LASER INTERFEROMETER GRAVITATIONAL WAVE OBSERVATORY
- LIGO -
CALIFORNIA INSTITUTE OF TECHNOLOGY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Technical Note **LIGO-T1900398-v1** 2019/09/09

Analisys of the correlations between the non stationary noise and the auxiliary channels in LIGO

Luca D'Onofrio[1], Gabriele Vajente[2]

[1]Universita' di Napoli Federico II, Complesso Universitario di Monte S.Angelo, I-80126 Napoli, Italy
[2]LIGO, California Institute of Technology, Pasadena, CA 91125, USA

Distribution of this document:

LIGO Scientific Collaboration

Abstract

Noise in LIGO detectors is not stationary. To track the time evolution of the noise we used the Band-Limited Root Mean Square (BLRMS). We analized different frequency bands in different time BLRMS segments for LIGO Livingston and LIGO Hanford O3 data. We used LASSO choosing an appropriate value for the hyper-parameter α to reduce the number of important channels to tens. We will show interesting results for LIGO Livingston in the frequency band $65-76$ Hz and for LIGO Hanford for the frequency band $1120-1400$ Hz.

1 Non stationary noise in LIGO

Data from the LIGO detectors typically contain non Gaussian and non stationary noise which arise due to instrumental and environmental conditions. This non stationary noise implies a time variations of the LIGO detectors sensitivity.
Operating the LIGO detectors at the sensitivity needed for gravitational-wave detection requires a large number of control loops and sensors to precisely control and monitor the instrumentation and environment. The channels that continually record information from those sensors allow to analyze the problematic variations in the sensitivity of the detector. However, due to the complexity of the system, the number of auxiliary sensor channels is around hundreds of thousands. The LIGO auxiliary channels include microphones, seismometers, magnetometers, photo-diodes, voltage monitors, feedback loop signals etc.
Searching through such a large amount of data to determine which sensors are most highly correlated with variations of the noise in the considered frequency range is an important and difficult challenge. Characterizing these correlations from the data and improving the background of LIGO's searches is crucial for reducing non-stationarity in the detectors and increasing the statistical significance of gravitational-wave candidate events.

2 Computation of Band-Limited Root Mean Square

To perform quantitative analysis on the noise floor variations over time, it is useful to compute the Band-Limited Root Mean Square (BLRMS). If $x(t)$ is a time series[1], the mean power density of $x(t)$, also called root mean square value (RMS), is defined as [1]:

$$RMS = \lim_{T\to\infty} \sqrt{\frac{1}{2T}\int_{-T}^{+T} x^2(t)dt} \tag{1}$$

that is linked to the auto-correlation function at zero delay. The power spectral density (PSD) is defined as the Fourier transform of the auto-correlation function R_{xx}:

$$PSD(\omega) = \int_{-\infty}^{+\infty} e^{-i\omega\tau} R_{xx}(\tau)d\tau \tag{2}$$

[1]We are considering ergodic process. A stochastic process is said to be ergodic if its statistical properties can be deduced from a single, sufficiently long sample of the process.

Table 1: Frequency bands for the computation of the BLRMS for the LIGO Hanford detector.

Starting frequency [Hz]	End frequency [Hz]	Starting frequency [Hz]	End frequency [Hz]
10	13	96	116
23	26	111	114
26	29	116	174
28	31	174	186
31	44	186	286
44	56	316	400
46	47	440	495
47	49	521	583
56	62	630	823
62	86	870	970
86	96	1120	1400
88	90		

where

$$R_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{2T}\int_{-T}^{+T} x(t+\tau)x(t)dt\,. \tag{3}$$

The physical units of the power spectrum are $[x]^2/Hz$ where $[x]$ is the physical unit of the considered time series. The BLRMS values of a signal in a frequency band $[\omega_1, \omega_2]$ is simply the integral of the PSD in that band:

$$BLRMS = \sqrt{\int_{\omega_1}^{\omega_2} PSD(\omega)d\omega}\,. \tag{4}$$

Using the PSD of LIGO's detectors, if the computation of the BLRMS is performed periodically it is possible to track the time evolution of the noise.
In this way, considering n frequency bands and a time T as the sampling rate to compute the BLRMS, the results are n new signals which give the noise power in each band as a function of time [2]. It is possible to obtain these signals at a larger sampling rate using overlapping time intervals for the computations. This is useful to have a better frequency resolution.
Once the n frequency bands have been selected, the time evolution of the BLRMS can be analyzed. In this work the chosen sampling rate is $T = 2.5$s while the chosen bands are in Table 3. For the computation of the PSD, we use the fast Fourier transform (FFT) with a time window of 5 seconds, scale it in units of PSD, integrate to get the BLRMS and then we move forward by 2.5 seconds and repeat the computation [3].
In time evolution of BLRMS, *glitches* could be observed. A glitch is a high-amplitude, short-duration excursion of unknown origin in the calibrated strain signal, which is what we use to compute the BLRMS. To identify glitches, for each sample in time we computed the median value and a dispersion (that is a sort of standard deviation of the data, but using median instead of mean) in a chosen time window. In this work we chose a time range of 40 seconds. We used the median value and not the mean value because the former is less dependent to the value of the glitch. It is possible to put a threshold (median plus three times the dispersion) on the BLRMS value to identify glitches and reject them using an interpolated value (see Figure 1).

Table 2: Frequency bands for the computation of the BLRMS for the LIGO Livingston detector.

Starting frequency [Hz]	**End frequency [Hz]**	**Starting frequency [Hz]**	**End frequency [Hz]**
10	13	115	190
18	22	190	210
22	27	210	290
27	29	290	480
29	40	526	590
40	54	590	650
54	65	650	885
65	76	885	970
75	115	1110	1430

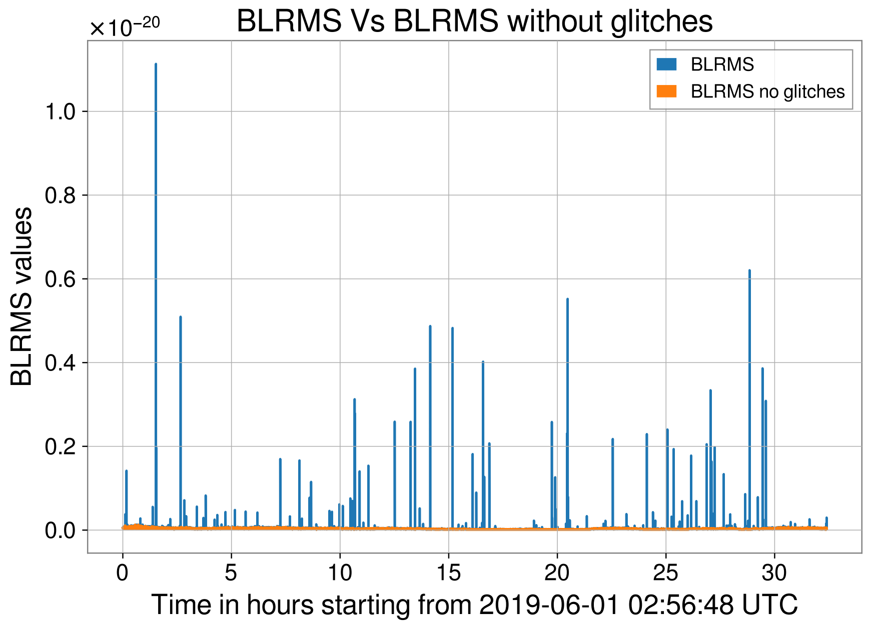

Figure 1: Plot of the BLRMS time evolution in the frequency band 47 – 49 Hz. The blue line shows the BLRMS time evolution loading from the available data while the orange line shows the same evolution excluding glitches using the described procedure. Orange line is better shown in Figure 2.

Using a similar procedure, it is possible to identify *lines*, that are strong sinusoidal contribution at a given frequency in the PSD. The BLRMS of a band containing a line might be completely dominated by the power in the line itself [1]. Therefore every fluctuation of the noise floor is hidden by the line amplitude.
Since the aim is to analyze the non stationary noise floor on long time scales (approximately days), glitches and lines are not relevant in this context.
The first step toward finding correlation between the noise variation and auxiliary channels is to compute the time evolution of the BLRMS for a certain frequency band without glitches and lines. This has already been computed with the code described in [3] and available in [2]. The next step is the use of statistical methods to find potential correlations, for example applying a multi-dimensional linear regression.

3 Multi-dimensional regression

The values at n different times y_i of the BLRMS in a given band can be viewed as the noisy version of a linear combination of the k auxiliary channels measured at the same times $x_{1\cdots n,1\cdots k}$ [4] where the first index runs over time samples and the second index over all auxiliary channels,

$$y_i = \beta_1 x_{i,1} + \cdots + \beta_k x_{i,k} + \varepsilon_i\,, \tag{5}$$

that can be rewritten as:

$$y = X\beta + \varepsilon \tag{6}$$

where

$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix} \qquad X = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i,1} & x_{i,2} & \cdots & x_{i,k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,k} \end{pmatrix} \qquad \beta = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_i \\ \vdots \\ \beta_k \end{pmatrix}\,. \tag{7}$$

The unknown coefficients β have to be estimated and ε is a source of additional noise, assumed to be normally distributed and considered as coming from the measurement of the y or also from the auxiliary channels.
The best estimation of the coefficients is obtained by minimization of the sum of squared errors S:

$$S = (y - \beta X)^T (y - \beta X) = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{k} \beta_j x_{i,j} \right)^2 \tag{8}$$

and if all the auxiliary signals are linearly independent, there is a unique solution. The value of β that minimizes S is found by differentiating the function S with respect to β and setting the result to zero. This gives the condition,

$$\frac{\partial S}{\partial \beta} = 2\beta^T X^T X - 2y^T X = 0\,, \tag{9}$$

and so, for the coefficients β [5]:

$$\hat{\beta} = (X^T X)^{-1} X^T y. \tag{10}$$

Equation (10) is sometimes called *"normal equation"*.
The advantage of this method to estimate the coefficients is that it is guaranteed to find the optimal solution analytically. This method is also useful to understand which amount of the total noise non-stationarity can actually be predicted from a given set of auxiliary channels.
However, if the datasets is large, it can be computationally too expensive to invert the matrix in this formula or simply, the matrix $X^T X$ may be singular (non-invertible). To avoid this problem, we compute the pseudo-inverse of the matrix X. The pseudo-inverse is equal to the matrix $(X^T X)^{-1} X^T$ if the columns of X are linearly independent. Otherwise it gives one of the possible solutions of the least square problem.
It is also easy to show that in some cases, standard linear regression can fail. We used simulated signals to analyze these cases. Chosen signals are random time series with 3600 points that corresponds to 60 hours for an auxiliary channel with sampling time of 1 minute. Using a low pass filter, we removed the high frequencies from these signals in order to reproduce the slow time evolution of the auxiliary channels. Choosing randomly twenty coefficients, we defined a new signal, the output signal, combining linearly the first twenty signals. A standard linear regression can be performed to compare the estimated coefficients to the real ones. If some Gaussian background noise is added to the input signals, the regression coefficients are close to the real ones, but also other coefficients, that are not linked to the output signal, can get large. The reconstrution is good, but it's hard to decide which of the input signals are relevant.
In addition, we simulated another random signal with the same characteristics of the input signals and we used it as output signal. In order to check for possible correlations with the input signals, we performed a linear regression and a good recontruction can be obtained if the number of considered signals is large enough. This is the most spectacular failure since we know there is no correlation at all. Linear regression is able to find a combination that does not exist due to the large number of signals.
Because of the great number of auxiliary channels in the LIGO detector, this problem of "overfitting" must be considered.

