Machine Learning Applications for Waveform Analysis

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Micah Cruz

August 2021

This thesis is dedicated to my parents, whose profound confidence in me have enabled my academic journey, and my fiancée, Taylor, who has been a constant source of happiness during the bleak abyss known as graduate school.

Acknowledgments

First and most importantly, I would like to thank my advisor, Dr. Nadia Fomin, for taking me on as her student. Her knowledge, patience, and support have enabled my growth as a researcher and physicist. She has inspired me not only as a scientist, but as a person who invests so much of her time in the well-being of others. I am incredibly grateful to have her as a mentor and a friend.

I would also like to thank Elliot Perryman, for introducing me to the field of machine learning, which started my research path towards my M.S. He has provided much valuable advice during my graduate studies.

I would like to thank Noah Birge for his expertise on the ^{45}Ca data, and Jimmy Caylor for his expertise on the BL2 data. Both of them made my work possible and greatly helped me to understand the data.

I would like to thank Di'Arra Mostella, who provided the BL2 pseudodata which enabled the generation of a predictive model. His work was vital to my analysis of BL2 data.

I would also like to thank Eli Carter for his contributions to both unsupervised and supervised learning analyses of BL2 data. He applied multiple promising methods that I had not implemented, and I am very intrigued to learn of his future work with the Fundamental Neutron Physics Group.

I would like to thank Grant Riley, who worked with our group as a post-doc for well over half of my time as a graduate student. He provided countless amounts of insight and taught me much about laboratory work.

I would also like to thank my committee, Dr. Kate Jones, Dr. Andrew Steiner, and Dr. Adrian Del Maestro. I am very grateful for the time they dedicated to reading and critiquing my work.

Lastly, I would like to thank all of the faculty at the University of Tennessee's Department of Physics and Astronomy. They have both challenged and encouraged me through my journey as a student.

Abstract

Since the later 20$^{\text{th}}$ century, the search for physics beyond the Standard Model (BSM) has been paramount to many nuclear and particle physicists. Neutron and nuclear beta decay experiments provide one avenue to search for evidence of BSM physics by contributing to the unitarity check of the Cabibbo-Kobayashi-Maskawa matrix. Many of these experiments detect neutron decay products as digitized waveforms. As computing power increases and novel algorithms are developed, it is compelling to investigate machine learning methods as an analytic tool for such waveform data. These methods can allow for very fast data exploration techniques, and if pseudodata is available predictive models can be built for tasks such as particle identification. This thesis will report machine learning analysis done for both the ^{45}Ca Beta Spectrum Measurement at LANL and the BL2 Neutron Lifetime Measurement at NIST.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Standard Model

The Standard Model of particle physics is a quantum field theory that describes the strong, weak, and electromagnetic interactions. Each particle interaction can be described using these fields, and the fundamental particles detailed by the Standard Model make up the known universe. These particles are shown in Figure 1.1. Though the Standard Model is one of the most noteworthy scientific achievements of the 20$^{\text{th}}$ century, there is evidence that it is not complete. It is lacking a description of the gravitational interaction, and the discovery of neutrino flavor oscillations proved the Standard Model's prediction of a zero mass neutrino to be incorrect. Furthermore, observations such as baryon asymmetry, dark matter, and dark energy are left unexplained by the Standard Model. Taking these deficits into consideration, it becomes intriguing to test the Standard Model and search for evidence of beyond Standard Model (BSM) physics.

1.2 Testing the Standard Model via Nuclear Beta Decay

Nuclear beta decay is one of the most fundamental interactions in particle physics. A nucleus of any size can undergo beta decay if it is energetically favorable; however, the

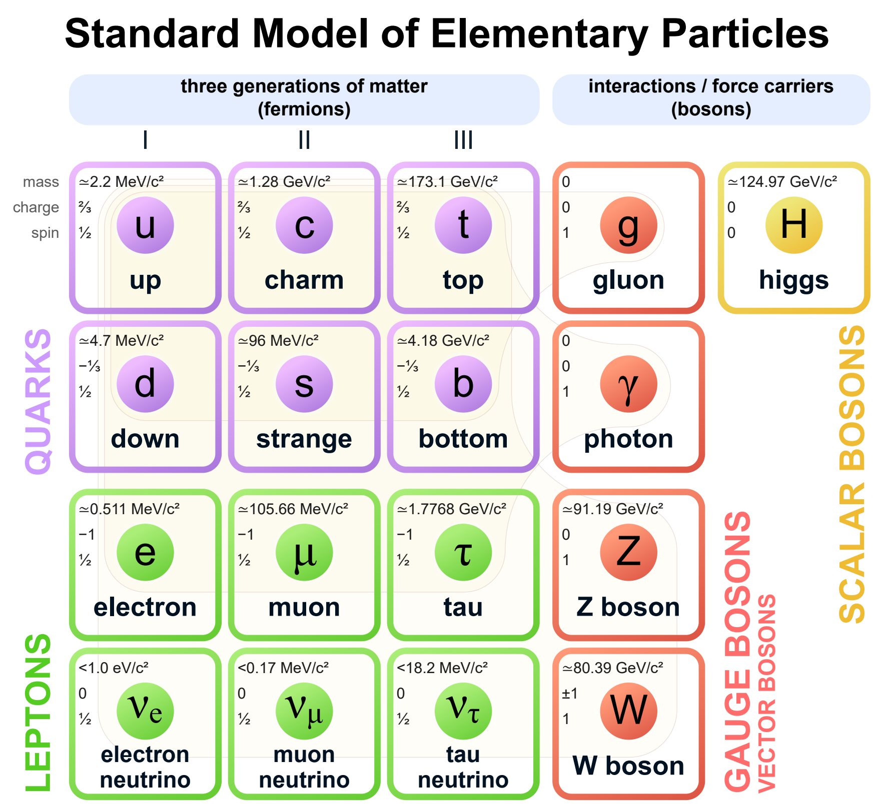

Figure 1.1: Particles included in the Standard Model[12]. The leptons and quarks are fermions (spin $\frac{1}{2}$ particles) and compose all known matter in the Universe. The vector bosons (spin 1 particles) mediate the fundamental interactions, and the Higgs is a scalar boson (spin 0 particle) that gives mass to the particles it interacts with.

most elementary example is that of free neutron beta decay. This is laid out in equation 1.1, where the neutron decays into a proton and a virtual W- boson. The W- boson then decays into an electron and an electron antineutrino.

$$n \rightarrow p + W^{-} \rightarrow p + e^{-} + \bar{v}_{e^{-}} \tag{1.1}$$

The W- boson mediates the flavour changing decay, where a down quark in the neutron transitions to an up quark, forming a proton. This process of quark mixing is one of nine predicted by the Standard Model of particle physics, and the mixing probabilities are given by the matrix elements of the CKM Matrix (Cabibbo-Kobayashi-Maskawa) [19], shown in equation 1.2.

$$\begin{bmatrix} d' \\ s' \\ b' \end{bmatrix} = \begin{bmatrix} V_{ud} & V_{us} & V_{ub} \\ V_{cd} & V_{cs} & V_{cb} \\ V_{td} & V_{ts} & V_{tb} \end{bmatrix} \begin{bmatrix} d \\ s \\ b \end{bmatrix} \tag{1.2}$$

This matrix should be unitary under the Standard Model. It is therefore compelling to experimentally verify the unitarity of this matrix, as any non-unitarity would be an indication of BSM physics. The first row unitarity test [19] (shown in equation 1.3) is generally taken most seriously, as the errors on $|V_{ud}|$ and $|V_{us}|$ are small, and $|V_{ub}|$ is small enough to have little effect on the unitarity check.

$$|V_{ud}|^2 + |V_{us}|^2 + |V_{ub}|^2 = 1 \tag{1.3}$$

$|V_{ud}|$ is the largest contributor to this unitarity check, and therefore a precise determination of $|V_{ud}|$ provides valuable insight into BSM physics. Equation 1.4 shows how $|V_{ud}|$ can be determined.

$$|V_{ud}|^2 = \frac{\Gamma}{|g_V|^2 G_F^2 (1 + 3|\lambda|^2)} \tag{1.4}$$

In the above equation, Γ is the neutron decay rate ($\Gamma = \frac{1}{\tau_n}$ where τ_n is the neutron lifetime), g_V is the vector weak nucleon form factor at zero momentum transfer, G_F is the Fermi weak coupling constant, and λ is the ratio of axial-vector to vector coupling constants, $\lambda = \frac{G_A}{G_V}$.

The unknown terms in this equation are Γ and λ. Thus, an experimental determination of both of these values is required in order to test the unitarity of the CKM matrix. Neutron lifetime experiments allow for determination of Γ, while measurement of neutron decay parameters allow λ to be measured. The following sections will go into detail on how these are experimentally measured.

1.3 Measurement of Neutron Beta Decay Parameters

Free neutron beta decay is characterized by the triple differential decay rate, given by equation 1.5 [9]:

$$\frac{dw}{dE_e d\Omega_e d\Omega_v} \propto p_e E_e (E_0 - E_e)^2 \Big[1 + a\frac{\vec{p_e} \cdot \vec{p_v}}{E_e E_v} + b\frac{m_e}{E_e} + \langle \vec{\sigma_n} \rangle \cdot (A\frac{\vec{p_e}}{E_e} + B\frac{\vec{p_v}}{E_v} + ...)\Big] \tag{1.5}$$

Where $p_{e(v)}$ is the electron (neutrino) momentum, $E_{e(v}$ is the electron (neutrino) energy, E_0 is the electron energy spectrum endpoint, and $\vec{\sigma}_n$ is the neutron spin. The lower-case terms (a, b) require an unpolarized neutron source, while the upper-case terms (A, B, ...) require a polarized neutron beam. The specific experiments being studied for this thesis utilized only unpolarized neutrons, so the upper-case terms fall out. This results in a simplified equation, shown in equation 1.6.

$$\frac{dw}{dE_e d\Omega_e d\Omega_v} \propto p_e E_e (E_0 - E_e)^2 \Big[1 + a\frac{\vec{p_e} \cdot \vec{p_v}}{E_e E_v} + b\frac{m_e}{E_e}\Big] \tag{1.6}$$

The lower-case term a is known as the electron-neutrino correlation, and it is related to λ (as seen in equation 1.4) by equation 1.7:

$$a = \frac{1 - |\lambda|^2}{1 + 3|\lambda|^2} \tag{1.7}$$

Thus, a determination of a provides a measurement of one of the two terms needed to measure $|V_{ud}|$ and search for evidence of BSM physics. While measurement of b does not probe the value of $|V_{ud}|$, it is proportional to scalar and tensor couplings theorized in BSM physics, and as such is zero in the Standard Model. A nonzero determination of b would therefore indicate possible physics beyond the Standard Model. While the discussion up to this point has included only free neutron beta decay, these effects can be similarly studied in nuclear beta decay, as in the ^{45}Ca experiment at Los Alamos National Lab. [2]

1.4 Measurement of Neutron Lifetime

The lifetime of the neutron must also be accurately determined in order to make the unitarity check from equation 1.3. There are two primary methods to measure the neutron lifetime: "beam" methods and "bottle" methods. These experiments rely upon "cold" and "ultracold" neutrons. Cold neutrons have kinetic energies ranging up to 25 meV, while ultracold neutrons have very low kinetic energies – up to 100 neV. Experiments using beam methods generally try to measure neutrons from a cold neutron beam and their decay products within some decay volume. Cold neutrons are preferred, since they spend longer time in the decay volume and neutron flux monitors are more efficient with lower energies. The most precise determination uses the Sussex-ILL-NIST technique [18], where a proton trap is used in the decay volume to trap and count the protons, while the neutron flux in the decay volume is also measured. Taking a ratio of the two rates allows the neutron lifetime to be determined.

