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

Technical Note **LIGO-T2300147–v** 2023/10/26

LIGO Seismic State Characterization using Machine Learning Techniques

Isaac Kelly

California Institute of Technology
LIGO Project, MS 18-34
Pasadena, CA 91125
Phone (626) 395-2129
Fax (626) 304-9834
E-mail: info@ligo.caltech.edu

Massachusetts Institute of Technology
LIGO Project, Room NW22-295
Cambridge, MA 02139
Phone (617) 253-4824
Fax (617) 253-7014
E-mail: info@ligo.mit.edu

LIGO Hanford Observatory
Route 10, Mile Marker 2
Richland, WA 99352
Phone (509) 372-8106
Fax (509) 372-8137
E-mail: info@ligo.caltech.edu

LIGO Livingston Observatory
19100 LIGO Lane
Livingston, LA 70754
Phone (225) 686-3100
Fax (225) 686-7189
E-mail: info@ligo.caltech.edu

http://www.ligo.caltech.edu/

Abstract

The sensitivity of the LIGO detector makes it susceptible to interference from many sources. Among these sources is seismic noise. As of Observing Run 3, LIGO detectors are not limited by seismic noise; however, glitches (temporary aberrations in the strain signal) and locklosses (complete loss of interferometer function) occur on a regular basis, and some correlation with seismic events is suspected. In order to test this hypothesis more carefully, these seismic events must be identified. Because of the lack of ground truth and the statistical difficulty of identifying both local and global trends on short and long timescales, a machine learning approach to this problem is indicated. Specifically, a clustering algorithm has been applied to seismometer data in order to separate time segments according to the type of seismic activity. The resulting classification of seismic activity has allowed correlations to be discovered between certain seismic states and periods of increased detector glitch rates.

1 Introduction

The Laser Interferometer Gravitational-wave Observatory (LIGO) is the world's largest interferometer, constructed specifically to detect gravitational waves. These distortions in spacetime appear as changes in the relative lengths of the interferometer arms, which causes a phase shift in the light reflected by the test masses. This phase shift constitutes the magnitude of the detector signal, indicating the current strain on spacetime. Analysis of the strain over time allows extraction of signals from individual events, and these events provide insight on astrophysical and relativistic phenomenae. Current technological limitations constrain the observable emissions to those from inspiraling compact binary objects.

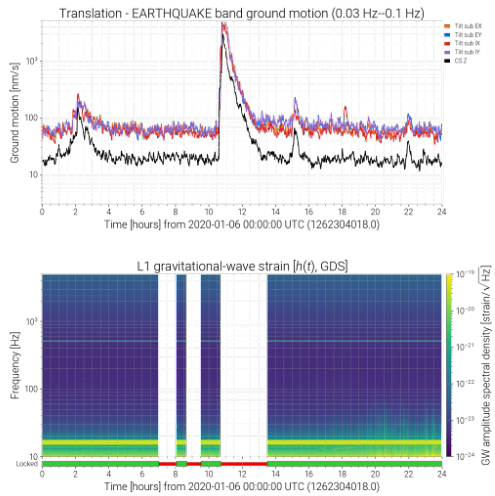

Figure 1: Earthquake event and corresponding data loss

Besides gravitational radiation, terrestrial detectors are subject to a range of other strains, all of which can interfere with the correct operation of the detectors. A significant part of this interference is ground motion; the 4-km interferometer arms are susceptible to distortion caused by movements in the earth beneath them. Even the strongest gravitational waves require a detector sensitivity of approximately $1 * 10^{-21}/\sqrt{Hz}$ to be evaluated with scientific significance.[1] One common type of ground motion, called the 'secondary microseism', is over 10 orders of magnitude stronger than the real signal at 10 Hz. [1] Earthquakes and human-caused (or anthropogenic) noise also cause distortions or loss of the strain signal. Ground motion is a significant contributor to noise and glitches in the detector (see figure 1) with both active and passive isolation utilized in the system to reduce its effects. [1] However, this isolation cannot completely nullify its effects, and so it is necessary to determine when the detector is being affected by this interference in order to find noise sources, test new isolation methods, and avoid misinterpretation of such noise as a gravitational wave event.

