DIPLOMARBEIT

The NUCLEUS Experiment: Raw Data Analysis with Machine Learning Methods

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Physik

eingereicht von

Leo Maran
Matrikelnummer 01506098

ausgeführt am Atominstitut
der Technischen Universität Wien
sowie dem Institut für Hochenergiephysik
der Österreichischen Akademie der Wissenschaften

Betreuung
Univ.-Prof. Dipl.-Phys. Dr. Jochen Schieck
DI Dr. Vasile Mihai Ghete

Wien, 11.05.2025

Abstract

The NUCLEUS experiment aims to measure coherent elastic neutrino-nucleus scattering at the Chooz nuclear power plant in France, using cryogenic particle detectors. The high sensitivity required to probe the expected signal range below $100\,\mathrm{eV/c^2}$ can only be achieved using complex and challenging analysis methods. The raw detector data consists of time series of voltage traces. During measurements, the detector output stream is stored and recorded as raw data. This data are made up of a noisy baseline, on which particle events and artifacts are superimposed. Particle events have to be located in this stream in a process called triggering. The targeted particle events are identified and distinguished from artifacts by their characteristic shape, which follows a parametric description with a priori unknown parameters. To calculate the energy deposited in the detector by such an event, the scale of the pulse has to be determined. This task, called pulse height estimation, together with assessing event-shape parameters, triggering and defining quality cuts aimed to exclude unwanted artifacts are crucial steps of raw data analysis. This work aims to provide machine learning based methods for these tasks in order to speed up raw data analysis while minimizing the need for manually defined cuts that might introduce human biases. For denoising raw pulses the use of autoencoders is explored, showing promising results. Furthermore, a shape based clustering method useful for data visualization is introduced, along with a neural network based method for triggering that combines the efficiency of current methods with artifact rejection capabilities. We also present a method for an automatized creation of an optimum filter, a linear filter currently used for both triggering and pulse height estimation, using information from the data stream. Finally, we present a new weakly supervised per-detector framework for training neural networks for quality cuts. This approach matches the performance of both handmade cuts and previous machine learning methods while being more efficient and requiring shorter training times. All methods have been implemented in the CAIT analysis framework used by NUCLEUS for data analysis.

Kurzfassung

NUCLEUS ist ein Experiment zum Nachweis von kohärenter elastischer Neutrino-Nukleus Streuung mit kryogenischen Detektoren am französischen Kernkraftwerk Chooz. Der erwartete Signalbereich liegt unterhalb von $100\,\mathrm{eV/c^2}$. Um die dafür nötige Empfindlichkeit zu erreichen, müssen komplexe Methoden zur Rohdatenanalyse eingesetzt werden. Die Rohdaten des Experiments bestehen aus Zeitserien von Spannungssignalen, der gesamte Datenstrom wird gespeichert. In diesem finden sich Teilchenereignisse und Artefakte, überlagert von einem Rauschsignal. Das Lokalisieren von Ereignissen im Datenstrom wird Triggering genannt. Teilchenereignisse können durch ihre charakteristische Pulsform identifiziert und von Artefakten unterschieden werden. Die erwartete Pulsform folgt einer parametrischen Beschreibung mit a-priori unbekannten Parametern. Um die im Detektor deponierte Energie zu bestimmen, muss die Pulsamplitude ermittelt werden. Dies ist, zusammen mit der Identifizierung der Parameter der Pulsform und der Definition von Kriterien zum Ausschließen von Artefakten, Teil der Rohdatenanalyse. In dieser Arbeit stellen wir auf maschinellem Lernen basierende Methoden zur Rohdatenanalyse vor, mit dem Ziel, die Analyse zu beschleunigen und den Bedarf an manuell definierten Ausschlusskriterien zu verringern. Zur Rauschunterdrückung in Rohdaten wird der Einsatz von Autoencoder-Netzwerken untersucht, mit vielversprechenden Ergebnissen. Des Weiteren stellen wir eine formbasierte Clusteringmethode zur Datenvisualisierung vor, sowie eine auf neuronalen Netzwerken basierende Methode für das Triggering, welche gute Effizienz mit Artefaktunterdrückung kombiniert. Außerdem wird eine Methode zur automatischen Berechnung eines Optimalfilters aus Rohdaten vorgestellt. Der Optimalfilter ist ein linearer Filter, der sowohl für das Triggering als auch zur Rekonstruktion von Pulshöhen verwendet wird. Schließlich stellen wir ein Framework zum Training von neuronalen Netzen zur Artefakterkennung vor. Die mit diesem Ansatz erzielten Ergebnisse erreichen die Qualität von händisch definierten Kriterien sowie vorheriger maschineller Lernmethoden, unter Verwendung von kleineren, schneller trainierbaren neuronalen Netzwerken. Alle Methoden wurden in das CAIT-Analyseframework implementiert, welches von NUCLEUS für die Datenanalyse verwendet wird.

Contents

Introduction **1**

1 The NUCLEUS experiment **4**
1.1 Brief Introduction to CEνNS . . . 4
1.2 Experimental setup . . . 5
1.2.1 Detectors and signal shape . . . 7
1.2.2 Background and shielding . . . 9
1.3 Data used in this work . . . 10

2 Raw data analysis methods **12**
2.1 Optimum filter . . . 12
2.1.1 The standard event . . . 14
2.2 Triggering . . . 15
2.3 Pulse height estimation . . . 15
2.4 Data simulation . . . 16
2.5 Quality cuts . . . 16
2.5.1 Artifacts . . . 17

3 Machine learning methods **20**
3.1 Denoising with artificial neural networks . . . 21
3.1.1 Scaling autoencoder . . . 21
3.1.2 Noise2Noise . . . 23
3.2 Shape-based clustering with k-means . . . 27
3.2.1 The algorithm . . . 27
3.3 Triggering using neural networks . . . 28
3.3.1 Dataset creation and training . . . 30
3.3.2 Locating events . . . 30
3.3.3 Results . . . 32
3.4 Automatic creation of the standard event, optimum filter and trigger thresholds . 40
3.4.1 AutoSev . . . 40
3.4.2 AutoOf and triggering . . . 42
3.5 AutoCut . . . 45
3.5.1 Training dataset creation . . . 47
3.5.2 Model . . . 47
3.5.3 Results . . . 49

Conclusion **55**

A Robust z-score **58**

B Model definitions **59**

Introduction

Since their proposal in 1930 and first detection in 1956 [1], the study of neutrinos and their interactions has been an active and fruitful field of research. Being the only fundamental particles in the Standard Model that only interact via the weak interaction, they offer unique opportunities to test the validity of the Standard Model. Due to the small interaction rates of most observable neutrino interactions, detecting neutrinos usually relies on either very large neutrino fluxes, like in the first neutrino detection or very large detectors, like Super-Kamiokande [2] (50 000 t of water surrounded by about 11200 photomultiplier tubes), Borexino [3] (278 t of liquid scintillator), KM3Net [4] (a large volume of the order of km^3 of sea-water in the Mediterranean equipped with Cherenkov neutrino telescopes) and IceCube [5] (an equally large volume of Antarctic ice equipped with Digital Optical Modules). The measured interactions usually have interaction energies exceeding 1.8 MeV, which is the boundary for inverse beta decay, with a recent observation of a muon induced neutrino with an energy of 120^{+110}_{-60} PeV in the KM3Net experiment [6].

Recently also a lower energy neutrino interaction has been the focus of research, the coherent elastic neutrino-nucleus scattering, or CEνNS for short. It has been predicted already in 1974[7] as a direct consequence of existence of the neutral current of the weak interaction. It happens if the interaction energy of a neutrino interacting with a nucleus via the neutral current is small enough for the neutrino to interact coherently with all nucleons. Since the interaction is elastic, the only experimental signature is a low energetic nuclear recoil. This makes CEνNS difficult to observe, even though it is thought to be the dominant neutrino interaction at low energies. Because of the enhanced cross section with respect to other neutrino interactions, detectors with lower masses are possible. The challenge lies however in reaching the low energy sensitivity necessary to resolve the CEνNS signal.

Due to this difficulty, CEνNS has only been recently observed, with the first observation happening in 2017 by the COHERENT collaboration [8] using CsI[Na] and neutrinos produced by the Spallation Neutron Source (SNS) at Oak Ridge National Laboratory. Currently, more efforts are being made with different detector technologies to push accuracy. Further observations were reported on accelerator neutrinos in 2021[9] using liquid Argon, on solar neutrinos in 2024 with 2.7 sigma using liquid Xenon[10] and on reactor antineutrinos in 2025 with 3.7 sigma using high purity Germanium, available as preprint[11].

Another experiment focusing on CEνNS of reactor antineutrinos is the NUCLEUS experiment [12] to be based at the Chooz nuclear power plant in France. Currently, the experiment is under commissioning at Underground Laboratory (UGL) at Technical University Munich, with relocation to Chooz planned for this year, followed by measurements in a technical run. The experimental site will be located between the two reactor cores, at distances of 102 m and 72 m from the respective cores. Due to the proximity to the reactor cores, the neutrino flux is expected to be high enough to measure CEνNS with gram-scale detectors. The detectors will be crystals of $CaWO_4$ and Al_2O_3, equipped with transition edge sensors, kept at cryogenic temperatures. Transition edge sensors are superconductors that are kept at their transition temperature. In

this configuration, a small change in their temperature results in a big change in their resistivity. This effect can be used to measure the temperature increase in the crystals caused by a CEνNS interaction. During measurements, the detector output stream, consisting of time series of voltage traces, is stored as raw data. The stream is made up of a noisy baseline, onto which particle events and artifacts are superimposed. Particle events have to be located on this stream in a process called triggering. A theoretical model of this type of detectors [13] provides a parametric description of the shape of particle events. The values of the parameters depend on the detector and configuration, are a priori unknown and have to be inferred from the data. This description can then be used to identify particle events and to distinguish them from artifacts. In addition, the model predicts that the energy deposited in the detector by a particle event scales with the pulse height of the event pulse. Close to the low energy threshold of the detectors, noise and pulse heights have similar magnitudes, which makes triggering and pulse height estimation difficult. Due to this, usually a linear matched filter, called the optimum filter, is employed[14]. Its form can be derived by requiring the linear filter giving the best signal to noise ratio for a given signal shape and noise spectrum. It can be used for noise suppression during triggering and as pulse height estimator. During triggering, some artifacts are also typically included. To exclude them, parameters reflecting the shape of each trace are calculated. On these parameters, cuts are defined with the aim to exclude as many artifacts as possible, while excluding as few event pulses as possible. These cuts are called quality cuts.

Machine learning (ML) methods have had a surge in popularity in recent years, due to their ability to master complex tasks without explicit instructions by learning from provided data. They have applications in many fields; for cryogenic detectors, ML methods were introduced for data analysis of the CRESST experiment. CRESST [15] is an experiment searching for dark matter, which uses detector technology which is very similar to the one used by NUCLEUS. Applications in CRESST include pulse height estimation [16], quality cuts [16, 17] and detector operation [17]. This work builds heavily on these efforts, also using the same analysis framework called CAIT [18].

Chapter 1 is going to give a short introduction to the physics of CEνNS and the NUCLEUS experiment, highlighting the experimental setup and theoretical description of the detectors. In Chapter 2, some methods currently used in raw data analysis of the NUCLEUS experiment are presented. The optimum filter, its derivation and uses for triggering and pulse height estimation is described, as well as common artifacts in the data and the procedure of defining quality cuts to exclude them. Chapter 3 is concerned with machine learning methods for raw data analysis. Section 3.1 explores the use of artificial neural networks for noise reduction in detector traces, yielding promising results for triggering and event parameter estimation. Section 3.2 introduces k-means clustering, a method for clustering detector traces, which can be used for dataset visualization, artifact rejection and event classification for the construction of labelled datasets. In Section 3.3 a new method for triggering is presented. It uses a neural network to detect particle events on the data stream. Its trigger efficiency for low energy pulses almost matches the efficiency of the optimum filter, while being able to recognize and reject artifacts already in the triggering process. Section 3.4 contains a framework for the complete automatization of optimum filter triggering. This encompasses a method to infer the event pulse shape from raw data without manual intervention and subsequent automatic calculation of the optimum filter, as well as a method for the automatic determination of sensible trigger thresholds for triggering with and without the optimum filter. Finally, in Section 3.5 a framework for artifact rejection is presented. It consists of a shallow neural network, that can be trained quickly per detector or configuration in a weakly supervised manner and yields results that match previous results obtained using much larger networks, as well as manually defined cuts.

Many of the methods presented in Chapter 3 can be combined, for further automatization.

For example, combining automatized triggering, clustering and the artifact rejection network, it is possible to process a raw data stream in order to get a dataset containing only the clean events present in the raw data stream with almost no manual interventions necessary. Another example is the use of clustering to dramatically speed up the creation of a dataset that can be used to train a network for triggering, which is able to combine triggering and artifact rejection into a single step. Scripts for these procedures were implemented in CAIT, being already available to be used.

Chapter 1

The NUCLEUS experiment

1.1 Brief Introduction to CEνNS

When a neutrino interacts weakly with a nucleus via a neutral current, if the energy of the exchanged virtual Z-boson is low enough such that its corresponding wavelength is of the order of the size of the nucleus, the constituents of the nucleus are not resolved and the interaction happens coherently over all nucleons. This interaction is called CEνNS; it is a direct consequence of the existence of the neutral current of the weak interaction [7]. It is conceptually similar to the electromagnetic case of electron-proton elastic scattering (see for example [19]). If the wavelengths of the exchanged virtual photons are larger or in the order of the proton radius, the constituents of the proton are not resolved and the interaction happens coherently with the proton as a quasi-point-like particle. It is possible to include first order contributions of the finite charge distribution of the proton by the inclusion of nuclear form factors, something that can also be done in the case of CEνNS.

The full derivation of the CEνNS cross section in the Standard Model can be found in [20], but we will give a short summary of the involved steps here for a better understanding of the steps and approximations involved.

In the Standard Model, the neutral current at the vertices is given by

$$J^{\mu}_{NC} = 2\sum_{f} g^{f}_{L} \bar{f}_{L}\gamma^{\mu} f_{L} + g^{f}_{R} \bar{f}_{R}\gamma^{\mu} f_{R} \tag{1.1}$$

where f stands for all elementary fermions, f_L and f_R their left and right handed components, and g^f_L and g^f_R for their respective couplings to the Z-boson, determined by charge and weak hypercharge. Since the Z-boson has a mass of around 91 GeV/c^2, for small momentum transfers its propagator becomes constant, and can be approximated with the fermi constant G_F divided by $\sqrt{2}$. With this, the amplitude of coherent neutrino nucleus scattering can be written as

$$i\mathcal{M}(\nu + N \to v + N) = -i\sqrt{2}G_F \langle N(k_2)|J^{\mu}_{NC}|N(p_2)\rangle \langle \nu(k_1)|J_{NC\mu}|\nu(p_1)\rangle \tag{1.2}$$

where p_1, k_1, p_2 and k_2 are the momenta of the initial neutrino, final neutrino, initial nucleus and final nucleus, respectively.

The part concerning the nucleus only depends on the quark sector in the neutral current, since the nucleus is made up of up and down quarks. To simplify the expression it is assumed that the nucleus does not violate parity, and to evaluate it some known electromagnetic properties of the nucleus are considered, introducing form factors $F(q^2)$ as function of the momentum transfer q

and the weak charge of the nucleus Q_W. The assumption of parity is satisfied in the sense that we statistically expect large nuclei to approximately have 0 spin.

The cross section can then be evaluated by considering that, since there are only left handed neutrinos in the Standard Model, the right handed case should vanish and with the use of kinematic relations. One obtains

$$\frac{d\sigma}{dE_R} = \frac{G_F^2}{4\pi} Q_W^2 F^2(q^2) m_N \left(1 - \frac{E_R}{E_R^{max}}\right) \tag{1.3}$$

where m_N is the mass of the target nucleus, E_R is the nuclear recoil energy, and E_R^{max} is the maximum recoil energy determined by kinematics and given by $E_R^{max} = 2E_\nu^2/(m_N + 2E_\nu)$, with the energy of the neutrino E_ν. The nuclear weak charge is given by $Q_W = N - Z\left(1 - 4 \cdot \sin^2\theta_W\right)$ with N the number of neutrons, Z the number of protons and θ_W the Weinberg angle.

The reactor antineutrinos targeted by NUCLEUS mainly stem from beta decays of fission products [21], which means they usually have energies below 10 MeV. The exchanged momentum q between a neutrino and a nucleus is associated with a length scale given by h/q. For a neutrino with an energy of 10 MeV, this value is in the order of 100 fm if its entire momentum is transferred in the interaction. This is well above the typical nuclear radii of around 5 fm, meaning that a neutral current interactions will be in the coherent regime, with form factor close to unity.

CEνNS is also the dominant neutrino interaction at energies less than $\sim$ 100 MeV. Because of the large cross section compared to other neutrino interactions, measurements may be done with smaller detectors. Instead of the typical tonnes or kilotonnes, detector masses can be in the order of grams or kilograms. Nevertheless observing CEνNS is not trivial, since the only experimental signature is a nuclear recoil with relatively low energy. The situation is further complicated by the fact that the CEνNS cross-section increases with the size of the target nucleus, whereas the recoil energy decreases for heavier nuclei.

The precise measurement of CEνNS is of great theoretical interest, since it is thought to be sensitive beyond the Standard Model physics[22, 23, 20, 24], including to sterile neutrinos and non-standard interactions.

1.2 Experimental setup

The NUCLEUS experiment [12] aims to measure CEνNS of reactor antineutrinos at the Chooz nuclear power plant, located in the north of France. The plant consists of two pressurized water reactors, each running at a nominal power of 4.25 $GW_{thermal}$. These reactors provide a steady flow of electron antineutrinos, mainly originating from beta decays of ^{235}U, ^{239}Pu, ^{241}Pu and ^{238}U [25]. The experiment will be located at the Very Near Site (VNS), a room in the basement of a building inside the protected area of the power plant, situated between the two reactor cores at distances of 72 m and 102 m of the two cores. Due to space restrictions and restricted access in the protected area, the experiment needs to be compact and low maintenance, thus a closed cycle 'dry' cryostat is employed. To reach the temperatures needed with such a cryostat, new vibration decoupling technologies have been devised [26]. See Fig. 1.2 for a schematic overview of the detector components and shielding.

Two measuring phases have been envisioned for the NUCLEUS experiment at Chooz: NUCLEUS-10 g, with a total detector mass of 10 g, which is planned to start operations soon, followed later by a second stage called NUCLEUS-1 kg with a detector mass of 1 kg.

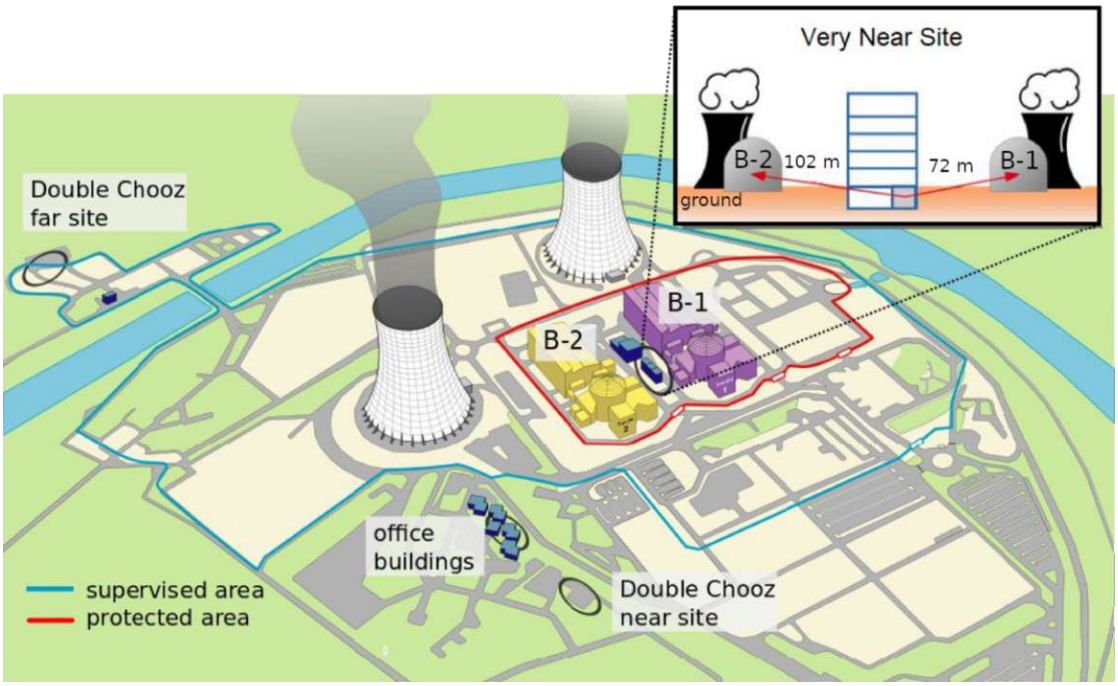


Figure 1.1: Schematic map of the Chooz power plant showing the location of the VNS. Graphic taken from [27].

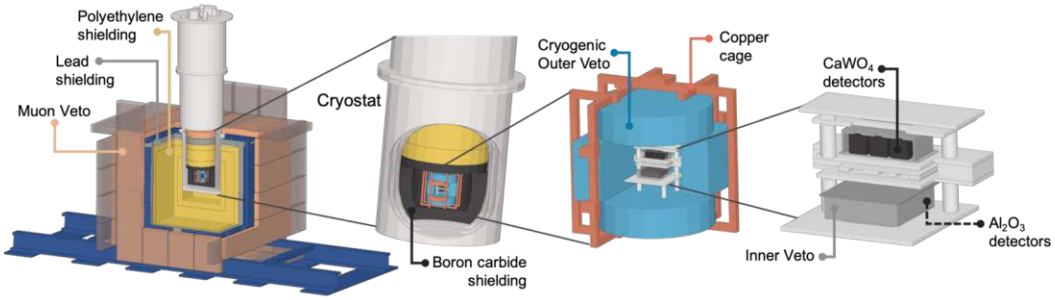


Figure 1.2: Schematic overview of the detector and its shielding, taken from [27].

1.2.1 Detectors and signal shape

The only experimental signal of CEνNS is a low energetic nuclear recoil. In cryogenic calorimeter detectors like the ones used by NUCLEUS, this recoil is detected by measuring the resulting temperature increase in the detector. The rise in temperature is proportional to the deposited energy via

$$\Delta T = \frac{\Delta E}{C} \tag{1.4}$$

where C denotes the heat capacity of the object and ΔE and ΔT are the changes in energy and temperature. This means that for increased sensitivity, small values of C are desirable in a detector. The Debye model of solid state physics (see for example ref. [28]) predicts that for low temperatures the heat capacity in a solid scales like $C \propto T^3$. Thus, detectors with high sensitivity usually operate at cryogenic temperatures. Since the heat capacity increases with mass, Eq. 1.4 also implies that small detectors are paramount to high sensitivities. The need for high sensitivity has however to be balanced against considerations of interaction probabilities. From Eq. 1.3 it is clear that the CEνNS cross section scales with the mass of the target nucleus, and the interaction probability in a detector also scales with its size. Luckily, there are some subtleties involved when applying Eq. 1.4 to cryogenic detectors, namely one has to consider electron and phonon subsystems that are involved separately.

The detectors of NUCLEUS consist of $CaWO_4$ and Al_2O_3 crystals, equipped with transition-edge sensors (TES) [29], see Fig. 1.3 for a picture of one such crystal. The working principle of these sensors is based on the phase transition between superconducting and normal conducting states. Superconductivity is a phenomenon that arises in some materials at very low temperatures, where below a critical temperature T_c the resistivity of the material rapidly drops to zero. If the superconductor is kept at its critical temperature, a relatively small change of temperature can cause a big change in resistance. A TES is made up of a superconductor operating at its critical temperature. By monitoring its resistance, a TES effectively acts as a very sensitive thermometer. See Fig. 1.4 for an example of a TES transition curve. For the TES readout, a circuit containing a super conducting quantum interference device (SQUID) is used.

It is important to note that the temperature a TES measures is determined by the temperature of its electron subsystem. To get an understanding of how the detector works, it is therefore necessary to address how the different phonon and electron subsystems of the TES and the detector crystal interact. The arguments presented here outline the ones made in ref. [13], see the reference for a more detailed and nuanced description of the detector model.

A neutrino interacting via CEνNS in the detector produces mainly optical phonons that rapidly decay into non-thermal acoustic phonons. In the relevant timescales, not many phonons thermalize in the bulk of the detector crystal. Instead, thermalization happens mostly in the TES and on the crystal surface, leading to two different time-depended signal components. To describe the influence this has on the detector signal, the thermal model seen in Fig. 1.5 is used. The different subsystems considered by it are the phonons in the absorber, the electrons in the TES and the heat bath. The phonons in the TES are neglected, since their heat capacity is very low. For the scope of the overview given here, it is also not necessary to include the thermal conductances that involve the phonons in the TES, instead only the resulting conductance G_{ea} between electrons in the thermometer and phonons in the crystal is considered.

