ANOMALY DETECTION AND PULSE SIMULATION FOR DIRECT DARK MATTER SEARCHES

A thesis submitted
in partial fulfilment of the requirements
for the degree of

MASTER OF SCIENCE
April 28, 2022

Aman Upadhyay
Roll: 1711017
Supervisor: Prof. Bedangadas Mohanty

to the
School of Physical Sciences
National Institute of Science Education and Research, HBNI

Declaration

Signature of the Student
Date:

The thesis work reported in the thesis entitled Anomaly detection and pulse simulation for direct dark matter searches was carried out under my supervision, in the school of physical sciences at NISER, Bhubaneswar, India.

Signature of the Supervisor
Date:

Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. I give special gratitude to my supervisor, Dr. Bedangadas Mohanty, whose contribution to stimulating suggestions and encouragement, helped me to complete my project.

Furthermore, I would also like to acknowledge with much appreciation the crucial role of the Dr. Varchaswi Kashyap and Dr. Ranbir Singh who helped me to use all the required equipment and the necessary materials to complete all the computational tasks. A special thanks go to Mouli Chaudhuri., who helped me to assemble the parts and gave suggestions about the project and this report. Last but not least, I am very grateful to my lab mates for their support and longanimity. Without their cooperation, completion of this project was difficult. I am thankful to my lab mates Danush S, Samir Banik, Dukhishyam Mallick, Ashish Pandav, Debasish Mallick, Vijay Iyer, Swati Saha, Prottay Das, Sudipta Das and Abhishek Deshmukh for helping me and creating a nice working atmosphere in the lab. I finally thank my parents and my friends for their support.

Abstract

This project has two parts. First, we discuss the application of an anomaly detection technique for data analysis of rare event search experiments. We will use a combination of t-SNE and DBSCAN algorithms to detect anomalies in rare event search data. Second, to build a simulation framework to generate raw pulses from particle interactions in semiconductor detectors used in direct dark matter searches. The simulations for the detector response will allow us to build better background models, that can be used to train and implement various machine learning techniques for future studies.

Contents

List of Figures

Page

List of Tables

Listings

Chapter 1

Introduction

Several astrophysical observations and surveys [1, 2] reveal that the total mass-energy budget of the universe is divided as follows: baryons and leptons make up about 5% of the total, dark energy makes up about 69%, and dark matter makes up the remaining 26 %[2][3]. The standard model accounts for the baryons, leptons, and bosons which constitute only 5% that makes 95% of the mass-energy budget of the universe unknown.

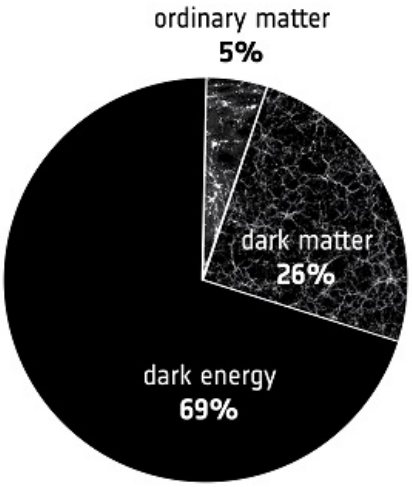

Figure 1.1: A pie chart showing the mass-energy budget of the universe.[4]

1.1 Evidence for the existence of Dark Matter

Fritz Zwicky in 1933, observed that the velocities of galaxies bound within the Coma Cluster exceeded those predicted by gravitation from visible objects[5]. This discrepancy is known as the "missing mass problem" and Zwicky coined the term dark matter for this missing mass. This section will discuss some evidence that comes from various astrophysical sources for the existence of dark matter.

1.1.1 Galaxy rotation curves

A galaxy is a collection of gas, dust, stars, and their star systems, all held together by gravity. Newtonian dynamics are still valid to understand the rotational dynamics of a spiral galaxy. The rotational velocity of an object in a spiral galaxy can be calculated using gravitational force and centripetal force. For an object, the rotational velocity v in the galactic disk at a distance r from the center of a galaxy with mass distribution $M(r)$ and G is the gravitational constant is given as,

$$v = \sqrt{\frac{GM(r)}{r}} \tag{1.1}$$

The formula predicts that the velocity should fall as $\frac{1}{\sqrt{r}}$ but the observed rotational velocities of galaxies were observed to be constant for values of r even beyond the luminous edge of the galaxies[6].

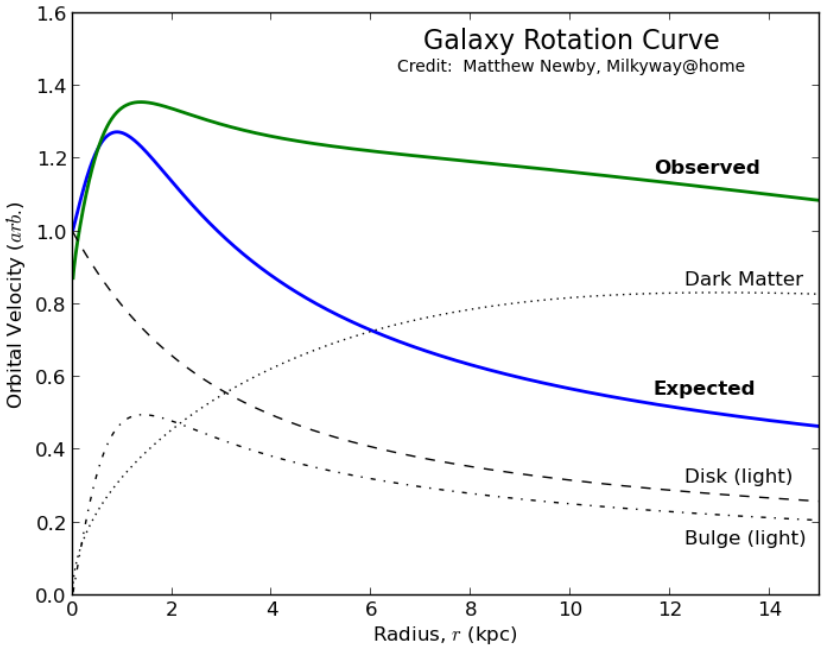

Figure 1.2: Difference between the observed and expected rotational velocity of an object vs distance from the center of the galaxy [7]

This behavior suggests that we are missing some mass in our calculations which is surrounding the galaxy and this mass is from a non-radiative source.

1.1.2 Gravitational lensing

According to Einstein's general theory of relativity any object with mass will wrap the space-time. He also predicted that this distortion can create a lens-like effect to an observer when light passes through space-time around a massive mass.

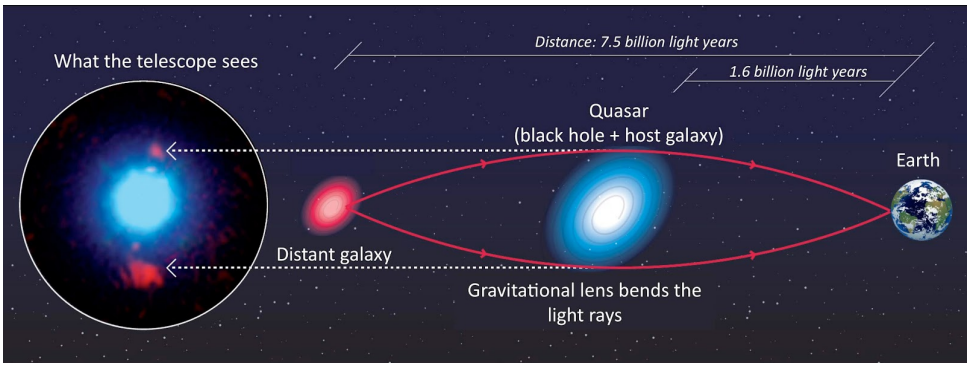

Figure 1.3: A schematic of the gravitational lensing effect due to a massive object.[8]

Fig: 1.3 gives a schematic of the bending of light around a galaxy and how it will look to an observer on the other side of the galaxy. Astronomers tried to deduce the mass of the galaxies using gravitational lensing and they consistently reported an excess in the masses of the galaxies when compared to the masses they got from their luminosity[9].

1.2 Search strategies for dark matter detection

Based on the observations we have on the dark matter we can infer some properties of dark matter. Dark matter is not observable via any of the telescopes we have, otherwise, we would have accounted for it in the galaxy rotation curve. This tells us that dark matter does not have any electromagnetic interactions. However, we know that they interact through gravitational force as they can be mapped by gravitational lensing. Other observations from large-scale structure formations suggest that dark matter is non-relativistic and stable.

Table 1.1: Properties of dark matter we know

Properties
No color charge
No electric charge
Non-relativistic
Stable on cosmological time scales
Interacts through gravity
Almost collisonless

One of the candidates for dark matter is Weekly Interacting Massive Particles (WIMPs). They are derived from supersymmetry theory (SUSY). WIMPs are massive and weakly interacting. There are 3 possible ways of detecting dark matter:

1. Indirect detection: When we detect standard model particles that are produced when dark matter particles annihilate with their anti-particles.

2. Direct detection: We directly detect dark matter when they scatter off Standard Model

particles, producing a signal that can be detected with sensitive detectors here on earth.

3. Collider production: In collider experiments, We try to create dark matter particles by colliding standard model particles at extremely high energies.

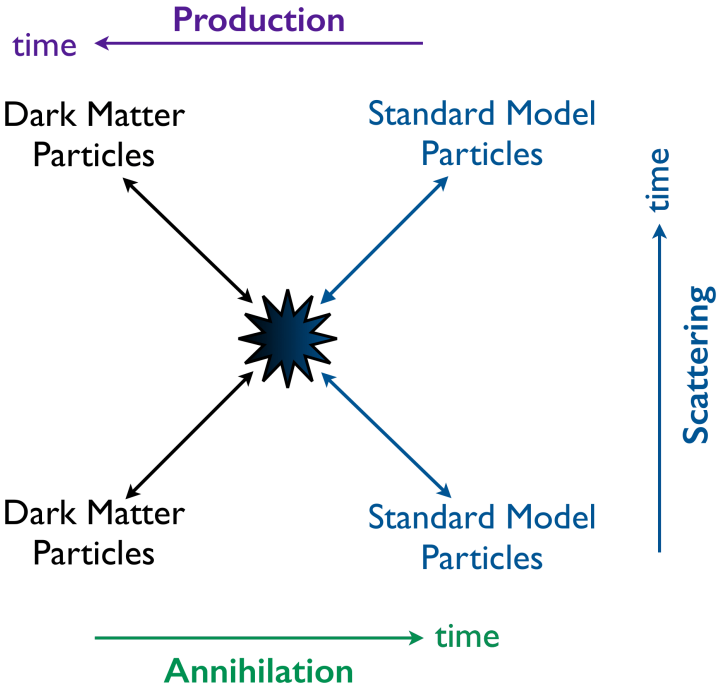

Figure 1.4: A diagram showing possible dark matter detection methods[10] .

1.3 Current status of direct dark matter searches

Multiple direct dark matter experiments have tried to detect dark matter. Some of the experiments are SuperCDMS, DAMIC, LUX, and CRESST[11]. For the detection of low mass dark matter candidates like WIMP, we use silicon and germanium in our detectors. These experiments are situated deep underground to shield from cosmic rays and other precautions are taken to shield all possible backgrounds. As the experiments get more sensitive they will reach a point where neutrinos will scatter off the detector nucleus. This will be a problem as neutrinos can not be shielded and would create an irreducible background. Fig: 1.5 shows an exclusion curve for various experiments searching for WIMPs.

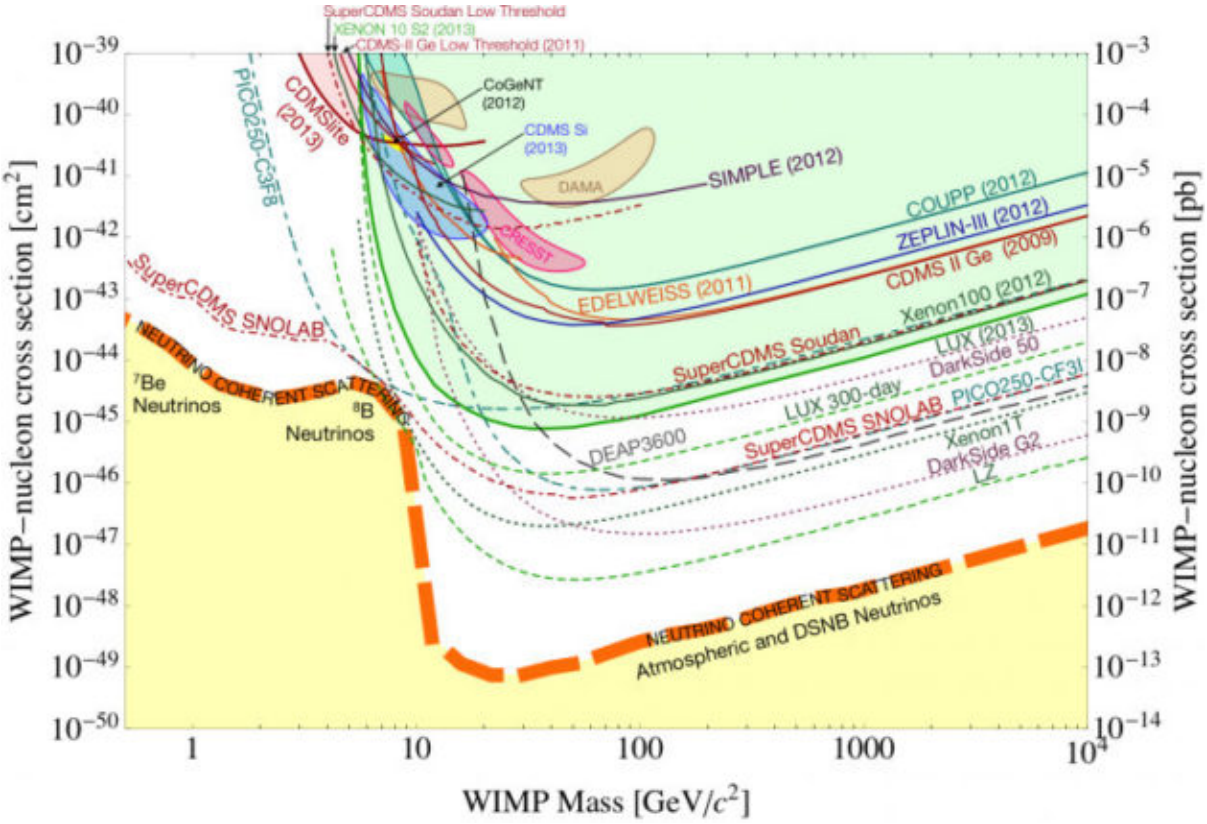

Figure 1.5: Results from various dark matter search experiments are shown as dark matter - nucleon cross-section vs dark matter mass. The region above a solid curve corresponding to an experiment has been excluded by it. The region below the solid curves is yet to be explored. The shaded region below the thick dotted line at the bottom of the plot corresponds to neutrino as an irreducible background [12]

Aim of this project: This project is divided into two parts: First is the application of anomaly detection techniques in data analysis of rare event search experiments. Anomaly detection (AD) is a data analysis technique that identifies data points that deviate from a dataset's normal behavior. Application of anomaly detection is being investigated by experiments like CMS and SuperCDMS. t-SNE and DBSCAN are techniques that can be used together to detect anomalies. These are unsupervised methods i.e. they don't need labeled data.

The second is to build a simulation framework to generate raw pulses from particle interactions in semiconductor detectors used in direct dark matter searches. The goal is to use the simulations for understanding the detector response to optimize an analysis. It will also help us to build a better background model which can be used to train and implement various machine learning techniques for our analysis.

Chapter 2

Detector response to particle interaction

Experiments like Super Cryogenic Dark Matter Search Soudan experiment (SuperCDMS Soudan) and Mitchell Institute Neutrino Experiment at Reactor (MINER) will use semiconductor detectors. SuperCDMS SNOLAB is designed to search for dark matter particles with masses $\lessapprox 10\text{GeV/c}^2$. [13]. It will be located approximately 2 km underground in Sudbury. MINER is a reactor based experiment that uses cryogenic detectors similar to those of SuperCDMS dark matter search. It is also developing new sapphire and veto detectors. It aims to detect coherent scattering of low energy neutrinos.

Coherent elastic neutrino-nucleus scattering (CEνNS) is a process in which a neutrino scatters from a nucleus by exchanging an electrically neutral Z boson. Coherent means that in the scattering process the neutrino interacts with the nucleus as a whole, and not with individual nucleons. The process is elastic and kinetic energy is conserved. But still the scattering cross-section is less compared to other standard model interactions. CEνNS and WIMPs both interact with a standard model atom through nuclear recoil and we can use this fact to discriminate other background events which interact through electron recoil when searching for WIMPs or CEνNS.

I will be discussing a germanium semiconductor dark matter detector. The detector is built of a single cylindrical crystal of germanium. Each detector has a charge and phonon readout system both at the top and bottom of the detector. Charge electrodes at the top and bottom of the detector generate an electric field in the detector as shown in Fig: 2.1[14].

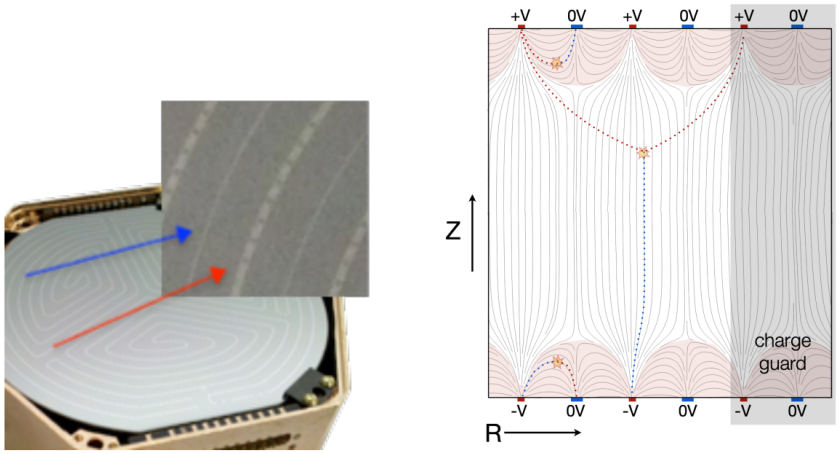

Figure 2.1: Left: Zoomed view of electrodes and phonon sensor arrays. Right: Detector divided into 4 channels A, B, C and D.[14]

The detectors are sensitive to charge and phonons, charges are collected using a FET (field effect transistor) sensor while phonons are collected using a TES (transition-edge sensors). The working principle of these two sensors will be discussed in section 2.5. The detector is divided into 4 channels A, B, C, and D. Each channel is an array of collection sensors, each channel produces one signal.