In order to prevent ground motion from propagating to the interferometer system, LIGO includes multiple seismic isolation devices. One of these is the Internal Seismic Isolation (ISI) system built into each optical chamber. One part of the ISI is a seismometer mounted on the ground outside the vacuum chamber in order to provide feed-forward correction. [2] These seismometers (one for each vacuum chamber) also allow the measurement of the respective ground motion at each chamber. Data from these seismometers is recorded and stored as time-series, and is available through the LIGO Network Data Service.

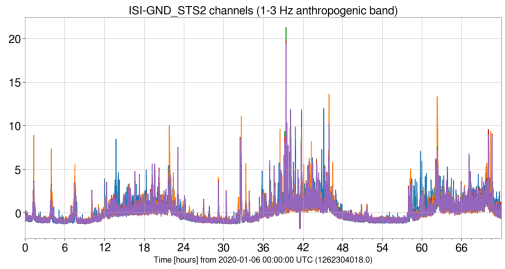

Figure 2: Periodic elevation in noise floor in the anthropogenic band during daytime hours, followed by reduction during nighttime hours.

We use traditional clustering methods to analyze this data in order to evaluate the seismic state of the detector. Time-series data from multiple seismometers is acquired and divided into segments of equal lengths. A feature extraction process is applied to these segments. Finally, the K-means clustering algorithm is applied to the resulting features, and the clusters compared to known seismic states and changes in glitch rates. The K-means algorithm has already shown promise in identifying seismic states. Bernhardt et al [3] applied K-means to seismometer data, and then to microphones and accelerometers in the physical environment monitoring (PEM) system. Bernhardt's approach used raw timeseries data, taking a 2-hour segment from each point in time, while we are applying feature extraction and separating time segments altogether. Our pipeline successfully identifies known seismic states; microseism trends, daily and weekly anthropogenic noise elevations, and short-term earthquake events. Having identified these states, we show a strong correlation between periods of elevated seismic noise and increased detector glitch rate, and demonstrate the suitability of the clustering process for analysis of detector behavior.

2 Methods

2.1 Summary

A complete seismic analysis pipeline has been constructed and evaluated, enabling the entire process from data acquisition to final presentation of results. The fundamental structure is illustrated in Figure 3. Ground motion timeseries data from ISI seismometers is split into segments and run through a feature extraction process; the resulting dataset is clustered using the K-means algorithm. The clusters are evaluated for stability, mathematical goodness, and correlation with periods of increased glitch rate and known seismic states. Important parameters in this process include the length of the time segments, the configuration of the feature extraction process, and the initial conditions provided to the k-means algorithm; this parameter space has been extensively explored.

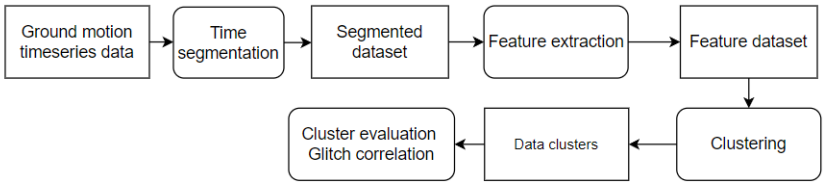

Figure 3: Pipeline structure. Raw ground motion data is processed by segmentation and feature extraction; K-means clustering of the resulting dataset groups time segments by Euclidean distance. The clusters are then evaluated for intrinsic goodness, their correlation with LIGO events of interest, and coherence with known seismic states.

2.2 Data acquisition

The data under consideration comes from the ISI seismic isolation system; specifically, a set of five seismometers at different isolation modules at the LIGO Livingston detector. These devices are mounted directly to the floor without isolation, so they accurately measure the motion and vibration of the ground beneath the detector. Timeseries data from these sensors is recorded and stored in Caltech's Network Data System, where it is available to any LIGO member through various software packages. This seismic data is available in a filtered form, having been separated into separate bands with a band-pass filter in order to ease visual identification of seismic activity. The bands span the effective range of the seismometer, 0.01Hz up to 30Hz, allowing the identification of a wide range of seismic phenomenae.

