BACHELOR RESEARCH PROJECT

Improving HPGe Detector Time Resolution Using Machine Learning

Author:
Martin N. Hagen
S4907957

First Supervisor:
Prof. Dr. Myroslav Kavatsyuk
Second Supervisor:
Prof. Dr. Nasser Kalantar-Nayestanaki

Faculty of Science and Engineering
University of Groningen
Groningen, the Netherlands
April-June 2024

Abstract

In this thesis, machine learning is applied to HPGe detector waveforms to cluster them with the goal of determining correlations that serve to improve detector time resolution. HPGe detectors are used in neutron cross section determination experiments; improvement of the moderate time resolution HPGe detectors possess is salient due to the neutron time-of-flight being closely linked to its energy, and in turn the reaction cross section. Close determination of neutron cross sections is important, as they play roles in nuclear reactor physics and design, astrophysics, radiation shielding and more. To investigate these correlations, the clustering algorithms KMeans and Hierarchical Clustering Algorithm (HCA) were applied to a version of the dataset reduced through principal component analysis. It was found that there was a strong correlation between the cluster index, corresponding to mean value of the first principal component $PC1$, and the time difference compared to a reference lanthanum bromide detector dT. Methods to apply corrections using clustering, as well as principal components analysis directly, are suggested and discussed.

Contents

1 Introduction

The nuclear cross section is a measure of the probability that a nuclear reaction will occur when two particles collide. The cross section is dependent on which two bodies collide and the center of mass energy of the collision. One of the processes that can occur when a neutron is incident on an atom involves the emission of one or more neutrons; these reactions are noted as (n, xn) reactions with x being the number of neutrons emitted following the interaction. The close determination of neutron cross sections is crucial in fields like nuclear astrophysics and radiation shielding, but (n, xn) reactions are especially salient in nuclear reactor physics and design; this is due to these reactions having relatively large cross sections, and by extension having a large influence on the neutron density in the reactor [1]. There is a distinct lack of experimental data on these reactions, however, which can be largely attributed to the difficulties related to these neutron measurements, including subpar neutron detection efficiency and the prevalence of other neutron emission processes. These difficulties can be circumvented by employing prompt γ-ray spectroscopy at neutron time-of-flight facilities [2]. This involves considering the subset of events that leaves the daughter nucleus excited, as gamma radiation is emitted upon de-excitation. By using this method, cross sections of only (n, xnγ) are acquired and can be used to deduce total inelastic (n, xn) cross sections.

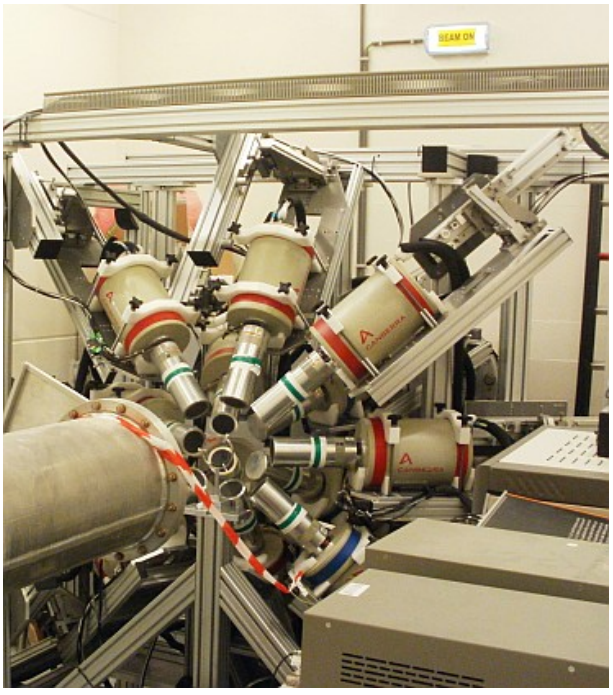


Figure 1: The GAINS experimental setup at GELINA in Geel, Belgium. [3]

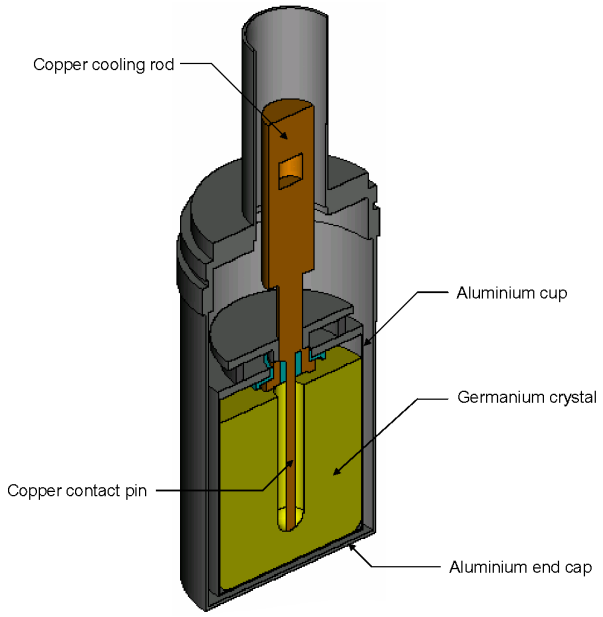


Figure 2: Cross section of an HPGe detector model, facing downwards. [4]

One of the premier neutron time-of-flight facilities employed for neutron cross section determination is GELINA, a linear accelerator located in Geel, Belgium. One of the experimental setups at this complex named GAINS (Gamma Array for Inelastic Neutron Scattering), shown in figure 1, is positioned at a 200 meter flight path of the white neutron beam source. The beam is incident on a sample surrounded by eight HPGe (high purity germanium) detectors, each detector being assigned a channel. These detectors are employed for their excellent energy resolution, but they provide only moderate time resolution compared to other ionizing radiation detectors. As seen in figure 2, HPGe detectors are coaxial and maintained under a high voltage under reverse bias. This voltage difference creates an electric field extending across the intrinsic or depleted region. When incident ionizing radiation interacts with the depleted volume material of a detector, electrons and corresponding holes are produced and swept by the electric field to the p and n electrodes [5]. The number of charge carriers produced is proportional to the energy deposited by the incident photon in the detector, and is converted to a voltage pulse that is outputted.

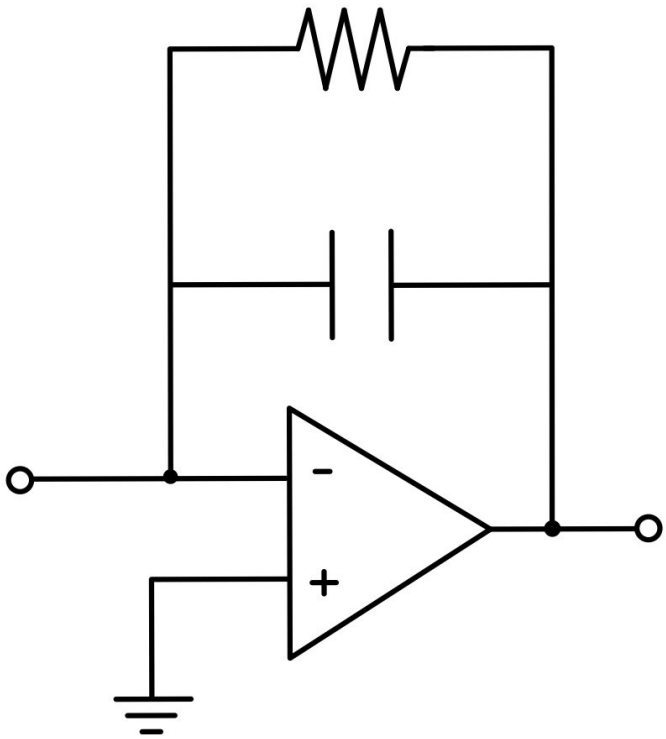


Figure 3: a diagram of a simple charge preamplifier circuit. A resistor is added over the capacitor, introducing exponential decay.

2 Methodology

2.1 Data preprocessing and resulting dataset

The signal data retrieved from the HPGe detectors underwent significant preprocessing before being usable for pursuing the aims of this thesis. The direct output of the HPGe detectors is a measure of charge deposited per time interval. The measure can be integrated to get the voltage per time interval. Integration is necessary due to the low number of electrons created in the germanium; only around ten thousand free electrons are created upon a single ionization event, meaning very little charge is measured. To process the charge measurement, one can either apply a voltage amplifier and integrate the result or simply apply a charge preamplifier circuit. The former approach introduces significant

noise in the integration stage compared to the latter. A charge preamplifier circuit, seen in figure 3, was chosen after consideration of the requirements of the experimental setup. If no method is implemented to reduce the resulting voltage signal amplitude, it would keep increasing indefinitely. In order to discharge the capacitor, a resistor was added above the capacitor which introduced exponential decay. A lower resistance of the inserted resistor would lead to a quicker decay rate of the signal but would introduce more noise, with the inverse being true for a higher resistance; the chosen resistance was decided keeping these consequences in mind. The decay of signal strength means that the recorded maximum amplitude recorded is lower than the true maximum, as some of the voltage will already have decayed. Amplitude determination was therefore performed with the use of a trapezoid algorithm to reshape the signal. Optimal shaping balances signal-to-noise ratio, pulse pile-up, and ballistic deficit. Trapezoidal shaping, which converts the exponentially decaying preamplifier signal into a trapezoidal form through convolution with time-dependent functions, offers a good compromise. The algorithm involves correcting the preamplifier signal for finite decay time to create a step function, then differentiating this to produce the trapezoid shape [6]. Following these procedures, the output of the detectors was processed to produce voltage signals suitable for further analysis.

The analogue HPGe output signals were all processed using a digitizer with a 14 bit resolution and 250 MHz sampling rate. Pulses, corresponding to gamma-detection are selected by applying a threshold to continuous data coming from the digitizer. If the threshold program observes a signal, a number of points before and after the threshold crossing is saved adding up to 650 points. The timestamp of the trigger is also saved. These threshold crossings are highly sensitive to noise, which both requires correction of the trigger timestamp and explicit noise suppression without suppressing waveforms. The latter was performed through isolating the first section of each waveform and performing linear regression to describe it with a 3rd order polynomial. The first 20% of the pulse proved adequate for all waveforms and was used to obtain precise timing of the pulse in the waveform.

