Signal distance measurement using auto-encoder latent space

AUTHORS

Marius Dragic
marius.dragic@student-cs.fr

Due : February 20th 2025

Table des matières

1 Abstract

Measuring the distance between two signals is a key issue in many scientific and industrial fields. Signals, whether acoustic, electrical, biological or other, are often complex representations of information derived from physical phenomena. Calculating a reliable distance between these signals enables us to solve various problems, such as pattern recognition, classification, clustering and anomaly detection. In this context, the aim is to develop an innovative method using an auto-encoder. This will enable the latent space to be structured in such a way that it respects specific properties for calculating distances. The distinguishing feature of this approach is its ability to manipulate signals in a transformed space where notions of proximity and similarity are optimized.

The proposed research paper is therefore an innovative way of measuring the distances between signals belonging to the same class. The following report focuses on several aspects, such as the definition of the class of signals we are going to study, an in-depth study of the latent space into which the signals are projected and the study of their reconstruction, which in particular enables us to detect anomalies and ensure that the autoencoder has learned a good underlying representation of the signals.

2 State of the art

2.1 Field of application

Today, the measurement of distance between signals has concrete and useful applications in many fields, despite the fact that such techniques are not yet perfectly developed :

- **Biomedical :** for the analysis of ECG, EEG, EMG signals, to detect anomalies and pathologies
- **Computer vision :** to measure the difference between facial features and perform facial recognition.
- **Finance and economics :** to analyze multivariate time series and evaluate similarities between market trends.
- **Clustering of all kinds :** to define a new measure of distance between temporal signals and to define new clustering methods adapted to this new measure.

Both in terms of domain and concrete application, distance measurement between signals is mainly used for anomaly measurement or pattern recognition, but the techniques developed today are not yet very reliable and robust, as the following section will show.

2.2 Existing methods for calculating distances between signals

There are more common ways to measure the distance between signals, the classical methods used are the following :

- **Euclidean distance :** this distance measurement is the most intuitive and measures the point-by-point distance between 2 signals, but it is not relevant because it is extremely sensitive to noise and phase shift.
- **Crossed Fourier Transform :** this method allows you to project signals into the frequency domain and compare their spectral characteristics. Signals with different frequencies/amplitudes will therefore be far apart. Nevertheless, this method is sensitive to noise, contraction/dilation, and insensitive to signal time shifts.

- **Dynamic Time Warping (DTW) :** This method is probably the most widely used today, as it aligns 2 time series to minimize their difference. Although this method is robust to contractions/dilatations, it is extremely costly in terms of computation (complexity in $O(n^2)$).

All these methods, although used in many fields of application, have drawbacks and a lack of robustness that we will try to overcome with the method we propose in the rest of the paper.

3 Proposed method

As stated above, our aim is to implement a method that is **robust to signal contractions, dilations, phase shifts and noise**. What's more, the method we propose in this paper allows us to construct an arbitrary distance measure, i.e. the measure will take into account only the characteristics of the signals we're interested in. Thus, the author decides to construct the measure of the signal of interest and then specifically train the auto encoder to construct a latent space that preserves this notion of distance.

3.1 Projection into latent space

The central idea is to use a convolutionnal autoencoder (CAE) to project signals into a latent space, and then measure the distance between these signals in this compressed space.

Traditional autoencoders are made up of two parts : an encoder and a decoder, which simultaneously learn to project the signal into a reduced-dimensional space known as the latent space, and to reconstruct it. It's sometimes difficult to understand how this kind of space is structured, which is why we're going to add a constraint to our autoencoder that will force it to preserve a notion of distance between the input space and the latent space, while minimizing the reconstruction error. We also decided to add a convolutional architecture to our autoencoder to make it even more efficient. Convolutional layers are particularly effective for detecting local patterns (such as trends, cycles or rapid changes) in time series.

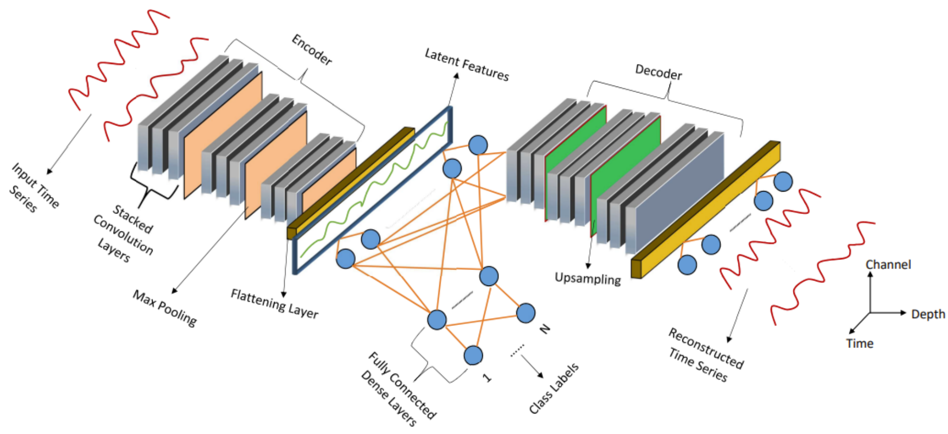

FIGURE 1 – Illustration of the Autoencoder architecture.

Furthermore, as shown in the diagram, it is then possible to connect another neural network to the latent space thus constructed, in order to perform all kinds of classification tasks in addition to measuring the distance between signals. The autoencoder will thus have been trained to structure a compressed latent space rich in information that can be exploited by clustering or supervised classification algorithms.

3.2 Defining the distance between signals

The initial challenge is defining a distance between two signals. It is necessary to emphasize that comparison of signals makes sense only in case when they are of the same nature. There is no sense in referring to distance from MRI signal to ECG signal. Therefore the approach suggested below compares the signals only of the same kind.

The CAE encoder maps every temporal signal S_i to a latent space, which is referred to as Z. The projection of S_i is a vector $z_i \in \mathbb{R}^d$, where d represent the dimension of the latent space. The latent vector in this case encodes the underlying nature of the signal while eliminating redundancy and retaining attributes relevant to the similarity metric. Continuing the above definition, we could define a distance between two signals S_i, S_j in latent space Z hrough some appropriate metric. We denote latent representations of signals S_i and S_j as latent representations z_i and z_j respectively. In this space we define a distance $d(z_i, z_j)$. There are 2 main possible choices depending on the desired properties :

Euclidean distance :

$$d(z_i, z_j) = \sqrt{\sum_{k=1}^{d} (z_{i,k} - z_{j,k})^2} = \|\mathbf{z}_i - \mathbf{z}_j\|_2 \tag{1}$$

This metric is simple, quick to calculate and is the most intuitive, but may lack robustness to non-linearities and complex signal distortions.

Cosine distance :

$$d(z_i, z_j) = 1 - \frac{\langle z_i, z_j \rangle}{\|z_i| \cdot \|z_j\|} \tag{2}$$

where $\langle z_i, z_j \rangle$ is the scalar product between z_i and z_j, and $\|z_i\|$ is the norm of z_i. This metric captures the alignment of signals in latent space, independent of their amplitude. This is the distance measure used in NLP models to measure the similarity between 2 vectorized words.

