LIGO

Understanding Interferometer Lock Losses with Machine Learning

Laurel White, Syracuse University
Mentor: Jamie Rollins, California Institute of Technology

LIGO

What is a lock loss?

- Detectors must be in a stable state known as “lock” in order to collect data
- In lock, there is:
  - Resonance in each of the cavities
  - Destructive interference at the output
- The automated lock acquisition system, Guardian, is used to obtain lock, but it is a time-consuming process
- A detector can be knocked out of lock, reducing its duty cycle
- Some lock losses are caused by earthquakes, while others have unknown causes

LIGO

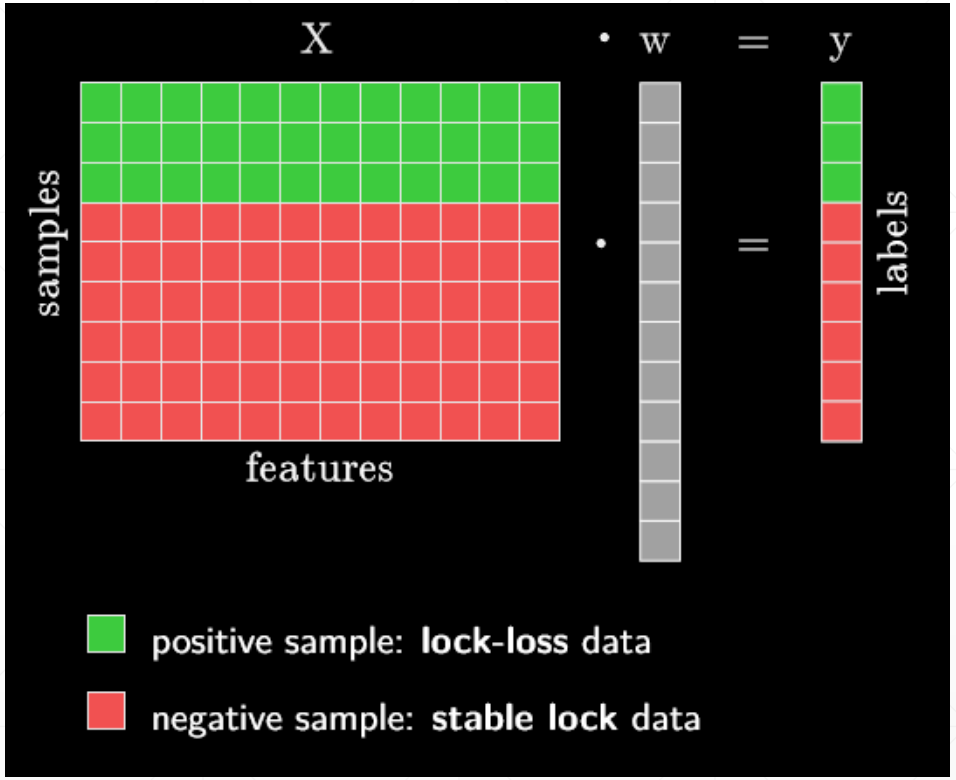
Our approach

- Besides the differential arm (DARM) channel which records gravitational-wave signals, there are thousands of auxiliary channels recording information about the detectors and their environments
  - Ex. SUS, PEM
- Our goal is to identify features in the auxiliary channel data that are characteristic of impending lock loss
  - Ex. Large spikes, unstable oscillations
- We want to use a regression algorithm to identify the most interesting features and channels out of thousands that we test and then a clustering algorithm to group lock losses with common features

LIGO

Our approach

- Compile extracted features into matrix **X**
- Use positive/negative labels to generate output vector **y**
- Solve for **w**, which maps **X** onto **y**



Courtesy of Jamie Rollins

LIGO

Overview

- Generating the dataset
- Methods and results
  - Feature extraction
  - Regression
  - Clustering
  - Follow-up of interesting results
- Conclusions and future work

LIGO

Generating the dataset

- Use O2 data from Hanford
- Channels: start with DQ channels, remove calibration channels, flat channels, and channels created during O2
- Positive samples: just prior to lock losses that occurred during observing mode
- Negative samples: stable, locked times spaced by 10,000 seconds and 1,000 seconds removed from the start and end of lock
- Sample length is a tunable parameter (we use 4 seconds)
- Left with 4,338 channels, 276 positive samples, and 1,086 negative samples

LIGO

Approach

LIGO

Approach

LIGO

Step 1: Feature Extraction

- Cannot use raw time series data
  - Channels are sampled at rates of up to 16,384 Hz
  - Too much data would hide interesting features in the regression
- We select a set of calculations that return single-value outputs for each sample
  - Difference between maximum and minimum
  - Complexity
  - Absolute energy
  - Mean value
  - Standard deviation
  - Variance
  - Kurtosis
  - Skewness
  - Absolute sum of changes
  - Mean absolute change
  - Permutation entropy

