# Udacity Machine-Learning Engineer Nanodegree

# Capstone Project Report

Eric G. Cavalcanti

(Dated: August 21, 2018)

# I. DEFINITION

## A. Project Overview

The aim of this project is to use machine learning techniques to estimate photon counts from the output of a transition-edge sensor (TES) detector illuminated by laser light. Transition-edge sensors provide, among other applications, unprecedented sensitivity for photon-number-resolved optical detection, with applications in quantum information and computation technologies [1, 2]. The capability of accurately and efficiently resolving photon numbers with TES detectors has the potential to open up higher-dimensional applications in optical quantum information processing.

TES detectors provide the highest efficiencies among photon-number resolving detectors [3]. One of the drawbacks of these sensors however are the slow response times and thus low detection rates. One of the motivations for this project is to explore whether machine learning techniques could provide insights on the structure and classification of TES traces, that could potentially translate into hardware-based classification of photon numbers and accelerate detection rates.

Further details on the TES detector used for this project can be found in an introductory set of notes [4], available in https://github.com/egcavalcanti/machine-learning, under `projects/capstone/photon_detector/TES_introductory_material.pdf`.

## B. Problem Statement

The data utilised in this project was produced by Geoff Gillett from the University of Queensland's Quantum Technology Laboratory. The dataset contains of a collection of output signals ("traces") produced by a TES illuminated by a laser source. As the energy of a pulse is proportional to the number of photons, photon numbers in a trace have been estimated via the area of the signal (and thus its energy). The distribution of pulse areas for each photon number value follows a slightly skewed Gaussian distribution. For low photon numbers, an estimate of photon numbers via pulse area appears to be quite accurate as the clusters are reasonably well separated. For higher photon numbers however, there is considerable overlap between the clusters, and thus an estimate by area alone seems to be insufficient to fully resolve photon counts.

The goal of the present analysis is to explore whether unsupervised learning techniques can provide an improvement on these estimates. A solution to the problem presented in this project would be a classification of the traces in the dataset into clusters corresponding to pulse photon numbers. Given that the pulse area follow approximately Gaussian distributions, an appropriate technique for this task is Gaussian Mixture Model (GMM) clustering. I will also apply Principal Component Analysis to reduce the 2048-dimensional pulse intensity vectors into a lower-dimensional space, to better visualise how features apart from the area can separate the data into photon-number clusters. I will compare the clustering produce with this technique with the area classification using the metric discussed in the following subsection.

### C. Metrics

Unfortunately, there is no independent estimate of the photon number in a particular pulse apart from the pulse area. This makes it impossible to determine the "true" number of photons detected by any given pulse. However, since the photon source produces optical coherent states, it is expected that the photon numbers follow a Poissonian distribution. For a coherent state the probability of detecting $n$ photons in a given time interval is given by:

$$P(n) = e^{-\langle n \rangle} \frac{\langle n \rangle^n}{n!} \ ,$$

where $\langle n \rangle$ is the expected number of photons in that interval. This can therefore serve as a metric to compare the technique produced as part of this work with the direct estimate via trace areas: ideally, this classification would be closer to a Poissonian distribution than the benchmark classification, indicating that it is closer to the expected photon-number distribution of the coherent laser source.

However, the GMM clustering classification was not appreciably superior to the benchmark as far as this metric is concerned. This was likely partly due to the limited size of the dataset used here, and thus low statistics for large photon numbers. On the other hand, as I will discuss in the following sections, the visualisation of the clusters based on the PCA component analysis give some evidence that the present clustering classification is more accurate; the small difference as far as the Poissonian distribution is concerned could be due to data points in neighbouring clusters swapping clusters while keeping ratios relatively

unchanged. The results in this exploratory project indicate that this analysis merits further exploration with a larger dataset.

## II. ANALYSIS

### A. Data Exploration

The TES data to be utilised in this work was produced by Geoff Gillett at the University of Queensland QT Lab, and permission has been given by that author to use the data in this work, and for its publication on Github. The directions to download the dataset are given in the README.md file found in https://github.com/egcavalcanti/machine-learning, under `projects/capstone/photon_detector`.

The file `TES_dataset.npz` contains several numpy arrays:

- `trace`: a numpy record array of $3 \times 10^5$ captured traces.
- `raw_hist`: a histogram created from $3 \times 10^7$ area measurements taken with the same optical input as the traces.
- `hist_x`: the trace area ($\sim$energy) values at the centre of the histogram bins.
- `smooth_hist`: `raw_hist` smoothed using a gaussian filter.
- `peaks`: a sequence of slices that divide the histogram data into 12 distinct peaks, based on the mid point between maxima.
- `max_i`: the indices of the first 12 peaks in the smoothed histogram.
- `std`: the std deviations of the first 12 peaks estimated from the FWHM.

The data in `raw_hist` has been used to fit Gaussians for photon numbers between 1 and 12, with standard deviations as recorded in `std` (Figure 1). These were then used to determine the slices in `peaks`, based on the mid points between maxima, which provide the benchmark area-based classification.