3.3 Synthetic signal dataset

To begin this project, we make the simplifying assumption that the signals of interest to us are defined as a sum of a 3rd-order polynomial and 8 cosine terms :

$$s(t) = \sum_{i=0}^{N_p} a_i t^i + \sum_{i=1}^{N_c} b_i \cos(w_i t) \tag{3}$$

Where :

- $a_i \in \mathbb{R}$: Coefficients of the polynomial
- $b_i \in \mathbb{R}^+$: Cosine amplitudes
- $w_i \in \mathbb{R}^+$: Cosine frequencies
- $\phi_i \in [0, 2\pi]$: cosine phase

For $N_p = np$ and $N_c = nc$, the signal without noise is thus entirely defined by the parameter vector :

$$\theta = (a_0, a_1, \ldots, a_{np-1}, b_1, b_2, \ldots, b_{nc}, w_1, w_2, \ldots, w_{nc}, \phi_1, \phi_2, \ldots, \phi_{nc}) \in \mathbb{R}^{np+3nc}$$

Parameter	Description
$F_e = 1024$	Sampling frequency
$n_{\textbf{size}} = 1024$	Size of the signals
$a_{\textbf{range}} = [-7, 7]$	Range of the coefficients a
$b_{\textbf{range}} = [0, 5]$	Range of the cosine amplitude b
$w_{\textbf{range}} = [0.5, 7]$	Range of angular frequencies w
$\phi_{\textbf{range}} = [0, 2\pi]$	Range of the phases ϕ
$n_{\textbf{poly}} = 3$	Number of polynomials
$n_{\textbf{cos}} = 8$	Number of cosines

TABLE 1 – Parameters of the synthetic generated signals

Here is an example of generated signals with the following parameters :

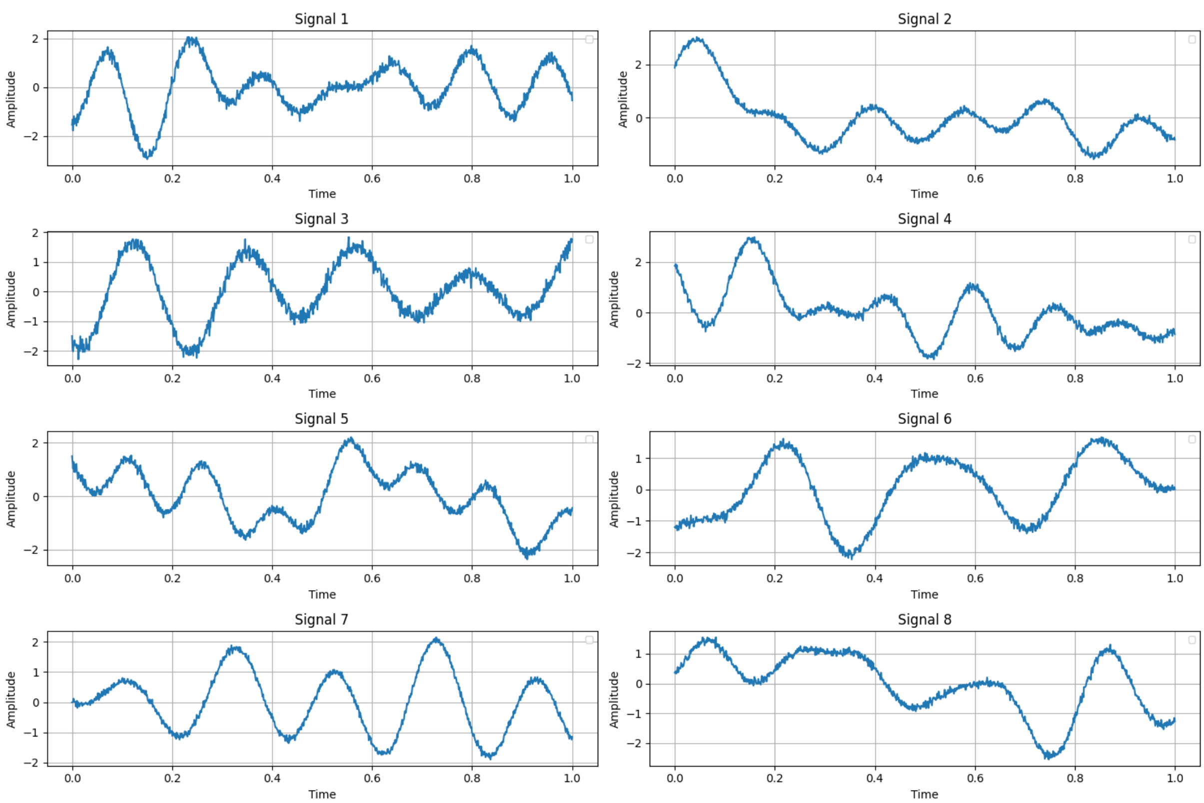

FIGURE 2 – Synthetic generation of signals

4 Modeling an auto-encoder preserving distance metrics

4.1 Auto-encoder architecture

An **autoencoder** is used to learn the $\mathcal{Z}$ latent space :

— The **encoder** $f_{\text{enc}}(s(t)) = \mathbf{z} \in \mathcal{Z}$ projects the signal into latent space.

— The **decoder** $f_{\text{dec}}(\mathbf{z}) = \hat{s}(t)$ reconstructs the original signal from $\mathbf{z}$.

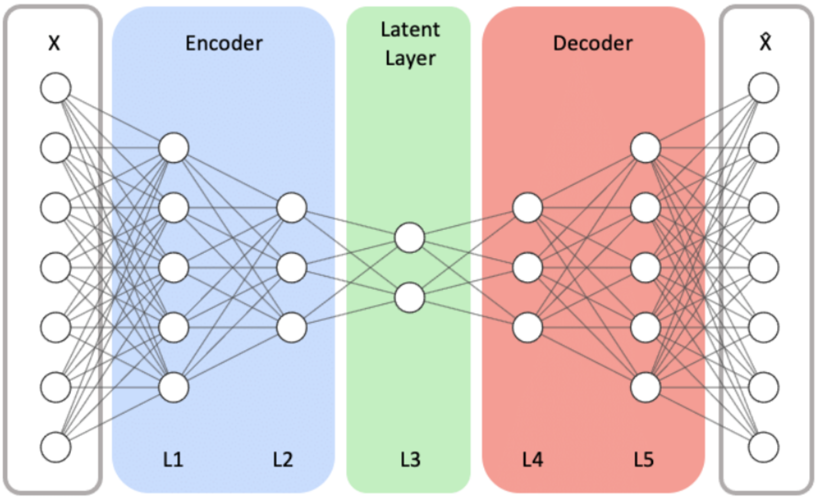

FIGURE 3 – Auto-encoder architecture visualisation

Layer	**Details**	**Output Shape**	**Description**
		Encoder	
Conv1	1 → 32, kernel=5, stride=2, padding=2	input_length/2	Extract features
Conv2	32 → 64, kernel=5, stride=2, padding=2	input_length/4	Decrease size, increase depth
Conv3	64 → 128, kernel=5, stride=2, padding=2	input_length/8	More features
Conv4	128 → 128, kernel=5, stride=2, padding=2	input_length/16	Deeper representation
Conv5	128 → 256, kernel=3, stride=2, padding=1	input_length/32	Final feature map
Flatten	-	256 × (input_length/32)	Flatten feature maps
FC Enc	Linear to latent space	latent_dim	Encodes latent representation
		Decoder	
FC Dec	Linear from latent space	256 × (input_length/32)	Map latent back to feature map
Deconv5	256 → 128, kernel=3, stride=2, padding=1, output_padding=1	input_length/16	Expand features
Deconv4	128 → 128, kernel=5, stride=2, padding=2, output_padding=1	input_length/8	Reconstruct step-by-step
Deconv3	128 → 64, kernel=5, stride=2, padding=2, output_padding=1	input_length/4	Continue expanding
Deconv2	64 → 32, kernel=5, stride=2, padding=2, output_padding=1	input_length/2	Upscale features
Deconv1	32 → 1, kernel=5, stride=2, padding=2, output_padding=1	input_length	Reconstruct signal

TABLE 2 – Summary of the convolutional autoencoder architecture.

Intuitively, the dimension of the latent space should be the same as the dimension of the parameter vector, but to ensure good signal reconstruction as well as efficient distance preservation, we choose a latent space of dimension $d_{latent} = 256$.

4.2 Auto-encoder's Loss

The autoencoder's loss function is a combination of two components : reconstruction loss and distance loss. The reconstruction loss ensures that the decoder can accurately reproduce the input signal from the latent representation. The distance loss ensures that the distances in the latent space reflect the distances between the parameters of the original signals. Mathematically, this is represented as $\mathcal{L} = \alpha\mathcal{L}_{\text{reconstruction}} + \beta\mathcal{L}_{\text{distance}}$ where α and β are hyperparameters. This design allows the latent space to preserve key signal characteristics related to distance.

