Investigating machine learning algorithms for signal and background discrimination in the ALPS TES detector

by Emre Toka

1st supervisor: Dr. Manuel Meyer
2nd supervisor: Dr. Gulden Othman

Institut für Experimentalphysik
Universität Hamburg
November 8, 2022

Eidesstattliche Erklärung

Ich versichere, dass ich die beigefügte schriftliche Bachelorarbeit selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle Stellen, die dem Wortlaut oder dem Sinn nach anderen Werken entnommen sind, habe ich in jedem einzelnen Fall unter genauer Angabe der Quelle deutlich als Entlehnung kenntlich gemacht. Dies gilt auch für alle Informationen, die dem Internet oder anderer elektronischer Datensammlungen entnommen wurden. Ich erkläre ferner, dass die von mir angefertigte Bachelorarbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung im Rahmen meines Studiums war. Die von mir eingereichte schriftliche Fassung entspricht jener auf dem elektronischen Speichermedium.

Ich bin damit einverstanden, dass die Bachelorarbeit veröffentlicht wird.

Ort, Datum

Unterschrift

Abstract

This thesis presents the analysis on light and background events measured with a Transition-Edge-Sensor. It will be implemented in the experimental setup of the Any Light Particle Search II experiment at DESY in Hamburg and is a key component of the setup. The Transition-Edge-Sensor is used for the detection of axions, which are postulated particles and a possible candidate for dark matter. They belong to the group of weakly interacting sub-eV particles. The analysis in this thesis includes the characterization of pulses from light and background events. Hereafter, machine learning algorithms have been applied to data of a test setup to determine their performance for signal and background discrimination. For this, the random forest algorithm and the K-nearest-neighbor algorithm have been used. Both classifiers provide promising results for the accuracy of signal and background discrimination. Especially, the evaluation by using the random forest algorithm gives the best accuracy of discrimination with $P_{acc-test} = 99.8826\%$. The statistical significance obtained for the K-nearest-neighbor algorithm, is slightly lower than the acceptance level of the model: $S = 3.12$. Lastly, the background events are investigated more in detail. For this purpose the Gaussian mixture model, which is a clustering algorithm based on machine learning, is applied to the background data. The goal of this procedure is to identify clusters from the background events recorded by the Transition-Edge-Sensor.

Zusammenfassung

Diese Arbeit präsentiert die Analyse von Licht- und Hintergrundereignissen, die mit einem Transition-Edge-Sensor gemessen wurden. Er wird in den experimentellen Aufbau des Any Light Particle Search II Experiments bei DESY in Hamburg implementiert und ist eine Schlüsselkomponente des Aufbaus. Der Transition-Edge-Sensor dient dem Nachweis von Axionen, die als Teilchen postuliert werden und ein möglicher Kandidat für dunkle Materie sind. Sie gehören zur Gruppe der schwach wechselwirkenden sub-eV-Teilchen. Die Analyse in dieser Arbeit umfasst die Charakterisierung von Pulsen aus Licht- und Hintergrundereignissen. Anschließend wurden Algorithmen des maschinellen Lernens auf Daten eines Testaufbaus angewandt, um ihre Leistung bei der Unterscheidung von Signal und Hintergrund zu bestimmen. Dazu wurden der Random-Forest- und der K-nearest-neighbor-Algorithmus verwendet. Beide Klassifikatoren liefern vielversprechende Ergebnisse für die Genauigkeit der Signal- und Hintergrundunterscheidung. Insbesondere die Auswertung mit dem Random-Forest-Algorithmus liefert die beste Unterscheidungsgenauigkeit mit $P_{acc-test} = 99,8826\%$. Die statistische Signifikanz, die für den K-nearest-neighbor-Algorithmus erhalten wurde, ist geringfügig niedriger als das Signifikanzniveau, bei dem das Model akzeptiert wird: $S = 3.12$. Schließlich werden auch die Hintergrundereignisse genauer untersucht. Zu diesem Zweck wird das Gaussian Mixture Model, ein auf maschinellem Lernen basierender Clustering-Algorithmus, auf die Hintergrunddaten angewendet. Ziel dieses Verfahrens ist es, Cluster aus den vom Transition-Edge-Sensor aufgezeichneten Hintergrundereignissen zu identifizieren.

Acknowledgments

Working on this thesis and also the way through my Bachelor studies have been an amazing experience. I learned very much and the most important thing for me is, that I developed myself. For the last few years, I would like to thank a few people.