After a particle interaction at the time t_0, the non-thermal phonons entering the TES are efficiently absorbed and thermalized by the electronic system, leading to a power input $P_e(t)$ that raises the electronic temperature of the thermometer. The rise in temperature in the crystal caused by phonons thermalizing on the surface is described by a power input $P_a(t)$. By noting that the temperature that determines the change in resistance in the TES, and thus the output signal, is the temperature of the electrons in the thermometer T_e, one can already explain

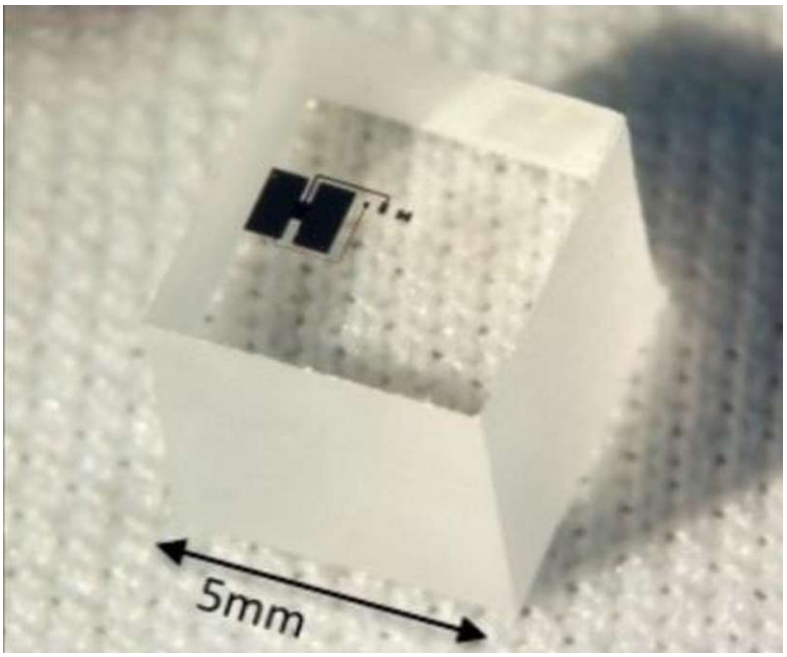


Figure 1.3: Picture of a Al_2O_3 detector crystal with a TES, taken from ref. [27]

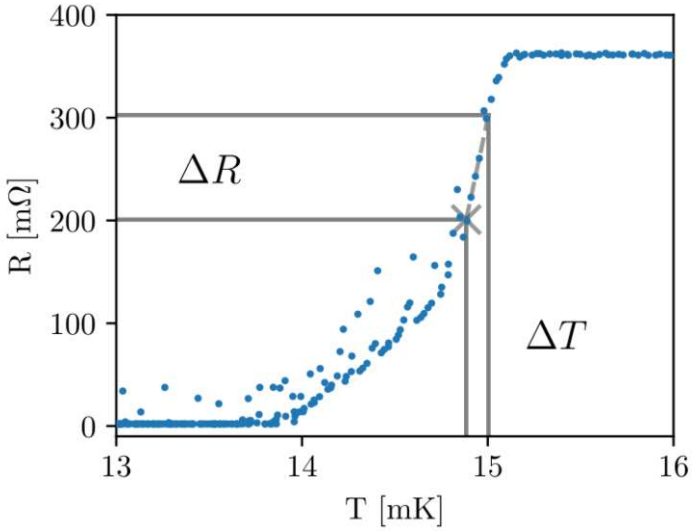


Figure 1.4: Transition curve of a Tungsten TES, taken from ref. [30]. The working point of the detector is marked by the gray x. In the region marked by ΔR and ΔT the relationship between resistance and temperature is approximately linear.

qualitatively that the power input $P_e(t)$ will lead to a signal change immediately, while the effects of $P_a(t)$ first have to pass through G_{ea} to have an effect on the output. In accordance with this, a fast and a slow component of the signal are observed experimentally. If we assume that a fraction ϵ of the high frequency phonons is thermalized in the thermometer and the rest in the crystal, we can write

$$P_e(t) = \Theta(t - t_0)\, \epsilon P_0 e^{-\frac{t-t_0}{\tau_n}} \tag{1.5}$$

and

$$P_a(t) = \Theta(t - t_0)\, (1 - \epsilon) P_0 e^{-\frac{t-t_0}{\tau_n}} \tag{1.6}$$

with the Heaviside function $\Theta(t)$, the time constant of the termalization of the high frequency phonons τ_n and the initial power input $P_0 = \frac{\Delta E}{\tau_n}$.

To get more information about the expected shape of the pulses, one has to consider the coupled equations for the temperature of the electrons in the thermometer T_e and the phonons in the absorber T_a:

$$C_e \frac{\mathrm{d}T_e}{\mathrm{d}t} + (T_e - T_a)G_{ea} + (T_e - T_b)G_{eb} = P_e(t) \tag{1.7}$$

$$C_a \frac{\mathrm{d}T_a}{\mathrm{d}t} + (T_a - T_e)G_{ea} + (T_a - T_b)G_{ab} = P_a(t) \tag{1.8}$$

with the heat capacities C_e and C_a of the electrons in the thermometer and the phonons in the absorber, and the thermal conductances G_{ea}, G_{eb} and G_{ab} as indicated in Fig. 1.5. With the initial condition $T_a(t = 0) = T_e(t = 0) = T_b$ and using Eqs. 1.5 and 1.6 one can solve Eqs. 1.7 and 1.8 to get the following expression for the thermometer signal $\Delta T_e = T_e(t) - T_b$:

$$\Delta T_e(t) = \Theta(t - t_0) \left[A_n \left(e^{-\frac{t-t_0}{\tau_n}} - e^{-\frac{t-t_0}{\tau_{in}}} \right) + A_t \left(e^{-\frac{t-t_0}{\tau_t}} - e^{-\frac{t-t_0}{\tau_n}} \right) \right] \tag{1.9}$$

Here A_n and A_t denote the amplitudes of the fast "non thermal" and the slower "thermal" component, τ_n and τ_t are the associated timescales and τ_{in} is an intrinsic timescale of the thermometer. For the full expressions, showing how these parameters connect to the parameters of the thermal model see ref. [13]. Experiments show that measured pulses are well described by Eq. 1.9.

The TES of the NUCLEUS experiment are designed to have a large value of A_n and $\tau_{in} \gg \tau_n$. In this case, the detector operates in the so called calorimetric mode and integrates the non-thermal input signal over time τ_{in}. The amplitude of the whole output signal is therefore expected to be proportional to the input energy ΔE. The proportionality can however only hold as long as the temperature change is proportional to the change in resistance in the TES, i.e. as long as the temperature stays within the boundaries indicated in Fig. 1.4. If the temperature exceeds this range, the detector goes into saturation, and pulses are distorted.

1.2.2 Background and shielding

The detectors of NUCLEUS will be deployed at the "Very Near Site" (VNS), a 24 m^2 room that is located in the basement of an office building in the protected area, between the two reactors, located at a distance of 72 m and 102 m from the reactor cores. At this distance, no relevant neutron background from the reactors is expected. Due to the relatively shallow overburden, cosmic rays and particles originating from cosmic ray air showers pose a relevant source of background, especially high energy neutrons are hazardous, since they can produce detector signals that closely mimic CEνNS events. To address the background, a sophisticated multilayer shielding strategy is employed. The outermost layers of the shielding are made up of a plastic scintillator based muon veto, followed by layers of low radioactivity Pb and borated high density

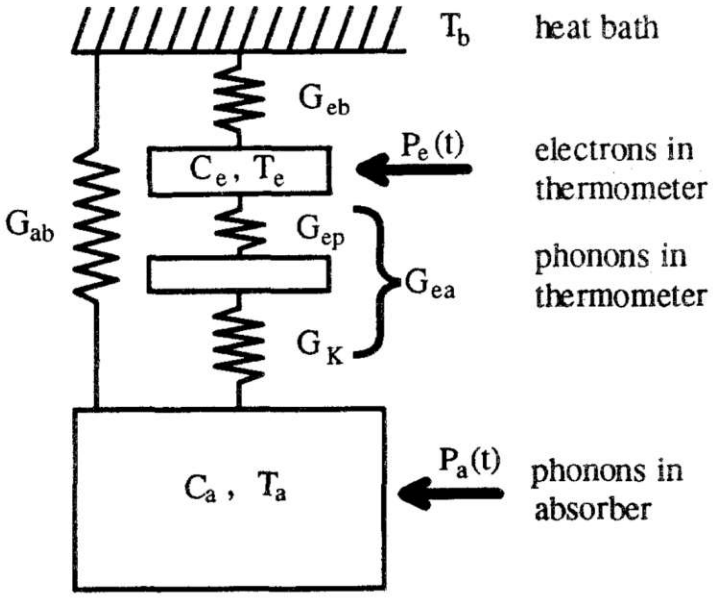


Figure 1.5: Thermal model of the detector, taken from ref. [13]. See text for description.

polyethylene. This configuration provides efficient rejection of muon-induced background events while minimizing the production of secondary particles by high-Z materials and attenuating ambient gamma rays and neutrons. To ensure 4π coverage of the shielding, these layers are continued inside the cryostat. Additionally, an almost 4π layer of boron carbide is present inside the cryostat, to provide further neutron attenuation. The detectors are surrounded by a cryogenic outer veto made of high purity Ge crystals, which acts to further reduce gamma ray as well as neutron and muon induced backgrounds. Finally, the detector holders are instrumented with TES and act as an inner veto to reject surface events and events related to the detector holder.

As mentioned in Section 1.2.1, two different detector materials will be employed, $CaWO_4$ and Al_2O_3. As seen in Eq. 1.3, the CEνNS cross-section increases with the mass of the target nucleus and its weak charge. Thus, the CEνNS rate on $CaWO_4$ is greatly enhanced with respect to Al_2O_3, while fast neutrons are expected to induce similar signatures in both materials due to scattering on O nuclei. This allows for an efficient experimental neutron induced background characterization during data taking, as no significant observation of CEνNS is expected on Al_2O_3. See Fig. 1.6 for a plot of expected CEνNS rates in different materials at the VNS. Note that it is not straightforward to calculate the exact shape of the reactor antineutrino spectrum, since many different nuclides contribute, and the initial fuel composition as well as the thermal history have to be taken into account. Furthermore, measurements only exist for the region above the threshold for inverse beta decay of around 1.8 MeV. Nonetheless, there are efforts to calculate the reactor neutrino flux also at lower energies [21, 31].

1.3 Data used in this work

During data taking the whole data stream is stored as raw data. For detector monitoring a heater pulse of varying amplitude is injected into the detector at regular intervals. The resulting

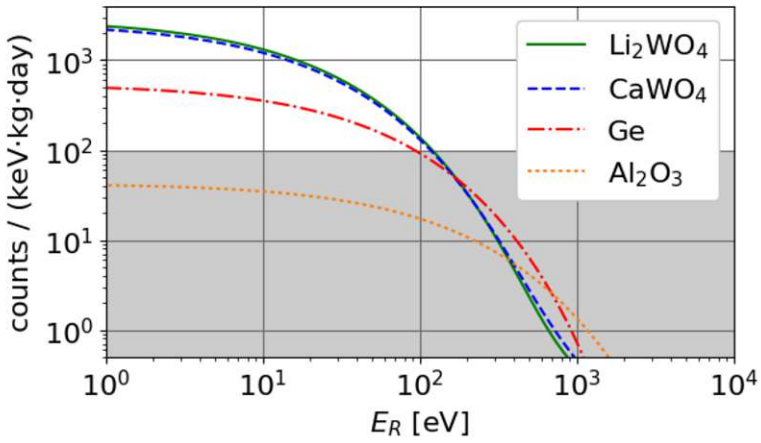


Figure 1.6: Expected CEνNS rate for different materials, calculated by using the expected neutrino flux at the VNS. The gray band indicates the background level hoped to be achieved by the NUCLEUS experiment. Graphic taken from [12].

pulses are called "test pulses". To identify these test pulses, additionally to the TES readout ADC channel, the heater input DAC channel is stored. The data used in this work were acquired during two runs: a commissioning run of eight weeks at TU Munich during Summer 2024, internally called the Long Background Run (LBR) [32] and the run that resulted in the observation of a nuclear recoil peak originating in neutron capture and the publication of ref. [33], called "Run 29". In Run 29, data were recorded with a $CaWO_4$ detector crystal, while LBR included data taking with a $CaWO_4$ detector, and a Al_2O_3 detector. In this work, four chunks of LBR data are used, each corresponding to around 100 h of data taking. Two of these chunks were taken using one $CaWO_4$ detector, with a sampling rate of 10 kHz. Internally and in the course of this work they are referred to as configuration 4 and 8. The other two chunks were taken with a double-TES equipped Al_2O_3 crystal and a sampling rate of 50 kHz, referred to as configuration 5 and 9. In between measurement chunks taken with the same detector, there is an interval of around a week. For the scope of this work, only data from the central cryogenic detectors are considered, since the dead time introduced by vetoes operating in anti-coincidence is integrated into the analysis process after the raw data analysis steps discussed here.

Chapter 2

Raw data analysis methods

The raw data stream of the cryogenic detectors of the NUCLEUS experiment consists of a time series of voltage traces. Particle events in the stream are expected to follow the parametric description of Eq. 1.9, superimposed with noise. Additionally, artifacts and noise fluctuations are present on the stream. Artifacts can have different shapes and are caused e.g. by highly energetic events or the detector readout electronics. During raw data analysis particle events have to be located on the stream in a process called triggering. During triggering the stream is searched for timespans where its value surpasses a threshold. One aims to set this trigger threshold in order to trigger as many particle events as possible while minimizing the amount of noise triggers, i.e. traces where the threshold is passed by a noise fluctuation. All traces that pass the threshold but do not contain particle event pulses at the correct time are called artifacts. In order to prevent them from distorting the results, they have to be excluded. In order to do that, so called quality cuts are defined on parameters describing the shape of traces. Finally, to assign an energy to each event pulse, its true pulse height has to be estimated.

2.1 Optimum filter

For triggering and for pulse height estimation in noisy conditions a noise suppressing filter is very useful. As described in Section 1.2.1 the expected signal shape is constant and proportional to the injected energy. If the power spectrum of the noise is also known, a matched filter called optimum filter [14, 34] can be calculated. It is a linear time-invariant filter that maximizes the signal to noise ratio (SNR). Given a measurement

$$x(t) = s(t) + n(t) \tag{2.1}$$

that consists of the target signal $s(t)$ and some noise $n(t)$, the output of a linear filter is given by the convolution of the $x(t)$ with the filter $h(t)$. Using the convolution theorem, this can be expressed as the inverse Fourier transform of the product of Fourier transformed signal and filter:

$$x_F(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H(\omega)\tilde{x}(\omega)e^{i\omega t}\, d\omega \tag{2.2}$$

where $H(\omega)$ and $\tilde{x}(\omega)$ denote the Fourier transforms of $h(t)$ and $x(t)$.

The SNR at time τ is given by the power of the target signal component at that time, divided by the power of the noise component at that time. With Eqs. 2.1 and 2.2 we can easily write

the filtered target signal as

$$s_F(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} H(\omega)\tilde{s}(\omega)e^{i\omega t}\, d\omega \tag{2.3}$$

with the Fourier transformed input signal $\tilde{s}(\omega)$. The signal power at time $t = \tau$ is therefore simply given by $s_F^2(\tau)$.

Since the noise is stochastic we cannot apply the same reasoning to get the desired expression, however by identifying that the expected power of the noise $\mathbb{E}[|n|^2]$ is equal to its autocorrelation $R_{XX}(0)$ we can use the Wiener-Khinchin theorem to write

$$\mathbb{E}[|n|^2] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} N(w)\, d\omega \tag{2.4}$$

with N being the noise power spectrum. The expected power of the filtered noise $\mathbb{E}[|n_F|^2]$ can be determined by applying the filter. For it we get

$$\mathbb{E}[|n_F|^2] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} |H(\omega)|^2 N(\omega)\, d\omega \tag{2.5}$$

Using Eqs. 2.3 and 2.5 we can identify the SNR to be

$$SNR = \frac{1}{\sqrt{2\pi}} \frac{|\int_{-\infty}^{\infty} H(\omega)\tilde{s}(\omega)e^{i\omega\tau}\, d\omega|^2}{\int_{-\infty}^{\infty} |H(\omega)|^2 N(\omega)\, d\omega} \tag{2.6}$$

This expression can be rewritten to

$$SNR = \frac{1}{\sqrt{2\pi}} \frac{|\int_{-\infty}^{\infty} (H(\omega)N(\omega)^{\frac{1}{2}})(\tilde{s}(\omega)N(\omega)^{-\frac{1}{2}}e^{i\omega\tau})\, d\omega|^2}{\int_{-\infty}^{\infty} |H(\omega)|^2 N(\omega)\, d\omega} \tag{2.7}$$

which allows for the identification of an upper bound of the SNR using the Cauchy-Schwarz inequality:

$$SNR \leq \frac{1}{\sqrt{2\pi}} \frac{\{\int_{-\infty}^{\infty} |H(\omega)|^2 N(\omega)\, d\omega\}\{\int_{-\infty}^{\infty} |\tilde{s}(\omega)|^2 N(\omega)^{-1}\, d\omega\}}{\int_{-\infty}^{\infty} |H(\omega)|^2 N(\omega)\, d\omega} \tag{2.8}$$

Putting together Eqs. 2.6 and 2.8 gives the simplified form of the inequality

$$\frac{|\int_{-\infty}^{\infty} H(\omega)\tilde{s}(\omega)e^{i\omega\tau}\, d\omega|^2}{\int_{-\infty}^{\infty} |H(\omega)|^2 N(\omega)\, d\omega} \leq \int_{-\infty}^{\infty} \frac{|\tilde{s}(\omega)|^2}{N(\omega)}\, d\omega \tag{2.9}$$

The signal to noise ratio reaches its maximum if the two sides are equal, which happens if the filter is

$$H(\omega) = K\frac{\tilde{s^*}(\omega)}{N(\omega)}e^{-i\omega\tau} \tag{2.10}$$

with a normalization constant K. This filter is called the optimum filter. Intuitively, it can be thought of as a frequency filter that suppresses noise frequencies but not signal frequencies. Due to this property it can be used for triggering and as pulse height estimator.

It has to be noted however that it does not preserve the shape of target events. Also, when it is applied to finite time traces, usually a window function is applied to the event before filtering. This is to avoid edge effects that might arise otherwise. Also, the edges of the filtered events are disregarded, which is again due to edge effects arising. To illustrate the effects of the optimum filter and possible edge effects that can arise due to its application consider Fig. 2.1.

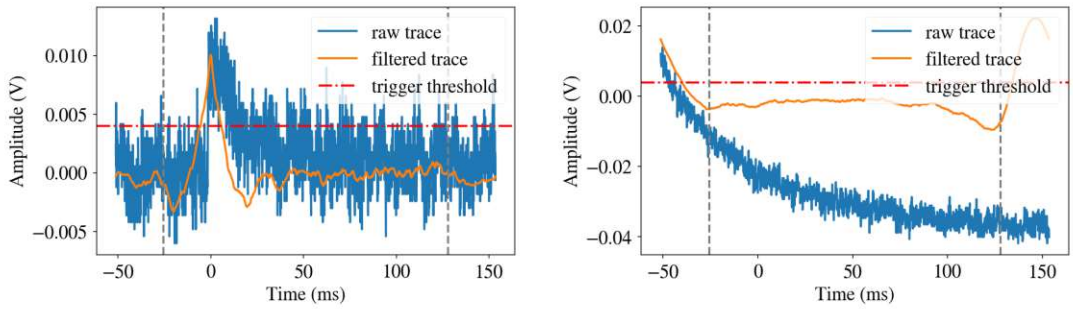


Figure 2.1: Illustration of the effects of the optimum filter. On the left, a particle event is shown. It is clearly visible how noise gets suppressed by the filter, with only the filtered event surpassing the trigger threshold. On the right, the tail of a large particle pulse is shown, illustrating the edge effects that can arise when the optimum filter is applied. The grey dashed lines in both plots indicate the edges of the filtered events that are excluded during triggering.

2.1.1 The standard event

For the calculation of the optimum filter, in addition to the average noise power spectrum, the signal shape must be known. As seen in Section 1.2.1, the expected pulse shape stays constant up to the saturation limit, scaling only linearly with energy. To find the signal shape corresponding to a detector at its current working point, strict cuts are applied to the triggered events, with the scope of selecting only traces with well-defined pulses.

If we assume that all the measured events y are well aligned in time, if every trace is scaled to the same value, we can treat each trace as a different measurement of the same ground truth. This ground truth corresponds to the sought-after detector pulse shape $z = (z_1, ..., z_n)$, consisting of n samples. To find the value of z for each sample, one has to minimize the task

$$\arg\min_z \mathbb{E}[L(z, y)] \tag{2.11}$$

where L denotes a loss function, acting as a measure of distance. In this formulation it is also apparent, that the task can be interpreted as a maximum likelihood estimation, with L being the negative log likelihood. With the commonly used L_2 norm $L(z, y) = (z - y)^2$, the expression is minimized by

$$z = \mathbb{E}[y] \tag{2.12}$$

corresponding to the arithmetic mean.

Intuitively this means that since the noise is assumed to be uncorrelated, the noise contributions are averaged out. This average pulse is called a standard event (SEV). To check its validity, it can be fit with the expected parametric pulse shape of Eq. 1.9.

During raw analysis the standard event is usually calculated by selecting a number of well defined particle pulses by some manually defined cuts on the data and subsequent calculation of the arithmetic mean of the selected traces. Finally, the standard event is normalized to height 1.

For the estimation of the noise power spectrum, a similar approach is chosen. It is crucial that the traces used for the calculation do not contain particle pulses, as this would degrade the performance of the optimum filter. Because of this, again clean traces are selected by cuts prior to the calculation.

2.2 Triggering

In the context of the NUCLEUS experiment, triggering denotes the localization of particle events on the raw data stream. This is done by considering a predefined number of samples at a time, called a record window. Due to the nature of the SQUIDs used for data acquisition, the offset from 0 of the sample values can change for each record window. Because of this, the offset has to be determined by calculating the mean of the first n samples for each record window. Usually n is chosen to be 1/8 times the length of the record window. The offset gets subtracted from the window, bringing its baseline to 0. Afterwards, if an optimum filter has been calculated, it can be applied to the window, if not, the unfiltered trace is considered. For a maximum in the record window to be considered a trigger candidate, it has to fulfil three conditions. Firstly, it has to surpass the trigger threshold, secondly it cannot be in the excluded region near the edges of the record window, and thirdly it cannot be too close to another sample already identified as trigger. If these conditions are fulfilled by a point, a new record window is defined with this point located at 1/4 of its length. If there is no higher maximum in the record window, the point of the maximum is saved as trigger, otherwise the window is resampled up to a predefined number of times. The target of this procedure is for all triggered events to be well aligned in time.

As described before, heater pulses, so called test pulses, are injected at regular intervals into the detector. To determine whether a triggered pulse belongs to this category, the DAC heater channel is considered. If a signal at the corresponding time is present, the pulse is considered as a test pulse.

Furthermore, during triggering usually also some noise baselines are triggered. These are needed among other things for the calculation of the noise power spectrum required for the optimum filter, event simulation, efficiency and resolution studies. To trigger a predefined number of baselines, a corresponding number of record windows located randomly in the stream is picked. It is checked that none of the resulting baselines overlap with test pulses.

Triggering usually happens as a two-step process. As a first step, events and baselines are triggered without the use of an optimum filter. The baselines are cleaned and a noise power spectrum (NPS) is calculated from them. Then the target events are considered, and the standard event is calculated as described above. With the standard event and NPS an optimum filter is calculated. This optimum filter can subsequently be used to redo the triggering of events with lower trigger thresholds to increase the sensitivity to low energetic pulses.

For the determination of the trigger threshold, multiple methods are available. One method is to estimate the noise trigger rate, see e.g. ref. [35], and to set the threshold in order to match a predefined expected noise trigger rate. There is however another approach, which also yields satisfactory results. It defines the trigger threshold as a multiple of the baseline resolution. To determine the baseline resolution, a number of cleaned baselines is taken. If the resolution with the optimum filter is to be calculated, the filter is applied. Then, one sample value in the centre of each baseline is taken. The sampled values are expected to follow a normal distribution centred around zero. To verify this, their histogram is fited with a normal distribution. The standard deviation of the fitted distribution is a measure of the magnitude of noise fluctuations and called baseline resolution.

