Improving the calibrated data using temporal patterns in Air Quality Sensor Monitoring Networks

Giacomini Nicolò

December 27, 2023

Abstract

In this paper I want to show and describe the results of a denoising applied to an IoT sensors system. The sensors gather data regarding the quality of air, in particular they measure the O_3 in the air in order to understand the pollution present in the air.

The experience will show how it is possible get better results despite the presence of noise. The noise of the sensors is due to bad quality of sensors, bad position of the sensors, weather conditions or electromagnetic interference. I will use Machine Learning techniques in order to get good results despite those problems. The goal is to show how can get good results using difference way.

In this experience I have available dataset from reference station data, more reliable sensors and low cost sensors. First I will understand the data and how it varies over time. Then, I will evaluate the dataset of low cost sensors comparing the RMSE calculated with the reference station dataset. Then I will denoise the data of the low cost sensor using two approaches: the eigenfaces and the Gavish *et al.* method. Finally I will compare the results between the two methods.

1 Introduction

Initially we have three different dataset:

- Reference station dataset: it is the dataset of the values measured by the sensors in the station. These values are the most reliable.
- MLR dataset: it is the dataset of the values measured by the Low Cost Sensors (LCSs). In this dataset the data are been calibrated using Multiple Linear Regression.
- SVR dataset: it is the other dataset of the values measured by the Low Cost Sensors (LCSs). In this dataset the data are been calibrated using Support Vector Regression.

We can see the data in the following plots:

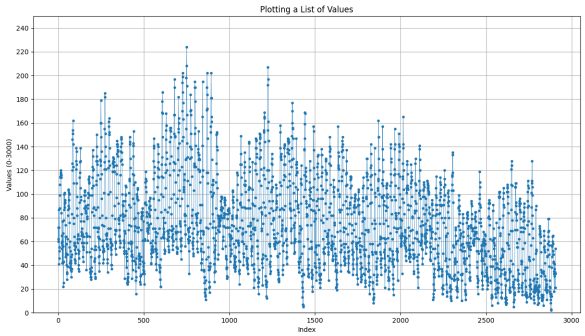

Figure 1: Reference data

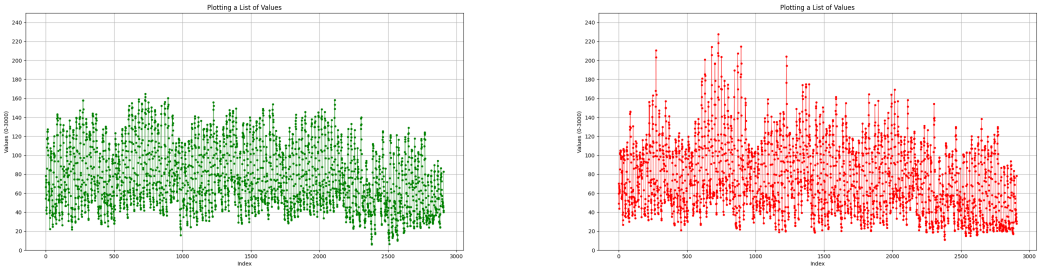

Figure 2: MLR data (green) and SVR data (red)

We can compare MLR and SVR plots with the reference plot. I took the data of the first 5 days in order to see better the differences. The plot is in the figure 3.

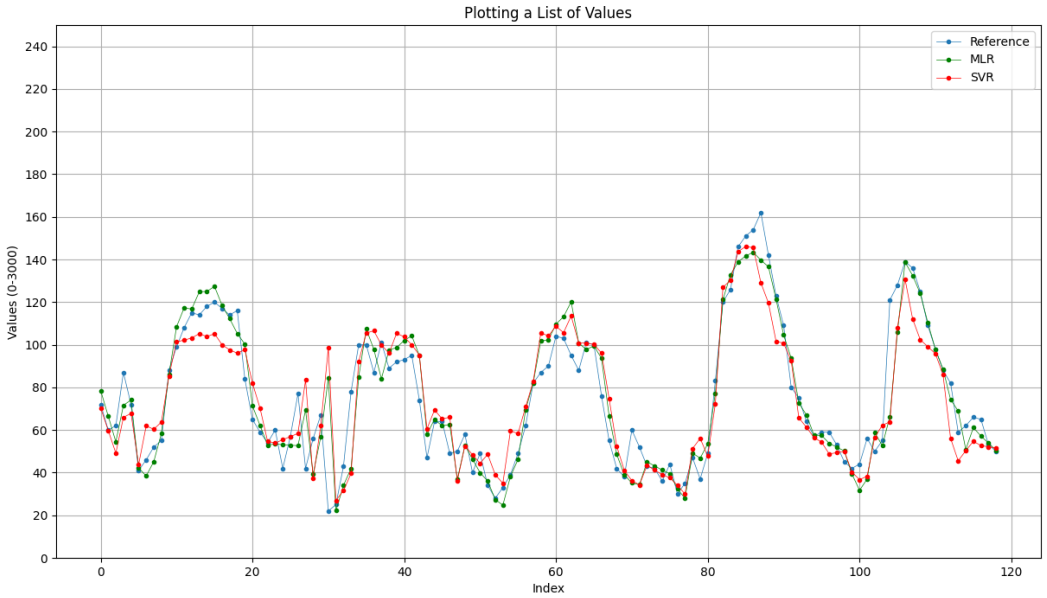

Figure 3: Reference data compared with MLR data and SVR data

2 Denoising data

2.1 Calculus of RMSE

The RMSE (Root Mean Squared Error) is a index that allows to understand the error and defines how much it is far the approximation with the reference station data. The formula is the following:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2} \tag{1}$$

In the case of the Multiple Linear Regression I get $RMSE = 13.9822$ and $R^2 = 0.8698$, while in the Single Vector Regression I get $RMSE = 12.6898$ and $R^2 = 0.8928$. The RMSE parameter it is used in order to measures the average of the errors between predicted values and observed values. A lower RMSE value indicates that the model prediction is closer to the real values, so it has better accuracy. Instead an higher RMSE value implies a larger difference between predicted and actual values, so it indicates that the model is not very good. The R-squared value is an other indicator used to indicate how much of the variation of a dependent variable is explained by an independent variable in a regression model. Usually it is considered as a percentage and a value grater or equal to 85% is considered good.