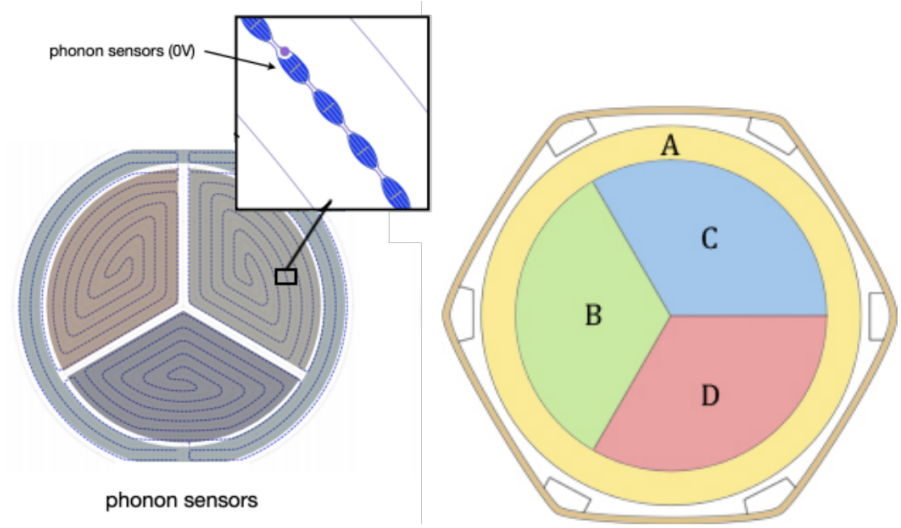

Figure 2.2: Left: Lines containing TES sensors. Right: Cross-section diagram of the electric field generated in the crystal by electrodes. In red rectangles, charge electrodes are shown and phonon sensors in blue[14]

2.1 Overview of the background sources

There are two types of background sources we need to keep an eye out for, first that come from near the detector or from within the detector itself they are the radiogenic backgrounds. Second, come from cosmic sources they are the cosmogenic backgrounds. Most of the background is blocked by the shielding that surrounds the detector but others need to be identified and

rejected in the analysis. In general, particle interactions are expected to scatter with either an electron or a nucleus in the detector we will refer to these two types as electron recoil and nucleus recoil[15][16]. The interaction type depends on the incoming particle, different type of backgrounds with their interaction type is listed in Table: 2.1. The total recoil energy depends on the kinematics of the interaction but each interaction releases two types of energy into the crystal of the detector: charge energy (from ionization of electrons in the crystal), and phonon energy (from vibrations of nuclei in the lattice). The mechanism of energy release for each type, and how much goes into each will be described in the next section. but the ratio of energy differs significantly between electron and nuclear recoil and allows us to discriminate between the two. Thus, the ionization yield, defined as the ratio of the measured charge and phonon recoil energy, creates a clearly defined method for identifying background-like (electron) from signal-like (nuclear) interactions.

Table 2.1: Summary of the background sources, what type of interaction they have with the detector

Cosmogenic Backgrounds	
Background Source	Recoil Type
Electrons and Photons	Electron Recoil
Muons	Electron Recoil
Neutrons	Nuclear Recoil
Neutrinos	Nuclear Recoil
Radiogenic Backgrounds	
Background Source	Recoil Type
Ge Activation	Electron Recoil
Pb Implantation	Electron Recoil and Nuclear Recoil

2.2 Energy deposition in crystal

We will discuss the detector response from the particle interaction up until the waveform is produced in the sensors. For every particle interaction in the detector, energy is deposited in the crystal. The total energy deposited in the detector comes from two processes: first from the incident particle and second as the charges accelerate in the crystal lattice due to the bias voltage across the detector. The energy deposited into the crystal from the incident particle is known as recoil energy (E_r). E_r is partially deposited into crystal lattice vibrations (phonons) and ionizing electrons.

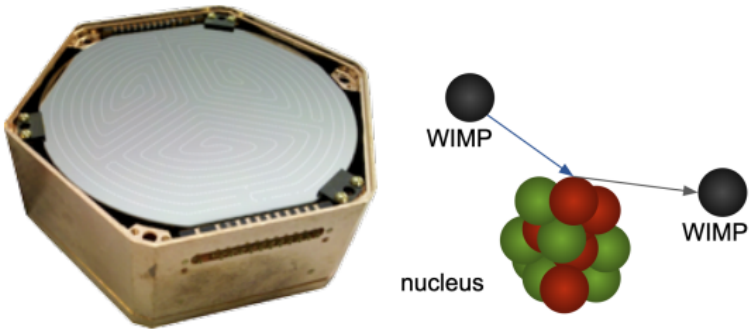

Figure 2.3: Ge detector used in SuperCDMS experiment and a cartoon of WIMP interacting with a nucleus in the germanium crystal[14]

After the initial interactions, phonons start moving in the direction based on their phase velocities. While the electron and holes start drifting in the crystal due to the bias voltage, this introduces a secondary source of energy in the crystal.

2.2.1 Primary energy

Recoil energy E_r, is transferred to the crystal in two systems: phonon system and charge system. Hence, E_r is divided into two energies $E_{pho,primary}$ Primary phonon energy and E_Q charge energy.

$$E_r = E_{pho,primary} + E_Q \tag{2.1}$$

The fraction of energy that goes in $E_{pho,primary}$ and E_Q is governed by expected yield T_{exp}. The expected yield is given by:

$$Y_{exp} = \frac{E_Q}{E_r} \tag{2.2}$$

Value of Y_{exp} depends if an interaction was an electron recoil or nuclear recoil. Since electron recoil does not produce primary phonons, all the recoil energy is converted into E_Q and $Y_{exp} = 1$. On the other hand, nuclear recoil goes through a complex process, and the energy is converted into both E_Q and $E_{pho,primary}$. In this case, Y_{exp} is given by Lindhard theory [17][18] which is described by condensed matter physics and is known as $Y_{Lindhard}$. So:

$$Y_{exp} = \begin{cases} Y_{Lindhard}, & \text{nuclear recoil} \\ 1, & \text{electron recoil} \end{cases} \tag{2.3}$$

According to the Lindhard theory, the value $Y_{Lindhard}$ only depends on the crystal material property (atomic number Z and mass A), E_r, and the stopping mechanism. Y_{Lindhard} is given by:

$$Y_{\text{Lindhard}}(E_r) = \frac{k \cdot g(\epsilon)}{1 + k \cdot g(\epsilon)} \tag{2.4}$$

where $\epsilon = 0.0115 E_r Z^{(-7/3)}$ [17] (the constant also takes into account the atomic mass of Ge) describes the amount of energy deposited per proton. $k = 0.133 Z^{(2/3)} A^{(-1/2)}$ describes the

amount of energy per electron/hole pair. And $g(\epsilon) = 3\epsilon^{(0.15)} + 0.7\epsilon^{(0.6)} + \epsilon$ [17] [18] describes the number of collisions with electrons in the crystal. We can fix the atomic mass and atomic number of the detector to get Y_{Lindhard}.

2.3 Number of phonon and electron-hole pairs produced

Debye model defines the maximum allowed phonon energy in any given material. This is denoted by E_{Debye} and depends on material density and temperature. For Ge at mK temperatures it has a value of 8.1 eV [19, 20, 21]. We can approximate that all phonons but one have the maximum allowed energy, $N_{pho,primary}$ is:

$$N_{pho,primary} = \text{Quotient}(\frac{E_{pho,primary}}{E_{Debye}}) + 1 \tag{2.5}$$

this gives us the maximum number of phonons possible with E_{Debye} energy plus one with the residual energy.

The total number of e/h pairs can be calculated using ionization energy ϵ_γ. The number of e/h $N_{e/h}$ pair is the number of the maximum number of electrons ionized by E_Q and any additional energy is lost to the crystal and can be neglected. For Ge $\epsilon_\gamma = 2.96eV$.

$$N_{e/h} = \text{Quotient}(\frac{E_Q}{\epsilon_\gamma}) \tag{2.6}$$

2.4 Total expected energy value

As electron and hole drift through the detector under the electric field produced by the bias voltage, they interact with the lattice structure in a process called Neganov-Trofimov-Luke (NTL) Gain, to produce what we call NTL phonons. The energy carried by these phonons is called NTL phonon energy $E_{pho,NTL}$. We can approximate that all the energy of the accelerated electrons is converted to NTL phonon energy we can estimate $E_{pho,NTL}$.

$$E_{pho,NTL} = N_{e/h} \cdot e\Delta V_{bias} = \frac{E_Q}{\epsilon_\gamma} \cdot e\Delta V_{bias} \tag{2.7}$$

where e is the charge of an electron and ΔV_{bias} is bias potential in the detector.

To summarize, we can calculate the total charge and phonon energies.

$$E_Q = E_r + Y_{Exp} \tag{2.8}$$

$$E_{pho} = E_{pho,primary} + E_{pho,NTL} \tag{2.9}$$

$$= E_r(1 + Y_{Exp} \cdot (\frac{e\Delta V}{\epsilon_\gamma} - 1)) \tag{2.10}$$

2.5 Charge and phonon measurement

Charge measurement: To collect the electrons and holes drifting in the detector, a potential difference V_b is applied across the crystal. In response to the electron and holes in the detector, image charges are generated in the electrodes, the amount of image charge created is given by Ramo theory [22] [14][23],

$$Q = q\phi_0(x) \tag{2.11}$$

where Q is the image charge, q is the drifting charge, and $\phi_0(x)$ is the weighting potential at position x. Weighting potential is used to weight the induced charge by the distance the charge travels through the lattice.

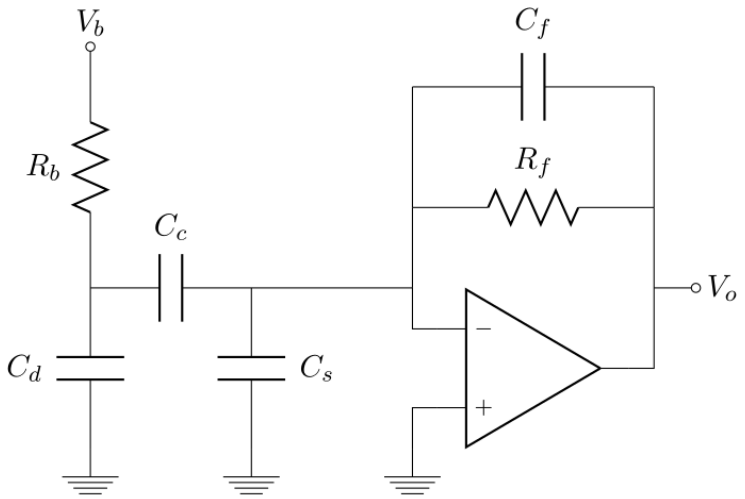

Figure 2.4: Read out circuit for ionization measurement[11]

A simplified readout circuit for ionization measurement is shown in Fig: 2.4, the detector is approximated as a capacitor C_d which is biased by V_b through a resistance R_b. Feedback capacitor C_f collects the charges induced in the detector, which increases the voltage at V_o our output voltage. C_C is a coupling capacitor and C_s is any stray capacitance. The capacitor then drains out through R_f with a time constant $\tau = R_f C_f$. The amplitude of the charge pulse is proportional to the image charge and can be used to calculate $N_{e/h}$ and E_Q[11].

Phonon measurement: The phonon sensor is known as Transition-edge-sensors (TES). These sensors are placed on the flat surface on both sides of the detector. The sensor consist of aluminum fins mounted on germanium using photo-lithography. The sensor also consists of tungsten strips (W) and is cooled down to superconducting state of aluminum ($T_c = 1.2K$) and tungsten ($T_c = 80mK$). If a phonon in the detector hits the surface where an aluminum fin is placed, it enters the fin. If the energy of the phonon is greater than the superconducting gap energy of aluminum $2\Delta_{AL} = 340\mu eV$, the phonon will break up a cooper pair. If the phonon energy is less than $2\Delta_{Al}$, they can not break any more cooper pairs and they are returned to the detector. Here the energy in them is lost to the sensor[14].

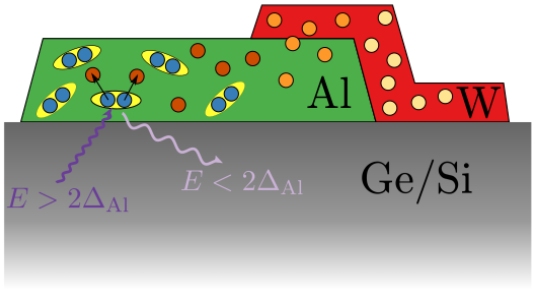

Figure 2.5: Cartoon depicting the cross-section of a TES module[11]

As shown in Fig: 2.5 the Al fins overlap with the W strip, this is called the bi-layer. The energy required to break up cooper pair in W is lower than that of Al, hence the quasi-particle created in Al diffuses through the bi-layer into W. Once the particles are inside W they scatter and lose energy. W has the smallest gap energy ($2\Delta_W < 2\Delta_{Bi} < 2\Delta_{Al}$) the particles can not go back and are trapped in W. The phonon energy is then converted into the electrical signal. W is held at the edge of its critical temperature, the phonons scattering in W increases the temperature of the W just enough to cross its T_c and have normal resistance.

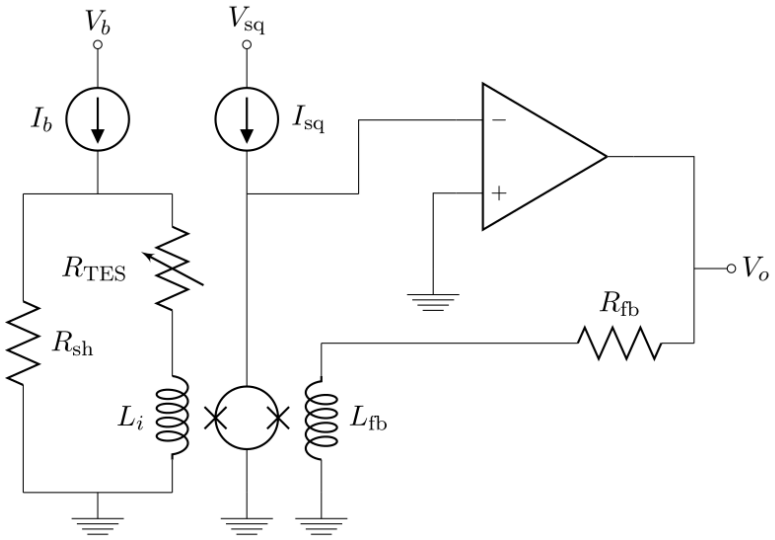

Figure 2.6: Readout circuit for phonon measurement[11]

Fig: 2.6 shows a simplified readout circuit for phonon measurement, the resistance in TES is depicted with a variable resistor R_{TES}. Any change in R_{TES} changes the current in the circuit. This changes the current through the inductor L_i which is coupled with a superconducting quantum interference devices (SQUID). SQUID can measure small changes in magnetic field and the array detects the change in magnetic flux from L_i. This change in the magnetic field changes the current in the feedback inductor L_{fb}, which is then amplified and read out through V_o.

Chapter 3

Anomaly detection

3.1 Introduction to machine learning

Machine learning (ML) is a branch of artificial intelligence. ML can be defined as a field of study that gives computers the ability to learn without being explicitly programmed [24]. It can perform tasks that involve recognition, diagnosis, planning, robot control, prediction, etc [25].

Machine learning is the automated process that extracts patterns from data and is then used to perform a specific task without using explicit instructions. In machine learning, a model is defined as an algorithm that has been tailored to recognize certain types of patterns. A neural network to predict the next few words of an incomplete sentence is an example of a model.

Advantages of ML are

- Identifies trends and patterns better and faster than a human
- No human intervention needed (automation)
- Can handle multidimensional and multi-variety data
- Scope of continuous improvement

3.2 Anomaly Detection

Anomaly detection (AD) is the identification of events and observations that differ significantly from the majority of data [26]. AD can be used to detect simulation/analysis bugs or unknown events which we may have missed in our modeling. The aim of this project is to investigate the application of AD techniques in the data analysis of dark matter search data. AD is an unsupervised ML algorithm, which means we do not need to train our model using a labeled dataset. This is particularly useful in cases where we don't have true or simulated data that can be used for training, given a dataset AD tries to differentiate between the events on basses of their similarity/dissimilarity with other events. In other words, unsupervised ML allows the

system to identify patterns within the system on its own. Unsupervised ML techniques do not understand what that pattern/cluster represents and after we have successfully extracted the pattern usually a human is needed to understand the context of these patterns.

For this project, we will use a combination of two algorithms to cluster our data, t-SNE and DBSCAN.

3.3 t-SNE

t-SNE stands for t - distributed Stochastic Neighbour Embedding [27]. It is a stochastic algorithm that is a process that has some randomness involved in it. Every time we run the algorithm, the results will be similar but not exactly the same. and the aim is to embed the data from a high-dimensional space to a lower-dimensional space. Before understanding the algorithm let us define some symbols: set of n points in our higher dimensional space is denoted by $X = \{X_1, X_2, X_3, ..., X_n\}$ and the set of points in two dimensional space by $Y = \{Y_1, Y_2, Y_3, ..., Y_n\}$. Given the set X with n points, we want to embed the data in set Y (in two dimensions) on n points, such that similarities between the points are conserved.

The first step is to calculate a similarity matrix p using the points in set X.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}, \text{ where} \tag{3.1}$$

$$p_{j|i} = \frac{exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} exp(-||x_i - x_k||^2/2\sigma_i^2)} \tag{3.2}$$

$p_{j|i}$ is the probability that the point X_i will take the point X_j as a neighbor under a Gaussian centered at X_i. Elements of the matrix $p_{j|i}$ will be higher values if they are closer than if they are far away. We don't need the probability of a point to be a neighbor of itself hence, we set $x_{i|i} = 0$. This also allows us to make sure, the sum of all possibilities for a point is 1. But this creates another problem, now $p_{i|j} \neq p_{j|i}$ due to the difference in denominator. To overcome this problem we calculate p_{ij} (a symmetric matrix) to make sure that for any point $p_{ij} = p_{ji}$.

The variance σ_i of the Gaussian is selected for each individual point using a user-defined parameter perplexity. We need this because in the dense region a smaller value of σ_i is required than in sparser regions. To overcome this the user-defined variable perplexity is chosen which depends on the entropy of the probability distribution of p_i. If the points are close by the entropy increases and with that the value of σ_i.

$$\text{perplexity} = 2^{(-\sum_j p_{j|i} log_2 p_{j|i})} \tag{3.3}$$

The second step is to randomly generate n, two-dimensional points in set Y, we then calculate a similarity matrix q using these points.

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{3.4}$$

We want the matrix q to be the same as matrix p, that is we want points y to show the same neighbor relationships as x. But in q we have used t-distribution with a single degree of freedom f(t) instead of Gaussian as in p. This is done to solve the crowding problem we face during embedding.

$$f(t) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}(1 + \frac{t^2}{\nu})^{-\frac{\nu+1}{2}} \tag{3.5}$$

The Crowding Problem: Let us start with a case where we have equidistant 11 data points in a 10-dimensional space, but there is no way to plot 11 equidistant points in a 2 d space. Other than that volume of a sphere scales as r^m where r is the radius of the square and m is the number of dimensions. We have fewer data in 2d and this creates a crowding of points. We use t-distribution as it has a heavier tail than Gaussian and this causes the points with moderate similarity to be mapped further apart in a lower dimension.

$$\text{volume of n ball or radius r} = \frac{\pi^{n/2}}{\Gamma(\frac{n}{2}+1)}r^n \tag{3.6}$$

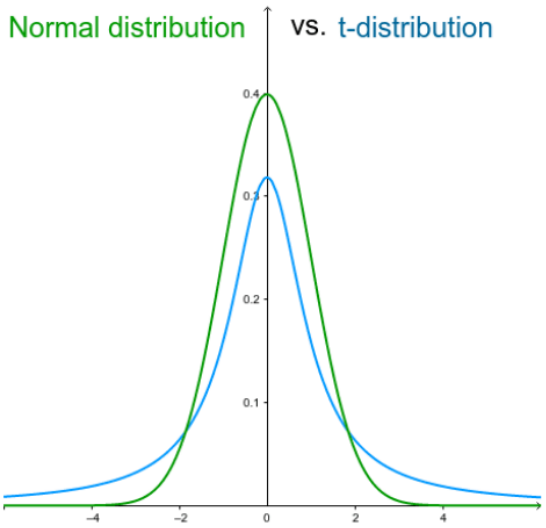

Figure 3.1: Comparison of Gaussian (Normal) distribution with a t-student distribution.