At this stage, a dataset generated by a ^{22}Na source was utilized to locate coincidences, defined to be events occurring within 100ns of each other. A new coincidence-only dataset was produced that was significantly smaller than the original. Each waveform was normalized to amplitude equal to 1 and truncated such that each waveform has been reduced from 650 to 128 data points. A selection of these waveforms can be seen in figure 4. Truncation led to some waveforms not reaching their original peak amplitude in the new dataset. The noise level is approximately constant for all the waveforms, but constitutes

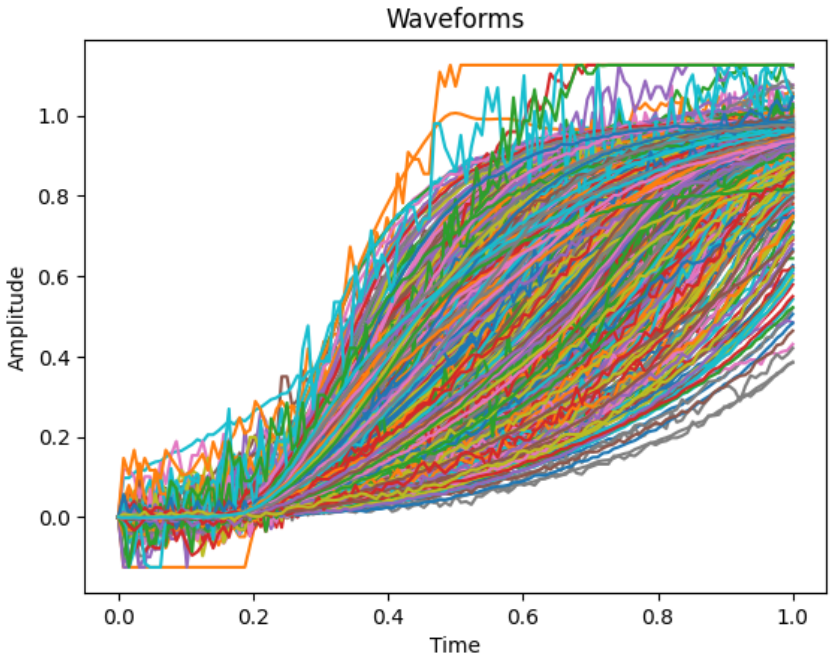


Figure 4: Plot of the first 750 waveforms in the initial dataset used. Time is in arbitrary units and the amplitude has been normalized.

a greater proportion of the maximum amplitude for waveforms with lower maxima; this is apparent in the plot, where some waveforms are noticeably noisier than the mean. The waveforms are also aligned with approximately proper timestamp, situated within 20 ns from the pulse start. This corresponds to about 0.2 in the arbitrary time units used. In order to ensure that the threshold is not dependent on the energy of the incoming photon, constant fraction discrimination was utilized. Constant fraction discrimination can determine amplitude-independent pulse timings. It accomplishes this by dividing the signal, in this case the charge output of the detector, into two, constituting 1 and 0.35 respectively. The former is delayed, here by 40 ns, and the latter is inverted ($f(x) \rightarrow -f(x)$) before the two are recombined. The resulting zero-crossing point is independent of the amplitude of the input signal [7]. By applying the procedure explained in this section, datasets suitable for applying machine learning could be produced with a number of waveforms, each possessing 128 amplitude points, the energy of the incident photon E, the time difference compared to the reference detector dT, and the channel through which the particular waveform was detected.

2.2 Clustering algorithms

Clustering analysis is an machine learning method in which the task is grouping objects in such a way that objects within a group, called clusters, are more similar to each other than to objects not in the same group. It is a form of unsupervised learning, meaning the data used is unlabeled; it in this way different from classification which utilizes labeled data and hence has more of an understanding of what the output groups should be. While cluster and class are non-equivalent, corresponding to clustering and classification analysis respectively, once clustering analysis has been applied to a dataset and clusters have been determined, further analysis is functionally identical on both. For this reason, the terms will be used interchangeably throughout this thesis. Most clustering algorithms require a cluster number input in order to perform analysis on a dataset. Several methods are available that aim to determine the optimal number of clusters; however as the the pulse shapes are expected to be a continuous distribution by considering the detector geometry, the choice of cluster number is arbitrary and should be based on the requirements of the application. Throughout this thesis, three clusters will be pictured and discussed as this number best showcased the found results, but the results can easily be generalized for any number of clusters.

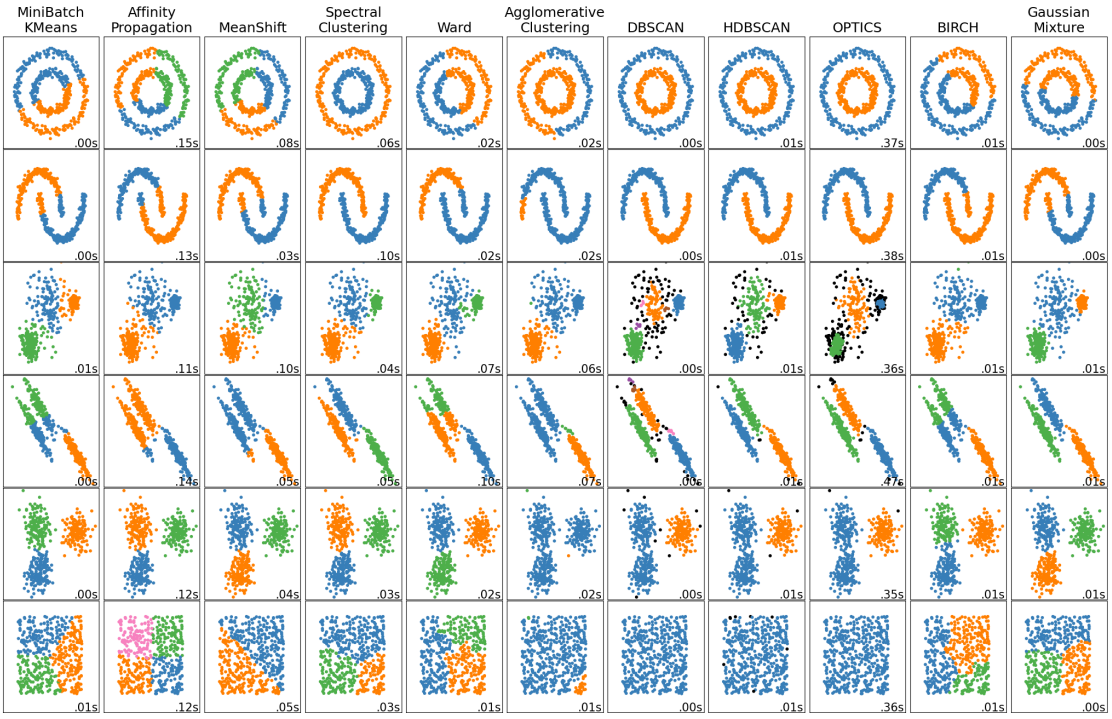


Figure 5: An overview of how different clustering algorithms perform on different datasets. The time taken to produce the results is shown in the lower-right corner of each plot. [8]

Within clustering analysis, there are a myriad of algorithms that vary greatly in working principle, implementation and time complexity. For this project, two algorithms were chosen, namely KMeans and hierarchical clustering analysis which will be referred to as hierarchy. Two algorithms were chosen to create an obvious sanity check; if the two algorithms produced vastly different results, that would either reveal an error in implementation or a difference in methodology between the chosen algorithms producing distinct clusters. KMeans was selected for this application due to it a being simple, efficient and effective algorithm. KMeans does not possess a method to assess the optimal number of clusters in a dataset and instead takes the cluster number n as an input. KMeans functions by placing n centroids in the parameter space according to initial conditions, and determines for each data point which centroid it is closest to and assigns it to that centroid. The centroid positions are then moved to the average position of all data points assigned to it; these two steps are iterated until a steady state is reached. This approach is highly efficient and scales incredibly well, with KMeans having a time complexity of $\mathcal{O}(n)$. Some drawbacks include centroid-based clustering approaches being highly sensitive to outliers and initial conditions, but these can be mitigated with proper outlier removal procedures. Another limitation is the inability of KMeans to detect more complex patterns in the data, illustrated in figure 5. The choice of second algorithm was made with this in mind, valuing complexity detection and transparency of clustering process over efficiency. Hierarchy, which aptly attempts to build a hierarchy of clusters and produces an easily digestible dendrogram was therefore a logical contender. This algorithm iteratively merges data points or clusters based on a dissimilarity measure. This measure takes the euclidean distance between data points in parameter space and a linkage criterion, wherein the dissimilarity between sets is determined by the pairwise distances among observations within those sets, as inputs [9]. Hierarchical clustering is much more computationally intensive than KMeans, a fact reflected in the poor time complexity of $\mathcal{O}(n^2 \log n)$ at best and $\mathcal{O}(n^3)$ at worst. Based on the continuous nature of the input data, however, KMeans is expected to be highly applicable; as Hierachy therefore primarily serves a comparative role, it was chosen despite its drawbacks.

The implementation of a method to quantitatively compare the outputs of the chosen algorithms was considered. Ensemble clustering is a deterministic method of producing an accurate consensus result from a collection of base clustering results and would fit this need [10]. Its incorporation was ultimately decided against, however. This is due to the approach being computationally expensive, relatively difficult to implement and ultimately not needed as the aim would be to eliminate the need for one of the clustering

algorithms. An alternative and much simpler approach was pursued and is discussed in section 3.4.

3 Clustering application

A prerequisite in this process for effectively applying clustering analysis is reducing the size of the dataset. Consideration of the form of the dataset, clear correlation between the amplitude points is seen. Correlations include early points having values close to zero, late points having values close to one and the values of intermediate points being somewhere in between. Points in close proximity also tend to have very similar values. Realizing this, a reduction of the number of parameters is expected to be feasible and was attempted.

3.1 Outlier removal

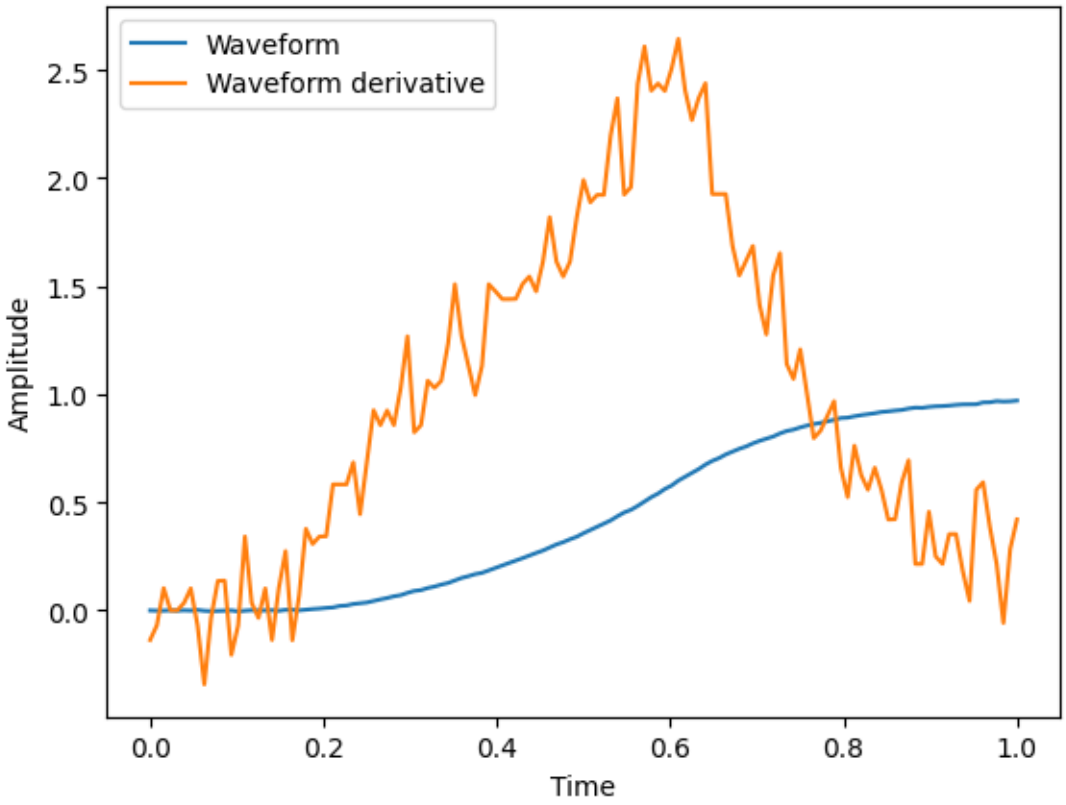


Figure 6: A single waveform from the original dataset plotted with its derivative.

