



Sandia National Laboratories

Operated for the U.S. Department of Energy's

National Nuclear Security Administration

by **Sandia Corporation**

FORTÉ Machine Learning

Matthew Kagie

July 19, 2016

Prepared by

Sandia National Laboratories

Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Recommended for Public Release Approval by

Name/Org: dcreviewer unset/dcdept unset

Date: dcdate unset

Contents

1	<i>Executive Summary</i>	1
2	<i>Introduction</i>	2
3	<i>Machine Learning</i>	4
4	<i>Noise Separation</i>	5
4.1	<i>Approach</i>	5
4.2	<i>Results</i>	8
4.3	<i>Other Methods Tried</i>	9
5	<i>Identifying Lightning Families</i>	10
5.1	<i>Approach</i>	10
5.2	<i>Results</i>	12
6	<i>Conclusion and Future Work</i>	18
A	<i>Groups</i>	20
B	<i>Deciles</i>	24
C	<i>Noise Synthesis</i>	29
D	<i>Data</i>	32
E	<i>Acronyms</i>	32

1 *Executive Summary*

Of the non-corrupted data collected by the Orbiting Experiment (FORTÉ) satellite's Photo-Diode Detector during the year 2001, I estimate that 7.9% of 914 894 signals are noise. My result differs dramatically from Guillen's estimate of 96%¹.

To arrive at this estimate, I used Gaussian mixture model (GMM) clustering—unsupervised machine learning—to aggregate the waveforms into groups based on the absolute value of the lowest 25 positive-frequency discrete Fourier transform coefficients. Then, I marked several of the groups as noise by inspecting a random sampling of waveforms from each group. Marking groups as either *noise* or *non-noise* is a supervised binary classification operation.

¹ Guillen did not publish, but his actual filtering results were available to me, and I derived the statistics from the size of his filtered data set.

After removing the signals in *noise* groups from further consideration, I clustered the remaining signals into families. Again, I used a GMM, but for the familial clustering I used a Non-Negative Matrix Factorization feature vector transform. The result was 9 distinct families of lightning signals, as well as a second stage of noise filtering.

To efficiently represent the entirety of the signal space, I broke each family into deciles based on their distance from the family mean. In this case, *distance* means the log-likelihood based on the GMM. Signals in lower deciles are more similar in shape and amplitude to their family average. I took the top 200 samples from each decile of each group, resulting in 18 000 signals. These signal approximately represent the entirety of the FORTÉ observations. To represent outliers, I also kept a *zoo* of the 1000 signals furthest from any family's average. All told, the resulting data set represents the FORTÉ data with a reduction of about 51:1.

To allow synthesis of an arbitrarily large number of test signals, I also captured each family's average signal and the time-sample covariance matrix over the signals in each family. Using these two pieces of information, I can synthesize new waveforms by using a Gaussian random realization from the family average and covariance matrix. I wrote a program to test the synthesis quality. The program shows me two signals on the screen, one synthesized and one randomly drawn from the data. I attempted to identify the synthesized signal. Although the synthesis is imperfect, in an A/B comparison I only correctly chose the synthesized signal 36% of the time.

2 Introduction

About 100 lightning strikes occur every second, giving the world plenty of lightning data to analyze². This paper used data collected by the FORTÉ satellite, a small satellite jointly launched by Los Alamos National Laboratory (LANL) and Sandia National Laboratories (SNL) in 1997 to collect optical and radio-frequency transient signals from lightning. An unshielded Photo-Diode Detector detects and records the optical signal at a sampling interval of 15 microseconds (μsec). Each lightning intensity profile analyzed in this paper consists of 128 samples, or 1.92msec of data³. This paper uses one year's FORTÉ Photo-Diode Detector (PDD) data, from 2001. The data set consists of 914 894 uncorrupted signals, that is, signals that contained all 128 samples.

As can be seen in Fig. 1, the detected signals include noise-like signals, ramps, particle excitations (signals too sharp to be lightning strikes,

² "Forte lightning data," Sandia National Laboratory, Tech. Rep., ca 2003

³ "Forte lightning data," Sandia National Laboratory, Tech. Rep., ca 2003

caused by high energy particles striking the PDD), various shapes for lightning intensity profiles, and other phenomena. The diversity of signals, along with the discrepant results from other filtering techniques, motivates more advanced analysis.

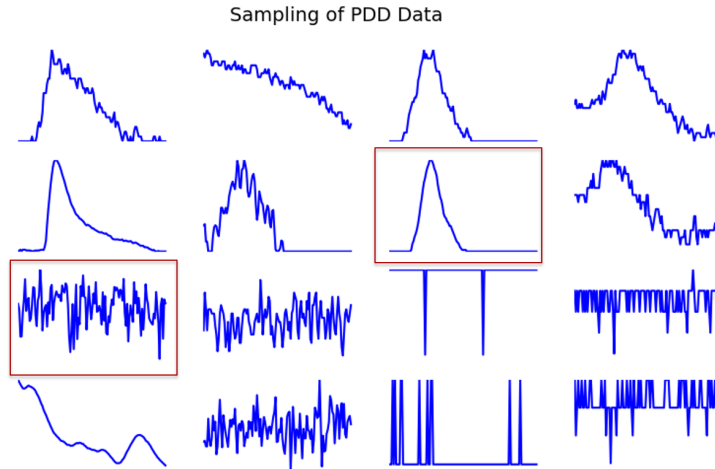


Figure 1: Examples of signals from 2001 FORTÉ PDD data. Highlighted signals will be examined later on.

Others' analyses eliminated between about 10% and about 96% of the FORTÉ triggers. Members of the technical staff of LANL exclude the 10% of the data collected during the day time, as they viewed it as too noisy for statistical analysis. Their analysis used 5 years of FORTÉ PDD data, from 1998-2002⁴. José Guillen, a member of the technical staff at SNL, removed 96% of the year 2001 triggers, leaving only 4% for further analysis. An unknown member of the technical staff of LANL filtered between 68-70%.

This paper describes a method to differentiate between actual lightning strikes recorded by the PDD and noise that was mistakenly collected, and reports on some results from applying that method. I develop a machine learning method to classify and remove noise signals at least as well as I could do by personally inspecting the data. Of the 914 894 signals, 72 096 were classified as noise (about 7.9%), which includes noise signals, particle excitation events, DC readings, ramps, and other non-lightning signals.

Once the data is filtered such that only lightning intensity profiles are present, I create familial clusters of lightning signals, as well as representative signals for each family. This is done to assist with statistical models of lightning occurrences world-wide as seen by the FORTÉ sensor, in addition to allowing for efficient sampling of the lightning space. I find 9 families of lightning and sample 18 000 signals based off

⁴ W. B. T.E.L. Light, S.M. Davis and A. Jacobson, "Global nighttime lightning flash rates and characteristics observed with the forte satellite," Los Alamos National Laboratory, Sandia National Laboratory, and Niagara University, Tech. Rep., jul 2003

of the probabilistic distribution of which family each signal belongs to. Finally, the “worst” 1000 signals in the filtered data set are used to create an anomaly data base for further analysis.

FORTÉ PDD records its signals in *counts*, which is related to the photocurrent through a piecewise linear transfer function. The three-stage gain compression of the optical signal supports larger dynamic range than a simple linear conversion. I used the counts data for noise filtering. The photocurrent is proportional to the optical irradiance, and is recovered from the count data by passing the signal through the functional inverse of the compression. This data, called the *irradiance* signal, is reported in units of irradiance (W/cm^2). I used the irradiance representation for all other analysis.

This paper is organized in three main parts. Section 3 consists of a brief overview of Machine Learning and its application to this paper. Section 4 discusses methods and results for filtering out most of the noise signals from the database. Section 5 explains the methods, results, and initial analysis of creating familial clusters of lightning.

3 Machine Learning

The purpose of this section is to give a *very* brief overview of Machine Learning (ML) and its applications in this paper. While there are many different types of ML algorithms, two are of importance to this paper, and those include unsupervised and supervised ML. Each will be discussed in more detail below.

The ML terminology in this paper is as follows. The inputs to the algorithms $x_1, \dots, x_n$ are called “feature vectors”, and consist of a vector of numbers. Each member in this vector is referred to as a “feature”. These feature vectors do not have to be the 128-dimensional time-series signal from the database. Instead, the PDD records can be processed through a “feature vector transform” that represent the same signal in a different way. This is described in more detail in later sections, as the transforms are application-specific.

One major area of machine learning is called Unsupervised Machine Learning (UML). UML is used for density estimation and data visualization, and for this paper UML is used to find structure in a given data set. For that reason, the training data for a UML algorithm consists of a set of input feature vectors, i.e. a set of waveforms, without any sort of labeling or any assumptions from the implementor. The UML algorithm looks for groups of waveforms that are most similar to each

The only assumption required is that all feature vectors have the same number of samples per waveform.

other, where the definition of similarity is application-dependent.

In contrast to UML, Supervised Machine Learning (SML) requires *labeled* data. The purpose of supervised machine learning is to assign an input feature vector an output, like a category, based on a learned function. This paper is concerned with classification, which is mapping each input feature vector to one category from a list of categories. To separate signals into either *noise* or *non-noise* is a binary classification task. To train and test the algorithm I created a *tagged* data set from a subset of the original data. I extracted a subset of the data and hand-labeled each waveform as either noise or non-noise. I then used this subset to test the performance of a SML algorithm. The separation of training and testing data is key to keeping the results of the SML algorithm uncorrupted⁵.

⁵ C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer Science + Business Media, LLC, 2006

4 Noise Separation

This section explains the derivation of the methodology and results of our noise separation technique.

4.1 Approach

The process for noise removal is explained briefly here, and each step is explained in more detail below. I created two data sets, a *testing* set and a *training* set. Each record in the testing subset was hand-labeled as either noise or non-noise. The training data was processed through the feature vector transform, and then a UML algorithm grouped the feature vectors into 10 clusters, and each group was inspected and labeled (as a whole group) as either noise or non-noise. The clustering model, learned by the algorithm, along with the post-clustering labeling, constitute a binary classification algorithm. Any waveform passed through the feature vector transform and then clustered will be labeled as noise or non-noise by the algorithm.

Applying the binary classification algorithm to the testing data produces an estimate of the expected effectiveness of the algorithm. Finally, I use the derived algorithm to filter the entire original data set.