After we have got the matrix q from randomly generated set Y, we want matrix q to be equal to matrix q, for that we create a cost function as defined below:

$$C = \sum_i \sum_j p_{ij} log \frac{p_{ij}}{q_{ij}} \tag{3.7}$$

Then minimization of the cost function is performed using gradient descent method. We will use points in set Y as the variables. The form of gradient descent for a point y_i replacing t-distribution in q with a Gaussian for simplicity is given by:

$$\frac{\delta C}{\delta y_i} = 2\sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j) \tag{3.8}$$

Physically this can be interpreted as a net force created by a set of springs between point y_i and all other points y_j. All the springs exert a net force in the direction $y_i - y_j$, and the force for spring between y_i and y_j will be attractive or repulsive depending on if the probability $p_{j|i} - q_{j|i}$ is greater than or less than $p_{i|j} - q_{i|j}$. The variable update formula is given by:

$$Y^{(n+1)} = Y^{(n)} - \eta \frac{\partial C}{\partial Y} + \alpha(n)(Y(n-1) - Y(n-2)) \tag{3.9}$$

where $Y(n)$ indicates the solution at iteration n, η indicates the learning rate, and $\alpha(n)$ represents the momentum at iteration n. To make sure that gradient descent does not get stuck in a local minimum we add a large momentum term is added to the gradient.

3.4 DBSCAN

Density-based spatial clustering of application with noise (DBSCAN) is a clustering method. Clustering is a technique that separates the data points into specific bunches or groups. DBSCAN uses spatial information and groups data based on a distance measurement and a minimum number of points[28]. It can divide the data into clusters and also separate noise points which are in low-density regions. The advantage of DBSCAN is we don't need to specify the number of clusters in the data a priori. It can also find arbitrary-shaped clusters unlike some other clustering algorithms like k-mean clustering.

There are two parameters we need to define:

1. **minPts:** Minimum number of points in a neighborhood to be called a cluster.
2. **eps** (ϵ)**:** A distance measure to define if a point is a neighbor.

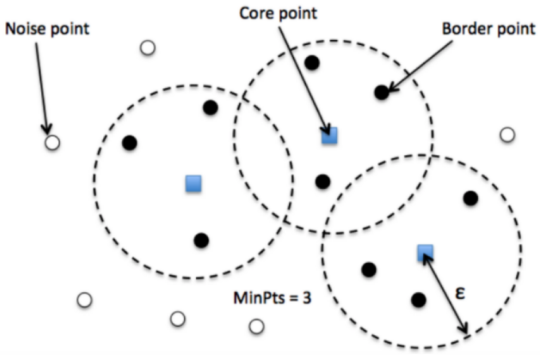

Figure 3.2: Cartoon describing different types of points in DBSCAN.

There are three types of points that are used in DBSCAN algorithm:

1. **Core points:** If a point has minPts number of points under ϵ distance from itself it is a core point.
2. **Border points:** If a point has at least one core point under ϵ distance from itself it is a border point.

3. **Noise points:** If a point has no core points or minPts of points under ϵ distance from itself it is a noise point.

The algorithm starts by picking a point randomly from the data set. Then it figures out if the point is a core, border, or noise point. Depending on this classification next step is taken. If the point is a core point, all the points are considered part of the cluster belonging to the first point and then start the process again with each point in the neighborhood.

if the point is a border point, the algorithm picks another point in the neighborhood of the last core point. If there are no points left in the neighborhood of the last cluster point it picks another point randomly from the set of unclassified points and starts the process over again.

If the point is a noise point, the algorithm jumps to a new point randomly chosen from the set of unclassified points.

3.5 Dataset

The dataset used for this analysis was taken from a sapphire scintillator detector which was fabricated at Texas A&M University, and used in the MINER experiment. It is made up of Al_2O_3 cylindrical crystal with a diameter of 7.6 cm and width of 0.4 cm with a mass of 73 gm. It produces phonons and photons, photons are produced through scintillation, and phonons are produced in the same manner as in the Ge detector described in section 2.2. To collect phonons Al fins are connected to the detector and they are collected using a TES module. There are 4 phonon channels (A, B, C, and D) as seen in Fig: 3.3. Aluminum has a lower mass than Silicon and hence can be used to search for low-energy neutrino.

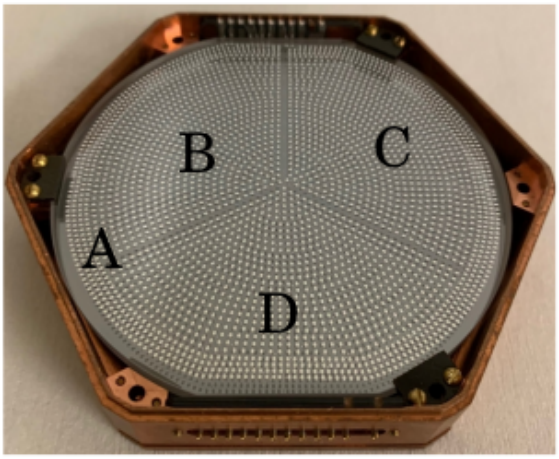

Figure 3.3: Sapphire detector with different channels.

The detector is placed in a dilution refrigerator approximately 4 meters away from a reactor core. The reactor is kept inside a pool which is covered with a concrete wall to prevent backgrounds like γ neutrons and cosmic muon. We want to observe the neutrinos from the reactor core and

measure phonon energy due to it. The reactor power is 1 MW and neutrino flux at 1 m from core is approximately $10^{12} cm^{-2} s^{-1}$.

Figure 3.4: Example of bad pulses

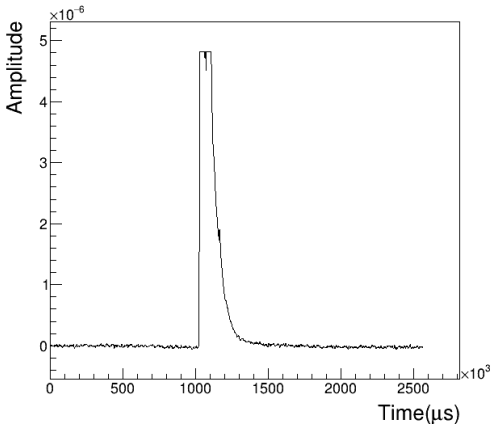

Figure 3.5: Saturated pulse

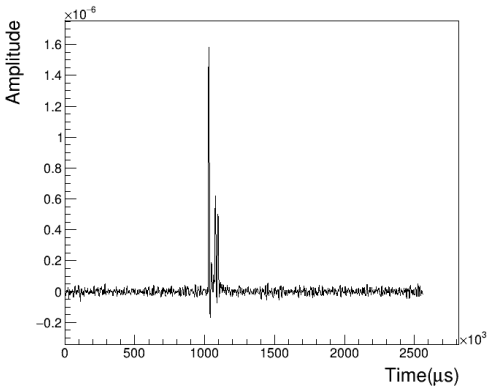

Figure 3.6: Bad pulse

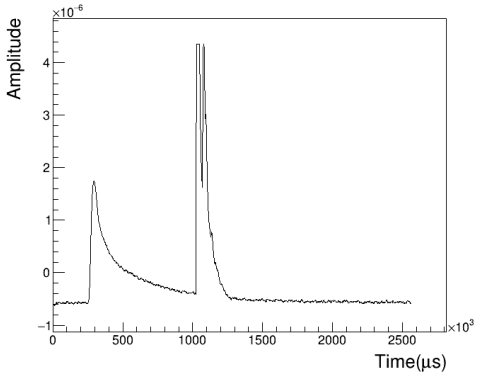

Figure 3.7: Pile-ups

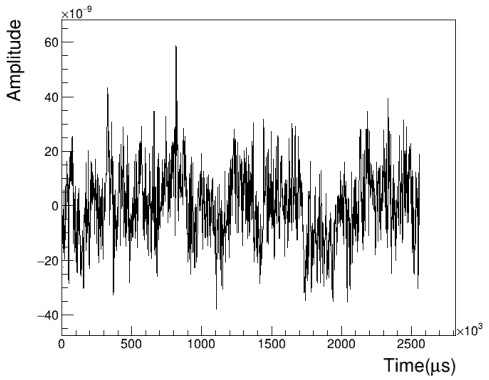

Figure 3.8: Noise

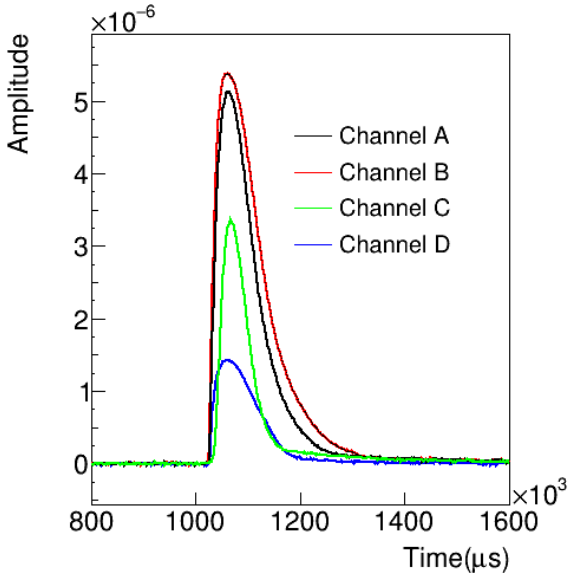

Figure 3.9: Example of a good pulse.

The pulses recorded can be good pulses or those consistent with noise Fig. 3.5. For measuring an event this weak, we need to make sure we use only the good pulses in our analysis. In our analysis, we considered 105250 events each with 4 channels in the sapphire detector. We further defined 10 variables for each pulse in an event which is 40 variables per event plus one variable which is calculated using all four pulses. The variables were:

1. **Prepulse STD:** Standard deviation of the first 400 bins of the pulse. Standard deviation (σ) is given by
$$\sigma = \sqrt{\frac{(x_i - \mu)^2}{N}} \tag{3.10}$$
where μ is the mean of the sample and N is the size of the sample.
2. **Postpulse STD:** Standard deviation of the last 256 bins of the pulse
3. **Max content:** Bin with the maximum amplitude of the pulse
4. **Min content:** Bin with minimum amplitude of the pulse
5. **Max tail:** Maximum amplitude in the last 512 bins of the pulse
6. **Rise time:** Time taken by the pulse to rise from 10% to 90% of its maximum amplitude
7. **Fall time:** Time taken by the pulse to fall from 90% to 10% of its maximum amplitude
8. **Full-Width Half Maximum(FWHM):** Width of the pulse at 50% of its maximum amplitude
9. **Full-Width 90% Maximum(FW90M):** Width of the pulse at 90% of its maximum amplitude
10. **Full-Width 10% Maximum(FWHM):** Width of the pulse at 10% of its maximum amplitude

11. **Amplitude bin STD:** standard deviation of bin number containing maximum amplitude of pulses from the 4 channels.

3.6 Result

3.6.1 t-SNE + DBSCAN on sapphire detector data

We started by converting the triggered data from 105250 events, each containing 4 pulses of length 2000 μs to a 41-dimensional feature vector with features described in the previous section. Then these 105250 vectors of 41 dimensions are emended to a two-dimensional space using t-SNE. After that, the data in the two-dimensional space was clustered using DBSCAN. The scripts for t-SNE and clustering can be found in Appendix A.

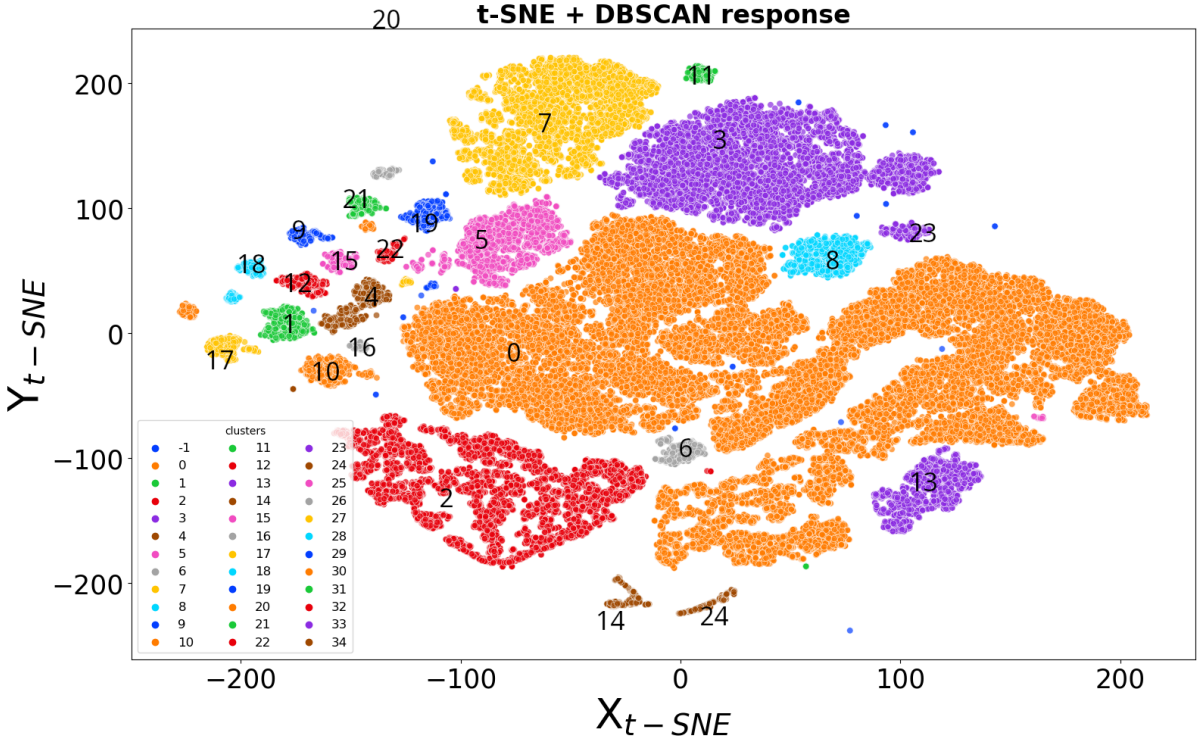

Figure 3.10: Result of t-SNE + DBSCAN on data taken from Sapphire detector.

We see that we were able to collect our data into 36 different clusters as shown in Fig: 3.10. The next step is to manually examine the contents of each cluster and determine what kind of pulse they correspond to. We also notice that cluster number 0 is a lot bigger than other clusters and we may need to divide that cluster.

Table 3.1: **cluster content**

Cluster	**Remark**	**Cluster**	**Remark**
-1	Un-clustered	17	Noise
0	Mixed	18	Noise
1	Noise	19	Noise
2	Saturated	20	Noise
3	Noise	21	Noise
4	Noise	22	Noise
5	Noise	23	Noise
6	Mixed	24	Pile up
7	Noise	25	signal
8	Noise	26	Noise
9	Noise	27	Noise
10	Noise	28	Noise
11	Noise	29	Noise
12	Noise	30	Noise
13	signal	31	signal
14	Pile up	32	Saturated
15	Noise	33	Noise
16	Noise	34	Saturated

Table 3.1 lists the types of pulses in the clusters, next we will try to examine these clusters to get signals.

Pileups

While examining clusters 14 and 24 we notice that all the pulses in this cluster were pile-ups. Fig: 3.12 shows a sample pulse from cluster 24. We can see in Fig: 3.11 that all the pulses with pile-up were detected by the t-SNE algorithm and separated from the other pulses.

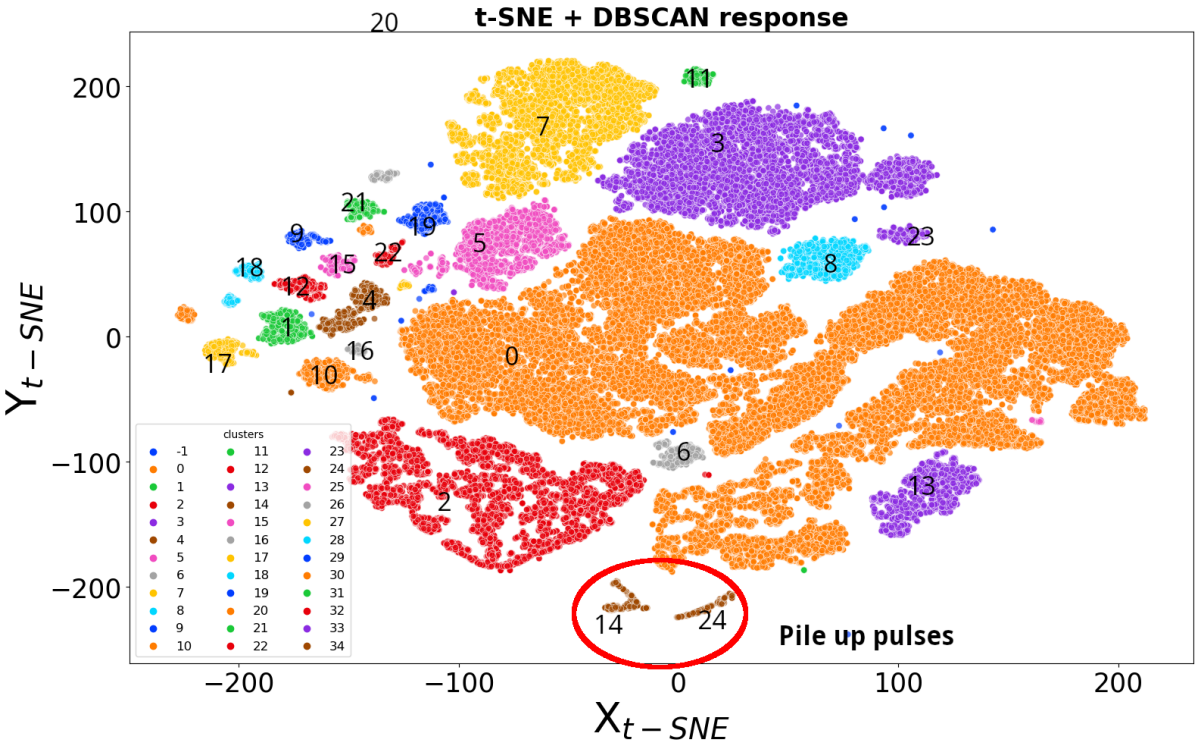

Figure 3.11: Clusters 14 and 24 containing pile up pulses marked by a red loop.

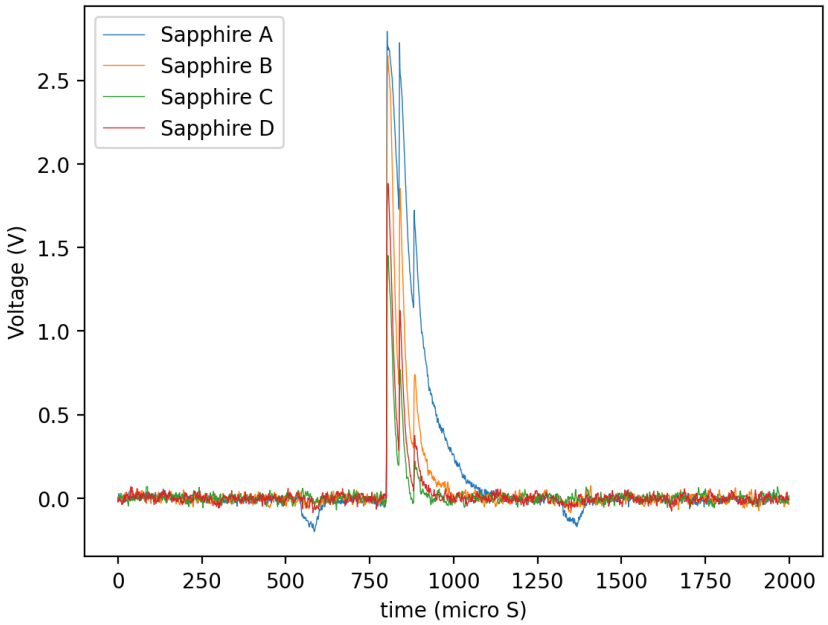

Figure 3.12: A piled-up pulse example from cluster 24.