2.3 Pulse height estimation

We have already seen that the energy deposited in the detector scales, up to the saturation limit, linearly with the pulse height. This means that to correctly assign an energy to a particle pulse, one needs an estimate of the true pulse height and a calibration factor. The calibration factor can be achieved with several methods, see e.g. ref. [36] and [37] for possible calibration

methods for the NUCLEUS experiment. For the estimation of pulse height also several methods are available. Taking the maximum of a pulse is the simplest approach. It has some shortcomings however, especially in the case of very small and very large pulses. In the case of small pulses the magnitude of noise fluctuations can be in the order of the pulse height, and thus add a significant bias, and in the case of large pulses the detector might go into saturation, meaning that the relation between pulse height and energy ceases to be linear.

Standard event fit A better and conceptually straightforward option is the so called standard event fit. It consists of fitting the event trace with the standard event, with the scale and timing of the SEV as free parameters. Since the standard event is normalized to height 1, the fitted scale can be used as an estimate for the true pulse height of a pulse. This works well for well defined pulses with magnitudes sufficiently above the noise level, but it can produce misleading results for very small pulses.

This method can also be adopted to enable pulse height estimation for events with pulse height above the saturation limit. This is due to the fact, that below the saturation limit the response of the detector is still expected to be linear even for large pulses. This means that the standard event fit can still be employed, considering only parts of the pulse that lie below the truncation limit. This is called truncated SEV fit [38] and makes it possible to extend the energy range of the detector.

Pulse height estimation with the optimum filter Even though the optimum filter does not preserve the pulse shape, it can be used as an unbiased estimator for the pulse height of non-saturated pulses. It it the state-of-the-art method for pulse height estimation of small pulses.

2.4 Data simulation

Simulated pulses are used for efficiency studies and for training neural networks. In the context of the cryogenic data used for this work, a simulated event consists of a noise baseline superimposed with a scaled and time shifted standard event.

The noise baselines used can either be noisy traces taken from the raw data stream, or be simulated using the calculated noise power spectrum. The use of recorded baselines yields simulated events that are closer to observed ones, however, baselines used need to be cleaned well to avoid using baselines that contain small pulses or artifacts. This is not a big problem in underground measurements with good background suppression but may be challenging with measurements that contain higher levels of background. To simulate traces that have a given noise power spectrum, a technique described in [39] is used. Baselines simulated in this way are free of signal events and artifacts, however they are only an approximation to what real baselines look like.

2.5 Quality cuts

On the data stream, in addition to particle events following the expected pulse shape, usually also other types of events are present, and get triggered alongside the particle events. To prevent these artifacts from distorting the measurement results, it is essential to remove them from the datasets. In order to do this, so called quality cuts are defined on the data. The goal of these cuts is to exclude as many artifacts as possible while preserving the integrity of the desired events. There are different kinds of cuts employed. Rate based cuts focus on the number of events per time: if there are regions with significantly anomalous event rate, they are excluded. Stability

cuts focus on the detector test pulses. If for a given injected heater amplitude the recorded pulse height deviates significantly from the mean, the detector might have changed the working point, therefore rendering recorded pulses during that time are incomparable to others. Furthermore, there are the anti-coincidence cuts with events from the active veto systems, which serve to exclude non-neutrino interactions.

While it is relatively straightforward to apply and automatize the aforementioned cuts, the last category of quality cuts is more involved and requires manual tuning. These cuts are shape-based cuts, and target artifacts that do not follow the expected pulse shape. In order to define these cuts, some values that reflect the pulse shape have to be calculated for each event. Values that are typically used are the raw pulse height, the offset between left and right baseline levels and the onset, rise and decay times of the pulse, among others. In CAIT, these parameters are called "main parameters", and their calculation is conveniently implemented. Note that in CAIT, in order to reduce the effect of noise fluctuations, a rolling average is employed prior to the calculation of these main parameters. Furthermore, the root mean square error of the standard event fit is a viable metric to define cuts on.

The cutoff values for each of these parameters have to be defined by hand for each measurement. For example, one might define a cut on the raw pulse height, excluding non physical events with heights larger than the saturation limit, a cut on the baseline difference, excluding steps and flux quantum losses, and a cut on the fit root mean square (RMS) error to exclude pile-ups (see next subsection for a short overview of common artifact types). While all of these cuts are qualitatively justified, determining the exact cutoff values requires examining the data and tuning the values manually. This process is not only time-consuming but also introduces the risk of biases.

2.5.1 Artifacts

All traces that get triggered but do not contain well defined event pulses are called artifacts. They can be caused by various reasons, and it is not always clear why and how they arise. There are however some common classes of artifacts that are present in many datasets, due for example to the characteristics of the detector readout electronics. The following is a short, non exhaustive summary of some recurring classes of artifacts.

Flux quantum losses

Flux quantum losses arise due to the properties of the SQUID used for the TES readout. Its response is periodic, rather than linear. This means that there is no fixed baseline level for the detector output, but it is rather linearized by electronics to some baseline level. This results in an ambiguity in the output. We call flux quantum loss an event where the detector output returns to a different baseline after an event than it had before. It happens mostly after large, saturated event pulses.

SQUID resets

Large steps in the signal can arise in a similar fashion as flux quantum losses due to the ambiguity of the SQUID signal output. If the output leaves a predefined range, it is reset by electronics, thus leading to a large step in the signal.

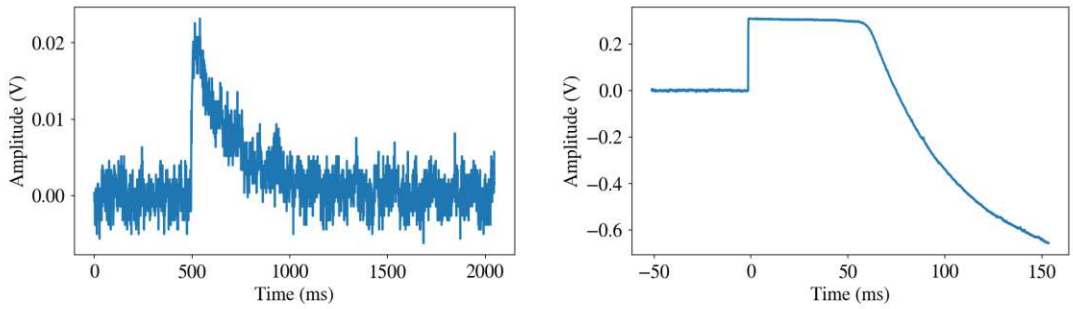


Figure 2.2: Examples of a well defined pulse on the left and a flux quantum loss on the right.

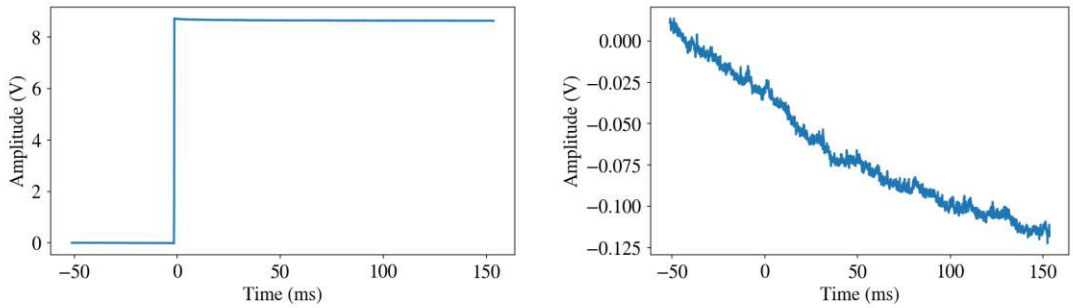


Figure 2.3: A SQUID reset, resulting in a large step in the output signal on the left, a triggered decaying baseline on the right.

Decaying baselines

If an event has a very long decay time, its tail can extend into a new record window. Due to the nature of the optimum filter, such a decaying baseline can exceed the trigger threshold after filtering.

Pile-up events

If there is more than one event inside a record window, it is called a pile-up event. These events can be difficult to deal with when defining cuts.

Spikes

These are thin spikes in the signal, presumably originating from detector electronics.

Early or late trigger

Sometimes events are triggered at the wrong time, resulting in events that are still in the record window, but not aligned on the time scale.

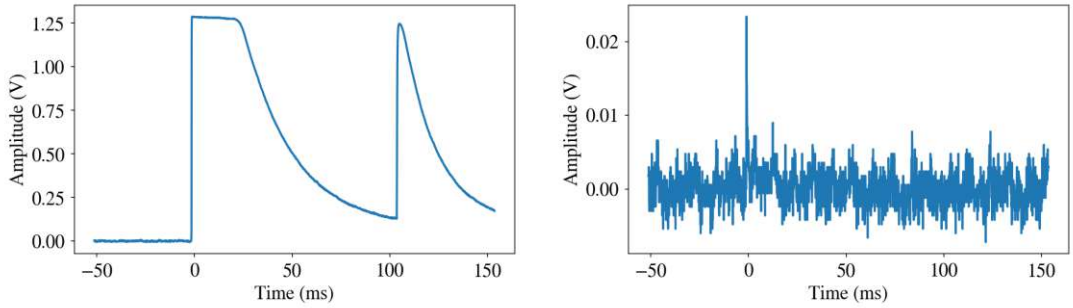


Figure 2.4: (left) A pile-up event, consisting of a heavily saturated and a less saturated pulse. (right) A spike event is visible.

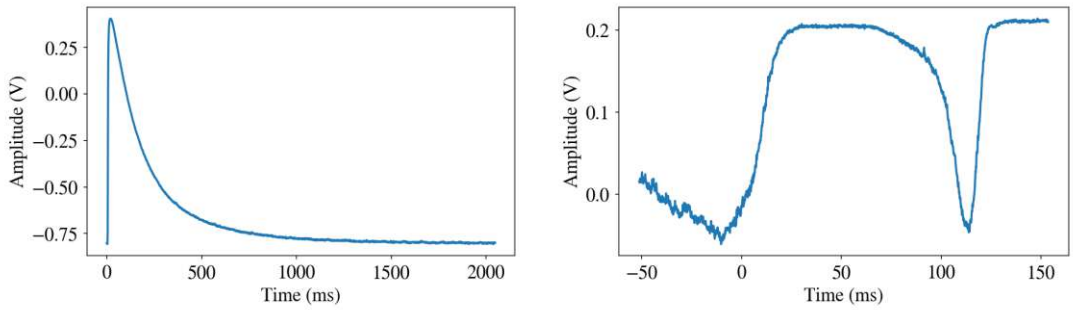


Figure 2.5: (left) The event was triggered too late, due to it being at the beginning of the data acquisition. (right) The artifact shown does not fit into any of the categories described above.

Chapter 3

Machine learning methods

According to reference [40], machine learning is "a field of computer science that studies algorithms and techniques for automating solutions to complex problems". It has gained prominence in recent years, mainly due to the rise of very large, pre-trained neural networks for natural language processing or image creation like ChatGPT. There are, however, many different methods and approaches to machine learning.

The use of some of these methods has already been explored for raw data analysis as well as detector operation in the CRESST experiment. The use of neural networks for artifact rejection has been explored in references [41], [42] and [17]. The use for detector operation with reinforcement learning is also described in reference [17]. In reference [16] a wider array of machine learning methods for pulse height estimation and data simulation are presented. Due to the similarities in detector technology of the CRESST and NUCLEUS experiments, these works have been an important foundation for the methods and results presented here.

In this chapter, new approaches and methods of incorporating machine learning methods into the raw data analysis of the NUCLEUS experiment are investigated. The use of a class of artificial neural networks called autoencoders is explored for noise reduction of raw event traces. A neural network based approach for triggering the raw data stream is presented. A new framework for quick training of shallow neural networks on new datasets for artifact rejection was also implemented. Moreover, a clustering method that can be used for data visualization and for the creation of labelled datasets is introduced. Finally, we also introduce a new method to extract the event pulse shape from raw, triggered detector data, which enables the automatic creation of a standard event and optimum filter, and thus the automatization of the triggering process.

Many of the methods presented can be combined, reinforcing the automatization aspect of machine learning. For example, by combining automatized triggering, clustering and the artifact rejection network, it is possible to obtain from a raw data stream a dataset containing only clean events with almost no manual interventions necessary. Another example is the use of clustering to dramatically speed up the creation of a dataset that can subsequently be used to train a neural network which is able to combine triggering and artifact rejection into a single step.

A general finding of this work concerning neural networks used with data of the NUCLEUS experiment is that smaller networks tend to match the performance of or even outperform bigger networks with sizes typical for similar analysis tasks. It is thought that this is due to the relative simplicity of the data used with respect to typical machine learning tasks, and should be taken into consideration for future works on the topic.

3.1 Denoising with artificial neural networks

In this section, two neural networks that have been trained for noise reduction in detector traces are presented. In the raw data analysis process, two tasks can be identified for which such a denoising algorithm could be used: pulse height estimation, and pulse characterization (estimation of rise and decay time, used for defining quality cuts). The pulse height estimation is currently usually done via the optimum filter. For pulse characterization, implementations may vary. In CAIT, a rolling average is performed to reduce noise, after which the parameters are calculated on this smoothed pulse.

The autoencoder presented in Subsection 3.1.1 has been optimized to reproduce the signal scale accurately, thus enabling an accurate estimation of pulse height. On the other hand, the Noise2Noise network presented in Subsection 3.1.2 has been tuned to reproduce signal shapes, while only reproducing scale accurately in as small signal region.

3.1.1 Scaling autoencoder

Autoencoders [43] are an important family of artificial neural networks that are used to learn efficient compressed representations of unlabelled data. An autoencoder typically consists of two parts: an encoder network and a decoder network. The encoder network takes the autoencoder input and outputs a latent representation of the input. The decoder network takes the latent representation as input, and outputs the final output of the autoencoder. In the simplest case, the network is trained by using the input as target output, i.e. the network is trained to encode the input into a latent representation, and subsequently reconstruct it. This way, autoencoders can learn to extract meaningful features of the data while ignoring parts that do not carry much information.

The latent representation of a trained autoencoder can be used for dimensionality reduction or as input to classification algorithms. Autoencoders also have applications as anomaly detection algorithms: if they are only trained with 'good' examples of data, they will fail to reconstruct anomalous data well, making the reconstruction error a metric for goodness of the input event. See [44] and [16] for applications of autoencoders as anomaly detectors with cryogenic detector data. Finally, autoencoders can also be employed as denoising algorithms. In this case the model is not trained to reconstruct the original input, but a different one.

In this section a denoising autoencoder is presented, a network that aims to reconstruct noiseless outputs from noisy inputs and is trained with noiseless and noisy versions of the same input. Therefore this network architecture relies on simulated data for training. The model proposed here is called scaling autoencoder because of the features of its architecture. It consists of two separate encoders, one encoding mainly the shape, the other encoding mainly the scale of the input.

The model is trained on a dataset consisting of events that have been simulated using a standard event and altered versions of a base NPS, and evaluated on a dataset simulated using the same standard event and real baselines.

Data used

The datasets used for training and evaluation of this model consisted of simulated data. In order to avoid overfitting and to be able to better asses how the model performs under changing noise conditions, a novel approach to simulate noise baselines has been chosen. For the training dataset, instead of using real baselines for the simulations, simulated baselines have been used. However, instead of using the calculated NPS of the real data directly, is has been altered beforehand by applying a magnitude warp [45] to the NPS. For this, a set of knots $\boldsymbol{u} = u_1, \ldots, u_j$ is defined.

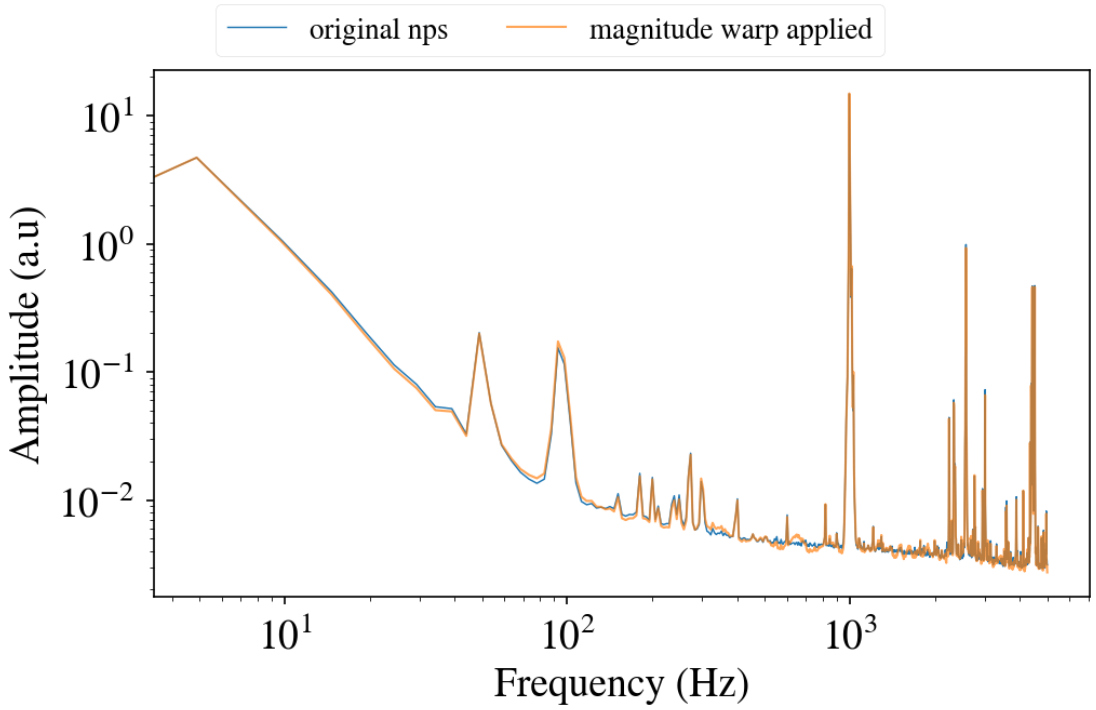


Figure 3.1: Comparison of magnitude warped NPS with unchanged NPS.

The value of each knot is sampled from a normal distribution. The values in between knots are interpolated with a cubic spline $S(\boldsymbol{u}) = s_1, \ldots, s_i$. For a given NPS $\boldsymbol{n} = n_1, \ldots, n_i$, the magnitude warped version is given by

$$\boldsymbol{n}_{scaled} = s_1 n_1, \ldots, s_t n_t, \ldots, s_i n_i \tag{3.1}$$

This can be viewed as crude way to mimic changing noise conditions, as the general shape of the NPS will stay the same, while the exact values are subject to change. See Fig. 3.1 for an example of a magnitude warped NPS.

The training dataset consisted of 500000 events simulated with 50 different, magnitude warped NPS based on the NPS from Run 29 and the standard event from Run 29. Additionally, 50000 simulated empty baselines have been part of the training dataset.

Model and Training

A sketch of the model architecture can be found in Fig. 3.2. The unscaled input is processed two times. Once, it is scaled to height 1 and processed with the shape encoder. It consists of a convolutional neural network, that performs well in shape recognition tasks. The output of the shape encoder is a lower dimensional representation of the scaled input. The unscaled input is also processed by the scale encoder without prior scaling. It consists of a Long Short-Term Memory (LSTM) [46] network followed by a fully connected layer. The output of the scale encoder is again a lower-dimensional representation of the input. The outputs of both encoder networks are then passed to the shape decoder, which employs again a convolutional neural network architecture. The scale encoder's output is processed again by a fully connected layer

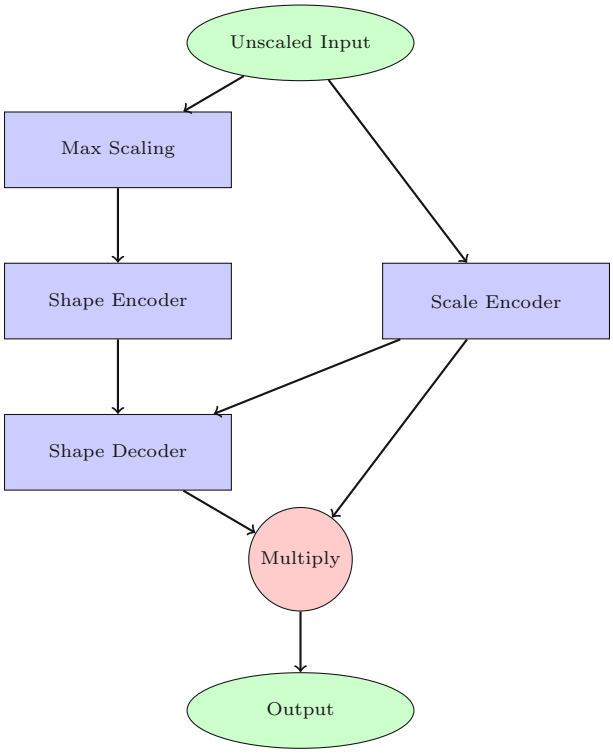


Figure 3.2: Scaling Autoencoder

to produce a scalar. This scalar is multiplied with the shape decoder's output to get the final output of the autoencoder. The loss used was the mean squared error between output and target.

The model has been trained for 150 epochs, taking around 4.5 hours.

Results

To evaluate the performance of the trained network, a test dataset, consisting of 18000 events simulated with real baselines and the same standard event as in training was used. The traces of the test dataset have been denoised with the network, and the results have been compared with pulse height estimations of the optimum filter. See Fig. 3.3 for an illustration of the models performance. In summary, denoising and pulse height estimation work reasonably well for larger pulses, yielding resolutions close to those of the optimum filter. However, for small pulses, the models performance deteriorates. Below a certain threshold, no pulse is detected, and the model struggles to distinguish between an empty baseline and a small pulse. Possible improvements could be achieved by exploring alternative models, adjusting hyperparameters, or improving the quality of the training data.

3.1.2 Noise2Noise

In ref. [47] it has been shown that for image denoising using machine learning algorithms, it is not necessary to have noiseless target images for training. Instead, due to the statistical nature of the problem, it is possible to train an algorithm with pairs of noisy images of the same thing, if the mean of the noise is zero.

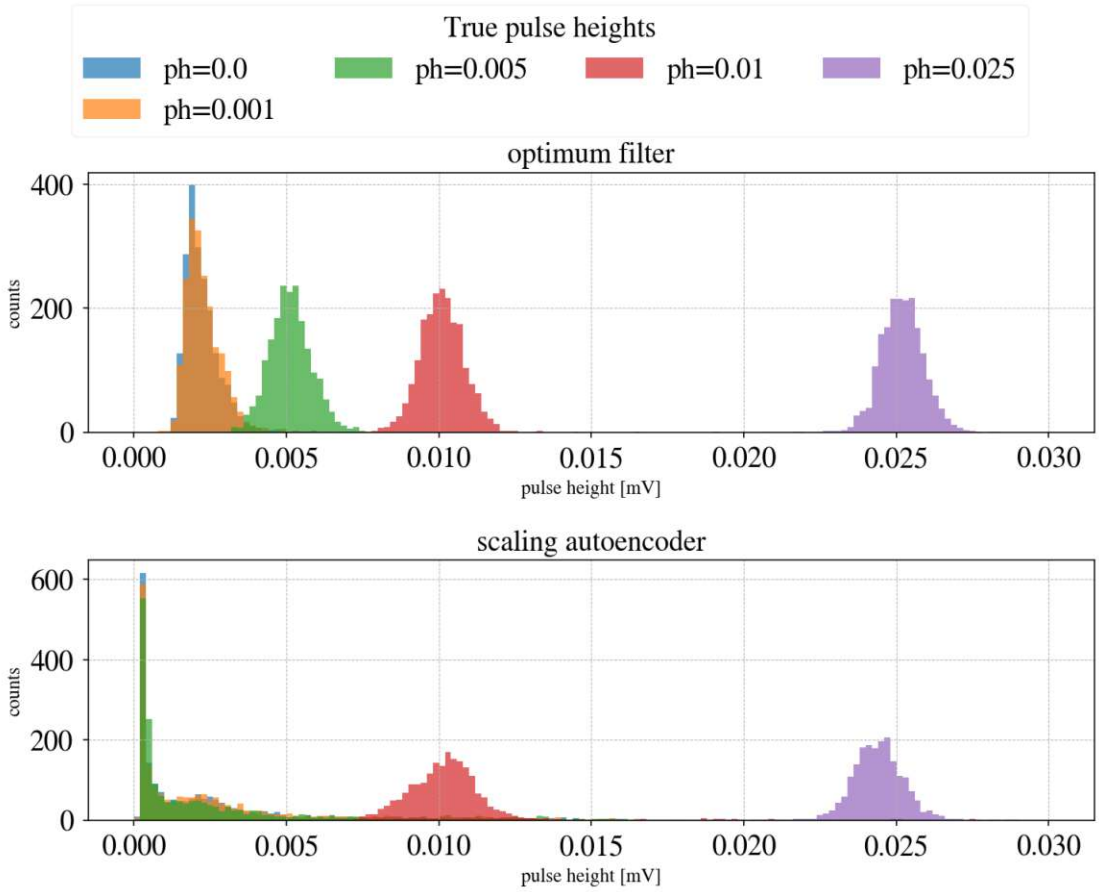


Figure 3.3: Comparison of pulse heights estimated with optimum filter and scaling autoencoder.

