Contrastive Multivariate Singular Spectrum Analysis

Abdi-Hakin Dirie[†] **Abubakar Abid**[‡] **James Zou**[‡]
[‡]Department of Electrical Engineering, Stanford University
[†]abdidirie0@gmail.com [‡]{a12d,jamesz}@stanford.edu

Abstract

We introduce Contrastive Multivariate Singular Spectrum Analysis, a novel unsupervised method for dimensionality reduction and signal decomposition of time series data. By utilizing an appropriate background dataset, the method transforms a target time series dataset in a way that evinces the subsignals that are *enhanced* in the target dataset, as opposed to only those that account for the greatest variance. This shifts the goal from finding signals that *explain* the most variance to signals that *matter* the most to the analyst. We demonstrate our method on an illustrative synthetic example, as well as show the utility of our method in the downstream clustering of electrocardiogram signals from the public MHEALTH dataset.

1 Introduction

Unsupervised dimensionality reduction is a key step in many applications, including visualization [8] [10], clustering [5] [11], and preprocessing for downstream supervised learning [13]. Principal Component Analysis (PCA) is one well-known technique for dimensionality reduction, which notably makes no assumptions about the ordering of the samples in the data matrix $X \in \mathbb{R}^{N \times D}$. Multivariate Singular Spectrum Analysis (MSSA) [7] is an extension of PCA for time series data, which been successfully applied in applications like signal decomposition and forecasting [6] [9] [12]. In MSSA, each row is read at a certain time step, and thus is influenced by the ordering of the samples. MSSA works primarily by identifying key oscillatory modes in a signal, which also makes it useful as a general-purpose signal denoiser. However, MSSA (and PCA) is limited to finding the principal components that capture the maximal variance in the data. In situations where the information of interest explains little overall variance, these methods fail to reveal it. Recently, extensions like contrastive PCA (cPCA) [1] have shown that utilizing a background dataset $Y \in \mathbb{R}^{M \times D}$ can help better discover structure in the foreground (target) X that is of interest to the analyst.

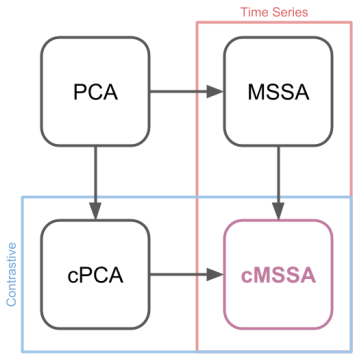


Figure 1: Schematic illustrating the relations among PCA, cPCA, MSSA, and cMSSA.

Contrastive Multivariate Singular Spectrum Analysis (cMSSA) generalizes cPCA and applies it to time series data. Figure 1 visualizes the relationships between the four methods. As a contrastive method, cMSSA emphasizes salient and unique sub-signals in time series data rather than just those that explain most of the signal variance. While standard MSSA is useful for denoising a signal, cMSSA additionally "denoises" signals of structured but irrelevant information.

Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2 Contrastive Multivariate Singular Spectrum Analysis

2.1 Standard MSSA

Consider a centered one-channel times series $\mathbf{x} \in \mathbb{R}^T$. We construct a Hankel matrix $H_\mathbf{x} \in \mathbb{R}^{T' \times W}$ with window size W as follows:

$$H_\mathbf{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_W \\ x_2 & x_3 & \dots & x_{W+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{T'} & x_{T'+1} & \dots & x_T \end{pmatrix}$$

where $T' = T - W + 1$. To extend to the multivariate case, let $X \in \mathbb{R}^{T \times D}$ be a D-channel time series that runs for T steps. We construct the Hankelized matrix H_X with window W by horizontally concatenating the per-channel Hankel matrices into a T'-by-DW matrix: $H_X = [H_{\mathbf{x}^{(1)}}; H_{\mathbf{x}^{(2)}}; \dots; H_{\mathbf{x}^{(D)}}]$. Next we compute the covariance matrix $C_X \in \mathbb{R}^{DW \times DW}$ for H_X. The next step is to perform the eigendecomposition on C_X, yielding DW eigenvectors. Of these we take the top K vectors with the largest corresponding eigenvalues. We denote $\mathbf{e}^{(k)}$ as the eigenvector with the kth largest eigenvalue. We collect the vectors into a matrix $E \in \mathbb{R}^{DW \times K}$.