Saturated pulses

t-SNE was also able to detect pulses that were saturated and separate them into different clusters.

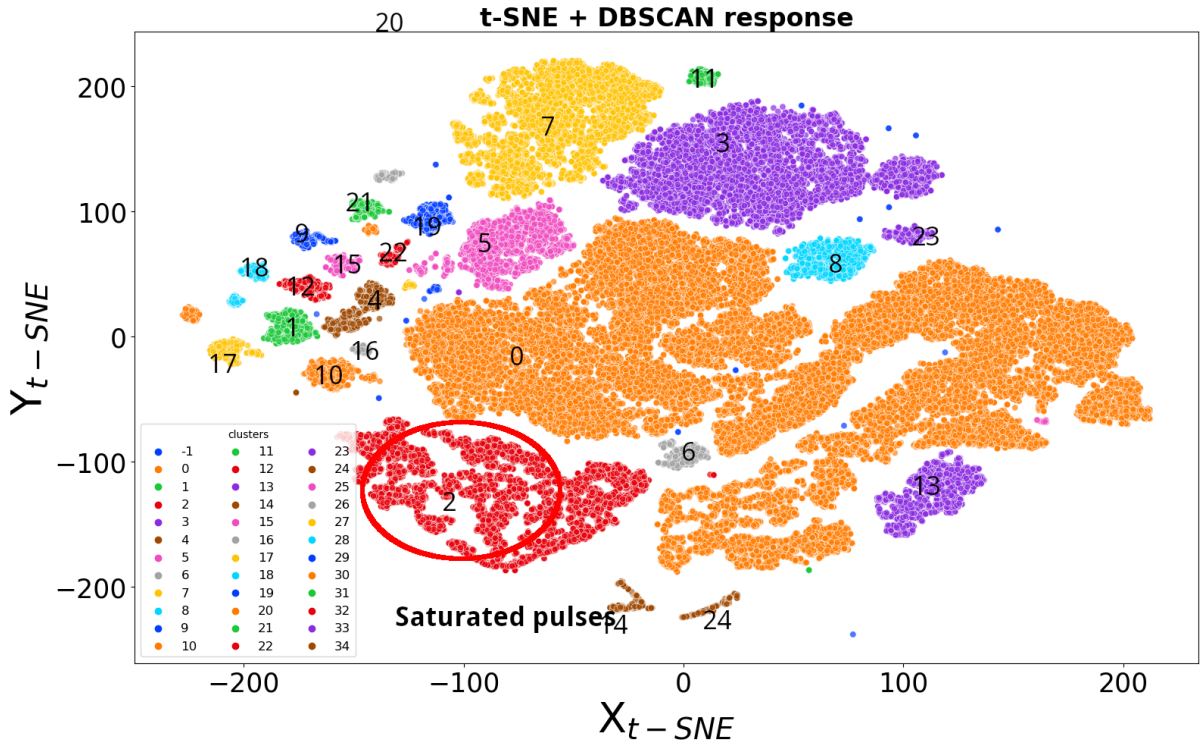

Figure 3.13: Clusters 2 containing saturated pulses marked by a red loop.

It is also noteworthy that t-SNE was even able to understand different types of saturated pulses as shown in Fig: 3.14, 3.15 and 3.16.

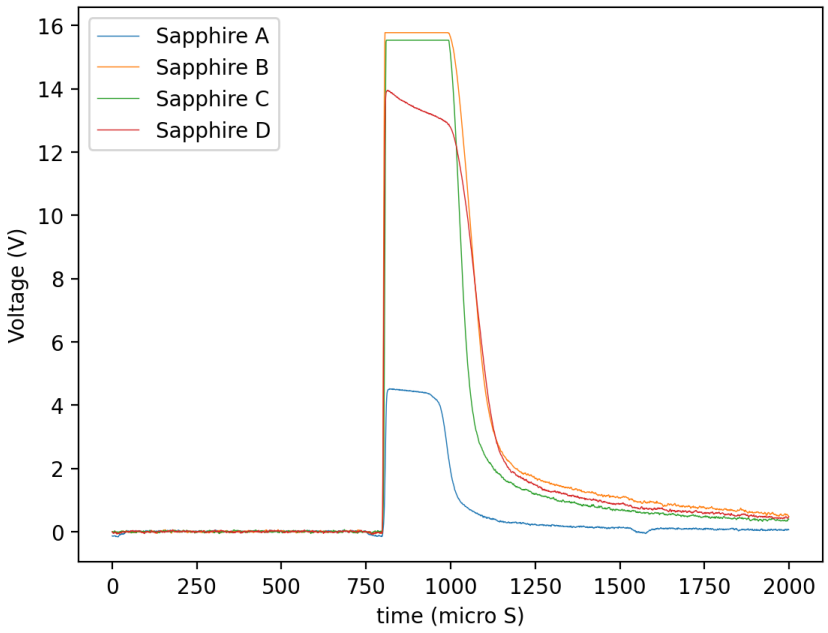

Figure 3.14: A saturated pulse example from cluster 2.

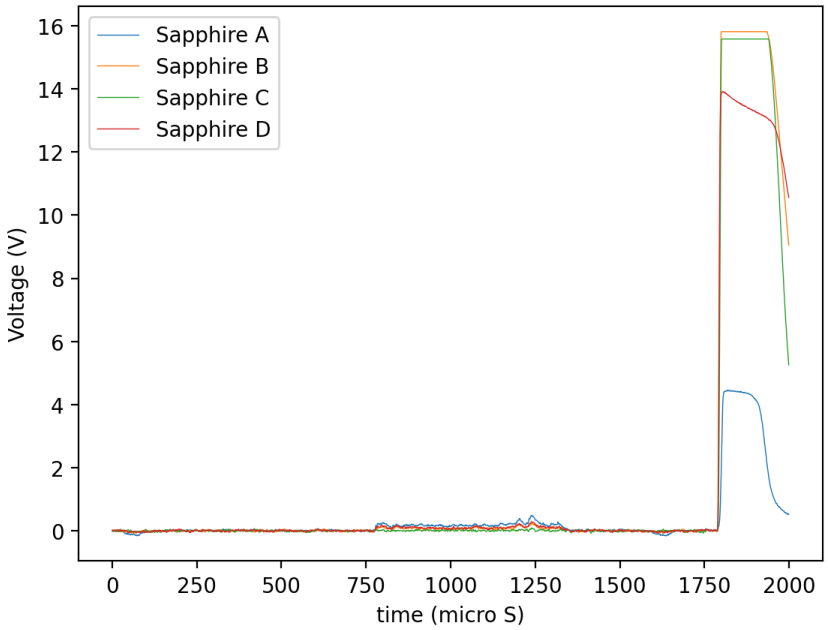

Figure 3.15: A saturated pulse example from cluster 32.

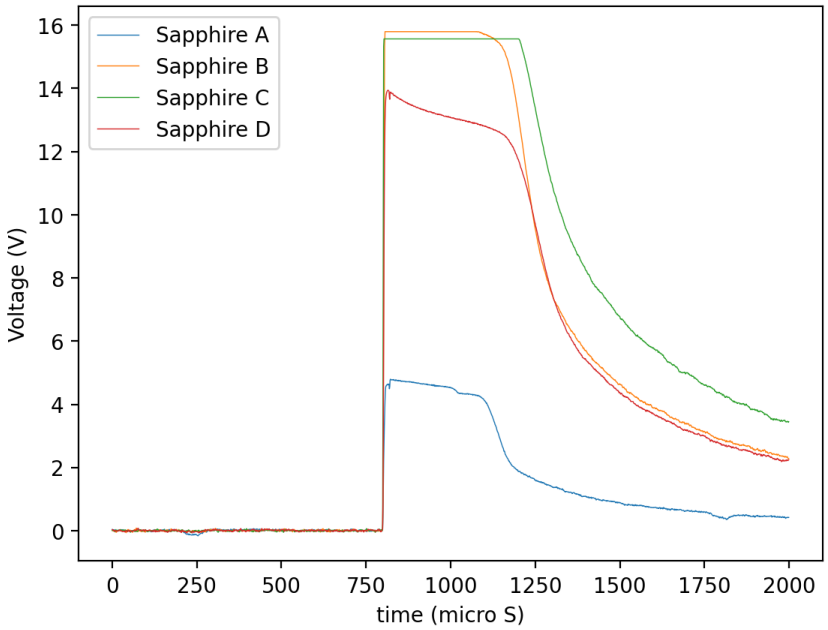

Figure 3.16: A saturated pulse example from cluster 34.

Noise

While examining the clusters, cluster number -1, 1, 3, 4, 5, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19, 20, 21, 22, 23, 26, 27, 28, 29, 30 and 33 all contained noise pulses. The location of these clusters are shown in Fig: 3.17.

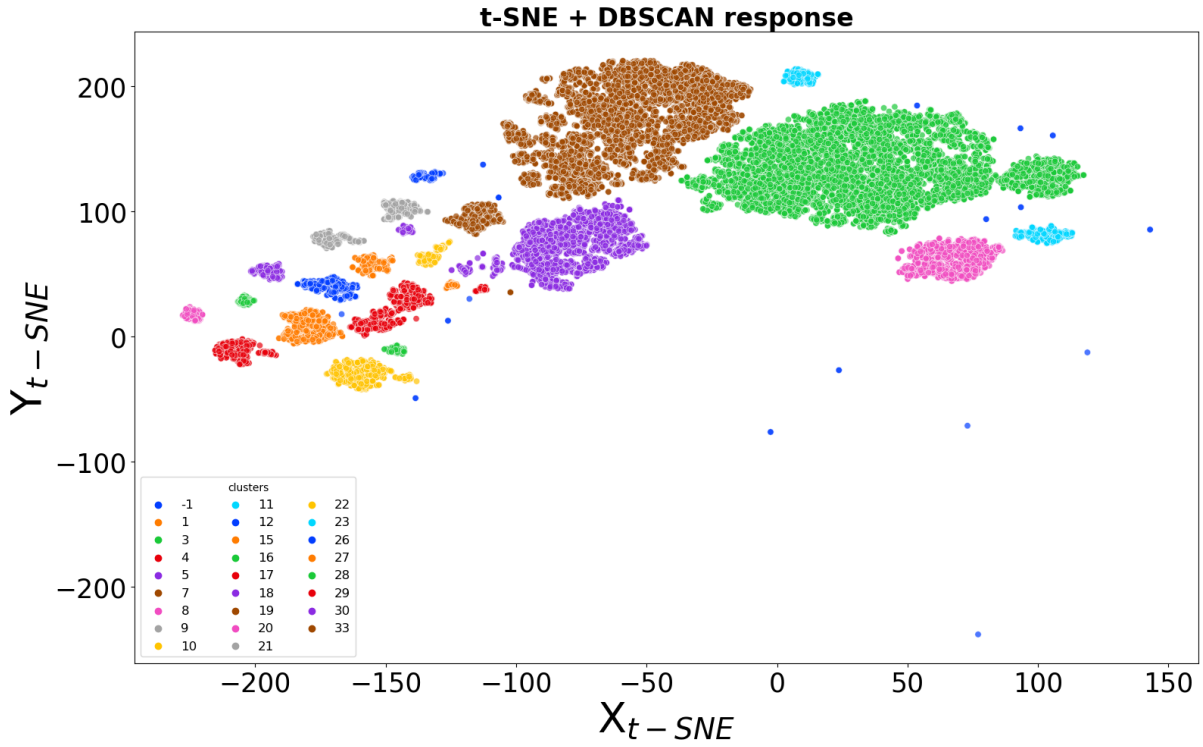

Figure 3.17: Clusters containing noise pulses.

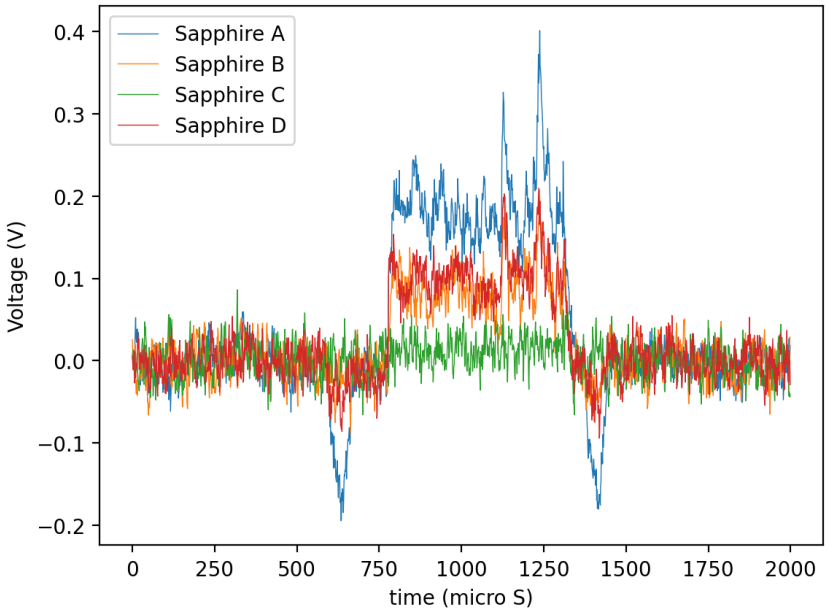

Figure 3.18: A noise event example from cluster 3.

Fig: 3.18 shows a pulse from cluster 3, all the pulses from the above-mentioned clusters were similar to this pulse.

Signal

Analyzing clusters 13, 25, and 31 revealed that they contained only signal pulses but there were two clusters 0 and 6 where there were mixed-signal and noise pulses in the cluster. Cluster 6 as seen in Fig: 3.19 is actually two clusters very close by less than 5 units. The ϵ used for DBSCAN algorithm is 5 units and it was not able to resolve the into two clusters.

Motivated by this observation, I tried to separate them using a linear cut in the t-SNE phase space. The straight-line $Y = \frac{6}{5} \cdot X - 80$ was chosen which is shown in Fig: 3.19, here X is tsne-2d-one variable and Y is tsne-2d-two variable. Using this line as a cut I was able to determine that pulses below this line were signals. I tried to extend the same idea on the much larger cluster 0 and most of the pulses below this line were signal events. Other clusters which were previously determined to be signal clusters also lie below this line, while noise above this line.

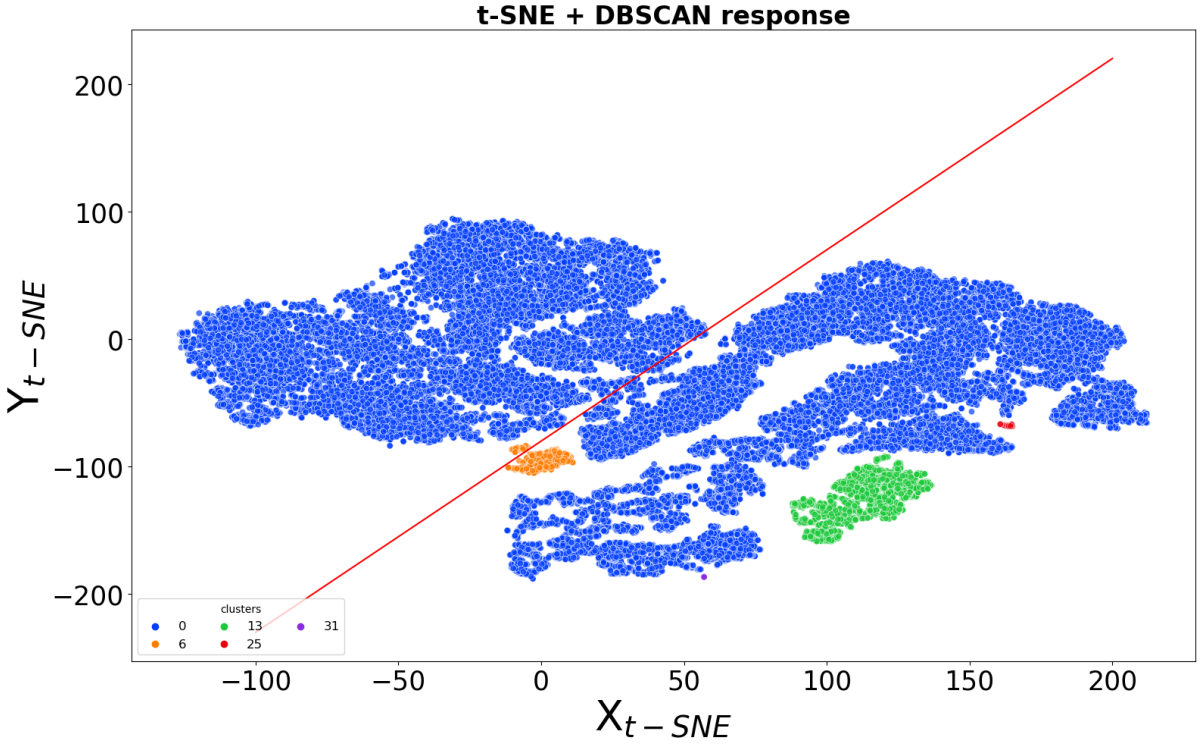

Figure 3.19: Clusters containing signal pulses with a linear cut shown in red ($Y_{t-SNE} = \frac{6}{5} X_{t-SNE} - 80$).

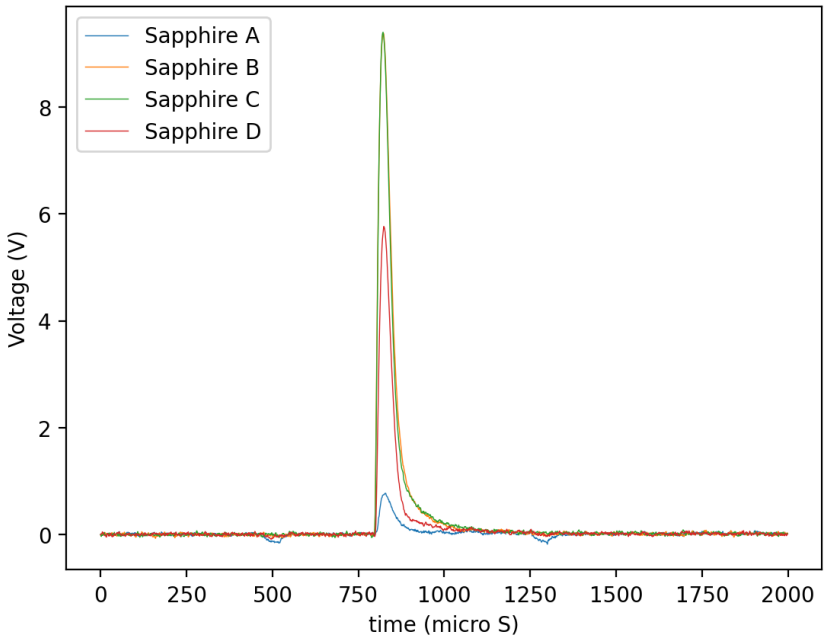

Figure 3.20: A signal pulse example from cluster 13.