To justify this, one can consider the training task of a model that is trained with pairs $(\hat{x}_i, y_i)$ of images, where $\hat{x}_i$ denotes model input, corresponding to the corrupted version of the clean target y_i. Training the model f_θ with parameters θ is equivalent to the minimization task

$$\arg\min_\theta \mathbb{E}[L(f_\theta(\hat{x}), y)] \tag{3.2}$$

where L denotes the loss function. When using the L_2 loss described in Section 2.1.1, this becomes

$$\arg\min_\theta \mathbb{E}[(f_\theta(\hat{x}) - y)^2] \tag{3.3}$$

Now consider that instead of using the clean target y, the noisy target $\hat{y} = y + n$ is used, where n is the noise, sampled from a zero mean distribution:

$$\arg\min_\theta \mathbb{E}[(f_\theta(\hat{x}) - y - n)^2] \tag{3.4}$$

Since the noise has zero mean, this expression is still minimized by $f_\theta(\hat{x}) = y$.

Intuitively, this means that since the process of training a neural network is an optimization problem, even though the gradient for each training step does not point towards the true minimum we are looking for, the average gradient of all training steps will.

Even though this architecture has been devised for the use with images, due to the specific nature of the problem it can be also be used for the denoising of the time series that make up the data discussed in this work. Namely, because the target signal has the same shape, regardless of the energy deposited in the detector (until the saturation limit), two particle pulses that have been scaled to the same height can be treated as two noisy realizations of the same underlying pulse shape.

A simple Noise2Noise neural network has been implemented in the course of this work, with its focus on reconstructing pulse shape.

Data used

Even though in principle the training with event pulses would be possible, for practical reasons training with test pulses has been chosen. The advantages of this are that test pulses are abundant. Also, because they can be grouped into pulses that have been injected by the same heater amplitude, also saturated pulses can be used for training, since we expect their underlying shape to be same, even though it doesn't follow the parametric description of the pulse shape. A disadvantage of using test pulses for training is that due to their different origin, test pulses are not expected to have the same pulse shape as particle pulses. To counteract this, the model is trained to extract the noise, rather than the event pulse, from the input. See below for a description of how the model is defined.

A dataset containing pairs of test pulses can be created largely automated. For this, first test pulses with the same test pulse amplitude are identified. To ensure that only similar pairs of test pulses are used, a threshold is placed on the pulse height difference as well as on the individual left-right baseline differences. Absolute values are avoided to facilitate the use with new datasets. Instead, cutoff values are calculated as percentages of the median values separately for each test pulse amplitude.

Model and Training

Due to the fact that the model is trained with test pulses, which may have a different pulse shape than event pulses, the focus of model training is shifted towards the characteristics of the

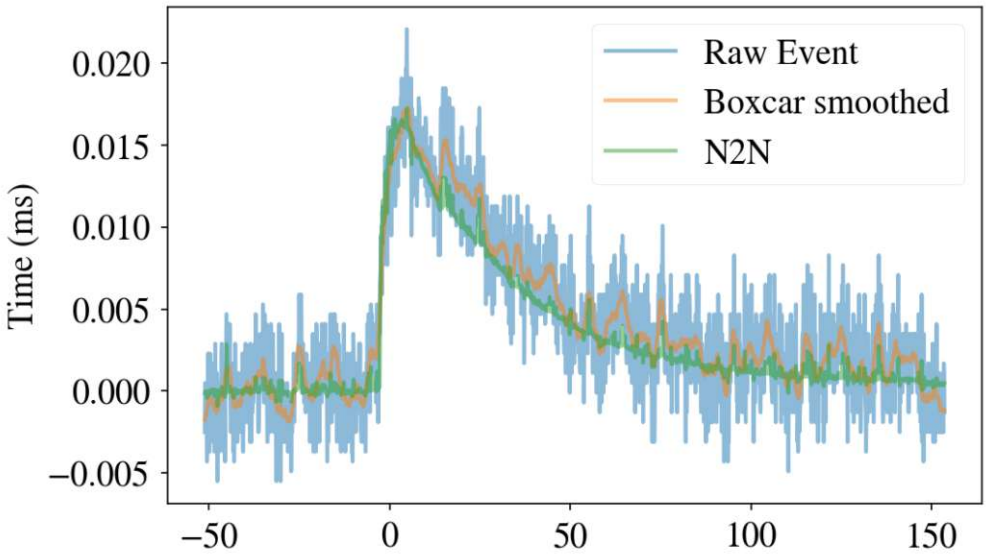


Figure 3.4: Event pulse, denoised by the Noise2Noise model.

noise, rather than the pulse shape. To achieve this, the model is trained to predict the noise residuals of each trace. The denoised version of the input is then calculated by subtracting the model output from the model input. The model itself consists of an LSTM followed by a fully connected layer. The training loss used was the mean squared error loss. Due to the small size of the model, having only about 50000 trainable parameters, training the model on a single GPU for 20 Epochs only takes around 10 minutes.

Results

The model trained on test pulses from a file succeeded to reduce noise considerably in particle events of the same file. For illustration, some figures are shown, each displaying a raw pulse, the version denoised by the model, and the pulse after applying a rolling average. This was included, since it is the method that is currently employed in CAIT to reduce noise before calculation of pulse parameters. In Fig. 3.4, a small event pulse is shown. The version denoised with the Noise2Noise model is significantly smoother than the boxcar smoothed pulse, while still maintaining important features such as rise time and onset.

In Fig. 3.5, two large, saturated pulses are depicted, with one leading to a flux quantum loss. Since the pulse height is on a much larger scale than the noise, the effects of denoising are not apparent. It can be noted however, that the boxcar smoothing washes out the edge of the pulse rise time, while Noise2Noise smoothing does not. In the plot on the right, it can also be seen that the model struggles with pulse shapes deviating from the expected one.

In summary the Noise2Noise autoencoder showed promising performance for denoising. Quantitative comparisons with the currently used boxcar smoothing method for pulse parameter estimation revealed that denoising with autoencoders performed better in some scenarios. The improvements were however not deemed significant enough to warrant the use of this more involved method of denoising.

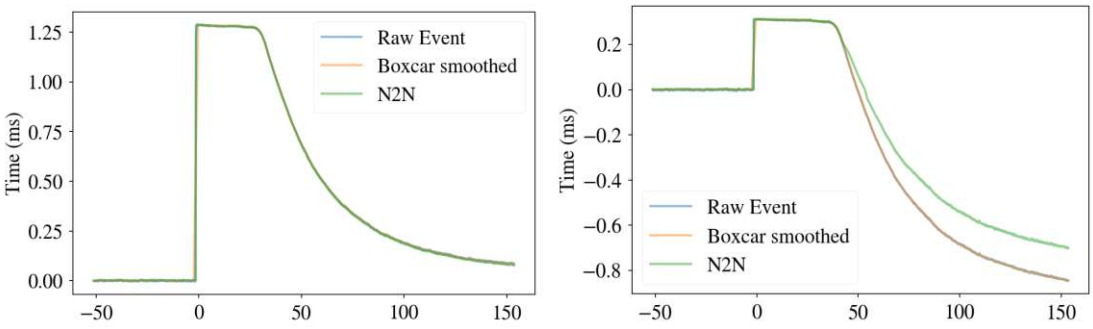


Figure 3.5: A large saturated pulse and flux quantum loss, denoised with different models.

3.2 Shape-based clustering with k-means

Clustering algorithms can be useful tools during raw data analysis. Their ability to group traces into similar clusters can be used for quickly getting an overview of the makeup of a dataset, visualization of cuts, and for the creation of labelled datasets that can be used for machine learning methods.

There are multiple ways to approach clustering traces of raw data. One possibility is to calculate parameters like rise and decay time for each trace and cluster the resulting values. In this case, the clustering itself is a relatively simple operation, that uses some distance metric in the resulting parameter space. However, the quality of the results hinges on the quality and expressivity of the parameters chosen, and the way they are calculated, which can vary for different datasets. Also, these parameters may be on different scales. To give a practical example, consider two of the artifacts described in Section 2.5.1: a step due to a squid reset and a decaying baseline. These can be identified by their left-right baseline difference, which is however on very different scales, and many clustering algorithms struggle to handle clusters that arise on different scales.

A different approach is to use an algorithm that clusters based on shape directly, instead of relying on proxies to describe features of the shape. In the literature many algorithms specialized for the clustering of time series can be found, see ref. [48] for some examples. While these methods work, due to the properties of the data used in this work, another algorithm generally outperformed these clustering methods specifically made for time series. The property we can use is the fact that the time series are expected to be aligned in time. Due to this alignment, the euclidean distance between two values with the same sample number becomes a viable clustering metric. For example, a well triggered event pulse, with pulse height significantly larger than the scale of the noise, has its maximum located at 1/4 of the record window. This means that a value at 1/4 of the record window that is significantly different to 1 after normalization points to the presence of an artifact, similarly values at the start and end of the record window can be expected to be close to 0.

3.2.1 The algorithm

The algorithm proposed to make use of this property is called k-means clustering (see reference [49] for an introduction). Mathematically, for a set of observations $(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n)$ the objective of the algorithm is to partition the observations into $k \leq n$ sets $S_1, ..., S_k$ by minimizing the within-cluster variance:

$$\arg\min_{\boldsymbol{S}} \sum_{i=1}^{k} \sum_{\boldsymbol{x}\in S_i} \|\boldsymbol{x} - \boldsymbol{\mu_i}\|^2 \tag{3.5}$$

where

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\boldsymbol{x}\in S_i} \boldsymbol{x} \tag{3.6}$$

is called the centroid of the cluster. Note that for a cluster containing only good particle pulses, this corresponds to the standard event. For application to the NUCLEUS raw data every event has its offset removed and is normalized to height 1 before clustering.

The k-means algorithm needs a predefined number of clusters. This value can be set by hand, through an educated guess, and subsequently fine-tuned. There is however a method to estimate the quality of clustering for a given number of clusters. By using it, one can automatize finding an appropriate number of clusters. The method is based on the silhouette score [50]. Given some samples that have been grouped into a cluster, the silhouette score is a measure of how well grouped the clusters are. The silhouette score of sample i is given by

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{3.7}$$

where a_i is a measure for the similarity of the sample i with the samples of the same cluster and b_i is a measure for the similarity of sample i with samples of all other clusters. s_i can take values between -1 and 1, with values close to 1 indicating well assigned clusters.

With this, the strategy to find the optimal numbers of clusters k is as follows: for a given range of values of k, k-means clustering is performed. For every iteration, the average silhouette score of samples is calculated. Since this calculation can be time intensive, it is possible to only sample a fraction of the events for this calculation. After all iterations, the number k with the highest corresponding silhouette score is taken to be the best number of clusters.

Performance and uses

The k-means algorithm with silhouette scoring has been implemented in CAIT. In Fig. 3.6 an example of clusters determined with k-means is shown. With this, the content of the dataset can be visualized, and the ratio of events to artifacts can be estimated. For every cluster the corresponding plot shows the centroid or mean event in black, some coloured example events contained in the cluster are also plotted. The number of events in the cluster can be seen on the top right of each plot. Note that in this case, the cluster with the lowest median RMS error of the standard event fit is also highlighted in green. Clusters 2 and 6 seem to contain good, non saturated lower energy pulses, clusters 0 and 7 saturated and higher energy pulses, clusters 1 and 4 contain flux quantum losses, cluster 3 contains steps, and cluster 5 contains decaying baselines.

The method worked well on all tested datasets. It proved useful in dataset visualization, giving an immediate overview of different event types and their quantity, thus making it possible to quickly evaluate data quality. It can furthermore be used to define cuts, and for the creation of labelled datasets for machine learning methods.

3.3 Triggering using neural networks

In this section, a framework is presented that uses an artificial neural network for triggering. It is based on the "overlapping parameter" algorithm described in references [51] and [52].

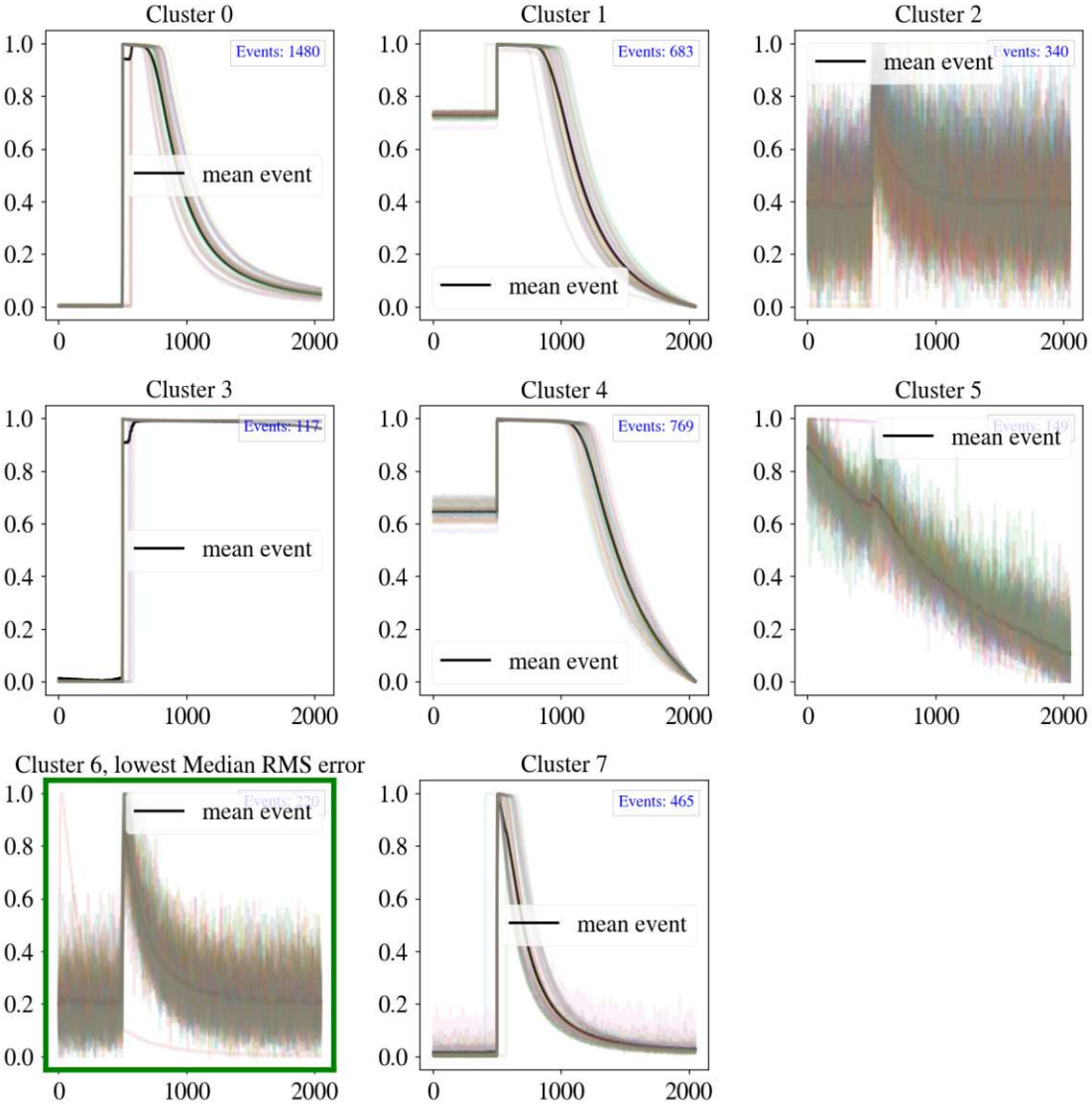


Figure 3.6: k-means clustering for the events of configuration 4 of the LBR. See text for discussion.

The underlying idea of the framework is to treat the problem of event detection in a data stream not as binary classification between the two cases, event present and event not present. Instead, the problem is viewed as a regression problem. For this, the time series, in this case the raw data stream, is divided into overlapping partitions p, also called windows. The predicted quantity for each partition is the overlapping parameter *op*. It denotes the overlap of the partition p_i with an event e and is given by

$$op(p_i, e) = \frac{\text{duration}(p_i \bigcap e)}{\text{duration}(p_i \bigcup e)} \tag{3.8}$$

The value of $op(p_i, e)$ is between 0 and 1. The width of event and windows need not be the same and can be chosen in order to fit the structure of the data. The overlap between windows, which corresponds to the step size when iterating over the data stream can also be freely chosen.

After a time series has been divided into partitions and the overlapping parameter has been predicted for each partition, the task of locating events is reduced to simply finding maxima in the predicted overlapping parameter.

3.3.1 Dataset creation and training

For training, a mix of simulated and real data has been used. The part consisting of real data was assembled by using conventionally triggered data that uses longer record windows than usual. The long record windows were chosen so that every trace containing an event also contains windows with a corresponding overlapping parameter of zero. The data were labelled into three categories. Traces with a single, well aligned pulse were used as positive examples, traces with only artifacts as negative examples and pulses with more than one pulse or misaligned pulses where excluded from the training set. It is important to exclude misaligned pulses because, in the calculation of the overlapping parameter, the pulse location was assumed to be at one quarter of the record window, meaning that different pulse locations would reduce the time resolution of the model. To facilitate data labelling, the k-means clustering algorithm described above has been used as a starting point for labelling. Since this clustering method is not perfect and unable to handle multiple events per trace, cleaning the dataset by hand is necessary. In order for this to be done, a new tool has been implemented in CAIT, the EventCurator. It offers a graphical interface for quick and easy cleaning of pre-clustered traces. Additionally, simulated events with small pulse heights and empty baselines were added. This was done to avoid them being under-represented in the training dataset and to improve resolution. After labelling, the overlapping parameter is calculated for each trace. In case of artifacts or baselines with no pulses present, it was taken to be zero everywhere.

In this work, the width of the events and the window width were chosen to be the same, 1024 samples for all datasets. In this configuration the overlapping parameter is confined between 0 and 1, with a sharp maximum at the exact overlap. The event region was defined to have the pulse maximum at 1/4 of the width. See Fig. 3.7 for an example trace used for training with the training *op* indicated.

The model used was a small convolutional neural network (CNN), consisting of three convolution layers with increasing channel number, followed by a fully connected layer. During training, the mean squared error between predicted and real overlapping parameter was used as loss function.

3.3.2 Locating events

To locate events on the stream, the whole stream is processed with the trained model. The resulting predicted overlapping parameter is confined between 0 and 1 and usually has clear

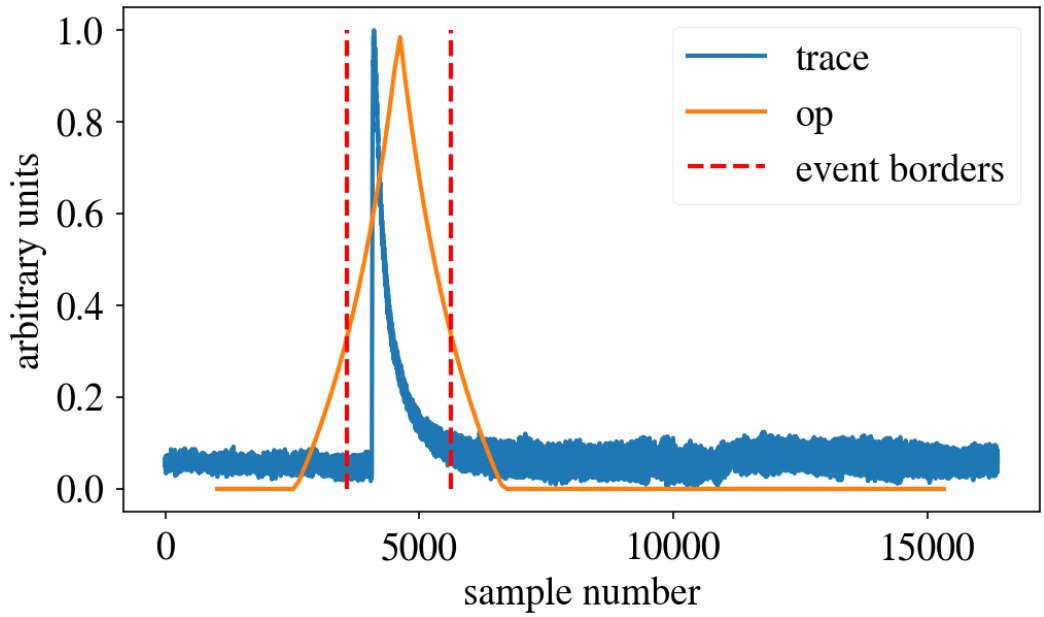


Figure 3.7: Example of a trace and *op* used for training

peaks that can easily be localized with conventional peak detecting algorithms. See Fig. 3.8 for an example of the predicted overlapping parameter of a file of Run 29. Note that this approach to triggering replaces the trigger threshold on pulse height with a trigger threshold on the overlapping parameter. This is a conceptual change, since the overlapping parameter does not represent the height of the pulse, but is a measure of quality a certain pulse has. In other words, small, well defined pulses may produce a high, well defined peak in the *op*, while big, deformed pulses might produce a smaller peak or no peak at all.

In Fig. 3.9, a section of the data stream and the corresponding predicted *op* are visible. The stream section contains one good pulse as well as a flux quantum loss. It can clearly be seen, that the pulse is located correctly (the corresponding region around the peak of the *op* has been marked), while the artifact, even though it has approximately the same raw pulse height as the event, does not lead to a peak in the overlapping parameter.

For further analysis, especially for the creation of a standard event, it is crucial that the events are located at the same point in the record window. This is usually defined to be at 1/4 of the record window, as seen for example in Fig. 3.7. However, the accuracy with which the *op*-trigger algorithm can locate pulses is determined by the step size parameter. This parameter indicates, how many samples the sliding window is moved after each calculation. Hence, the number of traces that need to be processed is

$$n_{windows} = \left\lfloor \frac{l_{stream} - w_{window}}{s_{step}} + 1 \right\rfloor \tag{3.9}$$

with the length of the stream in samples l_{stream}, the window width w_{window} and the step size s_{step}. So while $s_{step} = 1$ will deliver the best possible pulse location, a bigger step size will speed up the calculation considerably. To increase calculation efficiency, the possibility to use a two

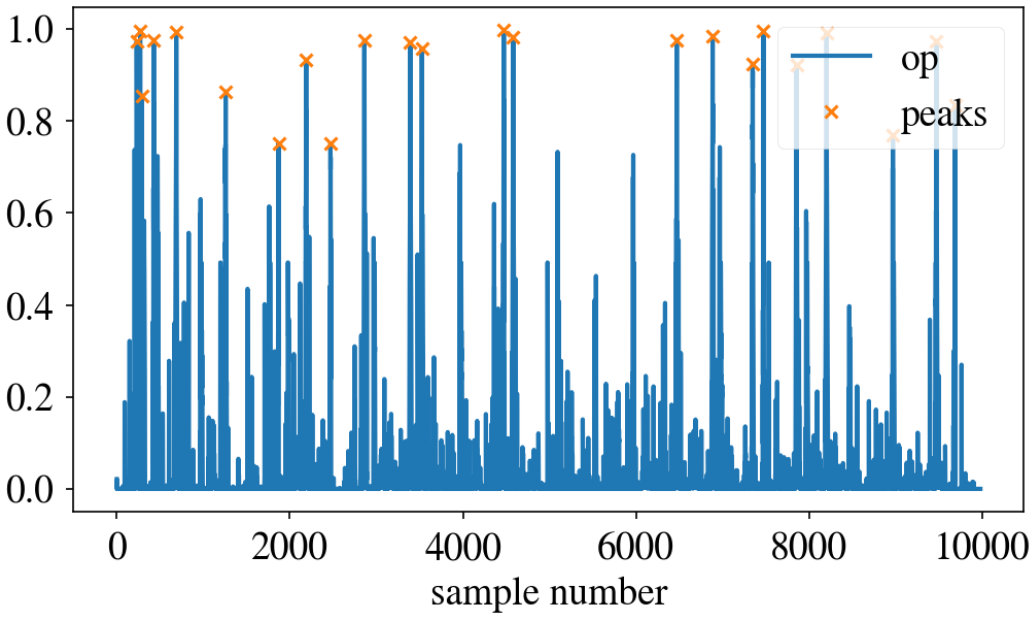


Figure 3.8: Section of the predicted *op* of a raw data stream. The triggered peaks are marked, the *op* threshold was set to 0.75

step algorithm has also been implemented. It works as follows: In the first step, the triggering is done with a big step size to increase calculation efficiency, and rough trigger locations are saved. Afterwards, for each rough trigger location, a window around the trigger is defined, and the exact location is found by iterating over this window with a step size of 1. With this two stage approach, good timing resolution is achieved while reducing processing time significantly.