2.3 Feature extraction

A timeseries dataset is inherently a highly-dimensional object; each timestep defines another dimension. As our analysis can span months, we must apply multiple techniques to flatten our dataset for ease of analysis. First, data is segmented by time; as we progress through the process of analysis, the time periods denoted by these segments remain constant, allowing comparison to other simultaneous events in the detector. Each of these segments contains thirty parallel timeseries: six separate bands for five different seismometers. Adjusting the length of these segments has the potential to drastically alter the final results of the process, so a wide range of lengths (between 30 seconds and 4 hours) has been explored. Once data is segmented, the feature extraction process can begin. Feature extraction is a common practice when dealing with timeseries data, as it can effectively reduce the dimensionality and complexity of a dataset. It involves a statistical analysis of a timeseries, reducing what may be a very long series into a simplified set of measured parameters known as features. A wide range of features is available in the **tsfresh** library, with over 700 possible feature-parameter combinations. However, many of these features are either redundant or actively harmful, especially when considering shorter timeseries. We have explored two extremes of feature count: first, a naive approach where all 700 datapoints are computed and passed to the clustering step, and then a much leaner attack where only eight parameters are calculated. The simplified extraction process is much faster (taking 10 seconds versus the 7 minutes required for comprehensive extraction) and additionally produces a more intelligible dataset. For an initial glimpse into the character of our data, we have applied a t-distributed stochastic

neighbor embedding (t-SNE) data reduction technique, which allows the visualization of an n-dimensional dataset in two or three dimensions. Figure 4 shows the difference between feature sets as computed by the t-SNE algorithm.

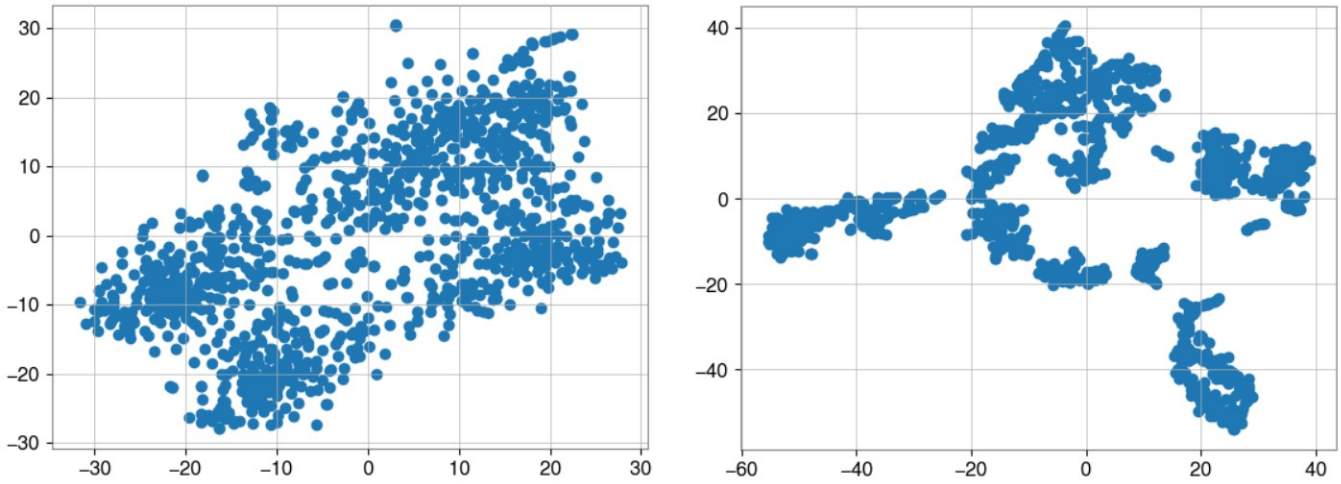

Figure 4: t-SNE representation of 72 hours of data, processed by tsfresh configured for (left) comprehensive extraction or (right) minimal extraction. t-SNE creates qualitative and unitless results which measure the similarity between datapoints. Note the difference in distributions, with the minimal extraction producing much tighter and more separate groups.

2.4 Clustering

After completing the extraction process, the dataset is ready for the application of the clustering algorithm. We have focused on the k-means [4] algorithm, as implemented in the **scikit** library. [5] This process separates the time segments (as represented by features) into k discrete clusters, based on their squared Euclidean distance from certain centroids; these centroids are initially selected randomly and are then refined as the algorithm proceeds. The k-means algorithm has two important parameters: first, the value of k, which determines the desired number of clusters; second, the initial centroid positions. These parameters affect the stability of the results (how clusterings change run to run), the quality of the clusters (how well the algorithm is able to select well separated groups), and the computations required to bring the algorithm to convergence. Because we are working with datasets in the megabyte range, performance is not currently a concern. However, stability and quality of results is crucial. To this end, we have explored a variety of mathematical evaluations of our clustering results.