For bottle methods, ultracold neutrons are trapped in a container (material [16] or magnetic [6]), and the number of neutrons is measured at specific time intervals in order to determine the decay rate. These extremely low energy neutrons are vital for bottle experiments, as they can be reflected and contained in the bottle. The neutron loss rate τ_l must be accounted for, which is related to the storage lifetime τ_s via $\tau_s^{-1} = \tau_n^{-1} + \tau_l^{-1}$. The number of neutrons at time t can be expressed with the typical exponential decay equation, given by equation 1.8.

$$N(t) = N_0 e^{\frac{-t}{\tau_s}} \tag{1.8}$$

Then the storage time can be expressed as $\tau_s = \frac{\Delta t}{\ln \frac{N_1}{N_2}}$ where $\Delta t = t_2 - t_1$, which then allows the neutron lifetime to be expressed as follows in equation 1.9.

$$\tau_n = \frac{\tau_s}{1 - \frac{\tau_s}{\tau_l}} \approx \frac{\Delta t}{\ln \frac{N_1}{N_2}} \Big[1 + \frac{\Delta t}{\tau_l \ln \frac{N_1}{N_2}} + (\frac{\Delta t}{\tau_l \ln \frac{N_1}{N_2}})^2 + ... \Big] \tag{1.9}$$

The most recent determination of the neutron lifetime with this method comes from the 2018 LANL publication, with a reported value of 877.7s $\pm$0.7s. [14]

Currently there is a 4σ discrepancy between the lifetime values as determined by these two methods. Figure 1.2 shows this discrepancy via the history of these beam vs. bottle experiments and their published values for the neutron lifetime. It is then possible to plot $|V_{ud}|$ vs. λ, as shown in Figure 1.3, in order to visualize how current measurements of λ and τ_n fit into the CKM unitarity test. The Particle Data Group lists the current value for $|V_{ud}| = 0.97370 \pm 0.00014$ [19], which is outside of the range for CKM unitarity; however, it is clear that then neutron lifetime discrepancy must first be resolved before any determination of $|V_{ud}|$ can be trusted.

1.5 Data Analysis and Applying Machine Learning Methods

Many of the experiments mentioned in this introduction digitize detector signals as a function of time, producing waveform data. These waveforms are traditionally analyzed multiple ways, such as shaping electronics or software filters, to extract the energy and timing information. With the recent developments in machine learning, many new analysis methods are available that have not been evaluated in this context.

Machine learning (ML) refers to a broad class of algorithms for data analysis that can study and learn from the data without being explicitly programmed to do so. These algorithms can be generalized into three categories: supervised learning, unsupervised learning, and reinforcement learning. Generally, only supervised and unsupervised algorithms are relevant for analysis of waveform data. A supervised learning algorithm will build a predictive model by learning from labeled training data, while an unsupervised algorithm will look for underlying patterns or outliers in the data set. Both types require relatively little user input compared to traditional analysis; however, they are regarded as a "black box" method especially when applied to physical data. ML methods are often not accurate enough to completely replace traditional analyses, but they offer novel uses for physicists, as will be

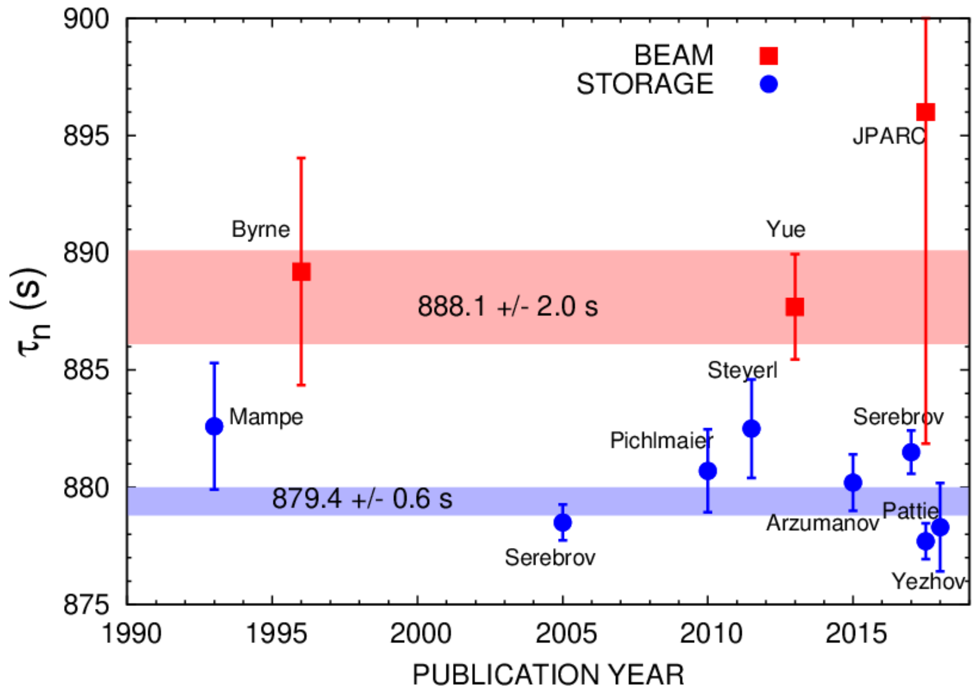

Figure 1.2: Neutron lifetime discrepancy through the past 30 years[5]

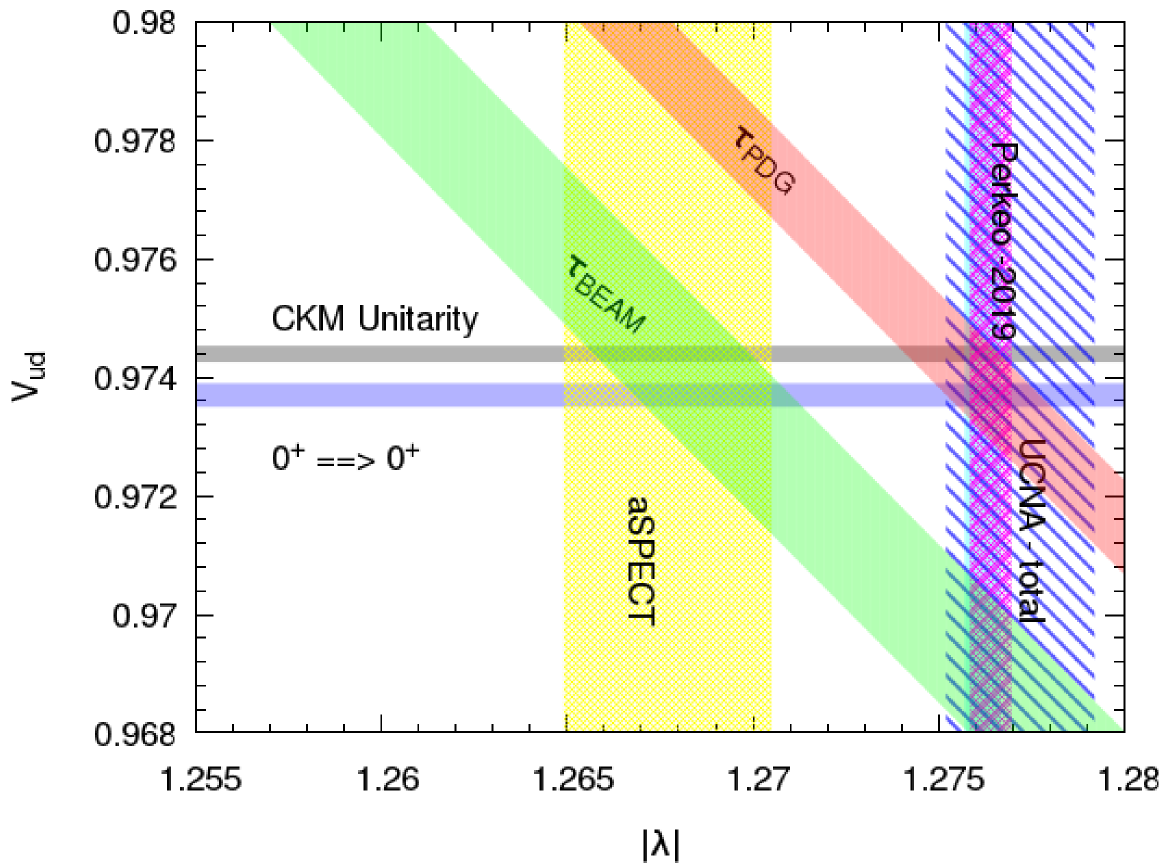

Figure 1.3: V_{ud} vs. $|\lambda|$ with various experimental measurements shown, showcasing the lack of consensus in the data, and as such it is unclear if the data is consistent with CKM unitarity[5]. Note that the best precision measurements for V_{ud} come from nuclear beta decay; however, free neutron beta decay measurements are also important as there are no nuclear structure uncertainties to consider, making the results easier to interpret.

shown in chapters 3 and 4. The work done for this thesis evaluated a number of machine learning methods to be used for particle identification in the NIST in-beam neutron lifetime measurement and for data exploration in the ^{45}Ca beta spectrum measurement at LANL.

Chapter 2

Features of Experimental Data

2.1 Overview

Before any machine learning analysis can be done, it is important to understand the features of the data set and how to pre-process the data for best results. The two data sets being used are the ^{45}Ca Beta Decay Spectrum data set and the NIST Neutron Lifetime Measurement data set. This section will outline the features from each data set and the considerations that were made before any machine learning analysis was done.

2.2 ^{45}Ca Beta Spectrum Data

This experiment aimed to make a measurement of the Fierz Interference term (b from Equation 1.5) by studying nuclear beta decay from ^{45}Ca. To do so, a water-based solution of ^{45}Ca was deposited onto a polymer film and dried. This film was placed in a superconducting solenoidal magnet, and electrons from the decay were guided by the magnetic field to be captured by silicon detectors [3]. These detectors consisted of 127 hexagonal segments referred to as detector pixels. A diagram of the experimental setup is shown in Figure 2.1.

Electrons were the primary event type being recorded, and these were present with a large range of energies. Electrons would sometimes backscatter, only partially depositing their energy before being captured on the second (or higher) incidence with the detector.

Other times, multiple electron events would "pile up" by being detected so close together that they were hard to distinguish from a single event. Since pseudodata for this experiment was not available, it was compelling to see if clustering algorithms (an unsupervised learning method) could identify these events, or other "hidden" features of the data.

Waveforms from the ^{45}Ca beta decay experiment were digitized and recorded with 3500 time bins, a peak energy corresponding to the energy of the incoming particle, and an exponential decay characteristic to the silicon detectors. One time bin corresponds to 4 nanoseconds, and the waveforms were recorded at 1 ADC = 0.15 keV. The data set being studied contained 4250 total waveforms. The data acquisition system was configured such that the data is buffered, a software filter is applied in the field-programmable gate array, and the output is used as a trigger. If the trigger threshold is met, the waveform is then read out with the triggering event beginning at time bin 1000. When an event is detected, neighboring pixels are also read to check for charge sharing. These neighboring pixels sometimes contained off time events, which should not be counted as they carry their own trigger. An example waveform for an electron event is shown in Figure 2.2.

While this data was being studied, oscillations in the baseline with a period of about 56 μs were noticed. These oscillations were thought to have originated from the physical shaking of the cooling lines, as they would disappear when the cooling system was off. Effects such as oscillations, backscattering, or pile up can all contribute to distortions in the measured electron energy spectrum, leading to incorrect results.

