How to Use the Time Series Augmentation Codes

Pre-requests:

1. Users need to have access to Python 3.10.14 (TensorFlow 2.13.1 required) and Matlab R2023b.

The authors recommend running Python codes on Jupyter Notebook. Running the codes on different versions of Python, TensorFlow, or Matlab may result in compatibility issues. However, most compatibility issues can be resolved by following the suggestions in the error messages.

2. Users need to have basic knowledge of machine learning or deep learning to tune the hyperparameters of convolutional variable autoencoder (CVAE).

Step-by-step instruction:

Step 1: Prepare the time series data for augmentation as a MATLAB dataset.

Save the time series data to be augmented as a $N_t \times 1$ vector of cells in Matlab and named it as "TimeSeries.mat", where N_t is the number of time series. Each cell includes a $N_s \times D$ matrix, where N_s is the number of time steps, D is the number of dimensions.

In the example folder, the "TimeSeries.mat" includes a 94×1 vector of cells. Each cell includes a 800×2 matrix.

Step 2: Run Python code "P1_CVAE.ipynb".

This step is used to train the CVAE, which may take a couple of hours, depending on the number and size of time series to be augmented and hyperparameters of CVAE (e.g., number of epochs). Note that the hyperparameters of CVAE are defined by the user.

Three folders (named with "Model", "Figure", and "Data") are automatically generated when "P1_CVAE.ipynb" is finished. The "Model" folder includes the checkpoint. The "Figure" folder includes the CVAE architecture. The "Data" folder includes the reconstructed time series, latent features, mean and variance of latent features, loss, and running time.

Step 3: Run Matlab codes “P2_Plot_Comp.m” and/or “P2_Plot_Comp_2D.m”.

These two Matlab codes are used to generate the figures to compare the original and reconstructed time series. Note that the parameters “Lat” and “Epoch” on lines 11 and 14 in “P2_Plot_Comp.m” and “P2_Plot_Comp_2D.m”. need to be consistent with the ones in “P1_CVAE.ipynb”.

In “P2_Plot_Comp.m”, for each dimension of one original time series, the original and reconstructed time series are plotted versus time steps. The user can change the parameter “ID” on line 34 to plot the time series of interest or write "ID" as a vector to plot comparisons for multiple time series.

In “P2_Plot_Comp_2D.m”, the plot is generated using two dimensions of the time series (2D). The user can select which two dimensions to plot on line 33. Similarly, the parameter “ID” on line 37 can be changed, as previously mentioned.

Step 4: Run Matlab code “P3_Aug_Lat.m”.

This Matlab code is used to generate latent features based on the mean and variance learned by the trained CVAE. Note that the parameters “Lat” and “Epoch” on lines 11 and 14 in “P3_Aug_Lat.m” need to be consistent with the ones in “P1_CVAE.ipynb”.

The “Num_Aug” is a user-defined parameter, which is the number of augmented time series generated from a single original time series.

Step 5: Run Python code “P4_CVAE_Aug.ipynb”.

This Python code loads the trained CVAE in step 1 to augment the time series. Note that all the hyperparameters of CVAE in “P4_CVAE_Aug.ipynb” need to be consistent with the ones in “P1_CVAE.ipynb”.

One folder named with “Data” and “Aug” is automatically generated when “P4_CVAE_Aug.ipynb” is finished. It includes the augmented time series.

Step 6: Run Matlab code “P5_Aug_Data.m”.

This Matlab code is used to post-process the time series data augmented from Step 5. Note that the parameters “Lat” and “Epoch” on lines 11 and 14 in “P5_Aug_Data.m” need to be consistent with the ones in “P1_CVAE.ipynb”.

The augmented time series data are saved as Matlab dataset “Aug_TimeSeries.mat”, which has the same structure as the original time series data “TimeSeries.mat”.

Step 7: Run Matlab code “P6_Plot_Aug.m” and/or “P6_Plot_Aug_2D.m”.

These two Matlab codes are used to generate the figures to compare the original and augmented time series.

Similar to Step 3, “P6_Plot_Aug.m” plots the original and augmented time series versus time steps. The user can change the parameter “ID” on line 22 to plot the time series of interest or write "ID" as a vector to plot comparisons for multiple time series.

In “P6_Plot_Aug_2D.m”, the plot is generated using two dimensions of the time series (2D). The user can select which two dimensions to plot on line 21. Similarly, the parameter “ID” on line 26 can be changed, as previously mentioned.