3.3.3 Results

First, the model was evaluated qualitatively on data from Run 29. Training was done on data without the CRAB source in place, then the data with the source in place was triggered with the trained model. In Fig. 3.10 the resulting low energy part of the spectrum can be seen, together with the published spectrum [36] from the same data. No cuts have been applied to the *op*-triggered data. The main features of the spectra coincide, the small peak around 175 eV was also noted in prior analyses of this data, it is due to a class of artifacts consisting of small steps. To expand on this promising result, more quantitative evaluations of trigger performance have been executed with data of configurations 4 and 5 of the LBR.

Efficiencies

Two more models have been trained with data from a segment of configuration 4 and 5 of the LBR each, used afterwards to trigger the whole configuration, respectively. This emulates what an actual workflow using this tool might look like. Note that for the double-TES detector of configuration 5, two separate models have been trained, one for each TES. In order to study the

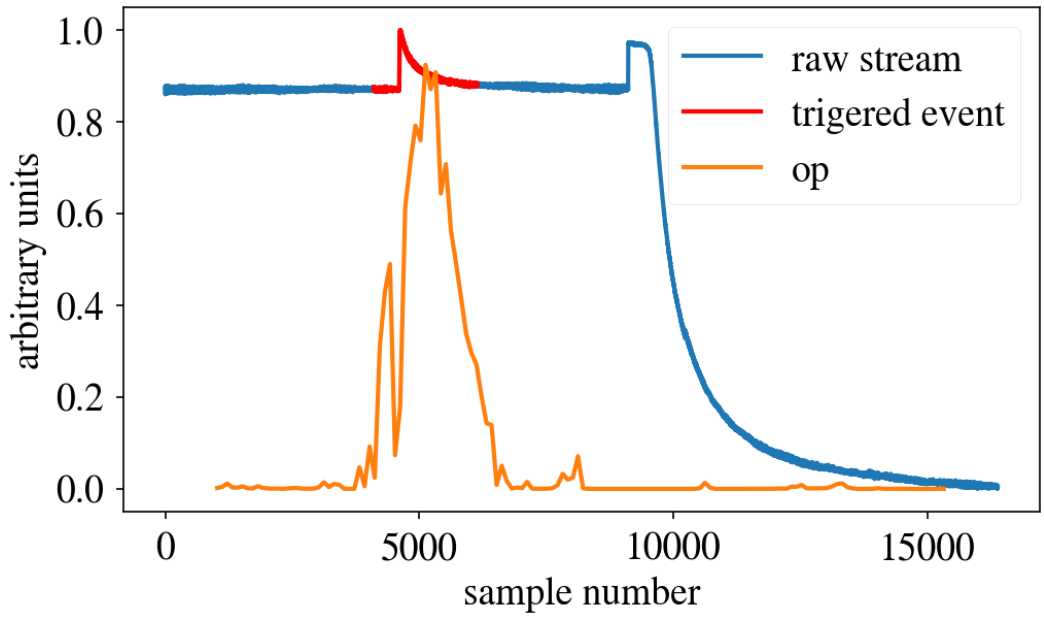


Figure 3.9: Section of the raw data stream containing a triggered event together with the predicted *op*.

trigger efficiency, for both configurations a dataset containing 50000 pulses has been simulated using real baselines. Pulses were simulated with pulse heights uniformly distributed between zero and fifty times the respective baseline resolution, as calculated using the optimum filter as described in Section 2.2. Subsequently the model has been applied to predict the overlapping parameter for every simulated event, as well as for all baselines in the dataset. The latter was done to get an estimate of the noise trigger rate. To avoid biases, no cuts were applied to the baselines beforehand. Since the locations of these baselines are chosen at random from the stream, the noise trigger rate estimated this way is expected to be greater than zero, since particle events or artifacts might be present on some the traces. To see if a trace would have been triggered, the maximum of its predicted *op* was considered.

Fig. 3.11 shows the predicted *op* values for the simulated test set of configuration 4 as well as the receiver operating characteristic (ROC) curve, calculated using the percentage of all simulated events where the predicted *op* passes the threshold as true positives, and all baselines where the predicted *op* passes threshold as false positives. From both plots it is visible that a cutoff value for the *op* of 0.55 is a good choice, balancing type I and type II errors. Figure 3.12 compares the resulting trigger efficiency for *op*-triggering and optimum filter triggering on the same simulated dataset, with the cutoff value for the *op* being 0.55, and the respective cutoff value for the optimum filter corresponding to 10 times the baseline resolution. For this dataset and model, the optimum filter is superior, having a sharper rise at low pulse heights.

Figures 3.13 and 3.14 show the same respective plots for both detectors of configuration 5. There, the results show a better performance, especially in terms of trigger efficiency.

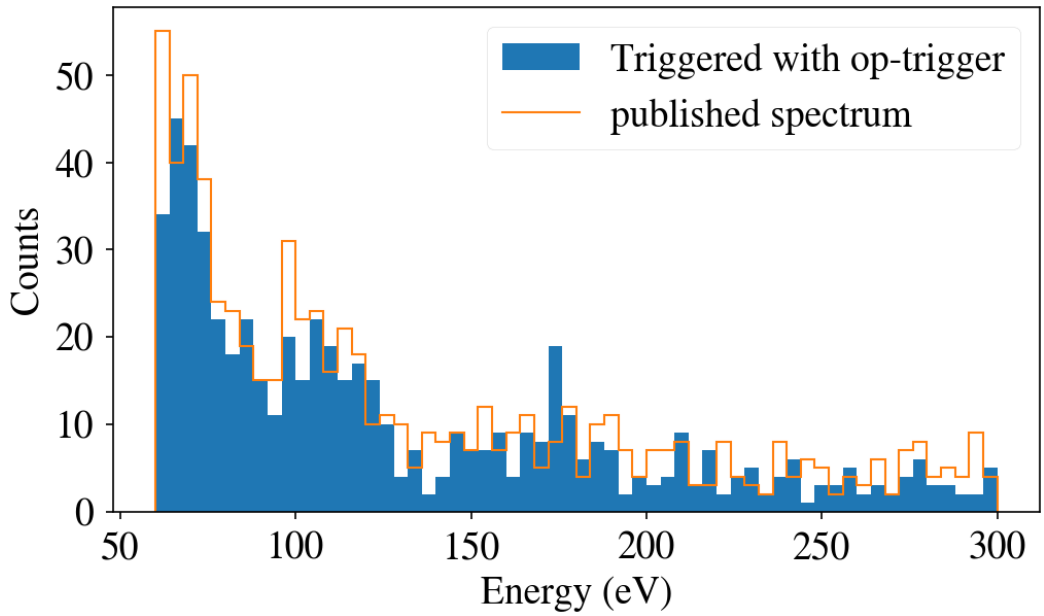


Figure 3.10: Comparison of data triggered with *op*-triggering, without any cuts applied, and published spectrum from the same data.

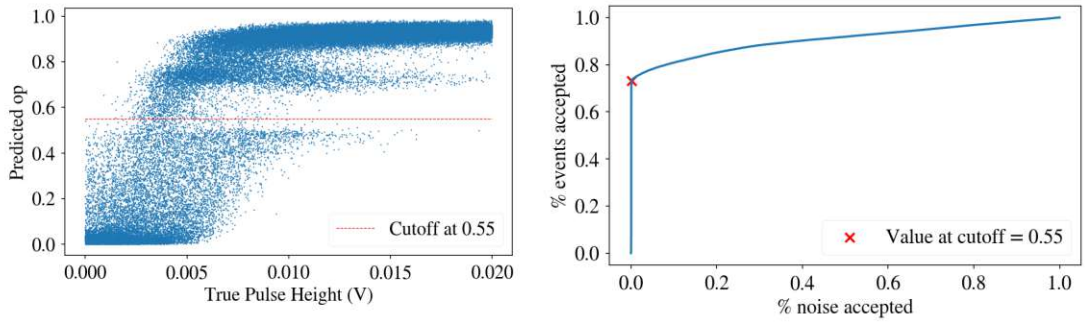


Figure 3.11: Left: Maximum of predicted *op* for the simulated test set of configuration 4. Right: ROC curve for the same dataset.

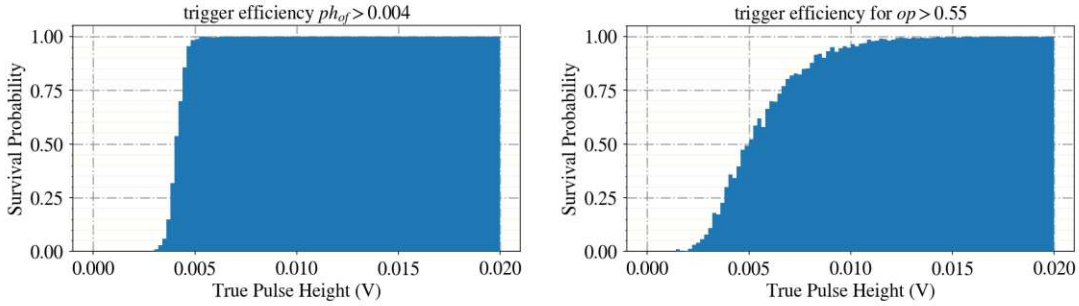


Figure 3.12: Comparison of trigger efficiencies for small pulse heights of configuration 4. Left: the efficiency using the optimum filter with a cutoff value at 10 times the baseline resolution. Right: the efficiency achieved with the *op* on the same dataset.

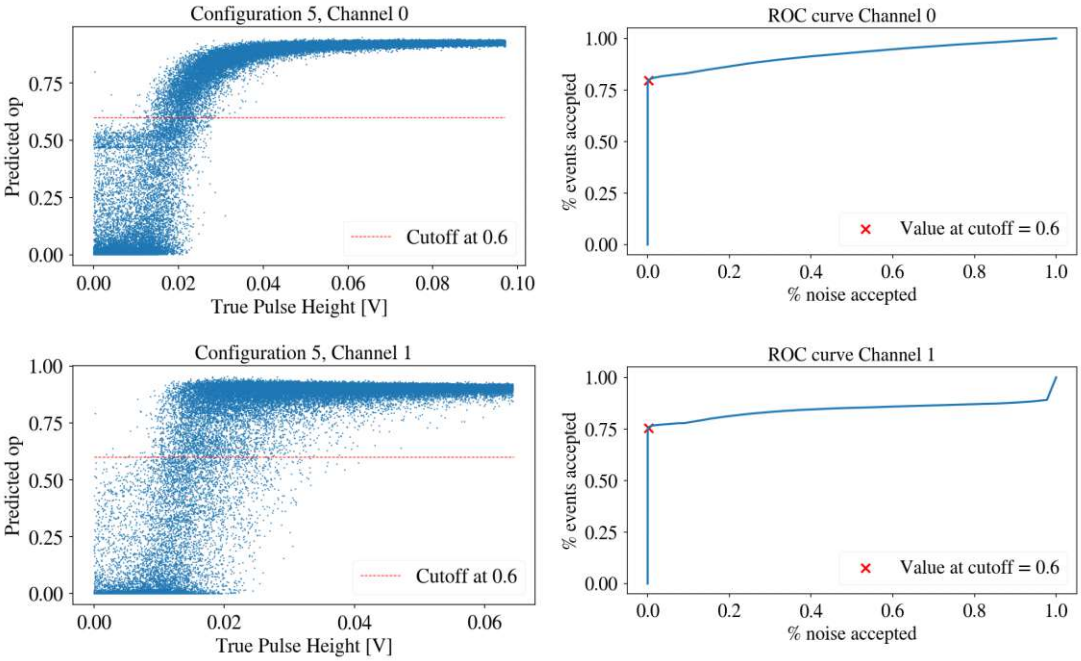


Figure 3.13: Predicted *op* and ROC curves for both detectors of configuration 5.

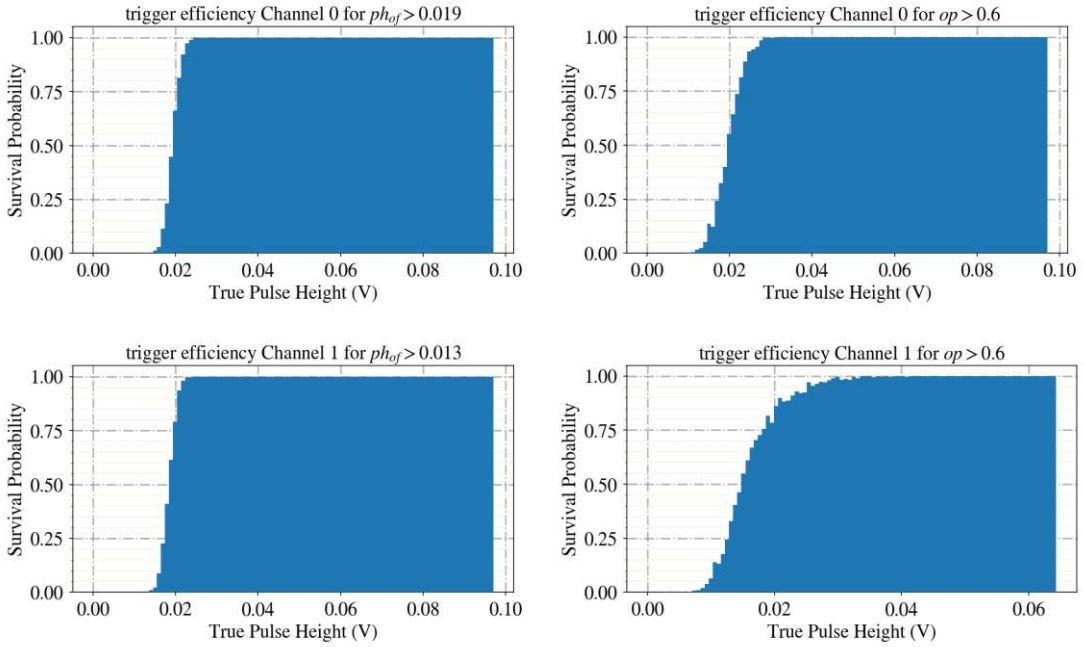


Figure 3.14: Comparison of optimum filter and *op* trigger efficiencies for both detectors of configuration 5. The optimum filter cutoff is chosen to be 10 times the baseline resolution in both cases.

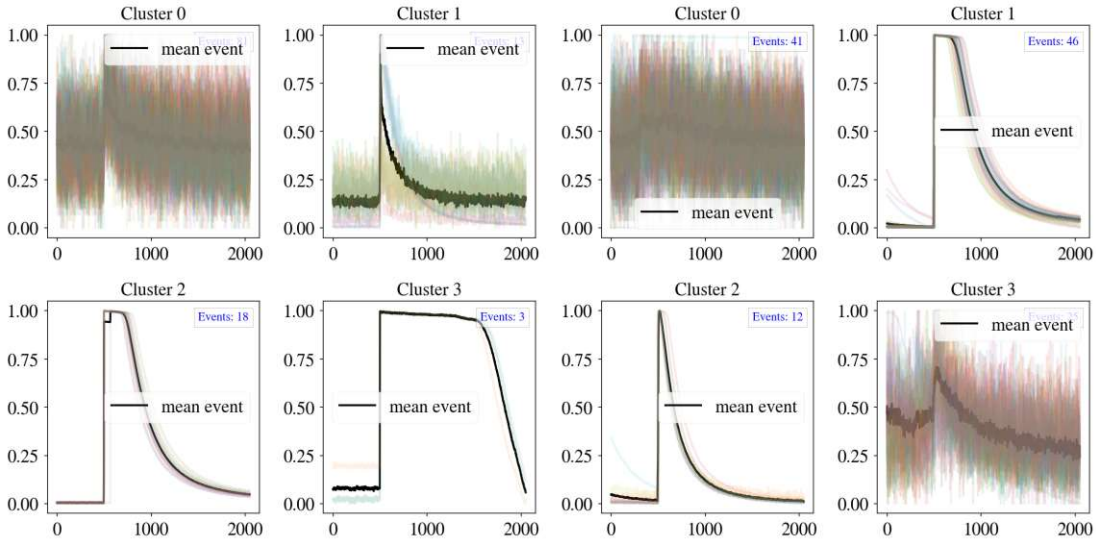
Comparison of triggered events

To better evaluate the performance of the algorithm for triggering, the trained networks used for efficiency studies in the previous section have been used to actually trigger the raw data of configuration 4 and 5, using the threshold values for the overlapping parameter that have been indicated above. The resulting triggered events have been compared to events that have been triggered using the optimum filter method, with a a threshold value of 10 times the baseline resolution. The optimum filter is expected to deliver more triggers, since it can not discriminate between pulses and artifacts. To be able to compare results nonetheless, a cut has been defined on the of-triggered data, aiming to exclude as many artifacts as possible while keeping valid particle pulses. The remaining events have been compared to the ones triggered with the overlapping parameter without cuts applied. To evaluate whether a single event has been triggered by both methods, the trigger time stamps are considered. If two time stamps lie within a predefined timespan, the triggered event is taken to be the same.

Configuration 4 In configuration 4, for two triggers to be considered triggering the same event, they need to happen within 30 ms of each other. In total, the optimum filter triggering yielded 4223 events, the overlapping parameter triggering yielded 2500 events. Of these events, 2389 have been triggered by both methods. After application of a cut on the optimum filter-triggered events, 2491 events remain. Of these, 2376 events are also present in the overlapping parameter-triggered dataset. The numbers are also reported in Tab. 3.1 for a comprehensive overview. It is worth to look a at the shape of some of the events in the respective groups. Since it is assumed that the events that are present in both datasets and survive the cuts are well defined particle events, and that the events that are excluded from the optimum filter dataset by the cut are artifacts, the main interest lies on events that are present either only in the overlapping parameter dataset, or only in the optimum filter dataset after application of cuts. To get an overview of the events, the previously described k-means clustering has been applied using 4 clusters for each dataset. In Fig. 3.15 the results for both datasets can be seen. The events only triggered by the optimum filter consist, as clustered, of 81 good, low energetic pulses, 13 higher energy pulses, 18 saturated pulses and 3 flux quantum losses.

In the corresponding dataset of events only triggered with the overlapping parameter framework, one cluster of saturated events, and one cluster of well defined particle events is visible. The corresponding events are also triggered by the optimum filter, but are wrongly excluded by the quality cut.

Configuration 5 The same procedure as with configuration 4 has been employed also for the raw dataset of configuration 5. The data stream has been triggered with both methods, with the thresholds being again 10 times the baseline resolution for the optimum filter, and 0.6 for the overlapping parameter, as indicated in Fig. 3.13. Both detector channels have been triggered; however for easier comparison, only the results for one channel are presented, since the two channels behave similarly. In this dataset, for two events to be considered the same, their triggers had to be within 5 ms. The number of raw triggers is 10832 for the optimum filter and 2449 for the overlapping parameter, of these, 1579 events are present in both datasets. Again, a cut was defined on the optimum filter triggered data, after which 1640 events remained in the dataset. Of these events, 1375 where also triggered by the overlapping parameter. The numbers can be found again in Tab. 3.1. For illustration of the differences in triggered data, the clusters calculated with k-means clusterings of the events that are only present either in the optimum filter triggered dataset after the application of the cuts, or in the overlapping parameter dataset, are shown in Fig. 3.16. It has to be noted that on this dataset, to evaluate if a trigger is a noise



(a) Optimum filter dataset (b) Overlapping parameter dataset

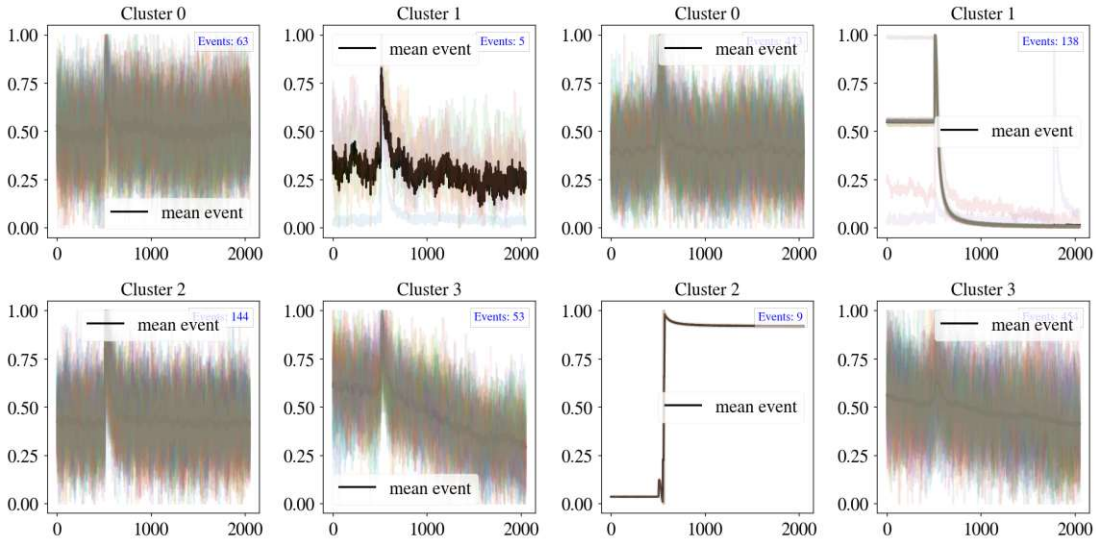
Figure 3.15: Configuration 4: Clusters of events that are only present in either the optimum filter triggered dataset, after application of cuts, or the overlapping parameter dataset.

	Configuration 4		Configuration 5 Channel 0	
	raw triggers	after cut	raw triggers	after cut
optimum filter	4223	2491	10832	1640
overlapping parameter	2500	2500	2449	2449
present in both	2389	2376	1579	1375

Table 3.1: Number of events triggered with different methods on raw datasets of configuration 4 and 5. Note that the cut has been defined on and applied to the optimum filter dataset exclusively.

trigger, both channels need to be considered, since a pulse present in one channel causes both channels to trigger. In the figure, it can be seen that the events only present in the optimum filter dataset are made up mainly of very low energetic pulses, that could also be noise triggers, with a couple of valid, higher energy pulses present as well. In the overlapping parameter dataset, there are also a lot of low energetic pulses and possibly noise triggers, but also some step and flux quantum loss artifacts. These are also triggered by the optimum filter, but removed by the cuts applied.

Time resolution As described above, the two stage approach to triggering with the overlapping parameter framework leads to greatly decreased computing times, while maintaining good time resolution of triggers. Since the training of the neural network is done with events that have been triggered with the optimum filter, the network is trained to reproduce the timing of the optimum filter. Because of this, the time resolution of the trigger algorithm is limited by the time resolution of the optimum filter. Hence, to evaluate the time resolution of the overlapping parameter triggering, the trigger time stamps it delivers are compared to the trigger time



(a) Optimum filter dataset (b) Overlapping parameter dataset

Figure 3.16: Configuration 5, Channel 0: Clusters of events that are only present in either the optimum filter triggered dataset, after application of cuts, or the overlapping parameter dataset.

stamps of triggers achieved using the optimum filter. To do this, the difference in trigger times for all events that have been triggered by both methods is calculated as $\Delta t_{trig} = t^{op}_{trig} - t^{of}_{trig}$. In Fig. 3.17 the resulting values are plotted. It can be seen that the differences in timing are mostly close to 0. In the case of configuration 4, there is a small peak at around 8 ms. It is mainly due to strongly saturated events.

Conclusion In conclusion the overlapping parameter triggering yielded good results. The efficiencies for small pulse heights, as evaluated on simulated datasets, are not yet on par with the optimum filter. This could however still be improved upon with different models and bigger training datasets. The framework can in the future also be combined with the optimum filter by filtering each trace prior to processing it with the neural network.

The trigger performance for larger pulses was very good, combining high trigger efficiency

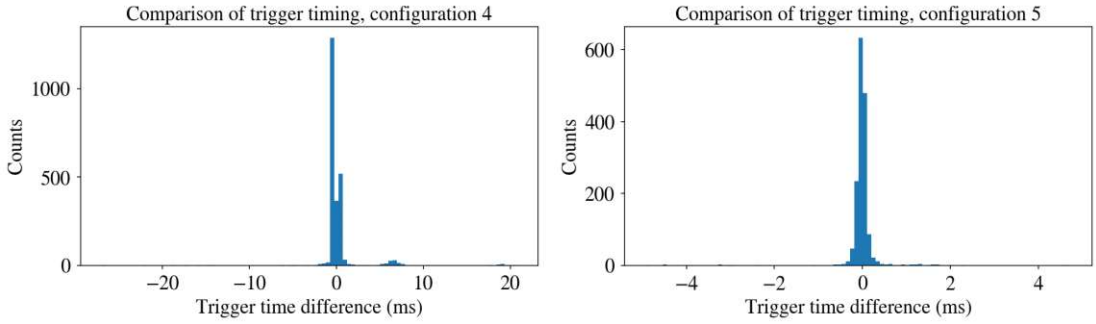


Figure 3.17: Comparison of trigger timing for configuration 4 and 5.

with artifact rejection capabilities. In configuration 4, almost no artifacts where triggered, and while in configuration 5 some where triggered, their amount was far less than the corresponding number that is triggered with the optimum filter.