Auto-encoder's Loss function

$$\mathcal{L} = \alpha\mathcal{L}_{\text{reconstruction}} + \beta\mathcal{L}_{\text{distance}} \tag{4}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\|s_i(t) - \hat{s}_i(t)\|^2 + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\|\theta_i - \theta_j\|_2 - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2 \tag{5}$$

4.3 Advantages of an auto-encoder

Instead of training an auto-encoder, it would have been possible to train a model to decode the parameters of interest in a signal. Nevertheless, the properties of an auto-encoder are particularly interesting in the case of this project for the following reasons :

- The latent space of an auto-encoder is rich in information and can implicitly model more than just a distance measure. The latent space can encode complex time series characteristics such as relations between harmonics, local peaks and similarities in patterns. It is possible to exploit the richness of the latent space by performing unsupervised clustering with this new notion of defined distance, or supervised classification with any other type of model from the latent space.
- Furthermore, if latent space is dense and well-trained, it can be used for signal generation and interpolation. Consider two signals S_i and S_j : they can be interpolated by taking latent vectors that lie between the latent representations of these 2 signals. Such a model therefore possesses interesting generative properties.
- Auto-encoders are often used for unsupervised anomaly detection. A signal can be considered abnormal if its reconstruction error exceeds a certain threshold, which cannot be done with a neural network that decodes parameters.

For all the reasons set out above, we have chosen to address the problem of measuring distance between time series using an auto-encoder.

4.4 Auto-encoder training

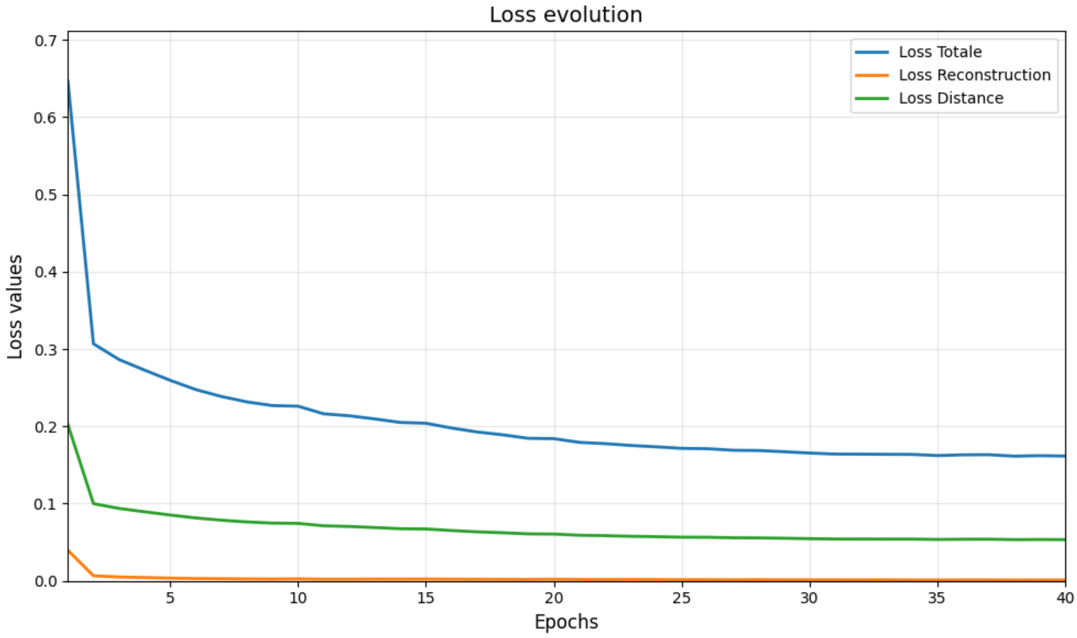

FIGURE 4 – Training loss evolution

The convolutional auto-encoder presented above is trained over 40 epochs on 20000 artificial signal, with a learning rate $lr = 0.001$ and an Adam optimizer. To improve learning, the signals are normalized with a MinMaxScaler, which transforms them to an image interval of [-1, 1]. The input parameters with respect to which we wish to define the notion of distance are normalized to [0,1] so that each class of parameters has the same weight in learning to preserve distances, so we have the following normalization on the parameters :

$$\begin{cases} a_{\text{norm}} = \dfrac{a - a_{\text{min}}}{a_{\text{max}} - a_{\text{min}}} \\ b_{\text{norm}} = \dfrac{b - b_{\text{min}}}{b_{\text{max}} - b_{\text{min}}} \\ w_{\text{norm}} = \dfrac{w - w_{\text{min}}}{w_{\text{max}} - w_{\text{min}}} \\ \phi_{\text{norm}} = \dfrac{\phi - \phi_{\text{min}}}{\phi_{\text{max}} - \phi_{\text{min}}} \end{cases}$$

After learning, we finally obtain the following results :

Epoch	Total Loss	Recon	Dist
40/40	0.1615	0.0012	0.0534

TABLE 3 – Training results at epoch 40

5 Results

5.1 Signal distance measurement

Three distance measurements were carried out, and 2 measurement methods are used, the Euclidean distance between latent vectors and the cosine distance, which measures perpendicularity.

- Distance between two sines phase-shifted by $\frac{\pi}{2}$.
- Signals generated very close together with just a slight phase shift, but very noisy in order to see if the noise has an influence on the measurement.
- Signals whose parameters in the starting space are very far apart.

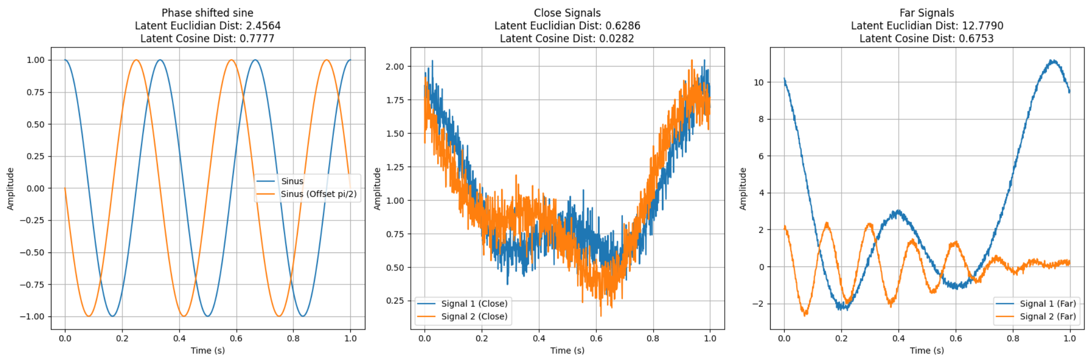

FIGURE 5 – Signals distance measurement

The distance measurement seems rather coherent, with signals that are visually similar (respectively far) being close in terms of distance (respectively far). What's more, adding noise to nearby signals doesn't seem to interfere with the measurement, since they seem to remain relatively close in terms of euclidean and cosine measurements. It's also interesting to note that the 2 sines out of phase by $\frac{\pi}{2}$ are further apart in terms of cosine distance than the 2 signals generated deliberately far apart. This may seem surprising, but it should be borne in mind that cosine distance measures perpendicularity, and in the eyes of the usual scalar product $\sin(x)$ and $\sin(x + \frac{\pi}{2})$ are orthogonal. So a scalar product of 0.78, as shown on the graph, seems rather appropriate to the situation.

To ensure that the notion of distance is consistent with all the parameters, we generate a reference signal at random, then vary its parameters independently of each other and plot the evolution of the distance with respect to the reference signal as shown in the graph below :