4 Auxiliary channels

To achieve the required sensitivity for detecting gravitational waves, it is necessary to identify, control and eventually remove environmental disturbances. Therefore, the LIGO interferometers are equipped with thousands of sensors, the *auxiliary channels*, to monitor instrumental and environmental conditions.
The LIGO channels shown in this work have long and complicated names, using a series of abbreviations to specify the exact location, subsystem, and type of sensors being recorded, according to a convention described in [6]. Channel names follow this standard:

- Characters 1 and 2 show the IFO (interferometer) the channel belongs to (H1 for LIGO Hanford, L1 for LIGO Livingston, V1 for Virgo, G1 for GEO600).

Table 3: Principal subsystems of LIGO detectors used in this work.

ASC	alignment sensing and control
LSC	length sensing and control
PEM	physical environment monitor
PSL	prestabilized laser
SEI	seismic isolation
SUS	suspension
IMC	input mode cleaner
ISI	internal seismic isolation
HPI	hidraulic external pre-isolation
SQZ	squeezed light

- Character 3 = : (colon)
- Characters 4,5,6 show which subsystem the channel belongs to.
- Character 7 = - (hyphen)
- The rest of the name describes the channel

For example: H1:ASC−AS_D_DC_YAW_OUTPUT is the channel which samples the DC signal (signal coming from the low frequency power) for the D photodiode yaw rotation at the antisymmetric port of the Hanford 4km interferometer. In the table 3, there is a list of subsystems of LIGO detectors used in this work.

5 Overfitting problems

In order to find a correlation between the BLRMS time evolution and the auxiliary channel time series using linear regression as a first step, we considered the data segment from 1243393026 GPS time to 1243509653 GPS time (around 33 hours) in the frequency band 47-49 Hz for LIGO Hanford detector. We chose this band because there is a bump that we would like to characterize [7].
In this first part, only the ASC subsystem (Alignment Sensing and Control) channels are selected. For this subsystem there are 2331 channels. Since the auxiliary channels time series are fast signals, we chose to consider the minute trends of these channels. These trends are time series with sampling time of 1 minute. For a particular channel, the corresponding trend is obtained computing the mean and the root mean square over 60 seconds of the channel time series. From each mean value M_i and root mean square value RMS_i, it is possible to compute the minute trend of the standard deviation SD_i:

$$SD_i = \sqrt{(RMS_i)^2 - M_i^2}\,. \tag{11}$$

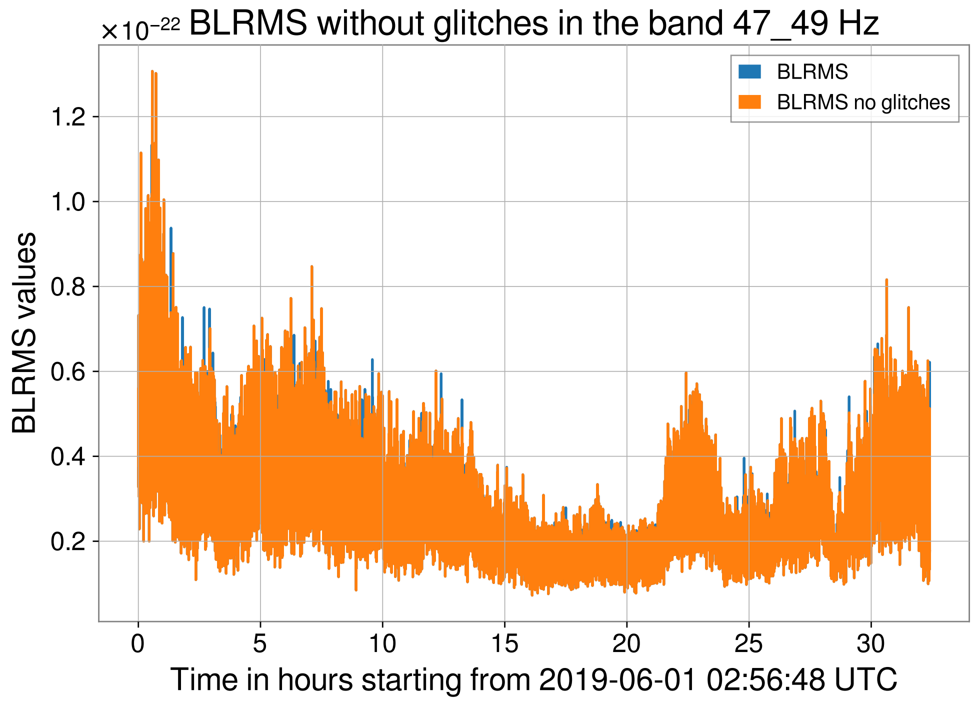

Figure 2: Plot of the BLRMS time evolution without glitches. The blue line shows the BLRMS time evolution after the procedure to remove glitches while the orange line shows the same evolution excluding remaining glitches using again the same procedure with a time range of 40 seconds. In this case, there are no remaining glitches and the two plots are almost coincident.

We use standard deviation because it is the root mean square of a signal's variation about the mean, rather than about zero.
Possible glitches in the auxiliary channels' time series could negatively influence the linear regression. To identify glitches in the mean and the standard deviation time evolution of each channel, we applied the method used for the BLRMS values.

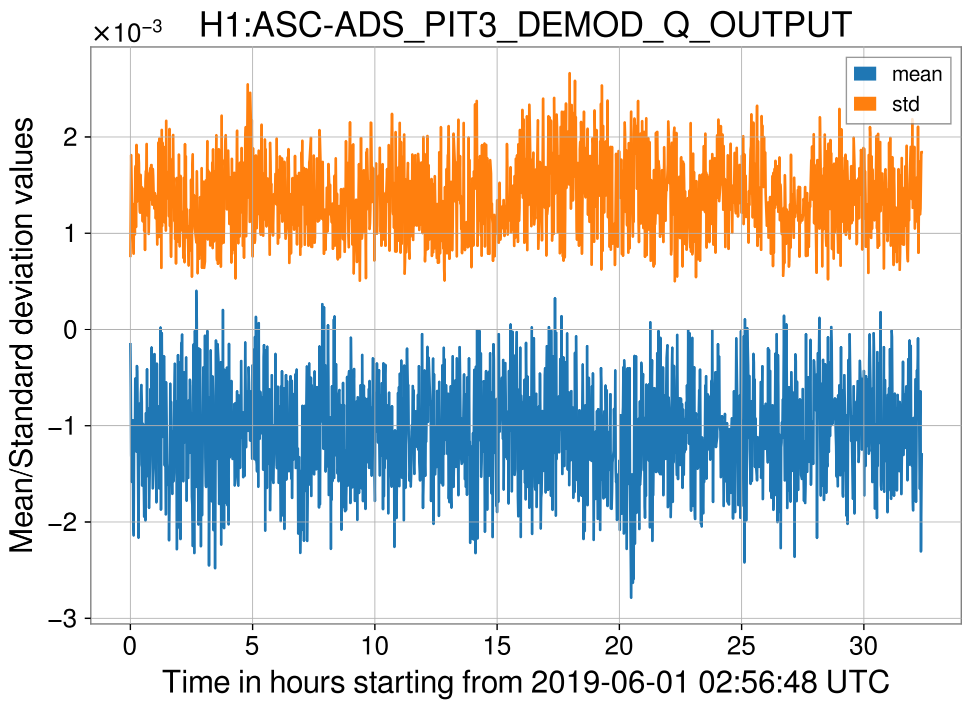

Figure 3: Plot of mean and standard deviation for the channel H1:ASC−ADS_PIT3_DEMOD_Q_OUTPUT, chosen for example, in the selected time range.

Since the sampling time for the auxiliary channels is 1 minute, to perform linear regression we need to resample the BLRMS data to the times of the auxiliary channels. We performed this resample using the average of the BLRMS in an interval of size 60 seconds centered around the considered auxiliary times.
Using the resampled BLRMS data and the auxiliary channel time series, it is possible to construct the y values and the matrix X of the equation 10. To estimate the β coefficients we used the function *pinv()* from the numpy python package [8] that calculates the pseudo-inverse of a matrix using its singular-value decomposition and including all large singular values.
In order to check for possible overfitting problems we separated the BLRMS dataset in a training set and in a test set of equal dimension. Then, we standardize the time series values by removing the mean and scaling to unit variance. For each channel and for the BLRMS, the mean and the variance are computed in the training set.
In the training set the β coefficients are estimated (see Figure 5) and used to predict the BLRMS values. In this way it is possible to compare the predicted values with the training and with the test set and plot the residuals. The residual plot, showed in Figure 6 and 7, is the plot of the difference between the BLRMS values in the test set and the predicted values. Ideally, the residual should follow the same distribution as the added noise (ε in equation 6) which in real life situations is not

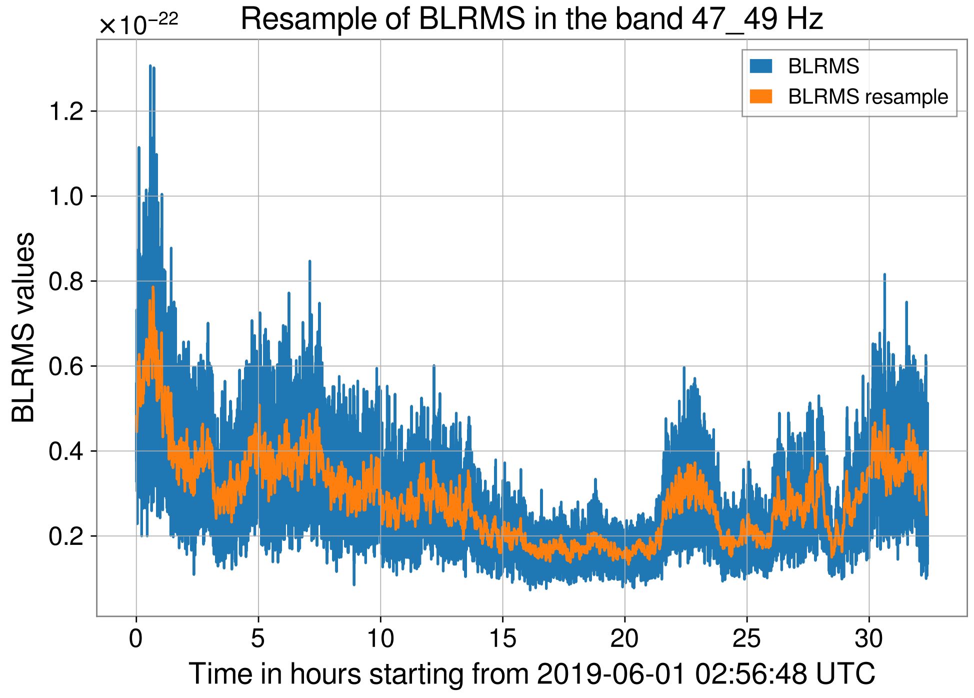

Figure 4: Plot of the resampled BLRMS. The blue line is equal to the orange line in Figure 2. The orange line shows the BLRMS time evolution with a sampling time of 60 seconds.

