Physical Therapy Project Report

Guangze Yu, Wuji Shan

I. Abstract

During our analysis process, we first reconstructed the time series data of GRF to show the force patterns. Then we conducted several unsupervised dimension reduction models, including the functional data analysis, the singular spectrum analysis (SSA), and traditional Autoencoder. In order to refine our models and have more practical discussions, we trained our SSA and Autoencoder models to cluster each data set we chose into four groups. We generally based the percentage of different estimated group members matches with the percentage of different real group members.

This report presents our analysis in four parts: introduction, data and methods, results, and conclusion.

II. Introduction

Mechanical loading has been implicated in knee osteoarthritis pathogenesis, suggesting that interventions aimed at changing joint loading may be vital in reducing the burden of knee osteoarthritis.

Our clients are from the Boston University Movement & Applied imaging lab, and they are conducting descriptive analyses of a large cohort (the multicenter osteoarthritis study) of the knees to compare ground reaction force (GRF) patterns during walking among legs defined by the presence or absence of knee pain and/or radiographic knee osteoarthritis.

Our clients conducted this study using the data from the Multicenter Osteoarthritis Study, including three-dimensional GRF data recorded at 1,000 Hz from both cohorts and extracted several potential features. The clients' expectation is that we could perform various methods to apply new features to describe the participants' motions and achieve some particular description outcomes.

The general flow for this report is from Functional Data Analysis, to smooth the curve, SSA, to extract the single signal for the principle components, and lastly auto-encoder and K-mean clustering.

III. Data and Methods

The analyses included data from 2824 legs contributed by 1576 individuals, including 15969 trails. The received data has already been cleaned by our clients. Each trail is already removed by obvious error and without missing values. We selected ML_GRF_stance_N, V_GRF_stance_N, and AP_GRF_stance_N data sets as our main focus of exploration, which are time-normalized stances at ground reaction force from separately medial-lateral, vertical, and anterior-posterior positions. The reason for the variable selection is based on the necessity of practical significance. The movement of knees with or without osteoarthritis disease directly relates to the shape of the above three data sets. The main goal of our project is to assess the potential of dimension reduction.

In general, our project mainly investigates two dimension reduction directions: Singular Spectrum Analysis, and feature selection(autoencoder). Singular spectrum analysis mainly includes MSSA and FSSA. Be noticed, that MSSA is not a dimension reduction technique. FSSA is a dimension reduction technique after functional data analysis. The detailed concept will be discussed in the following section.

1. Functional Data Analysis (FDA)

Functional data analysis analyzes data providing information about curves, surfaces, or anything else varying over a continuum. The idea supporting FDA is to relate discrete observations from time series with a form of function that represents the entire function as a single observation. FDA is one of the widely used noise reduction methods for smoothing curves. The curve is defined as a vector in an infinite-dimensional vector space and then constructed for a B -spline basis. We choose 23 basis functions and the smoothing constant of lambda 0.5 as our initial choice. Then, we smooth our previous three variables candidates.

Below are our first 1000 observations' results after smoothing. Color lines represent different observations, while the black line represents the mean time series curve of the 1000 observations.