First, I would like to thank Dr. Manuel Meyer, that he gave me the opportunity to work on a current project as this one. I had a lot of fun in all programming tasks and am thankful for that experience.

I would like to thank Dr. Gulden Othman for the support and input she gave me for my work.

I want to thank the Croissant group. Without you the tough courses of the last few years would have been even tougher.

Furthermore I would like to thank my friends for all the support you gave me over the time of my bachelor studies and for all the fun we had, where physics was not the topic. Finally, I want to say thank you to my family, especially to my parents. Thank you for all the support you gave me when I decided to study physics and that you believed in me.

Contents

1 Introduction

The search of particles beyond the Standard Model is a huge field of study in physics. One of the unsolved problems in the research field of physics is the explanation of dark matter. In the universe dark matter plays a major role. It is postulated that the contribution of dark matter to the entire energy in the universe is up to 30% (see [1] for review).

In the field of physics there are many approaches to identify the existence of dark matter. The Any Light Particle Search II (see [2]) at DESY is one experiment that provides an experimental setup and therefore an approach to detect dark matter. As a solution of the Strong CP-problem (see section 2), axions are theoretical particles which are postulated mathematically and are one of the candidates for dark matter. They belong to the group of weakly interacting sub-eV particles (WISPs). For the detection of axions, the Any Light Particle Search II experiment in Hamburg implemented a light shining through a wall system. It is based on the axion-photon conversion and therefore the main goal is to detect photons converted from axions. A major role in the experiment has the detector, for which two types of detectors are considered. First the heterodyne detector will be build in, which is not the subject of this thesis (see [2] for description of heterodyne detection). In this thesis, the focus is on the foreseen second detector in Any Light Particle Search II: A Transition-Edge-Sensor. In order to characterize the detector and its data acquisition it is important to analyse the detection scheme of light and background events with the goal to distinguish light pulses from background pulses. This will be the main part of this work.

The analysis of data is one of many important tasks, especially in particle physics. By examining data, it is possible to make statements about the physical processes and their occurrences in an experiment. Currently, machine learning is a great field which comes into operation in many areas for the purpose of analysis in particle physics. In particular for classification tasks the usage of machine learning is a common method [3]. In relation to the Any Light Particle Search II experiment and its Transition-Edge-Sensor, machine learning could help to discriminate actual signal events and background events which are detected as well.

This thesis focuses on the analysis of the data with the goal to understand the properties of the Transition-Edge-Sensor and the events measured. Since the detector is highly sensitive, the recorded background has a significant contribution to the data in total. In addition to that, a very small signal rate is anticipated for what reason the background must be suppressed as efficiently as possible. In order to understand the recorded events, the discrimination of signal and background events are analysed to create a classification. For that purpose, machine learning algorithms are used and optimized to make sure that the results are as accurate as possible. Furthermore the background events are investigated to create a classification again with the goal to characterize different background sources which have an impact on the Transition-Edge-Sensor.

2 Theoretical foundations

This chapter introduces the fundamental physical aspects which are relevant for this thesis. At first, in section 2.1, a theoretical motivation for axions and axion-like-particles (ALPs), as an extension of the Standard Model is discussed. In section 2.2 the physical background of axions and axion-like-particles are introduced. After this, the setup for the Any-Light-Particle-Search experiment (ALPS II) and its functionality are briefly explained in section 2.3. Finally, in section 2.4, the underlying physical and technical properties of the Transition-edge sensor detector as it will be used for ALPS II are described.

2.1 Motivation for extension of the Standard Model

The Standard Model (SM) (see figure1) gives a fundamental description of known elementary particles and most of their interactions with each other (see [4] for review). It is built on the one hand of fermions such as quarks and leptons. On the other hand there are bosons, which are the force carriers. In the SM three of the four fundamental forces are well described: the electromagnetic force, the weak force and the strong force. Even though the characteristics of many physical processes are successfully explained by this model, in fact it is incomplete. One of the major incomplete explanations is the gravitational force, which can not be taken into account in the SM. Since dark matter interacts via gravitational forces, it can not be considered either. However, due to the observation that around 30% of the energy in the universe is made of dark matter, it is a great indication that further elementary particles beside the SM do exist [1]. This motivates the search of particles beyond the SM.