The traces used for the present analysis are found in `trace`. This record array contains 30022 entries with the following fields: `('size', 'tflags', 'eflags', 'time', 'area', 'pulse_length', 'pre_trigger', 'rises', 'dot_product', 'samples')`. In

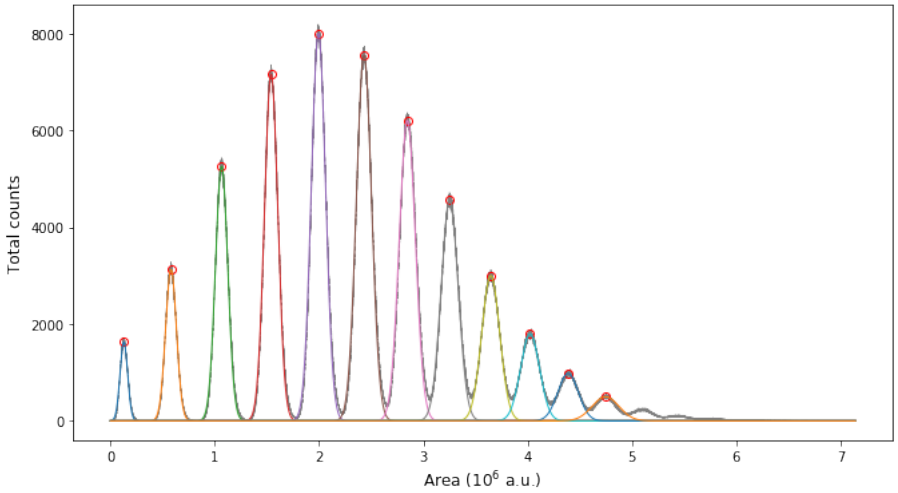


FIG. 1. A histogram of the energy measurements (area) is given in grey, and modelled as a mixture of 12 normal distributions (coloured). The data appears to be a mixture of skew-normal distributions but the Gaussian approximation is adequate to classify the traces.

this project, only the `'area'` and `'samples'` fields will be used. The data in `'samples'` consists 2048-dimensional vectors encoding the raw pulse intensity signals, triggered after the rise reaches a certain threshold. A plot of the first 5 traces is given in Figure 2.

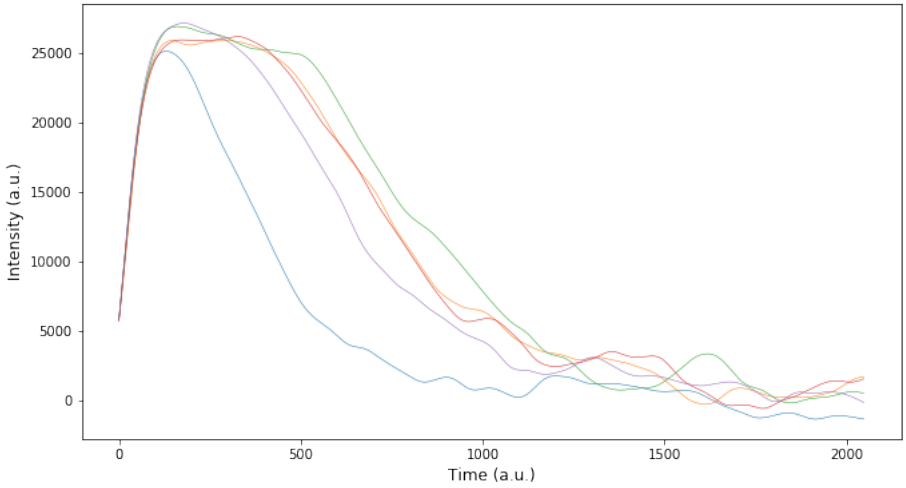


FIG. 2. A plot of the first 5 traces in `traces['samples']`. We see that the traces are characterised by a sharp rise, sometimes followed by a plateau, then a slower decline in intensity, with a noisy extended tail.

Although the other fields in `trace` could potentially be useful as features for unsupervised

learning, I will not consider them here, since one of the goals of this work is to determine whether some fast, built-in hardware processing can be used to accelerate the output of a TES, and thus none of that post-processed data would be available for this purpose.

As the histogram `raw_hist` was produced from $3 \times 10^7$ data points, while the data in `traces` contains only $3 \times 10^5$ traces, it is expected that this exploratory analysis may not necessarily produce more accurate results than the benchmark.

More details on the dataset and all the analysis in this work can be found in the notebook `photon_detector_capstone.ipynb` (also exported to `photon_detector_capstone.html`) contained in the same directory as the dataset.

### B. Exploratory Visualisation

Using the benchmark area classification, the traces dataset has been classified into 13 classes, one for each of the first 12 photon numbers, and the last corresponding to photon numbers larger than or equal to 13. A plot of the first 100 traces in each class is given in Figure 3.

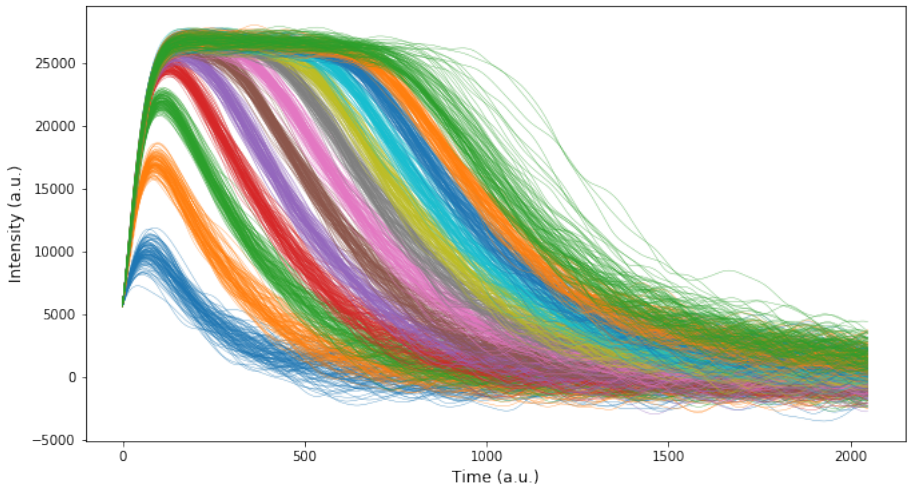


FIG. 3. A plot of the first 100 traces in each class, using the benchmark area classification on `traces['samples']`. Note that the classes for small photon numbers are well-separated, whereas the classes for larger photon numbers are overlapping.