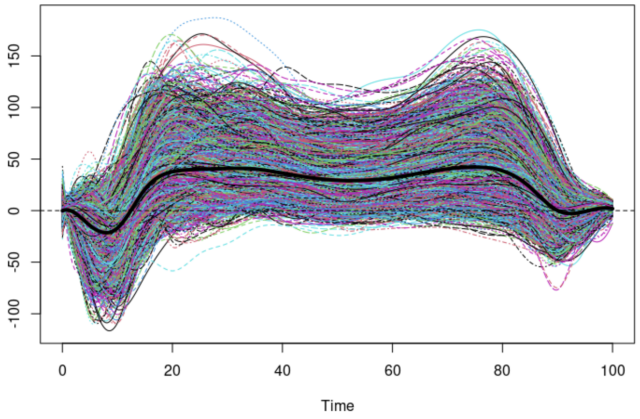


Figure 1 FDA result for ML_GRF_stance

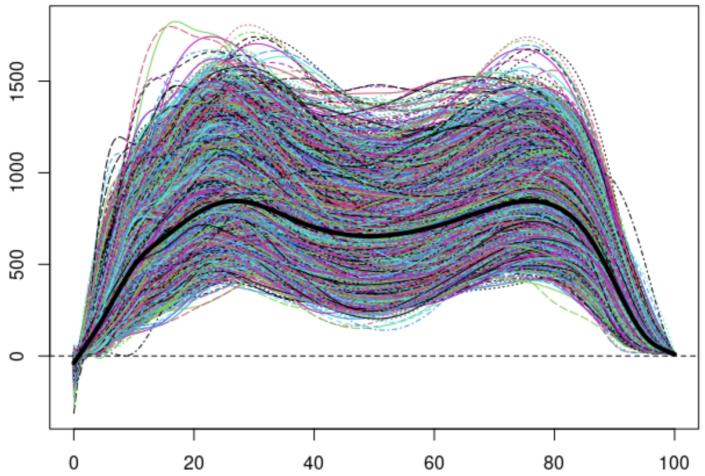


Figure 2 FDA result for V_GRF_stance

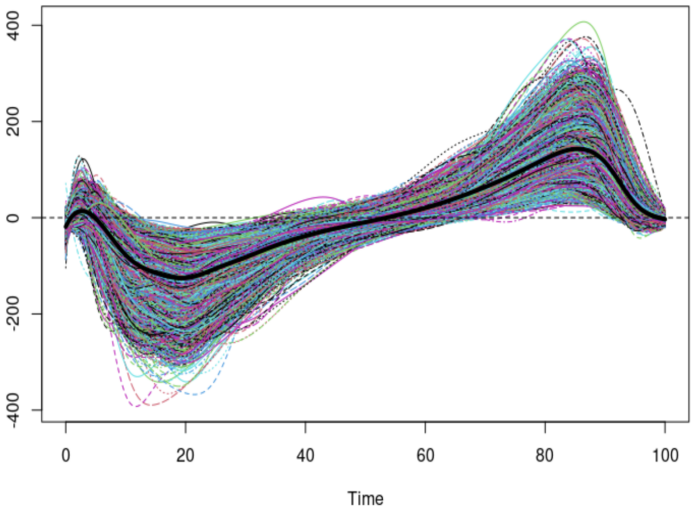


Figure 3 FDA result for AP_GRF_stance

From the above example observations, we find that the basic shape of three GRFs keeps the same after the smoothness, which is useful to keep track of useful information for our curve. This step is fundamental for our fellow analysis methods.

2. Singular Spectrum Analysis (SSA)

Singular spectrum analysis (SSA) is one of nonparametric analysis methods for time series data. The name of SSA comes from the spectrum of eigenvalues in singular value decomposition of a covariance matrix. There are a total of four steps to reconstruct the time series curve. We would use the univariate singular spectrum analysis as an example to illustrate the background information behind this method. Then, we expand the same idea to multivariate. The first step for SSA is to embed the trajectory matrix. The trajectory matrix is a linear map transforming the object X into an L×K matrix. L is defined by the window length, while K = N-L+1. N is the total length of the time series point. Below is one example trajectory matrix.

$$\mathcal{T}_{\text{1D-SSA}}(\mathbb{X}) = \begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_K \\ x_2 & x_3 & x_4 & \dots & x_{K+1} \\ x_3 & x_4 & x_5 & \dots & x_{K+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_L & x_{L+1} & x_{L+2} & \dots & x_N \end{pmatrix}.$$

Figure 4 Example of trajectory matrix

The second step is decomposition. Recalling the definition of singular value decomposition, we separate the trajectory matrix into the sum of d (rank X) matrix Xi. Xi is defined by singular value σ (the square root of eigenvalue) , corresponding eigenvector U and factor vector V. The basic formula should be summarized as below.

$$X = \sum_i X_i,\ X_i = \sigma_i U_i V_i^T$$

The third step is grouping. This step can be defined by the users who how to group the components. For example, we can group the first and the second matrix as the trend group.

The final step is restriction. We will base on the previous trending group to reconstruct the trajectory matrix. The basic formula should be summarized as below.

$$\mathbb{X} = \widetilde{\mathbb{X}}_1 + \ldots + \widetilde{\mathbb{X}}_m ,$$
$$\widetilde{\mathbb{X}}_k = \mathcal{T}^{-1} \circ \Pi_{\mathcal{H}}(\mathbf{X}_{I_k})$$

Figure 5 Equation of SSA reconstruction

After we talk about the basic definition of univariate SSA, we expand the same methodology to multivariate. Unfortunately, MSSA doesn't extract a single signal that describes the temporal evolution of the principle motion components but recover the specific oscillations for each channel. We transform the one-dimensional complex-valued series into a complex version. Also, the decomposition method for complex- valued space is Herimitian. We utilized the formula $U(t) = V_{GRF} + AP_{GRF} * i + ML_{GRF} * j$, where $i^2 = -1$, $and\ j^3 = -1$ to retrieve one line which aggregates these three data sets. Although MSSA is not a dimension reduction method, it provide new aspects for our analysis.

Below is one example graph during the process for row 388. We treated the first and second matrix with largest eigenvalue as trend and consider the rest as residuals. The Original Re, Re Trend, and Residuals Re represent correspondingly the part of V_{GRF}, and the Original lm, lm Trend, and Residuals lm represent correspondingly the part of $AP_{GRF} * i + ML_{GRF} * j$. Re represents the real number part, while lm represents the imaginary number part. From this example, we find that because we defined the V as the real number, the shape of original curve after reconstruction still keep the same. However, after the transformation, we lost the basic shape of AP and ML.