The 2001 FORTÉ PDD data contains 914 894 signals. From these I removed 880 signals for testing purposes. 440 of these records were non-noise, and 440 were noise. The non-noise subset contained a wide variety of the observed lightning shapes (including low SNR signals), while the noise signals contained examples of noise, particle

excitation, ramps, and other noise-like signals. Fig. 2 shows examples of the testing set.

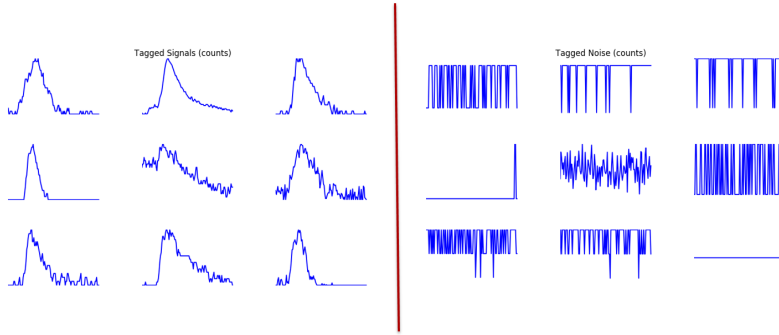


Figure 2: Examples of tagged test set.

The feature vector transformation is one of most important elements to this method. After examining many noise and non-noise signals it became apparent that the noise signals contained different frequency content than the lightning signals. The noise signals either consisted of high frequency content, almost strictly DC frequency content, or a few impulses. Therefore, I chose to group in the frequency domain, rather than the time domain.

After experimentation with various truncations of the discrete cosine transform (DCT) and magnitude fast Fourier transform (FFT), I settled on the absolute value of the first 25 positive-frequency coefficients of the 128 point FFT. These 25 values from each signal were the feature vector.

Fig. 3 shows an example of the feature vectors of the signals previously marked red in Fig. 1. The noise has a significantly higher direct current (DC) value, as well as frequency content across all coefficients. The lightning signal, on the other hand, has less magnitude at all coefficients, and the vast majority of the energy of the signal is contained in the first 10 coefficients. Differences like these are obvious in the feature-vector domain and allow the clustering algorithms to differentiate between groups.

Once these feature vectors are created, I apply data preprocessing methods to better prepare the data for clustering. These methods include making each feature zero-mean and scaling each feature to an absolute maximum of 1. It is important to remember the scaling factors used in this step, as the algorithm expects all new data being classified later on to be scaled by the same factors.

Once the data has been transformed and preprocessed, it is ready to

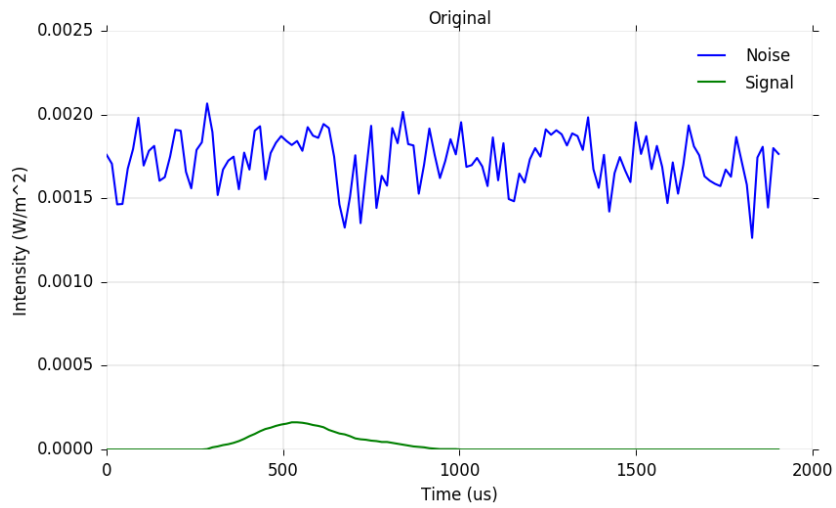
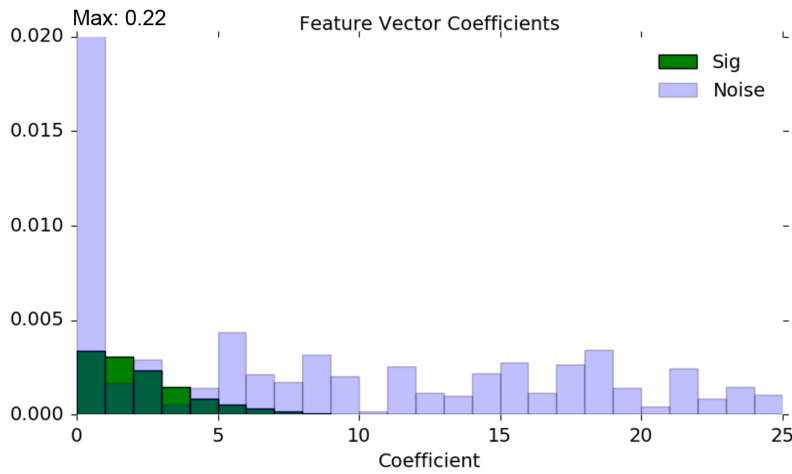


Figure 3: Noise separation feature vector example.



be clustered. I used Gaussian mixture model for clustering. GMMs are a linear superposition of Gaussian probability density functions. For the purpose of noise separation, the algorithm utilizes 10 different Gaussian distributions summed together. The idea behind a GMM is to iterate an Expectation-Maximization algorithm to maximize the log-likelihood of each feature vector being assigned to one of the 10 Gaussians. For more information, see ⁶.

The GMM algorithm clustered the training data set into 10 different groups. Examples of the clusters are in Figs. 4 and 5. I visually analyzed a random sample from each group and determined if the group represented non-noise or noise for the binary classification. Groups highlighted as red are noise groups.

⁶ C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer Science + Business Media, LLC, 2006

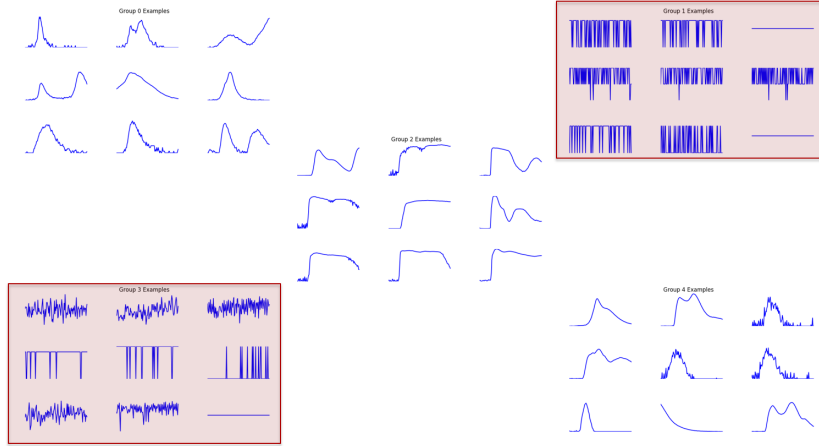


Figure 4: Groups 0-4.

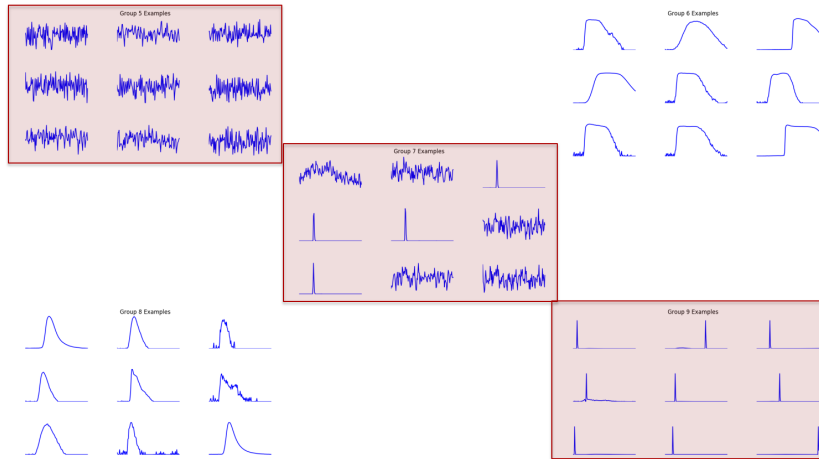


Figure 5: Groups 5-9.

4.2 Results

This section discusses the results of the noise separation methodology. First, I describe the performance, viz. the error rate, of the classification algorithm on the test data set and then report the noise statistics after the classification algorithm is applied to the entire data set.

Thirteen of the 880 tagged waveforms were misclassified, giving a misclassification rate of 1.7%. Fig. 6 shows the misclassified signals. Inspecting those signals, it is clear that human understanding and classification is as much a factor as the classification algorithm itself. For example, the first two misclassified signals are ramps. I originally tagged these as non-noise, since there appears to be structure in the signals. However, the signals are very different from a typical light-

ning intensity profile. Perhaps they are noise, or rather should be treated as noise regardless of their cause. The remainder of the misclassified signals I labeled as noise. However, many signals shown definite structure, and perhaps they are lightning signals showing quantization artifacts. The definition of noise and non-noise signals is not always clear and well-defined. These structured “noise” signals play an important role later in this paper, because they are not filtered out after the application of this algorithm.

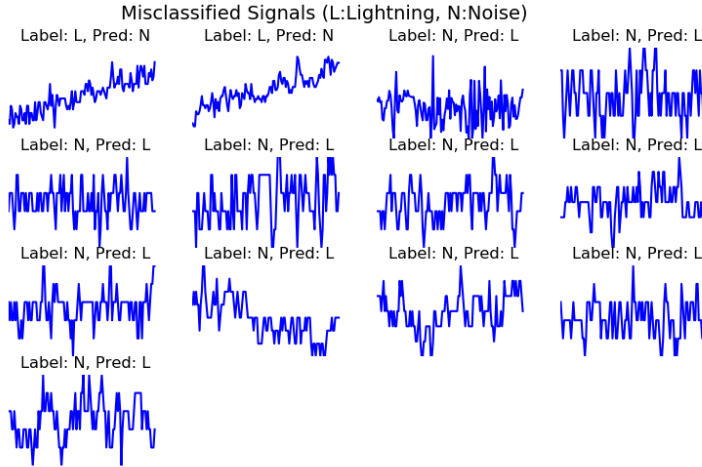


Figure 6: Misclassified signals.

Application of the classification method found 38 269 records were noise (4.2%). This number is significantly lower than previous work. These signals were removed from the data base, leaving 876 625 signals for familial cluster analysis—that is for the remainder of the paper.