To transform our original time series X, we have two options: (a) Project X into the principal component (PC) space defined by E: $A = H_X E$ or (b) use A to compute the kth reconstructed component (RC) $R^{(k)}$ as done in the SSA literature:

$$R_{tj}^{(k)} = \frac{1}{W_t} \sum_{t'=L_t}^{U_t} A_{t-t'+1,k} \cdot \mathbf{e}_{(j-1)W+t'}^{(k)}$$

where $L_t = \max(1, t - T + W)$, $U_t = \min(t, W)$, and $W_t = U_t - L_t + 1$. Summing up the reconstructed components reproduces a denoised version of the original signal. For our purposes, we opt instead to take the horizontal concatenation of the reconstructed components as the second transform: $R = [R^{(1)}; R^{(2)}; \dots; R^{(K)}]$. To handle multiple time series, we vertically stack each Hankelized matrix. The algorithm proceeds identically from there.

2.2 Contrastive MSSA

The modification to MSSA we introduce is via a new variable $\alpha \geq 0$ we call the *contrastive* hyperparameter. We construct H_Y for another D-channel times series Y (the background data) via the same process. It is not required that X and Y run for the same number of time steps, only that their channels are aligned. We compute a contrastive covariance matrix $C = C_X - \alpha C_Y$ and perform the eigendecomposition on C instead of C_X. The intuition for this is that by subtracting out a portion of the variance in Y, the remaining variance in X is likely to be highly specific to X but not Y. This is the key additional mechanism behind cMSSA — if $\alpha = 0$, then no contrast is performed, and cMSSA reduces down to just MSSA.

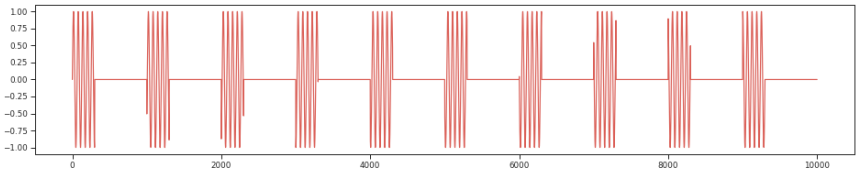
The choice of α is non-trivial. We outline a routine for auto-selecting a small number of promising values for α in the appendices.

3 Experiments

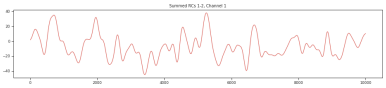
3.1 Synthetic example

To illustrate cMSSA, we present a simple synthetic example. We generate an artificial one-channel signal Y by sampling 500 sinusoids with different frequencies, amplitudes, phases, and vertical shifts. White Gaussian noise sample from $\mathcal{N}(0, 1)$ is added in as well. We generate X in the same manner, but add in a very specific sub-signal (Figure 2a) that has comparatively low variance compared to the whole time series. The signals X and Y are generated independently as to rule out simple signal differencing as an explanation. We take X as foreground and Y as background.

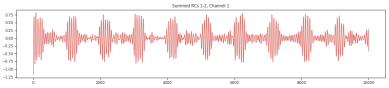
We set $W = 100$, $\alpha = 2$, and use only the top $K = 2$ RCs. Figure 2 displays the reconstructions computed by MSSA versus cMSSA, alongside the sub-signal that was injected into X. Specifically,



(a) Sub-signal specific to the foreground data X.



(b) Without contrast: X.



(c) After contrast.