Next we visualise the area classification applied to the traces dataset (Fig. 4). Note that the peaks are less well-defined normal distributions as compared to Fig. 1, due to the much

smaller size of the dataset used in this analysis. Also note that the area classification leads to sharp edges between the clusters for photon-number values larger than 6, due to overlap between the normal distributions of area. The goal of this work is to attempt to find a clustering that utilises more information in the pulse than is contained in the area, and thus to obtain more accurate classification of boundary cases.

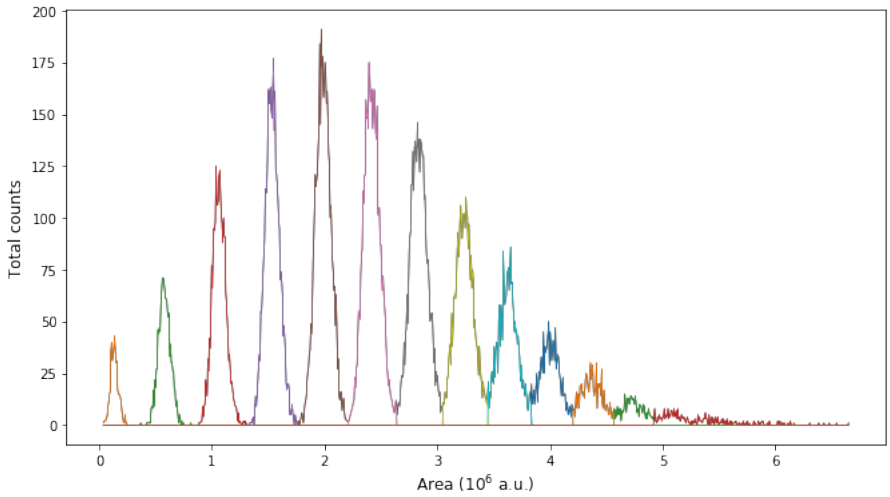


FIG. 4. A histogram of the energy measurements (area) on `traces['samples']`, with coloured lines representing each photon-number class using the benchmark area classification.

### C. Algorithms and Techniques

Due to the approximately normal distributions of areas as seen in the visualisations above, it is plausible to expect that the distributions of other features (e.g. pulse rise times, width, etc) will also follow approximately Gaussian distributions. A appropriate algorithm for this cluster classification problem is therefore Gaussian Mixture Models (GMM), which assumes that the data is composed of a mixture of normal distributions, and attempts to obtain the parameters of these distributions from the data. The following parameters of the scikit-learn implementation of GMM used in this work were tuned:

- `n_components`: the number of clusters to be modelled. These were set at 12 or 13, depending on whether we want to model the photon numbers beyond $n = 12$.
- `n_init`: The number of initialisations to run the algorithm. The best results are kept.

I have used 50 for the final results after dimensionality reduction but for the whole traces this was not feasible, and stayed at a maximum value of 3.

- `weights_init`: the initialisation value for the cluster weights. This was provided by the area classification weights for the reduced data clustering.

- `means_init`: the initialisation of the mean values for the clusters. This was obtained from the mean traces given by the area classification and significantly improved the final clustering results.

The raw 2048-dimensional vectors encoding the signal traces were reduced using the scikit-learn implementation of Principal Component Analysis (PCA). This utilises singular value decomposition of the data to project it to a lower-dimensional space with dimension given by the parameter `n_components`.

### D. Benchmark

The benchmark model for this analysis is the classification based on areas discussed above.

## III. METHODOLOGY

### A. Data Preprocessing

The data encoded in the TES dataset was first classified using an area-based classification, with boundaries given by the mid-points between peaks on the area histogram for the larger dataset of $3 \times 10^7$ traces (discussed in Section II A). This classification applied to the smaller dataset of $3 \times 10^5$ traces used here is displayed in Figure 4. I have then removed an extreme outlier with a very large value of the area from the original dataset. With this classification, I have calculated the means of the traces in each class (Fig. 5), as well as the weights of each class relative to the total number of traces in the dataset (i.e. the relative probability that the number of photons corresponding to each class is detected) (Fig. 6). These will later be used to initialise the clustering algorithm.