Complexity
- Take the diff:

$$\sum_{i=0}^{n-1}(x_{i+1} - x_i)$$

- Dot with itself
- Take square root

LIGO

Approach

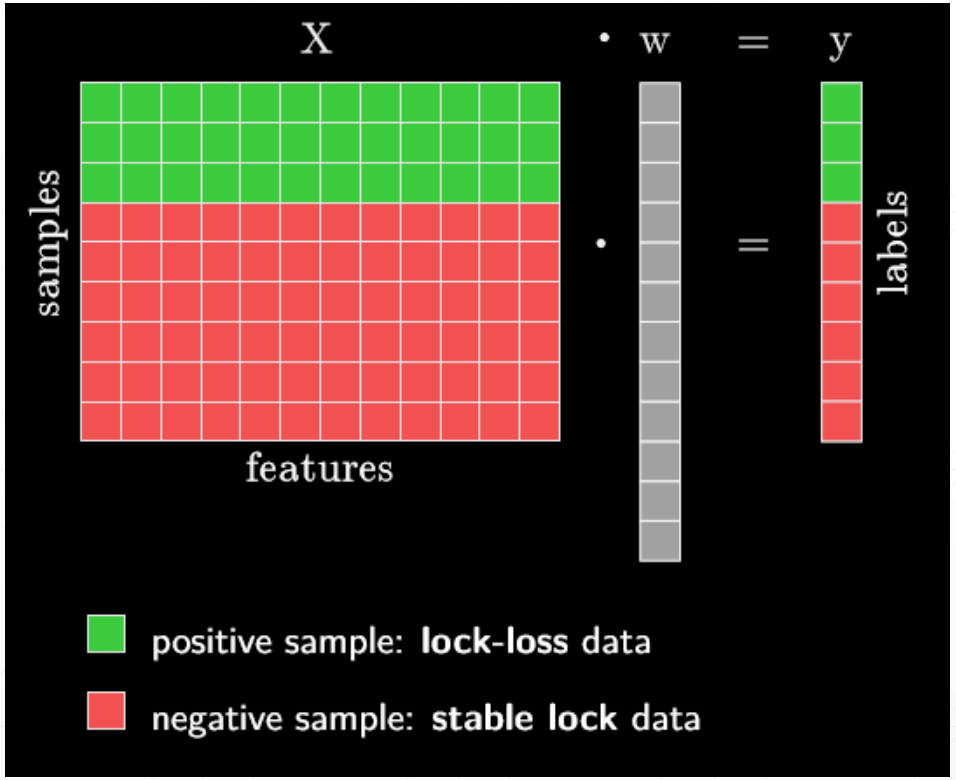
LIGO

Step 2: Regression

- Matrix is underdetermined, so we use Lasso regression
  - Alpha parameter controls degree of regularization
  - Use of L1-norm drives some coefficients to be zero so final model relies on less features

Ordinary least squares

$$\min_{\mathbf{w}} ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2$$



Courtesy of Jamie Rollins

LIGO

Step 2: Regression

- Matrix is underdetermined, so we use Lasso regression
  - Alpha parameter controls degree of regularization
  - Use of L1-norm drives some coefficients to be zero so final model relies on less features

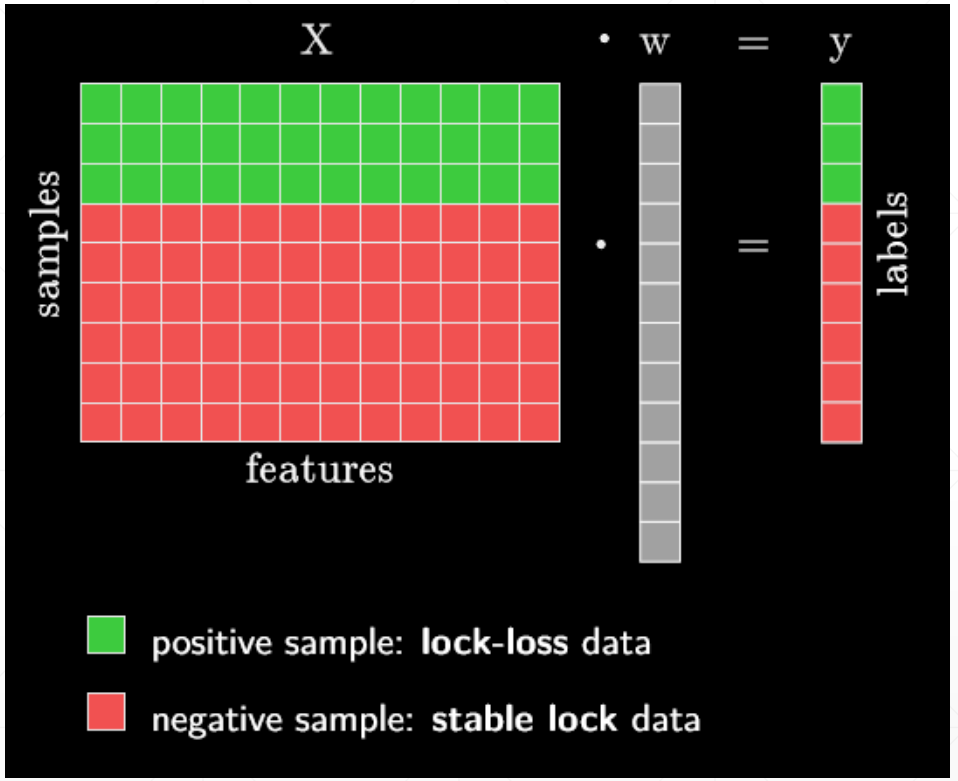
Ordinary least squares

$$\min_{\mathbf{w}} ||\mathbf{X}\mathbf{w} - \mathbf{y}||_2^2 + \alpha||\mathbf{w}||_1$$