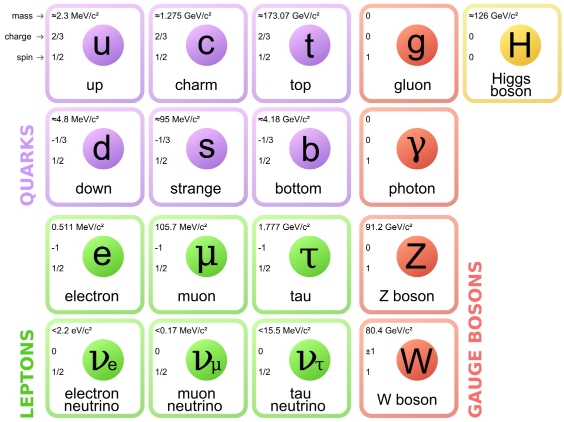

Figure 1: Table of Standard Model. It is a summary of all known elementary particles and it consists of the fermions(quarks and leptons) and the bosons, which are the force carriers. Figure from [4]

2.2 Axions & axion-like-particles

QCD-Axion

The axion is a possible candidate for dark matter. It is postulated as a solution for the strong CP-problem in quantum chromodynamics (QCD) and for that reason it is also called QCD-axion[5]. Experimentally, CP-violation is already known in weak interactions [6], but also QCD allows a CP-violating term in the Lagrangian:

$$\mathcal{L}_{CP-viol.} = \frac{\alpha_s}{4\pi}\theta \text{tr} G_{\mu\nu}\tilde{G}^{\mu\nu} \tag{1}$$

in which α_s is the strong coupling constant, $G_{\mu\nu}$ is the gluonic field strength and the parameter $\theta \in [0, 2\pi]$ describes an angle and can be found experimentally [7]. One well known approach to validate the strong CP-problem is the measurement of the electric dipole moment of the neutron. Theoretical considerations introduce an effective angle parameter $\bar{\theta} = \theta + det(M)$, where M is the quark mass matrix. From that, it can be derived that the electric dipole moment of a neutron is roughly [7]:

$$|d_n| \sim 10^{-16}|\bar{\theta}|e \text{ cm} \tag{2}$$

Experimentally, the measurement delivers an upper bound for the electric dipole moment of the neutron $d_n < 10^{-26}e$ cm [8] and thus for the effective angle parameter $|\bar{\theta}| < 10^{-10}$. The strong CP-problem describes the fact that the effective angle parameter $\bar{\theta}$ is suppressed to such a small value. One solution for this phenomenon was introduced by Peccei and Quinn, saying that a chiral symmetry gets spontaneously broken and generates a boson which is called ”axion” [9]. Axions are low in mass ($m_a <$ 1eV [2]) and are weakly interacting, which should not be mixed up with the weak force. It rather means that axions are rarely interacting and therefore they are categorized as Weakly interacting Sub-eV particles (WISP’s).

Axion-like-particles

The class of axion-like-particles(ALPs) can be derived from string theory [7]. In general, ALPs have the same characteristics as QCD-axions but do not solve the strong CP-problem. The main difference is that the coupling constant of ALPs with photons is not necessarily different than zero and in addition to that, the coupling constant is not proportional to its mass[7]. In the following, the QCD-axion and ALPs will be treated analogously as ALPs. ALPs can be generated by making use of the Primakoff effect, which states that photons can convert to ALPs by coupling to a magnetic field [2]. The conversion also occurs for ALPs to photons by the coupling of ALPs to a magnetic field. The Primakoff effect is the fundamental principle of the ALPS II experiment (see 2.3).

2.3 Experiment for the search of ALPs

The underlying concept of the Any Light Particle Search II (ALPS II) experiment is a light shining through a wall arrangement. All components of the experiment can be seen in figure 2.

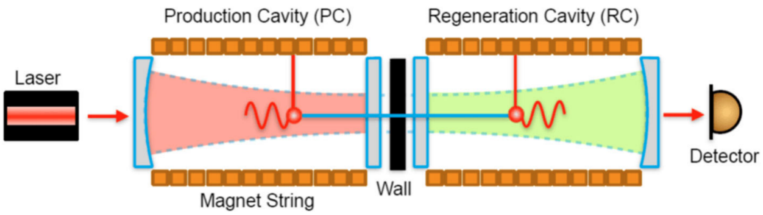

Figure 2: Experimental setup for the Any Light Particle search experiment [10]