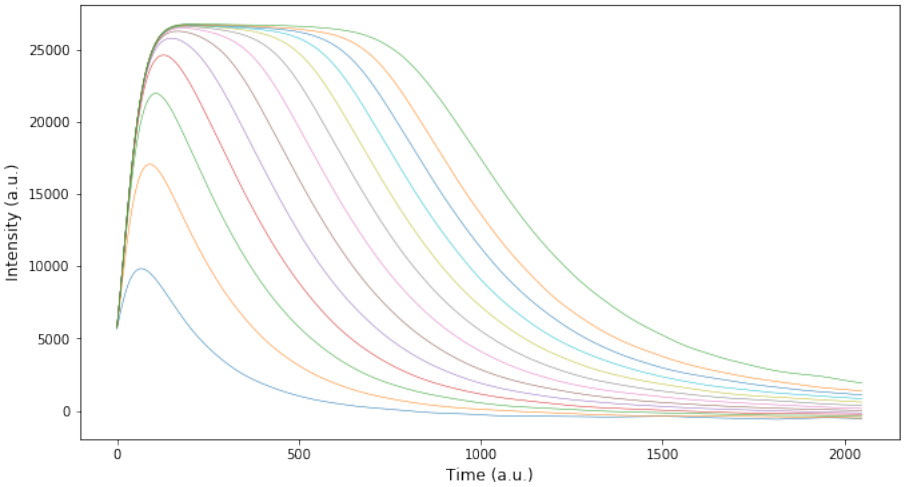


FIG. 5. A plot of the mean traces in each class, using the benchmark area classification on `traces['samples']`.

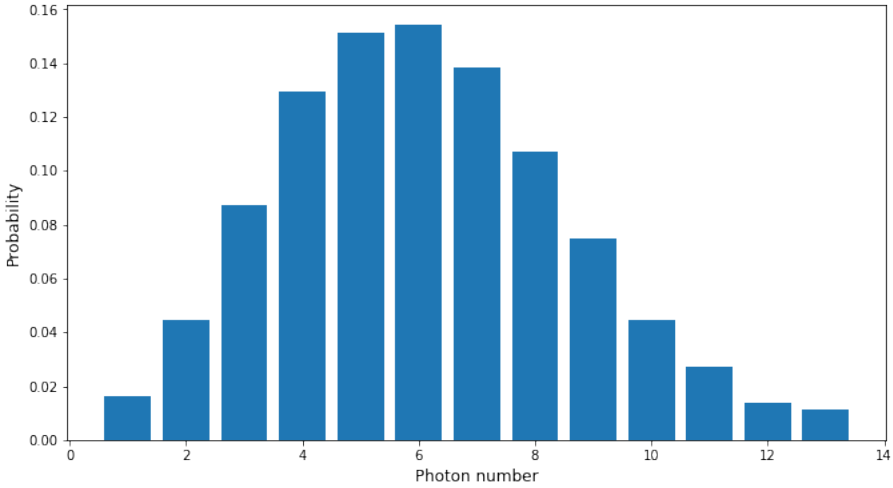


FIG. 6. A histogram of the relative probability for each photon-number class, using the benchmark area classification on `traces['samples']`.

### B. Implementation

The implementation proceeded by applying scikit-learn `GaussianMixture` unsupervised learning algorithm to the dataset in `traces['samples']`. The trained clusterer was then applied to the whole data and the cluster values for each trace were stored as an appended field in the record array, `traces['cluster']`. I initially varied the `n_components` parameter between 9 and 14, comparing their silhouette scores, to obtain some idea about how well the

algorithm performs with different numbers of clusters. However, this approach didn't seem to work well as for any value less than 12 the clusters tended to group together several of the photon-number peaks. Since we have a very good indication of roughly where the clusters are supposed to be, the idea of measuring their fitness by comparing silhouette scores seemed pointless – a visual inspection reveals already a much better comparison.

While the particular clustering achieved on a given run would vary depending on the random initialisation, a typical result is displayed in Figure 7 below. However, there are some oddities: Here we see that up to $n = 4$, the clusters are well defined, but from there on some peaks can be classified as separate clusters (as in $n = 6$) or two separate peaks are classified as a single cluster (as in $n = 10$ and 11). Clearly this classification is underperforming relative to the benchmark.

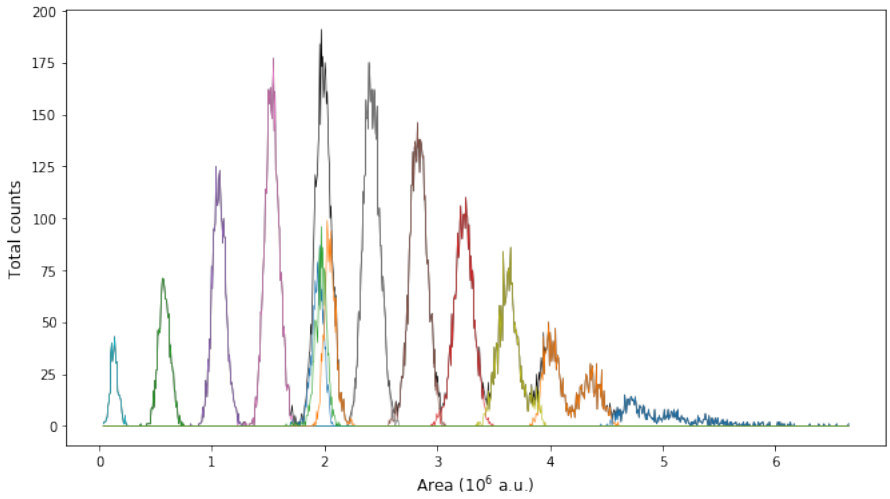


FIG. 7. A histogram of the energy measurements (area) on `traces`, with coloured lines representing each photon-number cluster obtained via GMM clustering on the raw data in `traces['samples']`.