Figure 2: Results of synthetic experiment. (a) The sub-signal that is present in the foreground X. (b) The non-contrastive reconstruction of X, which simply resembles the original sum-of-sinusoids signal. (c) The contrastive reconstruction of X, which teases out the desired sub-signal after using Y as background.

we see that the cMSSA reconstruction shown in Figure 2c yields a noisy approximation of the subsignal of interest, Figure 2a. Note that the variance of the noise is comparable to the variance of the sub-signal—more noise would eventually overpower cMSSA's ability to extract the sub-signal.

3.2 Clustering of electrocardiograms

Data: The data used in our experiments is taken from the public MHEALTH dataset [3]. In the dataset, 10 individuals were asked to perform 12 physical activities as various sensors recorded various motion data. In addition, the researchers also collected two-lead electrocardiogram (ECG) readings, which we take as dual-channel time series data. In addition to the 12 activities, there is a 13th NULL class that represents ECG signals collected between each activity but which don't have labels themselves. To increase the number of individual time series, we partition each one in half.

For our experiments, the foreground data are all time series labelled as either JOGGING, RUNNING, JUMPING, or CYCLING, 20 time series each for a total of 80. These four, being the more cardio-intensive of the 12, had much more signal activity that would be needed to be sifted through, exactly the type of environment cMSSA is intended to handle. For background data, we take all 272 time series belonging to the NULL class.

Setup: To evaluate the effectiveness of cMSSA over its non-contrastive counterpart, we run both cMSSA and MSSA with a variety of hyperparameter settings. For each fitted model, we transform the foreground data to both the PC and RC spaces. Once the transformations are had, we perform spectral clustering into 4 clusters and compare the resulting clusters to the activity labels on the time series data, which were hitherto withheld from the algorithms. There are 3 hyperparameters: the window size $W \in \{8, 16, 32, 84, 18\}$, the number of desired components $K \in \{1, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20\}$, and the contrastive parameter α. We set K only if the value is less than or equal to DW (where $D = 2$ in this case). For α, we used our automatic routine to compute five key values to try for each setting of W and K. Of these five, one of them is zero, representing standard MSSA. Altogether, we run 530 experiments, 106 of which are standard MSSA, and the remaining cMSSA.

The spectral clustering requires an affinity matrix $S \in \mathbb{R}^{N \times N}$ which contains the similarities between any pair of time series, where N is the number of times series we wish to cluster. Let $X^{(i)}$ and $X^{(j)}$ be two time series. Using the FastDTW metric [14] with a euclidean norm[1], we define the similarity S_{ij} to be $\frac{1}{\text{FASTDTW}(X^{(i)}, X^{(j)})+1}$. The cluster evaluation uses the well-rounded BCubed metric [2] to compute the precision, recall, and F1 harmonic mean for a particular cluster prediction. We also perform the evaluation in the model-free sense where we simply cluster the time series with no transformation as a basic baseline.

[1]FastDTW is not a symmetric metric, so we take the minimum between the two orderings of the operands.

Table 1: Best cMSSA and MSSA results in terms of maximum F1 score. Model-free clustering baseline also included. The transform column indicates which transform was applied before clustering. Best metric per column is bolded.

Model	W	K	Transform	Precision	Recall	F1
Model-free	-	-	None	50.49	48.82	49.54
MSSA	16	16	A	57.67	64.63	60.95
cMSSA ($\alpha = 12.41$)	128	1	A	**65.44**	**75.88**	**70.27**

Results: Table 1 reports the best representative contrastive and non-contrastive models, comparing both to the model-free baseline. We observe a number of things. First, both MSSA and cMSSA outperform the model-free baseline. Second, cMSSA has 9-10 point gains over cMSSA in each of precision, recall, and F1. Third, both find that using A over R as the transform yielded better results. Finally, and most interestingly, is the number of PCs used. Of the DW number of PCs available, MSSA gets its best performance using half (16 out of 32). The ratio is very different for cMSSA, which only uses one PC out of the maximum of 256 available. This highlights an interesting efficiency of cMSSA. By filtering out unnecessary components, the remaining not only account for less signal variance, but provide diminishing returns with each subsequent component used.