This experiment makes use of the conversion of ALPs to photons. It is build in a straight part of HERA (former used particle accelerator) at DESY. In this arrangement a laser with wavelength $\lambda = 1064$nm is used as a photon source and a detector (see next section 2.4) to record the signal. In the middle of the setup, in a distance of 106m [2] from each side, a barrier is placed, which the photons can not cross. A static magnetic field of $B = 5.3$T is created by dipole magnets from HERA. They are located on both sides of the barrier (see figure2). The fundamental idea is to make use of the effect that photons, which are passing through a magnetic field, can be converted to ALPs. This effect is the so called Primakoff effect[2]. The basic idea of the conversion process

can be seen in figure3. Since ALPs interact weakly with matter, they can pass the wall which is not valid for photons. After crossing the wall, this time the ALPs can couple with the magnetic field and convert back to photons again. The eventually back-converted photons are measured with the detector.
Since the coupling coefficient of ALPs to photons is rather low ($g_{a\gamma\gamma} \leq 2 \cdot 10^{-11}\text{GeV}^{-1}$ [2]), the expected photon rate is at 1 photon per day.

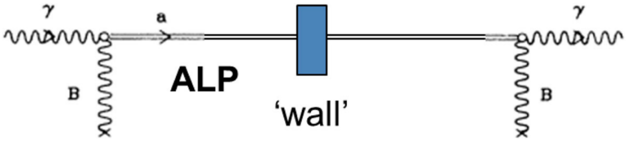

Figure 3: Photons propagating through a magnetic field get converted to ALPs which can pass the wall. Behind the wall, they convert back to photons in a second magnetic field. [10]

2.4 Detection system

For the ALPS II experiment, two different types of detection principles are considered to be used in the arrangement. First the heterodyne detector will be implemented, which is not covered in this thesis. As a second detector a Transition-Edge-Sensor (TES) is planned to be deployed and it is the main subject of this work. A TES is a superconducting microcalorimeter and makes use of its thermal and electrical components in a way that energy depositions of single photons can be detected. Since the expected photons arriving at the detector from ALPs conversion have low energy ($E_\gamma \approx$ 1eV) and the expected photon rate is rather low, the detector must be high in sensitivity and energy resolution [11]. Due to its characteristics, a TES is perfectly suited for the detection of photons in ALPS II.

2.4.1 Superconductivity

The funcionality of a Transition-edge sensor (TES) is based on superconductivity. Superconductivity is a specific property of certain conducting material. Below a certain temperature, the so called critical temperature T_C, the current flows without resistance. The fundamental reason for this behaviour of the conducting material is described by BCS-theory (J. Bardeen, L.N. Cooper, J.R. Schrieffer - 1957) which results in the creation of Cooper-Pairs. Two electrons with anti-parallel spin and opposite momenta in a lattice can be attracted to each other and create a Cooper-pair [12]. Although electrons are fermions, the Cooper-pairs behave like bosons. They can stay in the same quantum-mechanical state and as a result, the system of Cooper-pairs can be described as one macroscopic wave function [13]. A supercurrent arises and flows with zero resistance [14].

2.4.2 TES-detector

The core component of the TES-detector is the microchip. In the setup of ALPS II a Tungsten microchip (25µm × 25µm, 20nm thick) is used and is developed by NIST (National Institute for Standards and Technology, USA) [11]. The absorptivity of pure Tungsten is rather low. Measurements of the quantum efficiency for pure Tungsten at 1310nm and 1550nm gives a quantum efficiency $QE \approx$ 20% [14]. To increase the absorptivity and with that the quantum efficiency the microchip has been stacked with multiple layers of other material, mainly Silicon (for further information see [14], [11]). Through that, the absorptivity of the microchip increased to 97% for 1550nm [14].

The working principle in the TES for ALPS II is based on the change of resistance of the Tungsten chip. It is on one side connected to a thermal bath which will be cooled down. The exact temperature of the thermal bath in the ALPS II setup is not known yet. Previous characterizations of the TES made it possible to cool down the bath to $T \approx 80\text{mK} \pm 25\text{mK}$ [14]. On the other side, the chip is connected to an electrical circuit. The circuit is connected to a constant bias voltage, which is applied through the TES.