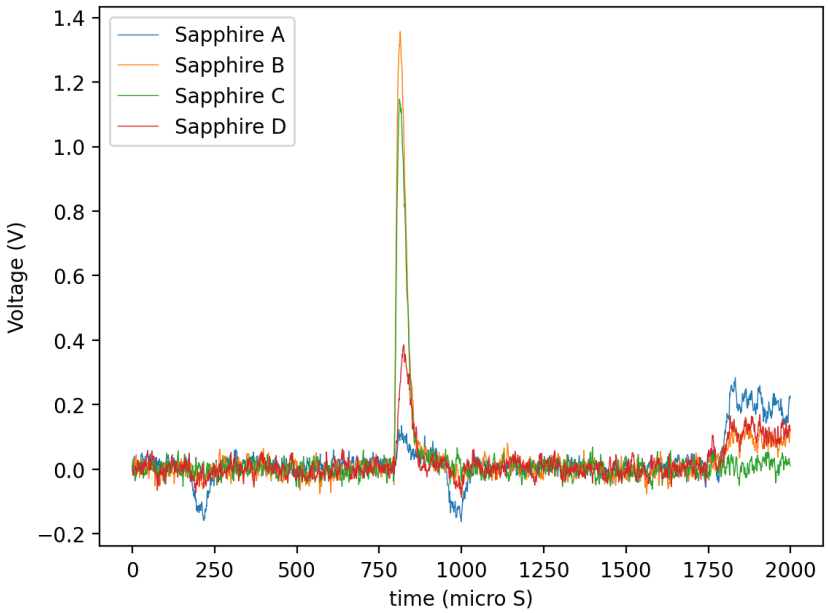

Figure 3.21: A signal pulse in cluster 6 below the line.

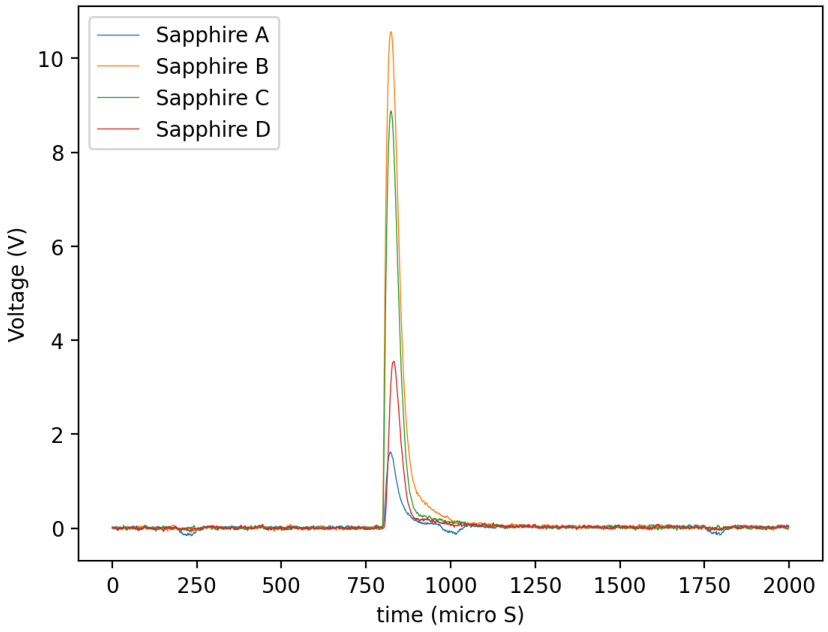

Figure 3.22: A signal pulse in cluster 0 below the line.

3.6.2 Quality of separation with number of events

Even though t-SNE + DBSCAN were able to cluster the data into different pulses, but it struggled to completely distinguish some noise from the signal which we see as cluster 0 in Sec. 3.6.1. A filtration at this point was done such that clusters that contain only the bad pulses were removed and clusters with good pulses or mixed pulses were kept. We used the anomaly detection algorithm again on this filtered data which gave us a much better result this time.

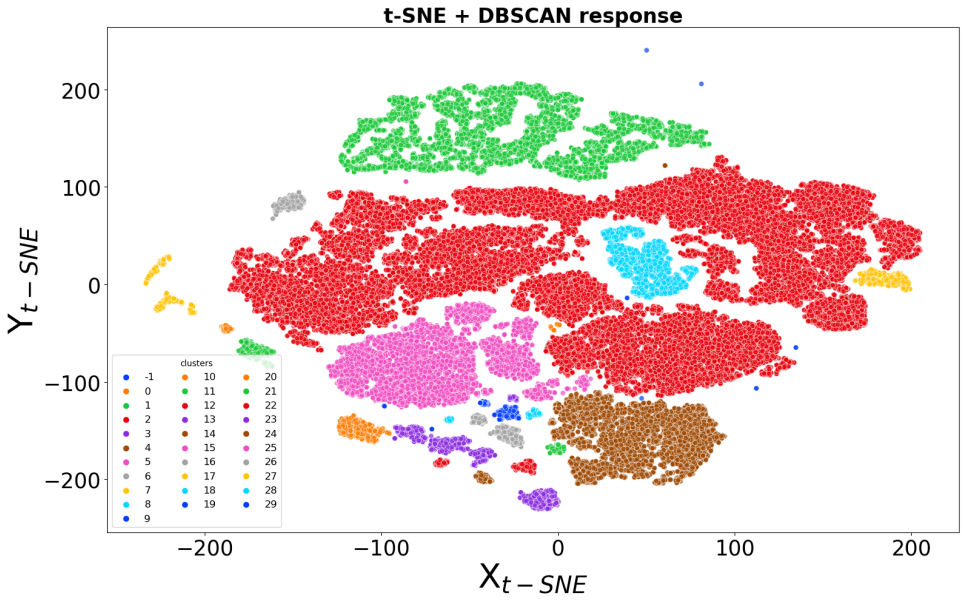

Figure 3.23: Result of t-SNE + DBSCAN on dataset taken from sapphire detector with 105250 events and perplexity 100.

In the first filtration step data was divided into 31 clusters out of which we again see similar results as in the last section. There were clusters with pure Noise and pure signal pulses but cluster 2 still was not been able to fully separate and there was a mix of pulses in this cluster.

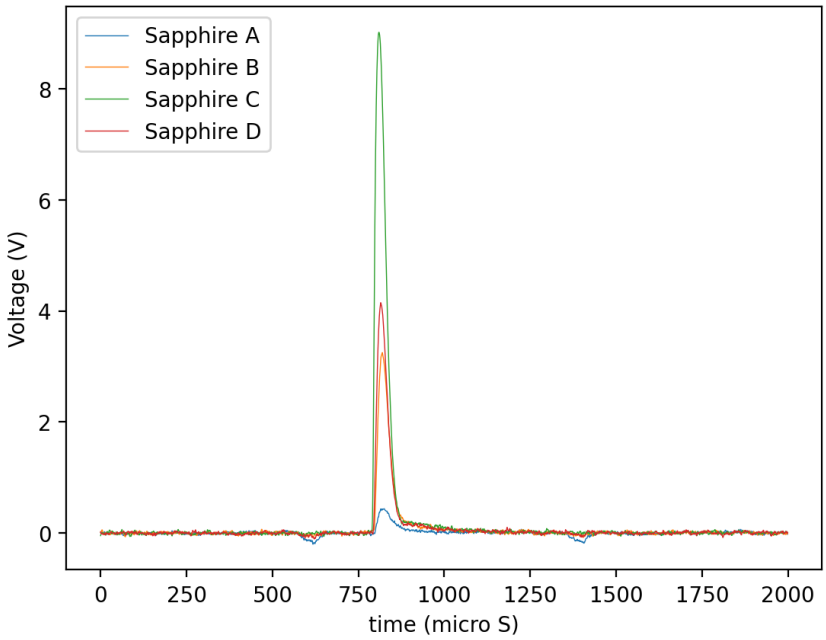

Figure 3.24: Pulse example from cluster 2.

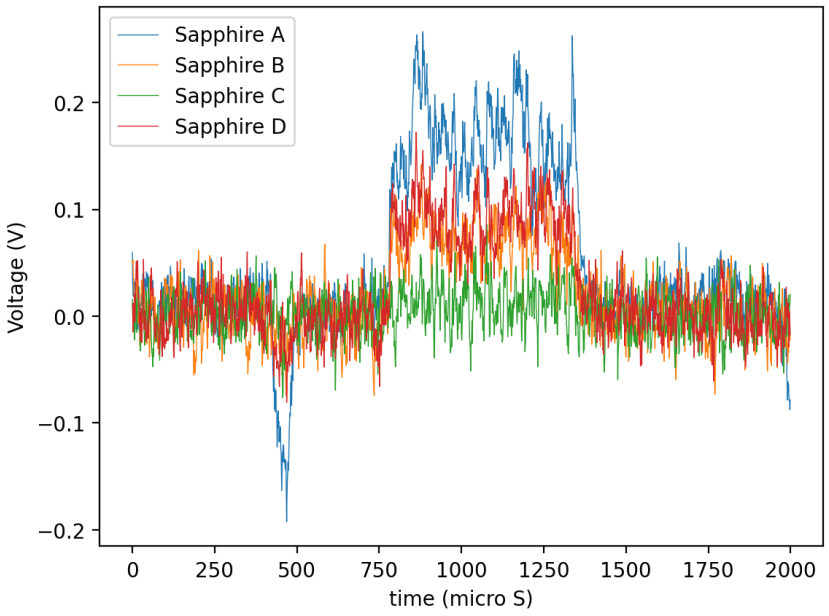

Figure 3.25: Noise pulse example from cluster 2.

We performed a filtration at this step by removing any cluster which contained all noise pulses and tried to perform t-SNE + DBSCAN again on the filtered dataset. The results from this process are shown in 3.26.

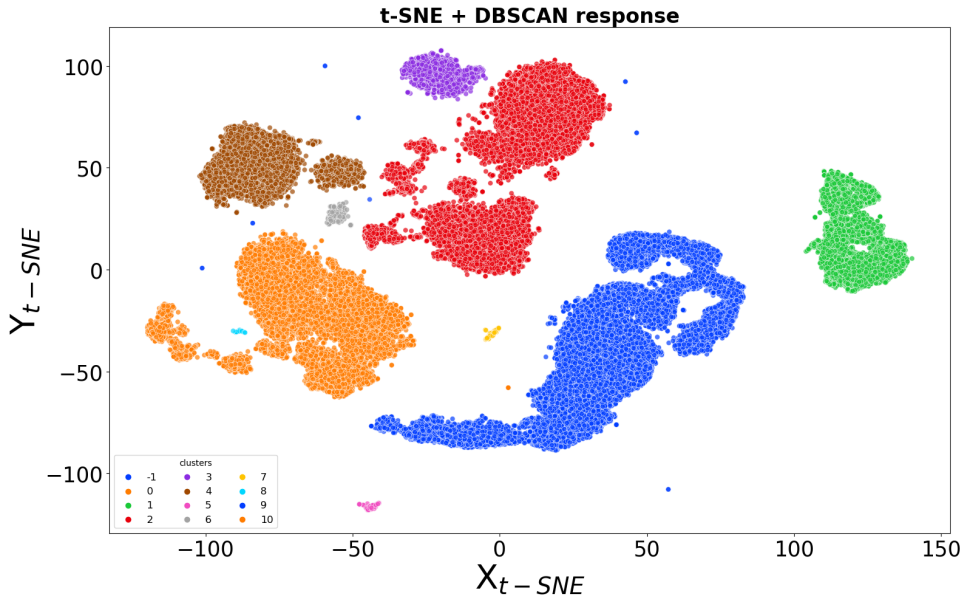

Figure 3.26: Result of t-SNE + DBSCAN on filtered data from Fig: 3.23, with 57648 events and perplexity 100

One of the probable reasons that the same algorithm when used again on the same dataset, just filtered, gave us a much better result might have to do with the number of events in our dataset. While filtering we decreased the number of events the algorithm had to go through and hence provide a better result. We will try to check this in the next section.

3.6.3 Quality of separation with perplexity

Perplexity is a hyper-parameter that the user has to provide to the t-SNE algorithm. The value of perplexity is used to define the variance σ_i of the Gaussian function for each individual point. We need this because in the dense region a smaller value of σ_i is required than in sparser regions. To overcome this the user-defined variable perplexity is chosen which depends on the entropy of the probability distribution of p_i. If the points are close by the entropy increases and with that the value of σ_i. Perplexity is defined as:

$$\text{perplexity} = 2^{(-\sum_j p_{j|i} log_2 p_{j|i})} \tag{3.11}$$

For our problem the events lie in approximately the same space and increasing or decreasing the number of points in each dataset changes the density of these points. As perplexity is the value that helps us to tune the algorithm for event density, we decided to run a computational experiment to try to understand the relationship between the number of points and perplexity for our dataset.

We decided to use our algorithm for a range of the number of points over a range of perplexity and then check the results for each trial. There were 4 samples taken from our complete dataset with 25000, 50000, 75000, and 100000 events. For all 4 samples, we ran the algorithm for 5 values of perplexity: 50, 75, 100, 150, and 175.

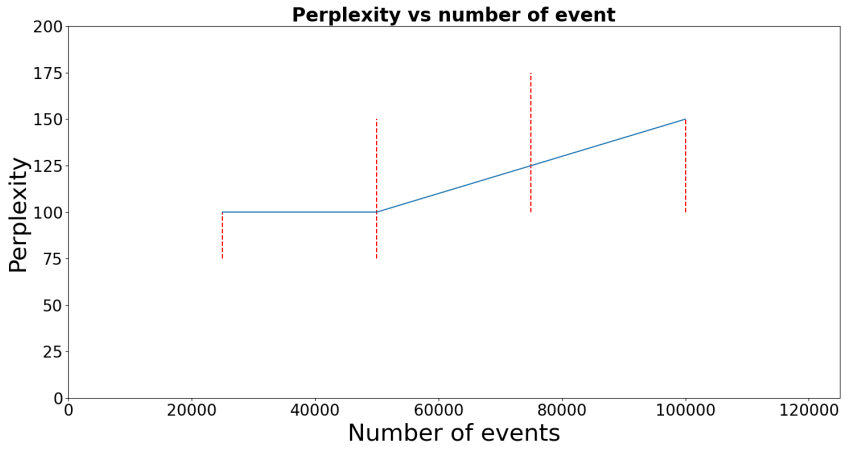

Figure 3.27: Perplexity vs Number of events.

In Fig: 3.27 we see a plot of perplexity vs the number of points. Dashed lines show the range of perplexity for the same number of points in the dataset where the algorithm was able to separate the noise and bad pulses from good pulses and the blue line joins the points of best separability. This check was done manually by looking at pulses in each cluster.

3.6.4 Measure of the quality of separation

Finally, we run this algorithm with all the events i.e. 10500 and perplexity 150. This divided our dataset into 24 clusters.

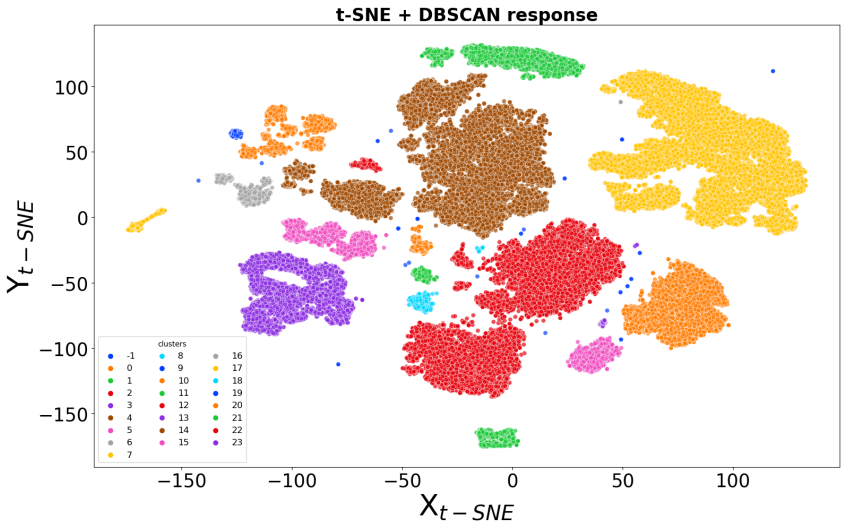

Figure 3.28: Result of t-SNE + DBSCAN on dataset taken from sapphire detector with 105250 events and perplexity 150.

Table 3.2: **cluster contents**

Cluster	Remark	Cluster	Remark
-1	Un-clustered	12	Noise
0	Noise	13	Noise
1	Saturated	14	Noise
2	Noise	15	Saturated
3	Saturated	16	Saturated
4	Noise	17	Good
5	Noise	18	Noise
6	Noise	19	Pile up
7	Pile up	20	Noise
8	Noise	21	Good pulse
9	Noise	22	Good pulse
10	Noise	23	Noise
11	Noise	-	-

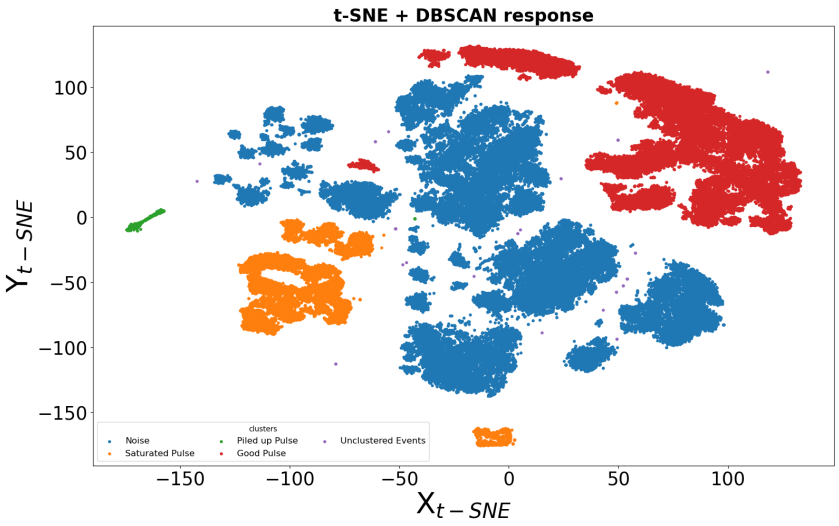

Figure 3.29: Showing t-SNE + DBSCAN response of sapphire detector from Fig: 3.28 with labeled clusters.

From Table: 3.2 we can see that all the good pulses were only in clusters 17, 21, and 22, this is also plotted in Fig: 3.29. Out of which cluster 17 comprising of events that have made very clear pulses in all three internal channels Fig: 3.30. Cluster 21 had all the pulses with very small amplitude. Other properties of the pulses in cluster 21 were that the amplitude of all three inner channels (B, C, and D) are very close and channel A has negative peaks Fig: 3.31. Cluster 22 had pulses with very high amplitudes reaching near 14 V which tells us that this cluster has pulses from high energy particles Fig: 3.32.