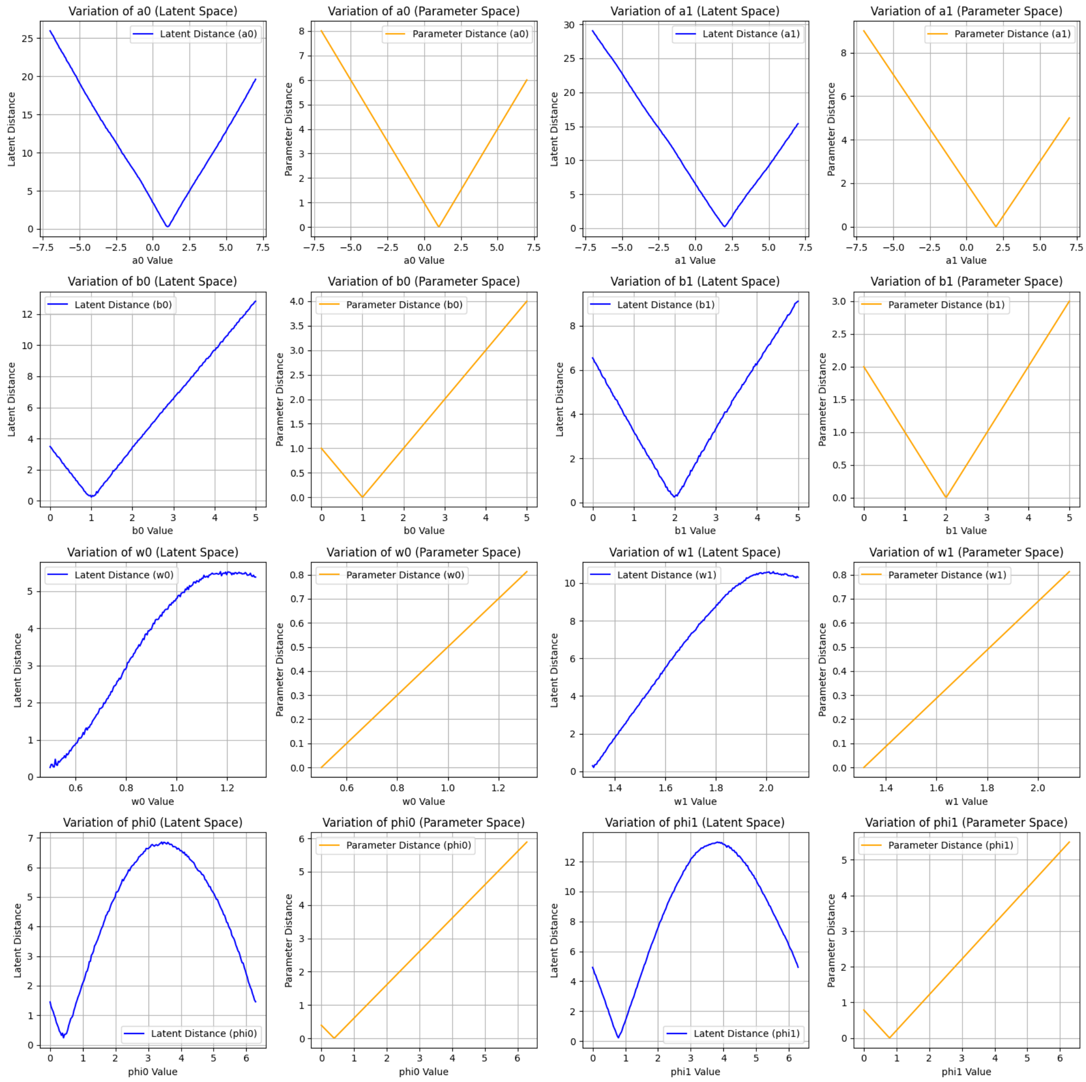

FIGURE 6 – Distance to a reference signal by varying the basic parameters. In orange, the evolution of the distance to a parameter in input space ; in blue, the evolution of the distance between the reference signal and the modified one in latent space.

The linearity relationships of the parameters a_i and b_i are well respected, and on the graph the distance to the reference signal evolves linearly when only one of these parameters is varied. The tricky part is encoding phase and frequency, since these parameters are not consistent with a notion of linear distance. Nevertheless, we try to encode the spectral distance $d_w = |w_i - w_j|$, which is linear (see orange graph). The evolution of the distance by varying the frequency parameters (cf. blue graph) is almost linear, which is satisfaying since we wanted the auto-encoder to be able to model the spectral distance from its latent space. Moreover, when the phase is varied, the distance between the reference signal takes the form of an absolute value of sine(x), which is consistent with the fact that once the phase has covered 2pi, we fall back to a distance of 0.

Conclusion : Distance measurement

From the graphs above, we can conclude that the auto-encoder has succeeded in encoding the notion of distance in its latent space. Nevertheless, frequency distances should be monitored when generalizing to other signals, as this distance seems more difficult to encode due to non-linearities.

5.2 Signal reconstruction

As shown in the following figure, the auto-encoder manages to reconstruct the noisy signal perfectly. This shows that the auto-encoder succeeds in projecting the signals into its latent space of dimension 256, i.e. 4 times smaller. It would be entirely possible to further reduce the dimensions of the latent space for the sake of data compression, but as dimensionality reduction is not the aim of the project, we prefer to favor a good reconstruction rather than a low-dimensional latent space, so as to be able to carry out anomaly detection using this convolutional auto-encoder.

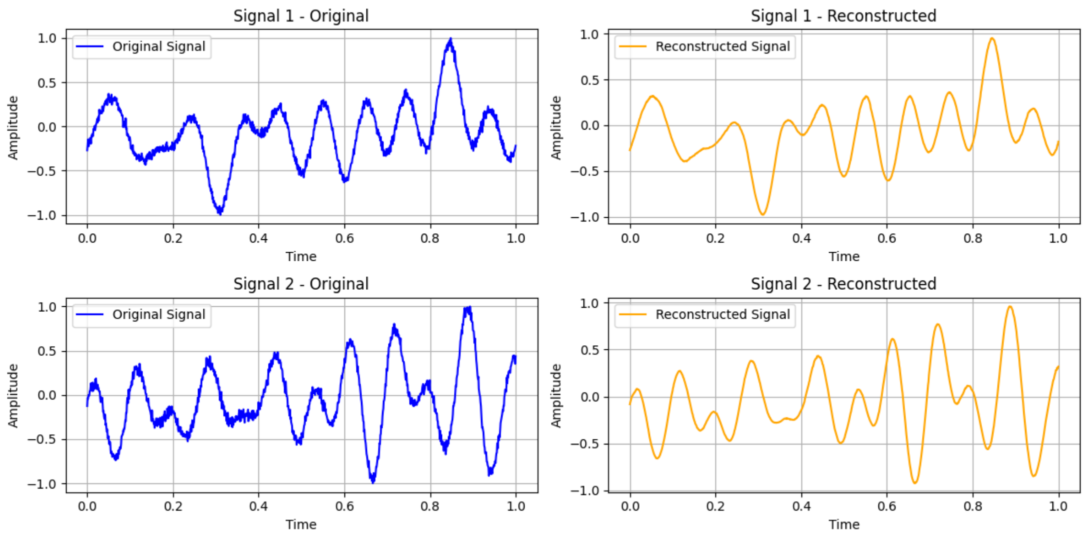

FIGURE 7 – Signals reconstruction by the convolutionnal auto-encoder

It is also interesting to note the denoising properties of such a model, which succeeds in extracting the main features of the signal, and, being unable to model the noise, reconstructs it in a completely denoised way.

The following graph shows the histogram of reconstruction errors between a traditional 6-layer auto-encoder and our convolutional auto-encoder. The difference in performance between the 2 types of model is striking, with the average reconstruction error divided by $\frac{36}{0.008} = 4500$. This illustrates just how much better convolutional layers are at capturing local patterns in noisy signals than classical auto-encoders.

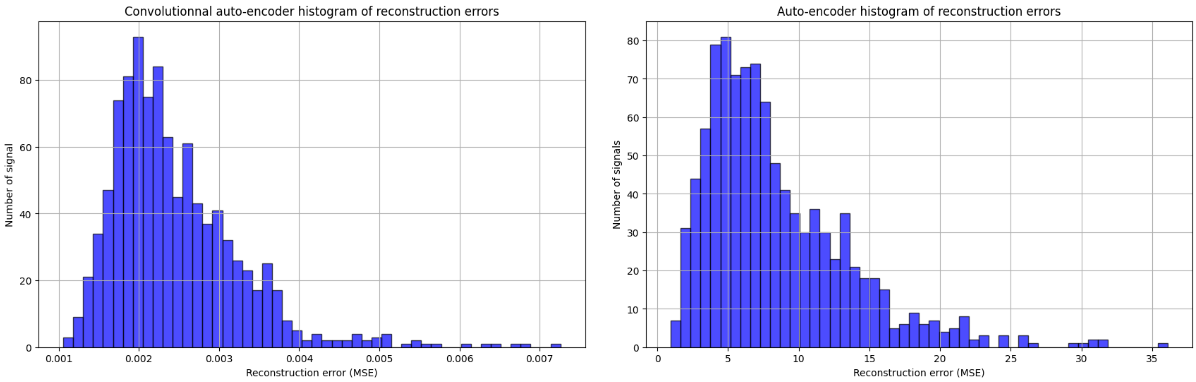

FIGURE 8 – Comparison of the two methods reconstruction errors

Conclusion : Signal reconstruction

Signal reconstruction is also very satisfaying, with the addition of convolutional layers that capture local signal patterns and dependencies. The distribution of reconstruction errors appears to follow a fish law, making it easy to detect abnormal signals by thresholding on rare anomalies at 5%, for example.