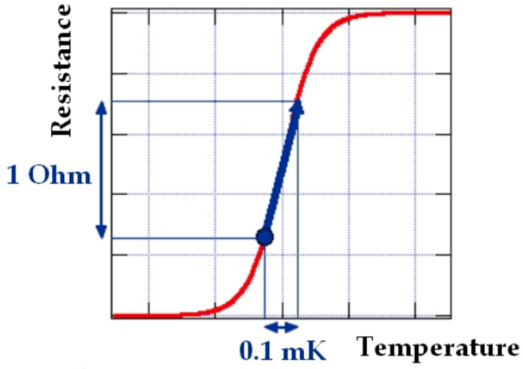

Figure 4: Behaviour of resistance and temperature for a scenario where one single photon with wavelength 1064nm deposits its energy to the Tungsten Chip. Figure from [15]

The microchip is adjusted to its working point, where it is in the transition region between conducting and superconducting. If now a 1064nm photon hits the chip, its energy is absorbed and the resistance increases by $\sim 1\Omega$ for a temperature increase of $\sim 0.1\text{mK}$ [15]. As a result, the TES is not superconducting anymore, but rather normal conducting, depending on the energy deposition. This change in the transition region of the TES can be seen in figure 4.

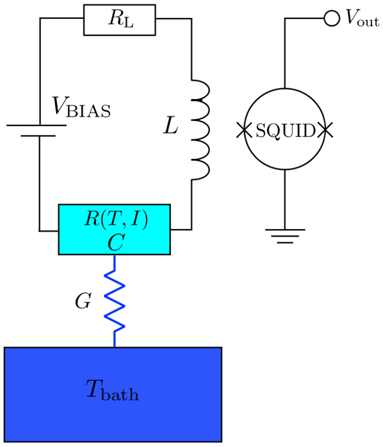

Figure 5: Circuit of a transition-edge sensor. The Tungsten chip with its resistance R(T, I) as a function of the temperature T and the current I is connected to a thermal bath with thermal conductivity G. A bias voltage V_{bias} is applied through the circuit. The electromagnetic coil L creates the magnetic field which is read out by SQUIDs. The resistance $\mathrm{R_L}$ describes the parasitic resistance of the coil. Figure from [14]

For the purpose of registering the change in current, due to the change in conductivity, an electromagnetic coil is incorporated. When the current is changing, the strength of the magnetic field will change as well. A general sketch of the electrical circuit is given in figure 5. The change of magnetic field strength can be read out by a Superconducting Quantum Interference Device (SQUID). SQUIDs are useful to detect slight changes in magnetic field strengths. It consists of a superconducting loop with 2 zones in the loop, where the material is not a superconducter or at least the superconduction is weaker [16]. The external magnetic flux creates a current in the superconducting loop. Due to the so called Josephson junctions, the voltage is fluctuating periodically, which can be recorded. Since the change in voltage is rather small, it must be amplified with electronics.

Further information and detailed discussions about the properties of SQUIDs can be found in [16].

3 Preliminary work on data

This chapter introduces the preliminary work that is made for the further analysis. In section 3.1 the first processing of the data is described. Essencially, the fitting of the data with the purpose to obtain the parameters, which will be the base of the further work, with regard to use machine learning for classification tasks, are explained.
Hereafter, in section 3.2, the mathematical foundations of the Principal Component Analysis are summarized and it is applied on the parameters coming from the fitting process, with the goal to prepare the parameters and improve the outcome of the used machine learning algorithms.

3.1 Fitting procedure

As explained in section 2.4, the change of the magnetic field strength, detected by the SQUIDs, is converted to a voltage which os read out. The readout of the SQUIDs is the data which will be analyzed. A timeline measured with the detector contains the voltage values, and the time when the voltage is measured. A distinction is made between light pulses and background pulses. The data used in this thesis is obtained by using a 1064nm laser connected to the TES via an optical fiber. The pulses caused by the laser light are called light pulses. On the other hand, the data collected without a light source and where the fiber is unplugged are called background pulses. In particular, these kind of pulses are categorized in the group of the intrinsic background. Radioactive sources in the detector material or cosmic rays are examples for the intrinsic background pulses. There are also extrinsic background pulses, which describe the pulses where the fiber is connected to the TES, but no photons are send from the laser. Since the TES is cooled down to $\sim$ 80mK, the room in which the TES is located, can be interpreted as a black body. In particular, photons from the black body radiation which arise at the other end of the fiber at room temperature, can reach the TES via the fiber. However, this thesis only focusses on the intrinsic background. For the analysis, there are all together 2627 light pulses and 39580 background pulses collected over an observation time of 517.91 hours with 10000 data values in each timeline. The sampling rate for the readout is f = 50MHz. One of each pulses is shown in figure 6, which shows a light and a background pulse how it is commonly expected with its typical time length of 200μs.