Regularization

L1 norm (Lasso)

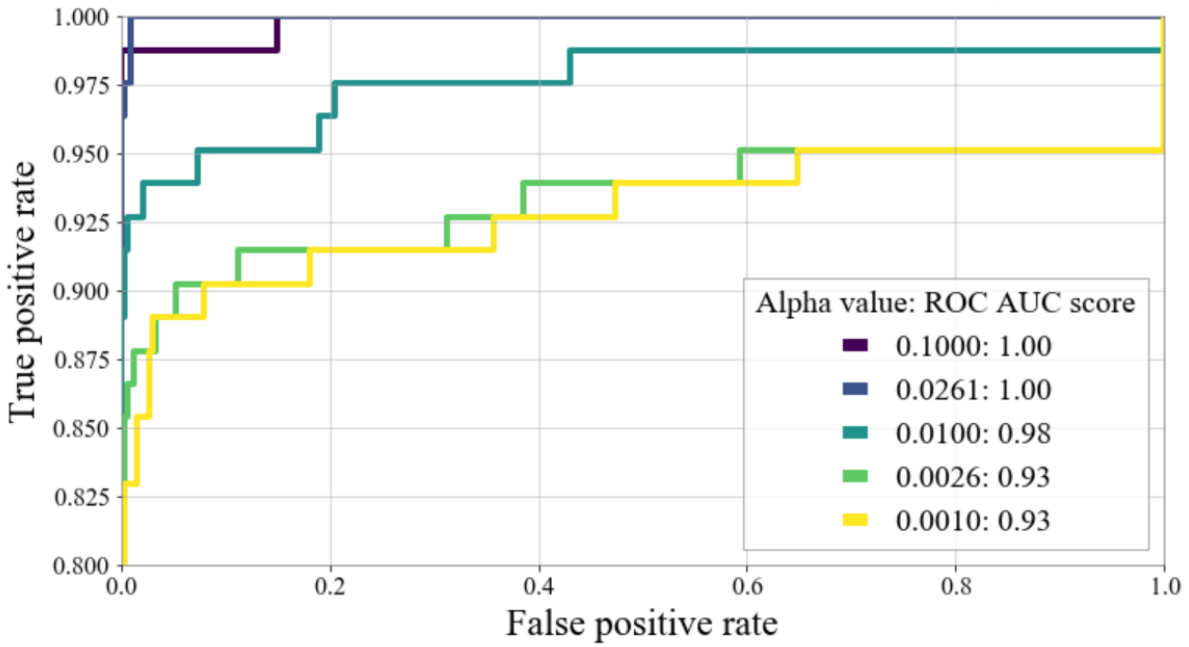
$$||\mathbf{w}||_1 = \sum_i |\mathbf{w}_i|$$

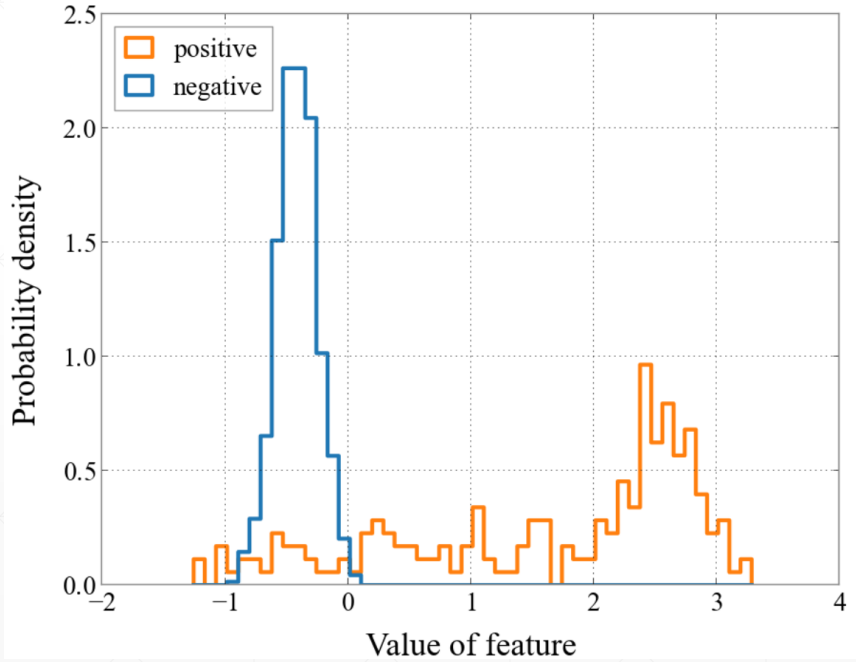


Courtesy of Jamie Rollins

Step 2: Regression

- Use receiver operator characteristic (ROC) curves with area under the curve (AUC) scores as well as histograms to evaluate success of regression