5.3 Parameter trajectory in latent space

To better understand the structure of latent space, we generate a reference signal with fixed parameters. The aim is to visualize how the latent space changes when one parameter varies independently of the others. To do this, we perform a PCA on the 2 main components of the latent space :

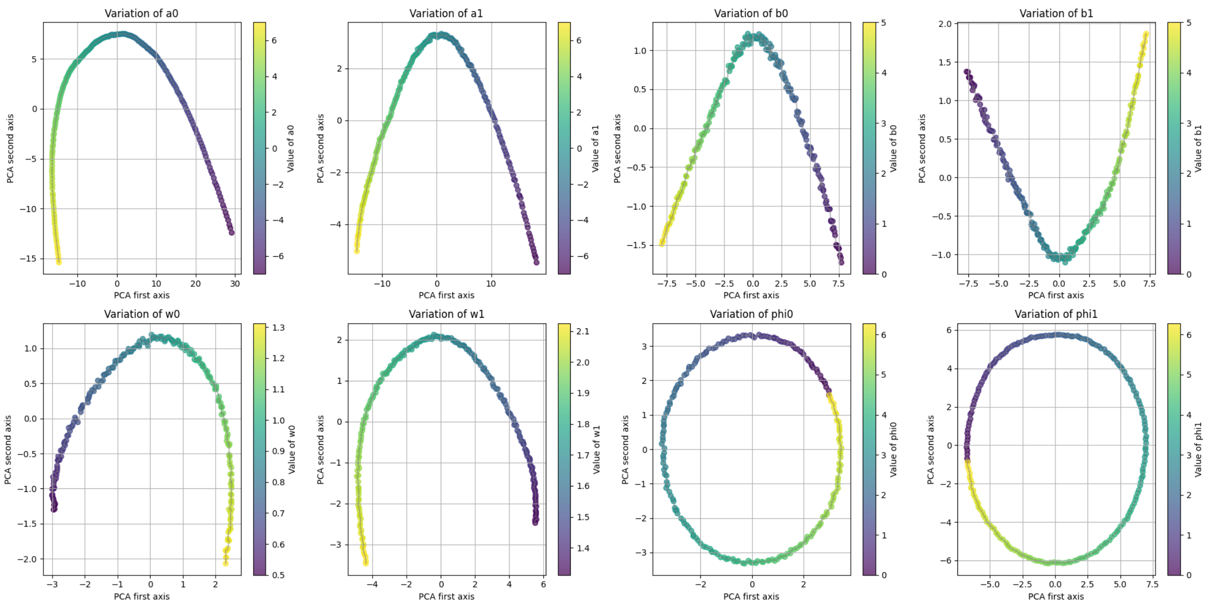

FIGURE 9 – Trajectory of parameters of interest in latent space

Clearly, the convolutional auto-encoder has learned to structure latent space. The arched and parabolic trajectories of the parameters a_i, b_i and w_i seem to be consistent with the notion of distance, as it has been proven part 5.1. Moreover, the circular trajectory when the phase varies from 0 to 2π is the expected one, ensuring that 2 sines out of phase by 2π have the same distance. So the loss that forces the auto-encoder to preserve the distance of the input parameters in the latent space has enabled the auto-encoder to structure its latent space coherently.

5.4 Cluster visualization in latent space

One of the interesting properties to exploit, as mentioned in 4.3, is clustering from latent space. To illustrate this property, 3 different classes of signals have been generated :

Cluster	a range	b range	w range	ϕ range
Rouge	$[-7, -4]$	$[0, 1]$	$[0.5, 2]$	$[0, 1]$
Vert	$[-3, 3]$	$[2, 3]$	$[3, 5]$	$[2, 3]$
Bleu	$[4, 7]$	$[4, 5]$	$[5, 7]$	$[4, 6]$

Table 4 – Parameters for clusters generated in latent space.

These 3 synthetically generated signals classes are then projected into latent space and visualized on the first 2 PCA components.

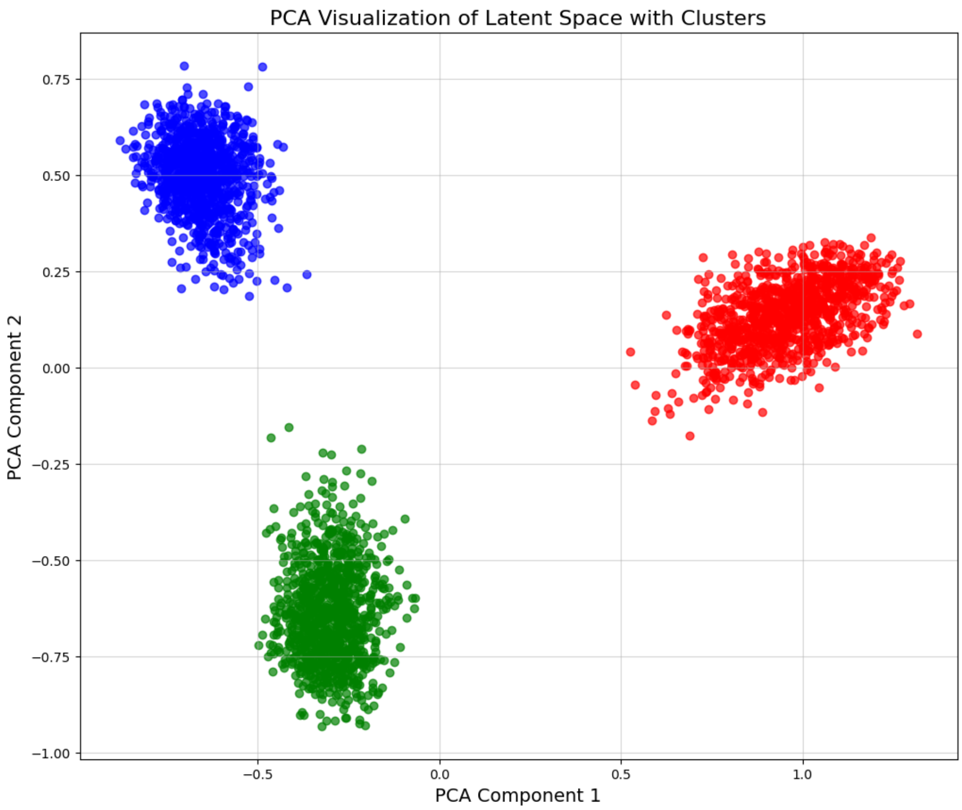

Figure 10 – Clusters of differents classes of signals

Unsurprisingly, the clusters are easily identifiable, and although we're in a trivial case with classes of signals whose parameters don't intersect, this illustrates the time series clustering possibilities available to the user of such a model, even in much more complex cases.