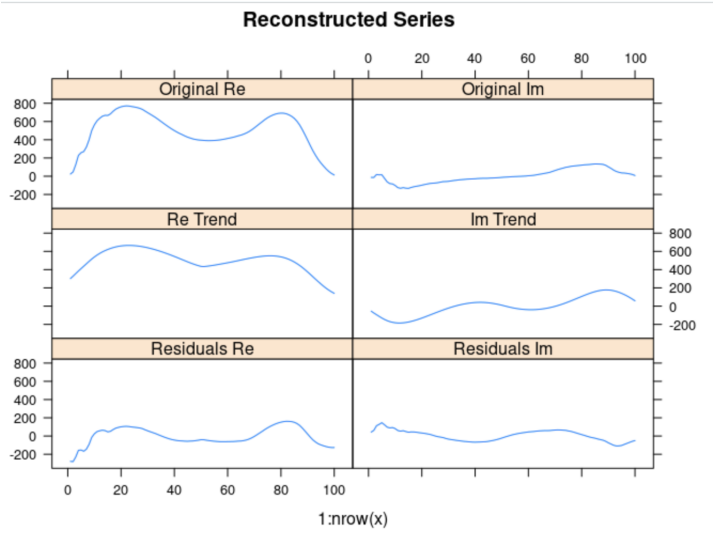


Figure 6 Example reconstructed series

Then, we want to have a clear pattern of the contribution of each eigenvector. We plot the top 8 eigenvectors and the associated reconstruction curves. The percentage associated with how much percentage of variance can be explained by this eigenvector. The first and second eigenvalues are the top two which contain the largest variance explained by the line. The blue line represents the real number, and the pink line represents the imaginary number. . We then did the same analysis on the whole data set, and we found that the mean variance which can be explained by the lines after reconstruction for in total of 15678 observations is 96.51%.

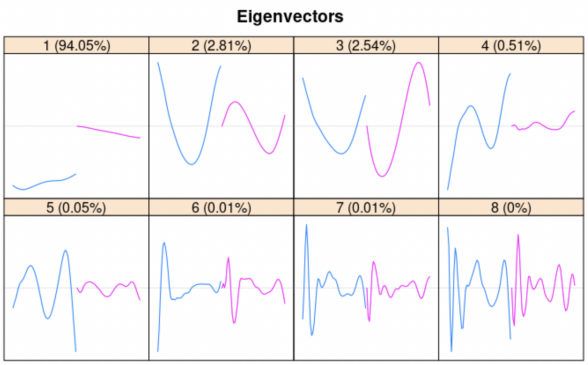


Figure 7 Top 8 contributed eigenvectors

Since MSSA is not a method of dimension reduction, we don't use time-series distance to calculate the distance among each observation. We get inspired by this idea and try to utilize them in the following part.

Then, we come to the second chapter of SSA. We combine FDA and SSA together. Recalled FDA is one type of smoothness method and SSA is a dimensional reduction method. We try to repeat the same procedure for three variables with the basis number 23 and group only by the first contribution as the trend. These plots are the original dataset with the first 100 observations. From the below graphs, we find that FDA keeps the shape of the original curves while SSA extract the trend of different curves.

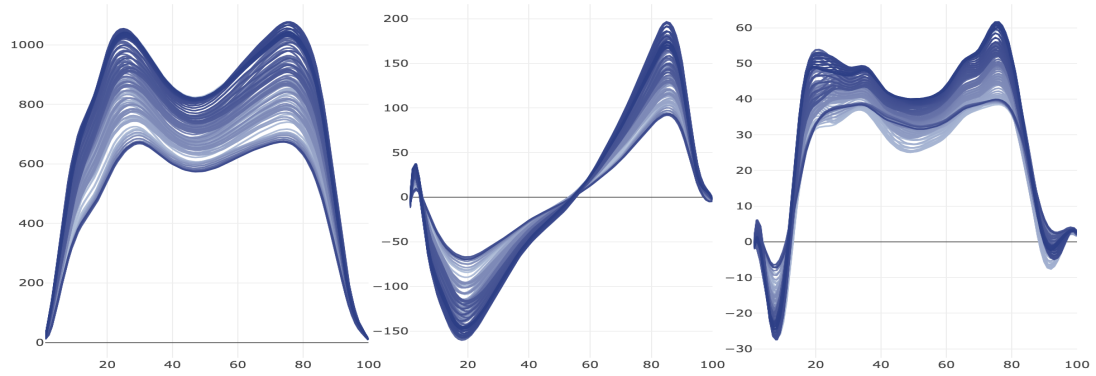


Figure 8 FSSA trends for the first 100 observations

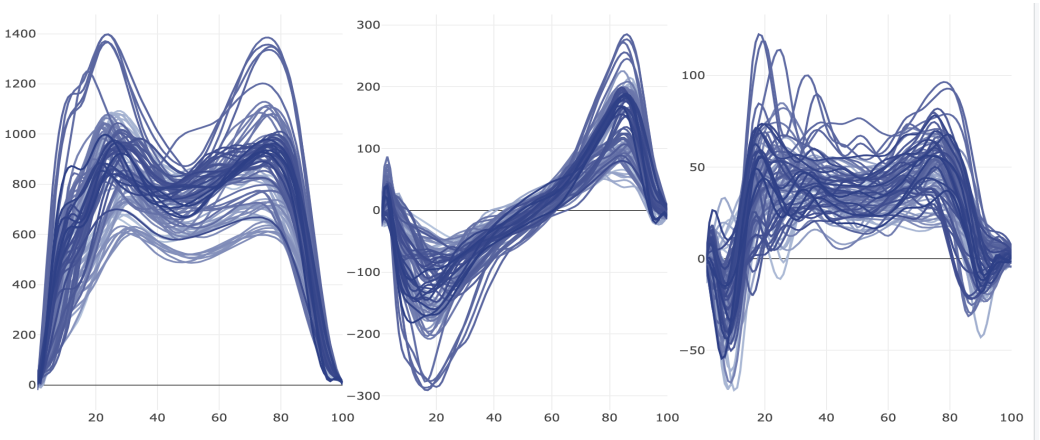


Figure 9 Original curve for the first 100 observations

Until now, SSA achieved a good result and fit. However, before performing the clustering analysis, we need to calculate distances for each observation in the data set. For FSSA, the calculator price will be high to calculate the time series distance, so we decided to try the other model to see whether it could give us a different perspective.