Before any dimensionality reduction approach can be applied, an outlier removal algorithm must be implemented. This is due to clustering algorithms, especially KMeans, being highly sensitive to outliers and noise. At this stage, there is not enough information in the dataset to properly filter the waveforms, and a relatively simplistic approach was therefore taken. This approach involved taking the derivative of each waveform in the dataset. A waveform plotted alongside its derivative is displayed in figure 6. If the derivative of the waveform at any points exceeds a certain threshold, this waveform was removed from the dataset. This filters out low-amplitude waveforms due to the relative noise levels being much more significant, which leads to larger amplitude jumps and thus higher derivatives. Using this method, only around 4% of the dataset was removed for the determined threshold, but the remaining dataset was considerably less noisy. By removing the outliers, both clustering algorithms employed, especially KMeans, should produce significantly better results.

3.2 Quantization of time and amplitude

The initial approach for dataset size reduction involved quantizing the amplitude and time. To do this, the maximum amplitude of each waveform was determined, followed by the determination of 10 through 90 percent of the maximum in increments of 10. The amplitude values in the dataset closest to these incremental percentages, along with the time values corresponding to these data points, were then saved and used to create a new dataset. These amplitudes and corresponding time were labeled Ax and Tx respectively with x being the increment used. The resulting data points, compared to the original waveform, can be seen in figure 7 This quantization process successfully reduced the 128 parameters to only 18, which should greatly reduce the computational resources required to apply clustering analysis. Two additional parameters named $rise20$ and $rise90$ were also including representing the time difference between 20 and 10 and 90 and 10 respectively. A correlation matrix was created for this new dataset and can be seen figure 8. This was achieved through the use of the pandas *corr* function which computes pairwise correlation of columns. The *corr* function method was chosen to be Spearman rank correlation as the conditions were not expected to be met for Pearson standard correlation coefficient. This method was iterated for each combination of columns and plotted in a matrix form. As this matrix is primarily for correlation visualization purposes, the absolute value of the correlations were taken to improve clarity. Many of the correlations displayed in the figure are expected; the time at which the signal reaches 40% will always

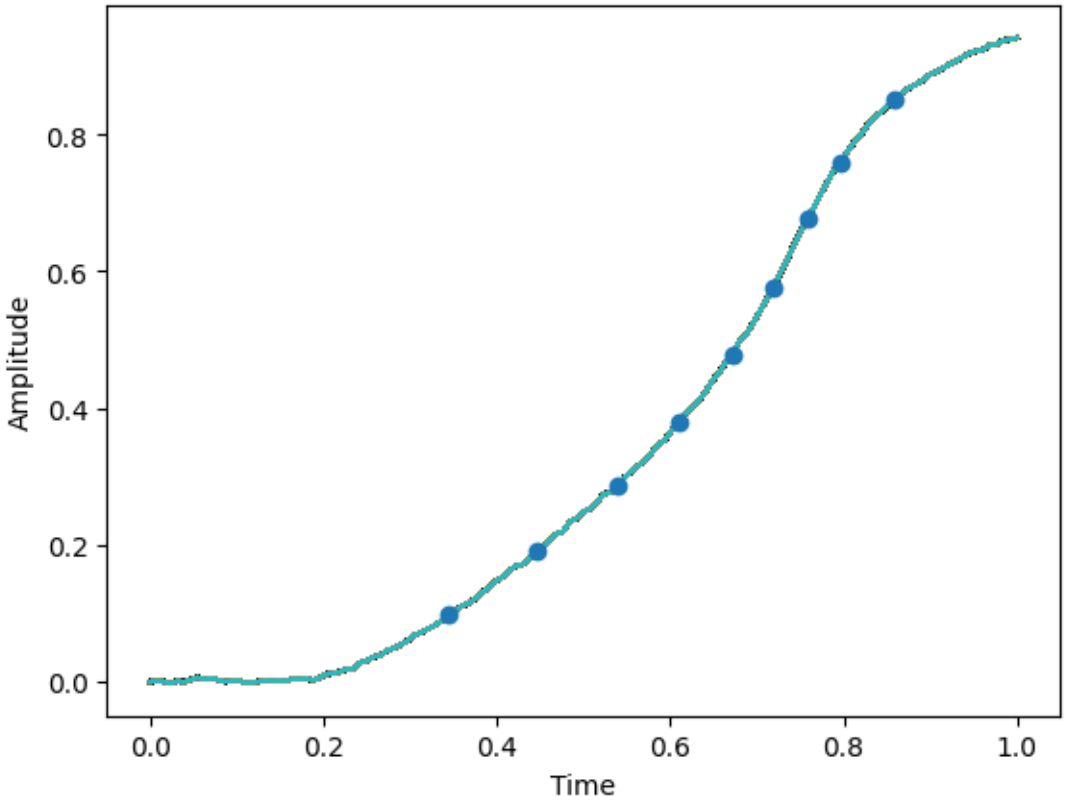


Figure 7: Example of a single waveform from the original dataset plotted alongside the data points to which it was reduced using the explained method.

be some time after it reaches 30%, for example. Most other correlations also have intuitive values, and the found parameters were therefore deemed promising enough to apply clustering analysis to.

Despite the promising correlation matrix and intuitive newfound parameters, the chosen clustering algorithms applied to the reduced dataset proved unsuccessful. The results of this procedure are shown in figure 9 and show that both algorithms produce seemingly nonsensical classes. While the clusters could have some underlying significance not readily apparent to qualitative inspection, no attempt at reaching meaningful conclusions using these clusters yielded usable results. Outliers and noise, poor choice of algorithms and dimensionality reduction could all contribute to the relatively poor results. Noise and outliers should be heavily suppressed due to the removal algorithm implemented and the clustering algorithms are proper, so these are likely not the cause of the issues observed. As the shortcomings of this approach were expected to be due to the dimensionality reduction, a different method must be employed.

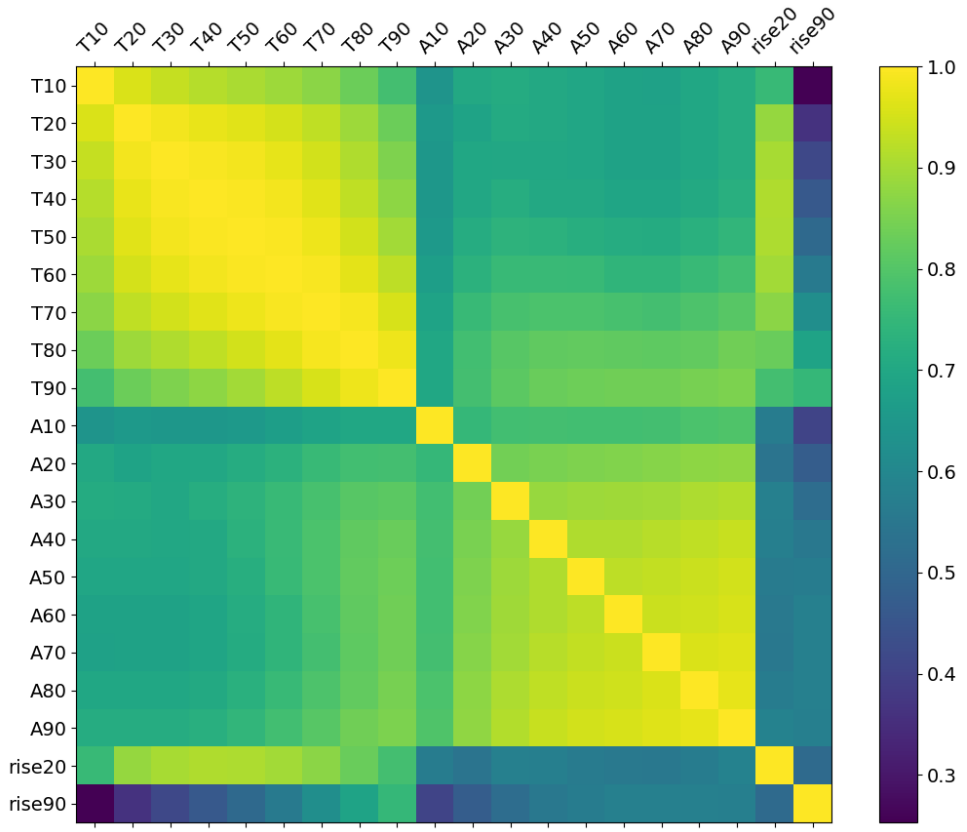


Figure 8: Correlation matrix of the parameters in the new dataset.

3.3 Principal component analysis

Principal component analysis, often shortened to PCA, is a dimensionality reduction method that relies on linear transformation of parameters followed by projection onto a parameter subspace. The determined principal components constitute an orthonormal basis in which distinct dimensions of the data are linearly uncorrelated. The numbering of the basis vectors, called principal components, reflects the amount of variance they capture such that the first principal component captures the most variance. To illustrate the working principle of PCA, the dataset in figure 10 was generated. The dataset plotted consists of two parameters shown on the x- and y-axis. By inspection, one can realize that a majority of the variance is along an axis at an approximately 45° angle to the x-axis. By choosing an orthonormal basis where one of the vectors is along this direction of most variance, shown in the figure as the red arrows, the basis will be optimized such that the greatest amount of information in the original dataset is captured in the fewest number of coordinates. This approach can be generalized for any number of input parameters and was chosen as the new dimensionality reduction procedure.

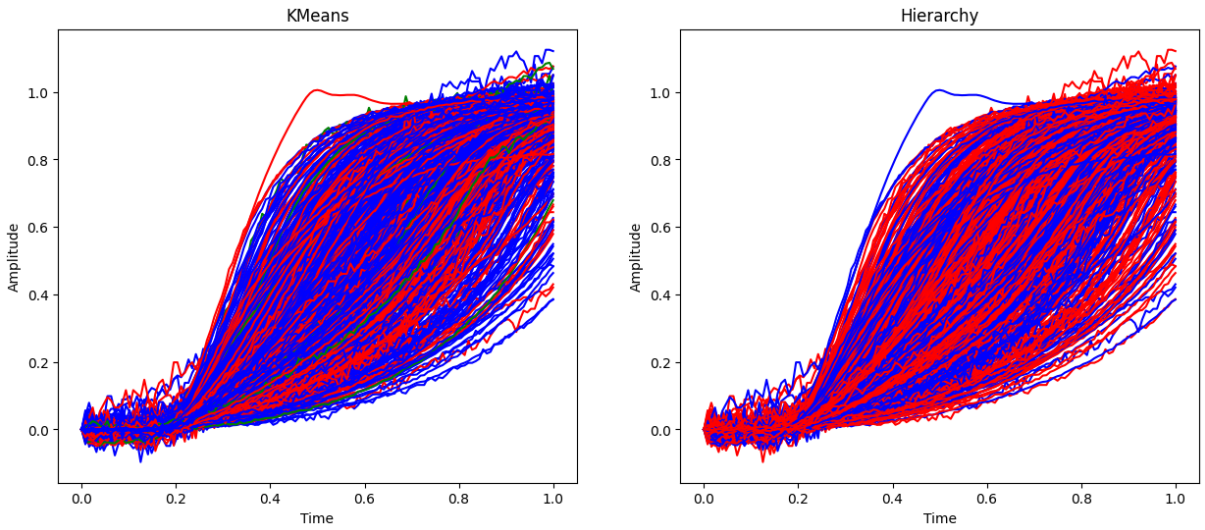


Figure 9: Plots of a selection of waveforms. Each wave is plotted in a color corresponding to the class into which it was clustered. The respective plot titles reflect which algorithm was employed.