measurable.
In case of overfitting, it is expected that the predicted values fit almost exactly the training set and fail to predict the test set. In other words, overfitting means that the model captures the patterns in the training data, but fails to generalize well to unseen data [9]. If a model suffers from overfitting, we also say that the model has a high variance, which can be caused by having too many parameters that lead to a model that is too complex given the underlying data.
According to Figure 6 and 7, the linear regression exactly fits the training test but the prediction on the test set is totally wrong. There is no relation between the BLRMS time evolution for this time range and this band but linear regression works perfectly on the training set anyway. The linear combination is the simplest model and the main problem is the great number of auxiliary channels used to fit the time evolution of the BLRMS. It is important to underline that, for now, we are just considering a small subset of the available auxiliary channels. Adding more channels would make the situation even worse.

6 Test of possible solutions using simulated data

In Section 3, we described overfitting problems using a simulated dataset. Using a similar dataset we can test different regression methods, comparing directly the estimated coefficients with the real ones and reducing computational cost.
The chosen dataset is composed by 20000 random time series $x_i(t)$ (input signals), each with 3600

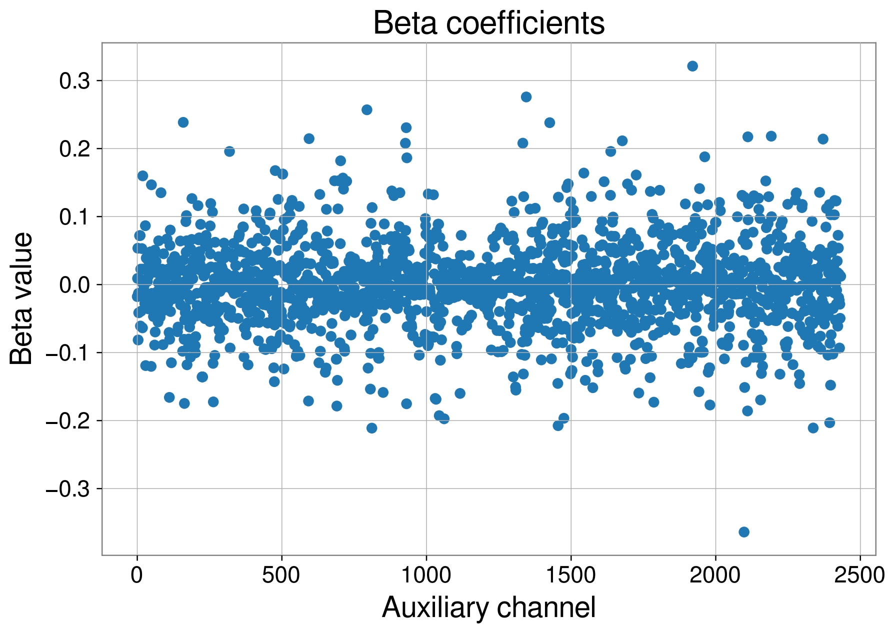

Figure 5: Plot of beta coefficients computed using linear regression for the training set. On the x axis, there is the index of ASC subsystem channel corresponding to the β value.

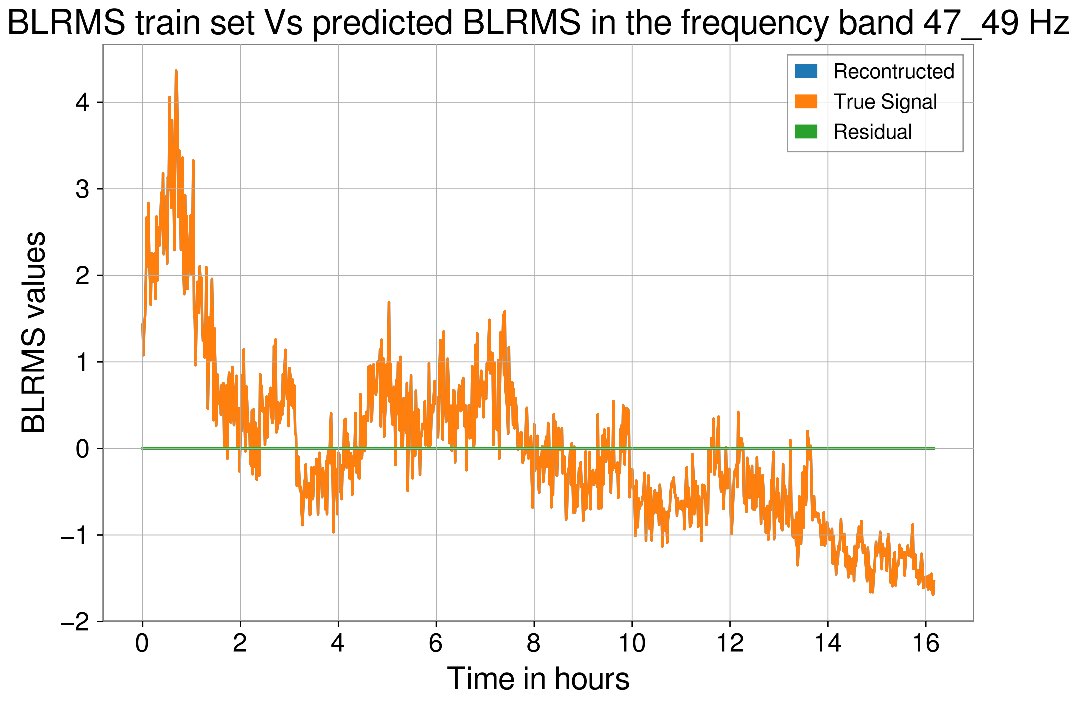

Figure 6: Plot of the predicted values and of the residuals for the train set.

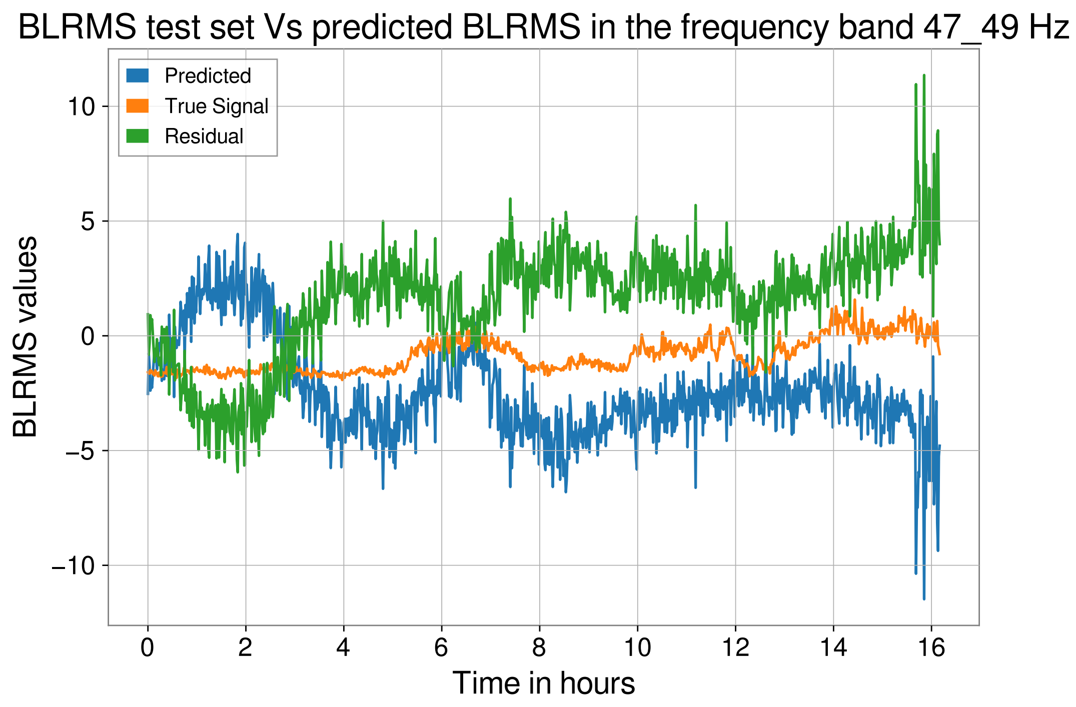

Figure 7: Plot of the predicted values and of the residuals for the test set.

points that corresponds to 60 hours for an auxiliary channel with sampling time of 1 minute. Using a low pass filter, we removed the high frequencies from input signals in order to simulate the slow time evolution of the auxiliary channels. We also added Gaussian background noise with zero mean and standard deviation $\sigma = 0.05$ to the input signals.
Choosing randomly twenty coefficients a_j, we defined a new signal $y(t)$ (output signal) combining linearly the first twenty input signals:

$$y(t) = \sum_{j=1}^{20} a_j x_j(t) + \varepsilon(t)\,. \tag{12}$$

The output signal is a known combination of the input signals. In order to evaluate a particular regression method, we can compare the real coefficients that for simulated data are known with the estimated coefficients.
Input signals and output signal are splitted in training set and test set equally distribuited. For each input signal and for the output signal, mean and standard deviation are computed in the training set and then used to scale to zero mean and unity standard deviation the time series (this operation is performed using the class StandardScaler of the package Sklearn [10]).
We found that LASSO Regression is the best method to improve the predictions reducing the number of features that, in our case, is the number of channels linked to the time evolution of the noise.

6.1 LASSO regression

LASSO, or Least Absolute Shrinkage and Selection Operator, is a linear regression model that performs L1 regularization[2]. The ”loss” function to minimize in the case of L1 regularization is the sum of the function S in (8) whit a penalty term equal to the absolute value of the magnitude of coefficients:

$$S_L = \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{k} \beta_j x_{i,j} \right)^2 + \alpha \sum_{j=1}^{k} |\beta_j| \,. \tag{13}$$

Figure 8 is contour plot of Lasso regression objective function. The elliptical contour plot represents the sum of square error term. The diamond shape in the middle indicates the constraint region. The optimal point is the common point between ellipse and diamond because gives a minimum value for the above function. There is a high probability that the optimum point falls in a corner point of diamond region. For the 2-dimensional case as in Figure 8, if an optimal point falls in the corner point, it means that one of the feature’s estimate is zero. Lasso regression helps for feature selection.