2.3 NIST Neutron Lifetime Data

The Beam Lifetime 2 (BL2) experiment at NIST measures the neutron lifetime by accurately counting both neutrons and protons [8, 13]. The neutron flux is monitored via capture on a ^{6}Li film. The resultant alpha particles and tritons are then detected by one of four silicon detectors. The protons, which have a maximum energy of 751 eV, are trapped in the decay volume by an 800V potential which constrains them longitudinal to the beam, and a 4.6T magnetic field which constrains them tranverse to the beam [13]. They are trapped for 10 ms before being released towards the silicon proton detector and accelerated by a 30kV

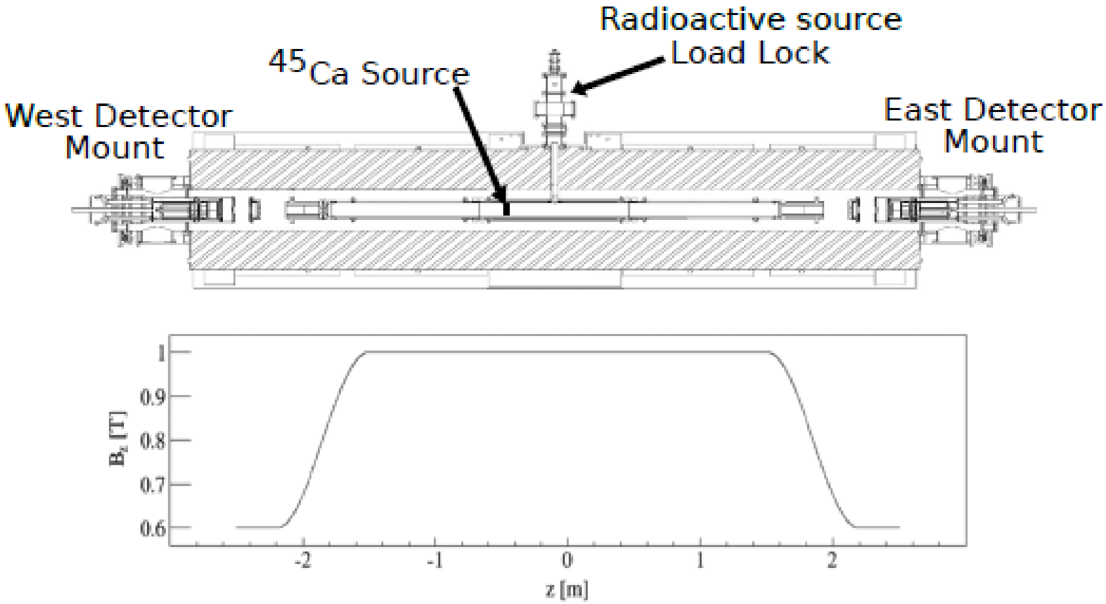

Figure 2.1: ^{45}Ca experimental diagram

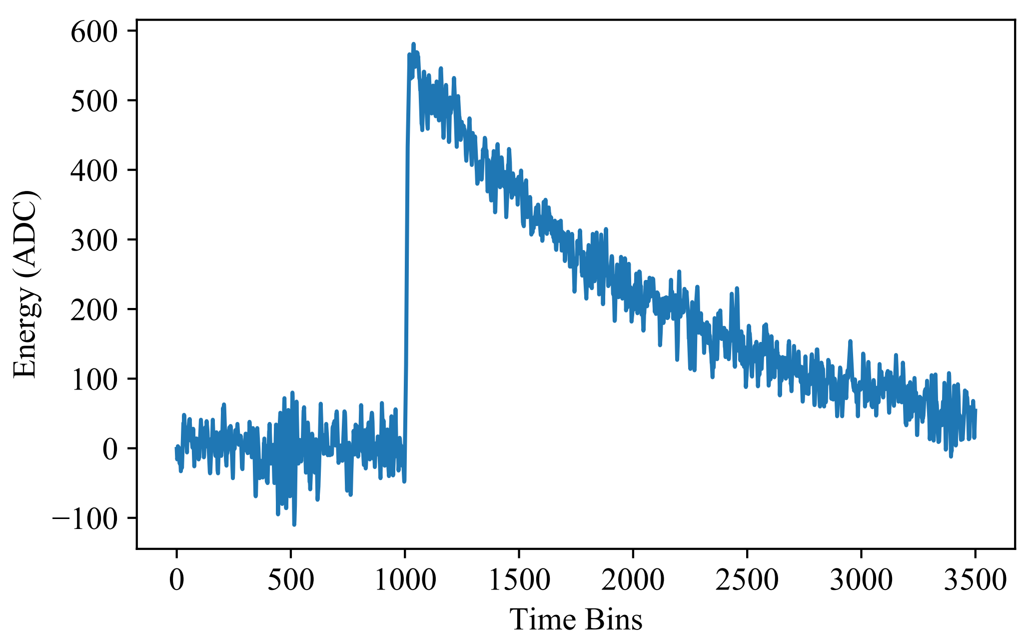

Figure 2.2: ^{45}Ca example waveform

potential in order to penetrate the detector's dead layer. A diagram of the experimental setup is shown in Figure 2.4.

Some protons that are not part of the trapping cycle are still detected. These untrapped protons are from real neutron decays in-flight, but they should not be counted. They are hard to distinguish from protons of interest, and are handled instead via background subtraction. The data that was studied for this thesis was exclusively proton detector data/pseudodata.

2.3.1 Pseudodata for Neutron Lifetime Measurement

Pseudodata for the neutron lifetime experiment was created by adding the RMS noise distribution from the real data to characteristic signals. The shape of these signals is governed by the response of the electronics, which dictates an exponential decay in voltage after the initial particle detection. Equation 2.1 approximates this shape.

$$V(t) \propto (\frac{1}{\tau})^n e^{-t/\tau} \tag{2.1}$$

Where n is a free parameter that is fit to the data.

Signals such as multi-proton events, electrons, and cosmic rays are also present in the data stream, and thus are also present in the pseudodata. The probabilities for these events in the pseudodata were extracted directly from experimental data.

The pseudodata was generated with 9 possible event types, given by Table 2.1. The most important events are those containing protons, as each proton must be correctly counted to make an accurate measurement of the neutron lifetime. The pseudodata used for training these algorithms used an equal split for number of events in each event type. This was to ensure that rarer event types had a statistically significant amount of data points to train on. When the predictive models were evaluated, they were run on a different set of pseudodata that had the same event distribution as experimental data.

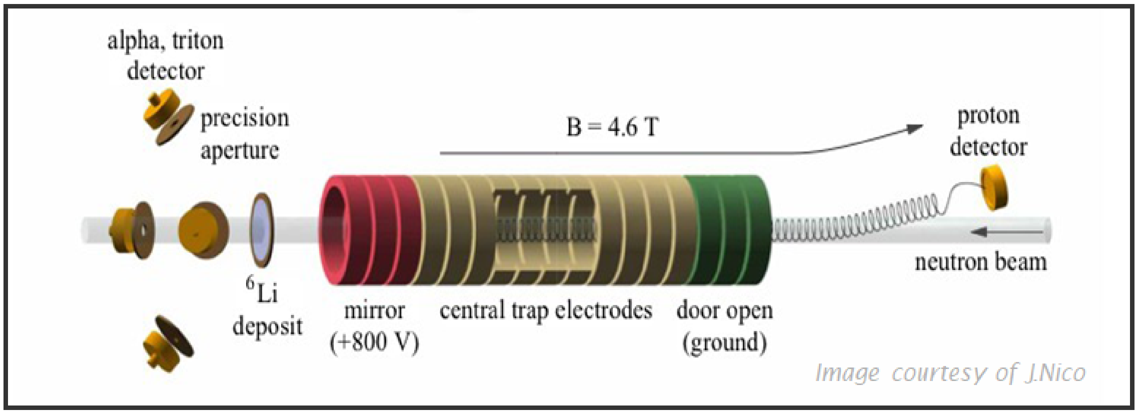

Figure 2.3: BL2 experimental diagram

Table 2.1: Pseudodata event types

0	Pure Noise
1	Single Proton Event
2	Double Proton Event
3	Triple Proton Event
4	Cosmic Ray Event
5	Electron Event
6	Electron + Proton Event
7	Cosmic + Proton Event
8	Untrapped Decay-in-Flight Proton Event

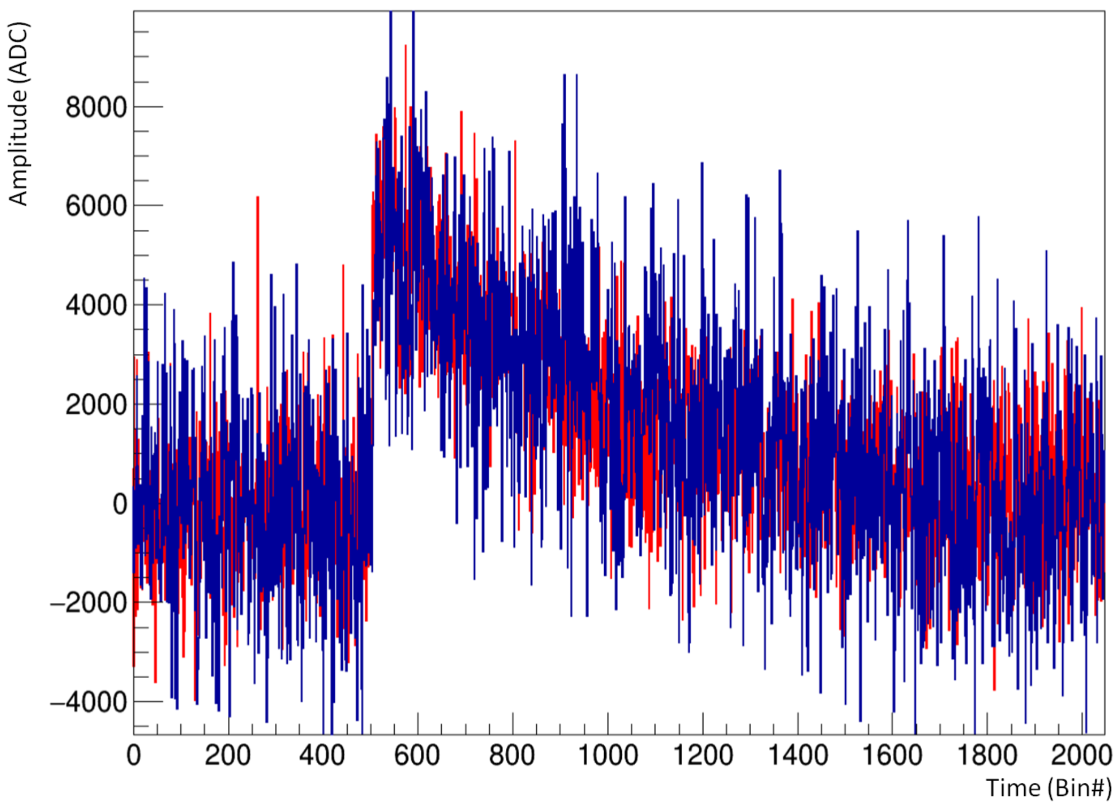

Figure 2.4: Comparison of pseudodata with real data from NIST BL2 neutron lifetime experiment

Chapter 3

Unsupervised Learning Analysis

3.1 Overview

The goal of any unsupervised learning study is to find underlying patterns in the data. While supervised learning requires a set of labeled data to train on, unsupervised methods allow one to explore an unlabeled data set with relatively minimal user input. This comes at the cost of reduced predictive ability and interpretability of the output when compared to supervised learning methods. A major strength of these methods versus traditional analysis techniques is allowing users a fast overview of the structure of the data, which can be used for diagnostic purposes or identifying anomalous events, especially when pseudodata is not readily available.

The first study applied unsupervised learning methods to energy spectrum data from the ^{45}Ca beta spectrum measurement at Los Alamos National Lab. This was an extension of a previous machine learning study on the same data set. By clustering the waveform data, oscillations in the baseline became immediately apparent, when previously they had taken years to notice. The goal of this study was to investigate more methods for data exploration to be used in this experiment and similar experiments in the future, such as the Nab experiment at Oak Ridge National Lab. In this way, any spurious or corrupt data could be identified more rapidly than traditional methods.