Complexity of H1:OMC-FPGA_DTONE_IN1_DQ

LIGO

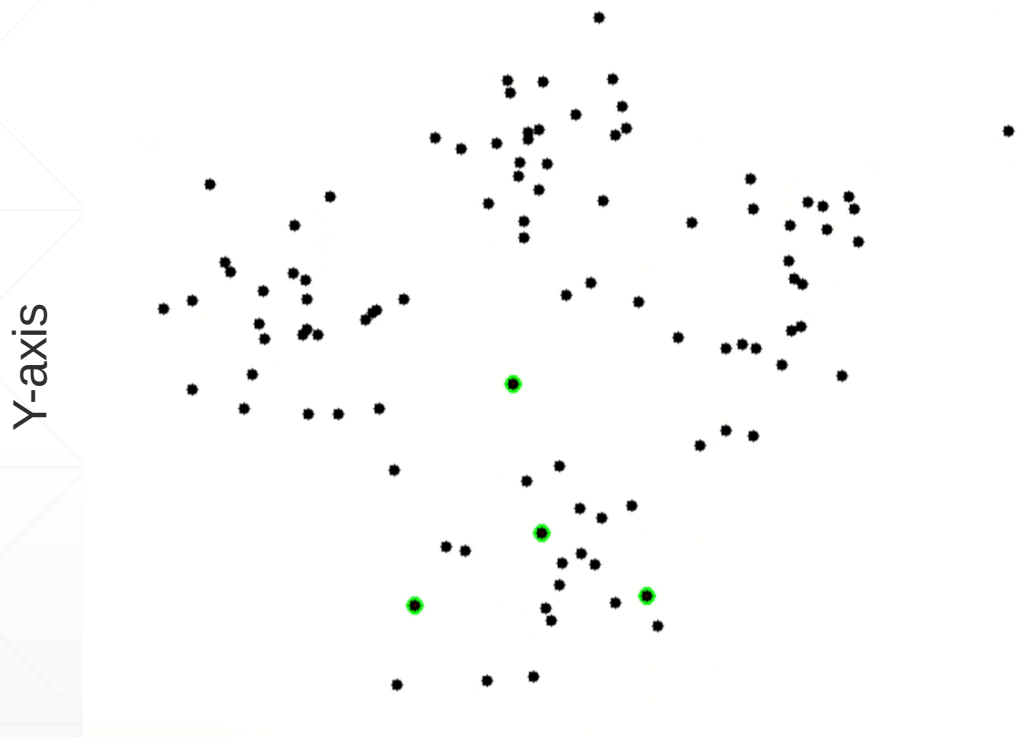
Approach

LIGO

Step 3: Clustering

- Attempt to cluster samples to find groups of lock losses that share common features
- K-means algorithm groups into any specified number of clusters
- No built-in functionality to figure out which features are most important for clustering