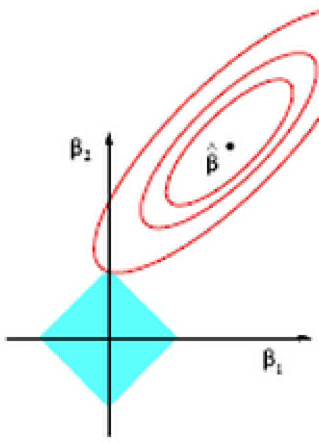

Figure 8: A geometrical 2-dimensional example of how LASSO sets certain auxiliary channel coefficients to zero [11].

It is well known [12] that more generally the L1 regularization term shrinks many components of the solution to zero, and thus performs feature selection. The number of selected features is controlled by the hyper-parameter α. α is a tunable parameter that penalizes overfitting by reducing or eliminating the contributions from unnecessary input variables.

[2]Ridge regression uses instead L2 regularization with a penalty term equal to the squared absolute value of the magnitude of coefficients. Ridge regression merely reduces the values of the coefficients of the less desirable explanatory variables [11].

6.2 Results

The hyper-parameter α controls the number of coefficients that are shrinked to zero. In Figure 9 there is the search for the best α value (α_{best}) for a simulated dataset. To compare LASSO performances with different value of α we used the Mean Squared Error (MSE) as metric. MSE is simply the averaged value of the sum of the squared errors:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y^{(i)} - \hat{y}^{(i)})^2, \tag{14}$$

where N is the number of samples, $y^{(i)}$ is the original value and $\hat{y}^{(i)}$ is the estimated value [9]. MSE is equal to the quantity S defined in equation 8, except for a constant.
α_{best} value is the value of α with the minimum MSE value for the test set.

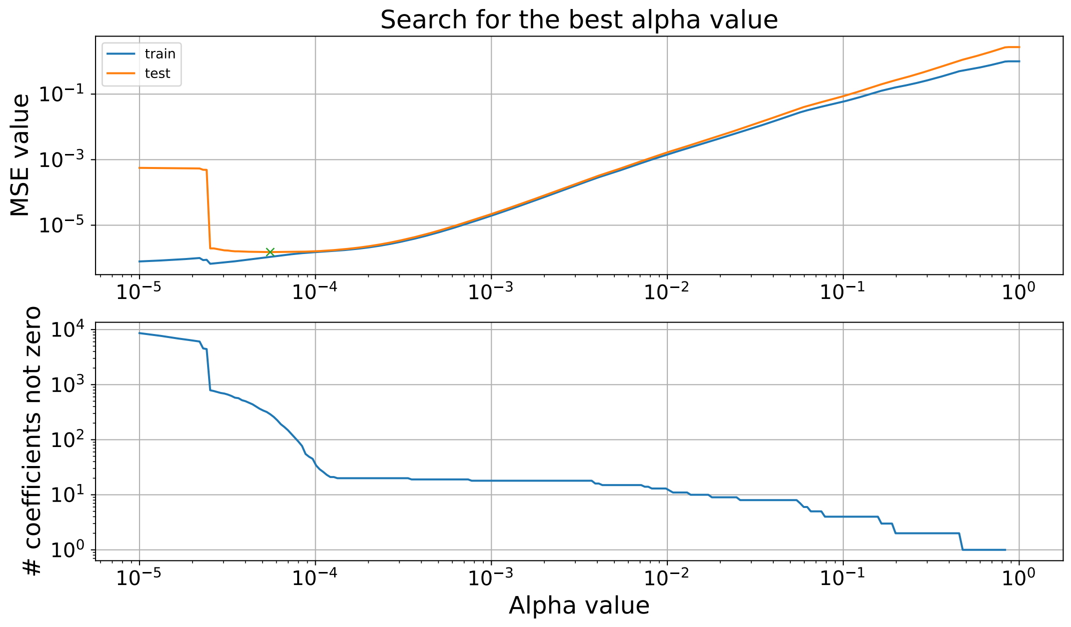

Figure 9: Search for α_{best} value for a simulated (according to 6) dataset and for 250 values of α in the range $10^{-5} \leqslant \alpha \leqslant 1$. In the top plot, in blue there are the values of MSE in the training set while in orange, the values of the MSE for the test set. The green " $\times$ " indicates the minimum MSE value for α_{best}. In the bottom plot, for each value of α it is shown the number of coefficients different from zero selected by LASSO.

Figure 9 shows that, for the chosen simulated dataset, the number of coefficients different from zero selected by LASSO using α_{best} is 290. In this case we know that the number of coefficients actually different from zero is 20. We can reduce the number of coefficients choosing a value of α near to the best one.
In the example in Figure 9, for $\alpha = 10^{-4}$ the value of MSE is almost equal (under 5%) to the MSE for α_{best} but the number of coefficients is 35 reduced by a factor ten.
Since LASSO selects also input signals that have no correlation with the output signal [3] we need a

[3]For our simulated dataset we know that only the first twenty input signals are real.

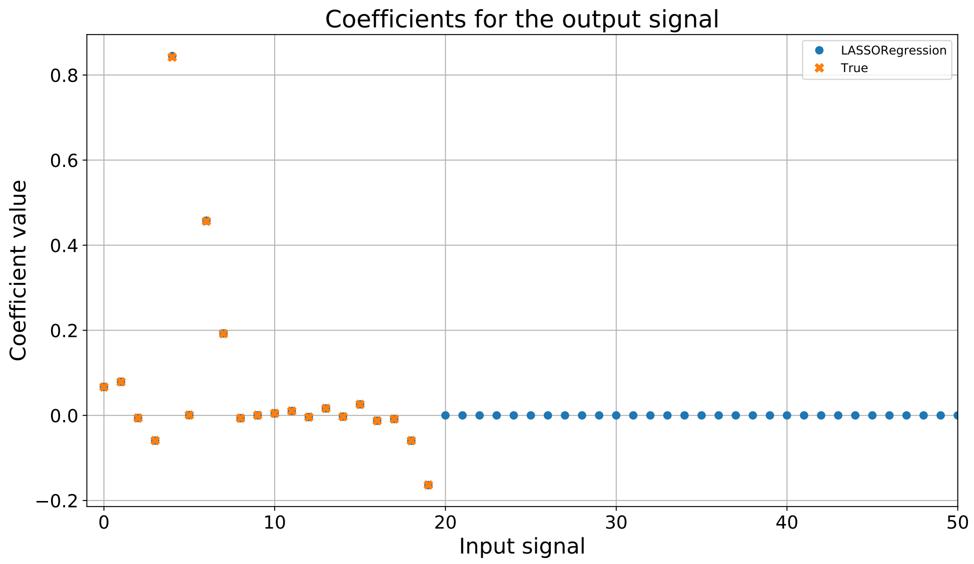

Figure 10: Plot of the values of the predicted coefficients by LASSO in blue and of the real coefficients in orange for the first 50 input signals.

Table 4: ε scores for $\alpha = 10^{-4}$. i is the index of the input signal.

i	ε_i
0	0.998
1	0.998
...	...
4	0.135
5	0.999
6	0.908
7	0.984
...	...
18	0.998
19	0.990
...	...
16793	1.000
17433	1.000
19230	1.000

way to quantify the importance of each channel. For this purpose we defined the ε score as:

$$\varepsilon_i = \frac{||y - \hat{\beta}_i x_i||}{||y||} \tag{15}$$

where i is the index of the input signal. ε_i is the normalized residual for the i-th input signal.
Figure 10 shows that the predicted values by LASSO for the first 20 input signals are almost equal to the real ones. Since simulated signals have the same range of variation, input signals with large value of the respective coefficient are more ”important” to reconstruct the output signal.
In Table 4, we report some of the ε scores for the 35 input signals with coefficient different from zero selected by LASSO with $\alpha = 10^{-4}$. As we expect for simulated data, the lowest values of ε correspond to the largest values for the coefficients.

7 O3 data

In the first example, described in Section 5, we showed overfitting problems of the standard linear regression using a 33-hour long BLRMS segment in the frequency band 47 – 49 Hz for the LIGO Hanford detector. We splitted this segment in two equal parts and used the first part as training set and the second part as test set.

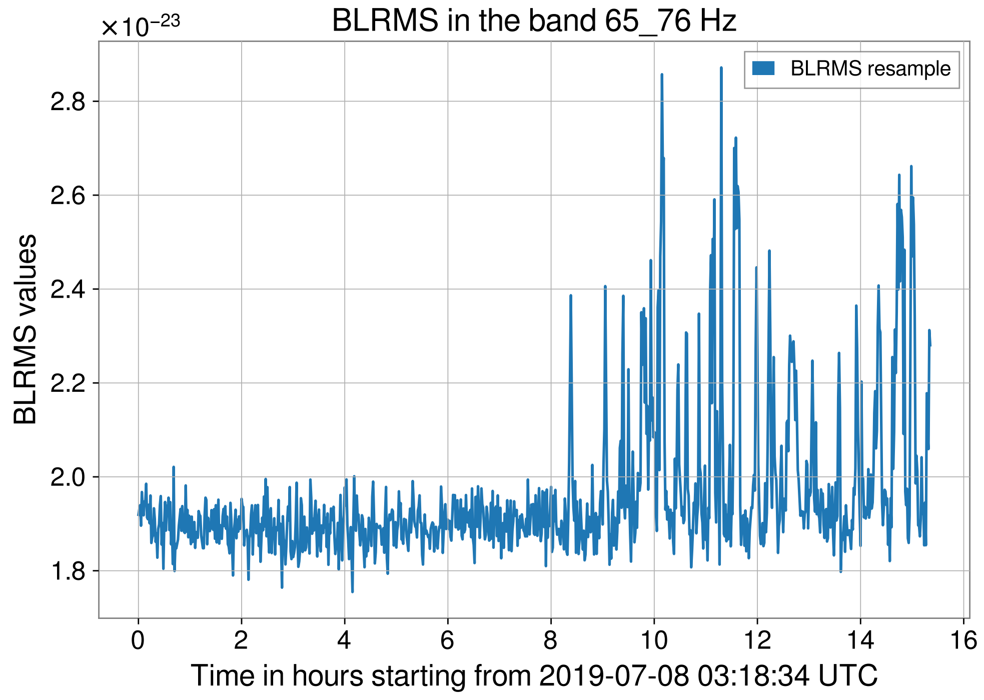

Figure 11: Plot of the BLRMS time evolution for a 16-hour long segment for LIGO Livingston detector in the frequency band 65 – 76 Hz.