On the other hand, this preliminary results reveals an encouraging feature: the clusters overlap, maintaining something close to a Gaussian distribution, instead of cutting off halfway between the peaks in the case where we use the area alone. This suggests that the algorithm is finding other features to detect the cluster apart from the area, and a refined version could outperform the benchmark.

### C. Refinement

Two approaches were used to refine on the results of the previous section. Firstly, a reasonable expectation was that providing the initial means and weights of each class as determined by area would improve the results, as the clusters would not tend to be initialised in boundary regions which could tend to separate single clusters or unite separate clusters. However, initialising with the full 2048-dimensional means proved to be very time-consuming and didn't on its own always provide satisfactory results. So this step was taken in conjunction with dimensionality reduction.

As each trace contains a time series of 2048 intensity values, this is a high-dimensional clustering problem, and thus the processing time would certainly benefit from dimensionality reduction. Furthermore, as it is one of the goals of this project to determine what other features apart from area are indicative of the photon numbers, dimensionality reduction was always expected to be one of the techniques to be used. I therefore applied Principal Component Analysis using scikit-learn's `PCA` algorithm to the traces dataset. An initial fit with $n_components = 20$ showed that almost 99.9% of the variance in the data was explained by the first 20 PCA dimensions. Indeed 86.8% of the variance is captured by the first dimension, which we can initially guess to be highly correlated with the pulse area. The first 3 dimensions already capture 96.9% and the first 6 capture 98.3% of the variance. Testing with different numbers of dimensions shows that $n = 6$ already provides very good cluster decomposition. This is a remarkable reduction from a vector in a 2048-dimensional space to a 6-d space.

Figure 8 shows a grid of scatter plots between the area of each pulse (as recorded in `traces['area']`) vs. each PCA dimension, and between every pair of PCA dimensions. We see from the graph between Area and PCA1 that, as expected, there is a high degree of correlation between these two parameters. There is also some correlation between area and PCA2 and PCA3, although not a linear relationship as that with PCA1. In the scatter plots between PCA1 and the other PCA dimensions, we can note the clear presence of approximately Gaussian clusters, very clearly defined for small values of PCA1 (and thus of area, and thus of photon number), but increasingly mixed as those values increase. These plots give some evidence that there should be further information in the pulses that allows for separating the clusters, apart from the area.

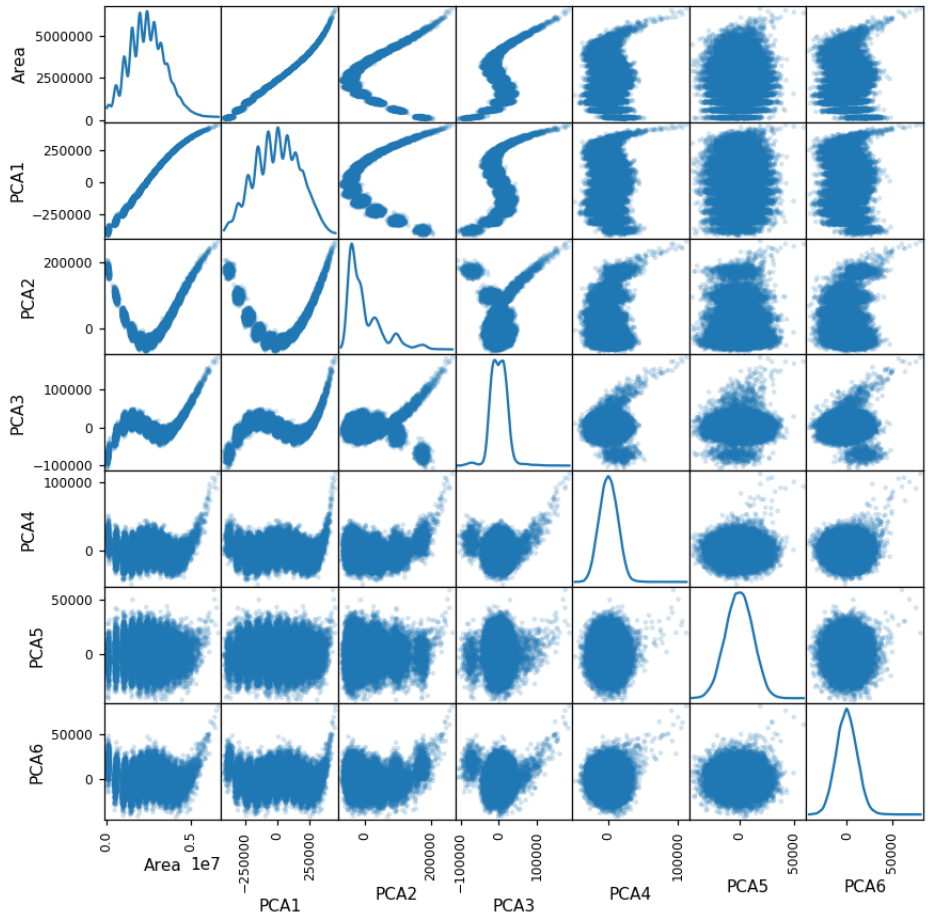


FIG. 8. A scatterplot grid between the area of a pulse and each of the PCA dimensions.