In order to get better results I removed the incomplete days, i.e. those days where there are not all the hourly measurements. There are 104 complete days in the dataset. Without the incomplete days I get these results:

- Multiple Linear Regression:
  - RMSE= 13.424
  - $R^2 = 0.8764$
- Single Vector Regression:
  - RMSE= 12.4545
  - $R^2 = 0.8936$

The results tell that the SVR is better than MLR, so the SVR data are close to the reality.

The following graphs are the plots of the day without the incomplete days:

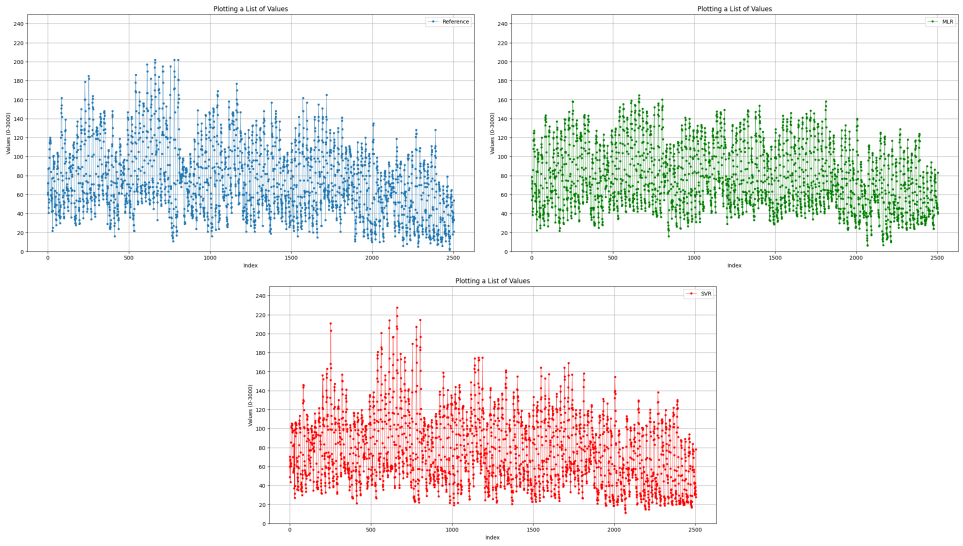

Figure 4: Reference data, MLR data and SVR data without incomplete days

2.2 Denoising data with the reference station

SVR dataset Now, I perform the denoising of the data. Initially I consider as a new reference data, the SVR dataset, because it has the lowest RMSE.

First of all, I calculate the average vector of all days of the reference dataset. I used this formula:

$$\Psi = \frac{1}{M} \sum_{i=0}^{M} x_i \tag{2}$$

where x_i are the vectors of reference dataset X where each vector is composed by 24 values, in my case $X = [x_1, ..., x_M] \in \mathbb{R}^{D \times M}$ where $D = 24$ and $M = 104$. Ψ represents the average set of data acquired by each sensor.

Then I calculate the Y matrix that contains the values of the SVR dataset. I compute the new Y' matrix (normalized matrix). In order to do this, I have to subtract to each vector of Y the Ψ vector, in this way I can calculate how much every vector of the dataset is far from the average vector. I use the following formula:

$$\hat{y_i} = y_i - \Psi \tag{3}$$

where y_i is each vector of Y matrix.

Now I have to calculate the approximation matrix, in this case I use the SVD decomposition. But I don't know which is the best low-rank approximation, so I check every low-rank approximation and I choose the low-rank with the minimum RMSE.

I calculate the normalized matrix also for the reference dataset (X matrix), like I did for the SVR dataset and then I calculate its SVD decomposition:

$$X' = U \Sigma V^T \tag{4}$$

where X' matrix is the normalized reference station dataset.

Then I use the following formula for the calculus:

$$\tilde{y_i} = \Psi + U_k U_k^T \hat{y_i} \tag{5}$$

where U_k is the matrix of rank k obtained by the SVD of the reference data, as shown in 4, and $\hat{y}_i$ is the vector of the computation in the formula 3. At the end I build the $\tilde{Y}$ matrix that is composed by the $\tilde{y}$ vectors. Once I get the $\tilde{Y}$ matrix I can calculate the RMSE value using as a reference the X matrix. So I repeat the formula 1 comparing every vector of X matrix with the corresponding vector of the $\tilde{Y}$ matrix. This table show the RMSE values that I obtained:

Rank	RMSE Value
1	15.3646
2	13.1128
3	12.2903
4	11.9974
5	11.6778
6	11.4834
7	11.4259
8	11.3899

Rank	RMSE Value
9	11.5449
10	11.5721
11	11.6774
12	11.7402
13	11.7846
14	11.8258
15	11.8464
16	11.9572

Rank	RMSE Value
17	12.0210
18	12.1014
19	12.1803
20	12.2474
21	12.3157
22	12.3589
23	12.4104
24	12.4545

Table 1: RMSE values based on the rank

We can see that the best approximation is the **rank 8**. This mean that if we use more than 8 values for each day in our dataset, we do not get better result, but, on the contrary, we add noise to the dataset which give us worse information. On the other hand, if we use less than 10 values for each day, we cannot describe well the dataset, so we lost information.

In the following plot, we can see how the RMSE change based on the low-rank approximation used in the SVD for the calculus of the matrix U_k.