4.3 Other Methods Tried

I selected the magnitude of the first 25 coefficients of FFT as a feature vector, with GMM as a clustering method based on its performance. The rejected methods, however, deserve some mention. All my methods followed the “cluster then classify” framework developed above, but I used several different combinations of feature vectors and clustering methods. In addition, I attempted one strictly classification method.

The feature vectors I tried included the time-series signal itself, a low-pass filtered version of the time-series signal, the first n coefficients of the DCT, Non-Negative Matrix Factorization basis expansion co-

efficient, and Dictionary Learning basis expansion coefficients. FFT coefficients generated the least number of misclassifications on the data set. Furthermore, its errors were the most difficult for a human to adjudicate indicating that the feature vector was performing about as well as a human.

Although GMM was my choice of clustering, I also clustered with k -means. However, the noise and non-noise separation appears to be non-spherical, and k -means produced very poor results. I also attempted a manifold clustering method called *spectral clustering*, which is very memory expensive, and could only be successfully completed on 1% of the data though would likely have produced more satisfying results. For that reason, I selected the much more memory-efficient GMM method.

I also trained and tested a Support Vector Machine on the data set. While the expected accuracy of the algorithm was similar to the chosen method, the training and application complexity and time were much larger. Support Vector Machines are not as scalable as the “cluster then classify” method.

5 Identifying Lightning Families

This section discusses the methodology, results, and analysis of the generation of the familial clusters.

It is instructional to look at a visualization of the filtered data set, like that in Fig. 7. The figure displays a data visualization technique known as Principal Component Analysis (PCA) (another UML technique). By plotting the data against the two most significant components, the entire high-dimensional set can be plotted. The PCA data has no obvious clusters. Rather, there is a large cluster in the middle with outliers on the fringe. With these observations we might not expect obvious familial clusters with low intra-cluster variance.

5.1 Approach

The approach for generating familial clusters is described briefly here, and in more depth in the following sections. I create a basis over the irradiance data using a technique called Non-Negative Matrix Factorization, described below. Each signal is projected onto this basis, and the coefficients of projection are considered the feature vectors. Clustering ensues.

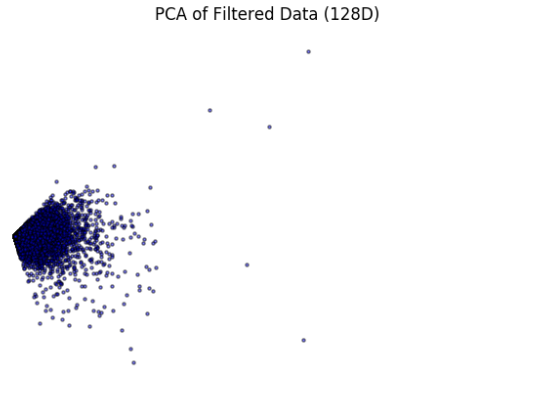


Figure 7: PCA of time-series signals.

Non-Negative Matrix Factorization (NNMF) approximates a matrix as a linear summation of positive components. It is analogous to the idea of creating a face by building it from scratch, starting with adding a head, then the eyes, the mouth, and so on. This sort of analysis of a matrix reveals the most important positive components to build up the matrix.

Formally, the idea is to solve the equation of

$$\min_{B \in \mathbb{R}^{d \times r}, C \in \mathbb{R}^{r \times n}} \|X - BC\|_F^2, B \geq 0, C \geq 0 \quad (1)$$

If we treat X as a matrix made up of waveforms (each column is a waveform), then, after solving (1), B represents a basis of the most important r positive waveforms, and C represents the projection coefficients onto the r waveforms for each of n initial waveforms.⁷

Fig. 8 shows the set of basis waveforms (B). These are used in the generation of the feature vectors, as described in the next section.

Once the NNMF basis set is created, generating the feature vectors is as simple as projecting each input signal onto the basis set. However, each input signal must be preprocessed before projection by normalizing the signal so that its L2 norm is unity. These coefficients are found in the matrix C . The 25 projection coefficients make up the feature vector. Fig. 8 shows the basis set consists of Gaussian-like bumps shifted in time, in a way downsampling the signal. This representation assists in identifying the shape of the signals. After the NNMF feature vector transform I apply a normalization technique identical to that done for

⁷ J. Romberg, "Structured matrix factorization," 2016, Georgia Tech ECE 6254 Spring 2016 class notes

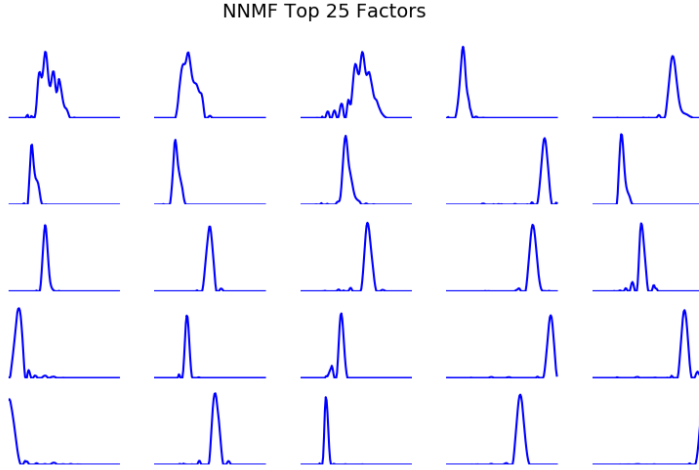


Figure 8: NNMF basis waveforms.

the FFT feature vectors in Section 4.1 to improve clustering efficiency.

Fig. 9 shows an example of the transform. The time-series representation of the signals is plotted in Fig. 9(a), while the transformed feature vectors (without normalization) can be seen in Fig. 9(b). The amplitude and time-shift differences are clearly visible in the NNMF domain.

Compare the PCA plot of the time-series signal in Fig. 7 to the NNMF signal in Fig. 10. While the signals still do not show evident clusters, they are spaced more uniformly and widely. This improves clustering.

With a new GMM, I clustered the records using transformed feature vectors into 10 clusters. These clusters are different than those in the noise filtering, as they are based on temporal shape, rather than frequency content.

5.2 Results

This section discusses the analysis and results of the familial clustering. First I discuss representing signal families with their family mean and covariance. Then, I analyze each group individually. I also describe how, independent of the mean-and-covariance representation, I capture an 18 000 signal sample that describes the families and a set of outliers. Finally, I demonstrate a lightning signal synthesis method based on the mean-and-covariance results.

As explained in the previous section, 10 groups were generated by the GMM clustering method. However, only 9 of the groups are statis-

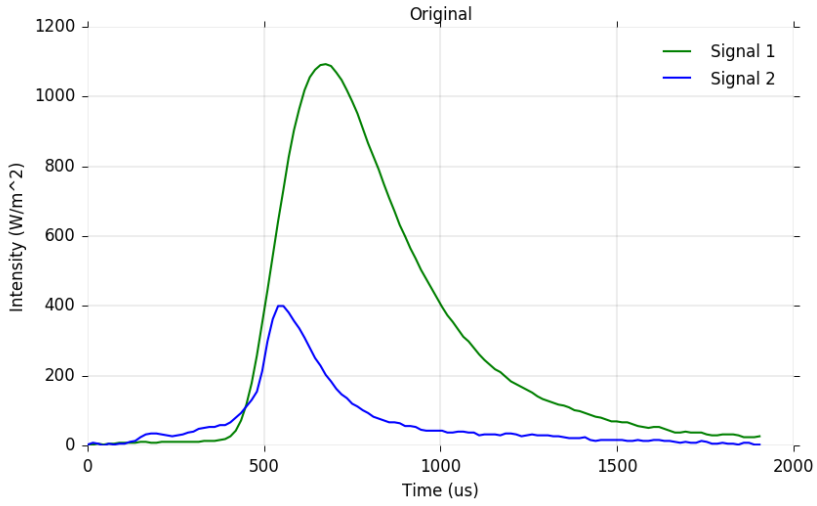
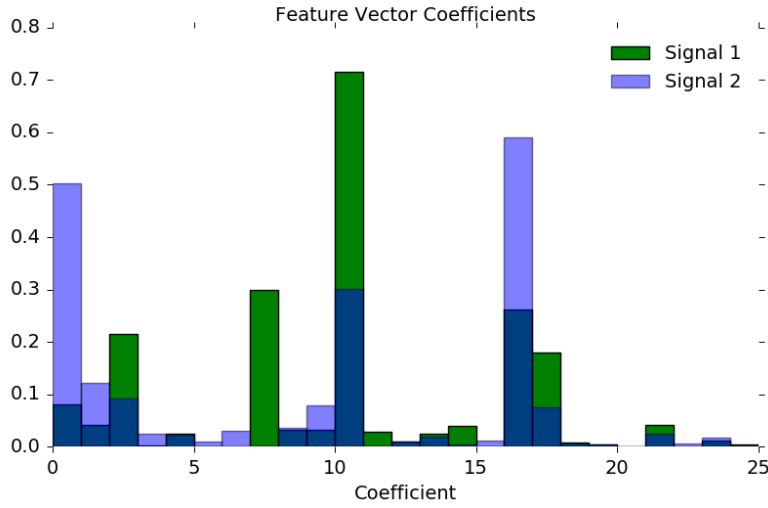


Figure 9: Familial clustering feature vector example.



tically useful. As mentioned in Section 4.2, the original noise filtering method misclassified very noisy signals with non-white structure. These signals are mostly useless. Fortunately, they tend to group together as a result of familial clustering. Therefore, the familial group containing the useless signals is removed in the rest of the analysis as a second stage of noise filtering. This group contains 33 827 signals (3.7%) which together with the first stage's 4.2% results in the 7.9% total noise fraction.

The representative signal for each of the remaining groups is defined as the average, or mean, signal of the group. It is the centroid of the

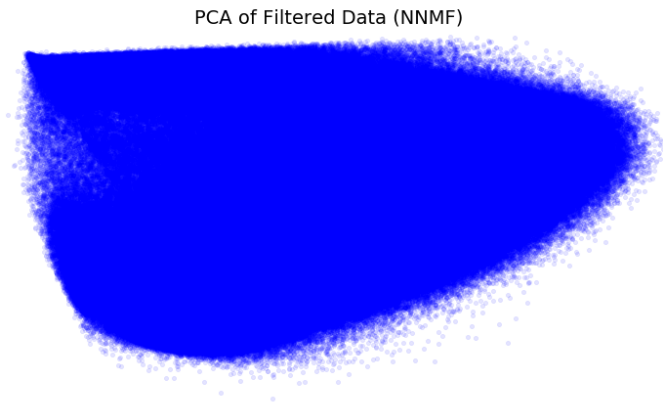


Figure 10: PCA of NNMF feature vectors.

cluster. Fig. 11 shows these signals (Group 1 is the noise group filtered in this stage). As the figure shows, 9 distinct families of lightning are present, though groups 2 and 7 are very similar. Nonetheless, analysis of the clusters individually differentiates the families. Fig. 12 displays the breakdown of the data set. Group 5 contains 35% of the signals recorded, by far the largest group. This sort of analysis can be used later for generation of a statistical model for lightning signals.