In the case of Figure 11, we can not use the same strategy because the time evolution of the BLRMS is almost flat in the training set. For this shorter segment (and in other similar cases) we need a new strategy. We decide to consider two different segment, approximately of the same length and consecutive, for the training and for the test set.

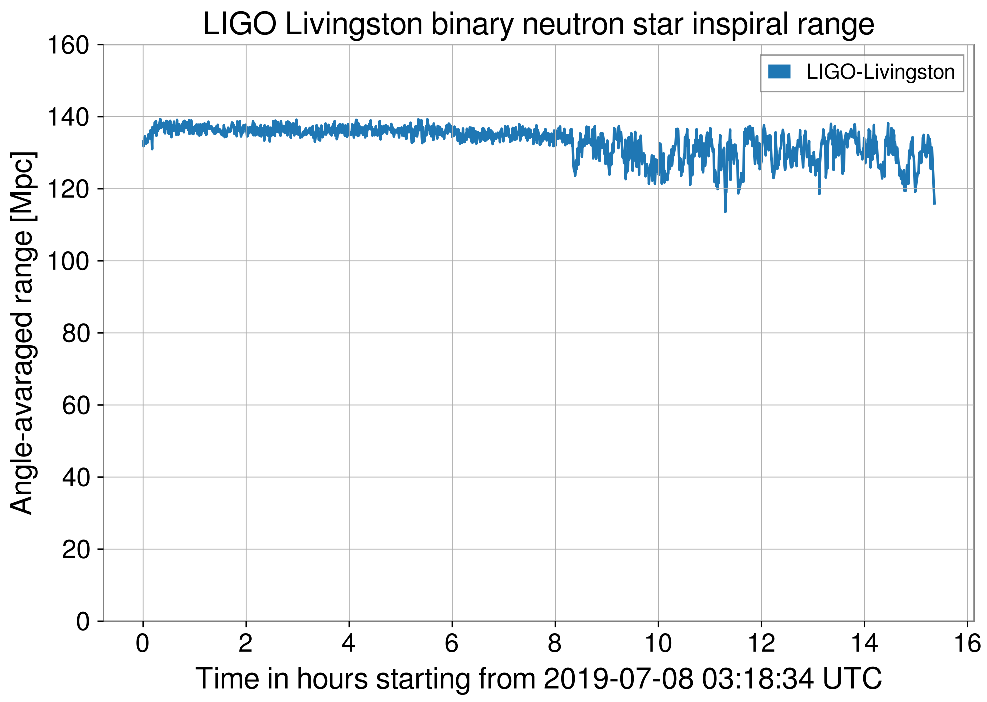

Figure 12: Plot of the binary neutron star inspiral range for LIGO Livingston detector for the time period of the BLRMS segment in Figure 11.

Figure 11 is also intersting because it shows a common behaviour of the LIGO Livingston detector of the last months; in the daily hours, there is a worsening of the sensitivity as it is also shown in Figure 12. The Binary Neutron Star (BNS) inspiral range, plotted in Figure 12, is defined as [13] the distance at which a single detector could observe the coalescence of a pair of 1.4 solar mass neutron stars with a signal-to-noise ratio of 8. BNS range varies significantly on minute and hourly timescales and is often used to quantify the variation of the noise in LIGO detectors as in [11]. The decrease in the BNS range in 12 corresponds to the increase of the BLRMS values in 11 in the frequency band $65-76$ Hz. It is important to underline that BNS range is not linked to a particular frequency band.
We analized different frequency bands in different time segments for LIGO Livingston and LIGO Hanford O3 data. We used LASSO choosing an appropriate α value to reduce the number of important channels to tens. In the next paragraphs, we will show interesting results for LIGO Livingston in the frequency band $65-76$ Hz and for LIGO Hanford for the frequency band $1120-1400$ Hz.

7.1 LIGO Livingston: 65-76 Hz

Firstly, we report results for two long segments (as in Figure 13) that are splitted in training and test set.

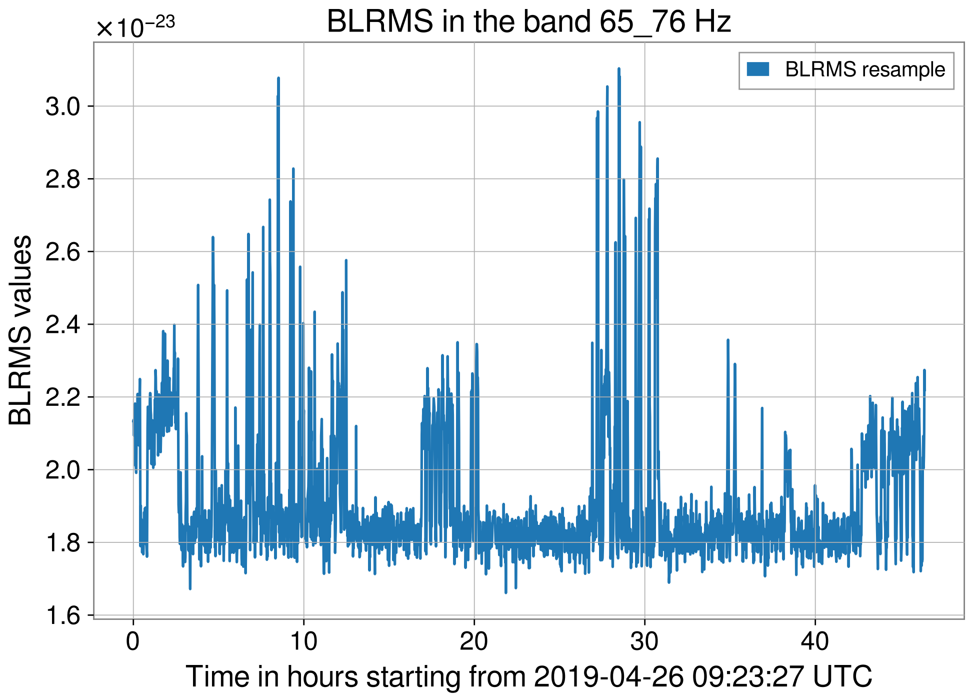

Figure 13: BLRMS time evolution starting from 1240305825 GPS time in the frequency band $65-76$ Hz for LIGO Livingston.

For the first segment we report also the search for α_{best} (see Figure 14). In order to reduce the number of relevant channels we chose $\alpha = 0.1$. In this case LASSO selects 18 channels and succeeds to predict several bumps in the BLRMS evolution. The values for the ε score for the first segment are reported in Table 5. In bold there are the two channels with the lowest ε score.
These channels are an accelerometer (PEM-CS_ACC_HAM6_OMC_X_MON) and a sismometer (HPI-HAM6_BLND_L4C_VP_INMON) that monitor the ground movement in the HAM6 vacuum chamber.
In the case of the second segment (Figure 16) LASSO selects 17 channels and again the two channels with the lowest ε scores are the same of the previous case (PEM-CS_ACC_HAM6_OMC_X_MON $\varepsilon = 0.801$ and HPI-HAM6_BLND_L4C_VP_INMON $\varepsilon = 0.643$)).
The two segments analyzed are one month away but the two most correlated channels are the same and linked each other.
To test this results we also considered two couples of shorter consecutive BLRMS segments. In the case of Figure 18 there is a good reconstruction using 24 chnnels. The two channels with lowest ε score are again the same described for the previous segments (PEM-CS_ACC_HAM6_OMC_X_MON $\varepsilon = 0.942$ and HPI-HAM6_BLND_L4C_VP_INMON $\varepsilon = 0.953$)).
In the case of Figure 17 the time series using 16 channels selected by LASSO does not predict the high variation in the test set. In these 16 channels (reported in Table 6 with the re-

spective ε score), the two HAM6 channels are not present. We decided to perform a standard linear regression using just the time series of PEM-CS_ACC_HAM6_OMC_X_MON and HPI-HAM6_BLND_L4C_VP_INMON.
Figure 19 shows that the standard linear regression using only these two channels can predict some bumps but not all the variation in this segment.

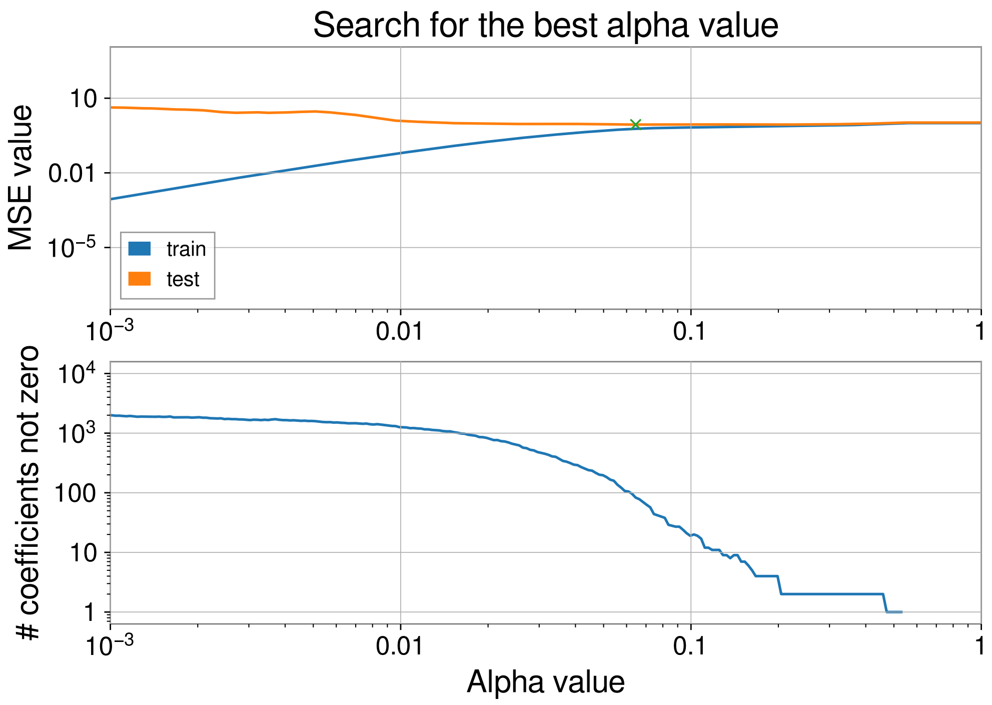

Figure 14: Search for α_{best} value for BLRMS segment starting from 1240305825 GPS time in the frequency band $65-76$ Hz for LIGO Livingston (Figure 13) and for 500 values of α in the range $10^{-3} \leqslant \alpha \leqslant 1$. In the top plot, in blue there are the values of MSE in the training set while in orange, the values of the MSE for the test set. The green " $\times$ " indicates the minimum MSE value for α_{best}. The bottom plot shows, for each value of α, the number of coefficients different from zero selected by LASSO.