As stated in Chapter 2, it is important to understand the data features before applying any machine learning methods. Due to the varying energies of the electrons, any clustering attempt done on the raw data would sort similar energy events together. In order to remove this factor, all of the waveforms were normalized individually to be between -1 and 1. In this way, any waveforms with differing shapes should stand out as outliers. A first pass study of these waveforms using a K-means clustering method revealed the oscillating baseline effect. This method clusters data around multiple centroids, with the number of clusters chosen by the user. However, notable drawbacks of the K-means clustering method include: a reliance on the number of clusters being specified as an input to the algorithm, no allowance for outlying data points outside of the clusters, and an inability to effectively cluster abnormally shaped data sets. The high-dimensional nature of the data set does not lend itself well to clustering methods that rely on a distance metric, due to the commonly referenced "curse of dimensionality" [11]. These dimensions may also contain correlated data, which could further skew the clustering results. Therefore it was compelling to first apply a dimensionality reduction algorithm, which would both orthogonalize the data set and reduce the number of dimensions, and then cluster the data set with a density-based method, which would address all of the noted drawbacks for the K-means method.

3.2 Dimensionality Reduction

The chosen method for dimensionality reduction was Principal Component Analysis (PCA) [10], which produced an orthogonalized and compressed data set that can be easily visualized in lower dimensions. A Singular Value Decomposition (SVD) [17] is applied to the data set, which consists of 4250 normalized waveforms with 3500 time bins. The SVD decomposes the data matrix as shown in equation 3.1.

$$M = U \times S \times V^T \tag{3.1}$$

S is a diagonal matrix of singular values, and the transformed data P is given by:

$$P = U \times S = M \times V \tag{3.2}$$

The singular values are ordered from largest to smallest, and the square of the singular value gives the variance explained by the corresponding principal component. Using this information, one can produce a scree plot, which plots the singular values vs. their order, from largest to smallest. Searching for the "elbow" in this plot gives a good indicator to the number of dimensions to reduce to. Adding dimensions past the elbow will have increasingly diminishing returns. In this case, the elbow appears at four principal components, as shown in Figure 3.1. Since the square of the singular value gives the data variance explained by the corresponding principal component, a graph of cumulative variance explained vs. number of principal components can be made, which will show how much information is retained at each dimensionality cut. At two principal components, 82.89% of the original data variance is preserved, as shown in Figure 3.2. Although the scree plot shows the elbow at four dimensions, the first two dimensions contain the large majority of the information. During the clustering process, multiple different choices were tried for dimensionality cuts. The best results were found using two-dimensional data. Thus, the original data set was projected to two principal components using equation 3.2. It is then useful to make a 2D scatter plot with the first two principal components as the x and y axes. Each data point will correspond to a unique waveform. This will allow for easy visualization of the clusters as they are formed. Furthermore, each principal component can be plotted in the original waveform-space. This gives interesting physical insight as to what the principal components look like and how a linear combination of the principal components could produce a waveform that is similar to the original. The 2D scatter plot is shown in Figure 3.3, and the principal components are shown in Figures 3.4 and 3.5.

3.3 Clustering ^{45}Ca Beta Spectrum Data with DBSCAN

Once the data has been compressed, it is ready to be passed into the clustering algorithm. The chosen clustering algorithm for this study was DBSCAN (density-based spacial application of clustering with noise) [4]. It was implemented using the Scikit-Learn package in

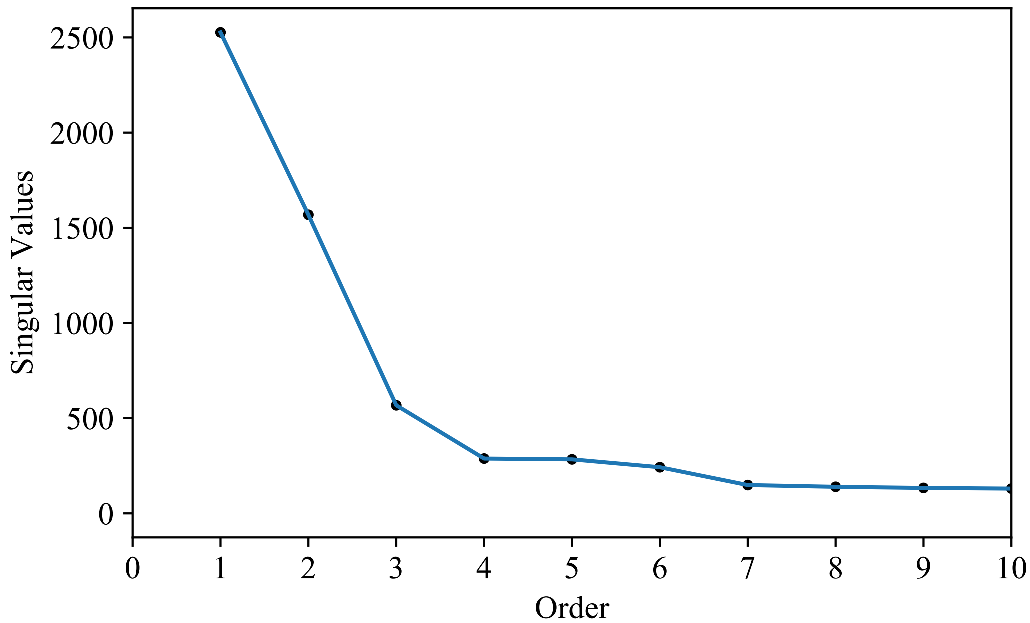

Figure 3.1: ^{45}Ca scree plot with elbow at 4 principal components

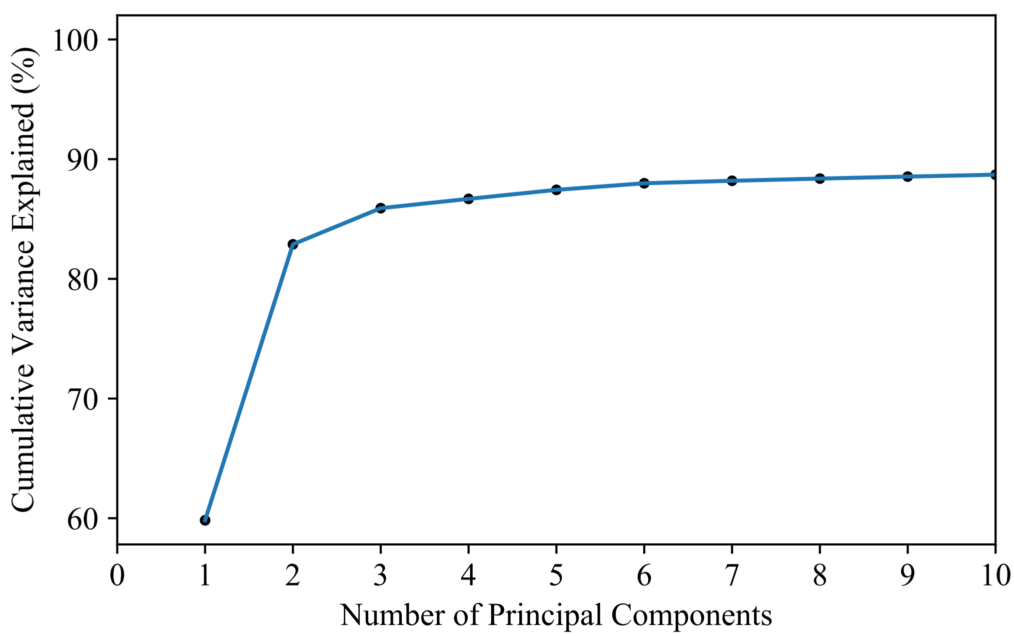

Figure 3.2: ^{45}Ca retained information after dimensionality reduction

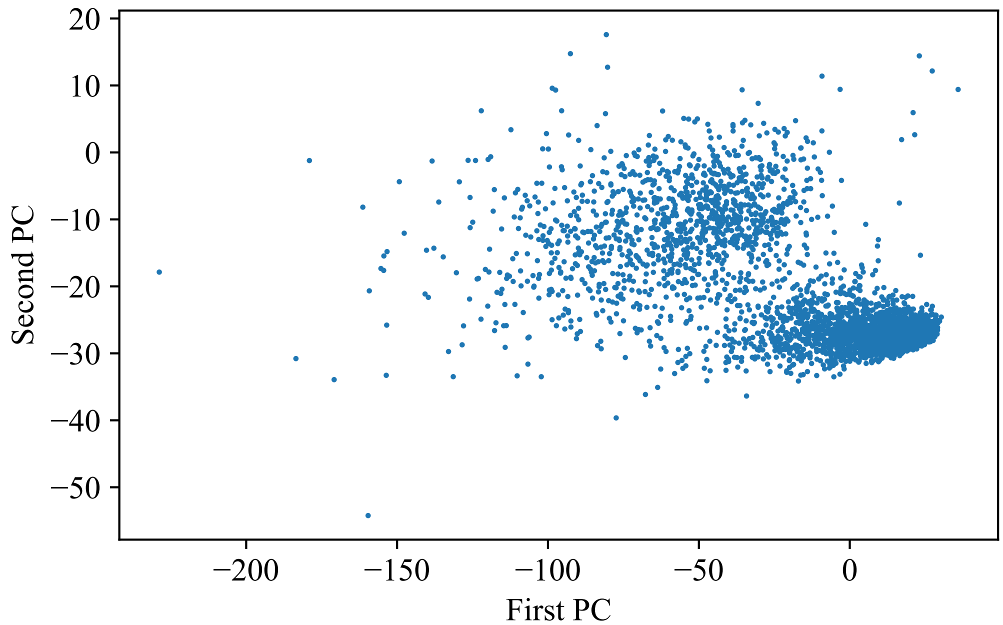

Figure 3.3: ^{45}Ca 2D scatter plot

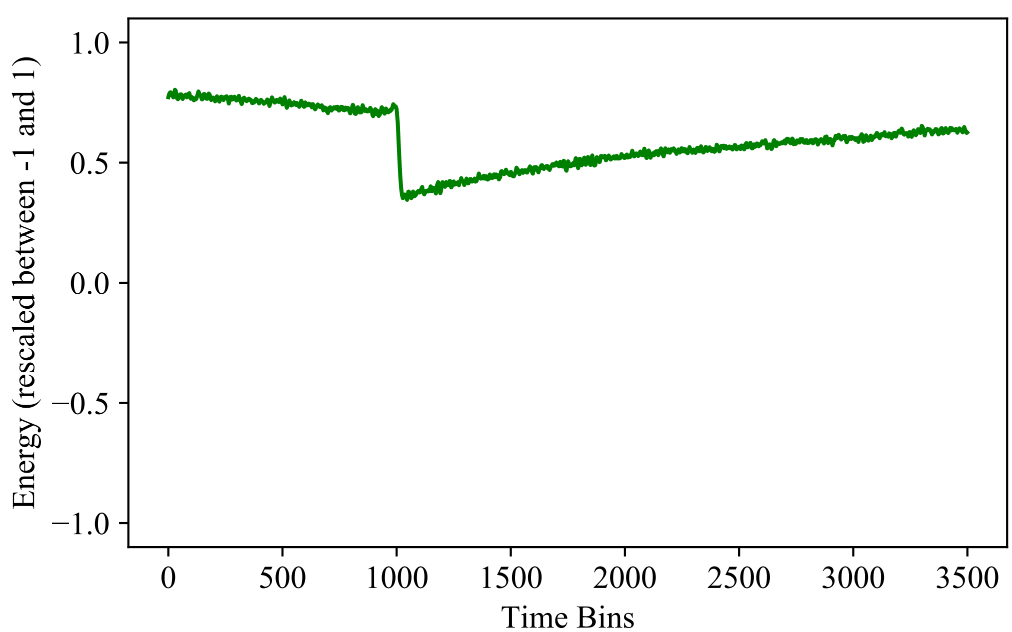

Figure 3.4: ^{45}Ca First principal component (note that while the individual waveforms were normalized such that the peak is at 1, this PC was formed from the normalized data set, and thus does not display the same shape. Instead, a linear combination of both this PC and the second PC can be used to approximate the original normalized waveforms.)