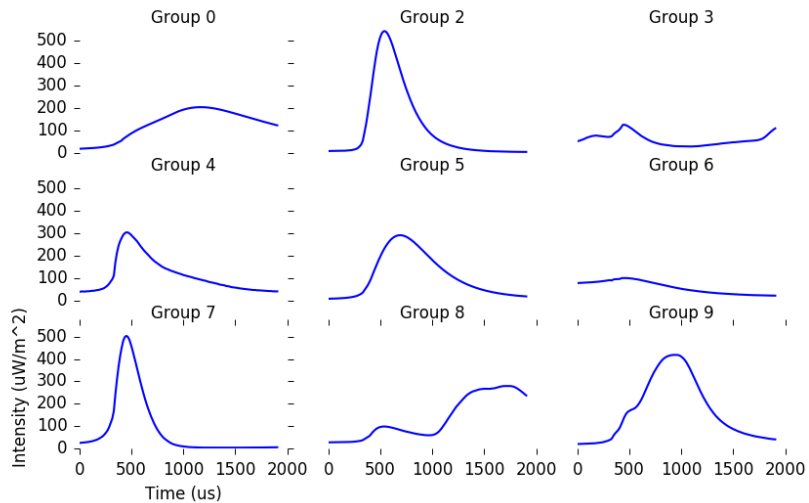


Figure 11: Group representative signals.

Fig. 13 is the same figure as Fig. 10, but labeled with group assignments. The NNMF feature vector transformation spreads out the input

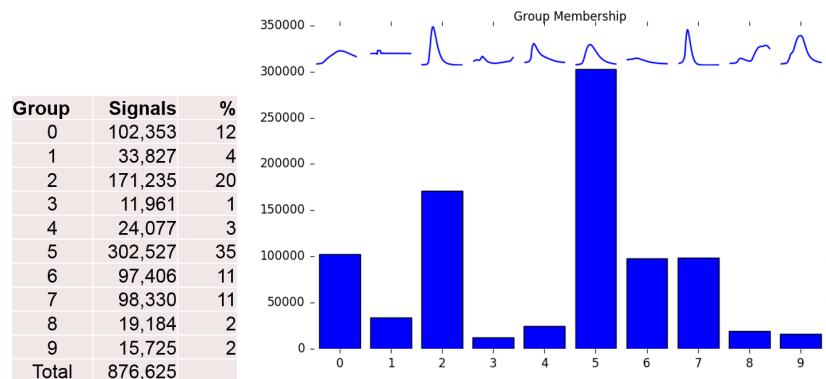


Figure 12: Data set broken down by group.

space, allowing for sections to be carved into clusters.

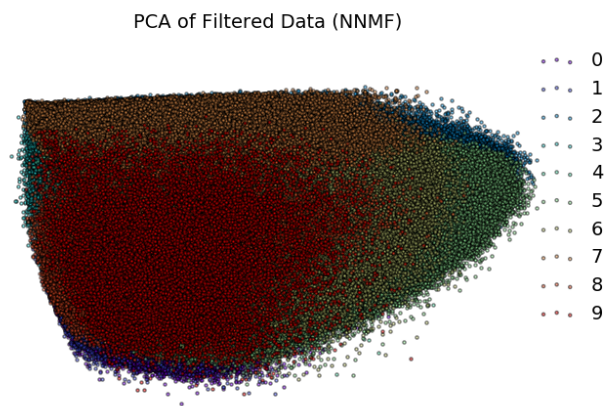


Figure 13: Labeled PCA of NNMF feature vectors.

Fig. 14 shows the mean signal and covariance matrix of Group 0; the remaining groups can be found in the Appendix. The covariance is of the time samples from within each family. Therefore, the portions of the plot that are yellow indicate parts of the lightning intensity profile that vary widely throughout the family. Group 0 has small variance at the front end, increasing dramatically throughout the hump of the profile. Group 0 can be described as starting low and flat, but having a wide hump starting around sample 20 (300 μ sec). Other groups have their own characteristics. For example, Group 7 is defined by signals with flat beginnings and ends and a very sharp hump. The magnitude of the peak of the hump is the variance of the group.

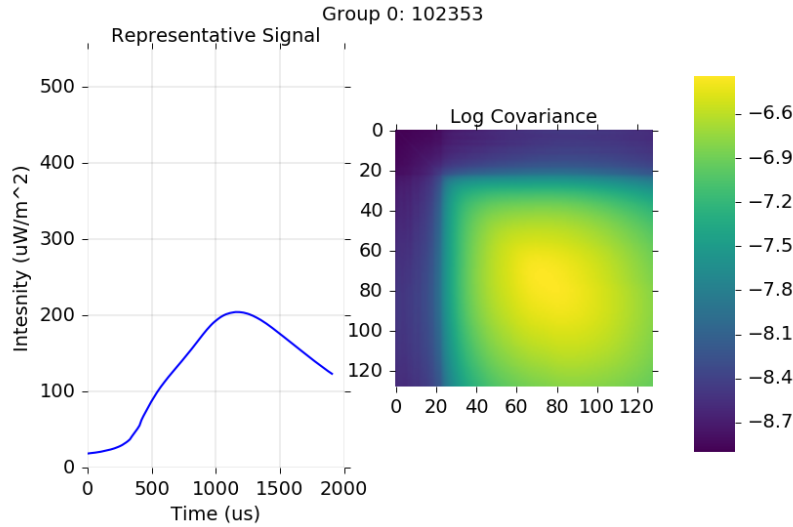


Figure 14: Group 0 representative signal and covariance matrix.

To build a set of signals that cover the entire data space, a cluster is broken into 10 deciles based on the distance metric of the GMM. The GMM distance metric is the log-likelihood of the signal under the GMM. The histogram of this breakdown for Group 0 can be seen in Fig. 15; the rest of the groups can be found in the Appendix. In this histogram, the higher the log probability, the more confident the clustering algorithm is that this waveform resides in this group and the more like the representative signal it is. Decile 1, the nearest-to-center 10% of Group 0, lies on the far right side of the graph. A signal from Decile 1 has been plotted in the legend next to Decile 1's label in purple. It is very similar to the representative signal (also plotted in the figure). The signal from Decile 2 is also less similar to the representative signal, though still looks mainly like the representative. As deciles increase, the signals differ more and more from the representative signal of the group. The trend is consistent for all of the groups, and makes sense. The more dissimilar a signal is to the group's centroid, the smaller the log-likelihood that the signal falls in that group.

Although there is variance in each group, particularly across the deciles, it is a sliding range. Decile to decile, the signals resemble each other significantly, for example Deciles 9 and 10 are similar. This is indicative that, although signals in Decile 10 are not obviously similar to the representative signal of Group 0, they are similar to other signals in the group.

With the decile representation, I sample the space of lightning intensity profiles by including elements from each group, and each decile

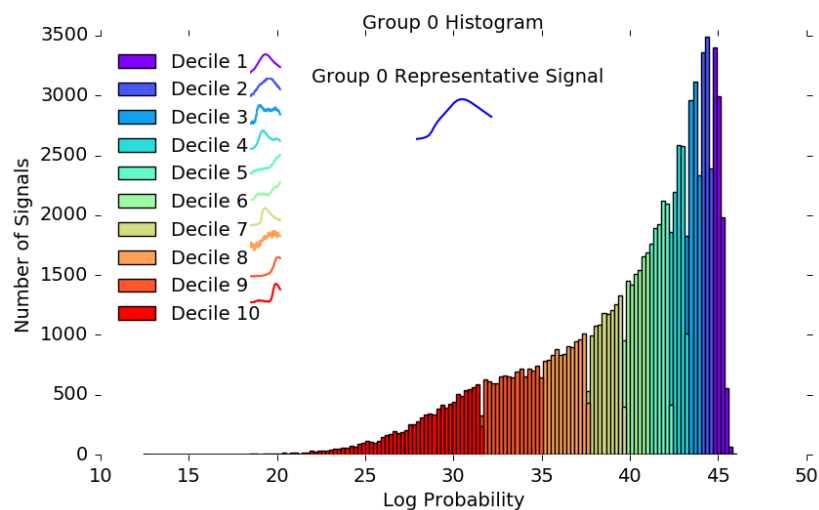


Figure 15: Group 0 decile break down.

of each group. Including samples from each decile mitigates the deficiency of large intra-cluster variation. By taking 200 of the signals with the highest log-likelihood from each of the 10 deciles of each of the 9 groups in this data set, I get a data set of 18 000 signals. The signals are included at the end of this paper.

Because the clusters show much overlap in the PCA representation, an additional zoo of the 1000 worst-classified signals have also been included. These signals are the oddballs of the lightning signals, and I believe examining them is very important. Fig. 16 shows examples from the zoo.

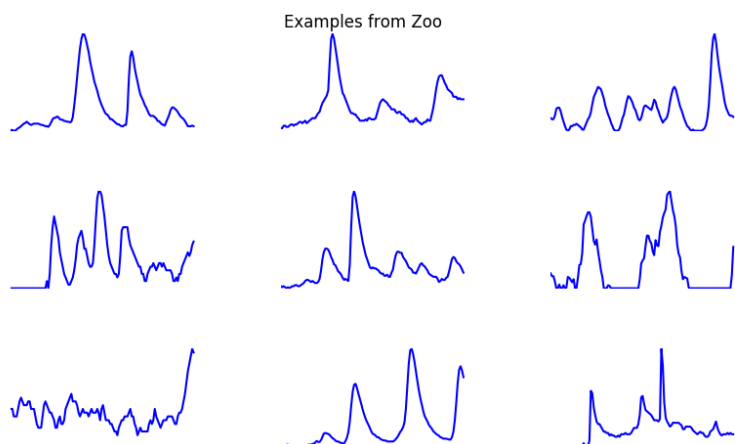


Figure 16: Examples from the zoo.