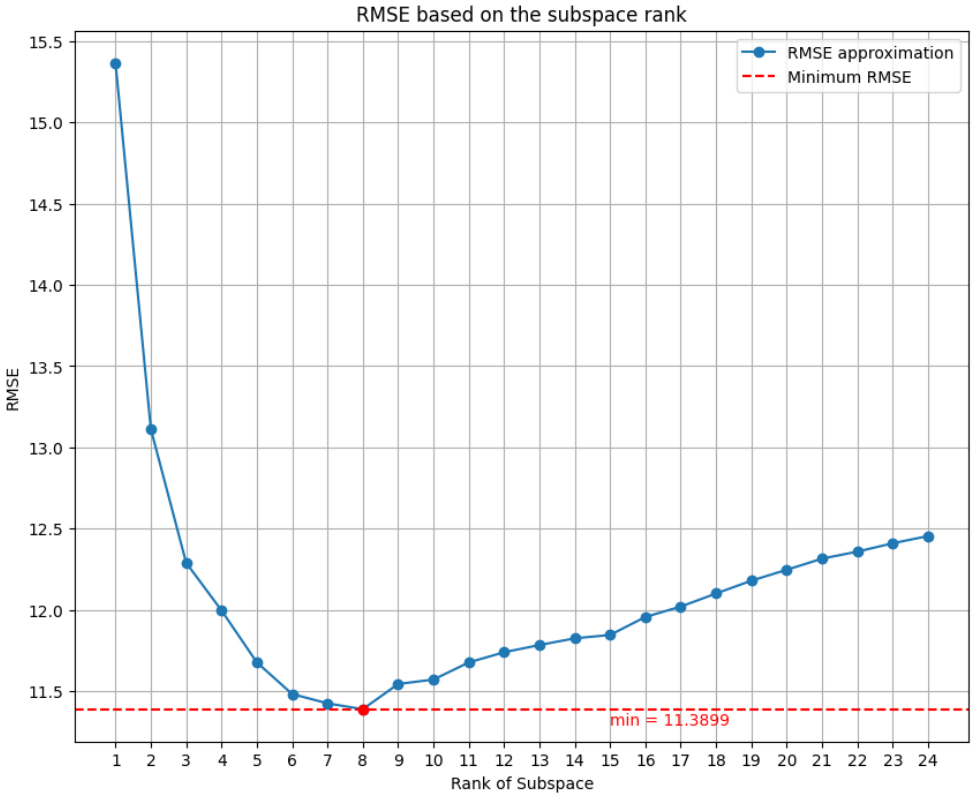

Figure 5: RMSE of the low-rank approximation of SVR

In the next plot we can see that also R-squared value increases with a rank of 8.

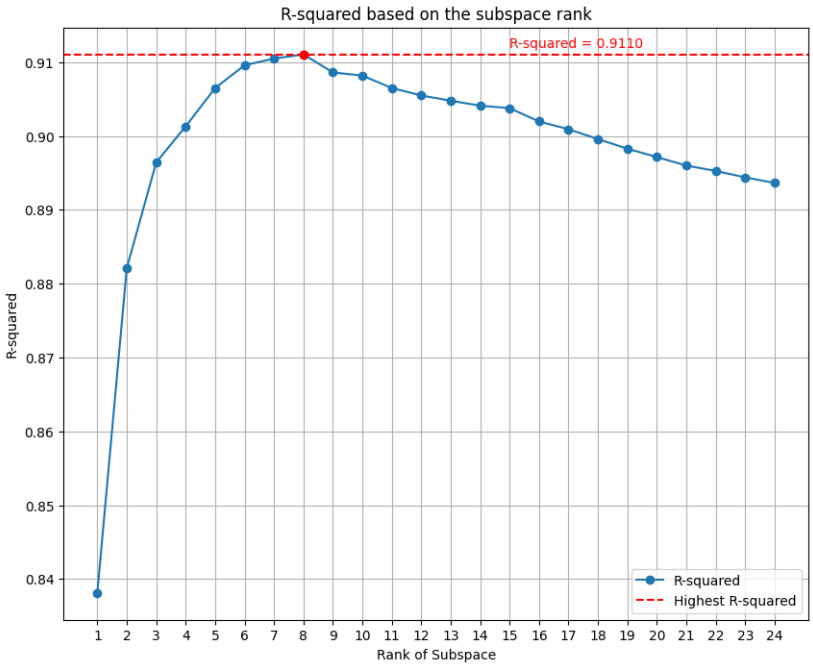

Figure 6: R-squared of the low-rank approximation of SVR

The following plot shows the difference between the reference station dataset and the denoised SVD dataset obtained by the low-rank approximation:

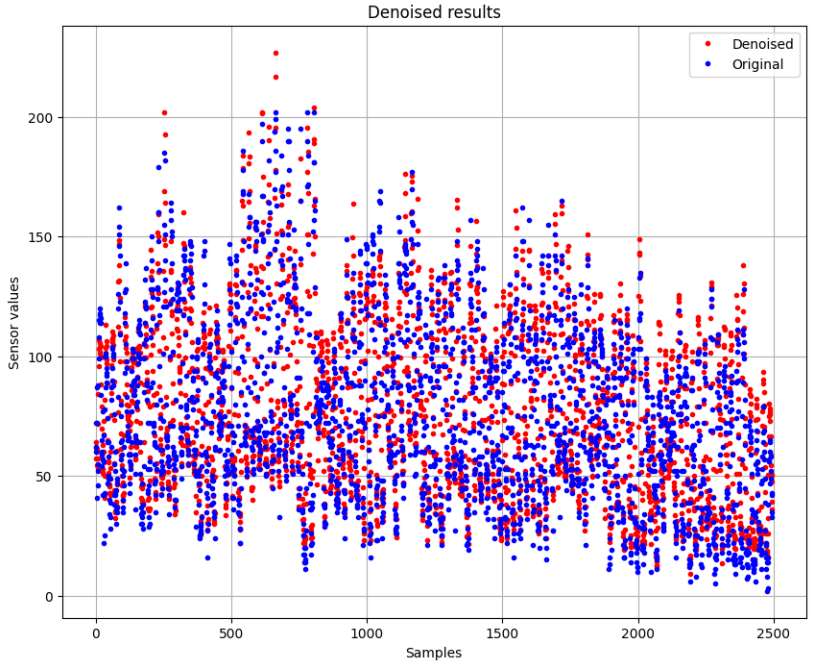

Figure 7: Denoised result and original result

Finally, in the last figure we can see better the approximation just plotting the first 5 days.