Figure 3 shows a more granular view of the general gains to be had from using cMSSA. For a particular setting of W and K, we plot the F1 score for the non-contrastive case vs the contrastive case. Due to the four values of αs used in the contrastive case, we take the model that had the greatest F1. The line is a visual guide – points below the line mean that the contrast was useful for a particular setting of the hyperparameters.

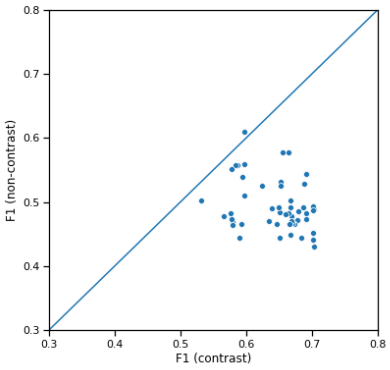


Figure 3: Plot of paired F1 scores. Each point is for a particular setting of W and K. The contrastive F1 score used is the maximum of the four runs (one per α tried) for that setting of the hyperparams. $x = y$ line drawn as guidance. The points look at only those where the transform used is A.

4 Conclusion

We have developed cMSSA as a general tool for dimensionality reduction and signal decomposition of temporal data. By introducing a background dataset, we can efficiently identify subsignals that are enhanced in one time series data relative to another. In an empirical experiment, we find that for virtually any setting of the hyperparameters, cMSSA is more effective at unsupervised clustering than MSSA, contingent on appropriate choices for the foreground and background data.

Some basic heuristics should be kept in mind when choosing to use cMSSA as an algorithm. First, the data ideally should exhibit periodic behavior, as MSSA (and by extension, cMSSA) is particular well suited to finding oscillatory signals. Second, X and Y should not be identical, but should share common structured signal such that the contrast retains some information in the foreground. As an example, the ECG foreground data consisted of subjects performing very specific activities, whereas the background consisted of a corpus of unlabelled ECG signals in which the participants performed no specific activity. We would expect a good amount of overlap in signal variance, but signals specific to the four activities would be under-represented in the background. Thus contrast is a plausible way to extract this signal.

Finally, we note that the only hyper parameter of cMSSA is the contrast strength, α. In our default algorithm, we developed an automatic scheme for selecting the most informative values of α (see the Appendix A). The experiments performed for this paper use the automatically generated values, and we believe this default will be sufficient in many use-cases of cMSSA. The user may also input specific values for α if more fine-tuned exploration is desired.

References

[1] A. Abid, M. J. Zhang, V. K. Bagaria, and J. Zou. Exploring patterns enriched in a dataset with contrastive principal component analysis. *Nature communications*, 9(1):2134, 2018.

[2] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information retrieval*, 12(4):461–486, 2009.

[3] O. Banos, R. Garcia, J. A. Holgado-Terriza, M. Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga. mhealthdroid: a novel framework for agile development of mobile health applications. In *International Workshop on Ambient Assisted Living*, pages 91–98. Springer, 2014.

[4] O. Banos, C. Villalonga, R. Garcia, A. Saez, M. Damas, J. A. Holgado-Terriza, S. Lee, H. Pomares, and I. Rojas. Design, implementation and validation of a novel open framework for agile development of mobile health applications. *Biomedical engineering online*, 14(2):S6, 2015.

[5] M. B. Cohen, S. Elder, C. Musco, C. Musco, and M. Persu. Dimensionality reduction for k-means clustering and low rank approximation. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 163–172. ACM, 2015.

[6] H. Hassani, S. Heravi, and A. Zhigljavsky. Forecasting european industrial production with singular spectrum analysis. 25:103–118, 03 2009.

[7] H. Hassani and R. Mahmoudvand. Multivariate singular spectrum analysis: A general view and new vector forecasting approach. *International Journal of Energy and Statistics*, 1(01):55–83, 2013.