3.4 Automatic creation of the standard event, optimum filter and trigger thresholds

In this section we introduce a method for the automatic creation of a standard event from raw data. With this method, it is possible to automatically calculate an optimum filter that can be used to estimate trigger thresholds, thus automatizing the complete optimum filter triggering routine.

3.4.1 AutoSev

The calculation of a standard event (SEV) is an important step in the analysis process. It is used to calculate the optimum filter and for pulse height estimation via fitting. The fit error of the standard event fit is an important quantity on which cuts can be defined.

In order to calculate a good, representative standard event, it is crucial that no artifacts are included in its calculation. To ensure this, the data has to be thoroughly inspected by hand, and cuts have to be defined to exclude unwanted traces. This has to be done manually for every detector run, since the shape of events and the noise are dependent on the detector itself, but also on the configuration and background conditions. However, in this work a new approach to automatize the standard event creation is presented. It has few parameters and requires no prior knowledge about the data, while delivering results that are very close to standard events calculated with manually defined cuts.

This approach, called AutoSev, hinges on several assumptions about the data: First of all, there need to be test pulses present. Since they are used for monitoring purposes of the detector, this is usually the case. Furthermore it is assumed that the shape of the test pulses is not completely dissimilar from the shape of real particle induced events, a condition that is also usually fulfilled, even though we do not expect the test pulses to have the same shape as particle pulses. Lastly, it is assumed that particle induced pulses follow the parametric description taken from [13] and given by Eq. 1.9.

The basic idea of the procedure is as follows: first, a standard event of test pulses is calculated as first approximation of the real standard event. Each test pulse has a corresponding heater amplitude or test pulse amplitude. If the heater amplitude is too high, then test pulses will start to saturate. To avoid this, only the lower half of heater amplitudes is considered. Also, the lowest heater amplitude is not considered, since it might be too noisy. On the remaining test pulses, another automatic cut is performed. For this, each test pulse amplitude is considered separately. For all events with the same test pulse amplitude, the pulse height and left-right baseline difference are considered. Their values are transformed into a robust z-score (see App. A). If both quantities do not exceed 1, than the test pulse is accepted for the standard event calculation. Through the use of the robust z-score, absolute cutoff values can be avoided, while still being able to reliably exclude outlier values. Testing with different datasets has shown these two quantities to be sufficient to exclude outliers.

The standard event derived from test pulses is then fitted to all particle events, yielding a fitted pulse height ph_{fit} as well as the root mean squared error of the fit rms_{fit}. With these two quantities, a normalized error

$$rms_{norm} = \frac{rms_{fit}}{ph_{fit}} \tag{3.10}$$

	Configuration 4		Conf. 5 Ch. 0		Conf. 5 Ch. 1	
	SEV	AutoSev	SEV	AutoSev	SEV	AutoSev
t_0	-1.612	-1.772	-0.084	-0.089	-0.086	-0.088
A_n	1.129	1.117	1.210	1.187	1.138	1.163
A_t	0.869	0.878	0.295	0.117	0.229	0.512
τ_n	41.561	43.072	0.640	0.919	0.730	0.515
τ_{in}	0.400	0.355	0.035	0.039	0.024	0.026
τ_t	13.166	13.271	2.735	5.700	3.823	2.169

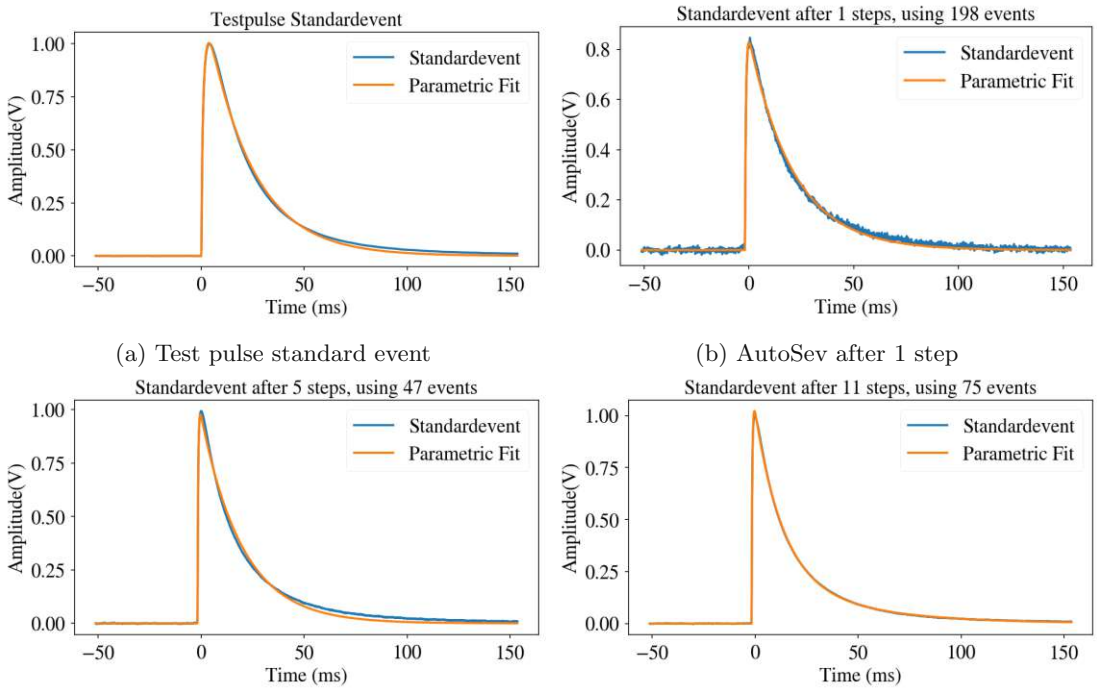
Table 3.2: Comparison of fit parameters between standard events calculated by using manually defined cuts and fit parameters of standard events calculated with AutoSev for different datasets.

is calculated. This quantity is then used to define a cut for the calculation of a new standard event. To avoid having to set the cutoff value for rms_{norm} manually, the following procedure is employed:

First, the n_0 events with the lowest rms_{norm} are taken and used to calculate a standard event. Then, this standard event is fitted with the parametric description of Eq. 1.9. The root mean squared error between the standard event and the fitted function rms_{par} is calculated. Now, the number n of events used for the standard event creation is increased. This will cause rms_{par} to decrease at first, since more good events are averaged, thus reducing the noise. However, at some point rms_{par} will start to increase again, when artifacts or events not following the parametric description are also included in the calculation. Thus, n is increased until rms_{par} stops to decrease. The resulting standard event is again fitted to all events, and the procedure is repeated iteratively until the number of events used for the calculation no longer changes or the maximum number of iterations is reached. After that, the standard event of the iteration that yielded the overall lowest deviation from the fitted parametric description is taken as final standard event.

Results

The method produces very good results on all tested datasets. Figure 3.18 shows the standard events calculated for different iteration steps on data of configuration 4. in Fig. 3.19 the corresponding plots for channel 1 of configuration 5 are shown. In both figures the test pulse standard event, while being almost noise free due to the high number of test pulses used in the calculation, has some deviations from the parametric descriptions, and a different shape than later iterations using event pulses. After the first step, the standard event is still noisy, even though a lot of events have been used in the calculation, indicating that mostly low energy pulses have been used. With an increasing number of iterations, the quality of the standard event and its acceptance with the parametric fit improves. After 11 iterations, the standard event is in very good agreement with the parametric fit. In Tab. 3.2 the fit parameters, as defined in Eq. 1.9, of the parametric fit of standard events calculated with manually defined cuts and standard events calculated with AutoSev are reported. Further investigations are needed to understand the fact that, for configuration 4, $\tau_n > \tau_t$, the values being in good agreement for SEV and AutoSev. In Fig. 3.20 the standard events calculated with AutoSev are compared to standard events calculated by using events that have been selected by manually defined cuts. It can be seen that the shapes of the standard events calculated with different methods are in good agreement for all considered datasets.



(a) Test pulse standard event

(b) AutoSev after 1 step

(c) AutoSev after 5 steps

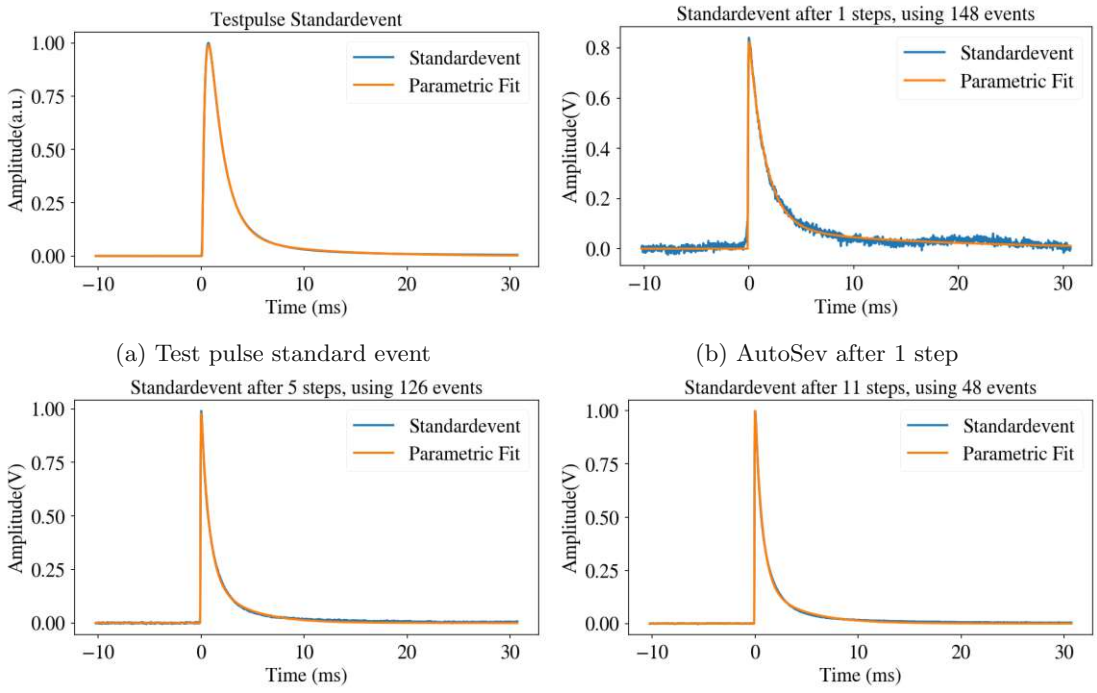
(d) AutoSev after 11 steps

Figure 3.18: Progress of the AutoSev for configuration 4. It can be clearly seen, that the standard event approaches the parametric fit with increasing number of iterations.

3.4.2 AutoOf and triggering

After having a method to automatically create standard events, all that is missing to automatically create an optimum filter is the NPS of clean noise baselines. The baselines need to be cleaned before the calculation of the NPS to avoid particle events or artifacts being present. Automatically selecting clean baselines for the NPS calculation can be done by using two quantities: the root mean square error of a fit with a cubic polynomial and the left-right baseline difference. The former can be used to exclude unwanted particle events and artifacts other than decaying baselines that might be present in the traces. To also be able to reliably exclude decaying baselines the latter is used. For this, the left-right baseline difference is scaled again via robust scaling (see App. A), such that an interpretable cutoff value on the scaled quantity can be defined. In case of the fit error, instead of applying robust scaling, one can exclude a certain percentile, e.g. the 50% of events with the highest fit error are excluded.

With the standard event and the noise power spectrum in place, an optimum filter can be calculated. An important quantity for triggering that is still missing for further automatization is the trigger threshold of the dataset. As mentioned in Section 2.2, it is possible to get an estimate for a reasonable trigger threshold by using a multiple of the baseline resolution, calculated with empty baselines. Each baseline is adjusted to remove its offset, and a specific point at the same position across all baselines is selected. Excluding outliers, the values at these points are expected to follow a Gaussian distribution centred around 0. Outliers are again removed through robust scaling and applying a cutoff based on the resulting z-values. The remaining values are then fitted to a Gaussian distribution, with its standard deviation serving as an approximation of the



(a) Test pulse standard event

(b) AutoSev after 1 step

(c) AutoSev after 5 steps

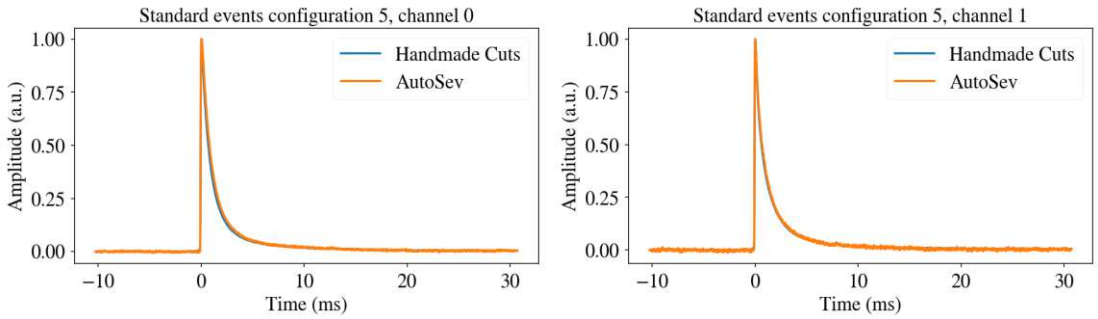
(d) AutoSev after 11 steps

Figure 3.19: Progress of the AutoSev for configuration 5, channel 1. Again, the progress when increasing the number of iterations is visible.

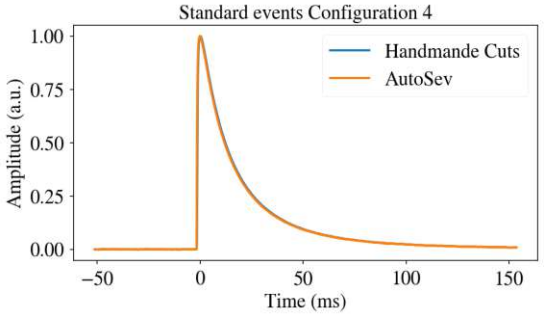
resolution. If the optimum filter is to be included in the calculation, all baselines are filtered with the optimum filter before the fit is calculated.

With these procedures in place, it is possible to completely automatize the optimum filter triggering process: First, the test pulses are triggered. For this, only a threshold on the DAC channel is needed, which is known a priori. Then baselines are triggered. The triggering of baselines consists of the selection of random locations in the raw data stream, hence it does not need any manually set parameters, except the number of baselines to be triggered. A reasonable number can be determined as function of the total length of the stream, e.g. by requiring 1 baseline per recorded second of data. The baselines are used to calculate the noise power spectrum, and to estimate the baseline resolution without the optimum filter. A multiple of the baseline resolution is taken as trigger threshold, in order to trigger the events. From these events, a standard event is calculated using AutoSev, which, together with the noise power spectrum is used to calculate an optimum filter. The baseline resolution is estimated again, this time using the optimum filter. Finally, the events are triggered again, using the optimum filter and a multiple of the corresponding baseline resolution as trigger threshold.

As opposed to the triggering using a neural network presented in Section 3.3, there are no black boxes involved, each step of the automatization can be inspected, and done manually if needed.



(a) Standard events of configuration 5, channel 0 (b) Standard events of configuration 5, channel 1



(c) Standard events for configuration 4.

Figure 3.20: Comparison of standard events calculated with AutoSev and by manually defined cuts

Results

In Fig. 3.21 the automatically calculated noise power spectra, as well as the noise power spectra calculated by using manually cleaned baselines can be seen. Even though there are some offsets between the two, the overall shape remains very similar for all configurations. Since the standard events were also similar, one expects the resulting optimum filter to be comparable as well. Figure 3.22 shows the Gaussian fits for the baseline resolution described above, with and without the use of an optimum filter. The resolutions calculated with the optimum filter for the different filters can be seen in Tab. 3.3. They are comparable, even though the resolution calculated with the automatically made filter has a lower value in all three cases.

To test the procedure, all available datasets have been triggered completely automated. The triggered events were compared to events that were triggered with a manually made optimum filter. For comparability, the trigger thresholds have been matched to 10 times baseline resolution in both cases. As expected, the triggered events were almost the same in all cases, with minimal timing differences. There were also some events that were triggered exclusively by one of the two methods. Closer inspection of those events revealed that they were mostly artifacts, namely decaying baselines. In conclusion, automatized optimum filter triggering is an approach that works well.

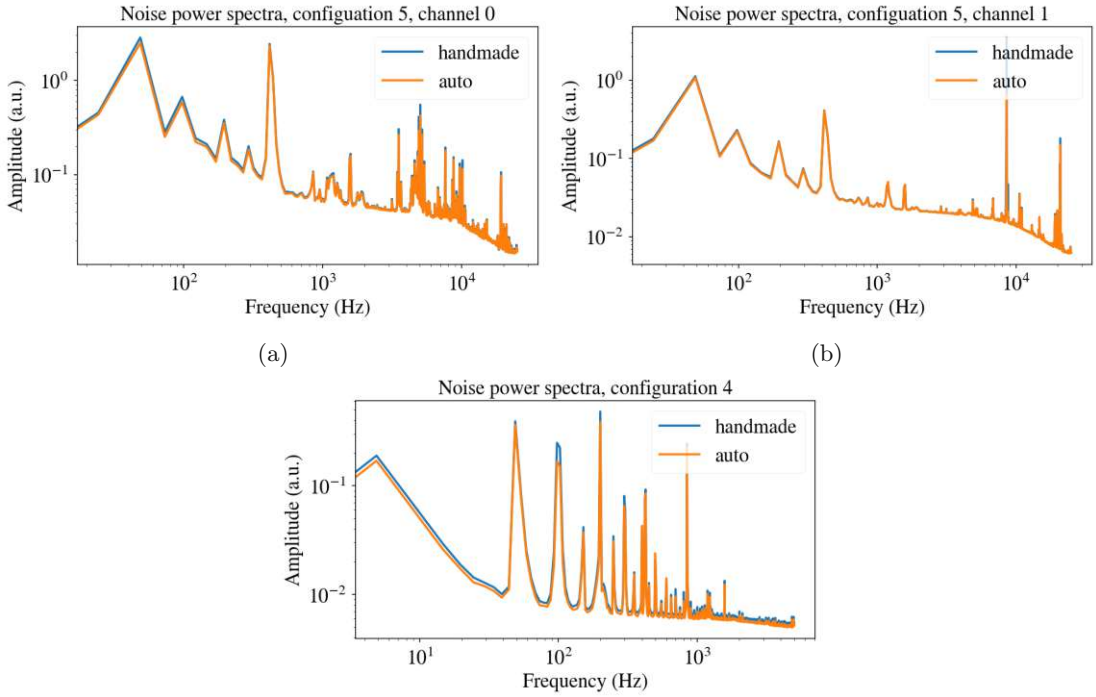


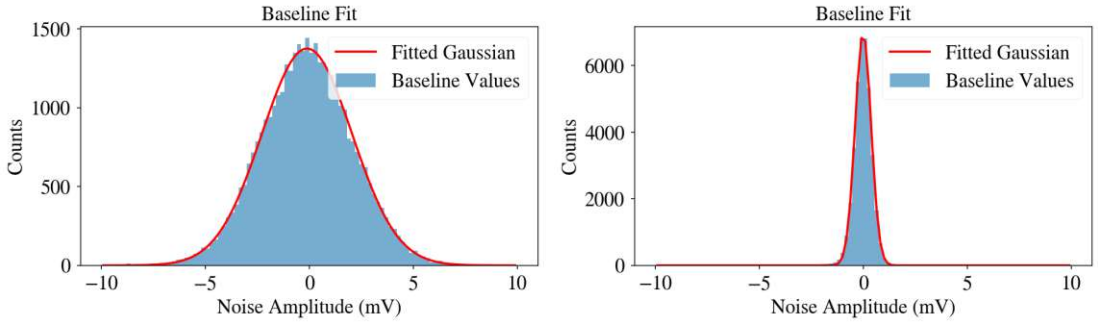
Figure 3.21: Comparison of noise power spectra calculated by using manually cleaned baselines and automatically cleaned baselines, for both channels of configuration 5, and configuration 4.

3.5 AutoCut

Building on the previous Section, the analysis of stream data can be further automatized by focusing on the exclusion of artifacts. In the context of machine learning, this can be viewed as binary classification problem: every trace is labelled either as artifact or good event pulse. Steps in this direction using machine learning have been described e.g. in reference [17]. There, the approach was to build a big dataset, consisting of labelled data of many detectors in different configurations, which were used to train a network to classify traces into good pulses and artifacts. By using a wide variety of training data, the network was thought to generalize well also to data from previously unseen detectors. The paper reports good success, with high classification accuracy. One problem that has been reported was the quality of labels in the dataset. While most labels were correct, some wrongly labelled events were still present.

	manually made filter	automatically made filter
Conf. 4	0.399 ± 0.002 mV	0.382 ± 0.002 mV
Conf. 5, Ch. 0	1.94 ± 0.00 mV	1.88 ± 0.00 mV
Conf. 5, Ch. 1	1.29 ± 0.00 mV	1.27 ± 0.00 mV

Table 3.3: Baseline resolution values for different datasets, calculated with a handmade optimum filter and automatically made optimum filter.



(a) without optimum filter

(b) with optimum filter

Figure 3.22: Fit for baseline resolution calculation of configuration 4 data. Notice the better resolution when using an optimum filter prior to fitting.

The framework presented here, called AutoCut, has a different approach to the aforementioned one. Instead of using a big diverse training set and large models, datasets and models are small. Instead of trying to train models that generalize well to different detectors and detector configurations, models are trained per detector or configuration. This keeps training times very short, while maintaining high classification performance.

The performance of machine learning classifiers is heavily influenced by the quality of the dataset used for training. A good training dataset should be balanced, hence containing a roughly equal number of both classes. Furthermore, interclass variability should be represented well. For positive samples this means including different pulse heights, and potentially saturated pulses. For negative samples, i.e. artifacts, this is more difficult, because even though they can usually be classified into different categories, new kinds of artifacts may always arise, and even known categories might or might not be present in datasets, depending on configurations and ambient factors.

Moreover, to make the per-detector approach feasible, dataset creation should not depend on hand labelling events, since that is a time intensive task. To overcome this problem, a weakly supervised, potentially even unsupervised framework is employed for dataset creation. It hinges on k-means clustering, described in Section 3.2, for the creation of a labelled, balanced training dataset. This dataset can subsequently be used to train a small neural network, that outputs a single number between 0 and 1 per trace. This number, which will be called the AutoCut-value or AC for short, points to the event likely being an artifact (values close to zero) or being a good event pulse (values close to 1).

Since the dataset creation is largely unsupervised, and the goal is not to get comparable accuracy metrics for different network architectures, but to classify traces correctly, the machine learning mantra of strict separation between training data and test data can be relaxed. The underlying reason is that we do not aim for a network that generalizes to datasets taken with different detectors, but for a network that generalizes from a part of a given dataset to the whole dataset. This means it is feasible to use a large dataset, taken with a specific detector at a specific configuration, create a smaller training dataset from it, and use a model trained on the training set to make predictions on the whole dataset. The process can also be thought of as self supervised training strategy: in the first step, a rough clustering together with the (potentially automatically calculated) standard event delivers a rough prediction about data quality. The inclusion of the standard event allows to prescribe a confidence level to each prediction. Training a model on traces with high classification confidence allows the network to improve the first

prediction and to make confident predictions even in edge cases.

3.5.1 Training dataset creation

As mentioned, the creation of training datasets is based on the already described k-means clustering. Once the clustering on a dataset is completed, there are two possibilities of continuing with dataset creation. The completely self supervised approach is to use the standard event fit to identify the best cluster. In this case, the cluster with the lowest average normalized fit RMS error is taken as the cluster containing good events, while examples from other clusters are taken to be artifacts. As consistency check, is is verified that this 'good' cluster also contains the event with the lowest normalized fit RMS error. While this method consistently identified the cluster containing the most well shaped particle pulses during testing correctly, in most cases there were also other clusters that contained viable events. Consider for example Fig. 3.23, showing clusters in configuration 4. There cluster 4, marked with green borders, has been correctly identified as 'best' cluster. However, also cluster 0, containing saturated events, and cluster 2, containing low energetic pulses, should be included in the training dataset as clusters containing viable particle pulses. Because of this, the weakly supervised approach consists of inspecting the clusters by hand, and manually choosing 'good' clusters, that will be used as positive training examples.