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (||x_i - \mu_j||^2)$$

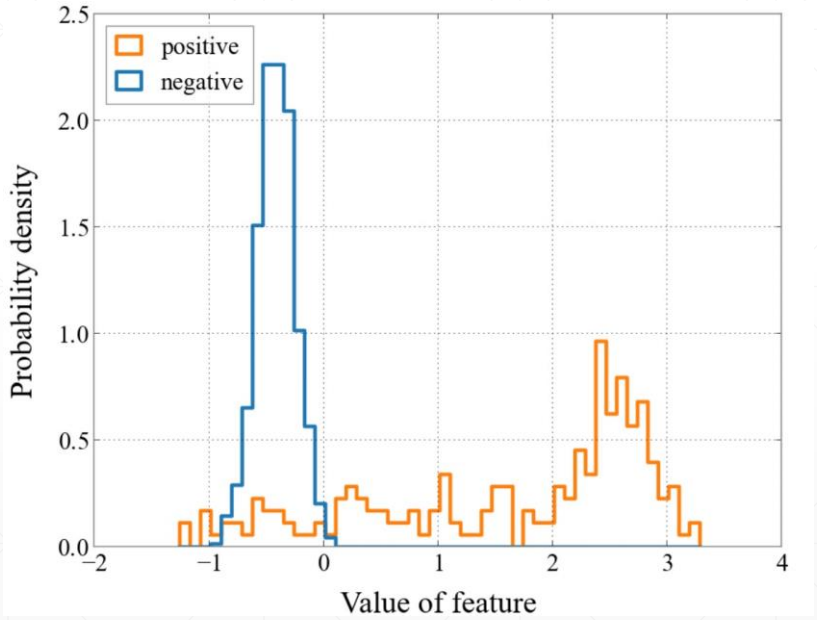


http://shabal.in/visuals/kmeans/6.html

LIGO

Step 3: Clustering

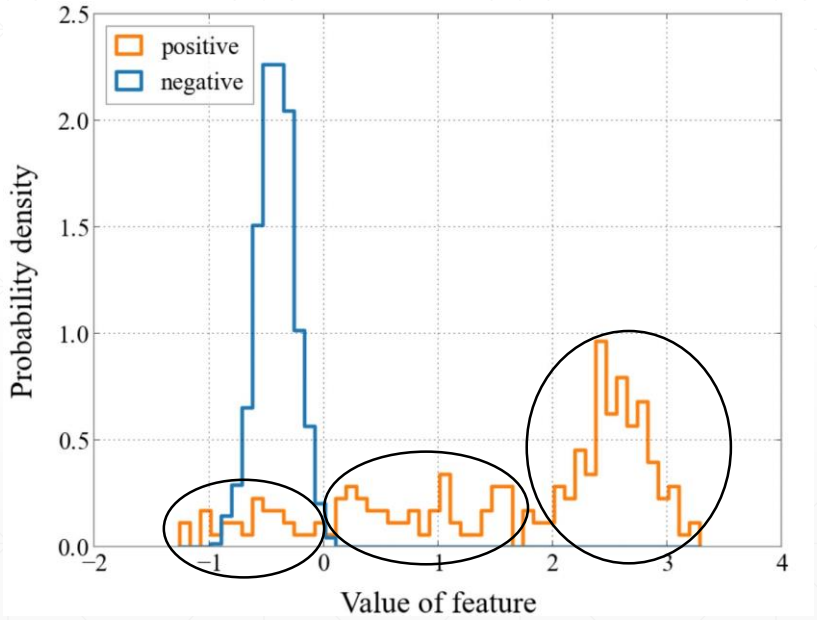
- Ideally would see clear clusters in histograms



Complexity of H1:OMC-FPGA_DTONE_IN1_DQ

Step 3: Clustering

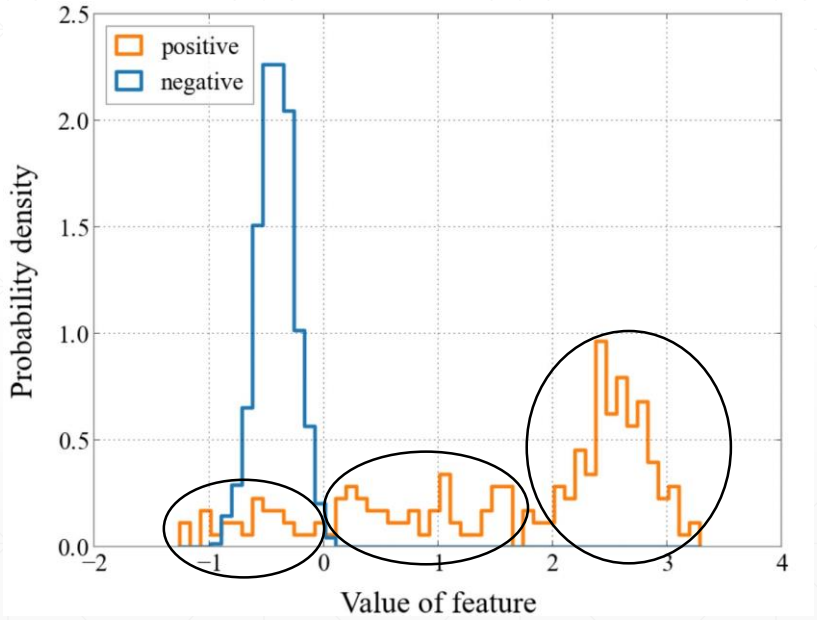
- Ideally would see clear clusters in histograms