[8] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[9] R. Mahmoudvand, F. Alehosseini, and P. Rodrigues. Forecasting mortality rate by singular spectrum analysis. 13:193–206, 11 2015.

[10] L. McInnes and J. Healy. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[11] D. Niu, J. Dy, and M. Jordan. Dimensionality reduction for spectral clustering. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 552–560, 2011.

[12] K. Patterson, H. Hassani, S. Heravi, and A. Zhigljavsky. Multivariate singular spectrum analysis for forecasting revisions to real-time data. *Journal of Applied Statistics*, 38(10):2183–2211, 2011.

[13] M. Pechenizkiy, A. Tsymbal, and S. Puuronen. Pca-based feature transformation for classification: Issues in medical diagnostics. In *Computer-Based Medical Systems, 2004. CBMS 2004. Proceedings. 17th IEEE Symposium on*, pages 535–540. IEEE, 2004.

[14] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[15] R. Vautard, P. Yiou, and M. Ghil. Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D: Nonlinear Phenomena*, 58(1-4):95–126, 1992.

Appendices

A Routine for candidate α generation

Algorithm 1 Procedure for generating several candidate αs.

Require: Minimum α to consider $\alpha_{\min}$, maximum α to consider $\alpha_{\max}$, total number of αs to consider n, and number of αs to return m, foreground signal X, background signal Y, window W, and number of components K.

procedure GENERATEALPHAS($\alpha_{\min}$, $\alpha_{\max}$, n, m, X, Y, W, K)
  $C \leftarrow$ LOGSPACE($\alpha_{\min}$, $\alpha_{\max}$, n) ▷ Get n αs spaced evenly in log-space.
  $C \leftarrow C \cup \{0\}$ ▷ Include zero in candidate set.

  for $\alpha^{(i)} \in C$ **do**
    $E^{(i)} \leftarrow$ GETEIGEN(X, Y, W, K, $\alpha^{(i)}$) ▷ Use cMSSA to compute eigenvectors.
  end for

  $S \leftarrow$ EMPTY($\mathbb{R}^{n+1 \times n+1}$) ▷ Initialize affinity matrix.
  for $i \in \{1, \ldots, n+1\}$ **do**
    for $j \in \{i, \ldots, n+1\}$ **do**
      $s \leftarrow \left\| {E^{(i)}}^T E^{(j)} \right\|_*$ ▷ Take nuclear norm of matrix product.
      $S_{i,j} \leftarrow s$
      $S_{j,i} \leftarrow s$
    end for
  end for

  $Z \leftarrow$ SPECTRAL(S, C, m) ▷ Get m clusters.

  $C^* \leftarrow \{0\}$ ▷ Set of best αs to return. Zero always included.
  for $z \in Z$ **do**
    if $0 \notin z$ **then** ▷ We ignore all αs that were clustered with zero.
      $\alpha^* \leftarrow$ MEDIOD(z, S) ▷ Get α with greatest mean affinity in its cluster.
      $C^* \leftarrow C^* \cup \{\alpha^*\}$
    end if
  end for
return C^*, set of m best αs, including zero.
end procedure

B Extra figures

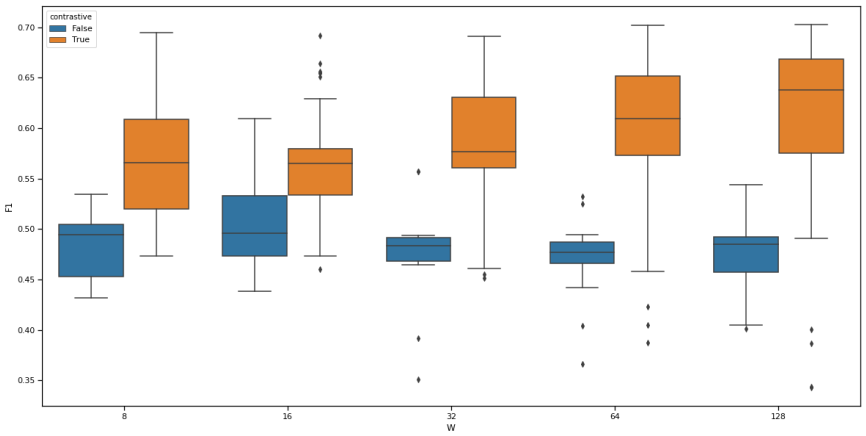


Figure 4: Box plots showing the distributions of F1 scores for particular values of W, differentiating between contrastive (orange) and non-contrastive (blue) runs.