3. Auto-Encoder

Auto-Encoder is a type of unsupervised artificial neural network used to learn how to efficiently compress and encode unlabeled data, and then learn the reconstruction of data and dimensionality reduction via learning how to ignore the data noise (insignificant data).

The Auto-Encoder consists of 4 main parts: encoder, where the model learns how to do the input dimension reduction and compression; latent dimension, which is the lowest possible dimension containing the compressed representation of the input; decoder, where the model learns how to perform the reconstruction from the encoded to be close to the original input; reconstruction loss, which measures the performance of the decoder and the closeness of the output to the input.

Below is the basic architecture graph of the Auto-Encoder.

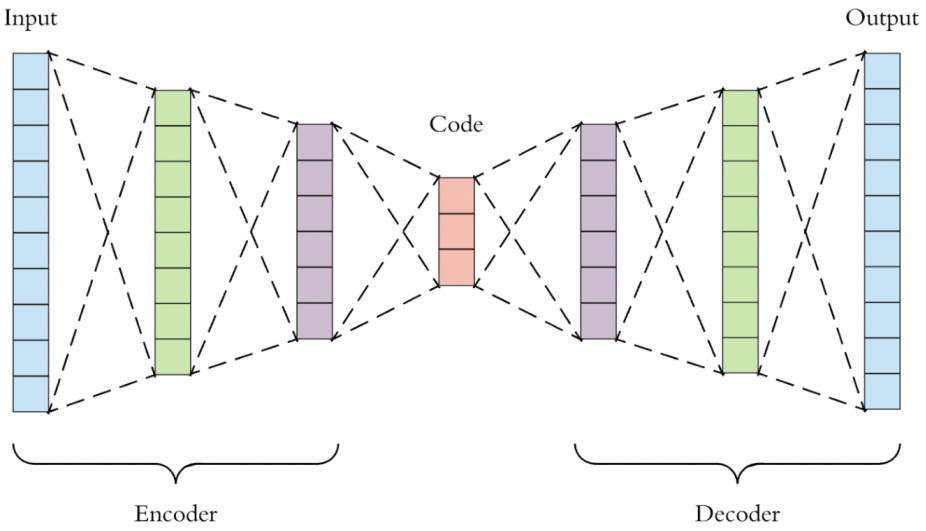


Figure 10 Auto-Encoder structure example

Here we built a tensor flow Keras reconstruction auto-encoder model with parameters that added the layer of the model. For our auto-encoder model, we select the first dense layer with 500 nodes and then decrease it gradually to latent dimension 10. The detail layers parameter shown below.

Layer (type)	Output Shape	Param #
dense_input (InputLayer)	[(None, 100)]	0
dense (Dense)	(None, 500)	50500
dense_1 (Dense)	(None, 250)	125250
dense_2 (Dense)	(None, 125)	31375
dense_3 (Dense)	(None, 50)	6300
dense_4 (Dense)	(None, 10)	510
sequential_1 (Sequential)	(None, 100)	214025

Total params: 427,960
Trainable params: 427,960
Non-trainable params: 0

Figure 11 Auto-Encoder model used

We conduct the same autoencoder to three different variables. Here we took the analysis on V_GRF_stance_N as the example, and below is the example reconstructed curve of the 500th observation:

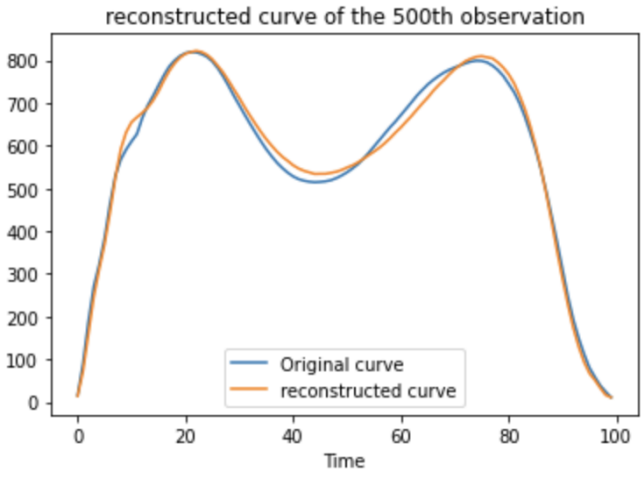


Figure 12 Reconstructed curve of the 500th observation

Below is the learning curve loss plot for V_GRF_stance, which is a plot of model learning performance over time. This shows that the model achieves a good fit in reconstructing the input, which holds steady throughout training. In order to avoid model overfitting, we stopped our training until the validation loss increased.