Given the mean and covariance matrix of a family of lightning signals, I can synthesize new signals. With a lightning intensity profile as a realization of a Gaussian random vector utilizing the mean and covariance matrix of a family of lightning, I can create an unlimited number of new signals. A random comparison of a true lightning signal and a synthesized lightning signal can be seen in Fig. 17. In this figure, Signal 0 is synthesized. The lightning synthesis typically creates reasonable intensity profiles.

One issue with this synthesis method is that, occasionally, intensity profiles with a very sharp, unnatural roll-off can occur (see Appendix). Further work to enforce a sort of smoothness requirement might be able to solve this problem. In addition, quantization effects of the sensor are not obvious in this method. I have hosted 2 blind tests in which I show one image of a synthesized lightning profile and one of a randomly selected profile from the data and ask an observer to identify the synthesized signal. While the initial test resulted in only 36% identification of the synthesized signal, once these two attributes were noticed, identification accuracy increased considerably.

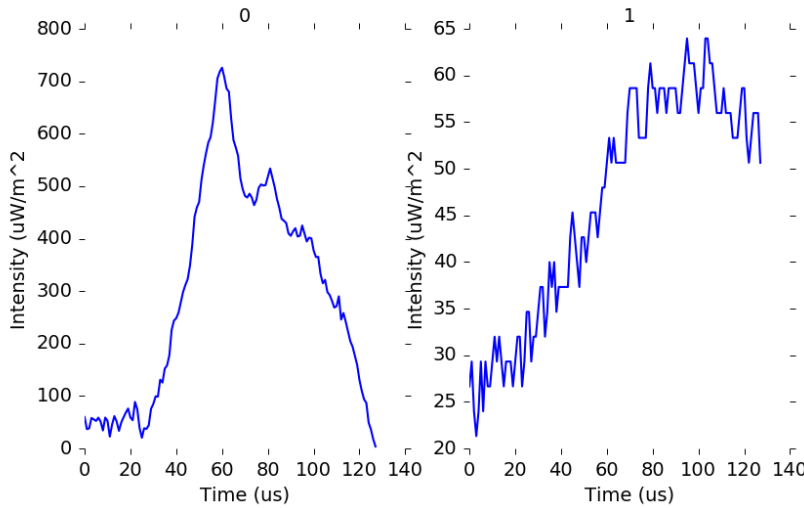


Figure 17: Noise synthesis. 0 is synthesized.

6 Conclusion and Future Work

Of the 914894 signals in the original FORTÉ PDD 2001 data set, I found 38269 to be noise in the first stage and 33827 to be noise in the second. I removed 7.9% of the data, a stark contrast to Guillen's 96%. I then divided the remaining 842798 signals into 9 different fami-

lies of lightning with representative signals and covariance matrices for use in statistical models and lightning synthesis. Just 18 000 signals efficiently cover the space of lightning intensity profiles seen by FORTÉ. With an additional 1000 signals from the zoo, I believe that our database represents the vast majority of the lightning intensity profile domain.

While this study has produced promising results, more work is still required. In terms of noise separation, the question of how we define noise must be better addressed. Are signals with very low Signal-to-noise Ratio (SNR) considered noise, or should they be included in the filtered data set? What about ramps and non-lightning looking shapes? The filtering method in this paper includes low SNR signals as lightning, but excludes ramps.

Furthermore, there is a lot of room in the area of familial clustering. While the results of this paper are positive, the intra-cluster variance is large in certain groups—as can be seen by the covariance matrix. There are visible differences inside the groups as well, especially in the higher deciles. This suggests that the space of the feature transform does not indicate all of the useful attributes of the signals. This concern is further displayed by Fig. 10, which shows one large blob of data points, as opposed to separated clusters. This is indicative that a different feature transform might produce better results.

One solution to this might be to subdivide each group into new groups, complete with representative signals and covariance matrices. As seen previously in Fig. 12, each family consists of a range of intensity profiles that slide across the space. The profiles in the further deciles differ from the representative signal, but they are very similar to the signals around them in the log-probability space. Further work might include searching for divisions inter-group.

Another question is how well the representative signals cover the entire space. While the breakdown and sampling of the deciles is efficient, a quality metric of the groups to indicate that the clusters cover the space well would be useful. The results of this paper are very promising, but more work can be done.

A Groups

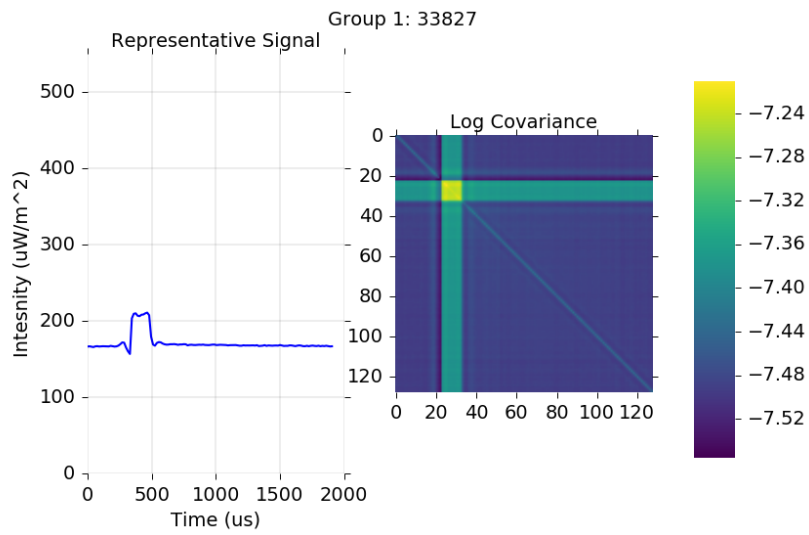


Figure 18: Group 1 representative signal and covariance matrix.

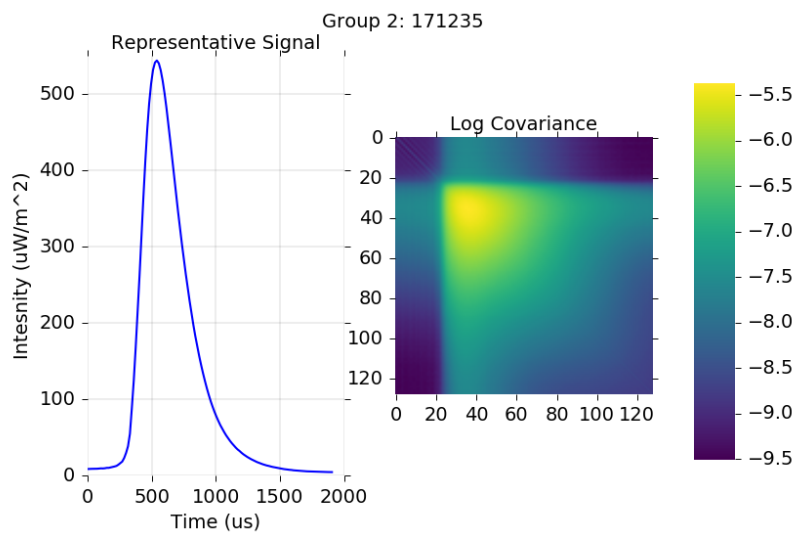


Figure 19: Group 2 representative signal and covariance matrix.

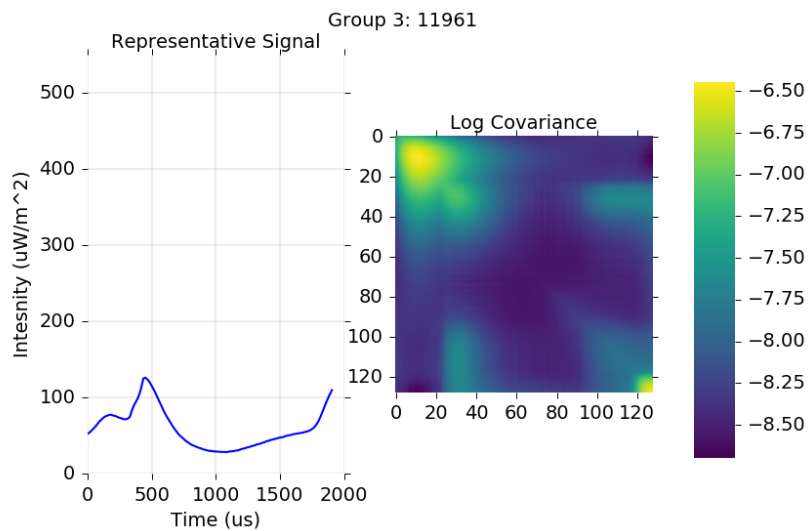


Figure 20: Group 3 representative signal and covariance matrix.

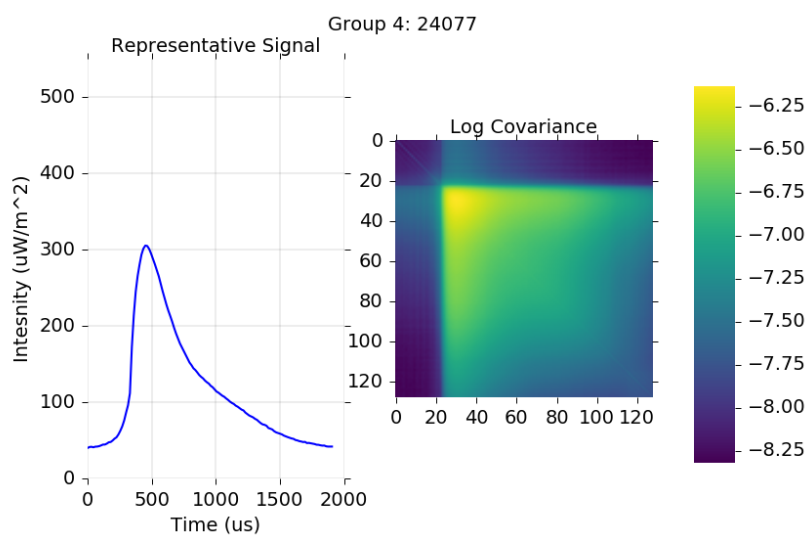


Figure 21: Group 4 representative signal and covariance matrix.