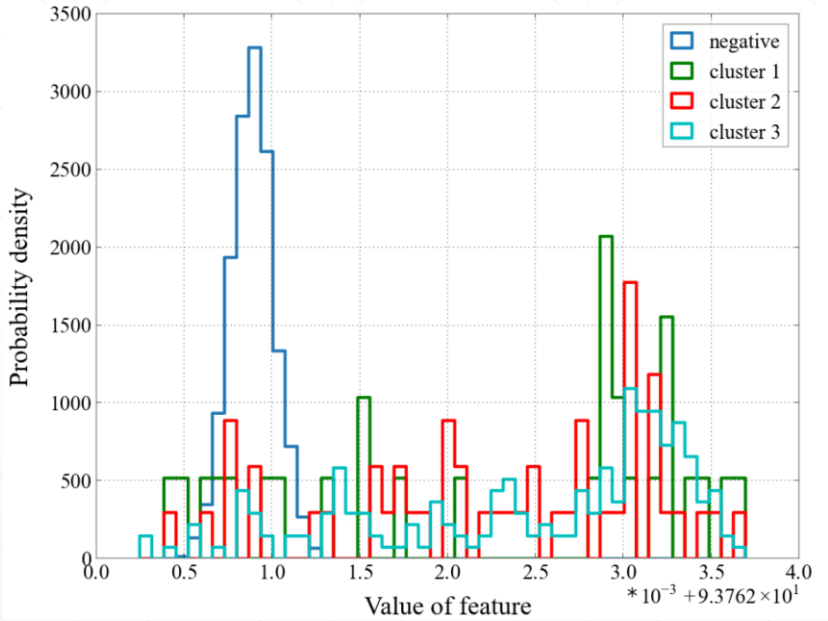


Complexity of H1:OMC-FPGA_DTONE_IN1_DQ

Step 3: Clustering

- Ideally would see clear clusters in histograms

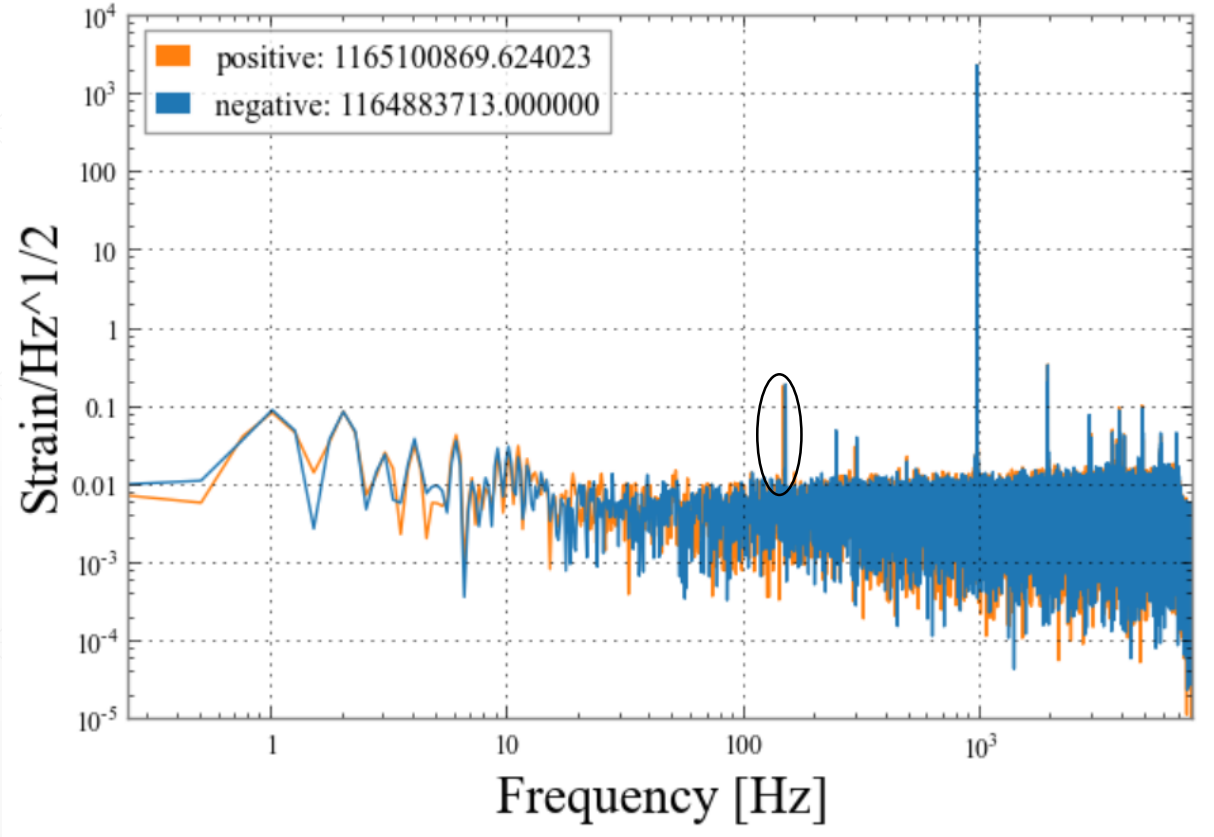




Complexity of H1:OMC-FPGA_DTONE_IN1_DQ

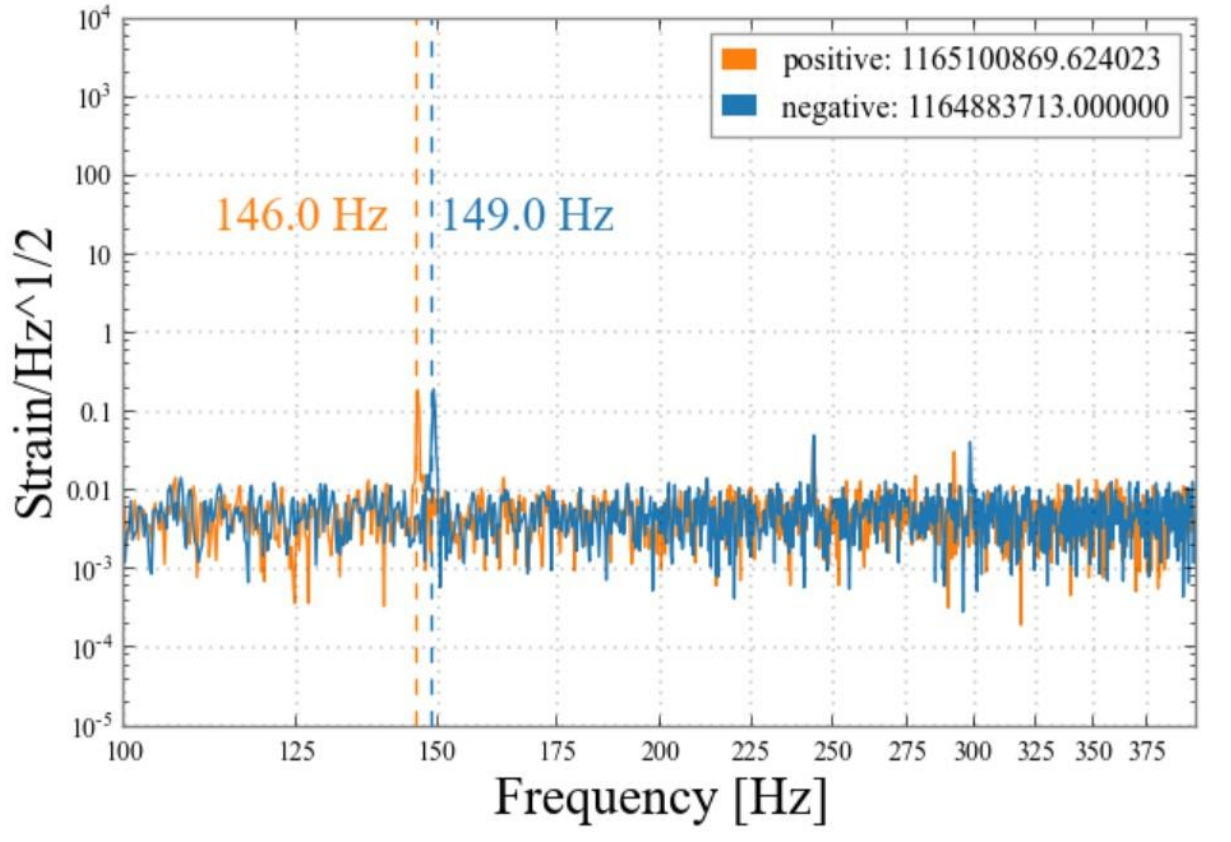
Follow-Up: H1:OMC-FPGA_DTONE_IN1_DQ

- When we plot the ASDs, we see that positive samples tend to have peaks at lower frequencies than negative samples