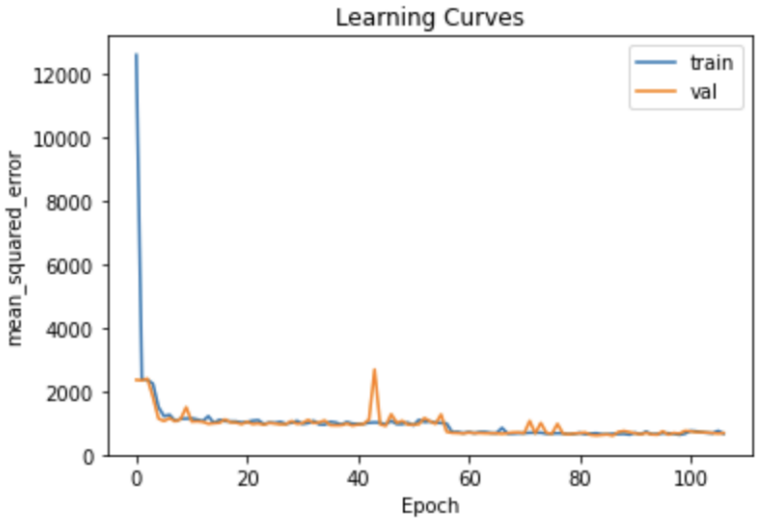


Figure 13 Learning curve for V_GRF_stance_N

Based on this auto-encoder training, we compressed the 100 points line into 10 points to get its compressed curve for the convenience of future cluster calculation and comparison.

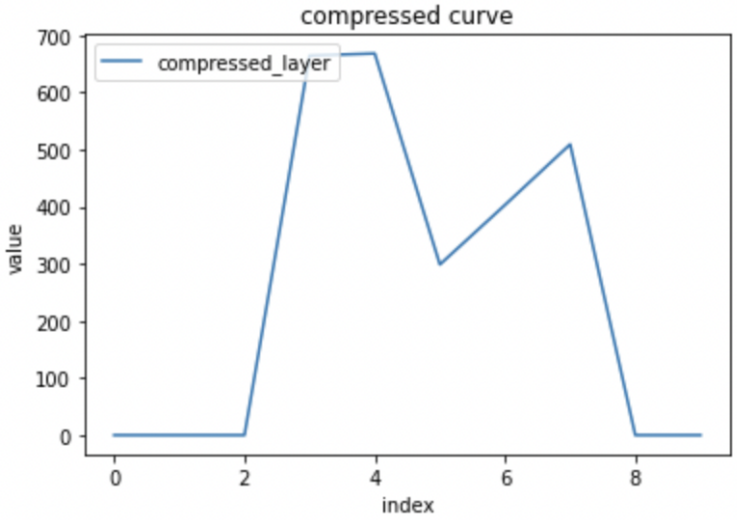


Figure 14 Compressed curve for V_GRF_stance_N

The last methodology that we use is Dynamic Time Warping (DTW). DTW is based on the curve shape information. We defined the window as 1 to exactly calculate the point-to-point Euclidean distance. The reason for the window length is that the latent dimension is already contained lots of information, if allow warping, we will compare different key components.

IV. Results

Based on the Auto-Encoder training and DTW distance, we used k-means to do the cluster analysis of these three data sets, and we clustered each of them into 4 groups. The reason we chose 4 groups is that our paper has true labels with 4 groups. In the clustering graphs below, we can see some obvious differences between each group for V_GRF_stance with autoencoder latent dim 10 and number of test observations 3140. The number of observations belonging to the first cluster is 1224, while the number for the second cluster is 488, the number for the third cluster is 403, and the number for the fourth cluster is 1025. Based on the brief recap of the paper, it seems that our cluster trend findings fit the actual groups, which correspond to the approximate percentage. Although we do not have the exact group belonging data, we believe that the client could apply this clustering.