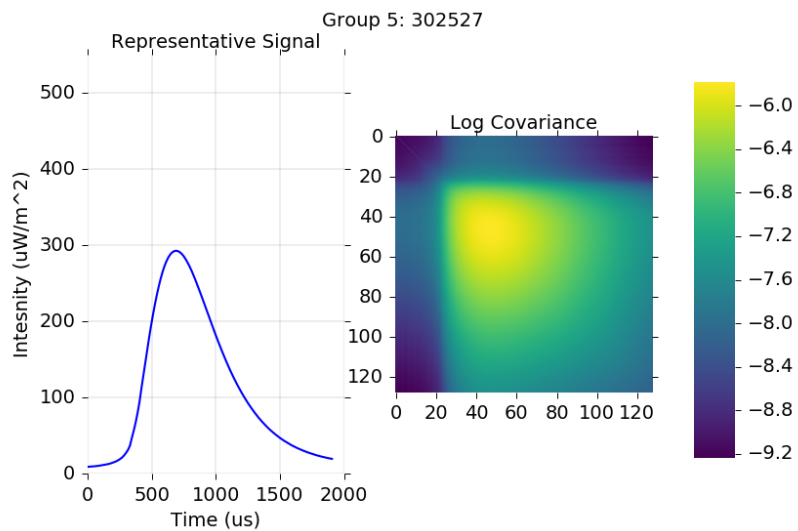


Figure 22: Group 5 representative signal and covariance matrix.

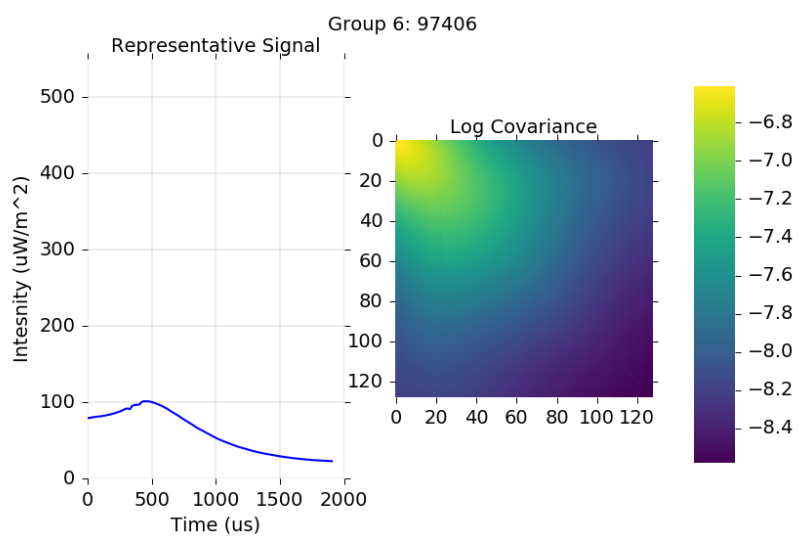


Figure 23: Group 6 representative signal and covariance matrix.

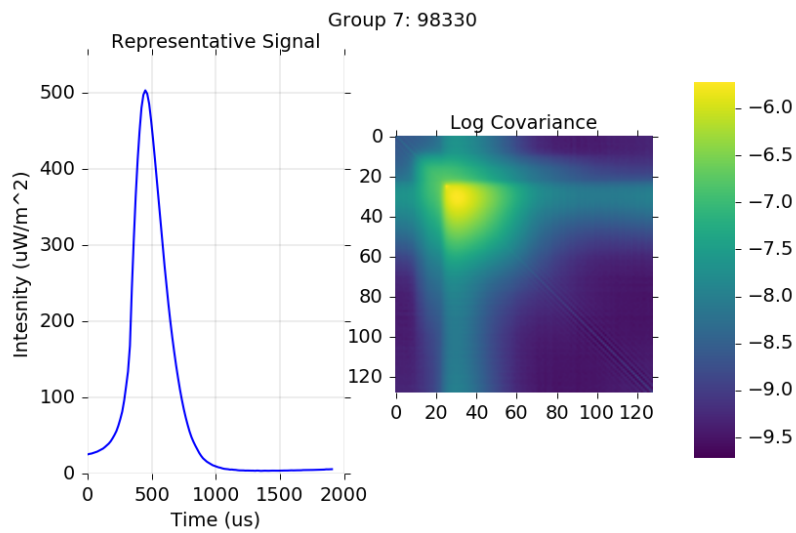


Figure 24: Group 7 representative signal and covariance matrix.

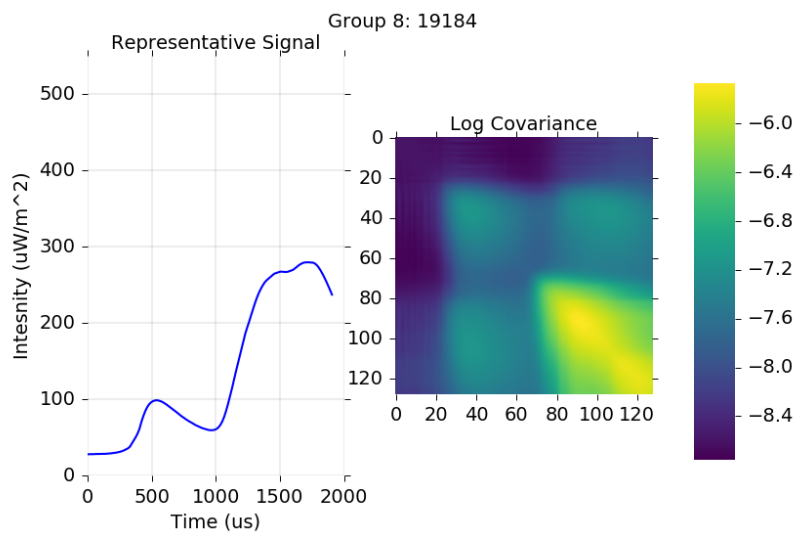


Figure 25: Group 8 representative signal and covariance matrix.

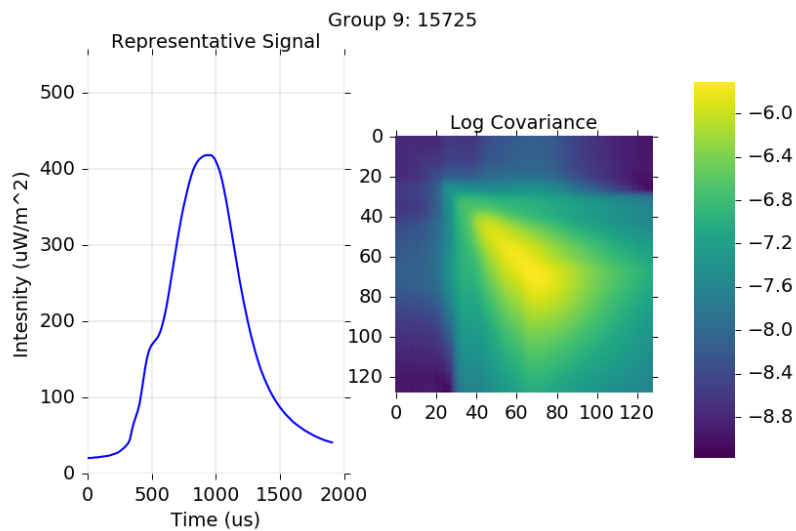


Figure 26: Group 9 representative signal and covariance matrix.

B Deciles

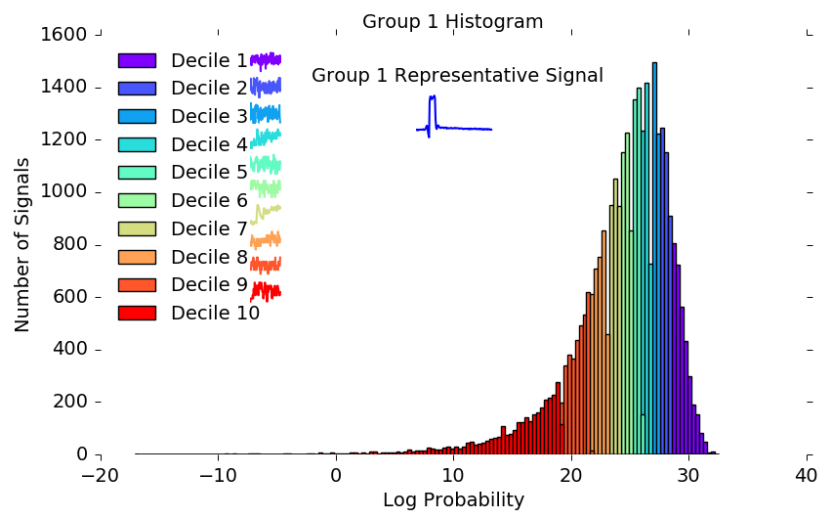


Figure 27: Group 1 decile break down.

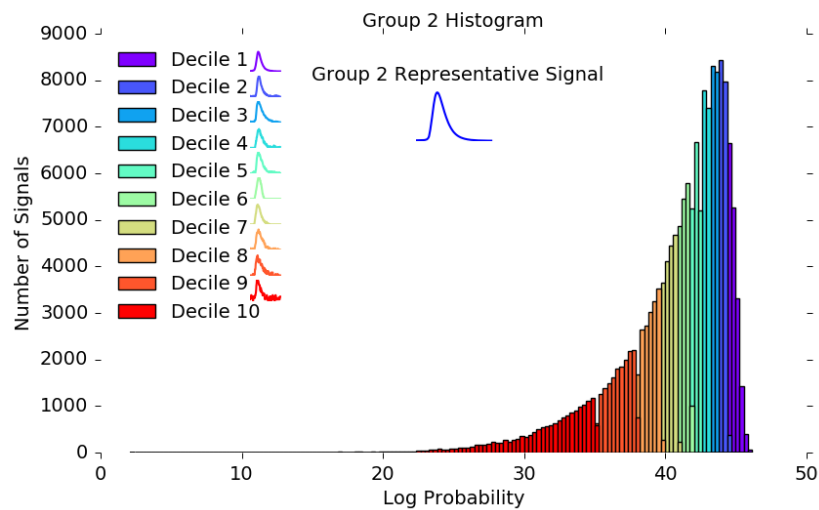


Figure 28: Group 2 decile break down.