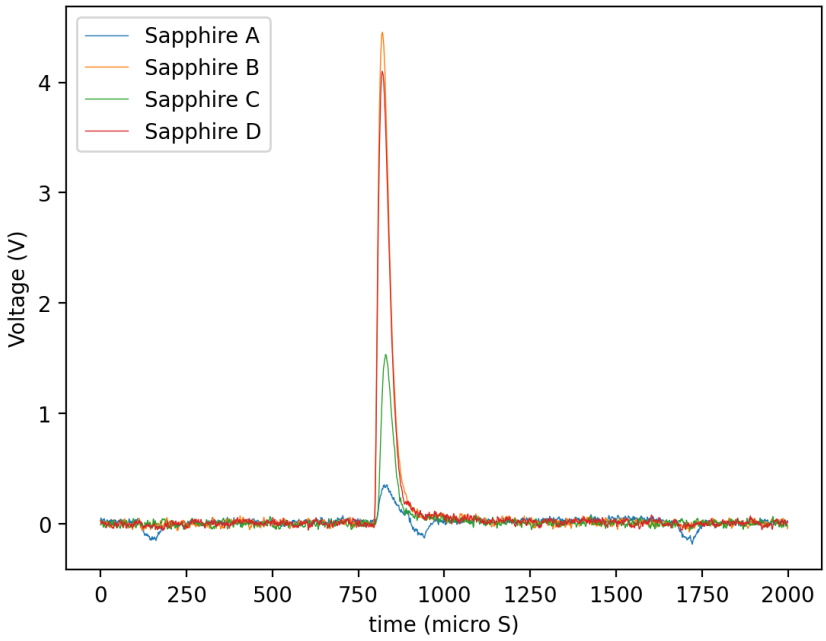

Figure 3.30: Sample pulse from cluster 17.

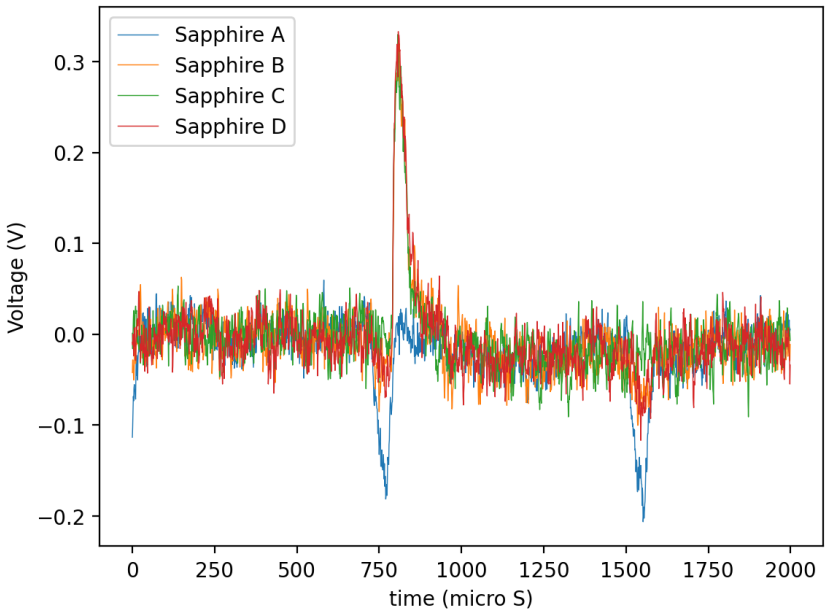

Figure 3.31: Sample pulse from cluster 21.

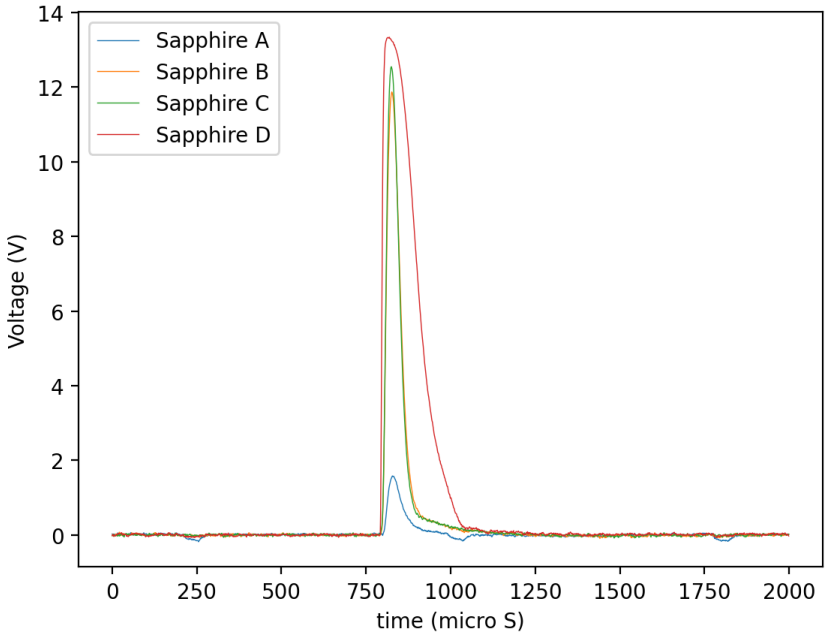

Figure 3.32: Sample pulse from cluster 22.

For checking the accuracy of this algorithm we had to manually check the clusters and count all the wrong classifications as there is no absolute true value given to us. I took 10% events from each cluster and plotted them which was around 10000 plots and then checked each one of them for miss classification. The results from them were that we had 50 misclassifications out of 3024 classified good pulses and 53 good pulses out of 7455 classified bad pulses. Hence using the above numbers we can calculate that the algorithm found 98.2 % of the good pulses with only 50 false positives. The confusion matrix for this classification is given below:

Table 3.3: Confusion Matrix

		Prediction		
		Pulse	Noise	Total
Actual value	Pulse	2974	50	$2974 + 50$
	Noise	53	7491	$53 + 7491$
	Total	$2974 + 53$	$50 + 7491$	10568

3.6.5 Optimum filter analysis

The optimum filter method tries to fit the pulse using a template to get the amplitude of the pulse, this provides us with the energy on each phonon and can also help us to triangulate the position of energy deposition in the detector [11]. The optimum filter is done by transforming the signal from a time domain to the frequency domain this helps in distinguishing the noisy part from the true signal in the data. If our signal has two components, a pulse template A(t) and Gaussian noise n(t). The signal can be written as,

$$S(t) = aA(t) + n(t) \tag{3.12}$$

here a is the factor by which we scale the template to get the pulse amplitude. Using Fourier transform we convert the signal and noise to is frequency space , $\tilde{S}(\nu)$, $\tilde{A}(\nu)$, and $\tilde{n}(\nu)$. We can then minimize the χ^2 of the pulse-template fit to get the best fit value for a. There might be a time difference between the template and the pulse, and to accommodate that we add an additional feature t_0 such that:

$$S(t) = aA(t - t_0) + n(t) \tag{3.13}$$

Partition plot

Once we have the amplitudes of pulses in all four channels we can try to position the source. We assume that a signal in a channel came from the center of the channel sector, using the amplitudes we can find approximate x and y coordinates using this formula [11]:

$$X = \frac{B_{OF}\cos(150)^\circ + C_{OF}\cos(270)^\circ + D_{OF}\cos(30)^\circ}{B_{OF} + C_{OF} + D_{OF}} \tag{3.14}$$

$$Y = \frac{B_{OF}\sin(150)^\circ + C_{OF}\sin(270)^\circ + D_{OF}\sin(30)^\circ}{B_{OF} + C_{OF} + D_{OF}} \tag{3.15}$$

where BD_{OF}, C_{OF} and B_{OF} are OF amplitude of the pulse in the D, C, and B channels respectively. X and Y coordinates calculated from the above formula can be now plotted to understand the position of a source.

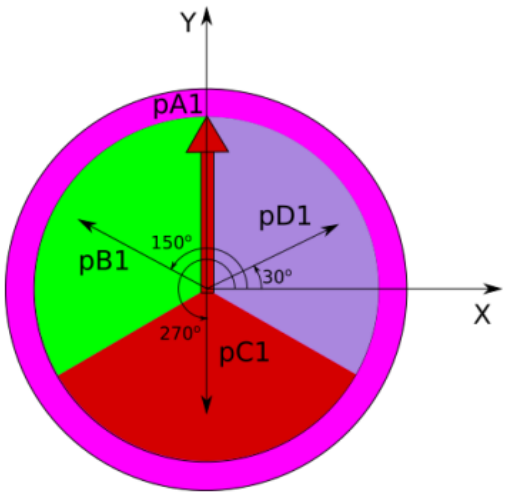

Figure 3.33: Detector configuration for partition plot.

The amplitude distribution of pulses in the dataset before and after filtration using t-SNE + DBSCAN is plotted in Fig: 3.34 and 3.35. We can see major differences in the distribution of filtered data and unfiltered data in the high amplitude and low amplitude regions. In the high amplitude region, a lot of pulses were saturated which were removed in the filtration step and in the low amplitude region there were noise pulses as shown in Fig: 3.35. The black histogram in Fig: 3.34 shows the pulse amplitude distribution for filtration using χ^2 cut. From Fig: 3.34

and 3.35 we can see that using χ^2 cut removes more events from the set than anomaly detection techniques. False negative rate calculated in section 3.6.4 is $\backsim$ 2% for our method. We need to inspect the pulses which are included in anomaly detection but not in χ^2 cut in future studies.

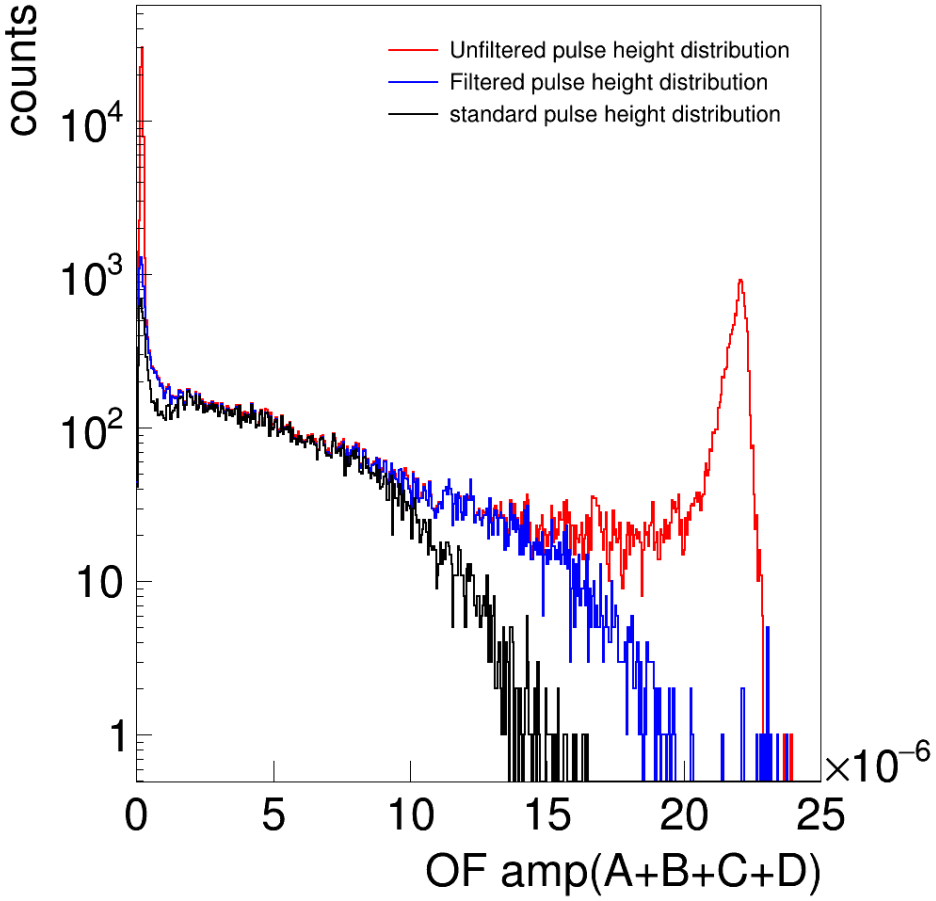

Figure 3.34: Pulse amplitude distribution for filtered and unfiltered data. The plot also includes pulse amplitude distribution using a cut on χ^2 values in black.

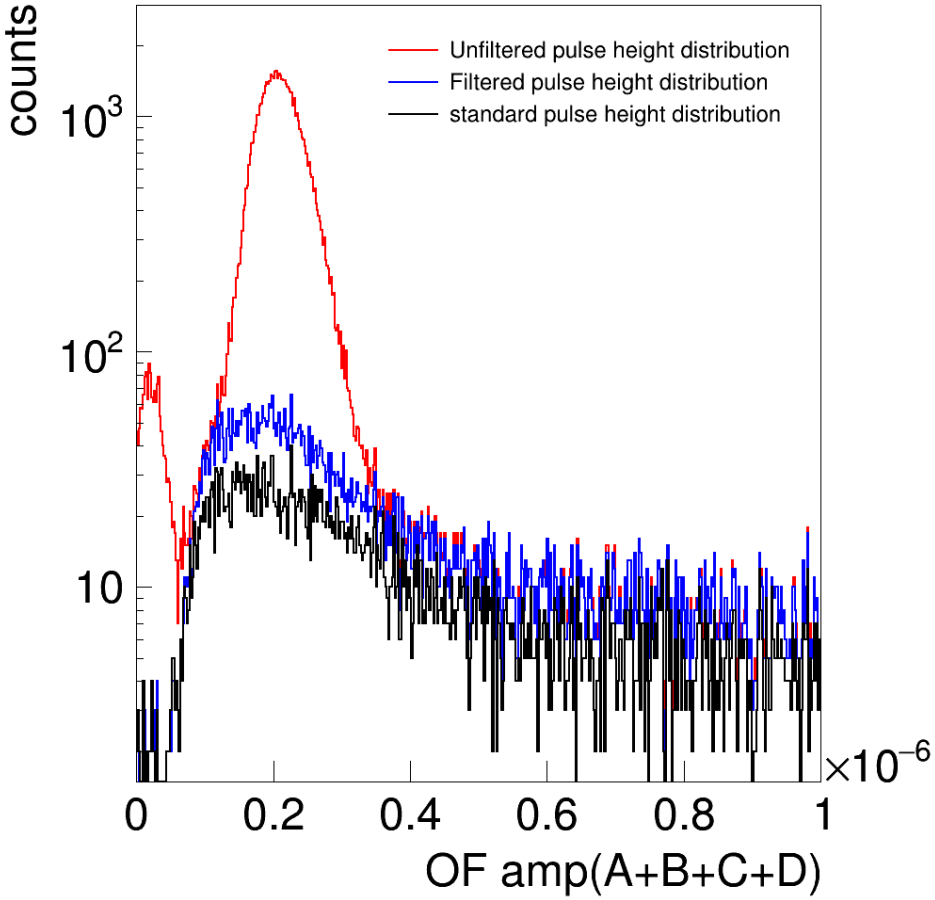

Figure 3.35: Pulse amplitude distribution for filtered and unfiltered data in the low amplitude region.The plot also includes pulse amplitude distribution using a cut on χ^2 values in black.

We can also see the difference in the χ^2 plot shown for unfiltered pulses Fig: 3.37 and filtered pulses in Fig: 3.37. The red line we see in the plot is a cut on χ^2 to remove bad pulses, this depends on the amplitude and we get this by eye estimation. We see a lot of pulses above the cut which are removed in the filtered pulses. We see the partition plot in Fig: 3.47 (a) for unfiltered data, we can see a black area and some pulses outside the triangle which is where a lot of saturated pulses are collected, as seen in the Fig: 3.47 (b) almost all pulses from these regions have been removed which were saturated leaving only good pulses. One more point to consider is if there is a noise pulse the amplitude will be similar in all three channels and according to the formula we have used this pulse will be located at (0,0). We do see a cluster at the origin in the unfiltered data plot at origin Fig: 3.47 (b), and a considerable decrease in filtered plot Fig: 3.47 (b), but some points still remain. This is due to the pulses in cluster 21 as shown in Fig 3.31. The pulses in this cluster are small pulses and have almost similar amplitudes in all three channels. They contribute to the bright spot at the origin. We can see in Fig: 3.47 (c) that is the same plot as Fig: 3.47 just without pulses in cluster 21, and we see that the spot at origin is almost gone.

Figure 3.36: Amplitude Vs χ^2 for unfiltered data

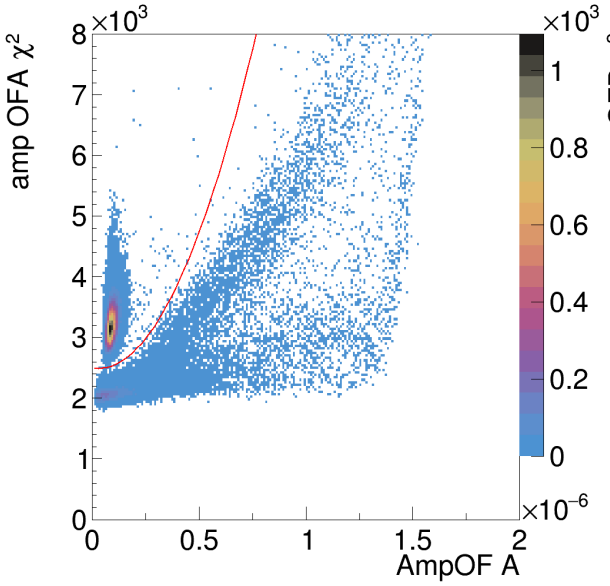

Figure 3.37: χ^2 Vs OF channel A.

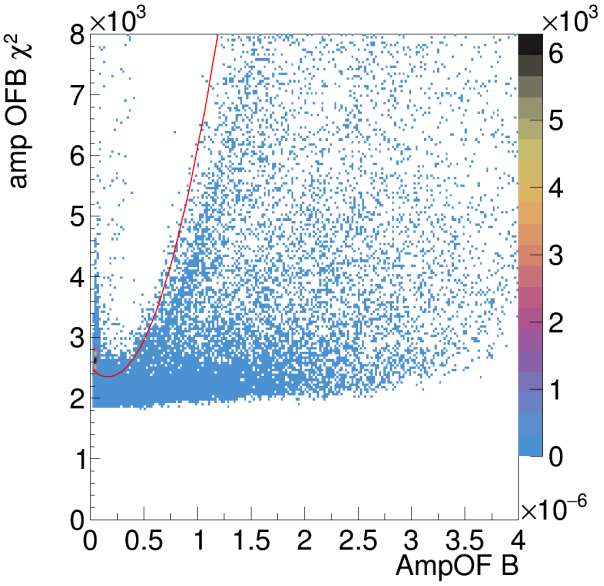

Figure 3.38: χ^2 Vs OF channel B.

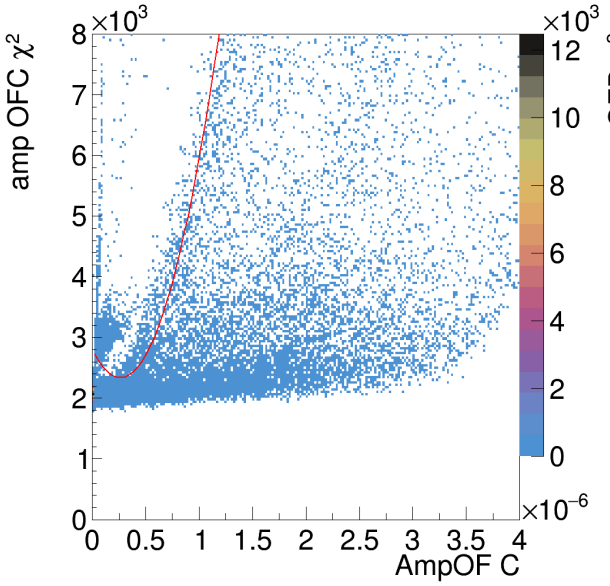

Figure 3.39: χ^2 Vs OF channel C.