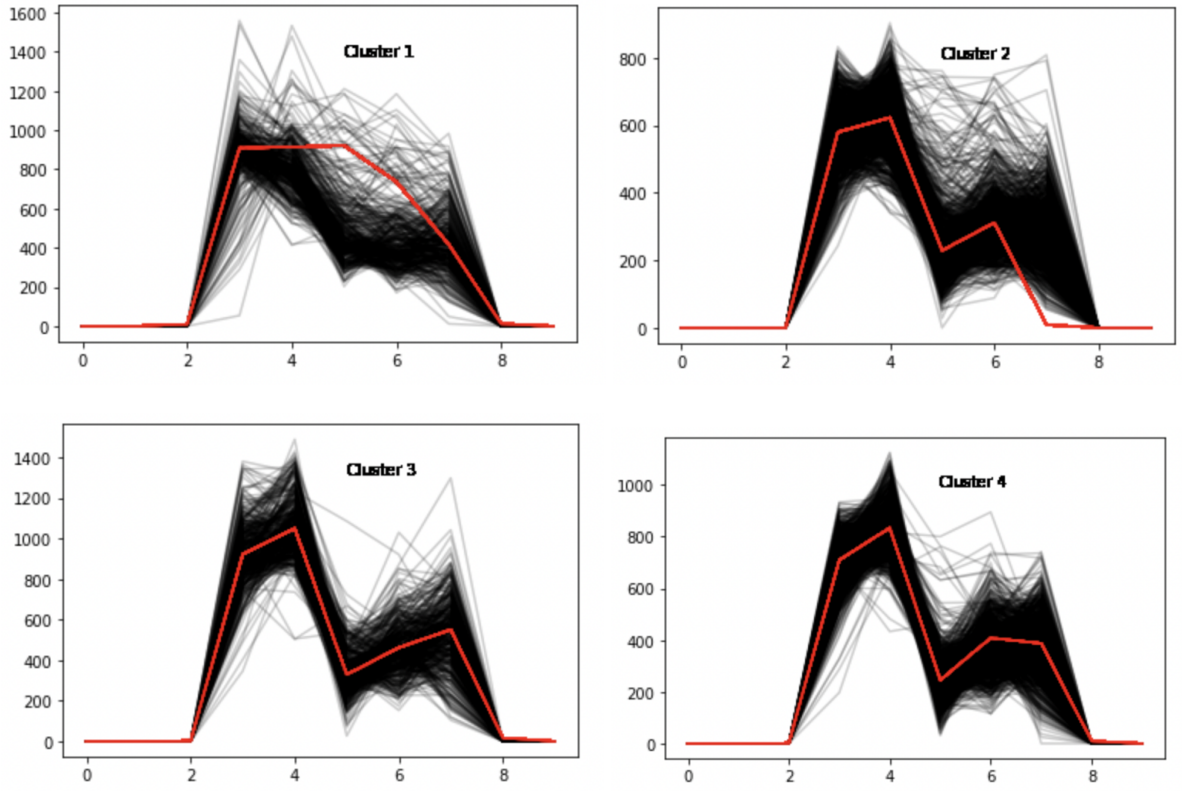


Figure 15 Clustering graph results for V_GRF_stance_N

V. Conclusion

In general, we assess different types of SSA, and an auto-encoder to show different aspects of dimension reduction. Although we don't have a method to approach the accuracy of our model, we still try to do as best as we can to cluster different groups.

For the next step, we should continue to work on our previous unfinished Variational auto-encoder to give more regulation methods to our naive models.

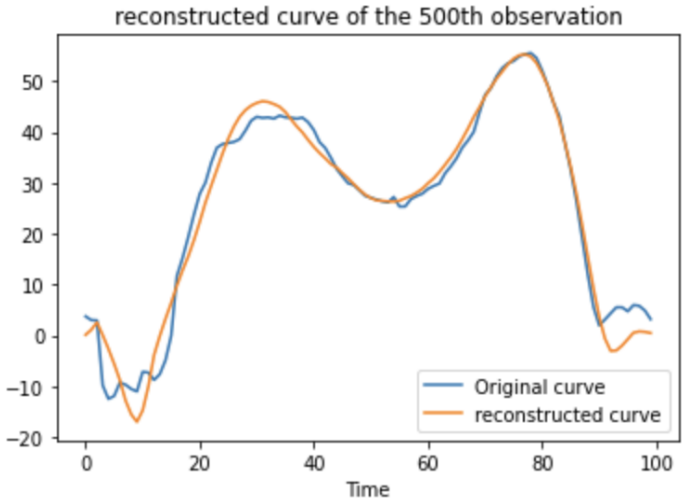
Works Cited

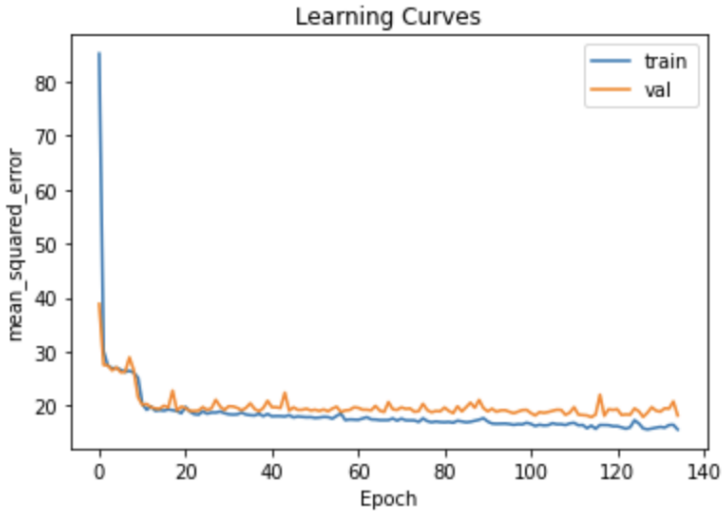
- Dertat, Arden. *Applied Deep Learning - Part 3: Autoencoders,* 3 October 2017, https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798
- Golyandina, Nina, et al. *Singular Spectrum Analysis with R,* Springer Berlin / Heidelberg, 25 June 2018.