2.5 Cluster analysis

Having completed the structure of the pipeline, parameter adjustments are necessary. We have discussed the factors affecting the configuration of the feature extraction set: the effects of dimensionality on the clustering step, as well as the performance issues in the feature extraction process. The configuration of the clustering algorithm is equally important. In

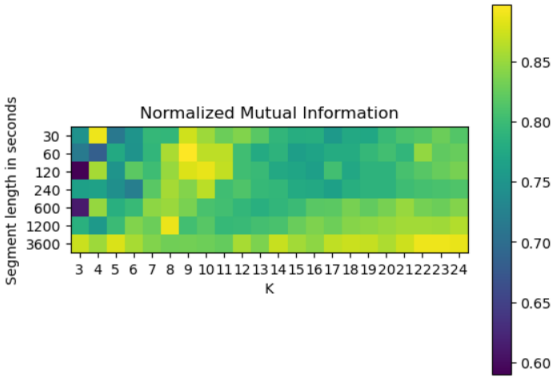

Figure 5: Normalized Mutual Information (NMI) score across 25 runs of single initialization kmeans++. We see a mean of approximately 0.8; this means that around 80% of clusters are identical between runs.

order to make sensible adjustments, measurements of cluster performance are necessary. First, we want to ensure that our results are repeatable. While the k-means process is mathematically deterministic, real implementations are not. Secondly, it is important to evaluate how well the k-means algorithm is able to create separated clusters.

One of the original goals of this project was the identification of new seismic states – the discovery of new types of seismic activity that had certain effects on the detectors. In order for these to be successfully identified by a clustering approach, the number of clusters (denoted by the parameter 'k') must be carefully adjusted and the results examined in detail. We have not yet been able to identify such new states; however, applying these methods to larger timescales and the corresponding diverse events should allow progress in this direction.

2.5.1 Stability

Depending on the clustering parameters, and especially the initial cluster centroids, running the same process on the same data multiple times can yield drastically different results. To address this problem, the parameters of the k-means++ initialization were adjusted to take multiple trials of centroid locations and choose the best. The initial randomness, when combined with sufficient sample size, creates greater consistency in the final result. In order to evaluate the similarity between various results, a few algorithms can be applied. In this work we have utilized normalized mutual information (NMI), adjusted rand index (ARI), and Fowlkes-Mallows score; any one of the three metrics is sufficient to establish the difference in stability between different clustering parameters. In Figures 5 and 6, we compare NMI scores for a varying number of kmeans++ initialization trials. Allowing kmeans++ to use more initialization attempts clearly increases the stability of the results.

Besides being important for the reliability of clustering results, increased stability can also point towards the correct range for the k parameter. Luxburg on clustering stability [6] notes that since the k-means algorithm creates sharp boundaries between clusters, a value of k that is too high will result in clusters being divided – placing a boundary in a high-density zone. Because the algorithm is not deterministic, the boundary will vary between runs. The result is a significant difference in results between runs because of that high density. Therefore,

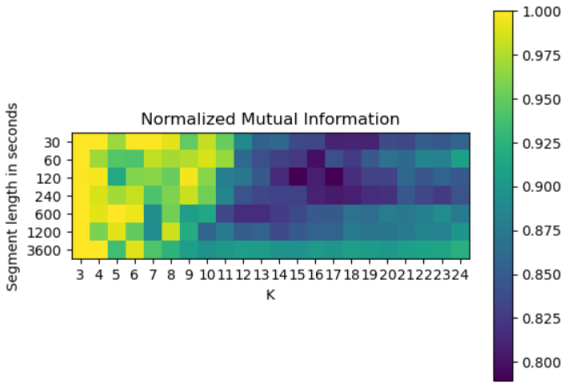

Figure 6: NMI score across 25 runs of 10-initialization kmeans++. All scores are above 0.8, with some as high as 0.98; similarity between multiple clustering runs is thus very high, and this initialization condition is more effective than that illustrated in Figure 4.

high instability generally indicates the k parameter being set too high. On the other hand, if k is too low, multiple 'true' clusters will be combined; this can cause similar instability as the algorithm struggles to make consistent divisions. Keeping this in mind, we have explored stability measurements in some detail beyond the broad improvement shown when modifying the kmeans++ initialization configuration. Figure 6 plots stability as k increases over a variety of segment lengths. We can see that as k reaches 10 to 12 stability decreases significantly across the board. Referring to Luxburg, we can see that we are experiencing instability due to k being too high. Thus we should be considering the range $k < 10$ for further exploration.