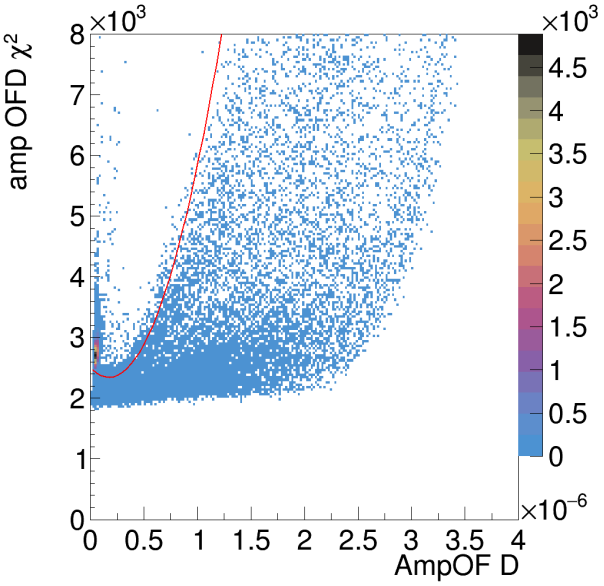

Figure 3.40: χ^2 Vs OF channel D.

Figure 3.41: Amplitude Vs χ^2 for filtered data

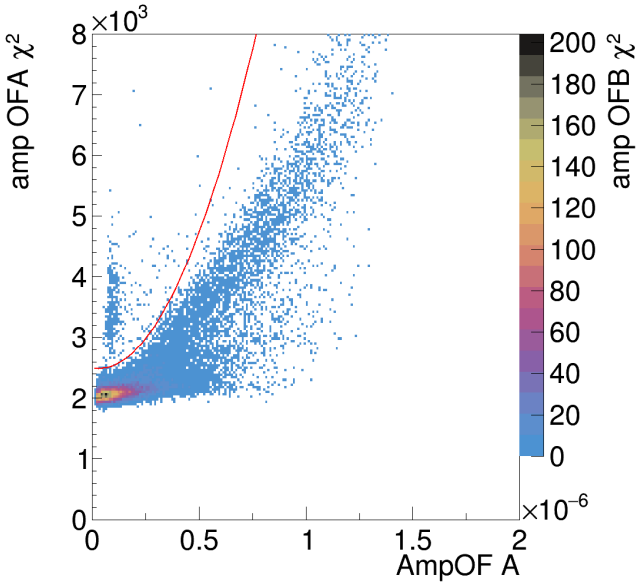

Figure 3.42: χ^2 Vs OF channel A.

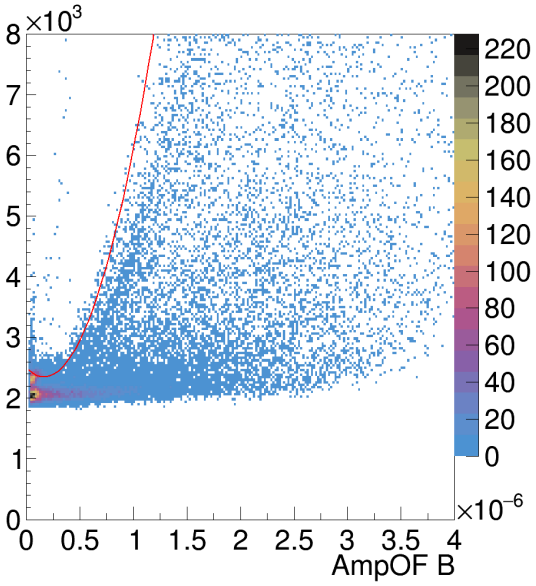

Figure 3.43: χ^2 Vs OF channel B.

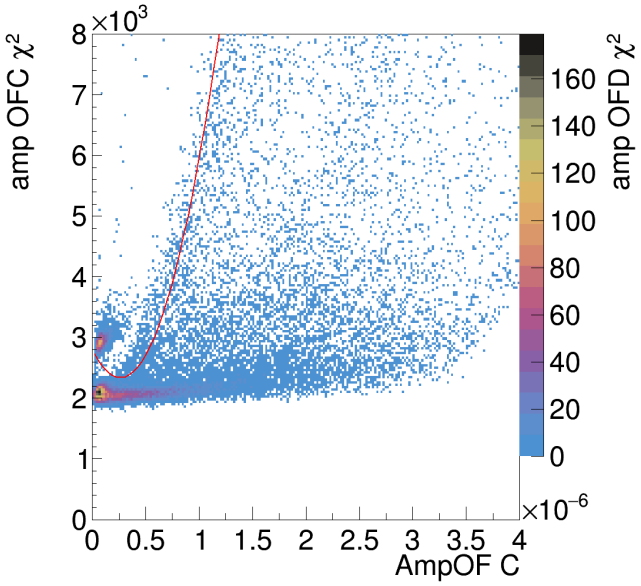

Figure 3.44: χ^2 Vs OF channel C.

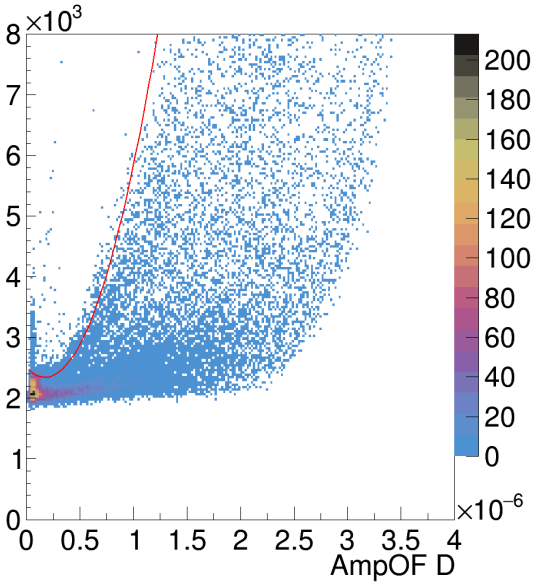

Figure 3.45: χ^2 Vs OF channel D.

Figure 3.46: Partition plot for filtered and unfiltered data

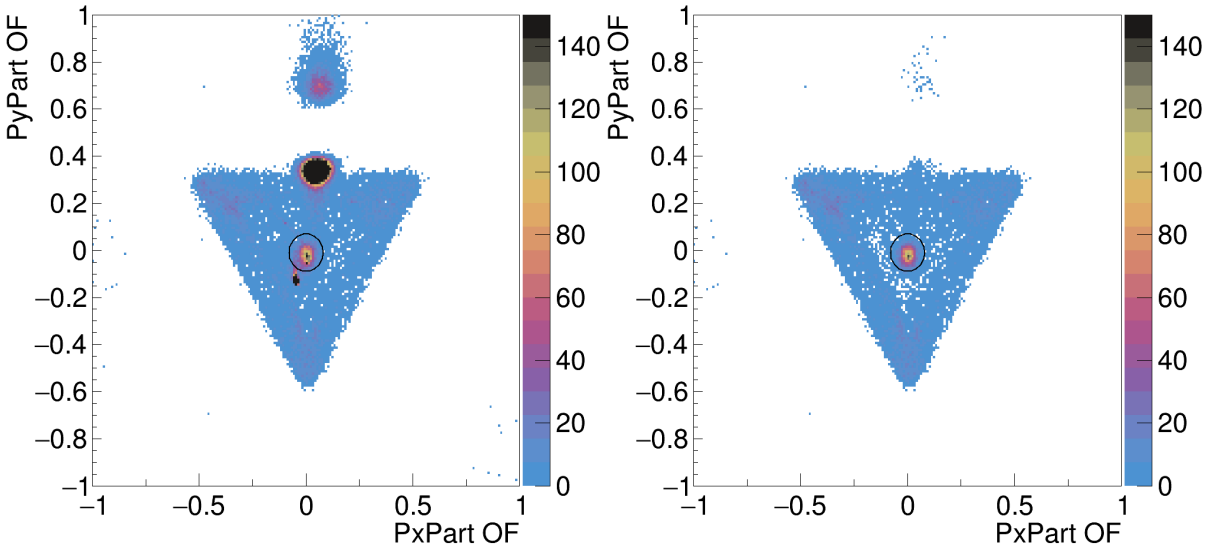

Figure 3.47: Partition plot for unfiltered data.

Figure 3.48: Partition plot for filtered data.

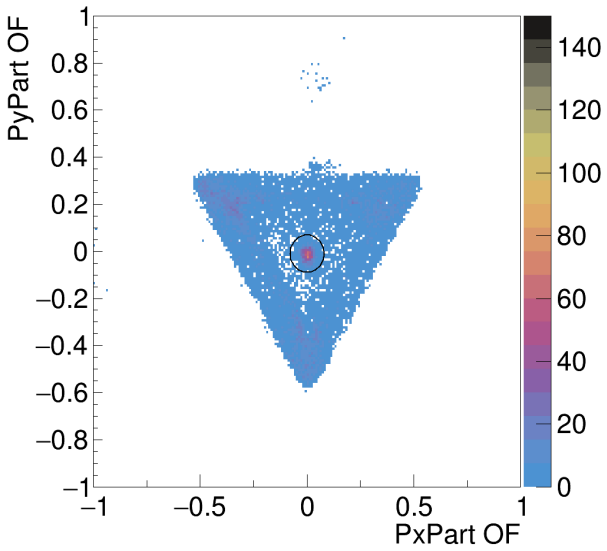

Figure 3.49: Partition plot for filtered data without cluster 21.

Chapter 4

Detector Simulation

4.1 Detector simulation

To simulate phonon pulses in a semiconductor detector we need to simulate:

1. Energy deposition mechanism in the crystal
2. Creation of primary phonon and electron/hole pairs
3. Propagation of charge and phonons through the crystal medium
4. Creation of secondary phonons due to charge propagation
5. TES readout

The first step in this simulation has already been implemented using Geant4 [29] which simulates a source and its interaction with the detector. γ particles are simulated as incoming particles with an energy distribution expected from U and Th background. A cylindrical Ge detector was used with a radius of 38 mm and a depth of 25 mm. It returns particle interactions in the detector crystal (whether electron or nuclear recoils), the amount of energy deposited, and the location where the interaction occurred. We have discussed how to get $E_{pho,primary}$, E_Q, $N_{pho,primary}$ and $N_{e/h}$ in section 2.2 Given the recoil energy from this simulation and eq 2.4 We calculated the $E_{pho,primary}$ and E_Q. Next using eq 2.5 and 2.6 $N_{pho,primary}$ and $N_{e/h}$ were also calculated.

4.2 Motion of phonons in the Crystal

After the creation of phonons in the detector, the path a phonon will take depends on its group velocity and impurities in the detector. The principle of propagation of phonons will be the same for primary and secondary phonons created due to the movement of electron-hole pairs in the lattice. We will consider two processes that will define the trajectory of a phonon: (a) scattering in the lattice due to mass defects and (b) a decay process.

(i)Impurity Scattering : The detector lattice can have impurities, for example, an isotope of the same material as the detector. A phonon can scatter off these impurities due to mass differences. The phonon could propagate in the lattice in one of the three modes in the germanium lattice : (i) longitudinal, (ii) slow-transverse, and (iii) fast-transverse. The fraction of phonons propagating in these modes fluctuates around the measured proportions of 10% longitudinal, 35% fast-transverse, and 55% slow-transverse [30] [14]. (ii)Phonon decay The phonon decays into two lower energy phonons. The decay process is called anharmonic decay and is only applicable to phonons with the longitudinal mode of propagation.

We find the isotope scattering rate of phonons in a germanium lattice from the results from [30] which depends on the frequency of a phonon and not its modes. From the rate, we calculate the next time step at which a scattering happens. During path calculation of the phonons, isotope scattering changes the direction of the phonons and their modes. We have approximated the wavelength to be in the order of μm, which is motivated by the calculation in [31]. First, a random phonon is picked at the pre-calculated time of scattering, the mode is changed such that the population fluctuates under 2% difference around the expected population of modes. The propagation direction for this phonon is now randomly chosen. A random direction is chosen by generating three random numbers and normalizing them to get a unit vector.
The decay process will create two phonons of lower energies from the incident phonons. The lifetime of this process is taken from [32]. A similar approach is implemented as scattering calculations to calculate the output of a phonon decay, first, the modes of outgoing phonons are selected randomly, making sure that the population fluctuates under a 2% difference around the expected population. Then for each phonon, a random propagation direction is selected preserving the momentum and energy of the interaction.
These phonons will keep on propagating in the lattice until either (i) they get absorbed by the TES sensors on the surface, or (ii) down converts to such low energies that the sensors can no longer detect, 340 micro eV in the case of detector discussed in Chapter 2. Finally, for all the outgoing phonons we check their coordinates and energies for absorption and repeat the process until there are no phonons left.

4.3 Results

In the plots of phonon energy Vs time, we do see a pulse where all the energy is deposited. But the pulse has a lot of fluctuations when compared to experimental data, even though we have not introduced any source of noise in our simulations. Another issue is that the width of the pulse is too small for the Ge detector typically the width of the pulse is of few 100 μS as seen in Fig: 4.2. Reasons for this might be that we are not simulating TES response for our absorbed phonons.

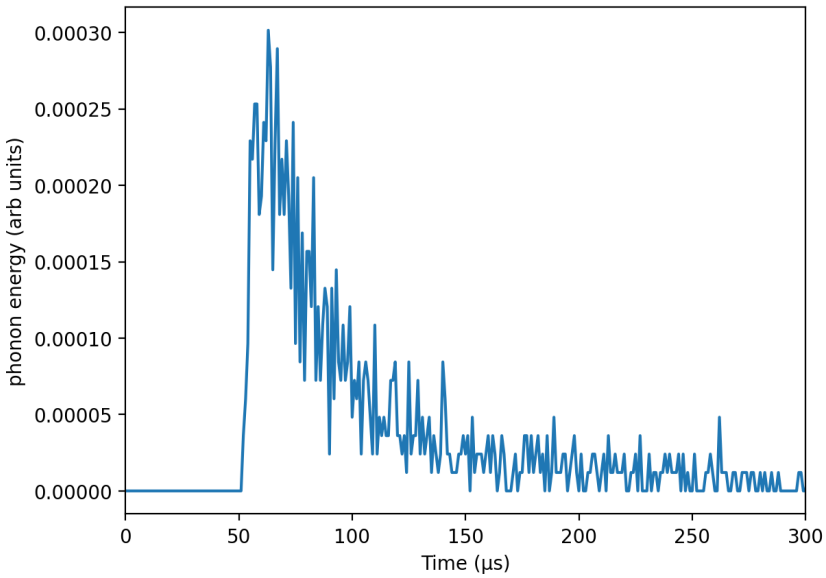

Figure 4.1: Phonon energy Vs time plot of a pulse from phonon propagation simulation.

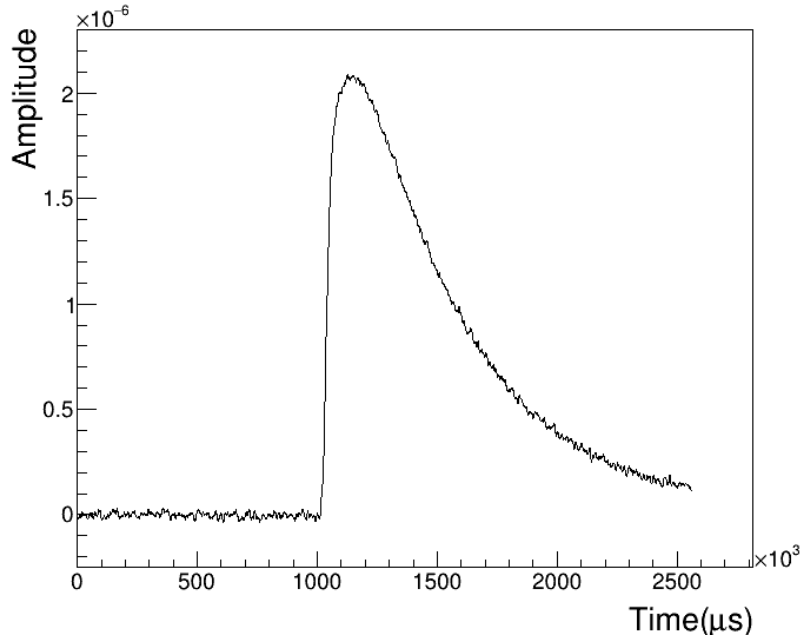

Figure 4.2: Sample pulse from a Ge detector.

We discussed how a phonon is measured using a TES sensor in section 2.5. TES sensors consist of a tungsten strip (W) n the TES sensor which is cooled to a superconducting state. When a phonon enters W it scatters and losses energy, increasing the temperature of W to cross its critical point and have normal resistance. This process is shown in Fig: 4.3 [33]. In the figure, color represents the temperature of the strip with red indicating higher temperature. After this, the strip begins to cool again (Fig: 4.3 (c)). If there is no other contribution then TES would return to its initial state with an exponential time constant [33]. We get the tail of the pulse in this phase of phonon measurement. As we are not simulating the TES response for

our absorbed phonons we do not see the full tail. The fluctuations we saw in our simulation will also be smoothed out as we will not be measuring phonon energy but voltage from TES module.

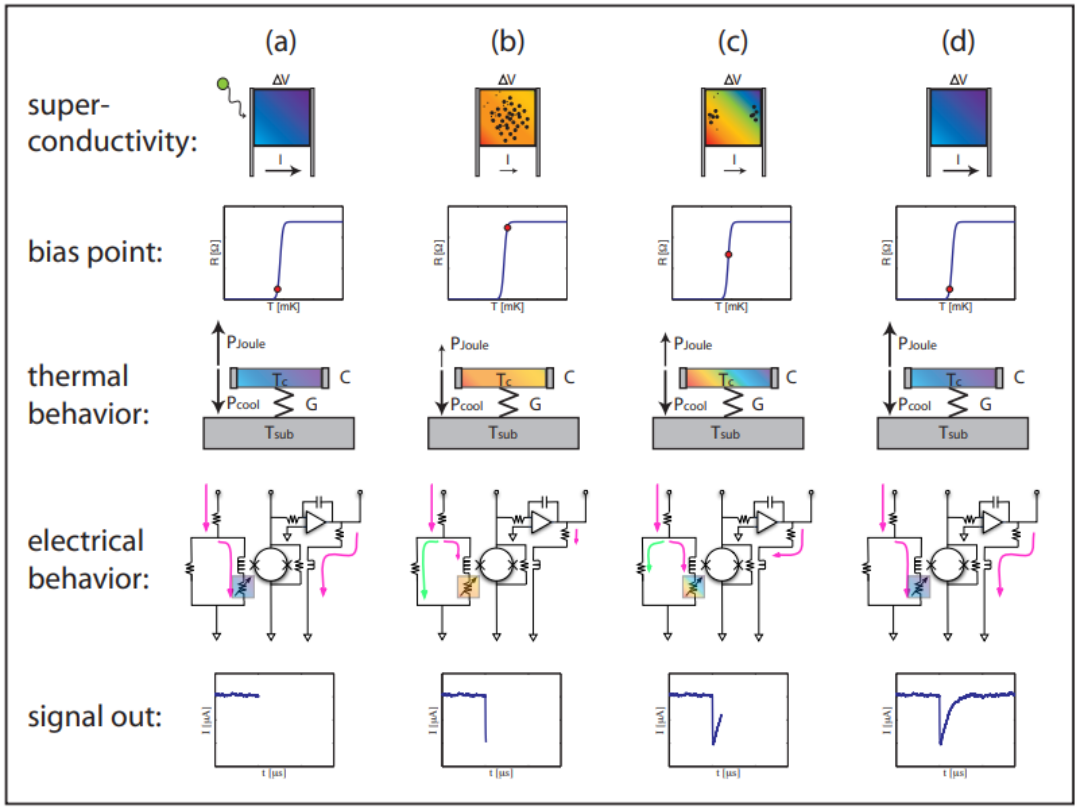

Figure 4.3: The response of the TES circuit to an event. Column (a), (b), (c) and (d) shows the stages of phonon detection. Column (c) shows the cooling of the W strip after phonon absorption when the tail of a pulse is formed. [33].