- This does not explain the higher complexity values for the positive samples



Follow-Up: H1:OMC-FPGA_DTONE_IN1_DQ

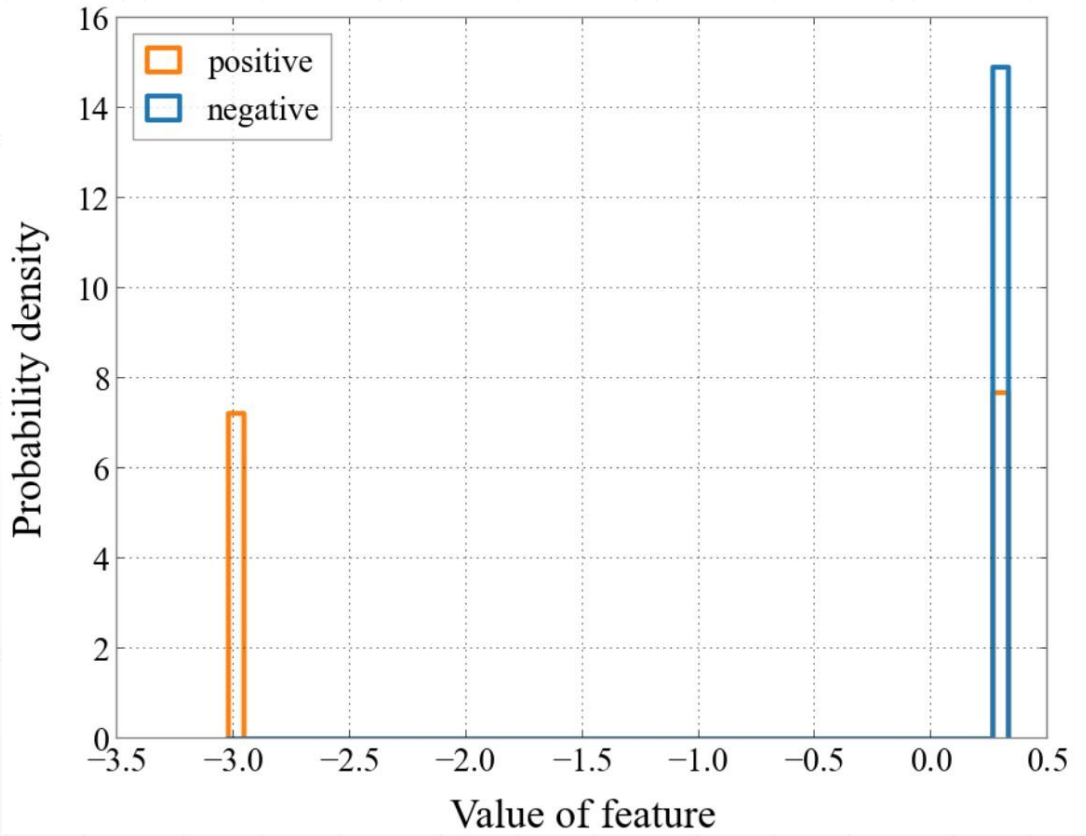
- When we plot the ASDs, we see that positive samples tend to have peaks at lower frequencies than negative samples
- This does not explain the higher complexity values for the positive samples