Python[15]. There are two required inputs to the algorithm (referred to as hyperparameters): the minimum number of points needed for a group to be a cluster (min_points), and the search distance to find points to add to the cluster (ϵ). Since ϵ is a distance metric, it is important to first define the distance metric being used. In this case, it is the normal Euclidean distance in two dimensions, given by equation 3.3.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2} \tag{3.3}$$

To find an optimal value for ϵ, the distance between each pair of points is found and plotted as a histogram. This is shown in Figure 3.6. The peak given in figure 3.7 gives a good starting point for ϵ values to try when running DBSCAN. After trying various values for ϵ and min_points, good clustering results were found at $\epsilon = 2$ and min_points $= 10$. Ten clusters were found, with 82.52% of the data existing in these clusters. To check if the clusters visually make sense, the 2D scatter plot can be color coded to show the clusters (shown in Figure 3.7). The contents of these clusters were then examined by averaging the waveforms in each cluster. This revealed multiple small clusters containing only noise data, but differentiated by the phase of oscillation in the baseline. The first cluster contained the majority of the data, which consisted of particle hits. Two smaller clusters of lower energy particle hits were also found. The second cluster contained the remaining clustered noise data. In Figures 3.8, 3.9, and 3.10 the colors of the averaged-waveforms correspond to the cluster colors in Figure 3.7.

Last to consider are the unclustered data points. Since they consist of many different types of waveforms, an average waveform does not give much information about the contents of the unclustered points. Manual inspection reveals that most of the particle hits off of t = 1000 were not clustered, as well as waveforms only showing a tail end. Multi-signal events were also unclustered, as well as many low energy particle hits (presumably because lower energy events have a lower signal-noise ratio, which makes the shape less defined). Figure 3.11 shows these events, with an off time hit (blue) in the top left, a multi-signal event (red) in the top right, a low energy event (gold) in the bottom left, and a tail end event (green) in the bottom left.

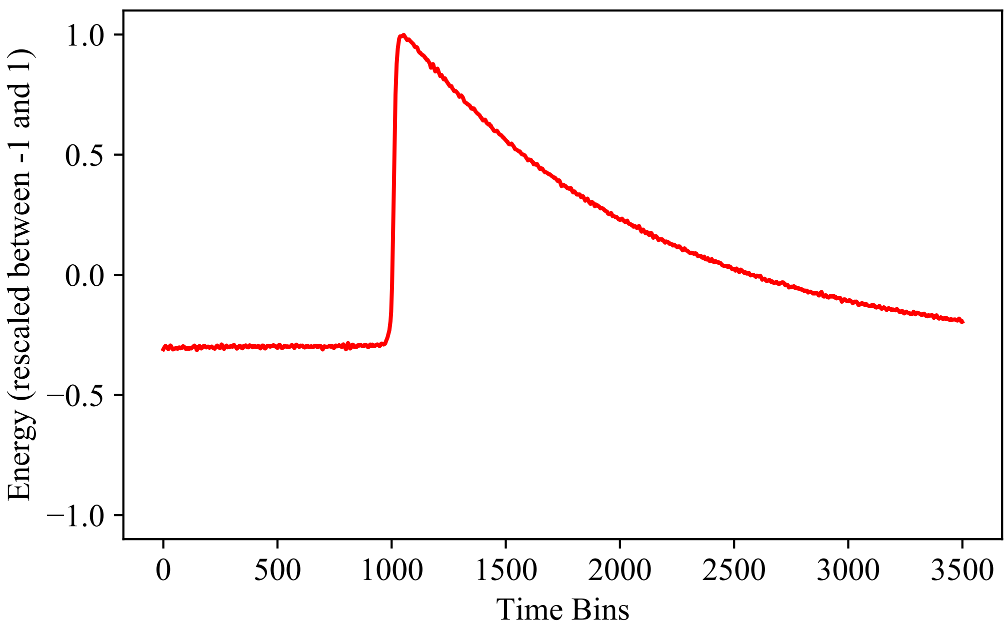

Figure 3.5: ^{45}Ca Second principal component

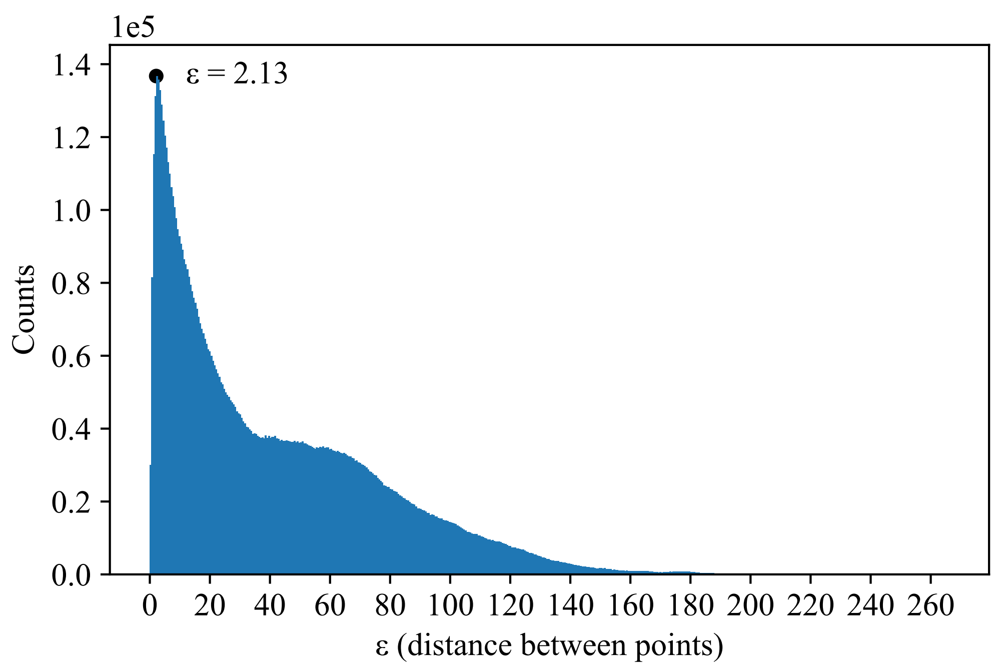

Figure 3.6: Histogram of point-to-point distances

3.4 Clustering ^{45}Ca Beta Spectrum Data with OPTICS

After clustering the data with DBSCAN, it was compelling to try an extension to DBSCAN known as OPTICS (ordering points to identify the clustering structure) [1]. This method allows for variable density data, and since Figure 3.6 displayed a range of distances between data points, it seemed to be a good application for this method. The algorithm works off the concept of "reachability," which is a distance that exists between every pair of data points. To define the reachability, the concept of a "core point" must first be defined. A core point is any point that has at least min_points within it's ϵ range. Similarly, the "core distance" is the distance from any point to the nearest core point. Then, the reachability of point a from point b is either the distance between a and b, or the core distance of a, whichever is larger. OPTICS calculates every reachability for every point, and annotates each point with its smallest reachability. As a drawback, calculating the reachability between each point in the data set leads to longer computation times than DBSCAN on average. A reachability plot can be produced, where the order by which the points were processed is the x-axis, and the reachability is the y-axis. This can be seen in Figure 3.12.

Clusters are then formed by looking for valleys in the reachability plot. Each peak represents a break and the start of a new cluster. The SciKit-Learn package includes the hyperparameter ξ which allows the user to define the slope steepness required for cluster differentiation. Alternatively, a single reachability value can be used as a cutoff, and when the trendline exceeds this cutoff value this marks a cluster break. However, this method simply reduces OPTICS back down to DBSCAN. Lastly, one can visually search for the cluster breaks and determine them manually.

The ξ method proved ineffective for this data set, as it would only find sparsely populated clusters with around ten or less events. Clustering with the visual method provided good results, although it is much slower to do manually and did not provide any new information when compared to DBSCAN. The color coded reachability plot can be seen in Figure 3.13, which is slightly zoomed in so that the cluster breaks are more apparent.

Once again a 2D plot can be made to visualize the clustering. As can be seen in Figure 3.14, these clusters contain more of the overall data when compared to the DBSCAN results.

Looking at the average waveforms of each cluster once again reveals some clusters in different phases of oscillation in the baseline. The first nine clusters showcase this effect, as seen in Figure 3.15.

Lastly, the average waveforms for the particle hit clusters can be shown, which are the last five clusters. Since these clusters were formed visually, some of them are imperfect and contain waveforms without particle hits. This distorts the average waveform when viewed. The average waveforms are shown in Figure 3.16.

Overall, the OPTICS method was equally as effective as DBSCAN, although significantly more time consuming due to the automatic cluster extraction methods not working well. The oscillation effects were easy to pick out from the formed clusters, and the good data could be isolated into a much smaller batch, albeit still with some bad waveforms (those lacking any particle hit). In its current state, OPTICS appears less favorable than DBSCAN, due to longer computing times plus manual cluster extraction and the already robust results of DBSCAN. However, other data sets may have larger variability in data density, in which case OPTICS would be more advantageous.

3.5 Conclusion

These studies allowed for detection of the known effect of the oscillating baseline in waveforms from the ^{45}Ca beta decay experiment. While this effect was already known, its detection required comparatively little work. Both DBSCAN and OPTICS algorithms were implemented in order to cluster this data, with roughly equivalent results, although OPTICS took considerably longer due to inherently longer computing times and manual extraction of the clusters. Clustering studies like these can be easily and quickly done many times throughout an experiment's life cycle, and can be applied to upcoming beta decay experiments such as the Nab experiment at Oak Ridge National Lab. They can provide early detection of unexpected effects or corrupt data, in addition to roughly categorizing events into distinct groups which can be useful to quickly extract a single event type.

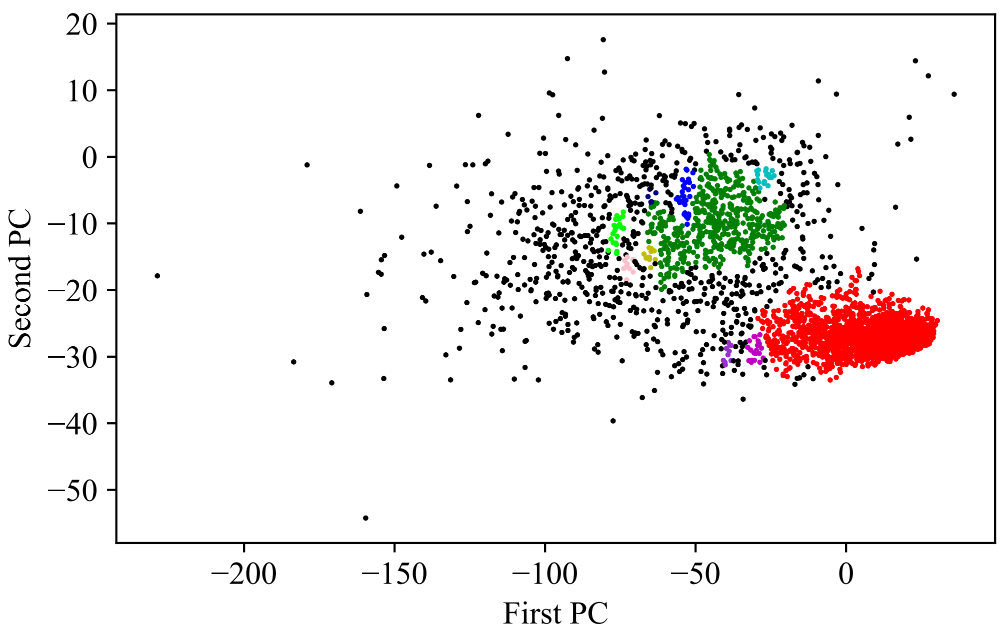

Figure 3.7: ^{45}Ca 2D scatter plot, color coded by clusters. Cluster colors correspond to colors of average waveforms shown in Figure 3.8, Figure 3.9, and Figure 3.10.