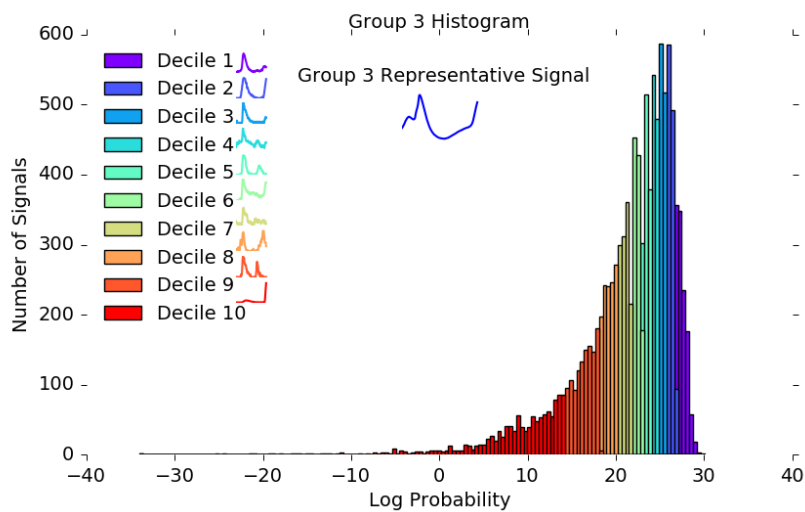


Figure 29: Group 3 decile break down.

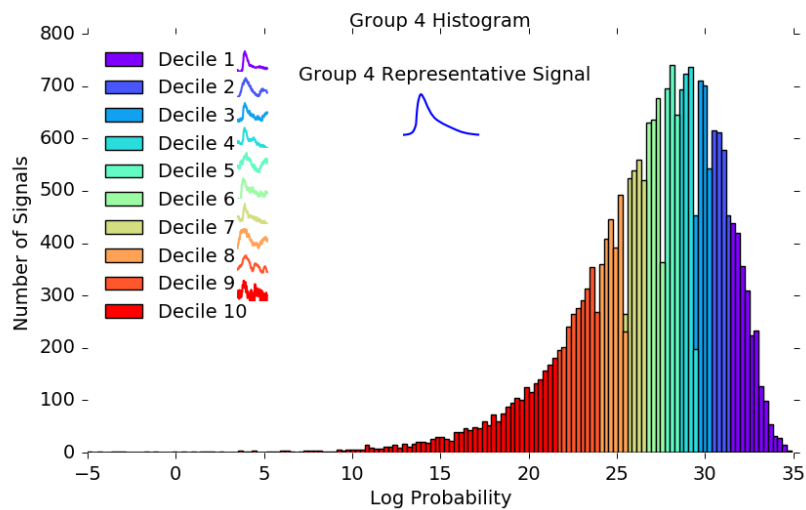


Figure 30: Group 4 decile break down.

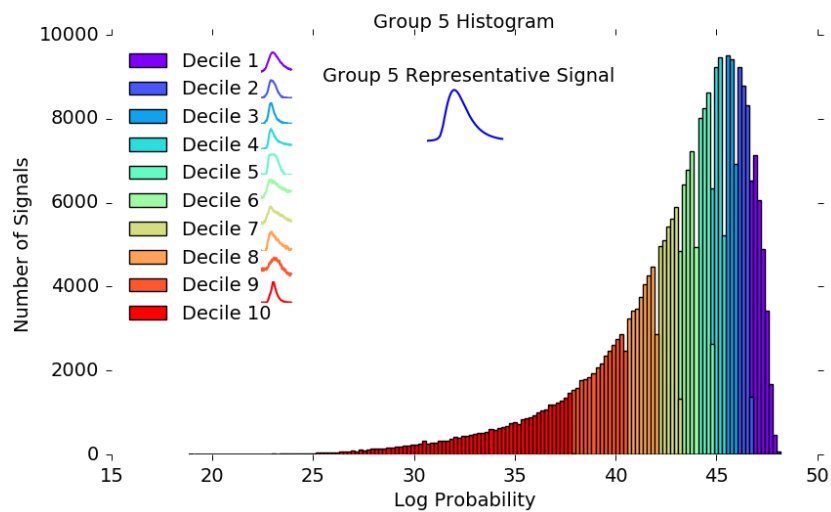


Figure 31: Group 5 decile break down.

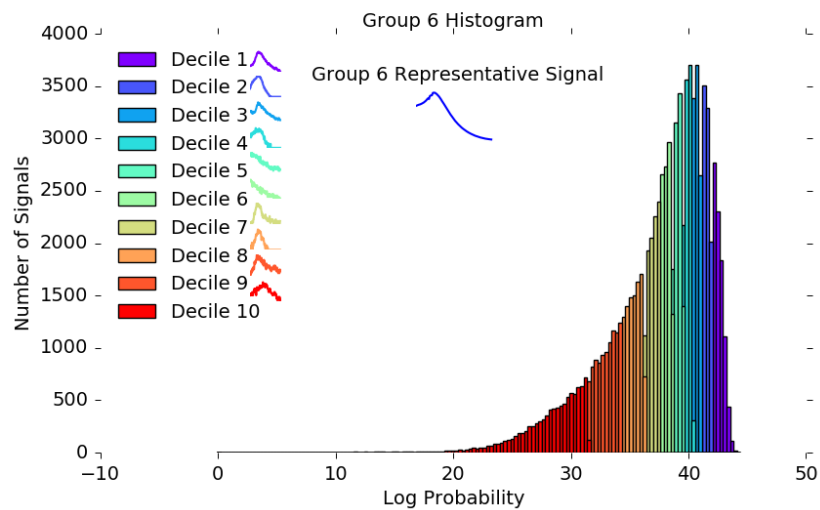


Figure 32: Group 6 decile break down.

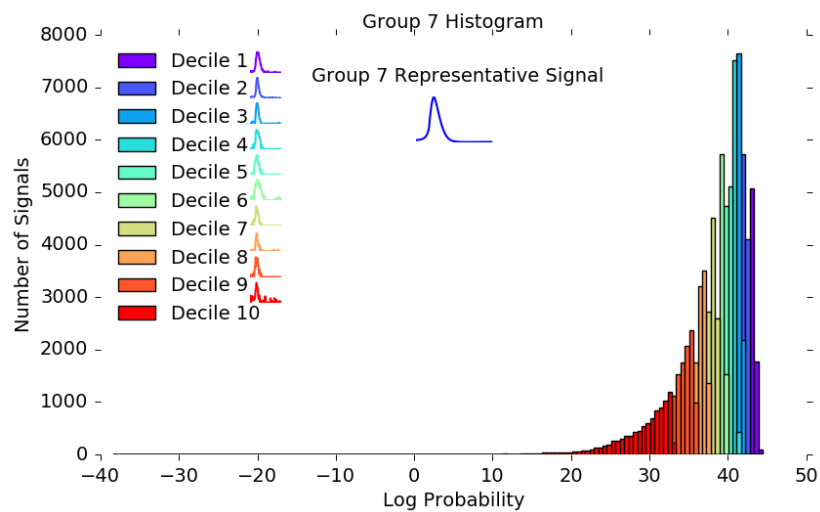


Figure 33: Group 7 decile break down.

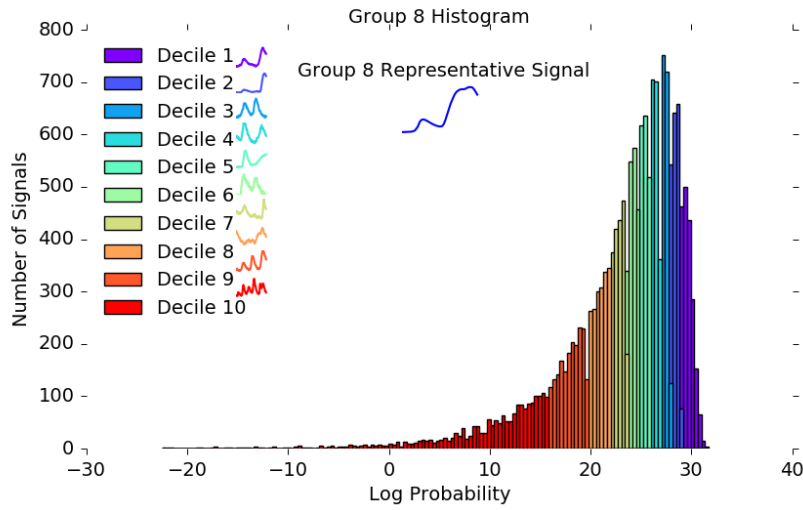


Figure 34: Group 8 decile break down.

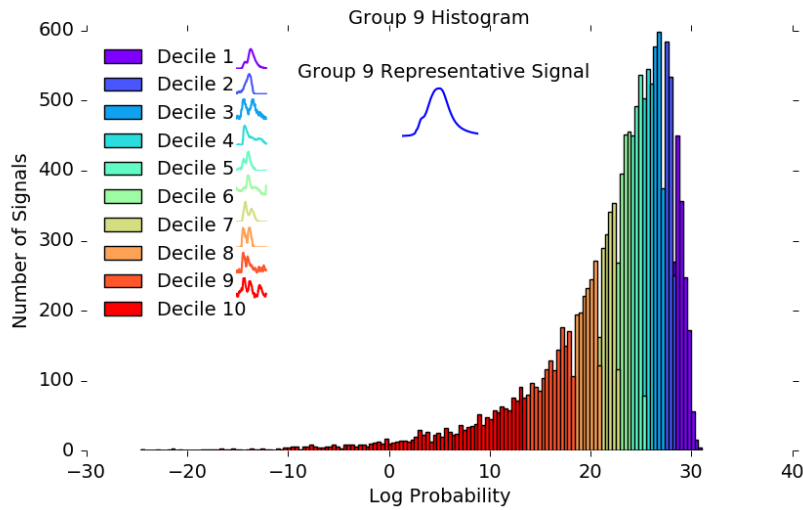


Figure 35: Group 9 decile break down.

C Noise Synthesis

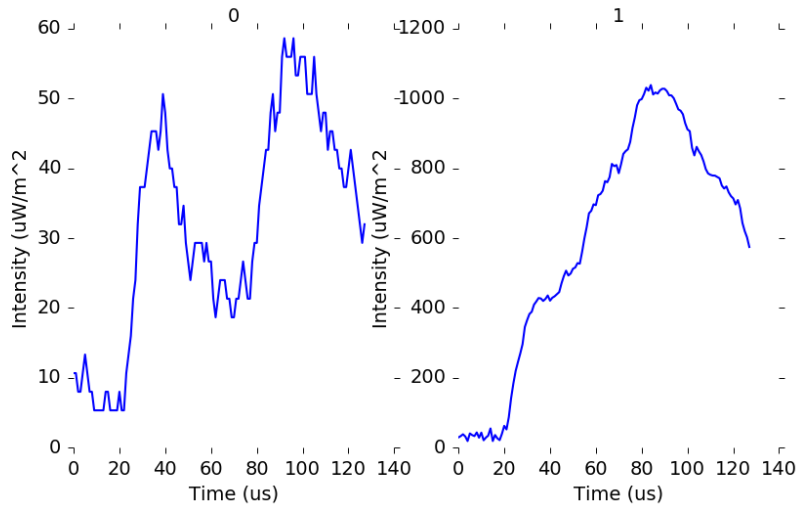


Figure 36: Noise synthesis. 1 is synthesized.