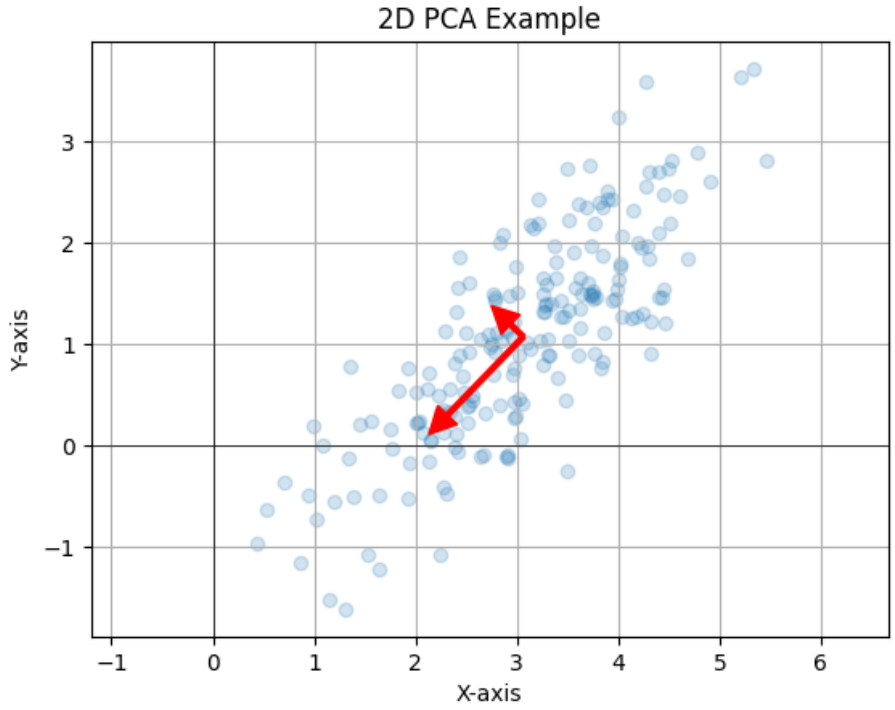


Figure 10: Example dataset generated to illustrate the working principle of principle component analysis.

Principle component analysis was applied to the original waveform dataset. The results are shown in figure 11. In figure 11(a), the explained variance ratio is plotted against the

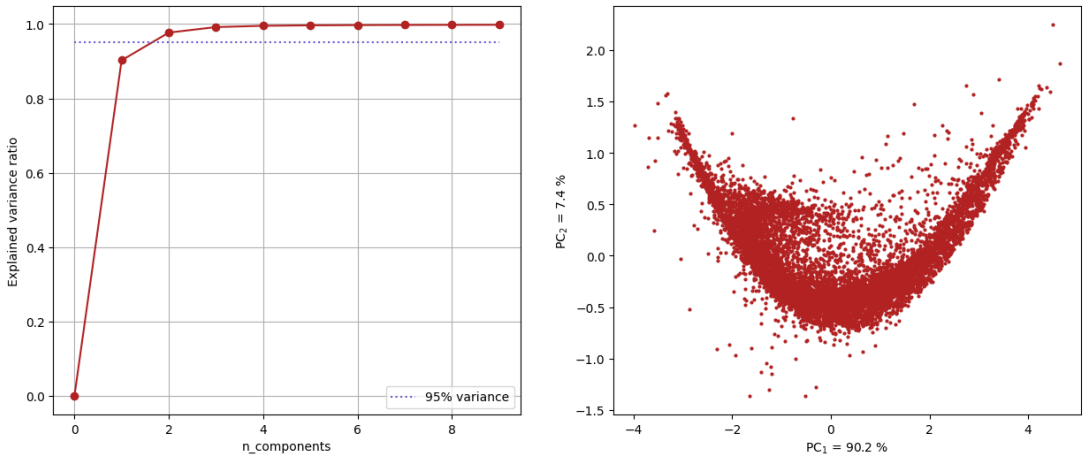


Figure 11: Plots of PCA results. (a) fraction of information in the original dataset captured in principal components as a function of number of principal components used. (b) scatter plot of the first and second principal coordinates for all waveforms.

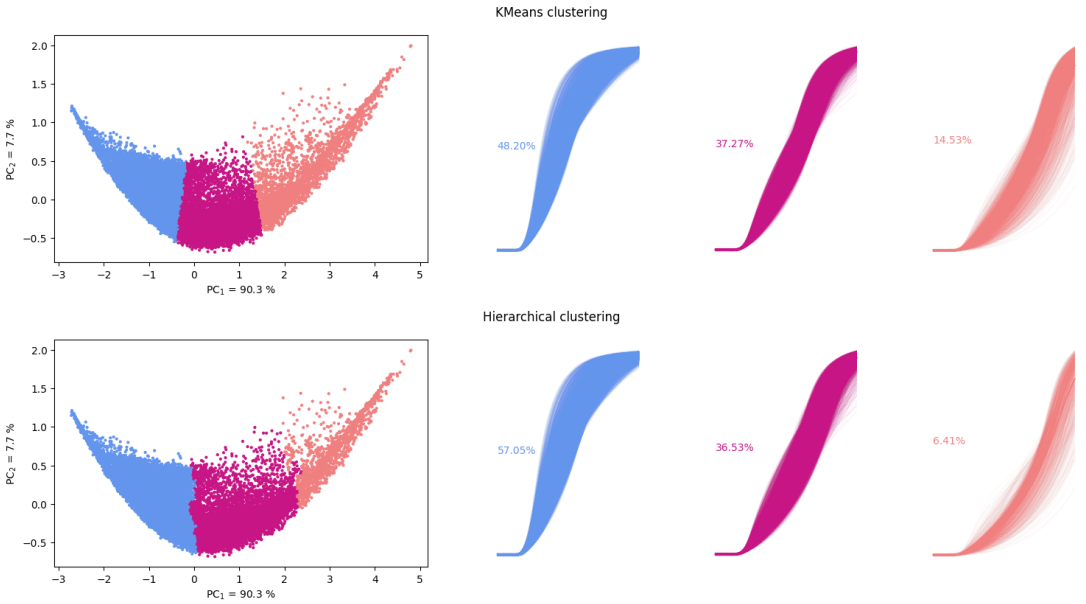


Figure 12: Results of applying the clustering algorithms to the principal components. On the right is the same PC scatter plot shown earlier, but with the points shown in the color corresponding to their class. On the right, composites of all the waveforms in each class is shown, along with the percentage of waveforms in the dataset found to be in that class.

number of components. Number of components refers to how many of the components containing the most variance are kept. The explained variance ratio is defined as the fraction of the information in the original dataset that is captured by a given number of principal components. also plotted is a dotted line marking 95% variance. This was chosen

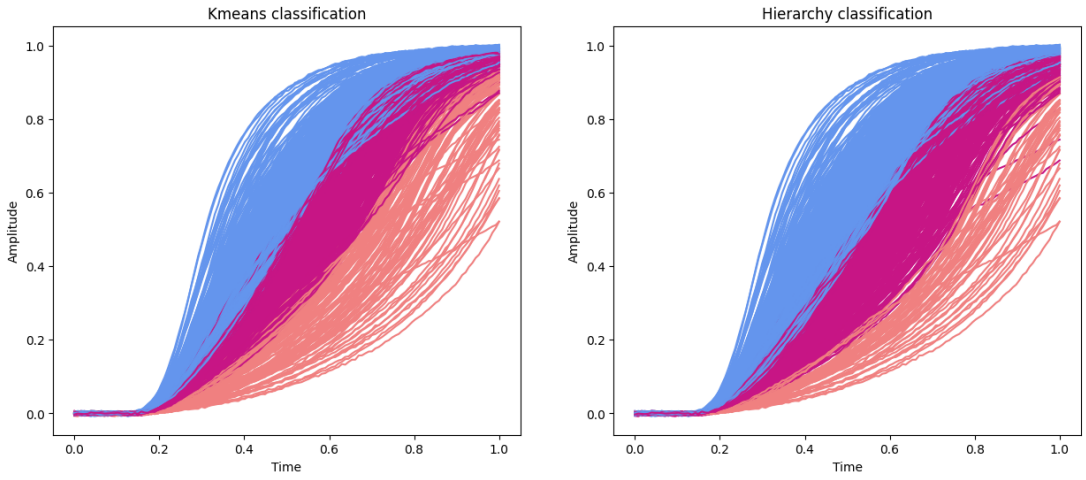


Figure 13: Plots of a selection of waveforms. Each wave is plotted in a color corresponding to the class into which it was clustered. The respective plot titles reflect which algorithm was employed.

as the threshold, such that the number of components used will be the lowest number containing more than 95% of the variance. As seen in the figure, two components were adequate to surpass this threshold. Throughout this project, many datasets were used; for all of them, two principal components surpassed the chosen threshold. Using this method, the 128 amplitude points in the original dataset were reduced to two principal components labelled $PC1$ and $PC2$. The information captured by each is illustrated in figure 14. The $PC1$ plot consists of a single peak in the middle. The $PC2$ plot has two peaks with one on either side of the middle. The first principal component therefore captures the variance at the approximate center of the waveforms, while the second component captures the variance closer to the start and end points. The third and fourth principal components are shown in the second row of the figure. This figure clearly illustrates the increasing frequency of the components, with the first component having a single peak and each subsequent component having one more extremum than the last. This suggests that subsequent components capture higher order harmonics, but as they contain little of the overall variance, these phenomena do not impact the overall waveforms significantly. In figure 11(b), a scatter plot of the first and second principal components for all waveforms is shown. In the axis titles, the percentage of variance captured by each coordinate is displayed. The significant dimensionality reduction achieved through the use of principal component analysis should enable more efficient application of clustering algorithms.

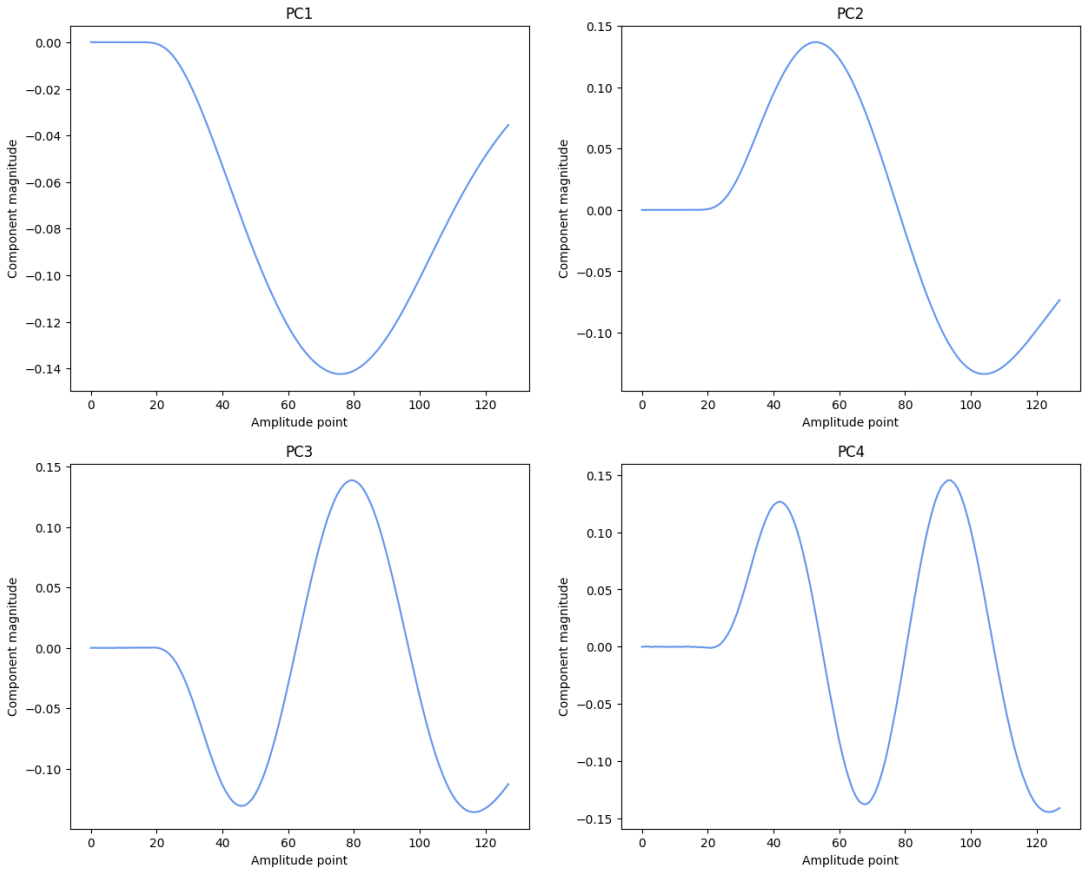


Figure 14: Plots of the magnitudes of each of the 128 amplitude points of the original dataset. The first row shows the two principal components used throughout this paper, while the lower row contains the next two components.