5.5 Properties of latent space

To better understand the properties of latent space, it is interesting to plot the distribution of the mean and variance of latent vector coordinates :

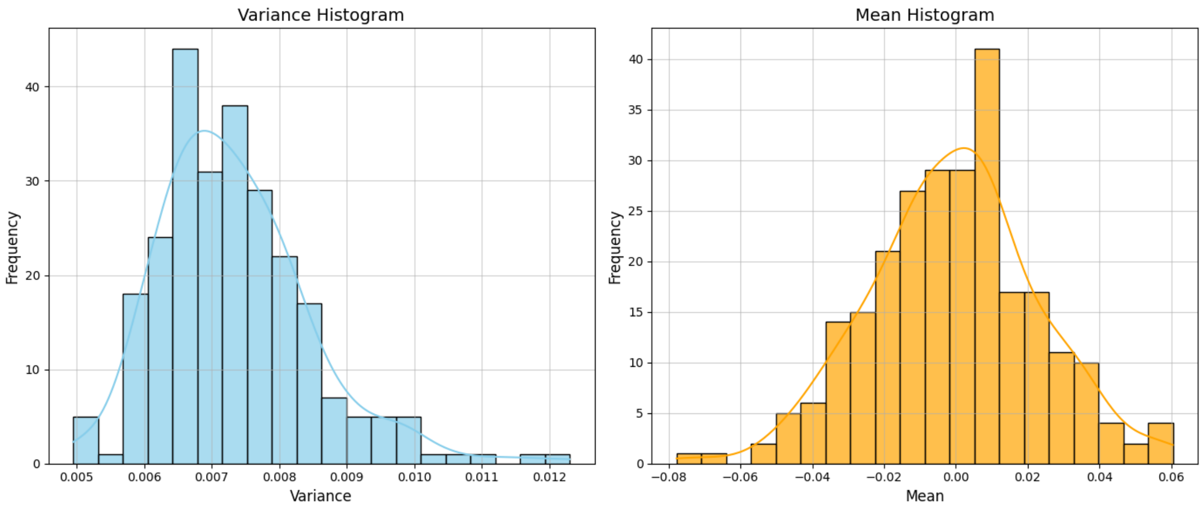

FIGURE 11 – Histogram of mean and variance coordinates of each latent space dimension

- The mean distribution of latent coordinates on each axis appears to follow a normal distribution centered at 0. On average, latent vector coordinates on each dimension are centered around 0, with some latent space axes off-center (at -0.38 or 0.06, for example).
- The variance of latent space coordinates on each axis appears to be on average $\sigma^2 = 0.04$. According to graphs each axis of the latent space seems to respect approximately the same properties, which could indicates that the latent space is isotropic

By plotting the PCA projection of 1000 signals on the 2 axes that best explain the variance, we can see that the latent space does distribute the signals uniformly :

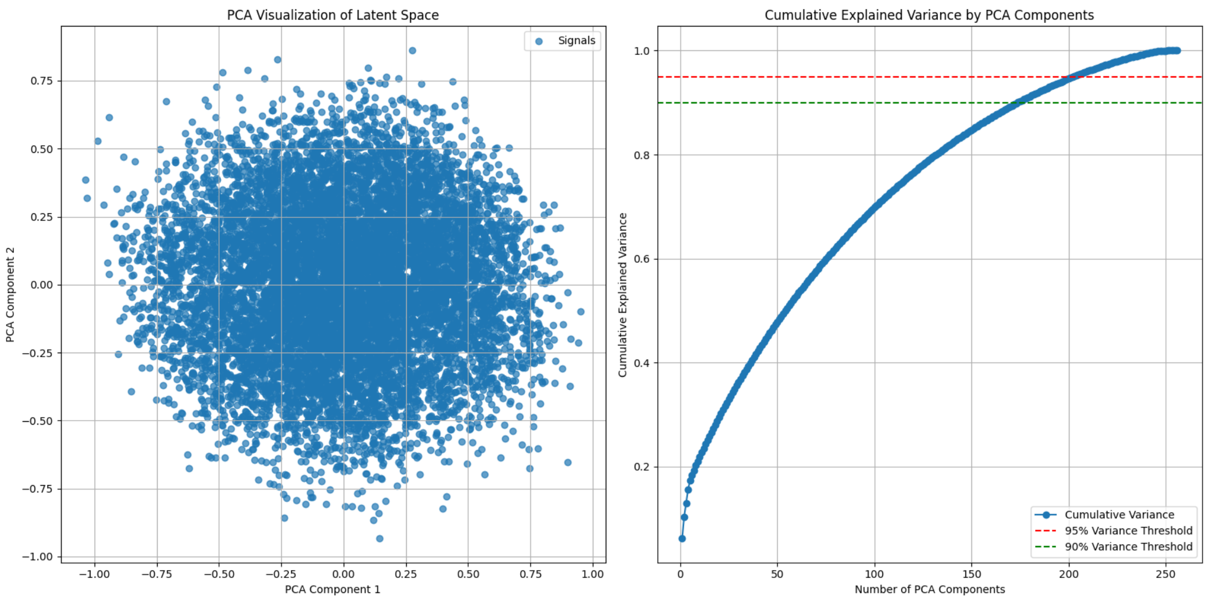

FIGURE 12 – PCA full dataset projection (left) and cumulative explained variance evolution (right)

In addition, we note that the explained variance of the latent space is 95% interpenetrable by the first 200 dimensions of the latent space, so it would be possible to perform a dimension reduction to reduce computational costs and further compress the data. However, for performance reasons, we prefer to keep the 256 dimensions, as we are not limited by the computational costs involved in this project. In addition, PCA projects the data into a space that allows maximum interpretation of the variance, but this projection could disturb the notion of distance defined in the initial latent space.

Conclusion : Latent space structure and properties

This detailed study of latent space has shown that it preserves and encodes the notion of inter-parameter distance. The structure thus defined not only enables time series clustering to be performed, but also allows inter-signal distance to be measured in a way that is robust to noise, contraction, dilation and phase shift in a reduced computation time, something that the methods presented in the state-of-the-art 2.2 were unable to do well.

6 ECG signals experiment

Now that we've demonstrated the effectiveness of our auto-encoder in preserving the notion of distance on synthetic signals and reconstructing them efficiently, it's time to test and train the auto-encoder model on a dataset of much more complex signals applied to a real life field.

6.1 ECG signals dataset

The dataset we want to test our auto-encoder on is an ECG open source signals dataset issued from **WaveForm Database**. The idea is to train our auto-encoder specifically on this dataset containing 48 different 30-minutes recordings, which we're going to slice into approximately 30,000 signals of 4 seconds and 1024 points. In this experiment, we suspect that our model will overfitter, but the aim of the experiment is not to generalize to a large class of signals, only to demonstrate the usefulness of such a model on a more complex class of signals applied to a real-life domain. Here is an overview of the real signals whose distances we are trying to measure :

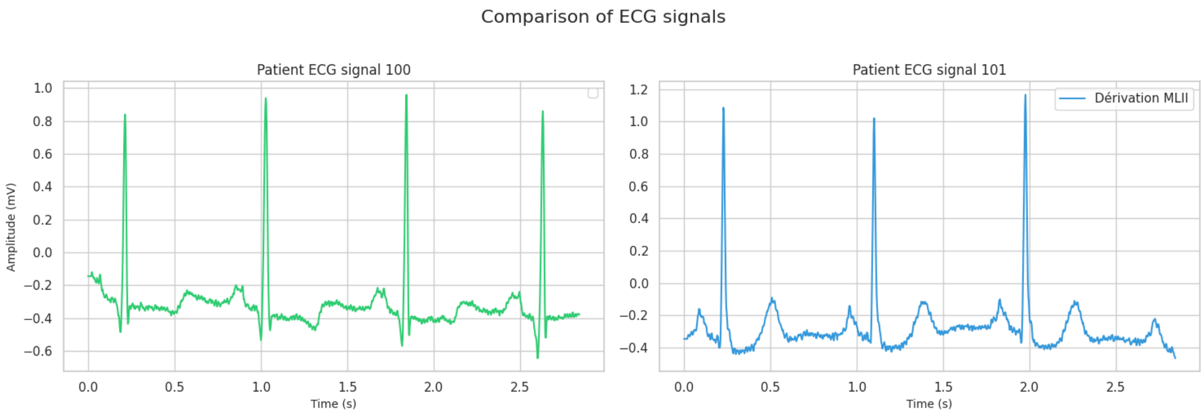

FIGURE 13 – ECG signals of the dataset

The ECG signal dataset is quite large, so it's possible to dispense with the synthetic signal generation described in part 3.3.

Note : about synthetic dataset generation

It's not uncommon to find datasets that don't contain enough signals. For this, we can perform data augmentation and approximate our initial dataset with synthetically generated signals, for example by performing a spectral analysis and generating signals containing the dominant frequencies/amplitudes of our dataset, or by using GAN generative models. Generating a synthetic dataset has several advantages :

- The number of signals contained in the dataset is arbitrary : we can deliberately choose a very large dataset in order to train our autoencoder on a wide range of signals.
- For each signal in the dataset we have exact knowledge of the parameters of interest on the basis of which we wish to construct a distance measure.
- We have control over the noise we wish to incorporate into the signals, in order to train the autoencoder to be robust to noisy signals.