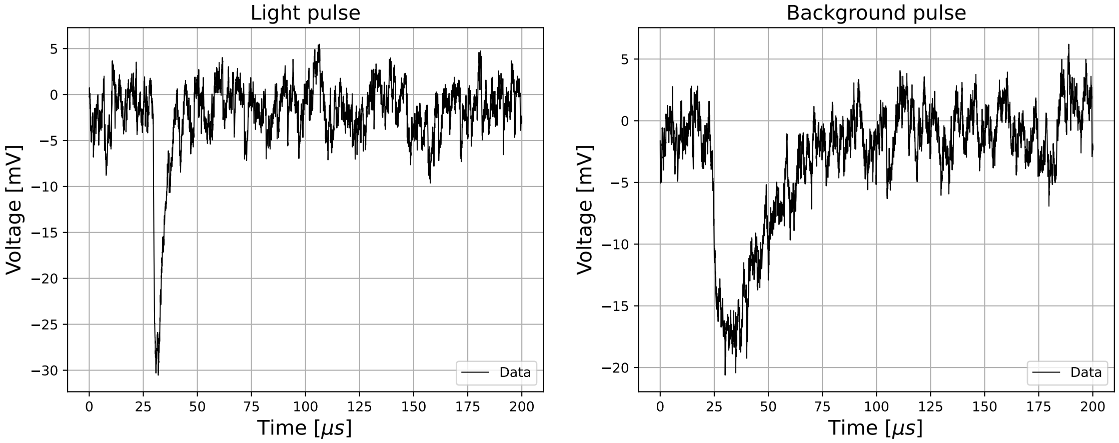

Figure 6: On the left a light pulse and on the right a background pulse. The shapes of the pulses are characteristic for light/background.

To achieve a better understanding of the data, it is helpful to fit a model function. With the aid of this function, one can obtain parameters which describe the shape and properties of the light or background pulses. The function used to fit the data is the following:

$$V(t) = c - 2A \cdot \left(\exp(-\frac{t - t_0}{\tau_{rise}}) + \exp(\frac{t - t_0}{\tau_{decay}}) \right)^{-1} \tag{3}$$

The parameters in this model function are the constant c, the amplitude A, the decay time τ_{decay}, the rise time τ_{rise} and the trigger time t_0 which will be determined through the fitting process. The fit is implemented with the Python module *iminuit*. For fitting the model to the data *iminuit* takes a so called cost function, which will be the function to minimize and in addition to that it takes an initial guess for the parameters. As a cost function, the built-in least sqaures function is used, which is calculating the sum of squares for the residuals of the model and the data [17]. The least-squares function needs also an error as in input which is chosen to $\sigma = 1.5$mV. With that one can find the fit parameters and their errors as well. Another advantage of *iminuit* is that the χ^2 can also be calculated through the fitting procedure, by which the reduced $\chi^2/ndof$ with ndof="number degrees of freedom" can be calculated easily. Further information about *iminuit* is given in the documentation in [17].

As mentioned above, the parameters can describe the properties of a pulse. Mathematically, the parameters represent the following:

- **Trigger time t_0:** The trigger time marks a time value at which the fitted pulse is between the starting point of the dip and the maximum value in voltage.
- **Rise time τ_{rise}:** The rise time describes how much time it takes for the pulse to get to the maximum value.
- **Decay time τ_{decay}:** The decay time characterizes the time the pulse takes to get back to normal level after the maximum.
- **Amplitude A:** The amplitude determines how high in voltage the fitted pulse is.
- **Constant c:** The constant represents a constant value at which the noise of the pulse is.

Since *iminuit* takes initial guesses for each parameter as an input for the fit, it is necessary to make these guesses as accurate as possible. By doing that, the minimizing process in *iminuit* can achieve a better estimation of the parameters. The choices of the starting values which are given in the Fit are the following:

- **Trigger time:** First, a lowpass filter has been applied on the data and after that, the derivative is calculated numerically. The peaks of the derivative are identified and they are used for the estimation for t_0. The purpose of the lowpass filter is to suppress the noise for high frequencies. With aid of the lowpass filter high frequencies are dampened. The minimum value of the derivative is chosen as the inital guess for t_0. The observation shows that the value for t_0 in the timelines fluctuates around 30µs.
- **Rise time:** $\tau_{rise} = 2$µs
- **Decay time:** $\tau_{decay} = 2$µs
- **Amplitude:** $A = 25$mV
- **Constant:** $c = 0$

As an example, the result of the fitting procedure is given in figure 7 where on the left the model is fit to a light pulse and on the right a background pulse. These shapes of the model curves are typical for light and background pulses, respectively.
Under consideration of the pulses, there are some differences between light and background pulses which can be seen by the shape. The light pulses seem to be sharper in its peak, which means that the rise and decay times are smaller than in a background pulse. In particular, the decay time for a background pulse is much higher. In addition to that, the peak of a light pulse is higher. Another noticeable aspect is that the integral of the pulses seem to be different. For that reason the pulse integral is also considered for the further work, especially in section 4 & 5.