3.4 Correspondence rate

In order to quantitatively compare the outputs of the clustering algorithms without requiring significant resources, a correspondence rate was implemented. This was presented as the percentage of waveforms in the dataset that were sorted into equivalent classes by both clustering algorithms. While this works in theory, it hinges on the algorithms labelling equivalent clusters the same. This is not by default the case, as the labels themselves, simply meaning the numbering, is arbitrary. To enable the correspondence rate calculation, a method must be implemented to enforce a predictable labelling scheme for both algorithms. The chosen method involved linking the cluster number to the mean $PC1$ value of the cluster. This was a logical approach, since the clusters of the PC scatter plots like the one in figure 12(b) were separated by vertical lines. The clustering was allowed to run and arbitrary labels were attributed; two new columns were appended to

the dataset containing the number of the cluster the waveform was assigned to according to both KMeans and Hierarchy. The mean value $PC1$ value of all cluster members was then calculated, and the cluster numbers were redefined; cluster 0 was now defined to be the cluster with the lowest mean $PC1$ value. Increasing cluster number now corresponded to increasing $PC1$ mean. This approach worked well for lower cluster numbers due to the vertical splits between clusters, due to the approach not being significantly affected by the differences between the algorithms. This approach starts to break down, however, when approaching ten clusters as some of the splits are horizontal. These clusters are much more sensitive to minuscule differences in clusters between the algorithms, and as such equivalent clusters are prone to not occupy the same cluster index. This would be possible to solve by extending the method to also consider mean $PC2$ value, but this was not pursued as this project mostly considered two to five clusters where this problem does not present itself. The correspondence rate was calculated for all the datasets used for this research, with all having between a 60% and 100% correspondence rate for sufficiently low cluster numbers. This is both an expected and sufficient result, and was an important tool in revealing and highlighting differences between clustering done by the different algorithms.

4 Additional parameter consideration

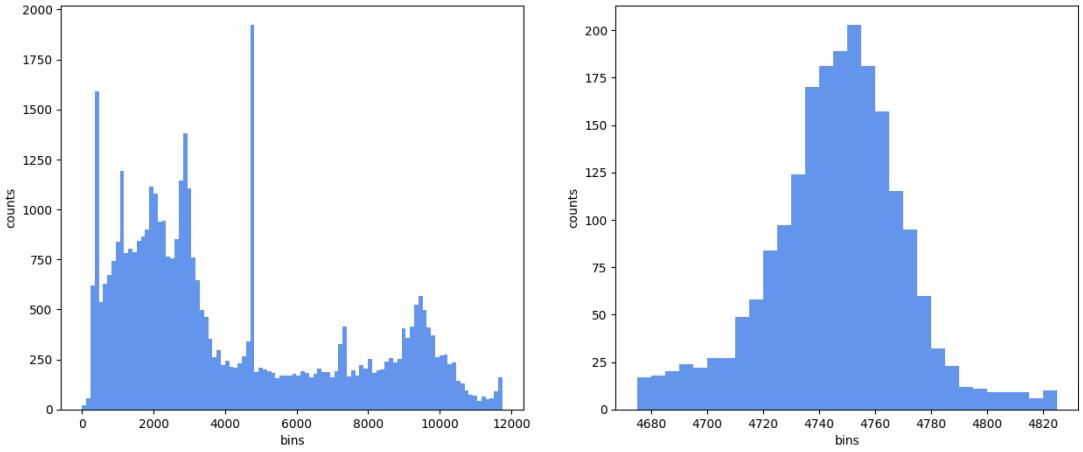


Figure 15: Histograms of energy values for all waveforms registered in a single channel. (a) shows the distribution of the entire dataset, while (b) is a magnification of the section of interest in the former. The bins are in arbitrary energy units.

With the clustering algorithms operational, additional parameters of the waveforms

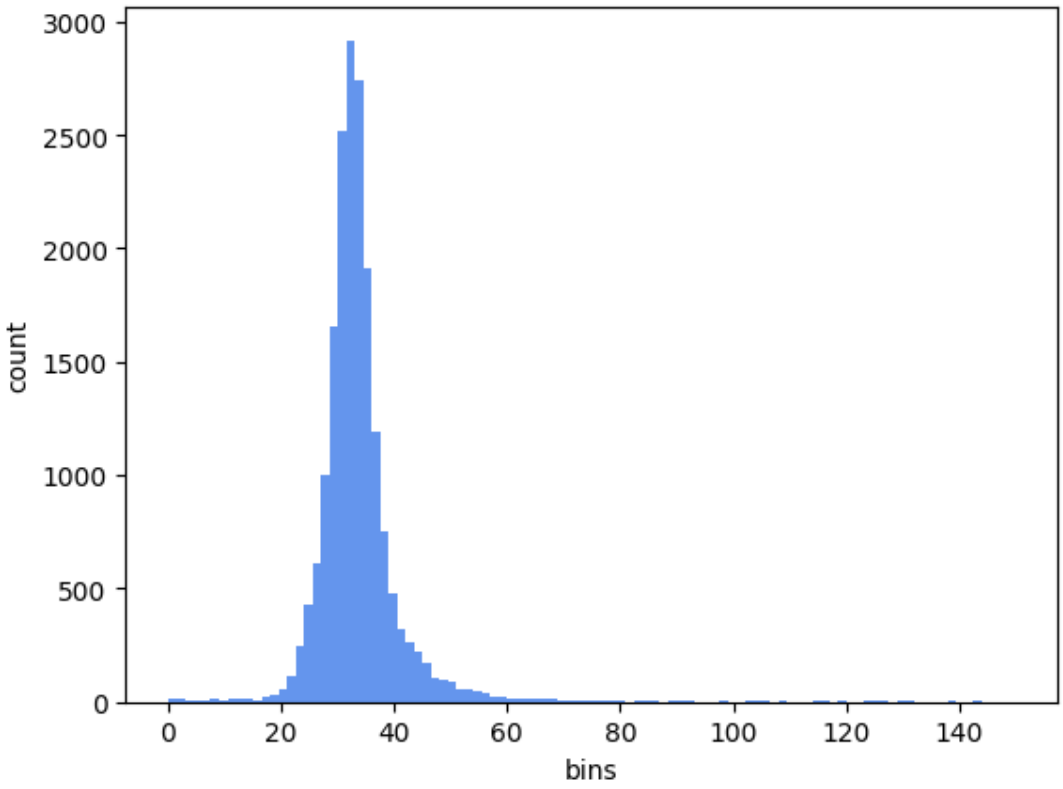
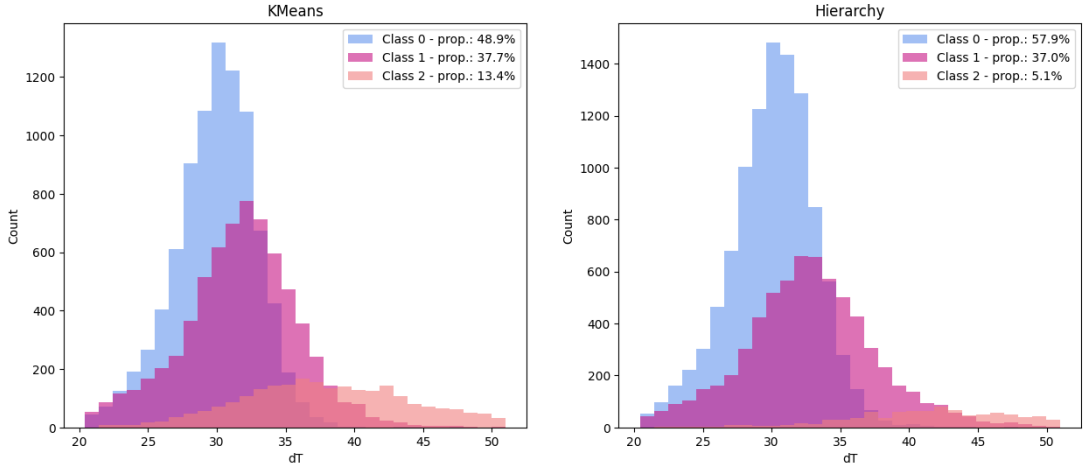


Figure 16: Histogram of dT values for all waveforms registered in a single channel. The bins are in arbitrary time units.

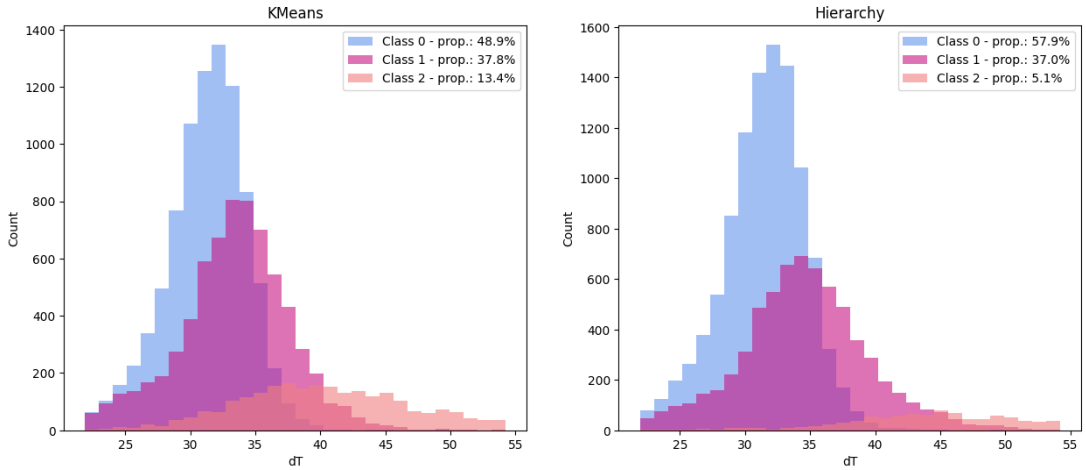
can be considered. This included the Ch, E and dT parameters. E refers to the energy deposited in the detector and the distribution seen in one of the datasets is shown in figure 15. The peak seen at around bin 4750 in (a) is the one expected of this experiment; a magnification of this section can be seen in 15(b). All waveforms in this dataset with E values outside this Gaussian were discarded. This served to replace the outlier removal procedure discussed in section 3.1. By selecting waveforms of the same energy the noise levels will also be similar, which ensures that the clustering is only dependent on the shape of the waveforms as the influence of noise is minimized. This approach is significantly more rigorous, having a proper basis for removal of certain waveforms beyond simply noise level. However, whereas the original outlier removal method removed only about 4% of the dataset, the E consideration removed about 97% of the dataset, or leaving a dataset containing only around 350 waveforms. This naturally lead to quite poor statistics, but was eventually solved by the generation of a new dataset of around 10000 waveforms all within the correct energy Gaussian. Ch refers to the channel, or which detector in the detector geometry present at the experiment the data was recorded

in. This served mostly as a check of the approach used, as only data from detector 0 had been used up until this point. With the exception of detector 4, 6 and 22 in the detector array, which were a different type of detector and hence not in the scope of this research, the approach described in this paper worked for all detectors in the experimental setup. Minor modifications were required to enable this however; minute differences between the detectors lead to displacements of the energy peak, and as such manual inspection was required to set proper bounds for the Gaussian. A possible extension of the work described here would be devising a program to automatically locate the correct peak in the dataset, set appropriate energy bounds and reduce the dataset accordingly. Lastly, dT is the time difference between the observation of an event in the HPGe detector and the reference lanthanum bromide detectors. The distribution of this parameter, using the new dataset described above to maintain proper statistics, can be seen in figure 16. The distribution appears to be a Gaussian with an exponential tail on the right side. With the proper considerations made using the newly introduced parameters, the correlation between the dT parameter and the clusters calculated previously can be determined.