LIGO

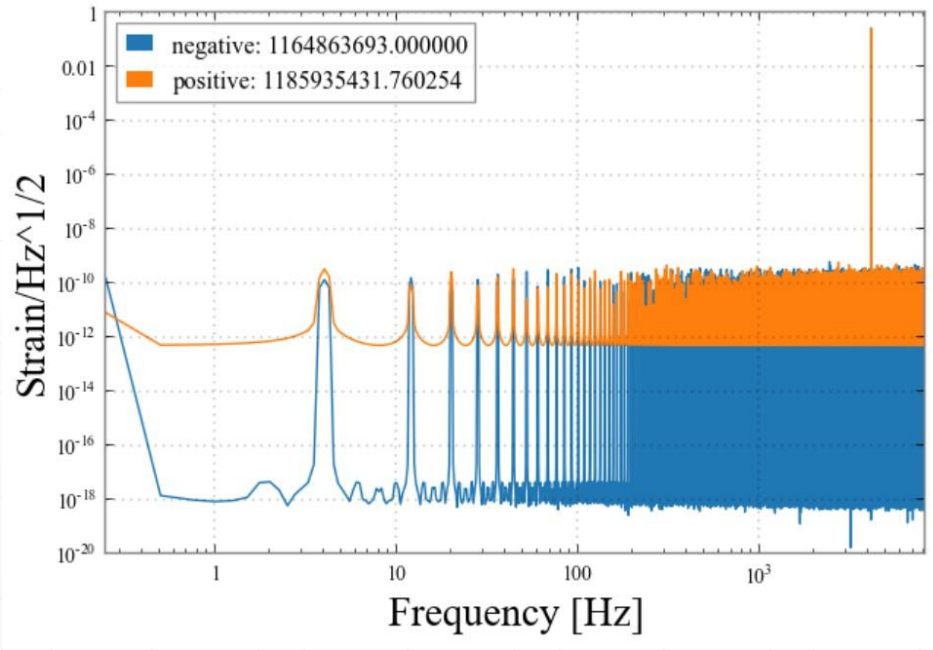
Follow-Up: H1:OMC-LSC_DITHER_OUT_DQ

- The permutation entropy always hovers around one value for the negative samples, but it is split between two values for the positive samples

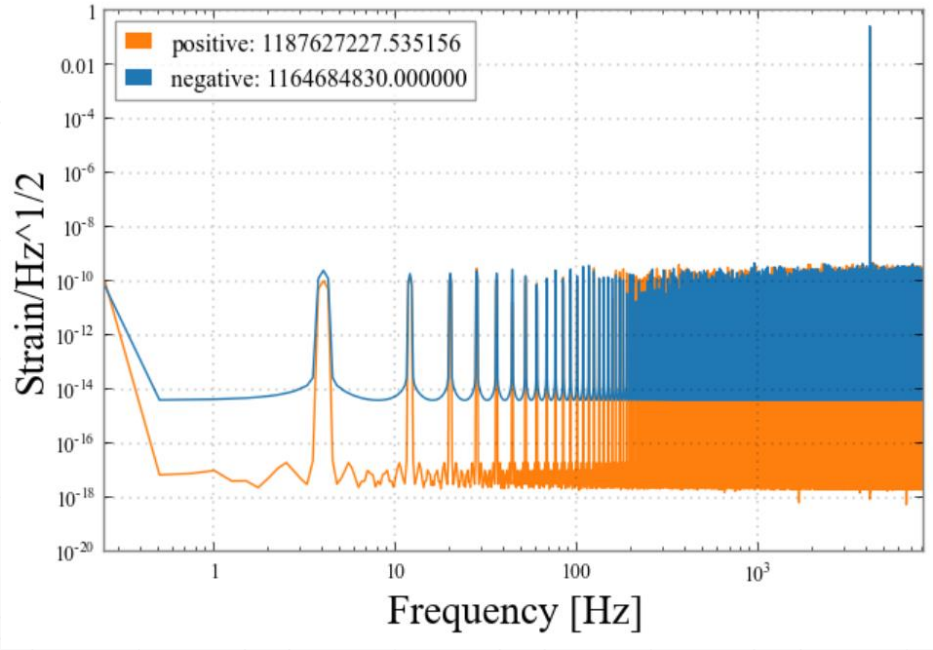


Follow-Up: H1:OMC-LSC_DITHER_OUT_DQ

- Plotting the ASDs shows no pattern among the noise levels for the positive or negative samples, regardless of the values of the permutation entropy



Both samples have high permutation entropy



The positive sample has low permutation entropy

LIGO

Conclusions

- Identified a small number of interesting features which differ between lock loss and clean times
- Did not find meaningful clusters

Future work

- Prune channel list
- Find new features that better distinguish positive from negative samples
- Refine clustering

LIGO

References

- Evans, M., et al. (2002). Lock acquisition of a gravitational-wave interferometer. *Optics Letters, 27(8).*
- Rollins, J. (2015, June 16). Advanced LIGO Guardian Documentation, Release 1447. Retrieved from https://dcc.ligo.org/public/0120/T1500292/001/AdvancedLIGOGuardian.pdf
- Rollins, J. (2017, July 20). Machine learning for lock loss analysis. Retrieved from https://dcc.ligo.org/public/0144/G1701409/001/main.pdf.
- Documentation of scikit-learn 0.20.3. (2018). Retrieved from https://scikit-learn.org/stable/documentation.html
- Documentation of tsfresh. (2019). Retrieved from https://tsfresh.readthedocs.io/en/latest/
- Documentation of EntroPy. (2019). Retrieved from https://raphaelvallat.com/entropy/build/html/index.html

LIGO

Acknowledgements

- My mentor, Jamie Rollins
- The LIGO SURF program and NSF
- My fellow SURF students

Thank you! Questions?