Following dimensionality reduction, the GMM algorithm was fitted to the reduced data, with the initialisation provided by the mean traces and weights calculated above. As processing time is much faster with the lower dimensional data, I was also able to perform 50 initialisations; the algorithm selects the best result out of those. Figure 9 shows a histogram of the pulse areas, with coloured lines representing the histograms for each photon-number cluster. We see that the results greatly improve on those in the previous section (Fig. 7): now all clusters up to $n = 11$ are well separated, and as before, this classification does not exhibit the sharp cut-offs that we get with the area-based benchmark classification. However, I was not able to tune the parameters of the model to separate $n = 12$ from $n = 13$. Even including a 13th set of values for the means and weights was not enough – the 13th

means value included all the data points up to the end of the spectrum, and this shifted the means towards the higher values. Perhaps a more careful initialisation with 14 clusters could do a better job.

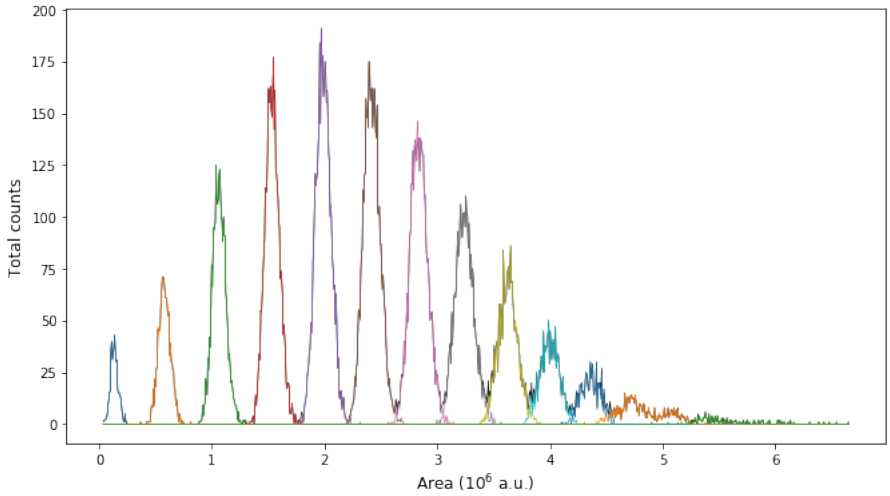


FIG. 9. A histogram of the energy measurements (area) on `traces`, with coloured lines representing each photon-number cluster obtained via GMM clustering on the PCA reduced data.

## IV. RESULTS

### A. Model Evaluation and Validation

As mentioned above, one of the features of the refined GMM clustering is that there is no longer a sharp boundary between the clusters in the area histogram, indicating that they are capturing other features of the data apart from area. This is further evidences by analysis of a 3d scatterplot of the data, coloured by clusters, using the first 3 PCA dimensions (Figure 10). Recall that the first 3 PCA dimensions capture 96.9% of the variance in the data. As we can see in the plot (and even better by manipulating the online 3d plot), there is a clear separation between the clusters up to $n = 11$, which cannot be obtained when looking at the area projection (closely related to PCA1) alone. This is strong visual evidence that the algorithm is correctly capturing the clusters and outperforming (at least up to $n = 11$) the benchmark model. It also provides a result for the other desired outcome of this project: it provides (at least) two other dimensions in the data which are highly relevant for photon-

number cluster separation, and which are not encoded in the area. This result suggests a direction for further research to determine what features of the traces these other dimensions are correlated with, and whether they could be obtained via a faster hardware-built method.

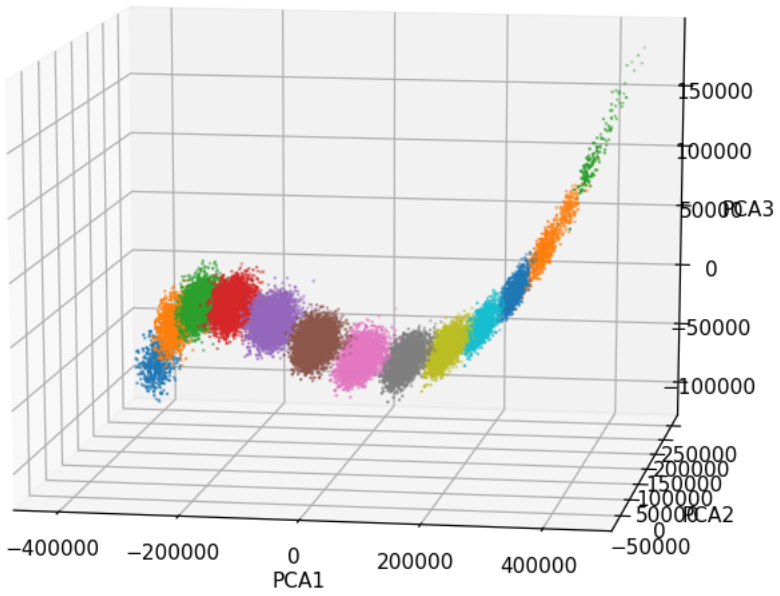


FIG. 10. A 3d scatterplot of the dataset along the first 3 PCA dimensions. The colours represent clusters obtained with the refined GMM algorithm.