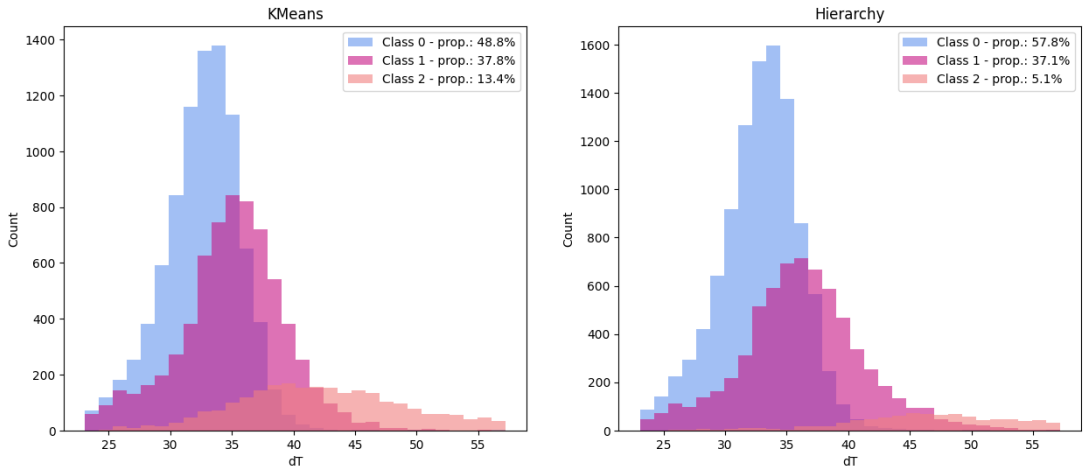
Threshold = 0.6%



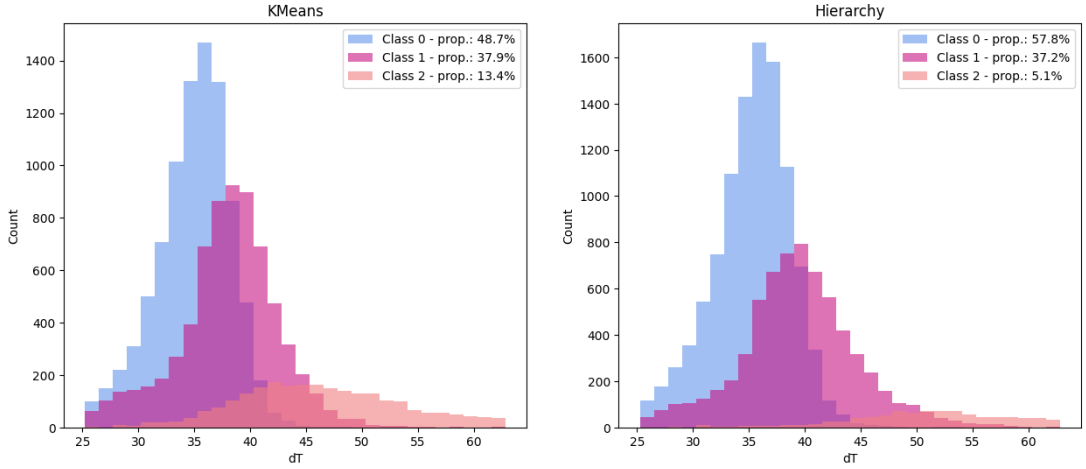
Threshold = 0.7%



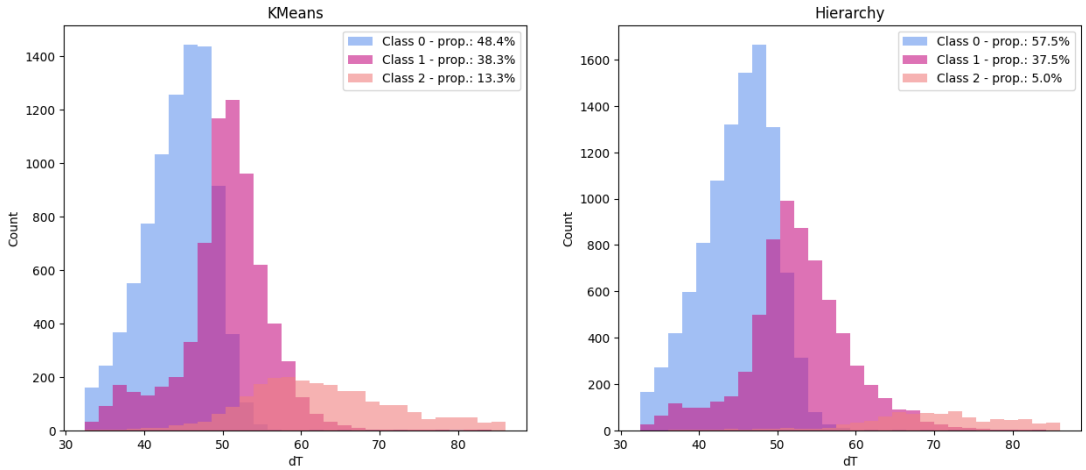
Threshold = 0.8%



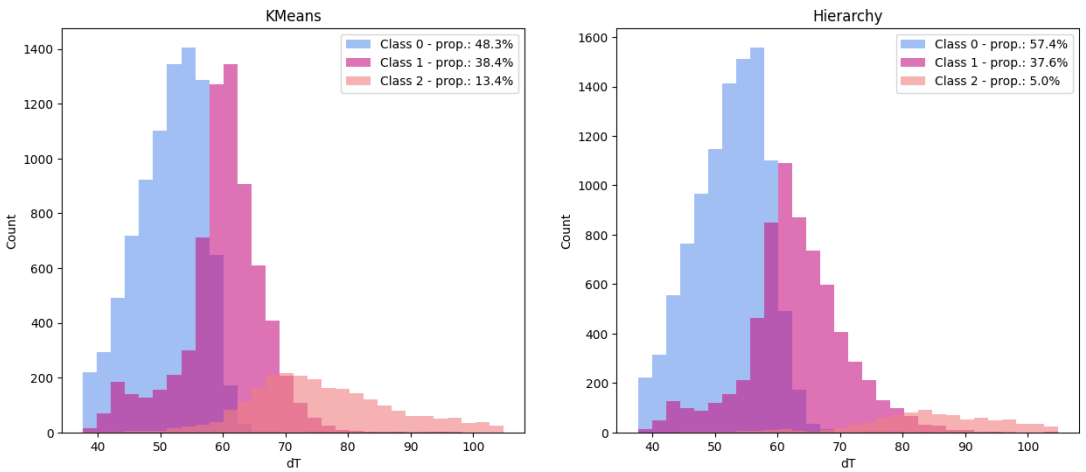
Threshold = 1%



Threshold = 2%



Threshold = 3%



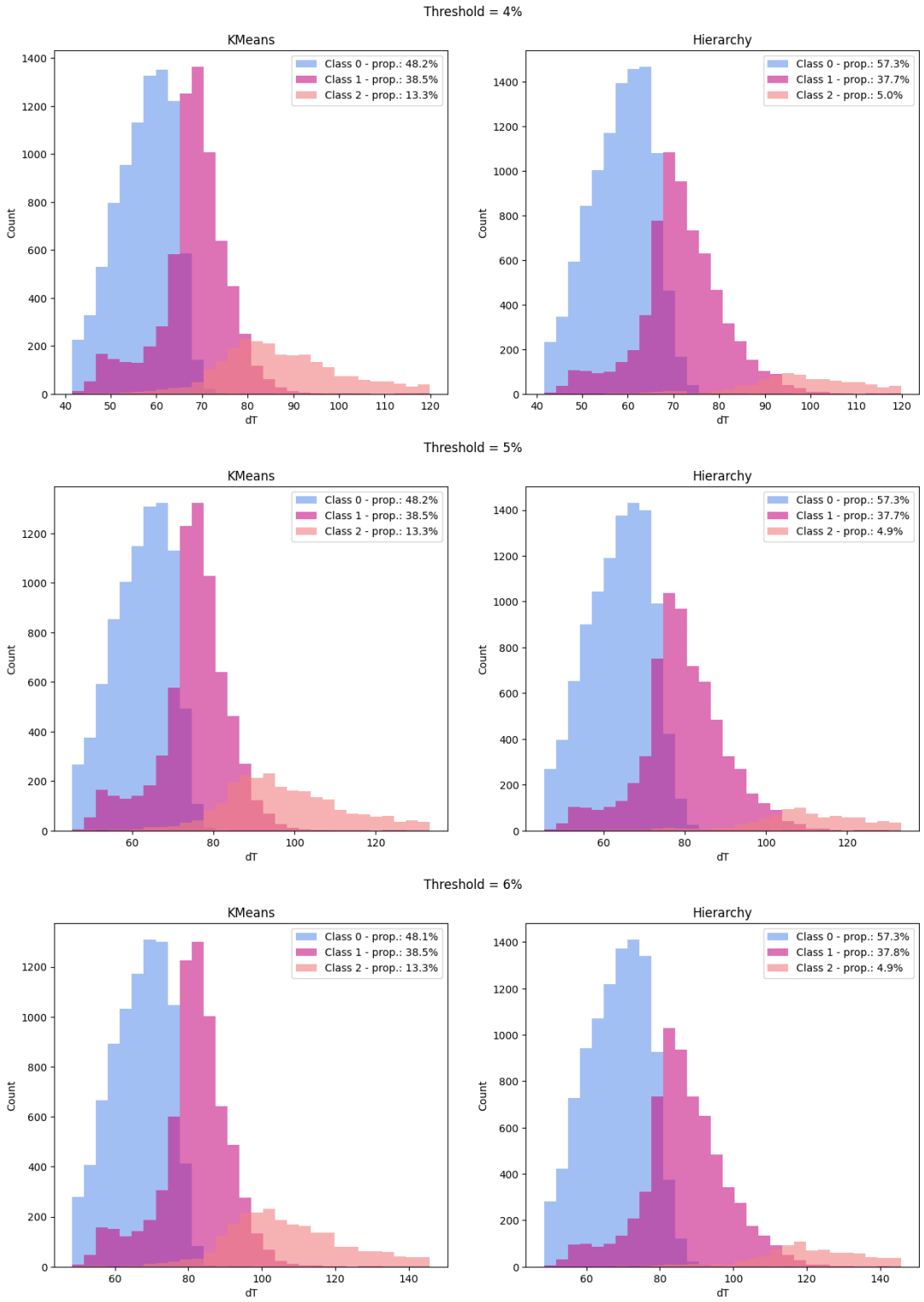


Figure 17: dT distribution split by cluster. Prop. refers to the proportion of the dataset sorted into this cluster and Delay is set in the timing algorithm.