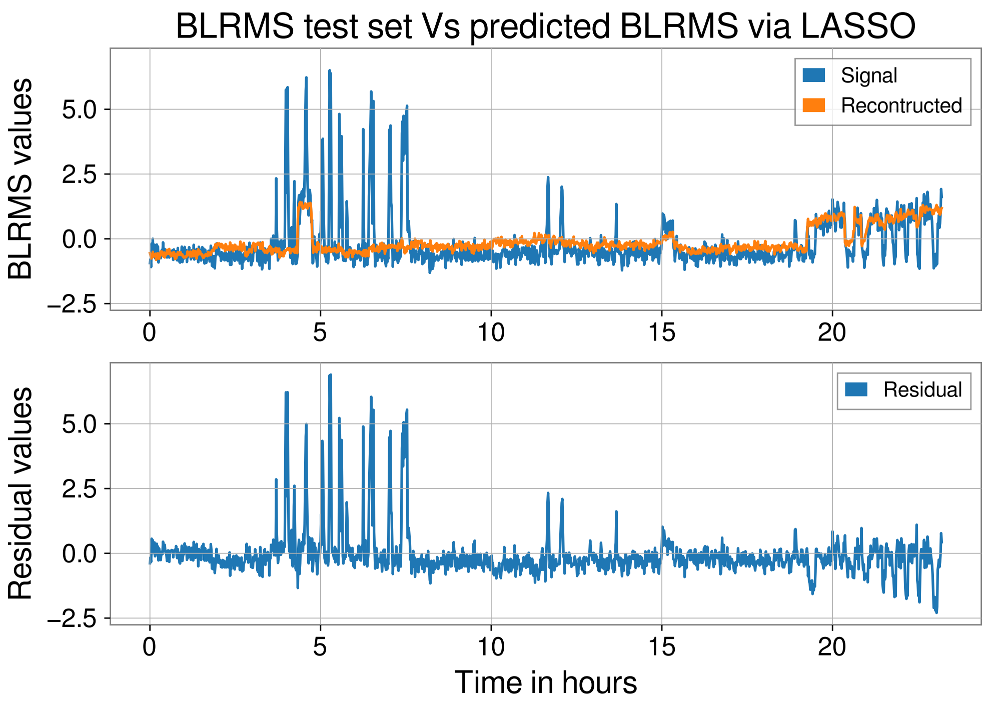

Figure 15: Plot of the predicted values and of the residuals for the test set of the BLRMS segment starting from 1240305825 GPS time in the frequency band 65 – 76 Hz for LIGO Livingston (Figure 13) using LASSO. In the top plot, the blue line is the time evolution of the BLRMS for the test set while the orange line is the predicted evolution using 17 channels selected by LASSO. In the bottom plot the difference between the real signal and the predicted signal is shown.

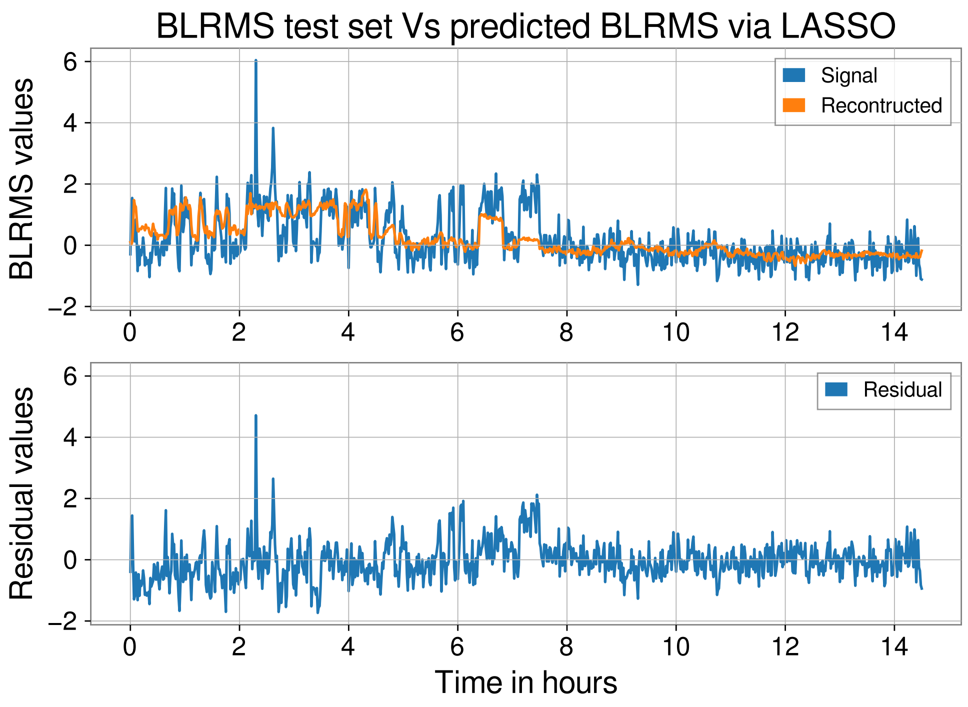

Figure 16: Plot of the predicted values and of the residuals for the test set (BLRMS segment starting from 1242787557 GPS time or May 25, 2019 02:45:39 UTC for LIGO Livingston) using LASSO. In the top plot, the blue line is the time evolution of the BLRMS for the test set while the orange line is the predicted evolution using 17 channels selected by LASSO. In the bottom plot the difference between the real signal and the predicted signal is shown.

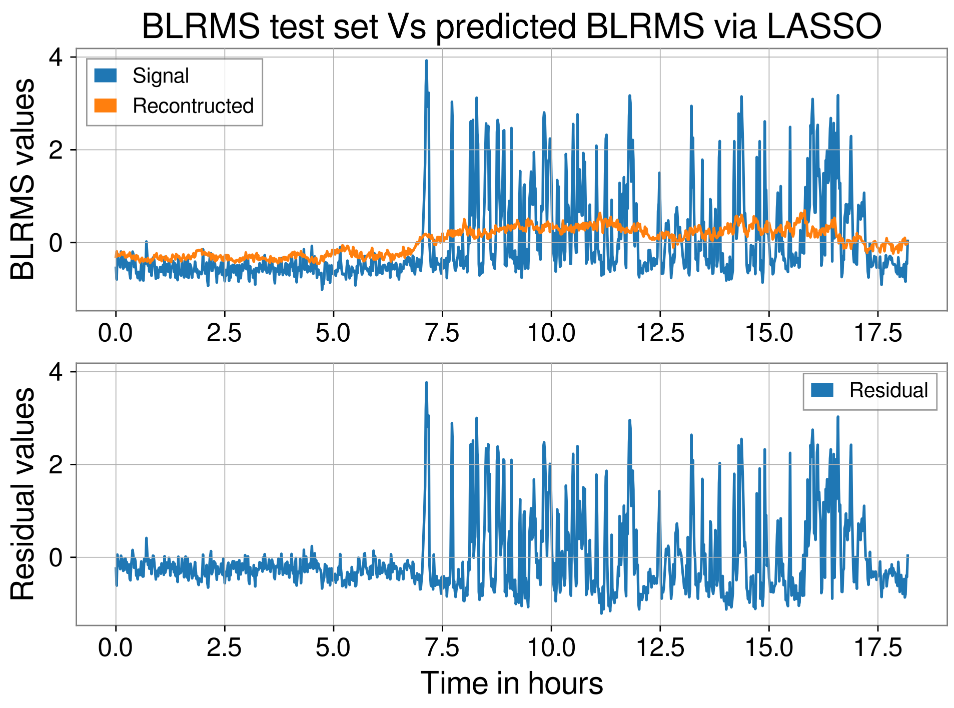

Figure 17: Plot of the predicted values and of the residuals for the test set starting from 1244608786 GPS time (Jun 15, 2019 04:39:25 UTC) and training set starting from 1244522008 GPS time (Jun 14, 2019 04:33:10 UTC) for LIGO Livingston. In the top plot, the blue line is the time evolution of the BLRMS for the test set while the orange line is the predicted evolution using 16 channels selected by LASSO. In the bottom plot the difference between the real signal and the predicted signal is shown.

Table 5: ε scores for channels selected by LASSO (Figure 15).

Channel name	ε score
L1:PEM-CS_ACC_HAM6_OMC_X_MON.std	0.843
L1:HPI-HAM6_BLND_L4C_VP_INMON.std	0.915
L1:ISI-HAM5_ISORAMP_RX_INMON.std	0.990
L1:ASC-POP_X_RF_I_PIT_NORM.mean	0.996
L1:LSC-REFLAIR_A_RF45_I_OUTPUT.mean	0.997
L1:ISI-HAM5_FF_RX_INMON.std	0.997
L1:PEM-EY_SEIS_VEA_FLOOR_Y_BLRMS_300MHZ1000.mean	0.998
L1:ISI-ETMX_ST2_ISO_Y_INMON.std	0.998
L1:ISI-ETMY_ST1_GNDSTSINF_C_Y_INMON.std	0.998
L1:SUS-ITMY_L1_WD_OSEMAC_UL_RMSMON.mean	0.999
L1:HPI-HAM2_BLRMS_RZ_30M_100M.mean	0.999
L1:HPI-ETMX_ISO_X_OUTPUT.std	0.999
L1:TCS-ITMX_CO2_PZT_MON_INMON.mean	1.000
L1:TCS-ITMX_CO2_PZT_MON_OUTPUT.mean	1.000
L1:HPI-ETMX_TWIST_FB_X_INMON.std	1.000
L1:ISI-ETMY_ST1_GNDSTSINF_C_Y_OUTPUT.std	1.000
L1:ISI-HAM5_ISORAMP_RX_OUTMON.std	1.000
L1:SEI-SEISMON_LLO_R50_ARRIVALTIME_MINS_2.std	1.000

Table 6: ε scores for channels selected by LASSO (Figure 17).