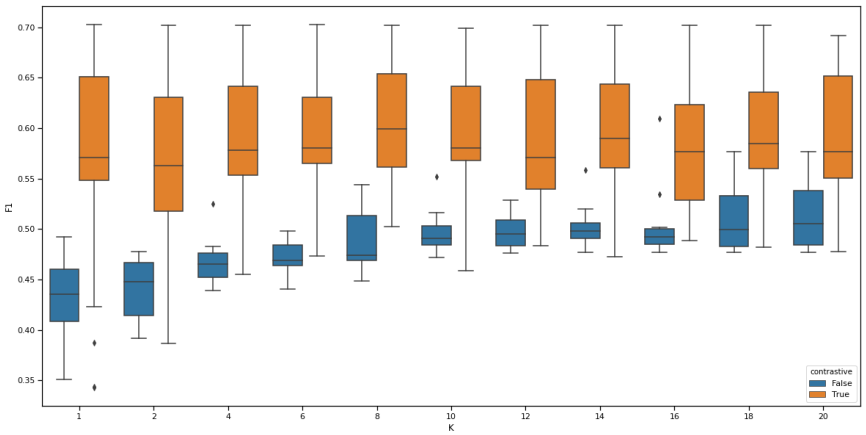
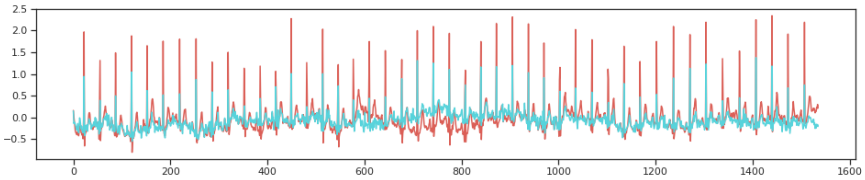
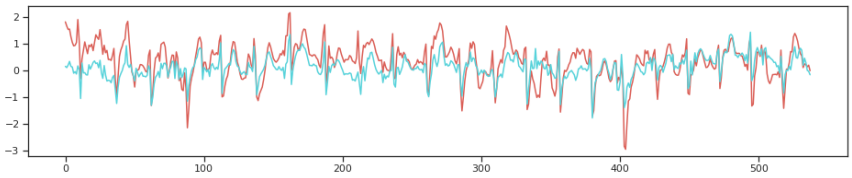


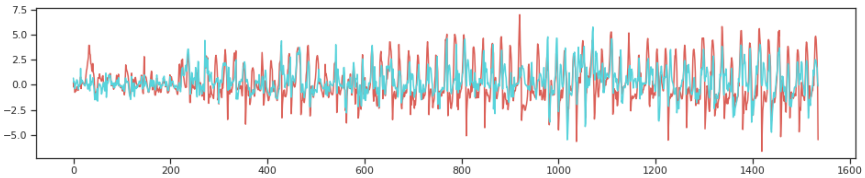
Figure 5: Box plots showing the distributions of F1 scores for particular values of K, differentiating between contrastive (orange) and non-contrastive (blue) runs.