5 dT-cluster correlation

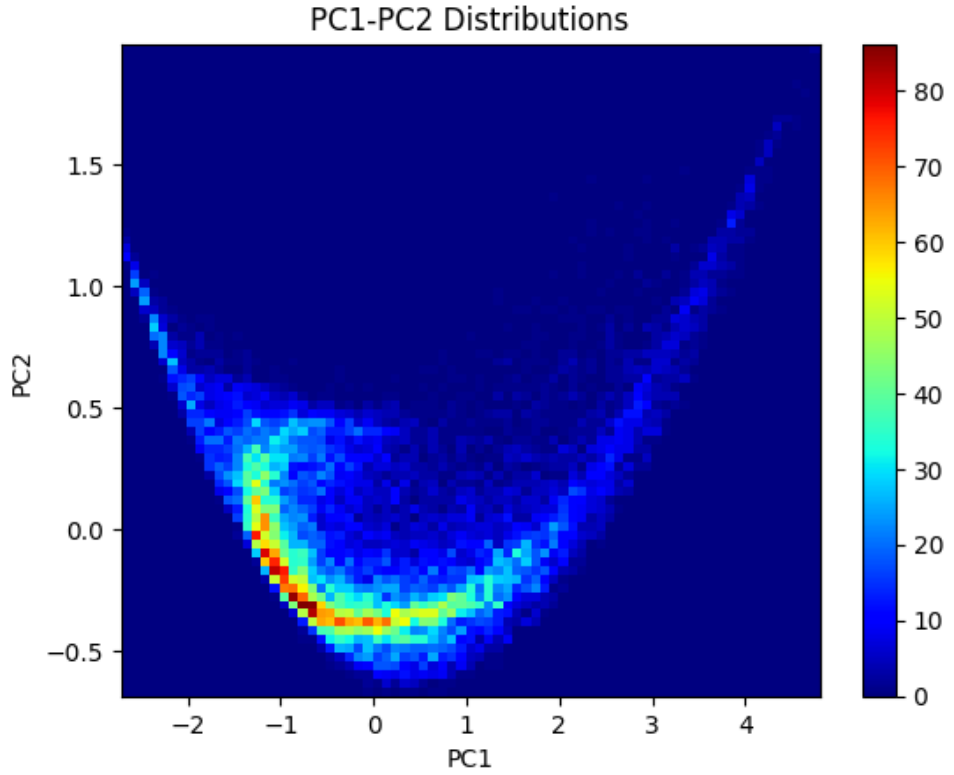


Figure 18: 2D histogram of the two principal components used throughout this thesis, effectively a density plot of figure 11(b).

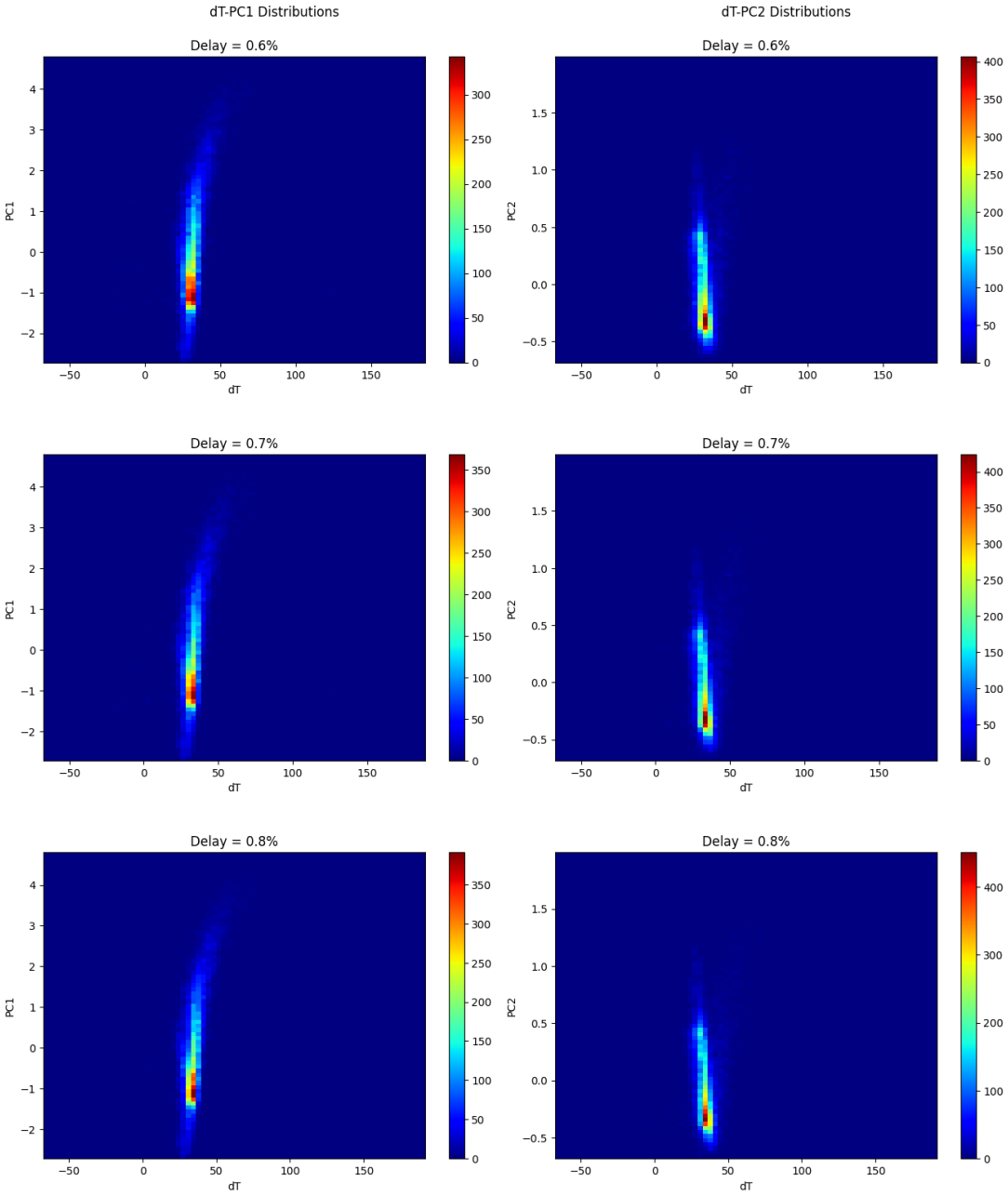
Seen in figure 17 is the same histogram as figure 16, but with the histogram divided into three overlapping histograms based on the cluster to which the waveform belongs. The threshold level used for each set of graphs refers to the threshold described in section 2.1. The results show a strong dependence of the dT parameter on the cluster; cluster number, corresponding to $PC1$ value, is proportional to dT value such that the increase in one leads to an increase in the other. The histograms vary considerably with delay. At lower delay values, the overlap is significant especially between class 0 and 1. With increasing delay, the split between the clusters becomes more significant with delays above 1% producing quite distinct peaks. These findings are significant, as this would allow for more precise corrections to be applied to the dT timing algorithm; different clusters could have varying corrections, which should significantly reduce the deviation of the distribution. A reduction in the variance and elimination of the exponential tail would lead to a narrowing of the distribution of dT, which would lead to an improvement in

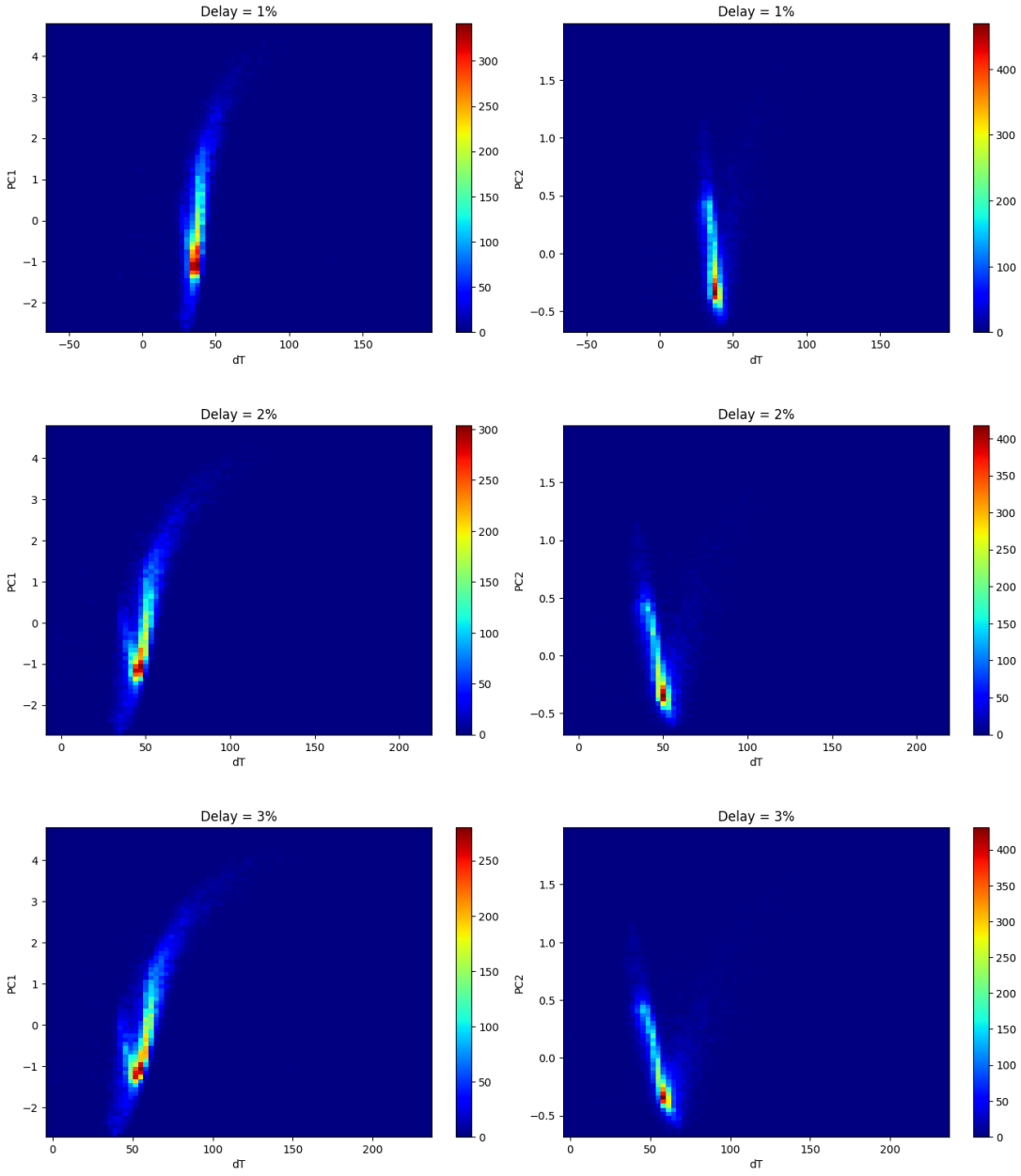
HPGe detector time resolution.