Furthermore, it can be seen in Fig. 3.23 that even 'good' clusters can contain artifacts: note the clearly wrongly timed event visible in cluster 4. Also, it can not be excluded that 'bad' clusters contain well shaped particle pulses. To stop the imperfections of clustering from degrading the dataset quality, a quality condition is introduced, making use of the normalized fit RMS error rms_{norm} of Eq. 3.10. For positive clusters, the 50% with the highest rms_{norm} are ignored, while for negative clusters the lower 25% of events with the lowest rms_{norm} are ignored. The remaining traces that belong to clusters labelled as particle events are then taken to be positive training examples. Then a number of traces that is chosen so that the dataset is approximately balanced is sampled from remaining events of each cluster labelled as containing artefacts.

This procedure allows for the creation of a training dataset with minimal human intervention, that is balanced not only between positive and negative samples, but also captures the interclass variability. Additionally, the possibility to include automatically cleaned empty baselines and simulated saturated events as positive training examples, and simulated pile-ups as negative training examples have been implemented in CAIT.

3.5.2 Model

The model used differs greatly from the ones used for example in reference [17]. While models described there have up to several millions of trainable parameters with an input size of 2048 samples, the model used for AutoCut has only around 80000 trainable parameters for the same input size. This allows for fast training, even without the use of GPUs.

The model used consists of only two layers: a convolutional layer followed then a fully connected layer. The convolutional layer has an unconventionally large kernel with the size of 75% of the input size. Usually, much smaller kernel sizes are used. The reason that such large kernels perform well in this instance is again rooted in the specifics of the problem, namely in the alignment. Since minimal padding is used, the choice of kernel size means that only very big features at certain positions in the record window can be correctly learned, which is well suited for the problem. For an illustration of learned filters see Fig. 3.24. It shows the filters of the convolutional layer of a trained model. It can be seen that all the filters containing pulse like features contain them in the first half of the filter kernel, illustrating the point about alignment. Following the convolutional layer, a fully connected layer calculates a single output value from

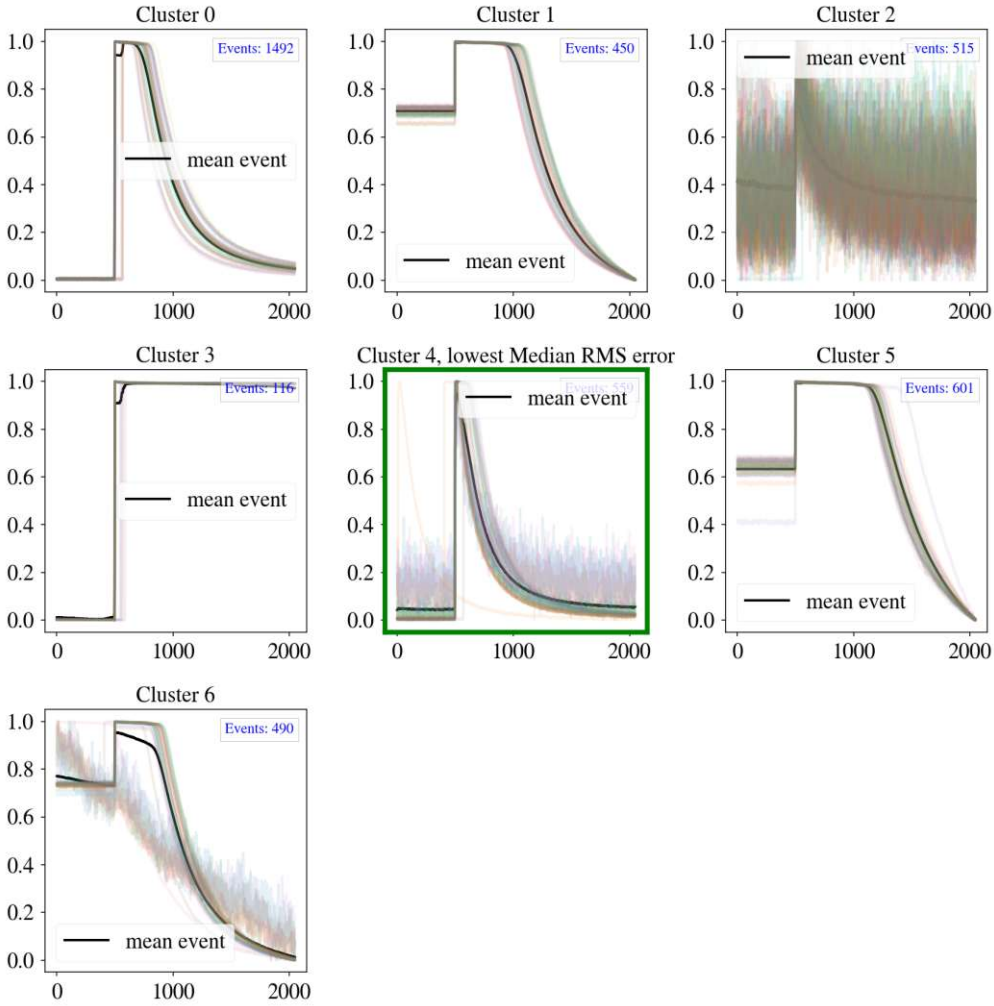


Figure 3.23: Visualization of clusters in data of configuration 4.

the output of the prior layer.

3.5.3 Results

To evaluate the performance of the network trained on data from configurations 4 and 5, two hand-labelled test sets have been made, taking data from configurations 8 and 9. Configuration 4 and 8 use the same $CaWO_4$ detectors and contain each about a week of data with two weeks in between. Configuration 5 and 9 use the same Al_2O_3 double-TES detectors and also each contain data from a week of data taking with two weeks in between. In the test sets, the events were labelled using k-means clustering, and each event was controlled by hand, changing its class if necessary, using the event curator. Note that for the double-TES data each channel has been labelled independently, and independent models have been trained for it. The test dataset of configuration 8 consisted of 4464 events, of which 2493 were classified as good events and 1971 as artifacts. The set of configuration 9 consisted of 9010 events. For channel 0, 1315 were classified as good and 7695 as artifacts. For channel 1, 1415 were classified as good and 7595 as artifacts. Note that there are a lot more triggered events in configuration 9, this is due to the fact that the TES operation was more unstable, and there are a lot of decaying baseline artifacts. For the scope of this work these have been classified as artifacts, even though a more differentiated approach might be sensible, since while some decaying baselines are purely artifacts of the optimum filtering, there are also decaying baselines that contain particle pulses, from which some information might be recoverable.

The AutoCut results were compared to results obtained with a manually defined cut. For this, on each test dataset, handmade cuts were defined on quantities such as left - right baseline difference, decay time and onset, aiming for the cleanest resulting dataset. For both the handmade cut and the AutoCut some relevant metrics were calculated, in case of the AutoCut as function of the AC cutoff value. The following metrics were chosen:

Recall, the proportion of true positive instances correctly identified by the model:

$$\text{Recall} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}}$$

Selectivity, the proportion of true negative instances correctly identified by the model:

$$\text{Selectivity} = \frac{\text{True Negatives (TN)}}{\text{True Negatives (TN)} + \text{False Positives (FP)}}$$

Balanced Accuracy, the average of recall and selectivity, providing a more balanced measure when classes are imbalanced:

$$\text{Balanced Accuracy} = \frac{\text{Recall} + \text{Selectivity}}{2}$$

Precision, the proportion of predicted positive instances that are correct:

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Positives (FP)}}$$

Plots of balanced accuracy and precision over recall are shown in Figs 3.25 and 3.26. The numbers for an AutoCut-value cutoff of 0.5 are listed in Tab. 3.4. Both in the plots as well as in the table it can be seen that the AutoCut matched the performance of the handmade cut in all tested datasets. For illustration, in Fig. 3.27 the traces of configuration 8 that have been wrongly classified by the manually defined cut are shown. The false positives are mainly flux quantum

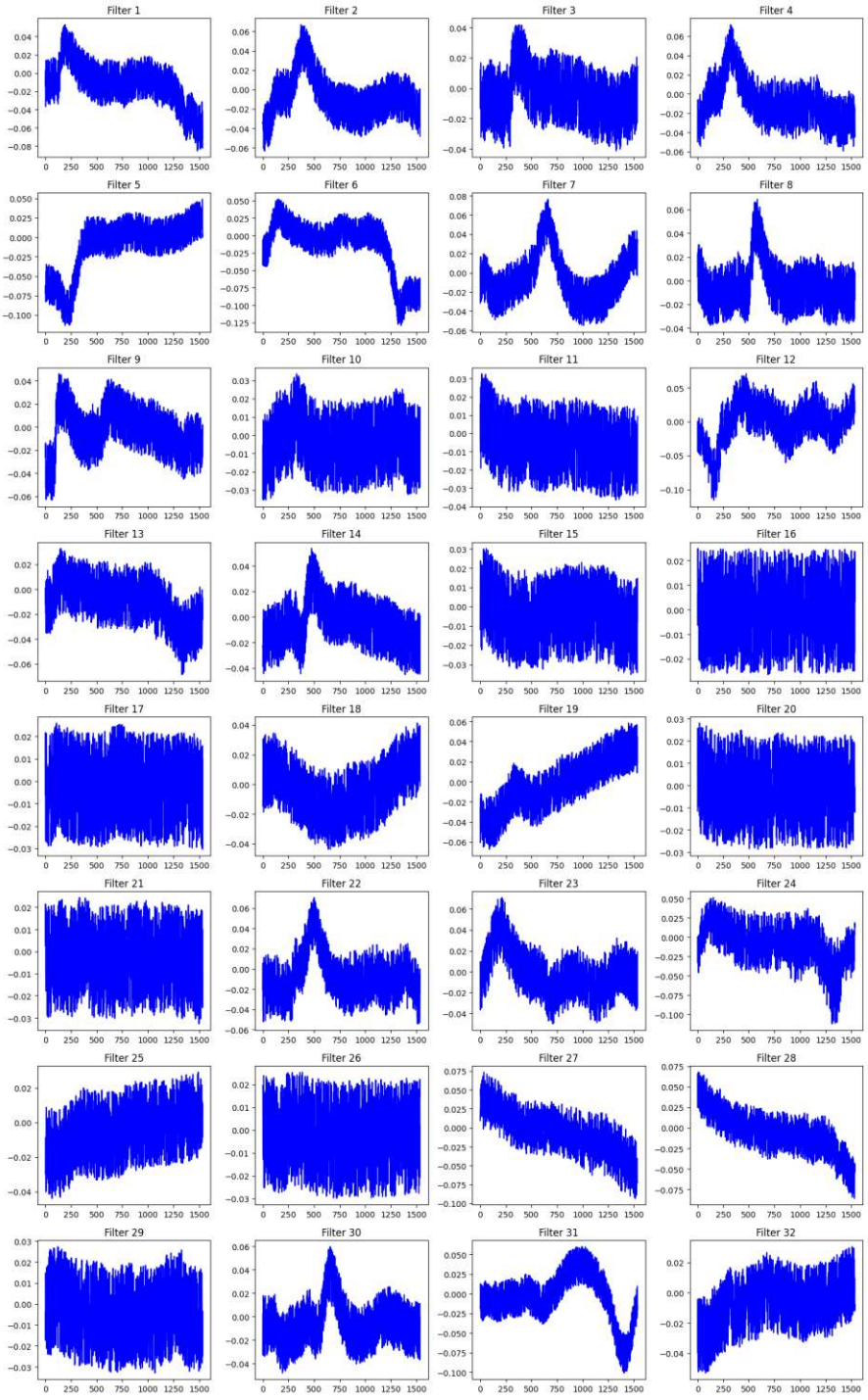


Figure 3.24: Example of CNN filters of a trained AutoCut model.

	Configuration 8		Configuration 9			
			Channel 0		Channel 1	
	Handmade	Autocut	Handmade	Autocut	Handmade	Autocut
True positive	2482	2456	1266	1278	1363	1376
True Negative	1915	1938	7657	7651	7510	7568
False Positive	56	33	38	44	74	27
False Negative	11	37	49	37	52	39
Bal. Accuracy	0.984	0.984	0.979	0.938	0.977	0.984
Precision	0.978	0.987	0.971	0.967	0.949	0.981
Recall	0.996	0.985	0.963	0.972	0.963	0.972

Table 3.4: Comparison of handmade cut with Autocut.

losses, decaying baselines, pile-ups and spikes. In the false negatives, a wrongly labelled spike event and a wrongly labelled noisy trace are visible. In Fig. 3.28 the corresponding plots for the AutoCut are shown. The false positives are mainly large flux quantum losses, pile ups and spikes, while the false negatives consist almost exclusively of lightly saturated pulses, indicating a possible under-representation in the training dataset.

Overall the AutoCut framework showed very promising performances on all tested datasets.

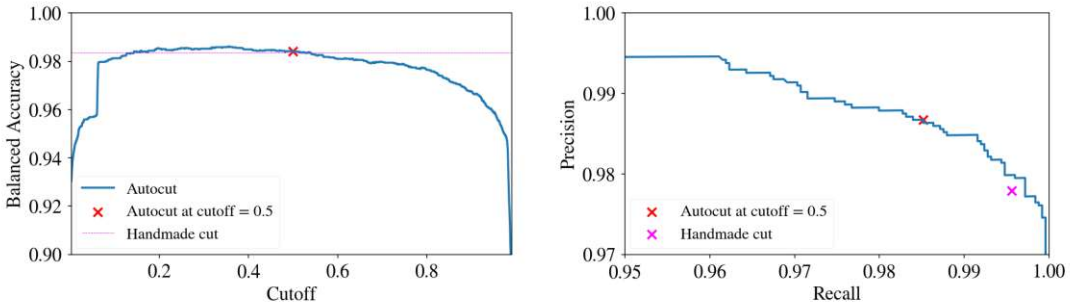


Figure 3.25: Comparison of balanced accuracy and precision over recall for configuration 8.

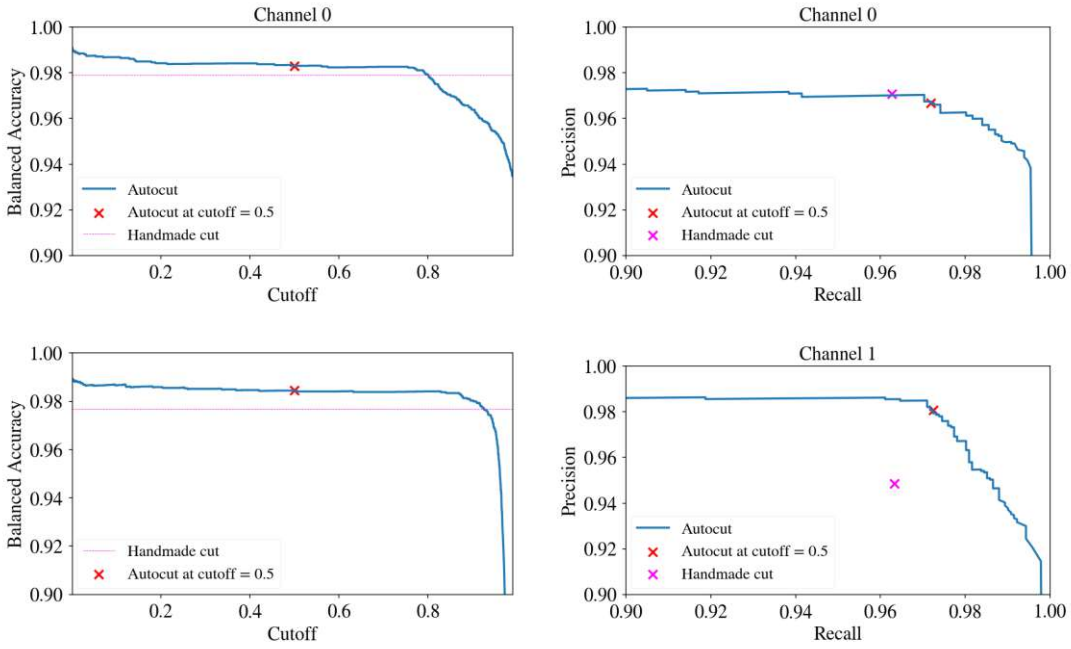


Figure 3.26: Comparison of balanced accuracy and precision over recall for both channels of configuration 9.

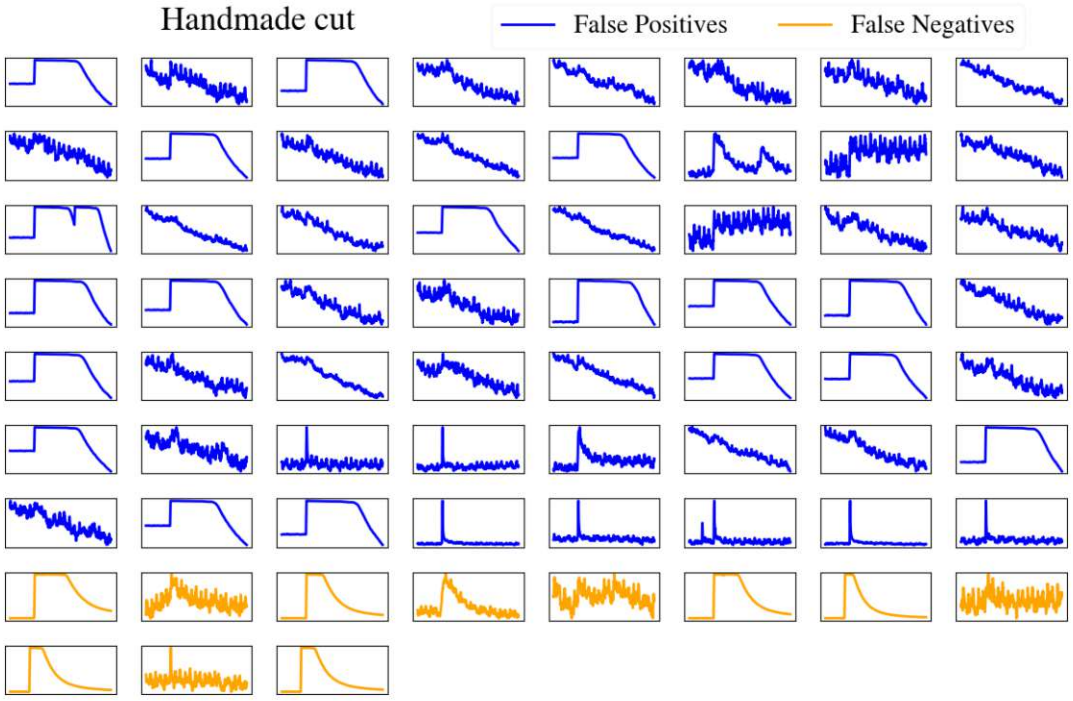


Figure 3.27: All false classifications for the handmade cut on configuration 8.

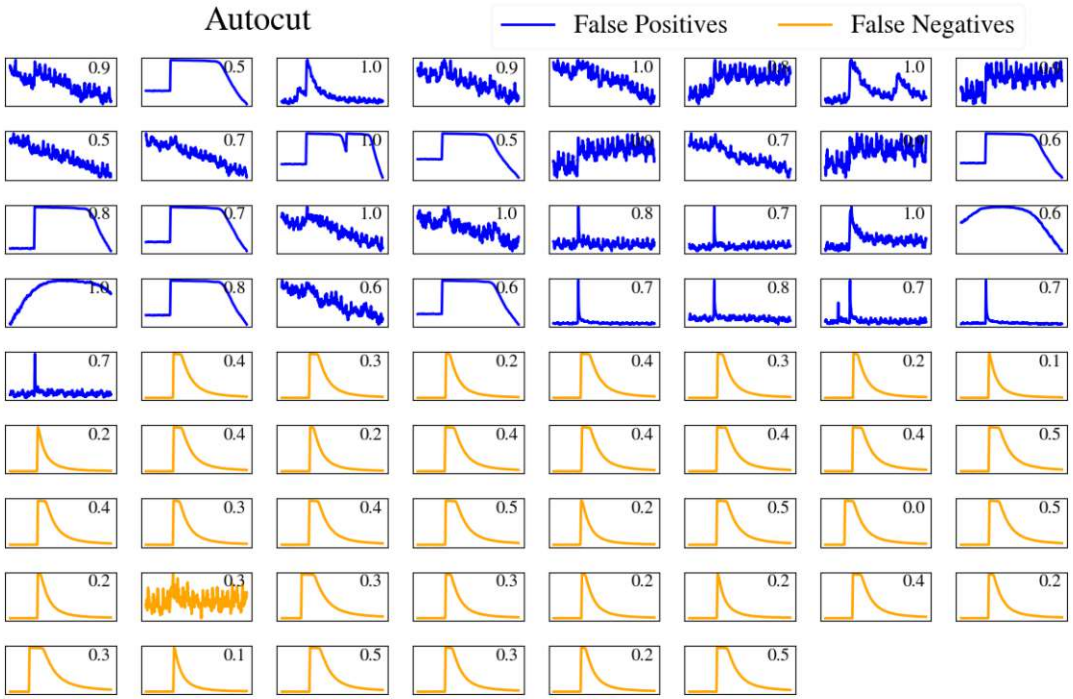


Figure 3.28: All wrong classifications for the Autocut on configuration 8. The numbers in the top right corner of each plot indicate the value of the AC for that trace.

Conclusion

In this work, the application of machine learning methods for raw data analysis of the NUCLEUS experiment has been explored.

In Section 3.1 the use of different kinds of neural networks for noise reduction of event traces has been investigated. The results obtained are promising; to consistently and significantly outperform state of the art techniques, more improvements are needed for these methods.

In Section 3.2 the k-means algorithm for clustering detector traces is introduced. In the configuration presented, it is used for clustering traces directly based on their shape and alignment instead of relying on calculated proxies for these values. The algorithm showed good performance and was used for the creation of labelled datasets during the course of this work. Also, it is a powerful tool for direct visualization of the contents of a dataset containing detector traces.

The trigger algorithm presented in Section 3.3 uses a trained neural network for the localization of particle pulses on the raw data stream. The neural network is trained using a labelled dataset that can efficiently be created using k-means clustering. The algorithm has the advantage of being able to differentiate particle events from artifacts already during triggering. Testing showed very promising results: even though the trigger efficiency of the state of the art optimum filter method is not yet matched for very low energetic pulses, for higher energetic pulses triggering and artifact rejection work very well, with datasets that have been triggered with the optimum filter and cleaned with manually defined cuts having largely the same make up as datasets triggered with the machine learning algorithm without cuts applied.

In Section 3.4 an algorithm for the automatic calculation of an optimum filter using raw data is presented. It can be used to trigger a raw data stream using the optimum filter and sensible trigger thresholds completely without user input. It consists of several parts, with the most involved one being the automatic selection of traces for the calculation of a standard event. This works in an iterative fashion by taking advantage the information provided by the parametric pulse shape description and the shape of test pulses. The algorithm yields standard events that are very similar to ones calculated by event selection with manually defined cuts. Further automatic cleaning for noise baselines via robust scaling enables the automatic estimation of the noise power spectrum and subsequent calculation of the optimum filter. Baseline resolution studies can be used to estimate trigger thresholds. Putting everything together enables completely automatized optimum filter based triggering, yielding almost identical results to triggering with an optimum filter calculated by using manually defined cuts.

In the final Section 3.5 a framework for artifact rejection is presented. It makes use of a comparatively small, shallow neural network that is intended to be trained on a per detector or configuration basis. For training, a labelled dataset is needed. The framework includes a weakly supervised method for the automatic creation of such a dataset, making use of k-means clustering and requiring only the selection of clusters as user input. This, together with quick training times enables rapid deployment of the method to previously unseen datasets. The artifact rejection power of the framework has been tested using a hand labelled dataset. The performance was

very good, matching manually defined cuts as well as previously reported results using artificial neural networks.

An overview of the ML methods developed in this work and their usage in the analysis of raw data is shown in Fig. 3.29.

All the methods developed in this work are already implemented by me in CAIT and are available for use in the analysis of raw data in the CRYOCLUSTER experiments (CRESST, COSINUS and NUCLEUS).

In conclusion, machine learning methods offer many possibilities for the automatization of raw data analysis tasks, potentially reducing human biases and facilitating the analysis of large amounts of data that might arise in the future of the NUCLEUS experiment.