### B. Justification

The accuracy of the clustering above was evaluated relative to the benchmark model by fitting to the expected Poisson distribution,

$$P(n) = e^{-\langle n \rangle} \frac{\langle n \rangle^n}{n!} \ .$$

This distribution has only one free parameter, the average number of photons, $\langle n \rangle$. As the clusters for $n = 12$ and above in the GMM clustering are clearly not reliable, I compare only the values for $n \leq 11$. The benchmark model produced a fit of $\langle n \rangle = 6.1131 \pm 0.0353$, and our refined GMM clustering produced a fit of $\langle n \rangle = 6.1082 \pm 0.0347$. Thus while the GMM model provided a slightly better fit, it was an essentially negligible improvement in the fit error. As shown in Figure 11, the best-fit Poisson curves for the benchmark and

GMM clustering are essentially indistinguishable. It is noteworthy that both distributions deviated somewhat from the Poissonian, more notably around $n = 5$ and $n = 6$. This seems to indicate that the quantum state prepared in this experiment was not very close to a coherent state as expected.

On one hand, this is a successful result in that the clustering algorithm produced results that are at least comparable to the benchmarks results via this metric. On the other hand, this shows that any potential advantage cannot be ascertained via proximity to the Poisson distribution alone. The 3d visualisation above gives evidence that the GMM clustering is more accurate. That this doesn't translate to a better fit to a Poissonian distribution could be explained by two factors: firstly the relative weight of the mis-classified data points are clearly quite small to start with, as we see from Fig. 4. Secondly, the improvement that we seem to obtain via the GMM clustering will mean that neighbouring clusters will swap data points, but keep their relative weights essentially unchanged. Thus the distribution will remain just as close to a Poissonian. Some independent method to ascertain the photon numbers could lead to a better estimate of the gain in accuracy, but this cannot be obtained with the experimental setup used, and is experimentally very challenging.

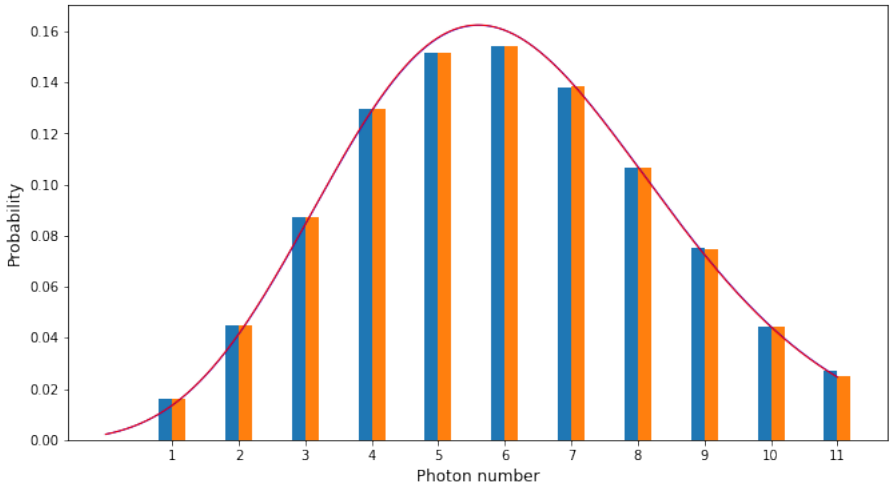


FIG. 11. A histogram of the photon number probabilities provided by the benchmark area classification (blue bars) and the refined GMM clustering algorithm (red bars). The red line represents the best-fit Poisson distribution for the GMM data, and coincides with a blue line representing the best fit for the benchmark data.

## V. CONCLUSION

### A. Free-form Visualisation

We now visualise the results by adding the cluster obtained via the refined GMM algorithm on the PCA data (represented as Cluster_PCA) to the scatterplot grid versus the area and each of the 6 PCA dimensions. This is shown in Figure 12. Here we again see that while the area is highly correlated with the cluster numbers, it is insufficient to separate between all neighbouring clusters, and that further information separating the clusters is obtained by considering the second and third PCA dimensions, and to a smaller extent the final three.