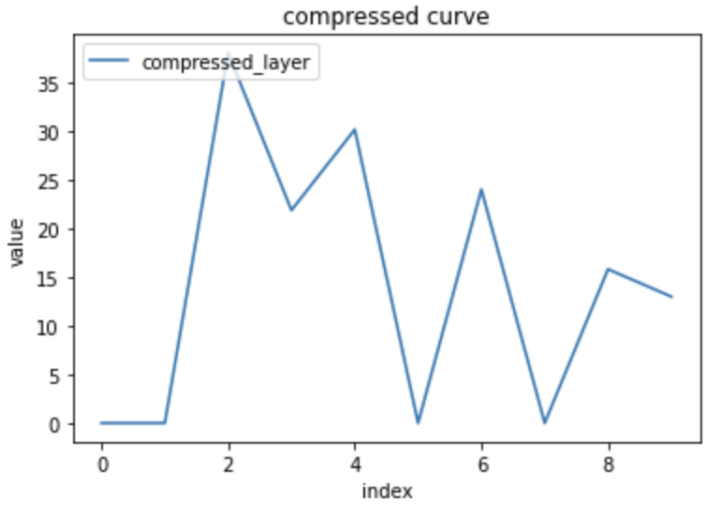
Appendix

- Auto-Encoder Fitting:

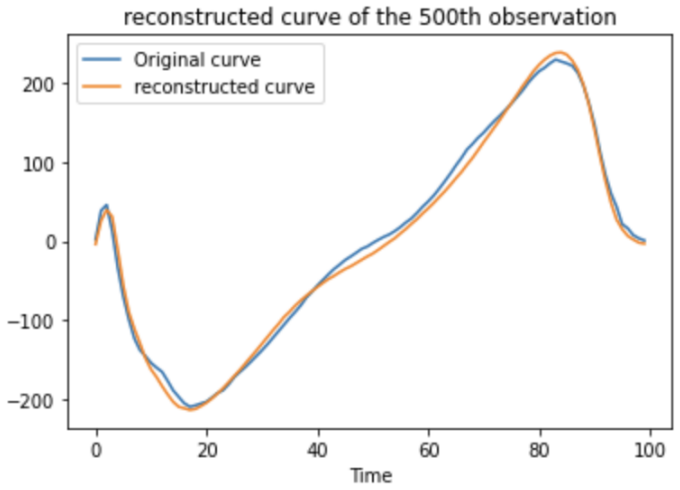
For ML_GRF_stance_N:

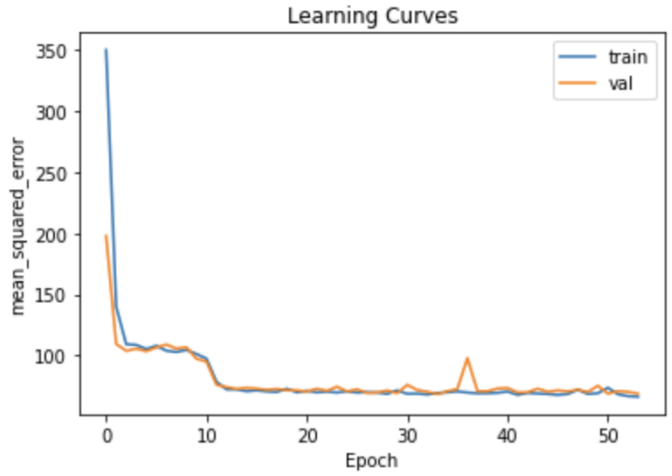


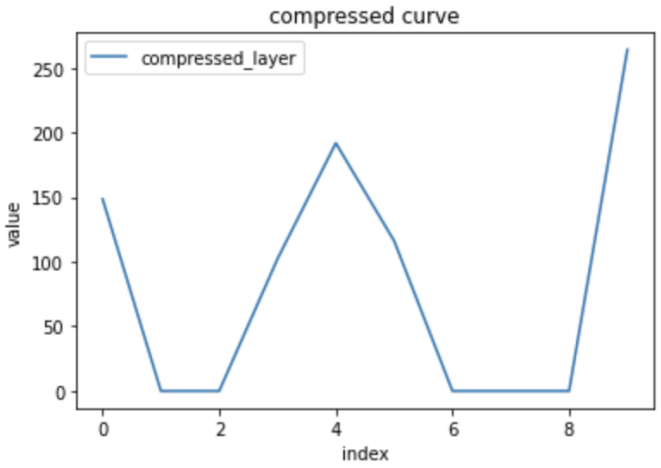




For AP_GRF_stance_N:

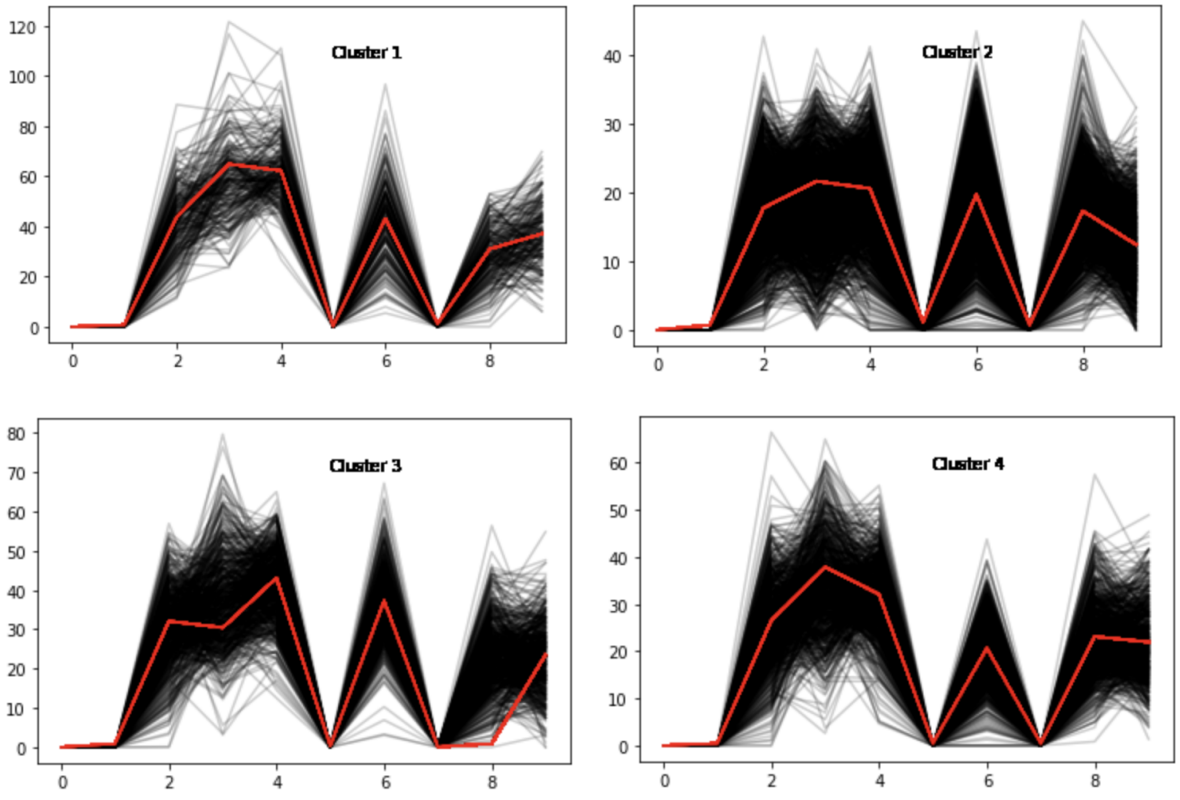




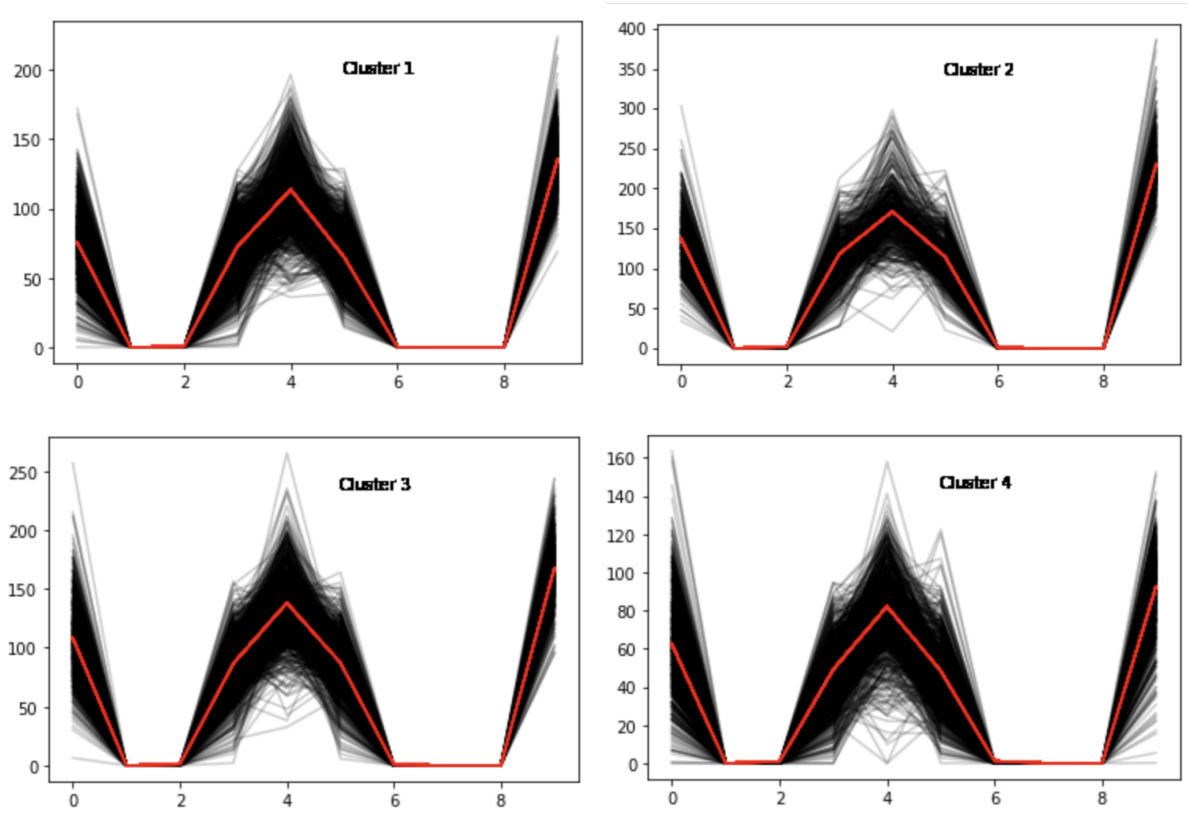


- **Cluster Results**

For ML_GRF_stance_N:



For AP_GRF_stance_N:


Cluster 1
Cluster 2
Cluster 3
Cluster 4