2.5.2 Intrinsic scores

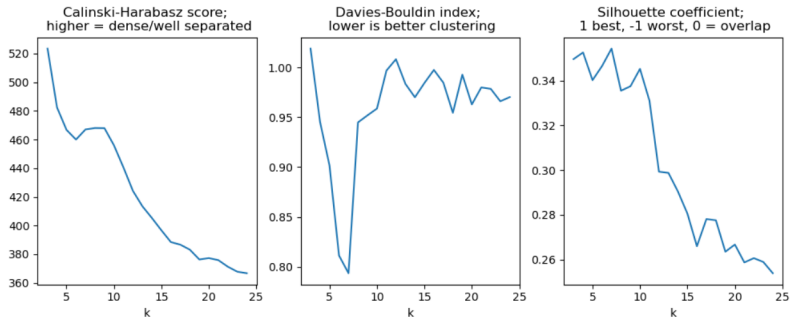

Figure 7: Cluster evaluation metrics for 72 hour dataset split into 240-second segments, evaluated over a range of k values. Each metric addresses a different statistical characteristic of the clusters; all three should be considered together for a measurement of the goodness of the clustering.

Intrinsic cluster evaluation metrics have been applied to our clustering results. The Davies-Bouldin, silhouette, and Calinski-Harabasz algorithms have been tested. Each metric attempts to characterize the quality of the clustering in terms of how well each cluster is separated from the others.

Figure 7 shows an example of these intrinsic scores, as computed from a 72 hour dataset in 240-second segments. All three scores generally agree that the range of k between 5 and 10 is promising, which accords well with our initial assumption of 4 or 5 known seismic states, plus a few outliers and possible new states. This agrees very nicely with what we demonstrated with stability metrics, which is a strong result! We have selected the range between 5 and 10 clusters as the most promising, and continue to explore results in this area.

3 Evaluation of Results

In many machine learning systems, a feedback system is created in order to allow automated refinement of the algorithm. This requires labelled data, constituting a supervised machine learning approach. Because of the complexity of seismic states, no labelled data is available.

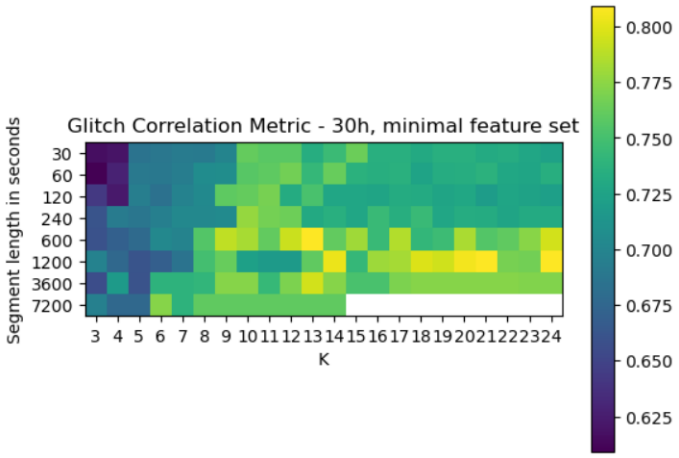

Figure 8: Heatmap of new metric – higher is better. Note the 'sweet spot' between k=6 and k=12, with slight variations based on segment length; additionally, the significant changes as k surpasses 4. These points of interest have given significant insight into clustering characteristics.

Intrinsic cluster metrics are helpful as a purely mathematical evaluation of cluster characteristics. However, cluster goodness and cluster usefulness are not intrinsically connected; well-separated clusters do not necessarily provide insight into seismic states or a glitch in LIGO. So we have designed and implemented a metric, the Glitch Correlation Metric, whereby LIGO glitches can be correlated with clustering results; this allows objective analysis of such results. The GCM is formulated as follows:

A timeseries T is composed of segments T_i of constant length l, each of which corresponds to a segment G_i of timeseries G. Each segment G_i contains a count g_i of glitches in that time period.

A clustering algorithm is applied to T with cluster count parameter K. This creates K discrete clusters C_j, each of which contains a set S_j of segments of T, corresponding to segments of G.

Glitch count G_j for each cluster is defined as $\sum g_i$ for all i in S_j.