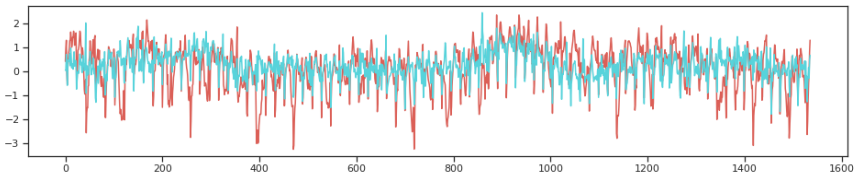
(a) Cycling



(b) Jumping

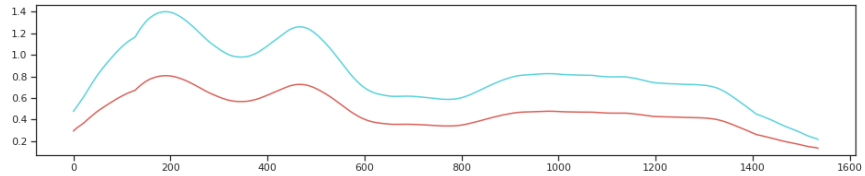


(c) Jogging

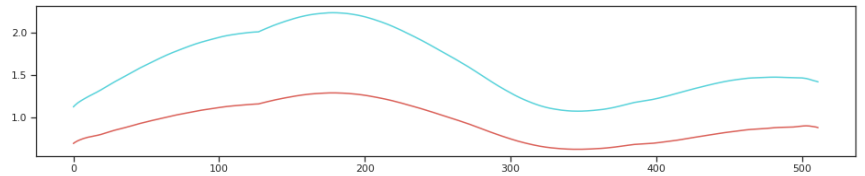


(d) Running

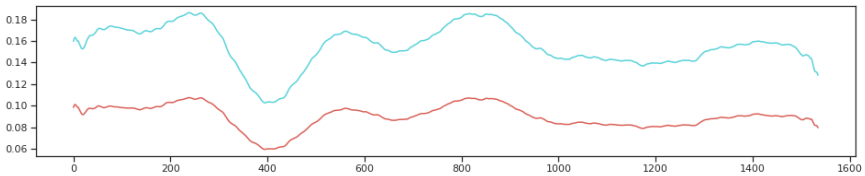
Figure 6: Per-activity random time series samples from the MHEALTH dataset. The dual channels are overlayed.