6.2 ECG distance selection

It's important to remember that defining the distance between signals is **arbitrary**. The only constraint is to have access to the parameters from which we want to define our notion of distance. A multitude of distances can be chosen, such as :

- **Distance between the main harmonics of the signal** : This distance measures the quadratic difference between the frequencies of the main harmonics of two signals s_i and s_j.
$$d(s_i, s_j) = \left(d_{\text{harmonique}_{si}} - d_{\text{harmonique}_{sj}}\right)^2 \tag{6}$$
- **Distance between ECG signal peak amplitudes** : This distance measures the difference between the peak amplitudes of two ECG signals. If A_i and A_j represent the peak amplitudes of the s_i and s_j.
$$d(s_i, s_j) = |A_i - A_j| \tag{7}$$
- **Distance between P wave intervals** : This distance measures the difference between the P wave intervals of two ECG signals, as illustrated in figure reffig :ecg-features. If T_{P_i} and T_{P_j} represent the P wave intervals of signals s_i and s_j.
$$d(s_i, s_j) = \left|T_{P_i} - T_{P_j}\right| \tag{8}$$

There are as many distance measurements as the author has the imagination to construct. We simply need to find the distances that are easiest for the auto-encoder to interpret, as we've seen in section 5.1 that the notion of frequency distance in latent space is more difficult to learn and interpret by the auto-encoder for example. In this example we've chosen the distance between the 20 dominant frequencies of the ECG signal with their associated amplitude and phase as the training distance. It's worth bearing in mind that this distance is not necessarily the most relevant, as it doesn't necessarily take into account the harmonics and characteristics of a healthy ECG signal.

$$d(s_i, s_j) = \sqrt{\sum_{k=1}^{20} \left(\alpha \left|w_{i,k} - w_{j,k}\right|^2 + \beta \left|A_{i,k} - A_{j,k}\right|^2 + \gamma \left|\phi_{i,k} - \phi_{j,k}\right|^2\right)} \tag{9}$$

In this way, the latent space will be structured according to this signal distance, and once trained, we'll no longer need to have signals labeled with phase, frequency and amplitude parameters ; the autoencoder will manage to naturally project the signals into its latent space to interpret the distance.

6.3 ECG parameter trajectory in latent space

As in section 5.3 we aim to study the trajectory of the parameters of interest in order to visualize the structure of the latent space. Here, the parameters of interest described in section 6.2 are the 20 dominant frequencies of the signal and their associated phases and amplitudes. By linearly varying each parameter individually, we obtain the following trajectories in latent space :

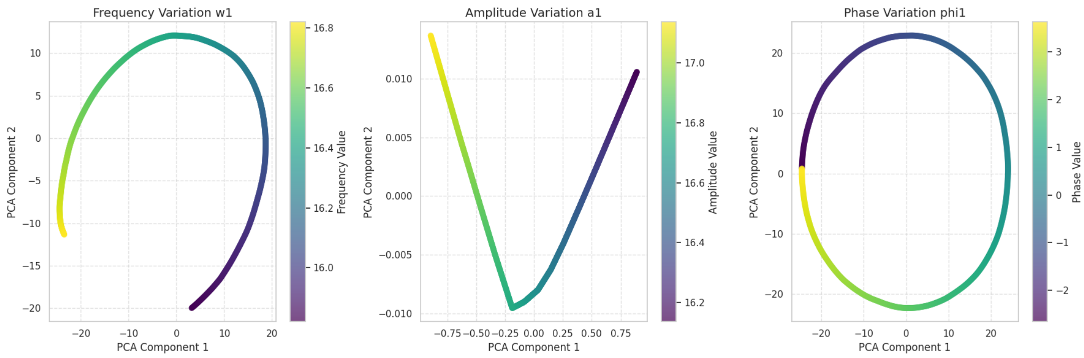

FIGURE 14 – Trajectory of parameters of interest in latent space

The results are similar to those in section 5.3 : the linear and trigonometric properties of the various parameters appear to be correctly encoded in latent space (although we only display variations of the first 3 parameters a_1, w_1, ϕ_1. Nevertheless, latent space encodes frequencies linearly over intervals of about 1Hz in length. If we vary the frequencies over an interval greater than 1Hz, the trajectory in latent space is much more erratic and non-linear, but still continuous (cf appendix 8.2), illustrating once again the difficulty of correctly encoding the frequencies of a signal.

6.4 ECG signals distance measurement

In this part, we check whether the distance measurement is coherent by verifying the proximity or otherwise of the signals as a function of their distance. To do this, from the signal dataset we display the 2 closest/distant signals and 2 signals taken at random :

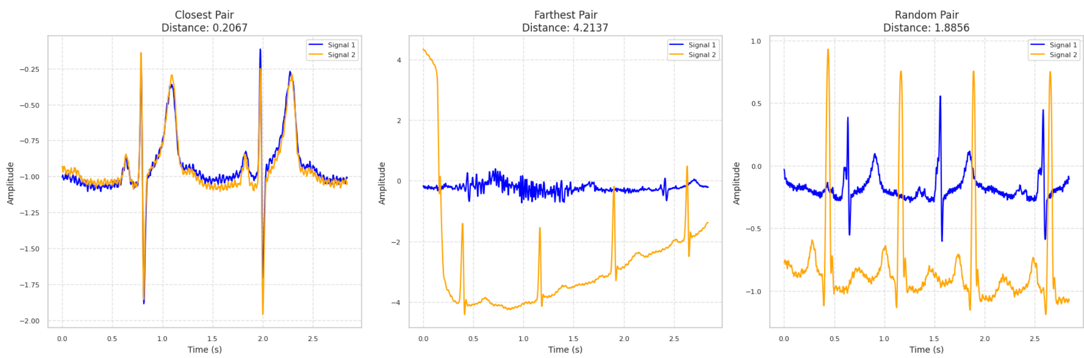

FIGURE 15 – Distance measurement beetween the farthest, closest and random signals

The distance measurements between the signals appear to be quite consistent, with a near-zero distance for quasi-identical signals with no phase shift, a high distance of 4.2 between an ecg signal and a noisy ecg signal, and finally a moderate distance between 2 fairly similar ECG signals from 2 different heart rates.

6.5 ECG signals auto-encoder reconstruction

In this last section, we check that, in addition to a latent space that correctly encodes the notion of distance, the reconstruction of the signals is acceptable, in order to conclude on the relevance and efficiency of the model applied to more complex signals.

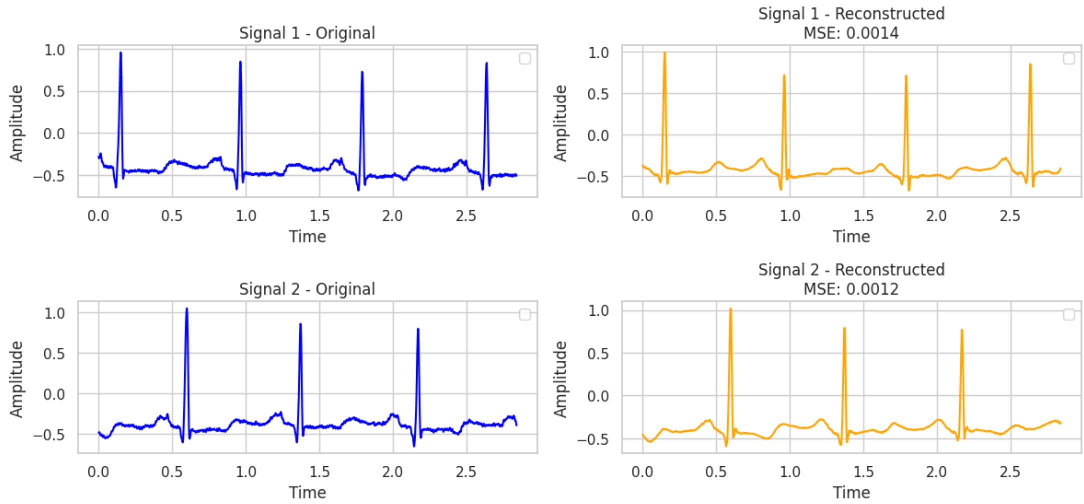

FIGURE 16 – ECG signals auto-encoder reconstruction

Signal reconstruction is clearly acceptable, with very low MSE errors of the order of 10^{-3}. We conclude that convolutional layers are once again effective on signals even with more complex frequency representations. It would therefore also be possible to perform anomaly detection on such signals, which would be very useful in the medical field for pathology detection.