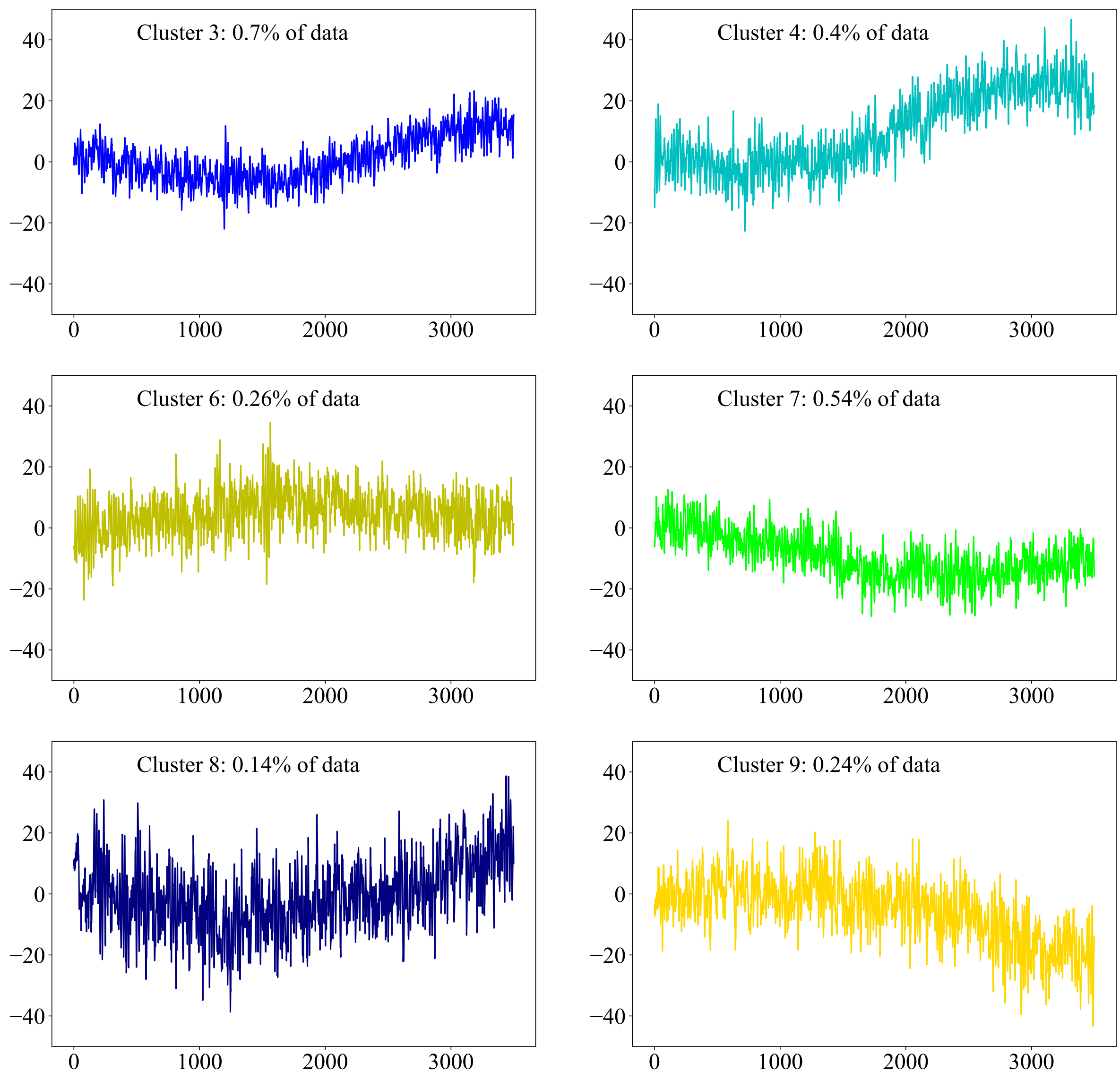

Figure 3.8: ^{45}Ca oscillations in baseline

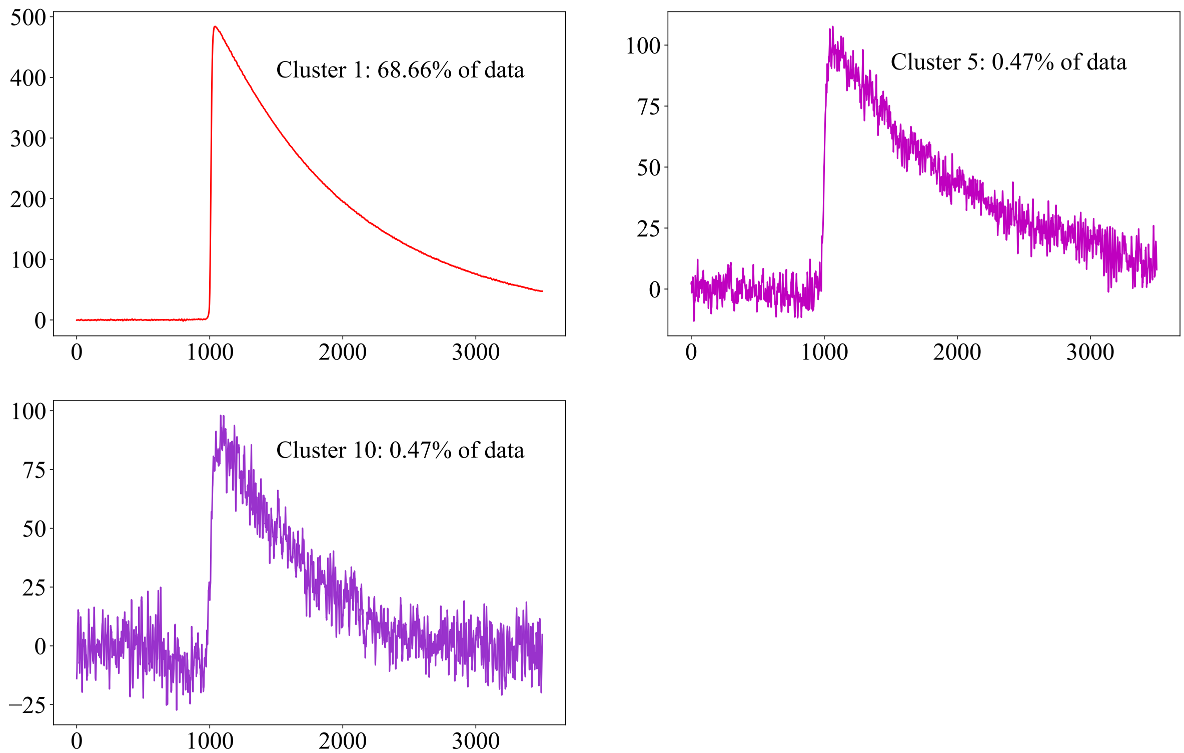

Figure 3.9: ^{45}Ca particle hits

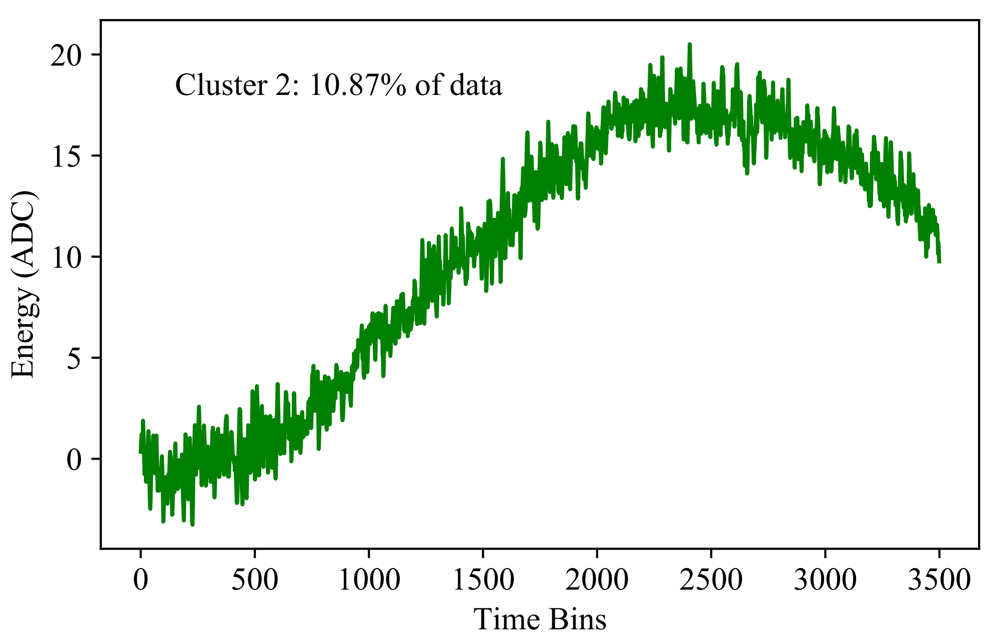

Figure 3.10: ^{45}Ca no particle hits

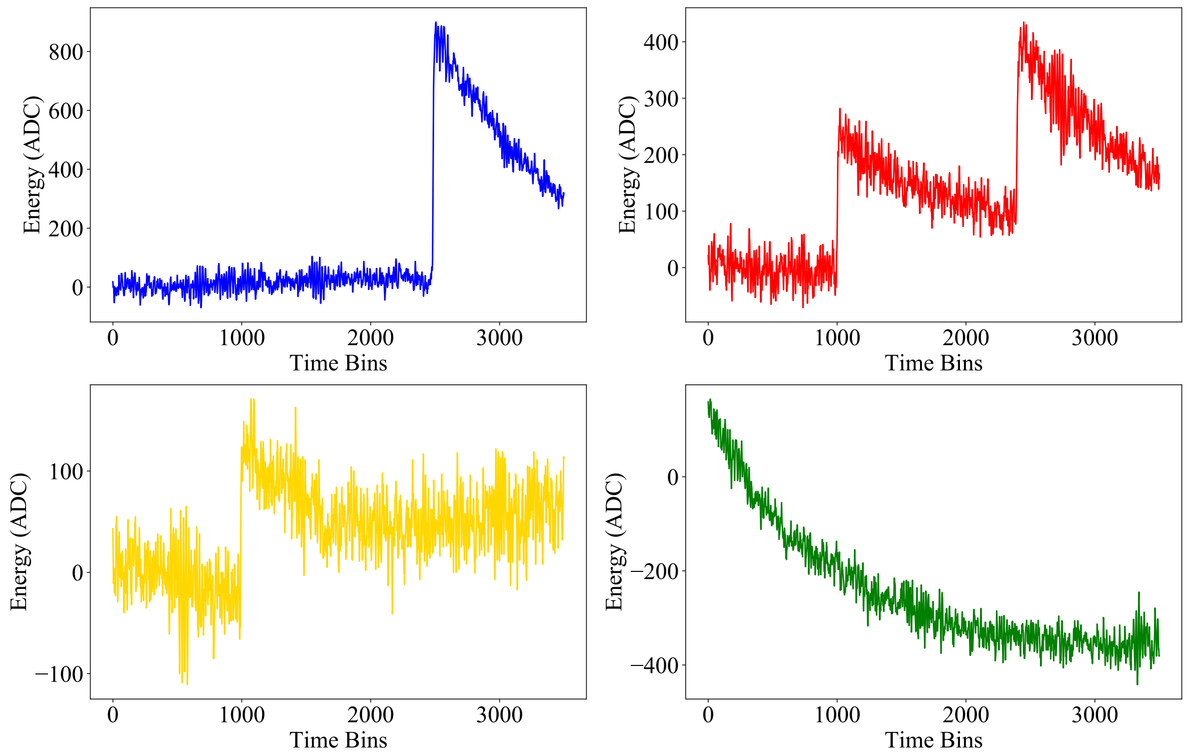

Figure 3.11: ^{45}Ca unclustered events (note that these waveform colors do not correspond to cluster colors. Unclustered events are represented by black data points).