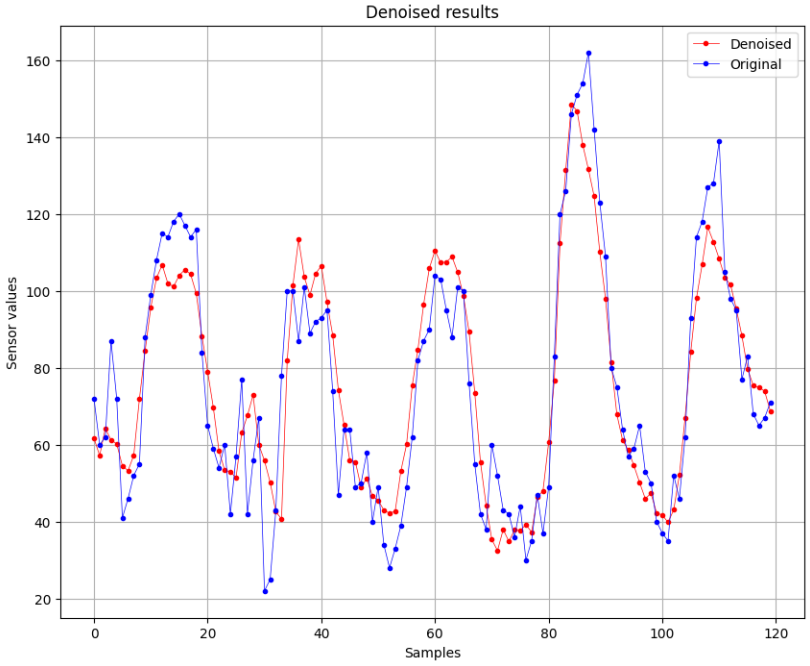

Figure 8: Denoised result and original result of 5 days

MLR dataset I tried to repeat the same procedure but using the data of MLR dataset. As I expected, the results are worse than the data of SVR:

Rank	RMSE Value
1	15.8460
2	14.5505
3	13.8506
4	13.5408
5	13.2643
6	13.1056
7	13.0482
8	12.9914

Rank	RMSE Value
9	12.9843
10	12.9774
11	13.0189
12	13.0738
13	13.0844
14	13.1153
15	13.1239
16	13.1876

Rank	RMSE Value
17	13.2139
18	13.2498
19	13.2759
20	13.3112
21	13.3485
22	13.3768
23	13.4041
24	13.4240

Table 2: RMSE values based on the rank

As we can see, the best low-rank approximation is the **rank 10**. In the following plot, we can see how the RMSE change based on the low-rank approximation used in the SVD for the calculus of the matrix U_k.

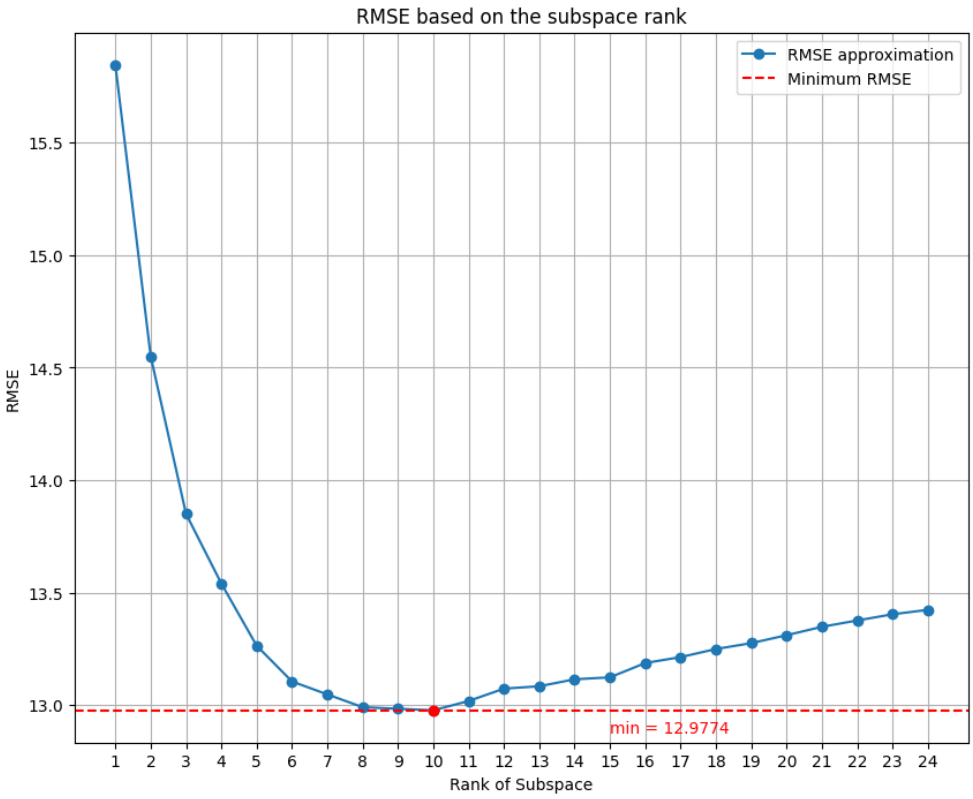

Figure 9: RMSE of the low-rank approximation of MLR

In the following plot we can also see how the R-square changes based on the rank. We can see that the best R-square value corresponds to the low-rank approximation.