Channel name	ε score
L1:PEM-EX_SEIS_VEA_FLOOR_Z_BLRMS_1HZ3.mean	0.907
L1:SUS-ITMY_R0_DAMP_R_INMON.mean	0.924
L1:LSC-POP_A_RF9_I_INMON.std	0.959
L1:SUS-ETMX_R0_NOISEMON_F2_INMON.std	0.974
L1:ISI-HAM2_GS13INF_H2_OUTPUT.std	0.985
L1:HPI-BS_BLRMS_RX_30M_100M.mean	0.986
L1:SUS-SR3_M1_WD_OSEMAC_LF_RMSMON.std	0.994
L1:SUS-ITMY_LKIN_Y_DEMOD_I_INMON.std	0.998
L1:ISI-HAM3_BLND_RZ_GS13_CUR_INMON.mean	1.000
L1:ISI-HAM3_BLND_RZ_GS13_NXT_INMON.mean	1.000
L1:ISI-HAM3_CAL_CART_RZ_INMON.mean	1.000
L1:LSC-POPAIR_A_RF9_DEMOD_RFMON.mean	1.000
L1:SUS-SR3_M3_OSEMINF_LR_OUTPUT.mean	1.000
L1:TCS-SIM_ITMY_SUB_DEFOCUS_RH_LF_OUTPUT.mean	1.000
L1:SQZ-LASER_IR_DC_DCCURRENT.std	1.000
L1:SUS-ETMX_R0_NOISEMON_F2_OUTPUT.std	1.000

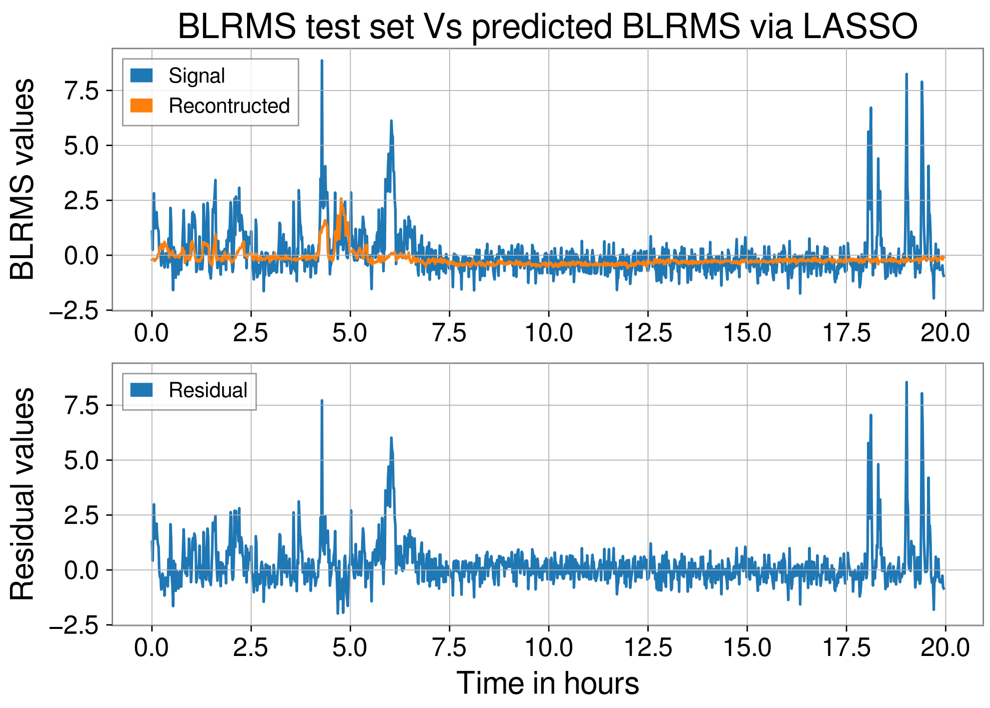

Figure 18: Plot of the predicted values and of the residuals for the test set starting from 1247850447 GPS time (Jul 22, 2019 17:07:09 UTC) and training set starting from 1247763673 GPS time (Jul 21, 2019 17:00:55 UTC) for LIGO Livingston. In the top plot, the blue line is the time evolution of the BLRMS for the test set while the orange line is the predicted evolution using 23 channels selected by LASSO. In the bottom plot the difference between the real signal and the predicted signal is shown.

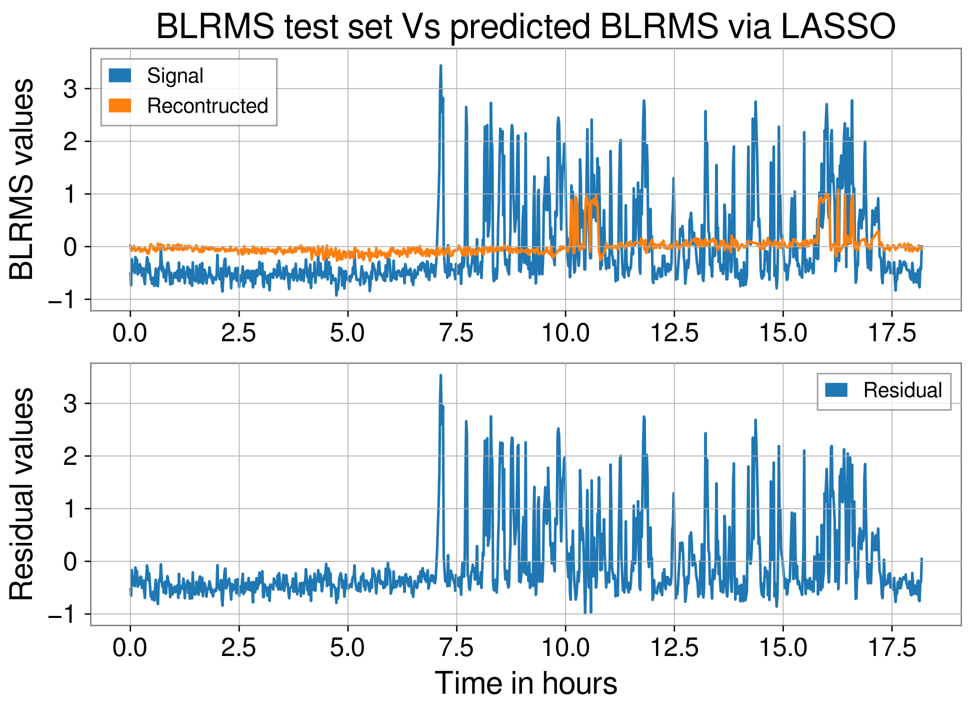

Figure 19: Plot of the predicted values and of the residuals for the test set starting from 1244608786 GPS time (Jun 15, 2019 04:39:25 UTC) and training set starting from 1244522008 GPS time (Jun 14, 2019 04:33:10 UTC) for LIGO Livingston. In the top plot, the blue line is the time evolution of the BLRMS for the test set while the orange line is the predicted evolution using standard linear regression with only the two channels linked to HAM6. In the bottom plot the difference between the real signal and the predicted signal is shown.

7.2 LIGO Hanford: 1120-1400 Hz

For LIGO Hanford we report results for two long segments that we splitted in two equal parts for the training and the test set.
In the case of Figure 21 there is a good reconstruction using only four channels, reported in Table 7. All these channels belongs to the Pre-stabilized Laser subsystem and are linked to the Frequency Stabilization Servo-System (FSS) and to the Pre-Mode Cleaner Cavity (PMC).
In the case of Figure 21, the orange line is the combination of 13 channels selected by LASSO. The shape of some bumps is predicted but not the amplitude. The two channels with the lowest ε scores are H1:IMC-REFL_A_DEMOD_RFMON ($\varepsilon = 0.94$) and H1:PSL-FSS_PC_MON_OUTPUT ($\varepsilon = 0.95$).

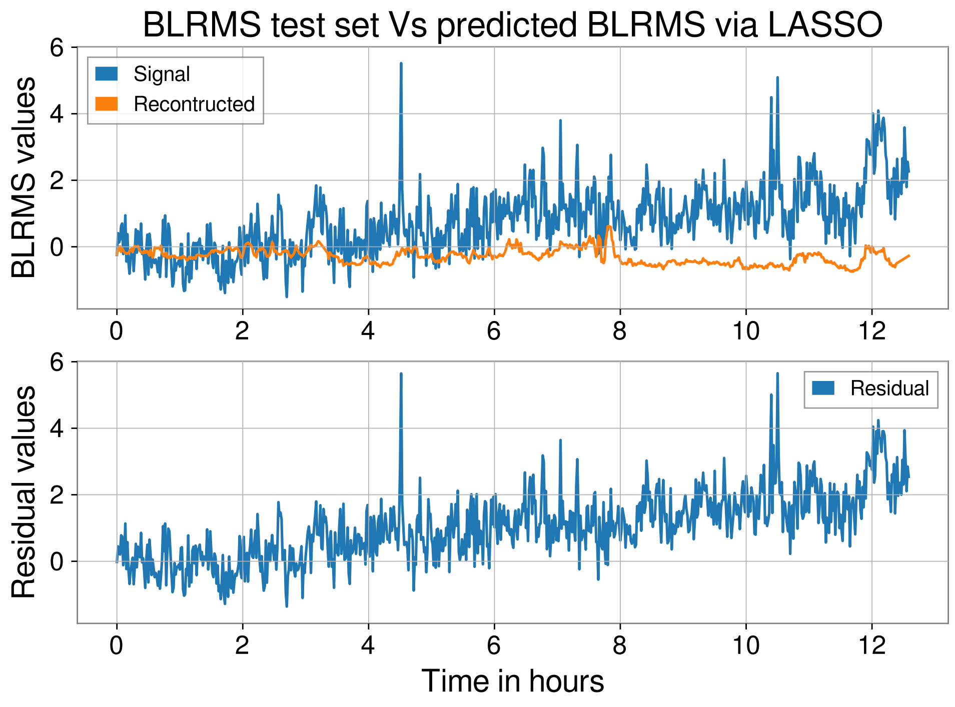

Figure 20: Plot of the predicted values and of the residuals for the test set (BLRMS segment starting from 1245857696 GPS time or Jun 29, 2019 15:34:38 UTC for LIGO Hanford) using LASSO. In the top plot, the blue line is the time evolution of the BLRMS for the test set, while the orange line is the predicted evolution using 13 channels selected by LASSO. In the bottom plot the difference between the real signal and the predicted signal is shown.

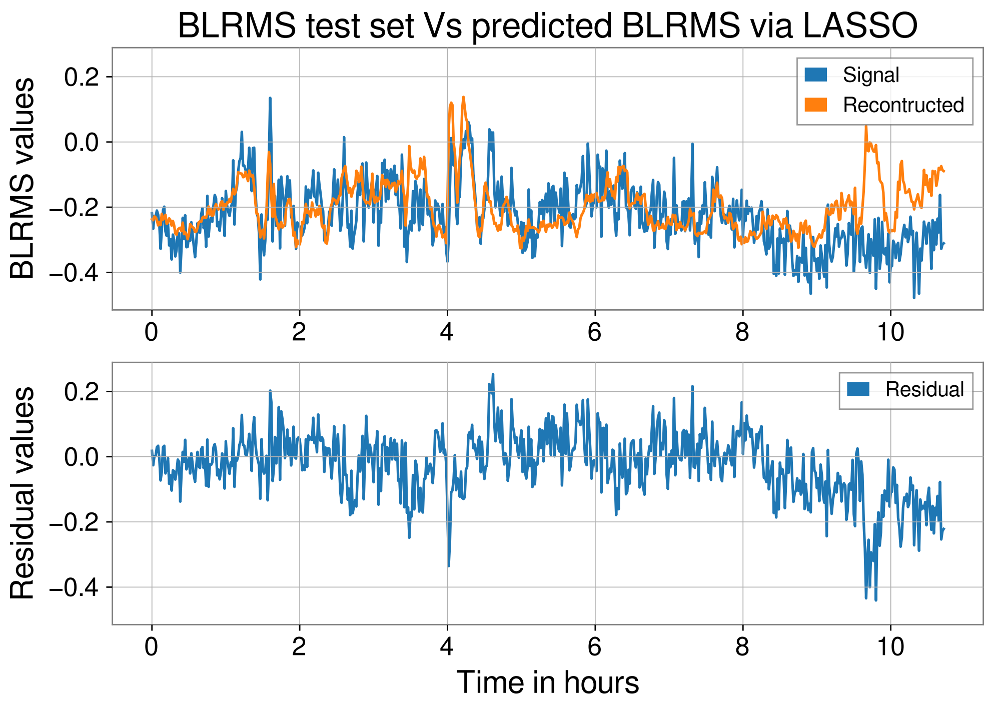

Figure 21: Plot of the predicted values and of the residuals for the test set (BLRMS segment starting from 1249000652 GPS time or Aug 05, 2019 00:37:14 UTC for LIGO Hanford) using LASSO. In the top plot, the blue line is the time evolution of the BLRMS for the test set while the orange line is the predicted evolution using 4 channels selected by LASSO. In the bottom plot the difference between the real signal and the predicted signal is shown.