Chapter 5

Conclusion and Outlook

5.1 Anomaly detection

We tried to use t-SNE + DBSCAN to separate the good and bad pulses from a dataset taken using a sapphire detector. We saw that the combination of these two algorithms was able to cluster the data into different pulses without using labeled good and bad pulse data. They were very good at detecting pile-ups, saturated pulses, and noise. We measured that the separability of the algorithm depends on the number of points in the dataset. It was also verified in the work that this can be corrected by changing the perplexity parameter in the algorithm. We also saw that we can extract events in the mixed and pure signal clusters and run t-SNE and DBSCAN on them again, this gives us an improvement in the separability as the number of points in the dataset is decreased. We filtered the good pulses detected by the t-SNE + DBSCAN algorithm and compared them to the unfiltered dataset. We found that there were significant improvements in decreasing the number of saturated pulses and noise pulses as shown in Fig: 3.34, 3.42, and 3.47. The accuracy was manually measured for the algorithm and it was found that it detects approximately 98.2% of the good pulses. It is needed to study the different signal clusters we have got and why t-SNE divided them into different clusters.

5.2 Detector Simulation

Using the energy deposition and coordinates of an event in the detector we have calculated the number of phonons and electron/hole pairs that will be generated for an event. We studied the propagation of phonons in the lattice using the number of phonons and their energy. Two physical processes have been considered and implemented while calculating phonon propagation in the lattice : (i) scattering in the lattice due to mass defects, and (ii) phonon decay. This same propagation steps can be used to propagate type of phonons.
We obtained the absorbed phonon energy vs time plot for each event. The shape of this plot match with the pulses we have seen experimentally from the detector readout. But the pulse which we have obtained has fluctuations and lacks the smoothly decreasing trend compared to experimental data. Potential reason for this observation could be that we have not simulated

the TES response for the absorbed phonons.
In the future it can be tried to simulate the creation of secondary phonons, and relaxation phonons. To understand the creation of secondary phonons, the trajectory of the electron/hole pair has to be determined. This will require obtaining the path of least resistance for electrons and holes by solving for potential in the lattice. Consequently, the TES response of the absorbed phonons could be simulated to get the detector readout.

References

[1] E. Komatsu, K. M. Smith, J. Dunkley, C. L. Bennett, B. Gold, G. Hinshaw, N. Jarosik, D. Larson, M. R. Nolta, L. Page, D. N. Spergel, M. Halpern, R. S. Hill, A. Kogut, M. Limon, S. S. Meyer, N. Odegard, G. S. Tucker, J. L. Weiland, E. Wollack, and E. L. Wright. "SEVEN-YEAR WILKINSON MICROWAVE ANISOTROPY PROBE (WMAP) OBSERVATIONS: COSMOLOGICAL INTERPRETATION". In: *The Astrophysical Journal Supplement Series* 192.2 (Jan. 2011), p. 18. DOI: 10.1088/0067-0049/192/2/18. URL: https://doi.org/10.1088/0067-0049/192/2/18.

[2] and P. A. R. Ade et al. "Planck2015 results". In: *Astronomy & Astrophysics* 594 (Sept. 2016), A13. DOI: 10.1051/0004-6361/201525830. URL: https://doi.org/10.1051/0004-6361/201525830.

[3] Andrew Liddle. *An introduction to modern cosmology; 2nd ed.* Chichester: Wiley, 2003. URL: https://cds.cern.ch/record/1010476.

[4] *Cosmic Energy Budget*. URL: https://sci.esa.int/web/euclid/-/cosmic-energy-budget.

[5] F. Zwicky. "Die Rotverschiebung von extragalaktischen Nebeln". In: *Helvetica Physica Acta* 6 (Jan. 1933), pp. 110–127.

[6] E. Corbelli and P. Salucci. "The extended rotation curve and the dark matter halo of M33". In: *Monthly Notices of the Royal Astronomical Society* 311.2 (Jan. 2000), pp. 441–447. DOI: 10.1046/j.1365-8711.2000.03075.x. URL: https://doi.org/10.1046/j.1365-8711.2000.03075.x.

[7] Written by: Matthew Newby. *Galaxy Rotation Curve*. URL: https://sites.temple.edu/profnewby/2019/05/04/galaxy-rotation-curve/.

[8] Richard S. Ellis. "Gravitational lensing: a unique probe of dark matter and dark energy". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 368.1914 (2010), pp. 967–987. DOI: 10.1098/rsta.2009.0209.

[9] Yashar D. Hezaveh, Neal Dalal, Daniel P. Marrone, Yao-Yuan Mao, Warren Morningstar, Di Wen, Roger D. Blandford, John E. Carlstrom, Christopher D. Fassnacht, Gilbert P.

Holder, and et al. "Detection Of Lensing Substructure Using Alma Observations Of The Dusty Galaxy Sdp.81". In: *The Astrophysical Journal* 823.1 (2016), p. 37. DOI: 10.3847/0004-637x/823/1/37.

[10] *Dark Matter*. URL: https://particleastro.brown.edu/dark-matter/.

[11] Mark David Pepin. "Low-Mass Dark Matter Search Results and Radiogenic Backgrounds for the Cryogenic Dark Matter Search". PhD thesis. Minnesota, USA, 2016.

[12] P. Cushman, C. Galbiati, D. N. McKinsey, H. Robertson, T. M. P. Tait, and D. Bauer. *Planck2015 results.*

[13] R. Agnese, A. J. Anderson, T. Aramaki, I. Arnquist, W. Baker, D. Barker, R. Basu Thakur, D. A. Bauer, A. Borgland, M. A. Bowles, P. L. Brink, R. Bunker, B. Cabrera, D. O. Caldwell, R. Calkins, C. Cartaro, D. G. Cerdeño, H. Chagani, Y. Chen, J. Cooley, B. Cornell, P. Cushman, M. Daal, P. C. F. Di Stefano, T. Doughty, L. Esteban, S. Fallows, E. Figueroa-Feliciano, M. Fritts, G. Gerbier, M. Ghaith, G. L. Godfrey, S. R. Golwala, J. Hall, H. R. Harris, T. Hofer, D. Holmgren, Z. Hong, E. Hoppe, L. Hsu, M. E. Huber, V. Iyer, D. Jardin, A. Jastram, M. H. Kelsey, A. Kennedy, A. Kubik, N. A. Kurinsky, A. Leder, B. Loer, E. Lopez Asamar, P. Lukens, R. Mahapatra, V. Mandic, N. Mast, N. Mirabolfathi, R. A. Moffatt, J. D. Morales Mendoza, J. L. Orrell, S. M. Oser, K. Page, W. A. Page, R. Partridge, M. Pepin, A. Phipps, S. Poudel, M. Pyle, H. Qiu, W. Rau, P. Redl, A. Reisetter, A. Roberts, A. E. Robinson, H. E. Rogers, T. Saab, B. Sadoulet, J. Sander, K. Schneck, R. W. Schnee, B. Serfass, D. Speller, M. Stein, J. Street, H. A. Tanaka, D. Toback, R. Underwood, A. N. Villano, B. von Krosigk, B. Welliver, J. S. Wilson, D. H. Wright, S. Yellin, J. J. Yen, B. A. Young, X. Zhang, and X. Zhao. "Projected sensitivity of the SuperCDMS SNOLAB experiment". In: *Phys. Rev. D* 95 (8 Apr. 2017), p. 082002. DOI: 10.1103/PhysRevD.95.082002. URL: https://link.aps.org/doi/10.1103/PhysRevD.95.082002.

[14] JORGE DANIEL MORALES MENDOZA. "SIMULATION OF THE CHARGE MEASUREMENTS FOR THE SUPERCDMS SOUDAN EXPERIMENT". PhD thesis. Texas, USA, 2020.

[15] Jonathan L. Feng. "Dark Matter Candidates from Particle Physics and Methods of Detection". In: *Annual Review of Astronomy and Astrophysics* 48.1 (Aug. 2010), pp. 495–545. DOI: 10.1146/annurev-astro-082708-101659. URL: https://doi.org/10.1146/annurev-astro-082708-101659.

[16] Anne M Green. "Dependence of direct detection signals on the WIMP velocity distribution". In: *Journal of Cosmology and Astroparticle Physics* 2010.10 (Oct. 2010), pp. 034–034. DOI: 10.1088/1475-7516/2010/10/034. URL: https://doi.org/10.1088/1475-7516/2010/10/034.

[17] J. Lindhard and V. Nielsen. "Integral Equations Governing Radiation Effects. (Notes on Atomic Collisions, III". In: *Kgl. Danske Videnskab., Selskab. Mat. Fys. Medd* 33.10 (1963).

[18] J. F. Ziegler. "The electronic and nuclear stopping of energetic ions". In: *Applied Physics Letters* 31.8 (1977).

[19] *CDMS Detector Monte Carlo Documentation*. 2012. URL: http://titus.stanford.edu/cgi-test/cvsweb.cgi/CDMS_DetectorMC/CDMS_DMC_%20Manual.pdf..

[20] F. E. Emery and T. A. Rabson. "Average Energy Expended Per Ionized Electron-Hole Pair in Silicon and Germanium as a Function of Temperature". In: *Physical Review* 140.6A (1965). DOI: 10.1103/physrev.140.a2089.

[21] R.h. Pehl, F.s. Goulding, D.a. Landis, and M. Lenzlinger. "Accurate determination of the ionization energy in semiconductor detectors". In: *Nuclear Instruments and Methods* 59.1 (1968), pp. 45–55. DOI: 10.1016/0029-554x(68)90342-x.

[22] Zhong He. "Review of the Shockley–Ramo theorem and its application in semiconductor gamma-ray detectors". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 463.1-2 (2001), pp. 250–267. DOI: 10.1016/s0168-9002(01)00223-6.

[23] Adam Anderson. "A search for light weakly-interacting massive particles with SuperCDMS and applications to neutrino physics". PhD thesis. Massachusetts, USA, 2015.

[24] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229.

[25] N.J Nilsson. *Introduction to machine learning*. 2005.

[26] Arthur Zimek and Erich Schubert. "Outlier Detection". In: *Encyclopedia of Database Systems*. Springer New York, 2017, pp. 1–5. DOI: 10.1007/978-1-4899-7993-3_80719-1. URL: https://doi.org/10.1007/978-1-4899-7993-3_80719-1.

[27] Laurens van der Maaten and Geoffrey Hinton. "Visualizing Data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605. URL: http://www.jmlr.org/papers/v9/vandermaaten08a.html.

[28] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, and Xiaowei Xu. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *AAAI* (1996). URL: https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf.

[29] S. Agostinelli, J. Allison, K. Amako, J. Apostolakis, H. Araujo, P. Arce, M. Asai, D. Axen, S. Banerjee, et al. "Geant4—a simulation toolkit". In: *Nuclear Instruments and Methods*

in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506.3 (2003), pp. 250–303. ISSN: 0168-9002.

[30] Shin-ichiro Tamura. "Isotope scattering of dispersive phonons in Ge". In: *Phys. Rev. B* 27 (2 Jan. 1983), pp. 858–866. DOI: 10.1103/PhysRevB.27.858. URL: https://link.aps.org/doi/10.1103/PhysRevB.27.858.

[31] I. A. Kaplunov, A. I. Kolesnikov, G. I. Kropotov, and V. E. Rogalin. "Optical Properties of Single-Crystal Germanium in the THz Range". In: *Optics and Spectroscopy* 126.3 (Mar. 2019), pp. 191–194. DOI: 10.1134/s0030400x19030093. URL: http://dx.doi.org/10.1134/S0030400X19030093.

[32] *Phonon Scattering in Condensed Matter*. Springer Berlin Heidelberg, 1984. DOI: 10.1007/978-3-642-82163-9. URL: http://dx.doi.org/10.1007/978-3-642-82163-9.

[33] Jennifer Anne Burney. "TRANSITION-EDGE SENSOR IMAGING ARRAYS FOR ASTROPHYSICS APPLICATIONS". PhD thesis. USA, 2007.

Appendix A

t-SNE +DBSCAN

Listing A.1: Data Generation Script V:1.0

```
from multiprocessing import Pool
from nptdms import TdmsFile as tdms
from sklearn.manifold import TSNE
from sklearn.cluster import DBSCAN
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np
import time
import os
'''
#does work
perplex = 35#@param {type:"number"}
niter = 5000#@param {type:"number"}
lr = 250 #@param {type:"number"}
ee =  170#@param {type:"number"}
'''

perplex = 40#@param {type:"number"}
niter = 6000#@param {type:"number"}
lr = 250 #@param {type:"number"}
ee =  200#@param {type:"number"}
'''

perplex = 30#@param {type:"number"}
niter = 2500#@param {type:"number"}
lr = 200 #@param {type:"number"}
```

```
ee =  150#@param {type:"number"}

'''
# read "data" pandas dataframe
data = pd.read_pickle("dataframe/ogdata.pkl")

data_subset = data.drop(['label','slno','file_name','group_name'],axis=1)
#print(data_subset)   #sanity check
data_subset_array = data_subset.to_numpy()

time_start = time.time()
tsne = TSNE(n_components=2, verbose=1, perplexity=perplex, n_iter=niter, early_exaggeration=ee,learning_rate=lr, n_jobs=-1)
print(data_subset_array)
tsne_results = tsne.fit_transform(data_subset_array)
print('t-SNE done! Time elapsed: {} seconds'.format(time.time()-time_start))


data['tsne-2d-one'] = tsne_results[:,0]
data['tsne-2d-two'] = tsne_results[:,1]
'''
plt.figure(figsize=(16,10))
sns.scatterplot(
    x="tsne-2d-one", y="tsne-2d-two",
    hue="label",
    palette=sns.color_palette("bright", 2),
    data=data,
    legend="full",
    alpha=0.7
)
plt.savefig('plots/t-sne/t-SNE1.png')
'''


data_subset = data.drop(["slno",'file_name','prepulse_std1', 'prepulse_std2', 'prepulse_std3', 'prepulse_std4', 'postpulse_std1', 'postpulse_std2', 'postpulse_std3', 'postpulse_std4', 'max_content1', 'max_content2',
```

```
    ↪ 'max_content3', 'max_content4', 'min_content1', ↙
    ↪ 'min_content2', 'min_content3', 'min_content4', ↙
    ↪ 'max_tail1', 'max_tail2', 'max_tail3', 'max_tail4', ↙
    ↪ 'rise_time1', 'rise_time2', 'rise_time3', 'rise_time4', ↙
    ↪ 'fall_time1', 'fall_time2', 'fall_time3', 'fall_time4', ↙
    ↪ 'fwhm1', 'fwhm2', 'fwhm3', 'fwhm4', 'fw90m1', 'fw90m2', ↙
    ↪ 'fw90m3', 'fw90m4', 'fw10m1', 'fw10m2', 'fw10m3', 'fw10m4', ↙
    ↪ 'max_time_std',"group_name", "label"],axis=1)
#print(data_subset)   #sanity check
data_subset_array = data_subset.to_numpy()

clustering = DBSCAN(eps=5, min_samples=4).fit(data_subset_array)
data['clustering'] = clustering.labels_[:]

data["legend"] = data["label"] + data["clustering"].astype(str)


print(data)
a_set = set(data['clustering'])
num_cluster = len(a_set)


plt.figure(figsize=(16,10))
sns.scatterplot(
    x="tsne-2d-one", y="tsne-2d-two",
    hue="clustering",
    palette=sns.color_palette("bright", num_cluster),
    data=data,
    legend="brief",
    alpha=0.7
)
plt.savefig('plots/t-sne/DBSCAN1.png')

data.to_pickle("dataframe/t_SNE+DBSCAN_1.pkl")
```

Appendix B

Pulse Simulation

¡¿

Listing B.1: Training Script

```
'''
import outputX.root
assign variables to edep, postion and type of particle
make function for yield
save edep, position, type, e_Q, E_ph_pri, N_ph_pri, N_eh
'''

import ROOT
from array import array
from tqdm import tqdm


def epsilon(e_r):
    return 0.0115*e_r*32**(-7/3)

def g_e(eps):
    return 3*(eps**(0.15))+0.7*(eps**(0.6))+eps

def k():
    return 0.133*(32**(2/3))*72.64**(-0.5)

def Yield(type_c, e_r):
    if(type_c==0):
        return 1
    else:
        return ((k()*g_e(epsilon(e_r)))/(1+k()*g_e(epsilon(e_r))))
```

```
27
28 inFileName = "DRU_output/output1.root"
29 inFile = ROOT.TFile.Open(inFileName,"READ")
30 inTree = inFile.Get("EDTree")
31
32 outFileName = "sim_output/output1.root"
33 outFile = ROOT.TFile.Open(outFileName,"RECREATE")
34
35 Edep =  array('d', [0.])
36 x =  array('d', [0.])
37 y =  array('d', [0.])
38 z =  array('d', [0.])
39 type_c =  array('d', [0.])
40 E_Q =  array('d', [0.])
41 E_Ph_pri = array('d', [0.])
42 N_Ph_pri = array('d', [0.])
43 N_eh = array('d', [0.])
44
45 outTree = ROOT.TTree("sim_output","sim_output")
46 outTree.Branch("Edep", Edep, "Edep/D");
47 outTree.Branch("x", x, "x/D");
48 outTree.Branch("y", y, "y/D");
49 outTree.Branch("z", y, "z/D");
50 outTree.Branch("type_c", type_c, "type_c/D");      #change when you ↙
    ↪ get type
51 outTree.Branch("E_Q", E_Q, "E_Q/D");
52 outTree.Branch("E_Ph_pri", E_Ph_pri, "E_Ph_pri/D");
53 outTree.Branch("N_Ph_pri", N_Ph_pri, "N_Ph_pri/D");
54 outTree.Branch("N_eh", N_eh, "N_eh/D");
55
56
57 for entryNum in tqdm(range(0,inTree.GetEntries())):
58     inTree.GetEntry(entryNum)
59     #print(getattr(inTree,"DepEnergy"),getattr(inTree,"x"),getattr(inTree,"y"),g
60     Edep[0] = getattr(inTree,"DepEnergy")
61     x[0] = getattr(inTree,"x")
62     y[0] = getattr(inTree,"y")
63     z[0] = getattr(inTree,"z")
64     #type_c[0] = getattr(inTree,"type_c")
65     type_c[0] = 1.0
66     E_Q[0] = ↙
```

```
        ↪ getattr(inTree,"DepEnergy")*Yield(1,getattr(inTree,"DepEnergy"))
67      E_Ph_pri[0] = ↙
        ↪ getattr(inTree,"DepEnergy")*(1-Yield(1,getattr(inTree,"DepEnergy")))
68      N_Ph_pri[0] = E_Ph_pri[0]//(8.1/1000)+1
69      N_eh[0] = E_Q[0]//(2.96/1000)
70      outTree.Fill()
71
72  inFile.Close()
73  outFile.cd()
74  outTree.Write()
75  outFile.Close()
```