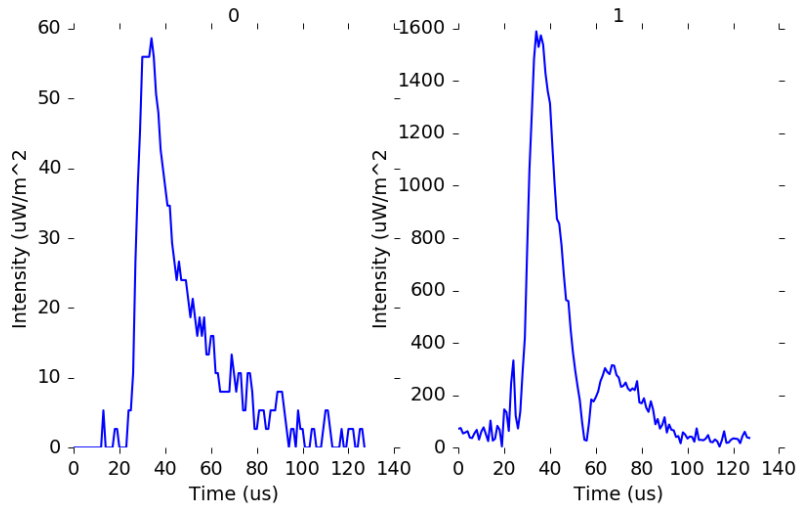


Figure 37: Noise synthesis. 1 is synthesized.

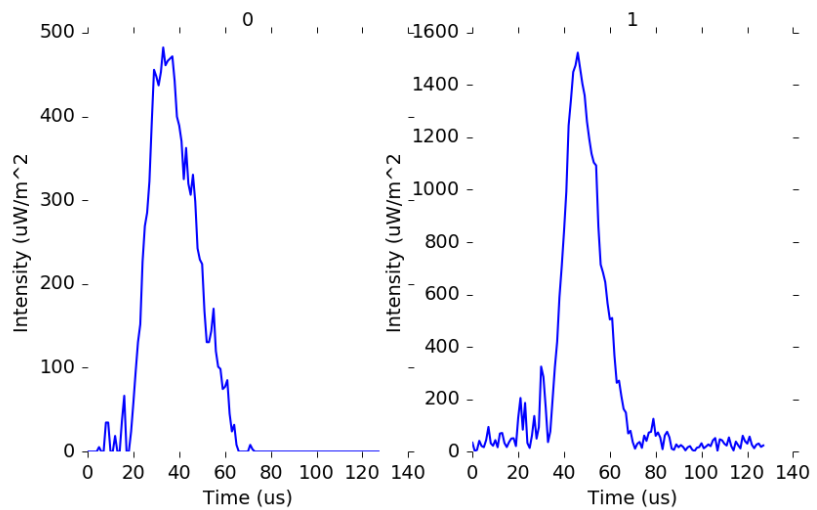


Figure 38: Noise synthesis. 1 is synthesized.

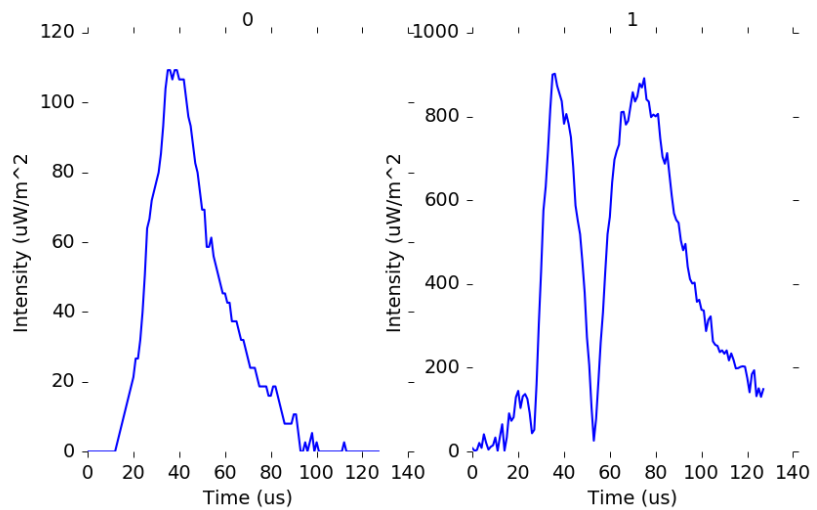


Figure 39: Noise synthesis. 1 is synthesized.

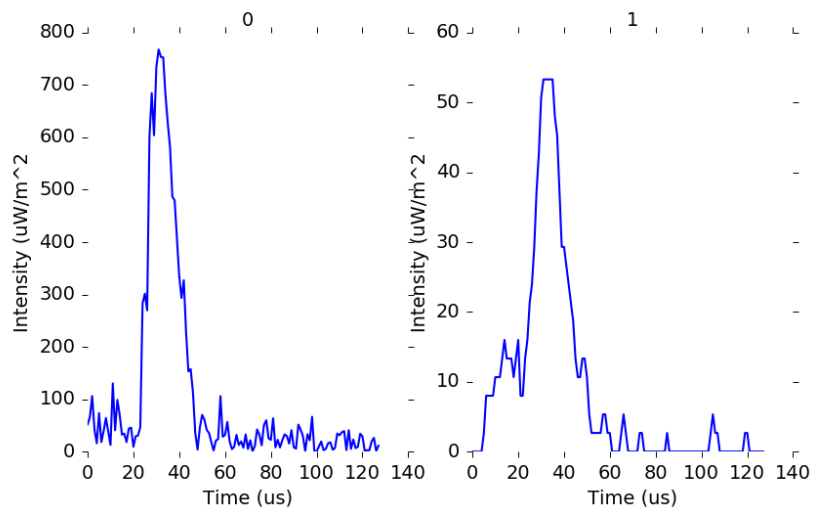


Figure 40: Noise synthesis. 0 is synthesized.

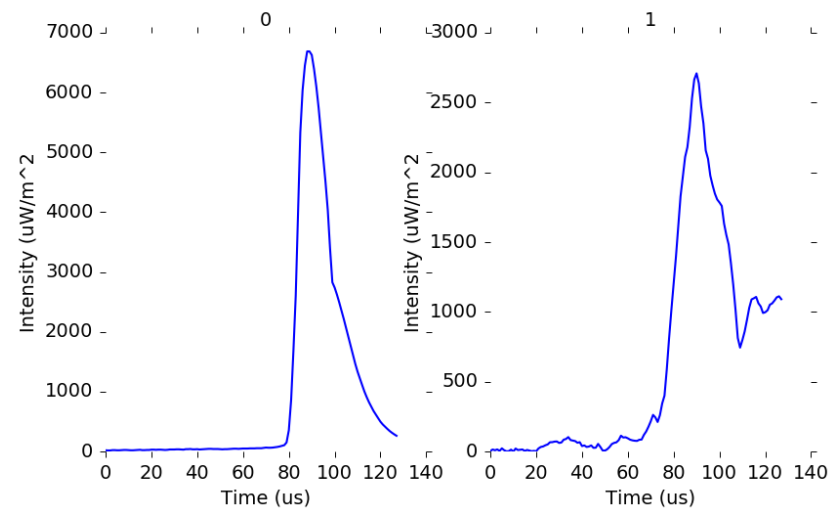


Figure 41: Noise synthesis. 1 is synthesized.

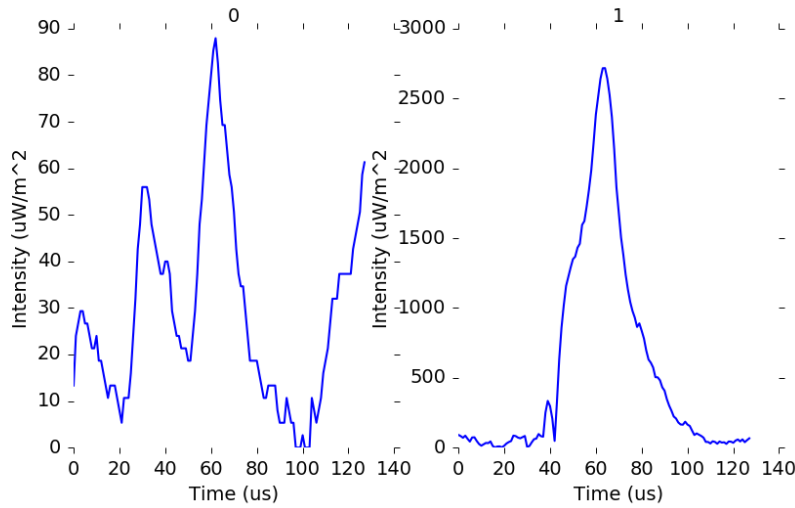


Figure 42: Noise synthesis. 1 is synthesized.

D Data

The attached zip-compressed data file includes all 18 000 signals from each decile of each group, the zoo of 1000 oddball signals, the times of each sample, each group's mean and covariance, and some statistics about the group sizes.



E Acronyms

DC	direct current
DCT	discrete cosine transform
FFT	fast Fourier transform
FORTÉ	Orbiting Experiment
GMM	Gaussian mixture model
LANL	Los Alamos National Laboratory
ML	Machine Learning
NNMF	Non-Negative Matrix Factorization
PCA	Principal Component Analysis

PDD	Photo-Diode Detector
SML	Supervised Machine Learning
SNL	Sandia National Laboratories
SNR	Signal-to-noise Ratio
UML	Unsupervised Machine Learning

References

- “Forte lightning data,” Sandia National Laboratory, Tech. Rep., ca 2003.
- W. B. T.E.L. Light, S.M. Davis and A. Jacobson, “Global nighttime lightning flash rates and characteristics observed with the forte satellite,” Los Alamos National Laboratory, Sandia National Laboratory, and Niagara University, Tech. Rep., jul 2003.
- C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer Science + Business Media, LLC, 2006.
- J. Romberg, “Structured matrix factorization,” 2016, Georgia Tech ECE 6254 Spring 2016 class notes.