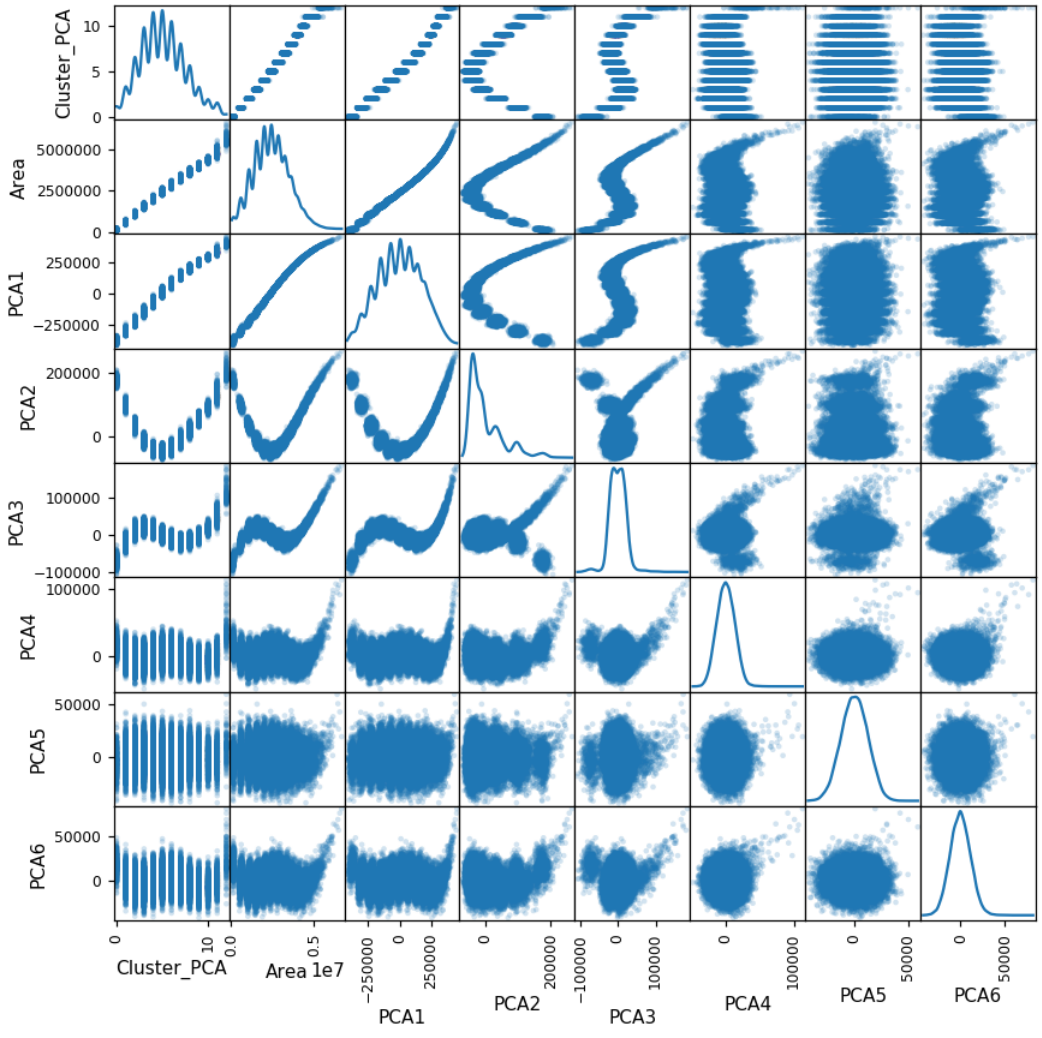


FIG. 12. A scatterplot grid between each pair of: the refined PCA cluster, the pulse area, and each of the 6 PCA dimensions.

### B. Reflection

End-to-end, this project can be summarised as follows:

- Search for an interesting and suitable problem. I was particularly interested in a project related to my area of expertise, quantum physics, and talking to a colleague who shares similar interests and has some knowledge of machine learning (Sally Shrapnel) this problem was suggested. Although this is not directly related to my theoretical research, it sounded like and interesting and potentially useful project to pursue.

- The next step was obtaining the dataset. This proved to be quite a lengthy process as the experiment had a much larger dataset than the one used here ( 120GB) on a cloud computing server, and obtaining the relevant permissions to access that data took some time and bureaucracy. I ended up opting to use a smaller but representative dataset provided by Sally with permission from the data owner, Geoff Gillett.

- The initial data exploration was rather lengthy as well, as the first dataset I obtained did not contain any comments on the meaning of the various fields saved in the .npz file, or on the background for how the data was obtained. It took some time for Geoff to finish a set of explanatory notes, and a Jupyter notebook with a brief explanation of the dataset. Having got this information greatly accelerated the process.

- From there on the process flowed a bit more steadily. Obtaining some visualisations and an initial clustering result was relatively straightforward, after some tinkering with the data. The initial clustering results were however clearly unsatisfactory.

- After some thought and looking up documentation, I found that providing good initialisations should improve the results, and implementing dimensionality reduction would speed up the results, as well as provide part of the information that I was seeking for this project. These implementations were somewhat more complicated than the initial ones, but have given rather satisfactory results.

- Due to the lack of an independent metric and the relative insensitivity of the Poissonian distribution to distinguish between the two methods, finding good visualisations for the results, in particular the 3d scatterplot, proved crucial to obtain information to evaluate the accuracy of the results.

### C. Improvement

It seems clear that using the larger dataset available for this experiment will improve the results, with much better statistics. The running time may become an issue – certainly before the PCA reduction, but perhaps not after that. It will be interesting to see how that affects the final clustering. It would also be interesting to attempt to analyse what the PCA dimensions 2 and 3 correlate with on the pulse traces: is it the rise time, or perhaps the pulse widths? It would also be interesting to attempt to explain the discrepancy between both the area classification and the GMM clustering relative to the expected Poisson distribution. Finally, it would be interesting to obtain data where prior information about the photon numbers in a pulse is given. This however is extremely challenging, and unlikely to be feasible with current technology for large photon numbers.

---


[1] D. H. Smith, G. Gillett, M. P. de Almeida, C. Branciard, A. Fedrizzi, T. J. Weinhold, A. Lita, B. Calkins, T. Gerrits, H. M. Wiseman, S. W. Nam, and A. G. White, Nature Communications **3**, 625 (2012), arXiv:1111.0829.

[2] B. G. Christensen, K. T. McCusker, J. B. Altepeter, B. Calkins, T. Gerrits, A. E. Lita, A. Miller, L. K. Shalm, Y. Zhang, S. W. Nam, N. Brunner, C. C. W. Lim, N. Gisin, and P. G. Kwiat, Physical Review Letters **111**, 130406 (2013), arXiv:1306.5772.

[3] M. D. Eisaman, J. Fan, A. Migdall, and S. V. Polyakov, Review of Scientific Instruments **82**, 071101 (2011), https://doi.org/10.1063/1.3610677.

[4] G. Gillett, *TES Introductory material* (2018).