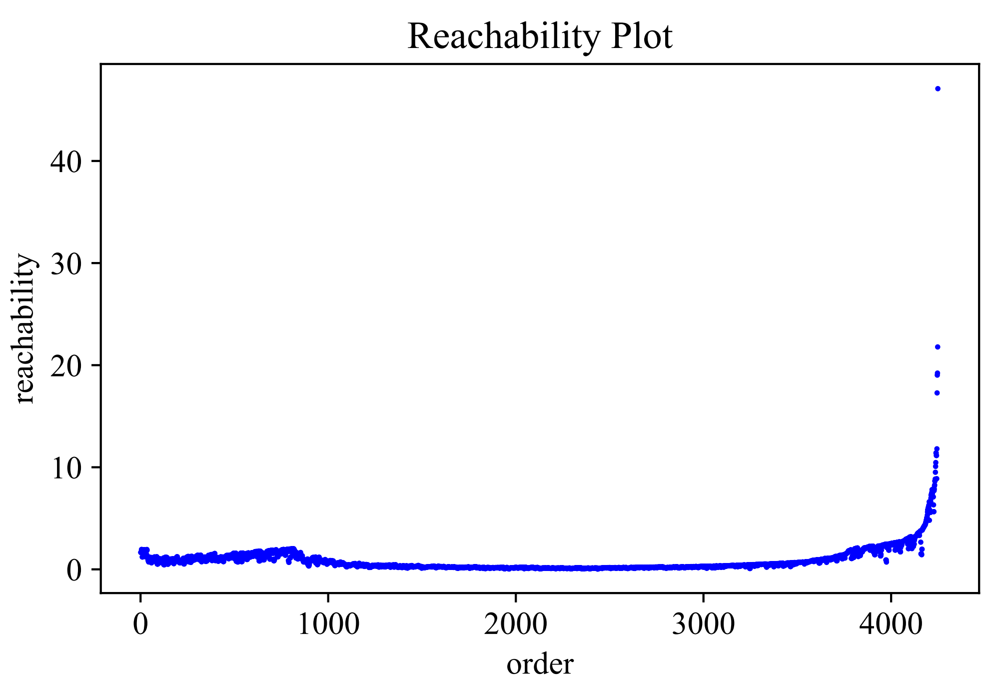

Figure 3.12: ^{45}Ca reachability plot

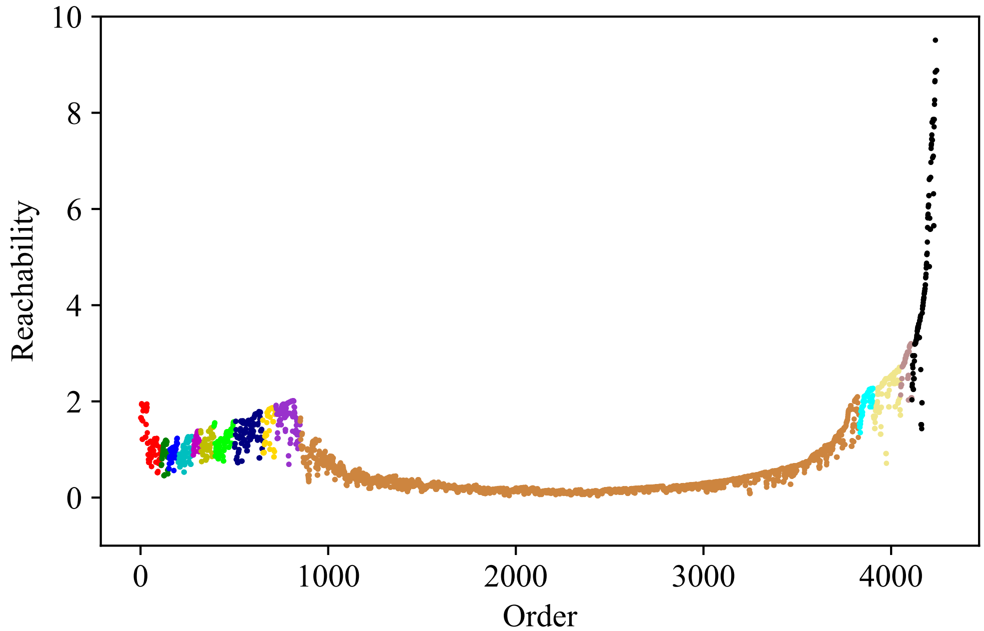

Figure 3.13: ^{45}Ca reachability plot color coded by cluster. Cluster colors correspond to average waveforms shown in Figure 3.15 and Figure 3.16.

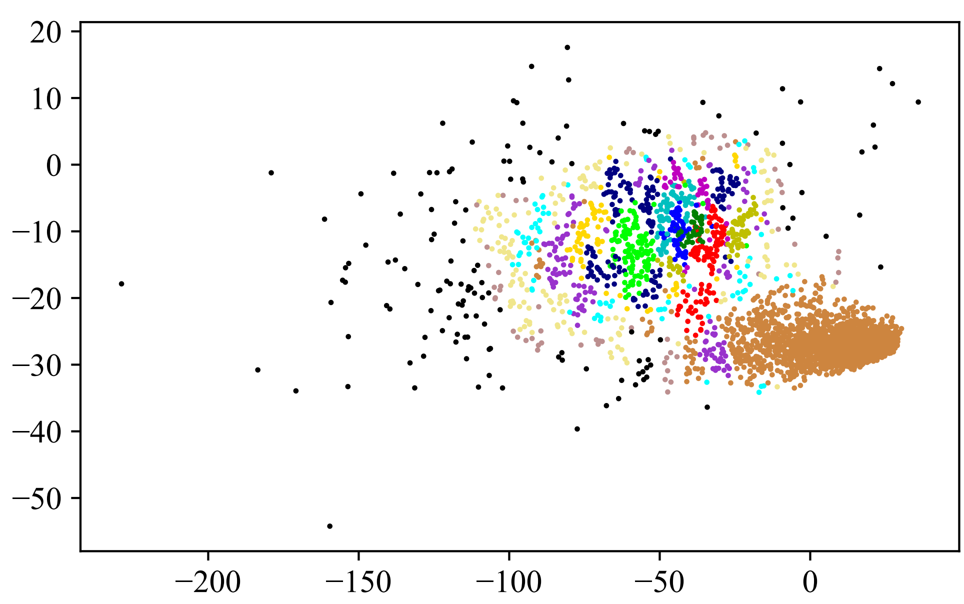

Figure 3.14: ^{45}Ca 2D plot color coded by clusters formed with OPTICS. Cluster colors correspond to average waveforms shown in Figure 3.15 and Figure 3.16.