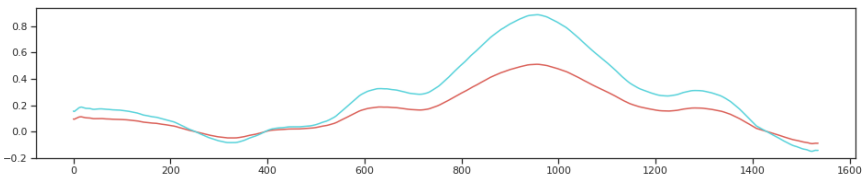
(a) Cycling



(b) Jumping

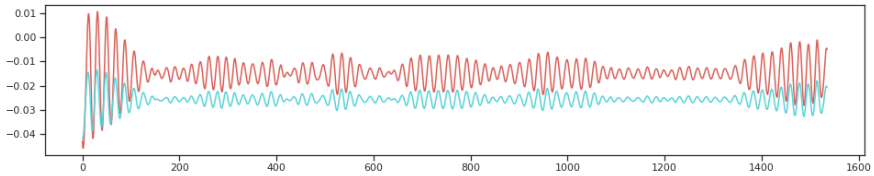


(c) Jogging

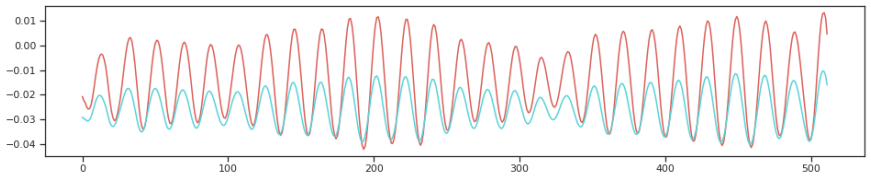


(d) Running

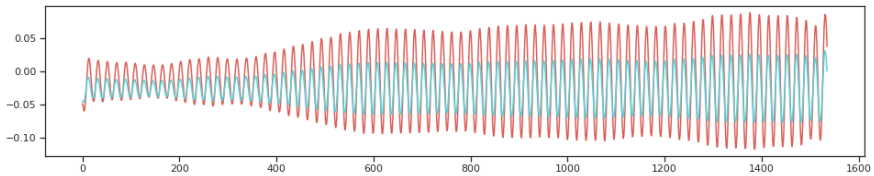
Figure 7: Per-activity random time series samples from the MHEALTH dataset, after performing MSSA with $W = 128$ and $K = 1$. The dual channels are overlayed.



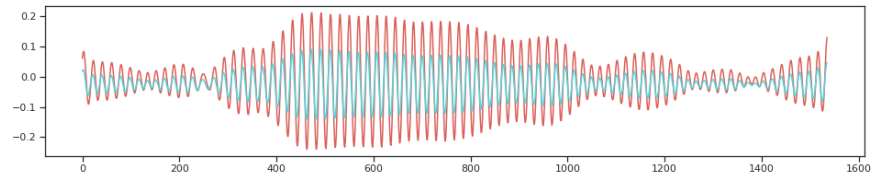
(a) Cycling



(b) Jumping

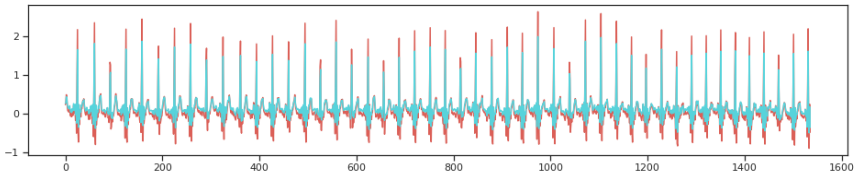


(c) Jogging

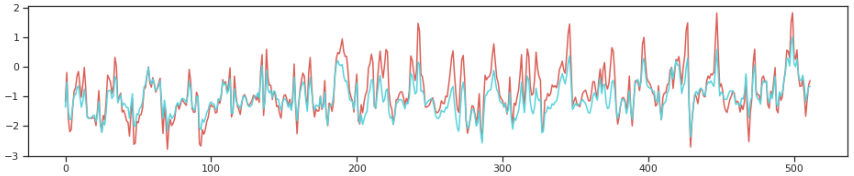


(d) Running

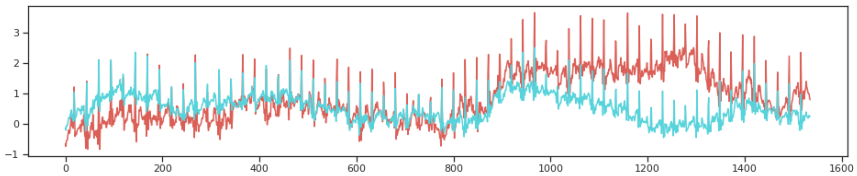
Figure 8: Per-activity random time series samples from the MHEALTH dataset, after performing cMSSA with $W = 128$, $K = 1$, and $\alpha \approx 12.41$. The dual channels are overlayed.



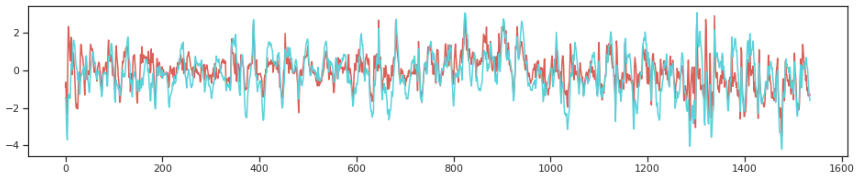
(a) Cycling



(b) Jumping



(c) Jogging



(d) Running

Figure 9: Per-activity random time series samples from the MHEALTH dataset, after performing MSSA with $W = 16$ and $K = 16$. The dual channels are overlayed.