations. The presented results showed better performance than both standard approaches, which requires ad-hoc feature engineering, and other topological approaches. Additionally we utilized a new a filtration method for “scanning" image data using zigzag persistence. In subsequent work we are exploring the theoretical foundations of this new filtration approach and formalize its scaling properties for use on high-resolution imagery.

We have demonstrated that using TopFusion for imputation for incomplete data can outperform imputation on structured feature vectors as well as direct imputation on the data. We hypothesize that the improved imputation performance is due to the stability of topological summaries. In future work we plan to experiment with more datasets to gain a better understanding on the performance of TopFusion against other state-of-the-art methods in multi-modality fusion.

References

[1] Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18, 2017.

[2] Paul Bendich, Herbert Edelsbrunner, and Michael Kerber. Computing robustness and persistence for images. *IEEE transactions on visualization and computer graphics*, 16(6):1251–1260, 2010.

[3] Gunnar Carlsson and Vin de Silva. Zigzag persistence. *Foundations of Computational Mathematics*, 10(4):367–405, April 2010.

[4] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *arXiv preprint arXiv:1710.04019*, 2017.

[5] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 263–271, 2005.

[6] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[7] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.

[8] Herbert Edelsbrunner and John Harer. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

[9] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.

[10] Tlamelo Emmanuel, Thabiso Maupong, Dimane Mpoeleng, Thabo Semong, Banyatsang Mphago, and Oteng Tabona. A survey on missing data in machine learning. *Journal of Big Data*, 8(1):1–37, 2021.

[11] Andrew M Fraser and Harry L Swinney. Independent coordinates for strange attractors from mutual information. *Physical review A*, 33(2):1134, feb 1986.

[12] Adélie Garin and Guillaume Tauzin. A topological" reading" lesson: Classification of mnist using tda. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1551–1556. IEEE, 2019.

[13] Kathryn Garside, Robin Henderson, Irina Makarenko, and Cristina Masoller. Topological data analysis of high resolution diabetic retinopathy images. *PloS one*, 14(5):e0217413, 2019.

[14] D.L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[15] Sebastian Jäger, Arndt Allhorn, and Felix Bießmann. A benchmark for data imputation methods. *Frontiers in big Data*, 4:693674, 2021.

[16] Alperen Karan and Atabey Kaygun. Time series classification via topological data analysis. *Expert Systems with Applications*, 183:115326, 2021.

[17] Bahador Khaleghi, Alaa Khamis, Fakhreddine O Karray, and Saiedeh N Razavi. Multisensor data fusion: A review of the state-of-the-art. *Information fusion*, 14(1):28–44, 2013.

[18] Anna Krakovská, Kristína Mezeiová, and Hana Budáčová. Use of false nearest neighbours for selecting variables and embedding parameters for state space reconstruction. *Journal of Complex Systems*, 2015, 2015.

[19] Dana Lahat, Tülay Adali, and Christian Jutten. Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477, 2015.

[20] Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

[21] Elizabeth Munch. A user's guide to topological data analysis. *Journal of Learning Analytics*, 4(2):47–61, July 2017.

[22] Audun Myers. *Dynamical Systems Analysis Using Topological Signal Processing*. Michigan State University, 2022.

[23] Audun Myers, Cliff Joslyn, Bill Kay, Emilie Purvine, Gregory Roek, and Madelyn Shapiro. Topological analysis of temporal hypergraphs. *arXiv preprint arXiv:2302.02857*, 2023.

[24] Audun Myers, Firas Khasawneh, and Elizabeth Munch. Temporal network analysis using zigzag persistence. *arXiv preprint arXiv:2205.11338*, 2022.

[25] Audun D Myers, Melih Yesilli, Sarah Tymochko, Firas Khasawneh, and Elizabeth Munch. Teaspoon: A comprehensive python package for topological signal processing. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, volume 30, page 033130. AIP Publishing, mar 2020.

[26] Norman H Packard, James P Crutchfield, J Doyne Farmer, and Robert S Shaw. Geometry from a time series. *Physical review letters*, 45(9):712, 1980.

[27] Jose A. Perea. A brief history of persistence. September 2018.

[28] Nalini Ravishanker and Renjie Chen. An introduction to persistent homology for time series. *Wiley Interdisciplinary Reviews: Computational Statistics*, 13(3):e1548, 2021.

[29] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM international conference on multimodal interaction*, pages 400–408, 2018.

[30] Floris Takens. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980: proceedings of a symposium held at the University of Warwick 1979/80*, pages 366–381. Springer, 2006.

[31] Sarah Tymochko, Elizabeth Munch, and Firas A. Khasawneh. Using zigzag persistent homology to detect hopf bifurcations in dynamical systems. *Algorithms*, 13(11):278, oct 2020.

[32] Valentin Vielzeuf, Alexis Lechervy, Stéphane Pateux, and Frédéric Jurie. Centralnet: a multilayer approach for multimodal fusion. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[33] Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, November 2004.