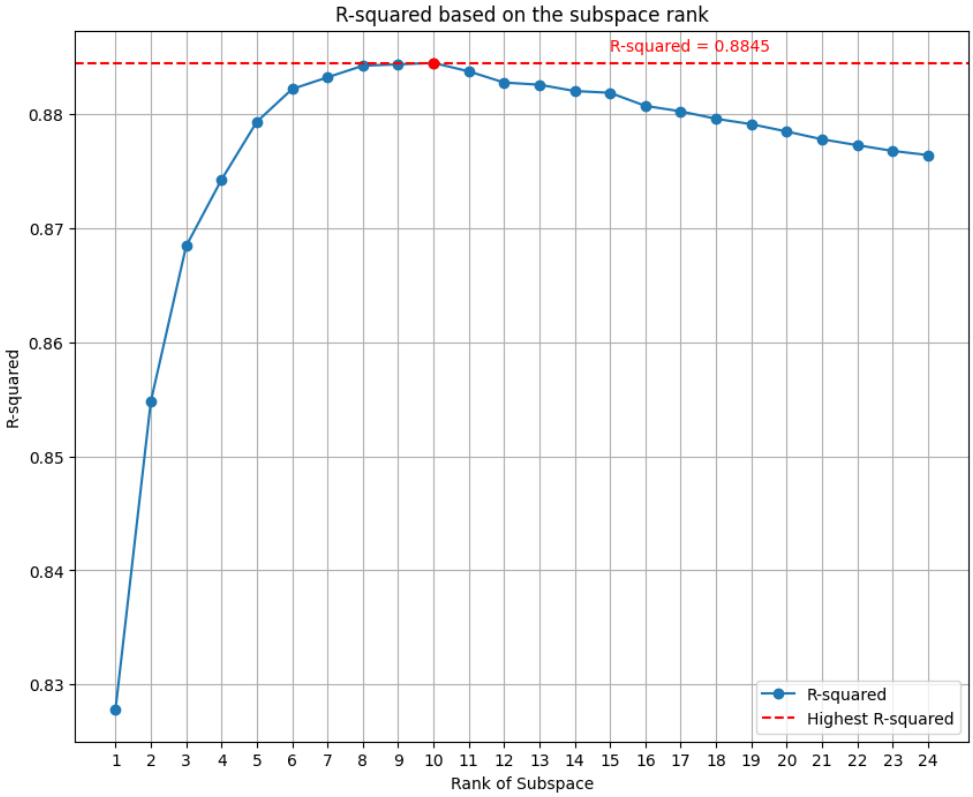

Figure 10: R-square of the low-rank approximation of MLR

The following plot shows the difference between the reference dataset and the new dataset obtained by the low-rank approximation:

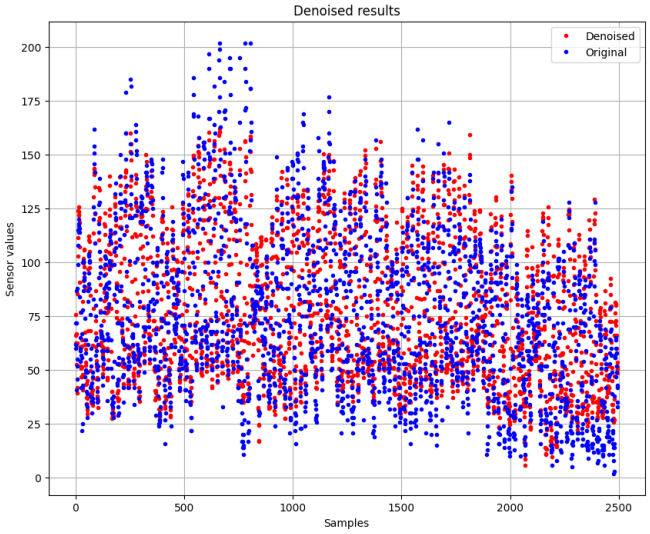

Figure 11: Denoised result and original result

Finally, in the last figure we can see better the approximation just plotting the first 5 days.

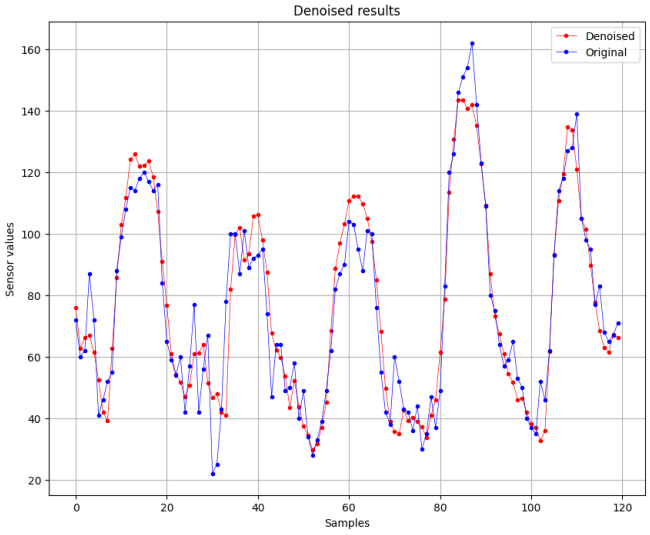

Figure 12: Denoised result and original result of 5 days

2.3 Denoising data with Gavish *et al.* method

Now I repeat the denoising of data using the Gavish *et al.* method. With this method we don't use the a reference dataset but we use just one dataset, then I will compare the results with the reference station dataset to check whether the obtained results are good or not. The main goal of this method is to try to find a low-rank approximation without use the reference station data. In the next pages I show the results that I obtained.

SVR dataset The first dataset on which I do the test is the SVR dataset. First of all I have to find the rank for the approximation. In order to do this, I find a threshold value. The threshold value is found by the following formula:

$$\tilde{\sigma} = \omega(\beta) \cdot \sigma_{median} \tag{6}$$

where $\beta = \frac{D}{M} = \frac{24}{104}$, σ_{median} is the median of the singular values obtained by the SVD decomposition and ω is the following function:

$$\omega(\beta) = 0.56 \cdot \beta^3 - 0.95 \cdot \beta^2 + 1.82 \cdot \beta + 1.43 \tag{7}$$

Once found the threshold value $\tilde{\sigma}$, I use the Σ matrix calculated by the SVD decomposition in order to find the rank approximation. In fact the Σ matrix contains on the diagonal the singular values in descending order. The minimum singular value above of the $\tilde{\sigma}$ defines the rank for the approximation. In this case I found the following results:

- $\sigma_{median} = 78.0116$
- $\omega(\beta) = 1.8063$
- Threshold $\tilde{\sigma} = 140.9115$
- Minimum rank $= 7$

In the case of the SVR dataset, the **minimum rank** found is **7**, as we can see in the following plot:

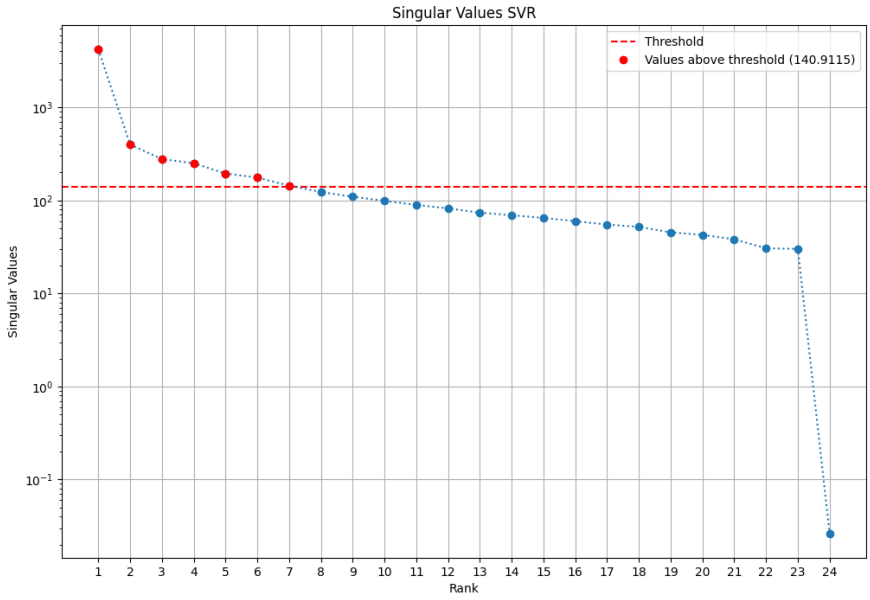

Figure 13: Singular values and threshold of SVR dataset

Now I can repeat the operation that I did in the previous method, but with some differences. In this case, since I have to denoising the data without use the reference station data, first I calculate the Ψ vector using the dataset (in this case SVR). Then I calculate the $\hat{y}_i$ subtracting Ψ for each vector of the matrix of the dataset. These vectors will compose the normalized matrix. After I calculate the U matrix of the normalized matrix using the SVD. Finally I get the denoising vectors using the formula 3 where Ψ, U and $\hat{y}_i$ have just been calculated and I obtain $\tilde{Y}_i$ that are the vectors of the denoised matrix.

I calculate again the RMSE using the reference station dataset in order to understand and evaluate the results. We can see the plot of RMSE in the following figure:

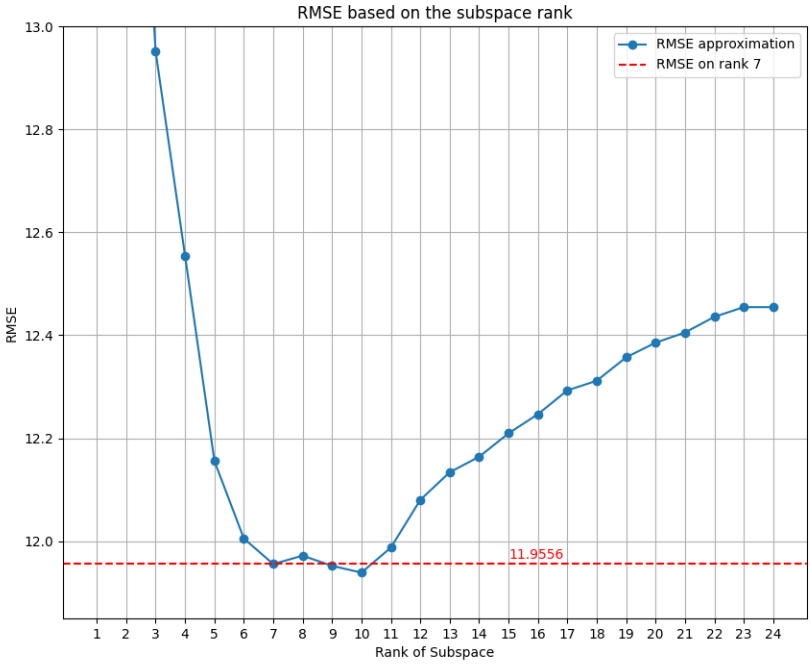

Figure 14: Rank approximation with Gavish *et al.* method

In the next plot we can see that the R-squared value on the rank 7 is slightly lower than the best R-squared value.

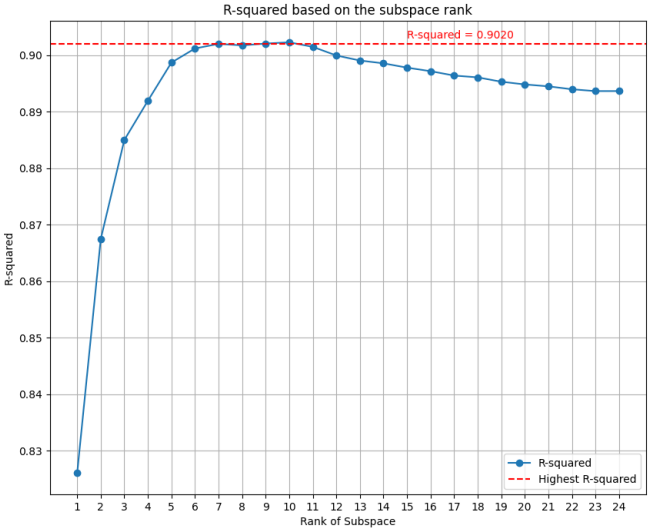

Figure 15: R-squared of the approximation of the Gavish *et al.* method

The following plot shows the difference between the reference dataset and the new dataset obtained by the low-rank approximation:

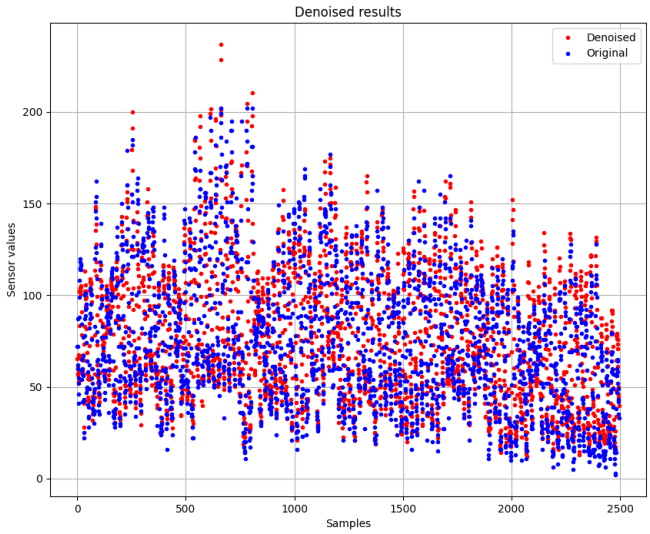

Figure 16: Denoised result and original result

Finally I plot the denoising of the first 5 days.