The mean glitch rate R_j for each cluster is then $\frac{G_j}{count(S_j)*l}$

The mean glitch rate R_T for timeseries T is similarly $\frac{\sum g_i}{count(T_i)*l}$

Now each cluster C_j is placed in a bin determined by comparison between glitch rates R_j and R_T.

$\sum_{below} = \sum^{R_j > R_T} G_j$

$\sum_{above} = \sum^{R_j < R_T} G_j$

And the final score $M = \frac{\sum_{above}}{\sum_{above} + \sum_{below}}$

This metric evaluates how well the clustering has captured periods of time with higher glitch rates; these time periods are important because one goal of the project, as stated above, is to identify exactly these periods. The score asymptotically ranges from 0 to 1, with higher results indicating a more effective capturing of high glitch rate periods with clusters. We consider every resulting cluster in this calculation, avoiding selection bias in comparisons.

A broad gridsearch has been completed with this metric, illustrated in Figure 8, and it has been used to objectively evaluate clusterings before moving to individual analysis.

Along with this specific mathematical approach, simple comparisons of glitch rates in different segments are powerful tools for evaluating more specific correlations. If a state has been identified as consistently containing high levels of seismic noise, then evaluating the glitch rate in that state as compared to the baseline rate can illustrate the relationship between that specific type of noise and the detector's performance. To this end, we selected some promising results and examined the relative glitch rates, using ANOVA testing [7] to verify the strength of correlations.

Figure 9 illustrates the result of the application of our pipeline to 240 hours of seismometer data. This dataset has been processed with minimal feature extraction after being sectioned into 240-second segments, then separated into 7 discrete clusters. Six representative channels are illustrated below a measurement of glitch rate, a single bar indicating the detector state, and a final bar which indicates the cluster identifier. A simple visual analysis can show that the clustering has identified the long-term trends in the 0.1-0.3Hz band which correspond to microseismic activity; additionally, the 3-10hz activity with its daily cycle can be recognized in the cluster pattern. What is more important is the difference in glitch rates between different clusters; the 'orange' cluster, appearing during elevated daily noise periods, exhibits a glitch rate of 0.019 glitches/second, while the 'cyan' cluster averages to 0.0085 glitches/second. These glitch rates are plotted in Figure 10.

We have applied an ANOVA test to confirm statistical significance, resulting in a p-value less than $1 * 10^{-}10$. This trend has been replicated across multiple datasets with differing cluster numbers and segment lengths. Thus, we can announce a strong correlation between certain periods of elevated seismic activity and detector glitches.