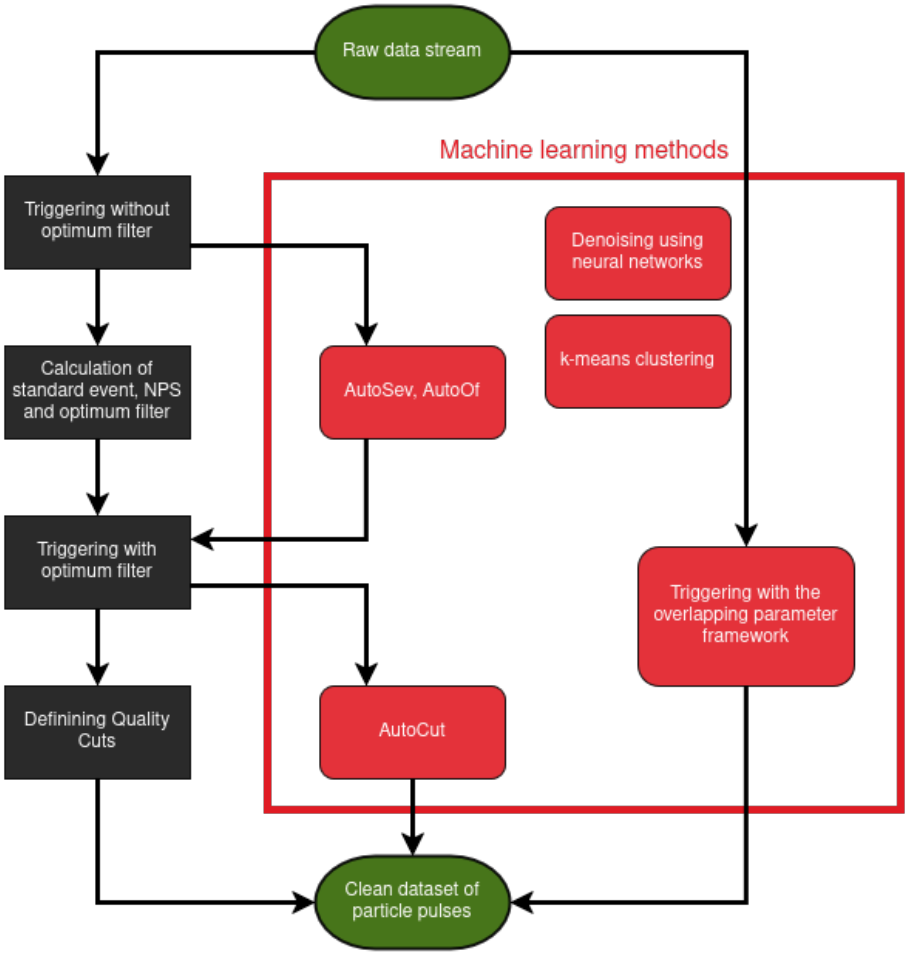


Figure 3.29: Overview of the ML methods developed in this work and their possible usage in the raw data analysis

Appendix A

Robust z-score

The z-score is a statistical measure that describes the position of a data point in relation to the mean of a dataset, in terms of standard deviations. Mathematically, this means the z-score Z is

$$Z = \frac{X - \mu}{\sigma} \tag{A.1}$$

with the sample mean μ and the sample standard deviation σ. This is useful because it can be used to transform different datasets to a comparable scale, regardless of their original units. It also allows for a scale independent, interpretable definition of cutoff values in units of σ.

The z-score is however sensitive to outliers, as it relies on sample mean and standard deviation, both of which can be heavily influenced by extreme values. To counteract this, the robust z-score [53] Z_{robust} is defined as

$$Z_{robust} = \frac{X - \tilde{X}}{\text{IQR}} \tag{A.2}$$

where IQR denotes the interquartile range, meaning the range between the first and third quartile, and $\tilde{X}$ denotes the median value of the dataset. Both the median and IQR are robust with respect to outliers.

Appendix B

Model definitions

All artificial neural networks presented in this work are implemented using the libraries PyTorch [54] and PyTorch Lightning [55].

Scaling Autoencoder The scaling autoencoder described in Section 3.1.1 has two branches, a LSTM branch and a convolutional branch, see Fig. 3.2 for illustration. The convolutional part of the network receives a version of the input that has been scaled to a maximum of 1, while the LSTM branch receives an unscaled version of the input. The The training loss used to train this network was the mean squared error (MSE) loss.

Encoder

- LSTM branch:
  - 3-layer LSTM with input size 8 and hidden size 80
  - Output passed through a linear layer to produce a vector of size 32
- Convolutional branch:
  - Conv1D(channels out=5, kernel=14, stride=2) → ReLU
  - Conv1D(channels out=5, kernel=8, stride=2) → ReLU
  - Conv1D(channels out=10, kernel=6, stride=2) → ReLU
  - Conv1D(channels out=10, kernel=3, stride=2) → ReLU
  - Conv1D(channels out=10, kernel=3, stride=2) → ReLU
  - Fully connected layer to output vector of size 64

Decoder

- Input: Concatenation of outputs of LSTM and Conv1D branches
- Linear layer followed by reshaping to (channels, length)
- Transposed convolutional layers:
  - ConvT(channels in=10, channels out=10, kernel=2, stride=2) → ReLU
  - ConvT(channels out=10, kernel=4, stride=2) → ReLU

  - ConvT(channels out=10, kernel=5, stride=2) → ReLU
  - ConvT(channels out=10, kernel=11, stride=2) → ReLU
  - ConvT(channels out=5, kernel=21, stride=2) → ReLU
  - ConvT(channels out=1, kernel=512, stride=1)
- Output is scaled by a the scalar output of a fully connected layer that takes the output of the LSTM branch of the encoder as input

Noise2Noise The Noise2Noise network consists of a LSTM followed by a fully connected layer. The input is padded at the beginning to avoid edge effects. The model output of the padded part of the input is ignored by the following fully connected layer. The training loss used was the MSE loss.

- Preprocessing: Reflective padding of 50 time steps is applied at the beginning of the sequence.
- LSTM Block:
  - 2-layer LSTM with hidden size 64
  - Dropout: 0.2 between LSTM layers
- Output Layer: A fully connected layer projects the LSTM output at each time step to a 1D noise estimate.
- Denoising: The estimated noise is subtracted from the input signal to produce the denoised output.

Overlapping parameter triggering network The neural network used for overlapping parameter triggering was a convolutional neural network with two convolution layers followed by a two fully connected layers. The loss used was the MSE loss.

- Convolutional layers:
  - Conv1D(channels out=64, kernel=512, stride=1, dilation=2) → MaxPool1D(4) → ReLU
  - Conv1D(channels out=32, kernel=128, stride=1, dilation=1) → MaxPool1D(4) → ReLU
- Fully connected layers:
  - Linear(output size=10) → ReLU
  - Linear(output size=1) → Sigmoid

AutoCut network The neural network trained for quality cuts consists of two layers, a convolutional layer and a fully connected layer. The loss used was the binary cross entropy loss.

- Convolution: Conv1D(channels out=32, kernel=1536, padding=204) → ReLU
- Fully connected: Linear layer with output dimension 1 → Sigmoid

References

[1] C. L. Cowan et al. "Detection of the Free Neutrino: a Confirmation". In: *Science* 124.3212 (July 1956). Publisher: American Association for the Advancement of Science, pp. 103–104. DOI: 10.1126/science.124.3212.103. URL: https://www.science.org/doi/10.1126/science.124.3212.103 (visited on 04/06/2025).

[2] S. Fukuda et al. "The Super-Kamiokande detector". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 501.2 (Apr. 2003), pp. 418–462. ISSN: 0168-9002. DOI: 10.1016/S0168-9002(03)00425-X. URL: https://www.sciencedirect.com/science/article/pii/S016890020300425X (visited on 04/06/2025).

[3] Gioacchino Ranucci et al. "Borexino". In: *Nuclear Physics B - Proceedings Supplements*. Neutrino 2000 91.1 (Jan. 2001), pp. 58–65. ISSN: 0920-5632. DOI: 10.1016/S0920-5632(00)00923-3. URL: https://www.sciencedirect.com/science/article/pii/S0920563200009233 (visited on 04/06/2025).

[4] S Adrián-Martínez et al. "Letter of intent for KM3NeT 2.0". en. In: *Journal of Physics G: Nuclear and Particle Physics* 43.8 (June 2016). Publisher: IOP Publishing, p. 084001. ISSN: 0954-3899. DOI: 10.1088/0954-3899/43/8/084001. URL: https://dx.doi.org/10.1088/0954-3899/43/8/084001 (visited on 04/06/2025).

[5] M.G. Aartsen et al. "The IceCube Neutrino Observatory: instrumentation and online systems". en. In: *Journal of Instrumentation* 12.03 (Mar. 2017), P03012. ISSN: 1748-0221. DOI: 10.1088/1748-0221/12/03/P03012. URL: https://dx.doi.org/10.1088/1748-0221/12/03/P03012 (visited on 04/06/2025).

[6] S. Aiello et al. "Observation of an ultra-high-energy cosmic neutrino with KM3NeT". en. In: *Nature* 638.8050 (Feb. 2025). Publisher: Nature Publishing Group, pp. 376–382. ISSN: 1476-4687. DOI: 10.1038/s41586-024-08543-1. URL: https://www.nature.com/articles/s41586-024-08543-1 (visited on 04/06/2025).

[7] Daniel Z. Freedman. "Coherent effects of a weak neutral current". In: *Physical Review D* 9.5 (Mar. 1974). Publisher: American Physical Society, pp. 1389–1392. DOI: 10.1103/PhysRevD.9.1389. URL: https://link.aps.org/doi/10.1103/PhysRevD.9.1389 (visited on 01/21/2025).

[8] D. Akimov et al. "Observation of coherent elastic neutrino-nucleus scattering". en. In: *Science* 357.6356 (Sept. 2017), pp. 1123–1126. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aao0990. URL: https://www.science.org/doi/10.1126/science.aao0990 (visited on 04/12/2023).

[9] D. Akimov et al. “First Measurement of Coherent Elastic Neutrino-Nucleus Scattering on Argon”. en. In: *Physical Review Letters* 126.1 (Jan. 2021), p. 012002. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.126.012002. URL: https://link.aps.org/doi/10.1103/PhysRevLett.126.012002 (visited on 04/12/2023).

[10] XENON Collaboration et al. “First Indication of Solar $ˆ{8}\mathrm{B}$ Neutrinos via Coherent Elastic Neutrino-Nucleus Scattering with XENONnT”. In: *Physical Review Letters* 133.19 (Nov. 2024). Publisher: American Physical Society, p. 191002. DOI: 10.1103/PhysRevLett.133.191002. URL: https://link.aps.org/doi/10.1103/PhysRevLett.133.191002 (visited on 01/20/2025).

[11] N. Ackermann et al. *First observation of reactor antineutrinos by coherent scattering*. arXiv:2501.05206 [hep-ex]. Jan. 2025. DOI: 10.48550/arXiv.2501.05206. URL: http://arxiv.org/abs/2501.05206 (visited on 01/20/2025).

[12] G. Angloher et al. “Exploring $$\hbox {CE}\nu \hbox {NS}$$ with NUCLEUS at the Chooz nuclear power plant”. en. In: *The European Physical Journal C* 79.12 (Dec. 2019), p. 1018. ISSN: 1434-6052. DOI: 10.1140/epjc/s10052-019-7454-4. URL: https://doi.org/10.1140/epjc/s10052-019-7454-4 (visited on 04/12/2023).

[13] F. Pröbst et al. “Model for cryogenic particle detectors with superconducting phase transition thermometers”. en. In: *Journal of Low Temperature Physics* 100.1 (July 1995), pp. 69–104. ISSN: 1573-7357. DOI: 10.1007/BF00753837. URL: https://doi.org/10.1007/BF00753837 (visited on 04/12/2023).

[14] E. Gatti and P. F. Manfredi. “Processing the signals from solid-state detectors in elementary-particle physics”. en. In: *La Rivista del Nuovo Cimento (1978-1999)* 9.1 (Jan. 1986), pp. 1–146. ISSN: 1826-9850. DOI: 10.1007/BF02822156. URL: https://doi.org/10.1007/BF02822156 (visited on 04/12/2023).

[15] A. H. Abdelhameed et al. “First results from the CRESST-III low-mass dark matter program”. en. In: *Physical Review D* 100.10 (Nov. 2019), p. 102002. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.100.102002. URL: https://link.aps.org/doi/10.1103/PhysRevD.100.102002 (visited on 06/21/2023).

[16] Felix Wagner. “Machine learning methods for the raw data analysis of cryogenic dark matter experiments”. en. In: (2020). Artwork Size: 123 pages Medium: application/pdf Publisher: TU Wien, 123 pages. DOI: 10.34726/HSS.2020.77322. URL: https://repositum.tuwien.at/handle/20.500.12708/15023 (visited on 04/12/2023).

[17] Felix Wagner. “Towards next-generation cryogenic dark matter searches with superconducting thermometers”. en. Accepted: 2023-12-28T11:31:02Z. Thesis. Technische Universität Wien, 2023. DOI: 10.34726/hss.2023.106550. URL: https://repositum.tuwien.at/handle/20.500.12708/190804 (visited on 04/09/2025).

[18] Felix Wagner et al. “Cait: Analysis Toolkit for Cryogenic Particle Detectors in Python”. en. In: *Computing and Software for Big Science* 6.1 (Dec. 2022), p. 19. ISSN: 2510-2044. DOI: 10.1007/s41781-022-00092-4. URL: https://doi.org/10.1007/s41781-022-00092-4 (visited on 01/20/2025).

[19] Mark Thomson. *Modern Particle Physics*. en. ISBN: 9781139525367 Publisher: Cambridge University Press. Sept. 2013. DOI: 10.1017/CBO9781139525367. URL: https://www.cambridge.org/highereducation/books/modern-particle-physics/CDFEBC9AE513DA60AA12DE015181A948 (visited on 01/21/2025).

[20] Manfred Lindner, Werner Rodejohann, and Xun-Jie Xu. "Coherent neutrino-nucleus scattering and new neutrino interactions". en. In: *Journal of High Energy Physics* 2017.3 (Mar. 2017), p. 97. ISSN: 1029-8479. DOI: 10.1007/JHEP03(2017)097. URL: http://link.springer.com/10.1007/JHEP03(2017)097 (visited on 04/12/2023).

[21] V. I. Kopeikin. "Flux and spectrum of reactor antineutrinos". en. In: *Physics of Atomic Nuclei* 75.2 (Feb. 2012), pp. 143–152. ISSN: 1562-692X. DOI: 10.1134/S1063778812020123. URL: https://doi.org/10.1134/S1063778812020123 (visited on 02/10/2025).

[22] Juan Barranco, Omar G Miranda, and Timur I Rashba. "Probing new physics with coherent neutrino scattering off nuclei". In: *Journal of High Energy Physics* 2005.12 (Dec. 2005), pp. 021–021. ISSN: 1029-8479. DOI: 10.1088/1126-6708/2005/12/021. URL: http://stacks.iop.org/1126-6708/2005/i=12/a=021?key=crossref.042c6879fade2a0d2f52b1905d6867e4 (visited on 04/12/2023).

[23] Bhaskar Dutta et al. "Sensitivity to Z-prime and non-standard neutrino interactions from ultra-low threshold neutrino-nucleus coherent scattering". In: *Physical Review D* 93.1 (Jan. 2016). arXiv:1508.07981 [hep-ph], p. 013015. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.93.013015. URL: http://arxiv.org/abs/1508.07981 (visited on 01/21/2025).

[24] M. Abdullah et al. "Coherent elastic neutrino-nucleus scattering: Terrestrial and astrophysical applications". In: (2022). Publisher: arXiv Version Number: 1. DOI: 10.48550/ARXIV.2203.07361. URL: https://arxiv.org/abs/2203.07361 (visited on 06/28/2023).

[25] Xin Qian and Jen-Chieh Peng. "Physics with Reactor Neutrinos". In: *Reports on Progress in Physics* 82.3 (Mar. 2019). arXiv:1801.05386 [hep-ex], p. 036201. ISSN: 0034-4885, 1361-6633. DOI: 10.1088/1361-6633/aae881. URL: http://arxiv.org/abs/1801.05386 (visited on 02/10/2025).

[26] The NUCLEUS collaboration et al. *Decoupling Pulse Tube Vibrations from a Dry Dilution Refrigerator at milli-Kelvin Temperatures*. arXiv:2501.04471 [physics]. Jan. 2025. DOI: 10.48550/arXiv.2501.04471. URL: http://arxiv.org/abs/2501.04471 (visited on 02/17/2025).

[27] *NUCLEUS*. URL: https://nucleus-experiment.org/ (visited on 02/17/2025).

[28] Dieter Meschede. "Festkörperphysik". de. In: *Gerthsen Physik*. Ed. by Dieter Meschede. Berlin, Heidelberg: Springer, 2015, pp. 853–926. ISBN: 978-3-662-45977-5. DOI: 10.1007/978-3-662-45977-5_19. URL: https://doi.org/10.1007/978-3-662-45977-5_19 (visited on 02/11/2025).

[29] Mario De Lucia et al. "Transition Edge Sensors: Physics and Applications". en. In: *Instruments* 8.4 (Dec. 2024). Number: 4 Publisher: Multidisciplinary Digital Publishing Institute, p. 47. ISSN: 2410-390X. DOI: 10.3390/instruments8040047. URL: https://www.mdpi.com/2410-390X/8/4/47 (visited on 05/02/2025).

[30] Johannes Felix Martin Rothe. "Low-Threshold Cryogenic Detectors for Low-Mass Dark Matter Search and Coherent Neutrino Scattering". PhD thesis. Technische Universität München, 2021. URL: https://mediatum.ub.tum.de/1576351 (visited on 02/19/2024).

[31] Lorenzo Périssé et al. "Comprehensive revision of the summation method for the prediction of reactor ${\overline{\ensuremath{\nu}}}_{e}$ fluxes and spectra". In: *Physical Review C* 108.5 (Nov. 2023). Publisher: American Physical Society, p. 055501. DOI: 10.1103/PhysRevC.108.055501. URL: https://link.aps.org/doi/10.1103/PhysRevC.108.055501 (visited on 05/03/2025).

[32] The NUCLEUS collaboration. "Commissioning of the NUCLEUS Experiment at the Technical University of Munich". In preparation.

[33] H. Abele et al. “Observation of a Nuclear Recoil Peak at the 100 eV Scale Induced by Neutron Capture”. en. In: *Physical Review Letters* 130.21 (May 2023), p. 211802. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.130.211802. URL: https://link.aps.org/doi/10.1103/PhysRevLett.130.211802 (visited on 06/21/2023).

[34] L.A. Vainshtein and V.D. Zubakov. *Extraction of Signals from Noise: By L.A. Wainstein and V.D. Zubakov*. Dover, 1970. URL: https://books.google.at/books?id=PTPSzAEACAAJ.

[35] M. Mancuso et al. “A method to define the energy threshold depending on noise level for rare event searches”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 940 (Oct. 2019), pp. 492–496. ISSN: 0168-9002. DOI: 10.1016/j.nima.2019.06.030. URL: https://www.sciencedirect.com/science/article/pii/S0168900219308708 (visited on 12/23/2024).

[36] CRAB Collaboration et al. *Observation of a nuclear recoil peak at the 100 eV scale induced by neutron capture*. arXiv:2211.03631 [nucl-ex, physics:physics]. Nov. 2022. DOI: 10.48550/arXiv.2211.03631. URL: http://arxiv.org/abs/2211.03631 (visited on 05/02/2023).

[37] Giorgio Del Castello. *Development of energy calibration and data analysis systems for the NUCLEUS experiment*. en. arXiv:2302.02843 [physics]. Feb. 2023. DOI: 10.48550/arXiv.2302.02843. URL: http://arxiv.org/abs/2302.02843 (visited on 01/15/2025).

[38] Martin Stahlberg. “Probing Low-Mass DarkMatter with CRESST-III - Data Analysis and First Results”. en. Artwork Size: 194 pages Medium: application/pdf Pages: 194 pages. PhD thesis. TU Wien, 2021. DOI: 10.34726/HSS.2021.45935. URL: https://repositum.tuwien.at/handle/20.500.12708/16833 (visited on 06/21/2023).

[39] M. Carrettoni and O. Cremonesi. “Generation of noise time series with arbitrary power spectrum”. In: *Computer Physics Communications* 181.12 (Dec. 2010), pp. 1982–1985. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2010.09.003. URL: https://www.sciencedirect.com/science/article/pii/S0010465510003486 (visited on 02/20/2024).

[40] Gopinath Rebala, Ajay Ravi, and Sanjay Churiwala. *An Introduction to Machine Learning*. en. Cham: Springer International Publishing, 2019. ISBN: 978-3-030-15728-9. DOI: 10.1007/978-3-030-15729-6. URL: http://link.springer.com/10.1007/978-3-030-15729-6 (visited on 04/04/2025).

[41] Felix Wagner. *Nonlinear pile-up separation with LSTM neural networks for cryogenic particle detectors*. arXiv:2112.06792 [physics]. Dec. 2021. DOI: 10.48550/arXiv.2112.06792. URL: http://arxiv.org/abs/2112.06792 (visited on 04/09/2025).

[42] G. Angloher et al. “Towards an automated data cleaning with deep learning in CRESST”. en. In: *The European Physical Journal Plus* 138.1 (Jan. 2023), p. 100. ISSN: 2190-5444. DOI: 10.1140/epjp/s13360-023-03674-2. URL: https://doi.org/10.1140/epjp/s13360-023-03674-2 (visited on 01/16/2024).

[43] Umberto Michelucci. “Autoencoders”. en. In: *Applied Deep Learning with TensorFlow 2: Learn to Implement Advanced Deep Learning Techniques with Python*. Ed. by Umberto Michelucci. Berkeley, CA: Apress, 2022, pp. 257–283. ISBN: 978-1-4842-8020-1. DOI: 10.1007/978-1-4842-8020-1_9. URL: https://doi.org/10.1007/978-1-4842-8020-1_9 (visited on 10/28/2024).

[44] Y. Ichinohe et al. "Application of Deep Learning to the Evaluation of Goodness in the Waveform Processing of Transition-Edge Sensor Calorimeters". en. In: *Journal of Low Temperature Physics* 209.5 (Dec. 2022), pp. 1008–1016. ISSN: 1573-7357. DOI: 10.1007/s10909-022-02719-7. URL: https://doi.org/10.1007/s10909-022-02719-7 (visited on 02/19/2024).

[45] Guillermo Iglesias et al. "Data Augmentation techniques in time series domain: a survey and taxonomy". en. In: *Neural Computing and Applications* 35.14 (May 2023), pp. 10123–10145. ISSN: 1433-3058. DOI: 10.1007/s00521-023-08459-3. URL: https://doi.org/10.1007/s00521-023-08459-3 (visited on 12/05/2024).

[46] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://ieeexplore.ieee.org/abstract/document/6795963 (visited on 05/07/2025).

[47] Jaakko Lehtinen et al. *Noise2Noise: Learning Image Restoration without Clean Data*. arXiv:1803.04189. Oct. 2018. DOI: 10.48550/arXiv.1803.04189. URL: http://arxiv.org/abs/1803.04189 (visited on 10/25/2024).

[48] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering – A decade review". In: *Information Systems* 53 (Oct. 2015), pp. 16–38. ISSN: 0306-4379. DOI: 10.1016/j.is.2015.04.007. URL: https://www.sciencedirect.com/science/article/pii/S0306437915000733 (visited on 11/18/2024).

[49] Junjie Wu. "Cluster Analysis and K-means Clustering: An Introduction". en. In: *Advances in K-means Clustering: A Data Mining Thinking*. Ed. by Junjie Wu. Berlin, Heidelberg: Springer, 2012, pp. 1–16. ISBN: 978-3-642-29807-3. DOI: 10.1007/978-3-642-29807-3_1. URL: https://doi.org/10.1007/978-3-642-29807-3_1 (visited on 11/18/2024).

[50] Peter J. Rousseeuw. "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis". en. In: *Journal of Computational and Applied Mathematics* 20 (Nov. 1987), pp. 53–65. ISSN: 03770427. DOI: 10.1016/0377-0427(87)90125-7. URL: https://linkinghub.elsevier.com/retrieve/pii/0377042787901257 (visited on 11/19/2024).

[51] Menouar Azib et al. *Event Detection in Time Series: Universal Deep Learning Approach*. arXiv:2311.15654 [cs, stat]. Dec. 2023. DOI: 10.48550/arXiv.2311.15654. URL: http://arxiv.org/abs/2311.15654 (visited on 06/12/2024).

[52] Menouar Azib et al. *A Comprehensive Python Library for Deep Learning-Based Event Detection in Multivariate Time Series Data and Information Retrieval in NLP*. arXiv:2310.16485 [cs] version: 2. Dec. 2023. DOI: 10.48550/arXiv.2310.16485. URL: http://arxiv.org/abs/2310.16485 (visited on 06/12/2024).

[53] Boris Iglewicz and David C. Hoaglin. *Volume 16: How to Detect and Handle Outliers*. en. Google-Books-ID: FuuiEAAAQBAJ. Quality Press, Jan. 1993. ISBN: 978-0-87389-260-5.

[54] *PyTorch*. en. URL: https://pytorch.org/ (visited on 04/23/2025).

[55] *Welcome to PyTorch Lightning — PyTorch Lightning 2.5.1 documentation*. URL: https://lightning.ai/docs/pytorch/stable/ (visited on 04/23/2025).