There are also clear differences visible between KMeans and Hierarchy for all the delays. One of these is the difference is proportion of the dataset found in each of the clusters. The relative abundance of waveforms in each cluster, specifically the high percentage found in the earlier clusters, can be understood by considering figure 19. Shown here is a 2D histogram of the two principal components used; it clearly shows that the density is much higher in the bottom left section of the plot, explaining the much higher percentage of waveforms found in earlier clusters. Despite this, Hierarchy consistently sorts around 10% more of the dataset into class 0 than KMeans. This class is already the largest class in both algorithms, so the greater proportion does not benefit the statistics significantly. The proportion in class 1 is approximately equivalent throughout, but class 2 has a much smaller proportion in Hierarchy than KMeans with about 13% in the former and 5% in the latter. This leads to much less significant statistics for Hierarchy, reflected in a very flat and wide distribution less suitable for analysis. KMeans also consistently produces a sharper peak in class 1 despite a similar proportion to Hierarchy, caused by a much lower overlap with class 2.

At this stage a conclusion can be drawn on the effectiveness of the two clustering algorithms that were implemented. As discussed in section 2.2, KMeans was implemented as the default solution with Hierarchy serving the role of sanity check and tool to potentially uncover deeper patterns not easily recognizable. This latter point lost applicability following the implementation of PCA; since only two components were used, the parameters could be easily visualized, and upon inspection it is clear that no pattern was seen that KMeans would not recognize. Because of this, the two algorithms produced very similar results as seen throughout this paper. As discussed above, there are differences in the outputted histograms in figure 17. While KMeans seems favorable, it is hard to say which of these algorithms' characteristics are favorable without considering its application. The similar outputs combined with the significantly lower time complexity of KMeans suggests that it is most suitable for this purpose and should be the designated default algorithm, with Hierarchy perhaps serving niche applications where its characteristics are required.

6 Direct application of Principal Components for dT correction




Delay = 1%
PC1
dT
Delay = 1%
PC2
dT
Delay = 2%
PC1
dT
Delay = 2%
PC2
dT
Delay = 3%
PC1
dT
Delay = 3%
PC2
dT

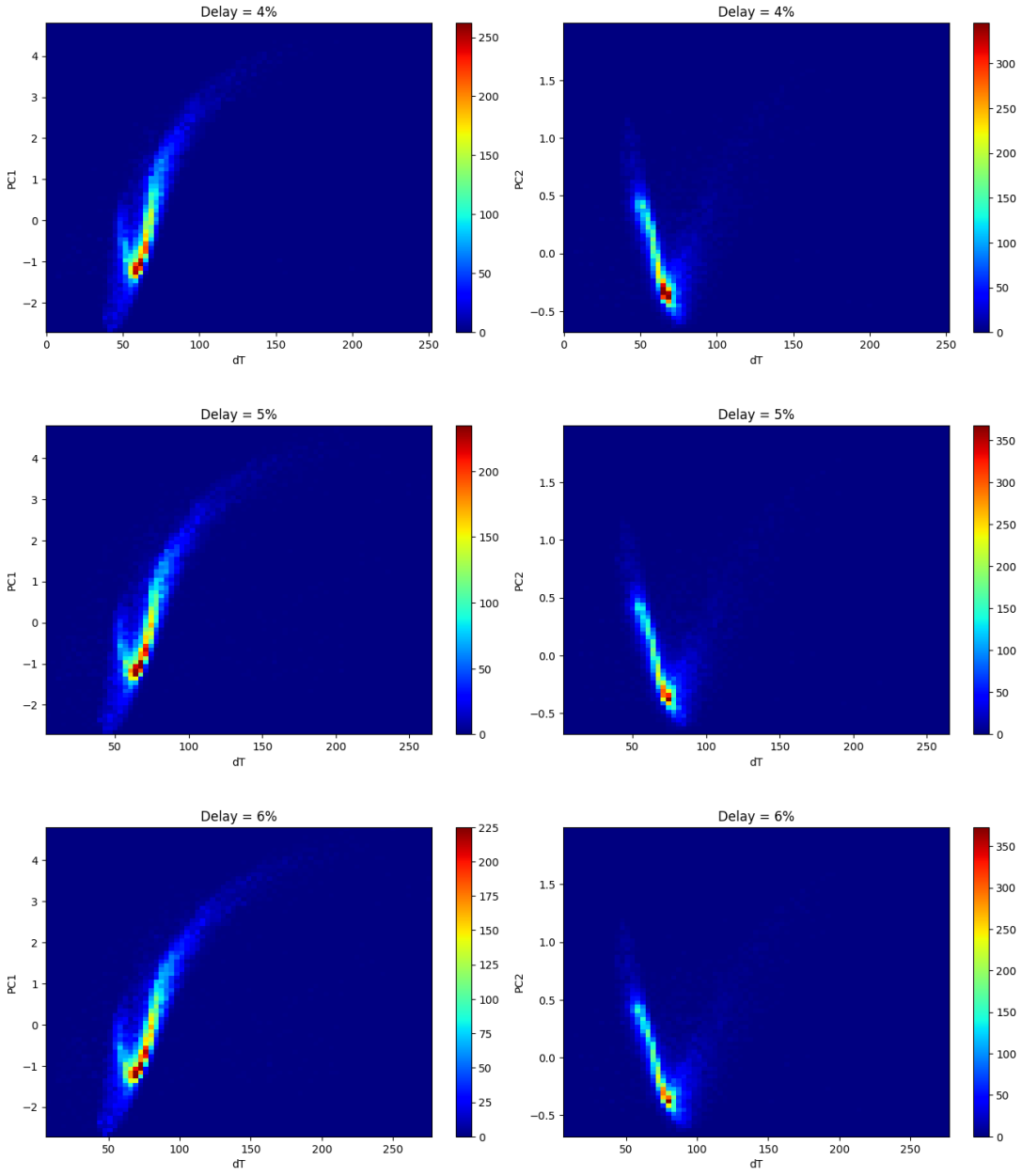


Figure 19: 2D histogram of dT and $PC1$ (left) and $PC2$ (right). All delays are shown.

By using the principal components directly, it may be possible to apply corrections to the waveforms without applying clustering first. Clustering has the downside of requiring a number of clusters to use, which is necessarily arbitrary. Low cluster numbers will have limited application due to relatively small splitting of the clusters, while high cluster numbers will struggle from low population in each leading to poor statistics. This approach could be bypassed entirely by considering the plots in figure 19. Here, a 2D histogram

of dT versus each principal component is shown and a trend emerges. For $PC1$, most of the data is in an approximately straight vertical line (same dT values), but a waning tail starts to form at higher $PC1$ values. This waning section is what becomes the later classes and what causes the exponential tail on the right in figure 16. A similar pattern is observed with $PC2$, but here the very little of the tail is straight; instead, the entire section is angled to the left. Additionally, a smaller tail pointing to the right can be seen. For this reason, in addition to the relatively small contribution of $PC2$ to the overall variance captured, means that $PC1$ would be more suitable for this type of analysis. Considering these findings, it would be possible to devise an algorithm that applies a correction based on the $PC1$ value; this would have the effect of straightening out the tail, such that the values of $PC1$ would be distributed more evenly around the mean. For this, delays around 1% should be suitable. Due to time constraints, attempting such an algorithm for this thesis was not feasible. If this method was implemented successfully, however, it would allow for the correction of the timing in a continuous manner rather than the arbitrarily clustered approach whose flaws were discussed above, which should yield significantly better results and is viewed as a natural extension to the work carried out in this thesis.

7 Conclusion

This paper aims to improvethe time resolution of HPGe detectors used at the GAINS experimental setup at GELINA, which investigates neutron cross sections. This improvement is crucial, as better time resolution enables more close determination of neutron time-of-flight, which is closely linked to the neutron energy. Precise knowledge of the neutron energy is important due to the great variance in cross section found in the resonance region. To improve the detector time resolution, the machine learning approach clustering was applied to a reduced form of the detector data. The two clustering algorithms KMeans and Hierarchical Clustering Algorithm, referred to throughout as Hierarchy, were chosen. The initial data reduction approach involved time and amplitude quantization, but did not produce meaningful results. The final data reduction approach used was principal component analysis, which successfully reduced the dataset to only two parameters containing above 95% of the variance of the original dataset and was well suited for clustering. Upon introducing the energy of the waveforms to the dataset, all waveforms not having energies corresponding to the expected decays were filtered out. The time difference compared to a reference detector was also added to the dataset, and

the distribution of this parameter observed through the lens of the clusters found prior. It was found that there was a strong correlation between cluster number, effectively mean value of the first principal component of the cluster, and dT. This lead to a "splitting" of the histogram, with the clusters forming approximately normal distributions with differing means. This effect was more drastic for higher delay values in the data preprocessing algorithm. These results are significant, as they allow for more precise corrections the waveforms; a correction could be determined for each cluster, resulting in the mean of all the clusters coinciding. This would reduce the deviation of the dT distribution, serving to improve the time resolution of HPGe detectors. This thesis makes no attempt at determining or applying such corrections, but views this as a natural extension of the work demonstrated. A more appealing extension is the treatment of the principal components directly; it was determined that with increasing $PC1$ values, dT stayed relatively constant before increasing. By fitting a function to this trend, appropriate corrections can be determined that treat the waveforms continuously, bypassing the arbitrariness of the number of clusters used and the issues this approach brings. In summary, this paper determined a strong correlation between clusters determined through machine learning applied to detector waveforms and the time difference between said detector and a reference detector with much better time resolution, and suggest approaches to use this correlation to improve HPGe detector time resolution. Extensions to the work carried out are also put forward that would serve to improve the results determined.

8 References

[1] M. Kerveno, A. Bacquias, C. Borcea, *et al.*, *From γ emissions to (n,xn) cross sections of interest: The role of gains and grapheme in nuclear reaction modeling*, en, Dec. 2015. DOI: 10.1140/epja/i2015-15167-y. [Online]. Available: http://dx.doi.org/10.1140/epja/i2015-15167-y.

[2] S. Lukic, "Measurement of (n,xn) reaction cross-sections using prompt gamma-ray spectroscopy at a beam with very high instantaneous flux," Oct. 2004.

[3] A. Negret, *Neutron induced reactions cross section measurements*. [Online]. Available: https://tandem.nipne.ro/nuclear_reactions.php.

[4] J. Boson, G. Ågren, L. B.-Å. Johansson, K. Lidström, and T. Nylén, "Improving calibration of hpge detectors for in situ measurements – a comparison of semi-empirical and monte carlo methods," 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:46549533.

[5] [Online]. Available: https://www.mirion.com/discover/knowledge-hub/articles/education/germanium-detector-types.

[6] L. Mihailescu, C. Borcea, and A. Plompen, "Data acquisition with a fast digitizer for large volume hpge detectors," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 578, no. 1, pp. 298–305, Jul. 2007. DOI: 10.1016/j.nima.2007.05.231.

[7] S. b. Granite, *Constant fraction discrimination: Tcspc*, Nov. 2023. [Online]. Available: https://www.edinst.com/blog/constant-fraction-discrimination/.

[8] *Comparing different clustering algorithms on toy datasets — scikit-learn.org*, https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html, [Accessed 09-06-2024].

[9] D. Xu and Y. Tian, *A Comprehensive Survey of Clustering Algorithms by Dongkuan Xu, Yingjie Tian*. 2015.

[10] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee, and R. Enayatifar, "From clustering to clustering ensemble selection: A review," *Engineering Applications of Artificial Intelligence*, vol. 104, p. 104 388, Sep. 2021. DOI: 10.1016/j.engappai.2021.104388.