Table 7: ε scores for the segment in Figure 21.

Channel name	ε score
H1:PSL-FSS_PC_MON_OUTPUT.mean	0.423
H1:PSL-FSS_PC_MON_OUTPUT.std	0.581
H1:PSL-FSS_PC_PP.std	0.776
H1:PSL-PMC_OSCILLATION_MON.std	0.863

8 Conclusion

Using BLRMS time series, we can select particular frequency bands to reconstruct the time evolution of the noise in LIGO detectors. We removed lines in the PSD and glitches in the BLRMS series in order to analyze the variation in the noise floor.
We used LASSO to solve overfitting problems of the standard linear regression. Using appropriate values of the hyper-parameter α we can reduce the number of channels to tens.
To quantify the importance of each channel we defined the ε score that is the normalized residual computed for single channel.

We showed interesting results for LIGO Livingston in the frequency band $65-76$ Hz using four different segments in the last four months of O3 data. There are good evidences that part of the variation in the BLRMS time series is linked to the two channels:

- L1:PEM-CS_ACC_HAM6_OMC_X_MON;
- L1:HPI-HAM6_BLND_L4C_VP_INMON .

These channels belong to different subsystems but both monitor the ground movement in the HAM6 vacuum chamber.

For LIGO Hanford we found interesting results for the frequency band $1120-1400$ Hz especially for a recent segment (4 Aug, 2019) where LASSO selects only four channels, belonging to the PSL subsystem. Further analysis is required.

A Random Forest

In Section 5, we showed the overfitting problems of standard linear regression.
In order to reduce these problems and to explore non parametric regressors[4], we chose Random Forest (RF). A RF is an ensemble of multiple Decision Trees trained via bagging method. RF fits a number of decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control overfitting. The sub-sample size is always the same as the original input sample size. As regressor, RF can be understood as the sum of piecewise linear functions in contrast to the global linear regression models that we discussed previously. RF usually has a better generalization performance than an individual decision tree. In facts, instead of searching for the very best feature when splitting a node RF searches the best feature among a random subset of features [14]. This results in a greater tree diversity which helps to decrease the model's variance. As impurity metric we used the Mean Squared Error (MSE). RF tries to minimize the impurity metric, that is the squared error function, using iterative methods and ensamble learning.
In the case of the example in Section 5, we set the number of trees and the maximum number of nodes for each tree using grid search. Grid search is a brute-force search paradigm where we specify a list of values[5] for different hyperparameters, and the computer evaluates the model performance for each combination of those to obtain the optimal combination of values from this set [9]. For the chosen set, we set the number of trees to 50 each limited by a maximum of 5 nodes.

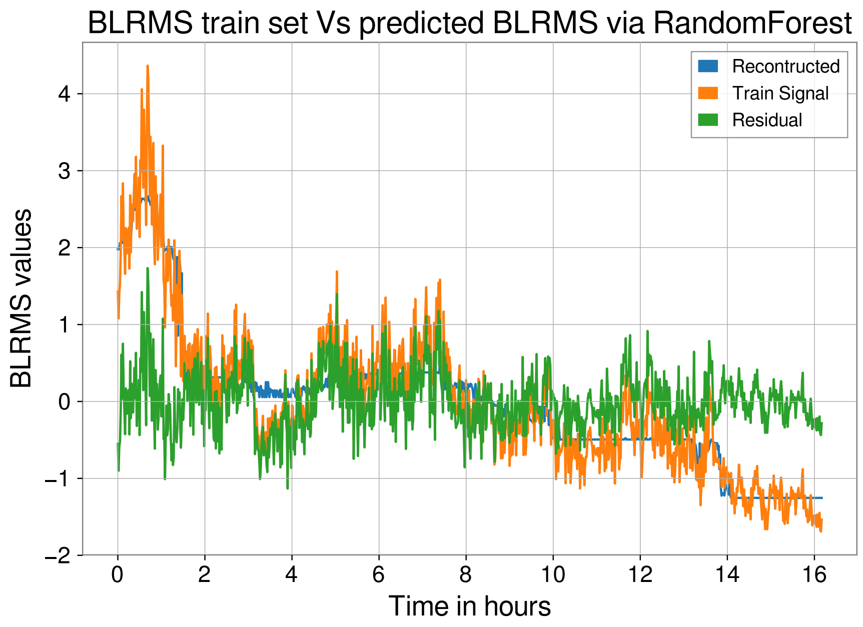

Figure 22: Plot of the predicted values and of the residuals for the train set via RandomForest.

[4]Machine learning algorithms can be grouped into parametric and non-parametric models. Using parametric models, we estimate parameters from the training dataset to learn a function that can classify new data points without requiring the original training dataset anymore. In contrast, non-parametric models can't be characterized by a fixed set of parameters, and the number of parameters grows with the training data [9].

[5]In this work the list for the number of trees is [30,50,70,100,120,140,160,180,200,250,300,500] while for the number of nodes for each tree is [5,7,9,12,15].

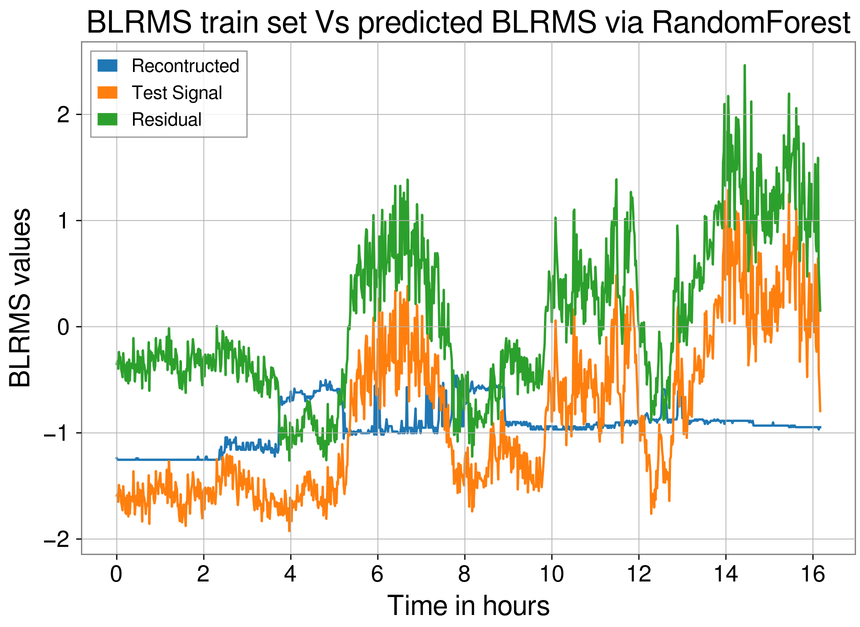

Figure 23: Plot of the predicted values and of the residuals for the training set via RandomForest.

The MSE value for the training set is $MSE_{train} = 0.15$ while for the test set is $MSE_{test} = 0.56$, while with linear regression MSE_{train} is practically zero and $MSE_{test} = 8.08$. Looking at the residual (see Figure 23), it is clear that the RF does not even get close to a good prediction. The reason the MSE is lower in the RF is that the prediction is just smaller in amplitude, but not any closer to the real values.
Using simulated data 6, it is possible to show that LASSO performance overcomes RF in terms of MSE_{test} values. Moreover, LASSO reduces the number of channels linked to the evolution of the noise.

References

[1] G. Vajente, ”Analysis of sensitivity and noise sources for the Virgo gravitational wave interferometer”, Tesi di perfezionamento, Scuola Normale Superiore di Pisa, https://gwic.ligo.org/thesisprize/2008/VajenteThesis.pdf (2008)

[2] G. Vajente, ”Band-limited RMS”, aLIGO LHO Logbook - posted 14:55, Monday 22 April 2019 (48668), https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=48668,

[3] G. Vajente, ”Compute band-limited RMS for LIGO O3 data”, https://git.ligo.org/gabriele-vajente/blrms (2019)

[4] G.Vajente, ”Study of dark-fringe noises slow non-stationarities during VSR1”, Virgo Internal note, VIR-009A-08 (2008)

[5] S. Pollock’s, ”The Classical Linear Regression Model”, Lectures at the University of Leicester, http://www.le.ac.uk/users/dsgp1/COURSES/MESOMET/ECMETXT/06mesmet.pdf

[6] D. Barker, ”aLIGO CDS Channel Naming Standards”, LIGO Scientific Collaboration, LIGO-T1000649-v1 https://dcc.ligo.org/public/0029/T990033/000/T990033-00.pdf (2011)

[7] aLIGO LHO Logbook, ”48Hz Peak in DARM”, https://alog.ligo-wa.caltech.edu/aLOG/index.php?callRep=50204, Wednesday 26 June 2019 (50204)

[8] NumPy v1.16 Manual, https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.pinv.html

[9] S. Raschka, V. Mirjialili, ”Python Machine Learning”, Expert Insight (Second Edition 2017)

[10] scikit-learn v0.21.3 Manual, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[11] M. Walker et al., ”Identifying correlations between LIGO’s astronomical range and auxiliary sensors using lasso regression”, Classical and Quantum Gravity 35(22),0264-9381 (2018)

[12] P. Zhao, B. Yu, ”On Model Selection Consistency of Lasso”, Journal of Machine Learning Research 7 (2006)

[13] The LIGO and Virgo collaboration, ”Sensitivity Achieved by the LIGO and Virgo Gravitational Wave Detectors during LIGOs Sixth and Virgos Second and Third Science Runs”, arXiv:1203.2674 (2012)

[14] A. Gron, ”Hands-On Machine Learning with Scikit-learn and TensorFlow”, O’Reilly, First Edition (2017)