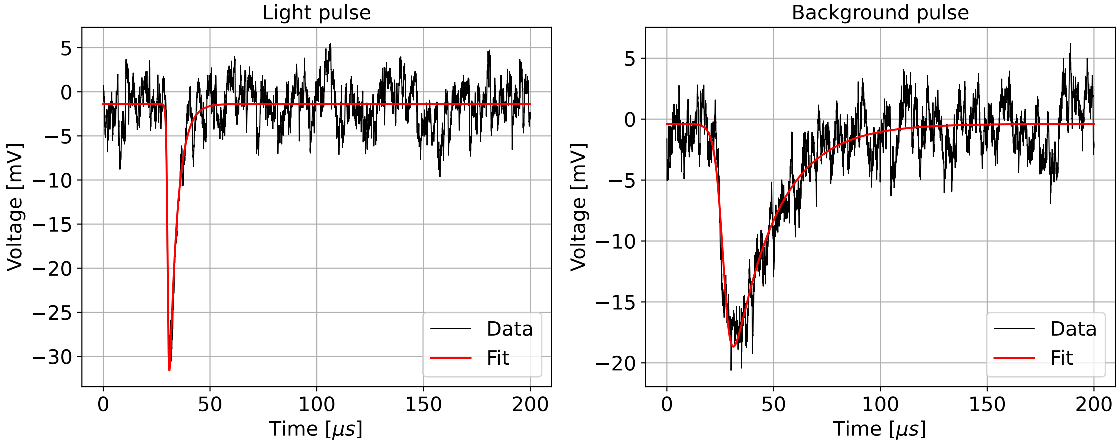

Figure 7: Fit of the model function $V(t)$ to pulses coming out of the TES. On the left a typical light pulse and on the right a typical background pulse is illustrated.

Trigger time t_0:	$30.201\mu s \pm 0.013\mu s$
Rise time τ_{rise}:	$0.344\mu s \pm 0.01\mu s$
Decay time τ_{decay}:	$3.84\mu s \pm 0.04\mu s$
Amplitude$A$:	$20.07mV \pm 0.15mV$
Constant c:	$-1.383mV \pm 0.016mV$
χ^2/ndof:	2.449

Table 1: Values of parameters for light pulse from figure7 which are achieved by the fit.

Trigger time t_0:	$26.82\mu s \pm 0.06\mu s$
Rise time τ_{rise}:	$2.10\mu s \pm 0.04\mu s$
Decay time τ_{decay}:	$20.13\mu s \pm 0.17\mu s$
Amplitude$A$:	$12.50mV \pm 0.17mV$
Constant c:	$-0.388mV \pm 0.020mV$
χ^2/ndof :	1.675

Table 2: Values of parameters for background pulse from figure7 which are achieved by the fit.

The parameters obtained by the fitting process for the light and background pulses in figure 7 can be seen in Table 1 & 2. As already visible by eye, the amplitude A is higher for the light pulse, and the rise- and decay times of the background pulse are larger. Especially the decay time is significantly higher.

The procedure of obtaining the parameters by fitting the model to the data is repeated for all timelines. Therefore, in total 42207 timelines are taken into consideration. To get an overview of the parameters and their distribution it is useful to plot histograms which can be seen in figure 8.