6.6 ECG signals experiment conclusion

This section demonstrated the usefulness of measuring distance between signals from latent space on more complex real life signals. The experiment showed that :

- The latent space structure is again consistent with distance measurement, although special attention needs to be paid to the more complicated frequency encoding, in the case of distance definition using frequency parameters.
- The distance measurement between signals seems perfectly coherent.
- Signal reconstruction is also largely satisfaying, enabling anomaly detection.

7 Conclusion and future work

This research paper proposes an innovative way of measuring distances between time series of the same class. The method implemented is the distance measurement of signals from the latent space of a convolutional auto-encoder. The main objective was to propose a distance measurement solution that is robust to noise, contractions, dilations, phase shifts and time shifts. To achieve this, the auto-encoder was trained not only to faithfully reconstruct the original signal, but also to project the signals into a latent space where the distances between the signals faithfully reflect the distances defined in the original parameter space.

Key results and contributions :

- Distance preservation in latent space : The auto-encoder succeeded in encoding a consistent notion of distance in latent space, especially for linear parameters such as amplitudes and polynomial coefficients. Although the encoding of frequencies was more complex due to their non-linear nature, the latent space still showed an ability to encode frequencies linearly by chunk.
- Signal reconstruction : The model proved highly efficient at reconstructing signals, even when they were noisy.
- Practical applications : This approach was first tested on synthetic signals, then successfully applied to complex ECG signal data. The results confirmed that distance measurement in latent space is reliable, and that ECG signal reconstruction is very satisfying. This opens up interesting prospects for applications such as anomaly detection or biomedical signal classification.
- Latent space structure : Analysis of the latent space has revealed a coherent organization, with well-defined parameter trajectories and the ability to group similar signals. This makes it possible to envisage additional applications, such as unsupervised clustering or the generation of new signals via interpolations in latent space.

In the future, we plan to extend the researches :

— First test the model on other time series in order to generalize the method to several classes of signals, notably more random and complex signals such as financial asset signals. For this, adopting a more complex convolutional autoencoder architecture such as UNET would certainly improve performance.

— The frequency data of signals are essential parameters to model properly, so defining a loss that focuses partly on an efficient projection of frequencies into latent space would considerably improve performance and generalization to other classes of signals.

— An in-depth comparison of distance measurement performance between our proposed method and the DTW, MSE between-signal and spectral-difference methods would enable us to more accurately demonstrate the value of such a measurement method.

Références

[1] Yao Qin et al. "A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction". In : *University of California, San Diego and NEC Laboratories America, Inc.* (2017). Proceedings of the 34th International Conference on Machine Learning (ICML). URL : https://arxiv.org/abs/1704.02971.

[2] Nutan Chen, Patrick van der Smagt et Botond Cseke. "Local Distance Preserving Auto-encoders using Continuous KNN Graphs". In : *Machine Learning Research Lab, Volkswagen Group, Germany* (2020). Germany. URL : https://arxiv.org/abs/2206.05909.

[3] Tsegamlak Terefe et al. "Auto-encodeur multi-taches pour le calcul de moyennes de series temporelles". In : *Atelier Apprentissage profond : Théorie et Applications (APTA), Journées Francophones Extraction et Gestion des Connaissances (EGC)*. Submitted on 10 Mar 2022. Montpellier, France, jan. 2021. URL : https://hal.science/hal-03604169v1.

[4] Miklovana Tuci. "Distance preserving Fermat VAE". Master's Thesis, Machine Learning, 120 credits. Mém. de mast. KTH Royal Institute of Technology, 2022. URL : https://www.diva-portal.org/smash/get/diva2:1711712/FULLTEXT01.pdf.

[5] Max F. Frenzel, Bogdan Telegat et Asahi Ushio. "Latent Space Cartography : Generalised Metric-Inspired Measures and Measure-Based Transformations for Generative Models". In : *Cogent Labs* (2023). Tokyo, Japan. URL : https://arxiv.org/abs/1902.02113.

8 Appendices

8.1 Appendix A : ECG signal features

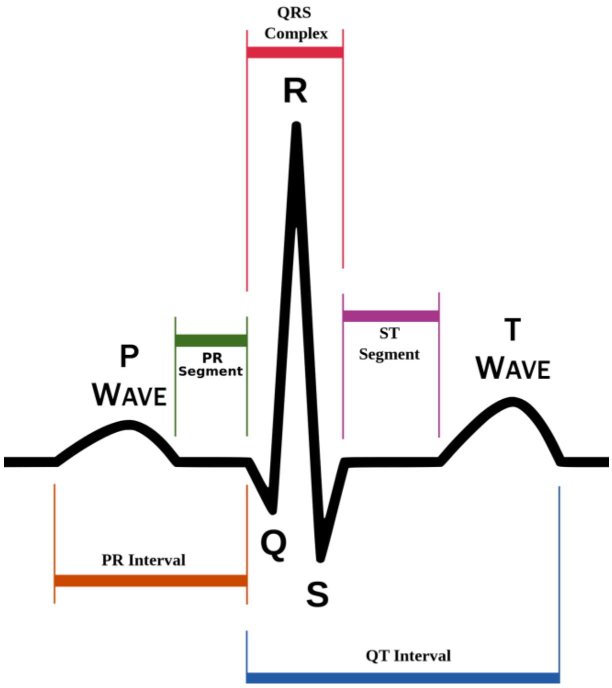

FIGURE 17 – Features of an ECG signal

This figure illustrates the main characteristics of an ECG signal from a healthy patient. Defining a notion of distance based on these characteristics could make it possible to define a notion of distance in latent space in relation to these parameters. Thus, a signal that is farther away in terms of distance from an ECG signal could be considered to reflect a pathological condition.

8.2 Appendix B : Erratic trajectory of frequency parameter

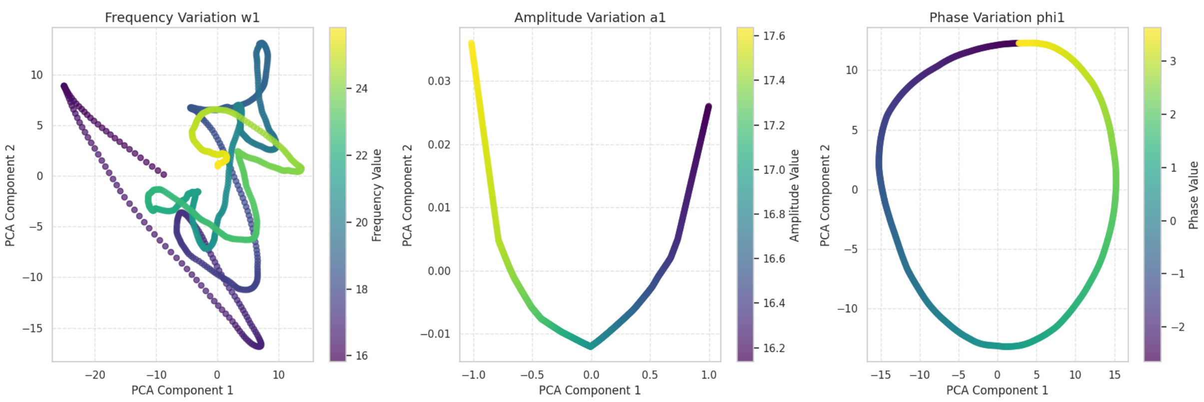

FIGURE 18 – Irregular trajectory of w parameter within latent space

Once again, this figure illustrates the difficulty of correctly encoding frequencies from latent space, although the trajectory remains continuous, illustrating that frequencies that are close in departure space are also close in arrival space. One hypothesis concerning the encoding of these frequencies is that the auto-encoder projects the frequencies onto linear representations by chunk, each frequency interval at a dedicated location in latent space. If we plot the PCA of the frequency trajectory on 1Hz slices each time, it turns out that the trajectories are quite linear or arched, proving that the auto-encoder has nevertheless managed to preserve some form of distance between frequencies in its latent space.