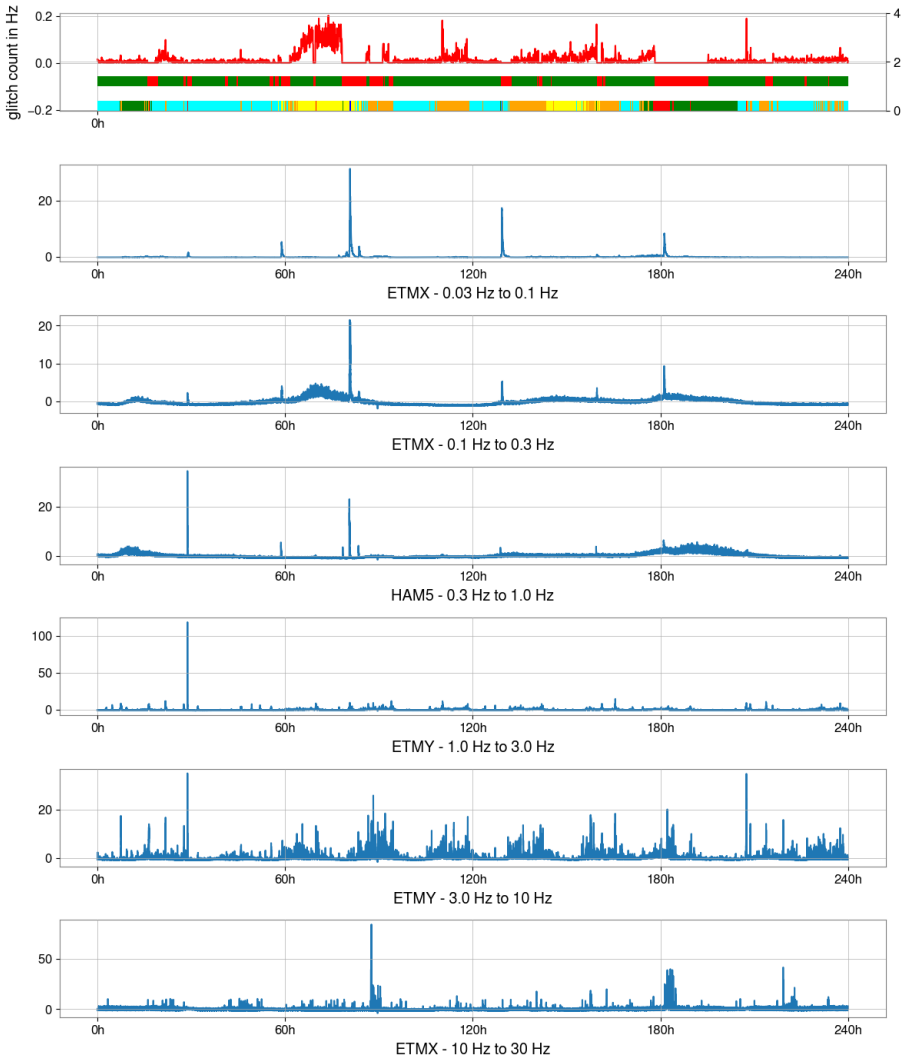

Figure 9: The result of the complete pipeline. Topmost plot represents glitch rate, detector state (green for locked), and cluster identifier. Below are six representative channels, one per band, in different locations. Note the correlation between the orange and cyan segments and the 3.0-10hz noise band.

4 Future Directions

Points of interest in LIGO strain data and the corresponding auxiliary channels include known noise events, false GW events, and other unexpected behaviors. Analysis of ground motion data from those times has provided insight into the factors that negatively affect the detector. Comparing glitch rates to seismic channels has already proven useful, as we have shown with our novel metric and specific analyses.

One extension of this project would be the use of clustering results as a predictive tool. If a unique state tends to precede locklosses, this could be a trigger for something akin to the current "earthquake mode" used by LIGO sites, which adjusts seismic isolation settings to handle increased ground motion while maintaining lock. Controllers and actuators could be adjusted to address the specific seismic motion that is linked to locklosses.

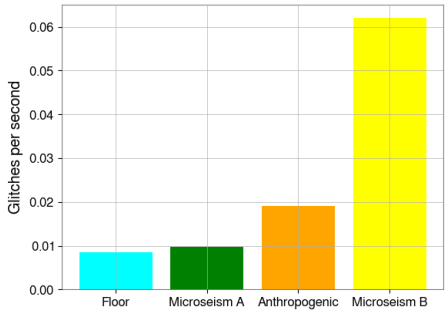

Figure 10: Results of glitch correlation – colors match Figure 9. Note the very high glitch rate in the yellow band, representing the second microseism event – this would be a good time period to examine in more detail.

The clustering method has been shown to be effective for evaluating large amounts of seismometer data. A similar approach could be applied to other environmental sensors at the LIGO sites, either for detection of anomalies (using slightly different algorithms) or for determining similar discrete states.

5 Acknowledgements

I would like to thank my mentors and supervisors at the University of California Riverside: Rutuja Gurav, Pooyan Goodarzi, Dr. Jon Richardson, and Dr. Vagelis Papalexakis. Thanks also to the Caltech SURF program, the LIGO collaboration, and the University of California Riverside.

References

[1] Rana X. Adhikari. Gravitational radiation detection with laser interferometry. *Reviews of Modern Physics*, 86(1):121–151, feb 2014.

[2] LIGO Scientific Collaboration (August 2014 LSC author list). Advanced LIGO. *Classical and Quantum Gravity*, 32(7):074001, mar 2015.

[3] Jacob Bernhardt. Data clustering techniques for the correlation of environmental noise to signals in ligo detectors. 2019.

[4] J MacQueen. Classification and analysis of multivariate observations. In *5th Berkeley Symp. Math. Statist. Probability*, pages 281–297. University of California Los Angeles LA USA, 1967.

[5] scikit-learn. https://scikit-learn.org/stable/. Accessed: 2023-05-17.

[6] Ulrike von Luxburg. Clustering stability: An overview. *Foundations and Trends® in Machine Learning*, 2(3):235–274, 2010.

[7] Lars Stehle and Svante Wold. Analysis of variance (anova). *Chemometrics and Intelligent Laboratory Systems*, 6(4):259–272, 1989.