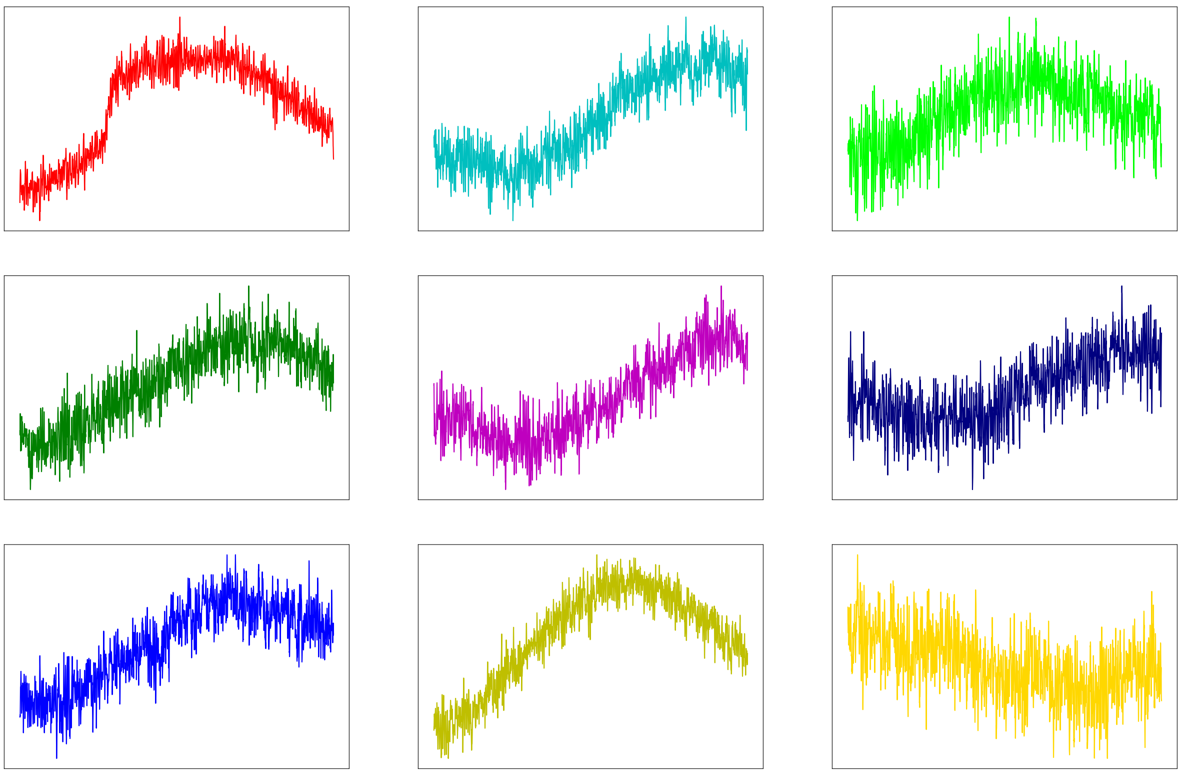

Figure 3.15: ^{45}Ca oscillations in baseline found by OPTICS

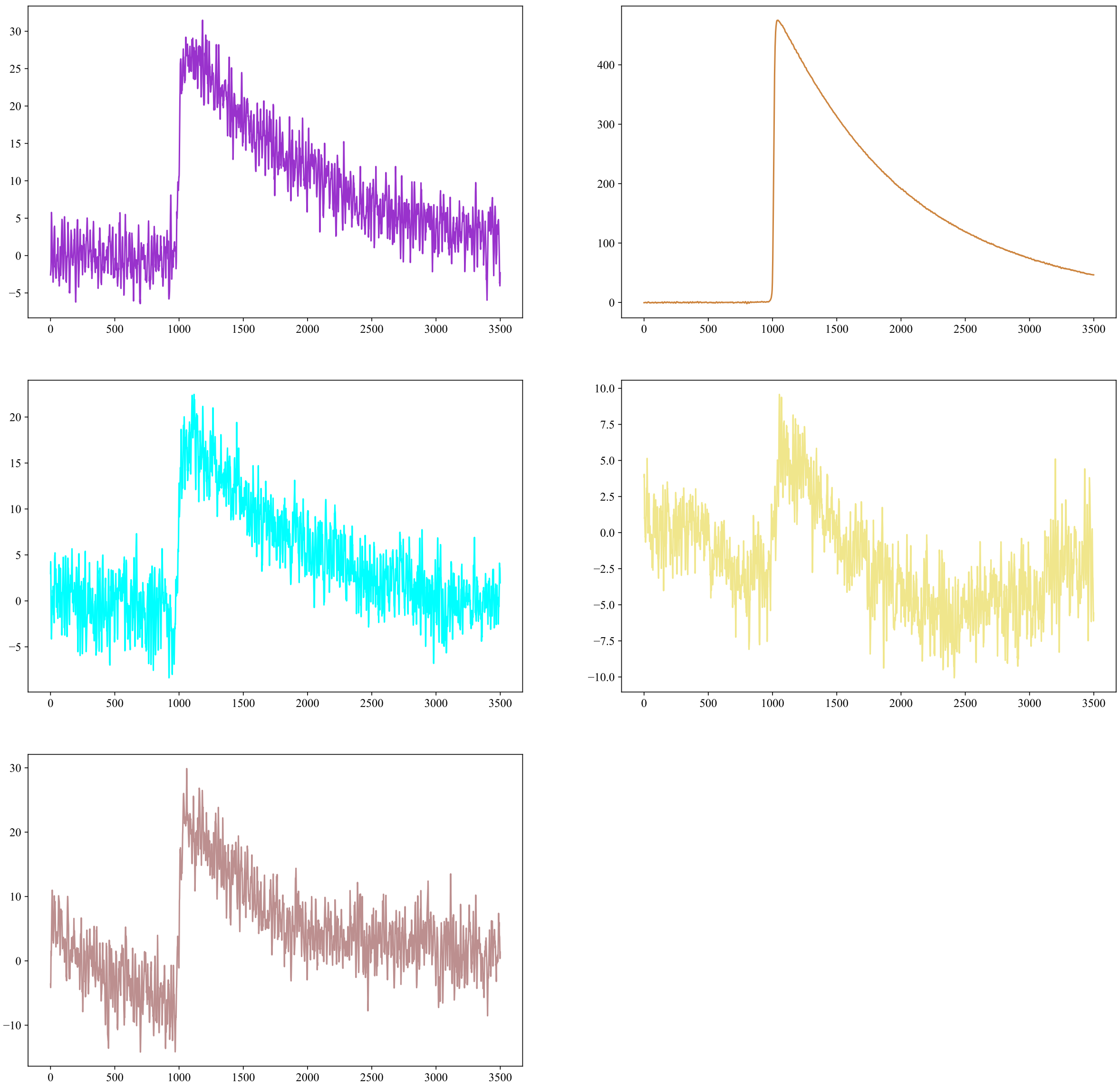

Figure 3.16: ^{45}Ca particle hits found by OPTICS

Chapter 4

Supervised Learning Analysis

4.1 Overview

In contrast to unsupervised learning, the goal of a supervised learning study is to make an informed prediction. This requires training an algorithm on previously generated and labeled data. This study made use of supervised learning to count proton events in data recorded for the neutron lifetime measurement at NIST using a cold neutron beam. This first required generation of labeled pseudodata that very accurately represented the real data. A predictive model was then built using a pseudodata set with an equal distribution of events, and tested on a separate pseudodata set with the same event distribution as the real BL2 data. After training and testing, a confusion matrix can be made which details the predictive ability of the algorithm by comparing the predicted labels to the actual labels.

An important part of every supervised learning study is the training-testing split. Data that is used to train an algorithm must be kept separate from data used to test or cross-validate an algorithm, in order to prevent bias or overfitting. The following study utilized 3-fold cross validation and a separate testing set that was held apart from the training data. The hyperparameters were optimized using grid searches, which allow many different algorithms to be trained with different combinations of hyperparameters. The best performing model was then chosen, and a finer tuned grid search was performed.

4.2 Random Forest Classification of NIST BL2 Neutron Lifetime Data

The algorithm chosen to build a predictive model for this data set was the SciKit-Learn[15] Random Forest Classifier (RFC) [7]. It operates as a 'forest' of decision trees, which consist of nodes with branching decisions. A question, formed by the algorithm, is asked about the data at each node. These questions are iteratively improved by maximizing some metric, such as information gain, entropy, or Gini impurity. A leaf node is reached when there are no more decisions to make, and thus the data can be classified. The two metrics used in this study were entropy and Gini impurity. The Gini impurity is given by equation 4.1, where C is the total classes and *P(i)* is the fraction of items labeled with class i in the set.

$$G = \sum_{i=1}^{C} P(i) \times (1 - P(i)) \tag{4.1}$$

The formula for entropy is given by equation 4.2, with C and *P(i)* defined similarly.

$$S = -\sum_{i=1}^{C} P(i) \times \log_2 P(i) \tag{4.2}$$

The decision tree then can predict a label for an event based on a series of questions asked by the algorithm. A random forest classifier considers the predictions of many decision trees, in order to reduce bias and get a more stable prediction.

This study implemented the Scikit-Learn package RFC. The hyperparameters to tune were:

- n_estimators: Number of trees in the forest
- max_depth: Maximum depth of a tree
- min_samples_split: Minimum number of samples required to split a node
- min_samples_leaf: Minimum number of samples required to be at a leaf node

- max_features: Number of features to consider when looking for a split. Allows for a specific value, $\sqrt{n\_features}$, or $\log_2(n\_features)$

- bootstrap: Chooses whether to use split the data set into 'bootstrap' instances and aggregate the results or train off the entire data set. 'True' corresponds to a split data set, while 'False' corresponds to training on the whole data set.

- criterion: Chooses the metric that measures the quality of a split (gini, entropy)

Multiple grid searches were performed to determine the optimal hyperparameters. These are reported in Table 4.1.

The trained algorithm was run on the test data set, and Table 4.2 reports the resulting confusion matrix. The diagonal matrix elements correspond to correct predictions by the model, while off-diagonal elements correspond to incorrect predictions.

This model's predictive accuracy was 99.39%, with most of the incorrect predictions being single proton events predicted as untrapped decay-in-flight proton events. This was expected, as these events appear identical to countable proton events. Since these are dealt with via background subtraction, this inaccuracy can be disregarded.

4.3 Conclusion

This study utilized pseudodata for the BL2 neutron lifetime measurement at NIST to generate a predictive model capable of particle identification for waveforms from the proton detector. A Random Forest Classifier was used to build this model, which had a predictive accuracy of 99.39% after being trained. While this accuracy is not yet high enough to replace traditional analysis, it can still provide use for experimenters by being used as a diagnostic tool to gain a live view of event distribution in the proton detector.

Table 4.1: Optimal Hyperparameters for Random Forest Classifier

Hyperparameter	**Value**
n_estimators	2000
max_depth	35
min_samples_split	4
min_samples_leaf	1
max_features	$\sqrt{n\_features}$
bootstrap	'False'
criterion	'Gini'

Table 4.2: Random Forest Confusion Matrix

Actual Label	Predicted Label								
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
0	2180	0	0	0	0	0	0	0	0
1	0	89520	2	0	0	0	37	0	482
2	0	2	5592	2	0	1	1	0	0
3	0	0	1	241	0	0	1	0	0
4	1	0	0	0	224	0	0	0	0
5	0	1	4	3	2	235	0	0	11
6	0	2	4	5	0	2	210	5	0
7	0	2	0	0	0	0	0	226	0
8	7	52	0	0	0	0	0	0	942

Chapter 5

Conclusion

5.1 Summary

The work done for this thesis used machine learning methods for multiple aspects of waveform analysis. The waveforms that were studied consist of data from both the ^{45}Ca Beta Spectrum Measurement and the NIST BL2 Neutron Lifetime Measurement, both of which contribute to current tests of the unitarity of the CKM matrix and consequently the search for beyond Standard Model physics. Clustering algorithms were applied to ^{45}Ca beta spectrum data as a method of data exploration. These algorithms, DBSCAN and OPTICS, were both able to pick out baseline oscillations in this waveform data. This effect was previously observed, but it had taken multiple years to notice via visual inspection of the data, while clustering the data provided immediate detection of the effect.

A predictive model was built for NIST BL2 neutron lifetime data that could classify events incident on the proton detector with an accuracy of 99.39%. This model was built using a Random Forest Classifier, which was trained and tested using labeled pseudodata. The trained model can be applied to online analysis to estimate event distribution in real time, which can be useful as a diagnostic tool.

5.2 Future Applications

These applications of machine learning algorithms to waveform data can serve as a proof of concept for upcoming beta decay experiments. Clustering studies like the ones shown in Chapter 3 can be used for the Nab experiment, which will take place at Oak Ridge National Lab. This experiment will make a measurement of a and b from Equation 1.5, and will detect outgoing protons and electrons from neutron decay. The waveforms from these decay products can then be clustered, and this can allow for much earlier detection of anomalous effects or corrupt data.

Future neutron lifetime measurements such as the BL3 experiment at NIST can also benefit from earlier implementation of machine learning methods. Alternative algorithms such as boosted random forests or neural networks may offer slight increases in predictive accuracy compared to the RFC used in Chapter 4. Biasing the model such that the misclassifications only fall in the upper or lower triangle of the confusion matrix (as in Table 4.2) could allow the experimental data to be reduced to a smaller data set before applying traditional analysis methods, potentially reducing the workload.

Bibliography

[1] Ankerst, M. et al. (1999). OPTICS: Ordering points to identify the clustering structure. pages 49–60. ACM Press. 23

[2] Birge, N. (2019). *Measurement of the Fierz Interference Term for Calcium-45*. PhD thesis, University of Tennesseee, Knoxville, TN. 5

[3] Broussard, L. J. et al. (2017). Detection System for Neutron β Decay Correlations in the UCNB and Nab experiments. *Nucl. Instrum. Meth. A*, 849:83–93. 10

[4] Ester, M. et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press. 18

[5] Fomin, N. (2020). BL3: Next generation beam experiment to measure the neutron lifetime. x, 7, 8

[6] Gonzalez, F. M. et al. (2021). An improved neutron lifetime measurement with UCNτ. 5

[7] Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1. 33

[8] Hoogerheide, S. F. et al. (2019). Progress on the BL2 beam measurement of the neutron lifetime. *EPJ Web Conf.*, 219:03002. 11

[9] Jackson, J. D., Treiman, S. B., and Wyld, H. W. (1957). Possible Tests of Time Reversal Invariance in Beta Decay. *Physical Review C*, 106(3):517 – 521. 4

[10] Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202. 17

[11] Keogh, E. and Mueen, A. (2017). *Curse of Dimensionality*, pages 314–315. Springer US, Boston, MA. 17

[12] MissMJ and Cush (2019). Standard model of elementary particles. `https://commons.wikimedia.org/wiki/File%3aLambdaPlaques.jpg"`. x, 2

[13] Nico, J. S. et al. (2005). Measurement of the neutron lifetime by counting trapped protons in a cold neutron beam. *Phys. Rev. C*, 71:055502. 11

[14] Pattie, R. W. et al. (2018). Measurement of the neutron lifetime using a magneto-gravitational trap and in situ detection. *Science*, 360(6389):627–632. 6

[15] Pedregosa, F. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. 21, 33

[16] Serebrov, A. P. et al. (2018). Neutron lifetime measurements with a large gravitational trap for ultracold neutrons. *Phys. Rev. C*, 97(5):055503. 5

[17] Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566. 17

[18] Yue, A. T. et al. (2013). Improved determination of the neutron lifetime. *Physics Review Letters*, 111(22):222501. 5

[19] Zyla, P. A. et al. (2020). Review of Particle Physics. *Progress of Theoretical and Experimental Physics*, 2020(8). 083C01. 3, 6

Vita

Micah Roy Cruz was born in Augusta, Georgia, and raised in Las Vegas, Nevada. He also lived in both Knoxville, Tennessee and Kennewick, Washington, before moving back to Knoxville to finish high school. During his final year of high school, his interest in mathematics dramatically increased as he studied calculus. He went on to attend the University of Tennessee, Knoxville, as an undergraduate, where he pursued a Bachelor's of Science in academic physics. After graduating in 2017, he returned as a graduate student to work with Dr. Nadia Fomin in the Fundamental Neutron Physics Group. Within this group, he was introduced to machine learning methods, which he found intriguing as they were a relatively unexplored tool in nuclear physics. He pursued a Master's of Science with the Fundamental Neutron Physics Group by studying applications of machine learning methods to neutron beta decay waveform data.