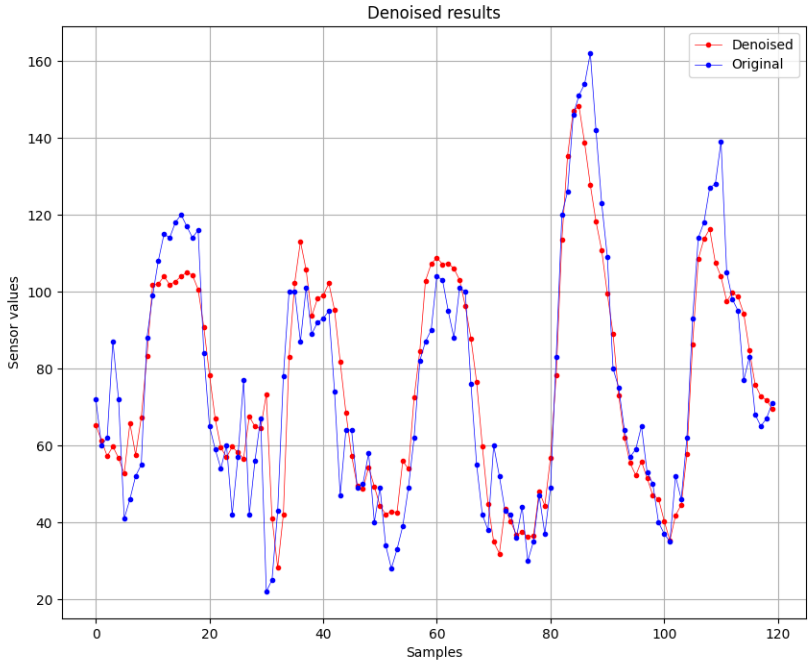

Figure 17: Denoised result and original result of 5 days

As we can see, the results are quite good, in fact the approximation is not perfect, but it is very close to the minimum rank that would be rank 10. In this case I am satisfied about the results.

MLR dataset Now I can repeat the test using the MLR dataset. In this case I found the following results:

- $\sigma_{median} = 58.6607$
- $\omega(\beta) = 1.8063$
- Threshold $\tilde{\sigma} = 105.9583$
- Minimum rank = 7

Also with the MLR dataset the minimum rank found is 7, as we can see in the following plot:

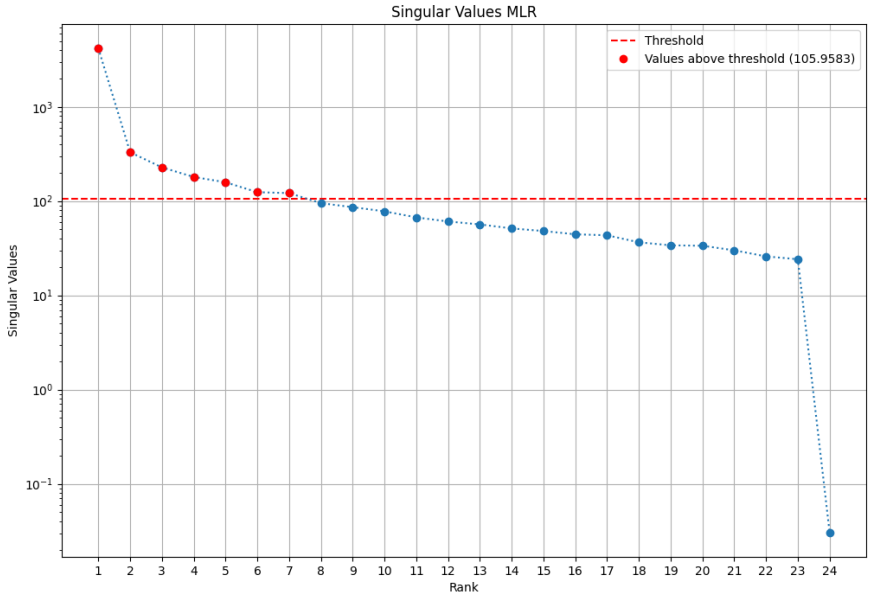

Figure 18: Singular values and threshold of MLR dataset

Here I repeat the same procedure that I used for the SVR dataset, in order to get the denoised matrix.

I calculate again the RMSE using the reference station dataset in order to evaluate the results. We can see the plot of RMSE in the following figure:

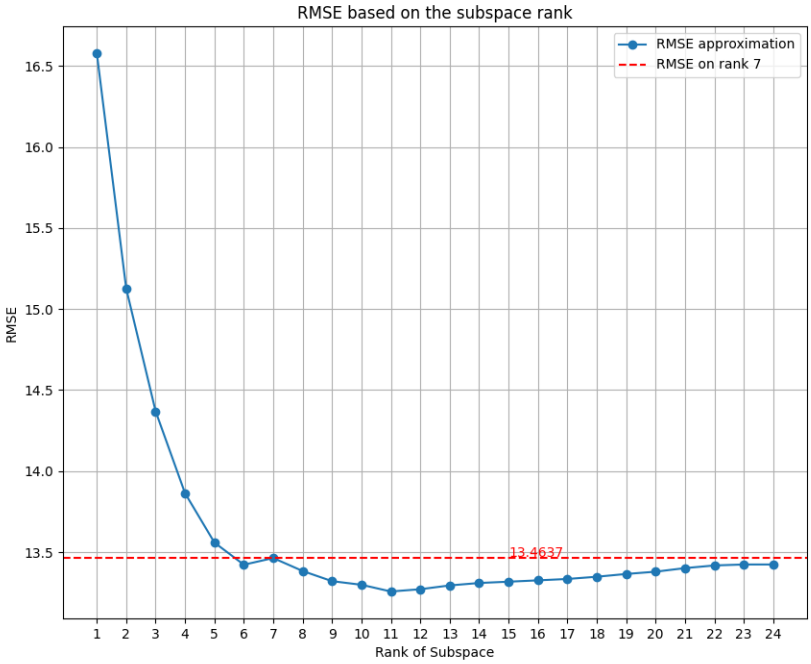

Figure 19: Rank approximation with Gavish *et al.* method

In the next plot we can see that the R-squared value on the rank 7 is very lower than the best R-squared value. The result is not so good.

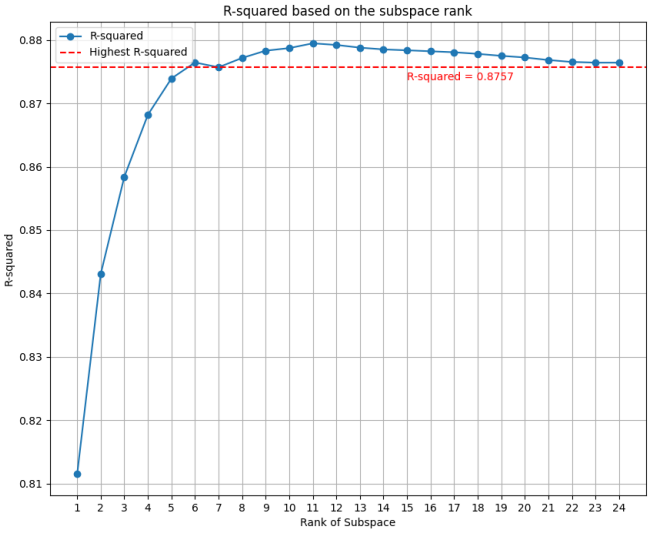

Figure 20: R-squared of the approximation of the Gavish *et al.* method

The following plot shows the difference between the reference dataset and the new dataset obtained by the low-rank approximation:

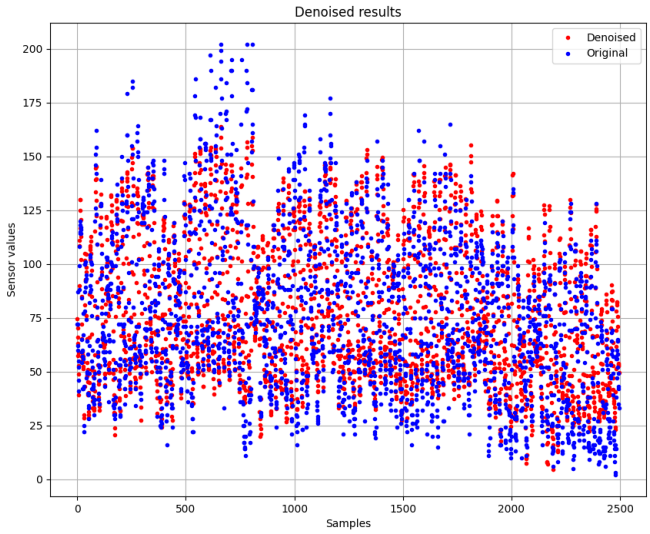

Figure 21: Denoised result and original result

Finally I plot the denoising of the first 5 days.

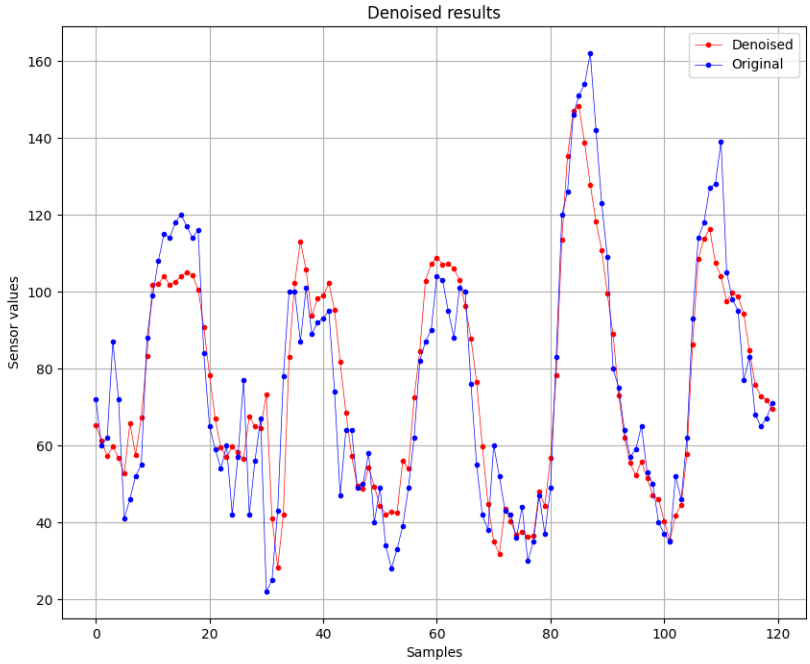

Figure 22: Denoised result and original result of 5 days

In this case the result is very bad, in fact we can see that it is possible a better approximation using some higher ranks.

3 Final comments

In these tests I could compare two different approaches to data denoising. In the first case I used reference station data, this method provides better results because we can compare the denoised dataset obtained with reliable data. Instead, with Gavish *et al.* method, I used only one dataset and tried to denoise it without any comparison with the experimental data.

Although the first impression is that having the reference station data provides better results, the outcomes I obtained are quite similar between the two approaches. Even if have a lot of experimental data is the best way, however this is not always possible in practice, due to too high costs. Despite of that the results show that we can denoise data efficiently even without reference station data.