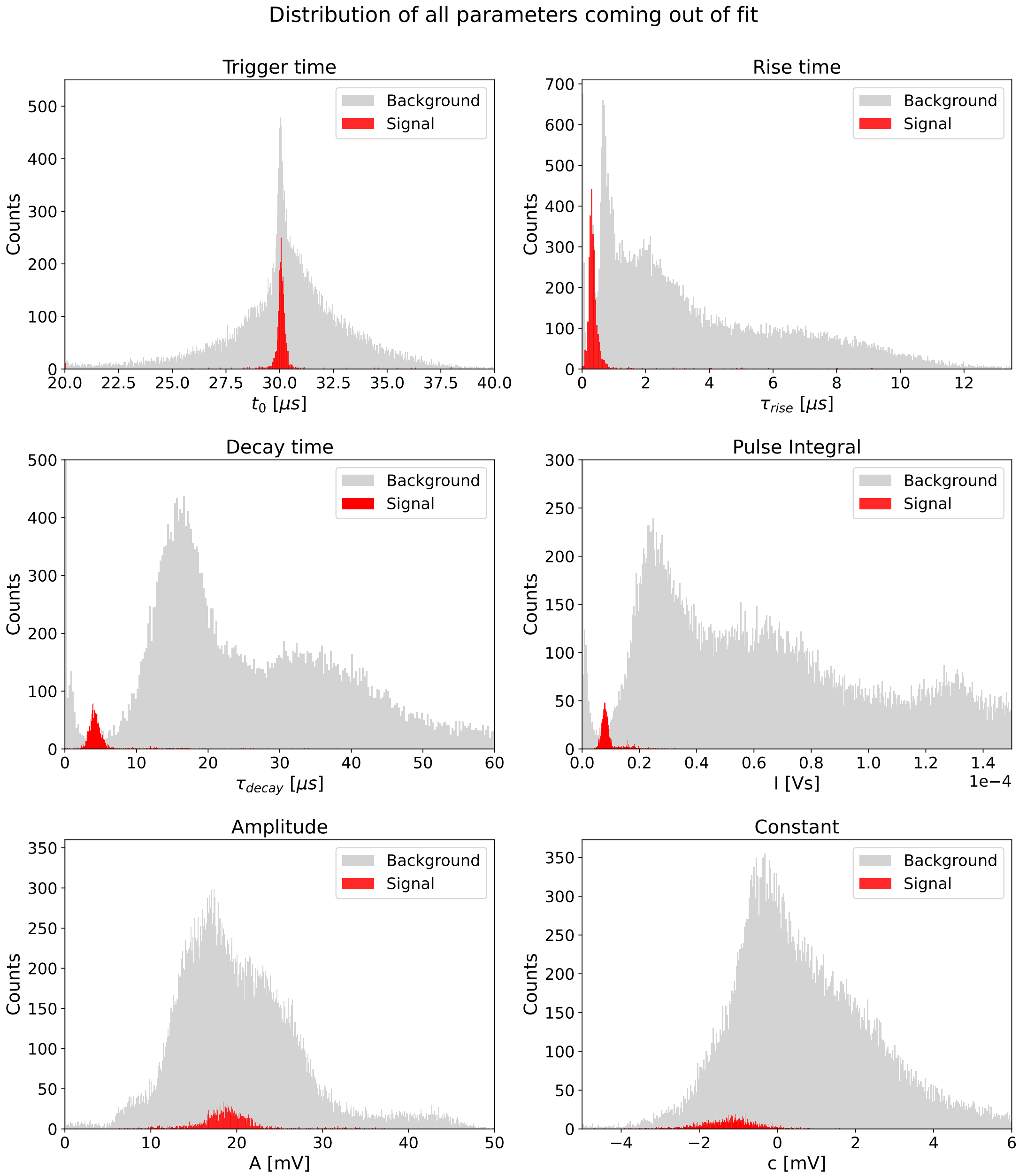

Figure 8: Distribution of the parameters for light and background pulses. The pulse integral for each timeline is also shown.

Signal pulse integral I_{signal}	$7.790 \cdot 10^{-6}$ Vs
Background pulse integral $I_{background}$	$25.538 \cdot 10^{-6}$ Vs

Table 3: Calculated integrals of fitted model function for signal and background pulse

In addition to the parameters as given in the function $V(t)$ from equation 3, the pulse integral is calculated. The integral of each timeline is computed by integrating over the fit function with the given parameters. After that the absolute value is taken. The unit of the pulse integral is $Voltage \times time$ $[Vs]$. For the given pulses in figure 7, the values for the pulse integrals are given in Table 3. Due to the small amount of light pulses in comparison to background pulses, the counts of the background are much higher. For that reason, the light counts in the histograms seem to be very small. By looking at the histogram of the pulse integral, it is noticeable that there are many pulses whose integral is closer to zero than the integral computed for light pulses. This could result from the fitting process, where some pulses can not be characterized properly. However, it could be beneficial to investigate the reduced χ^2. The distribution of the reduced χ^2 can be seen in figure 9.

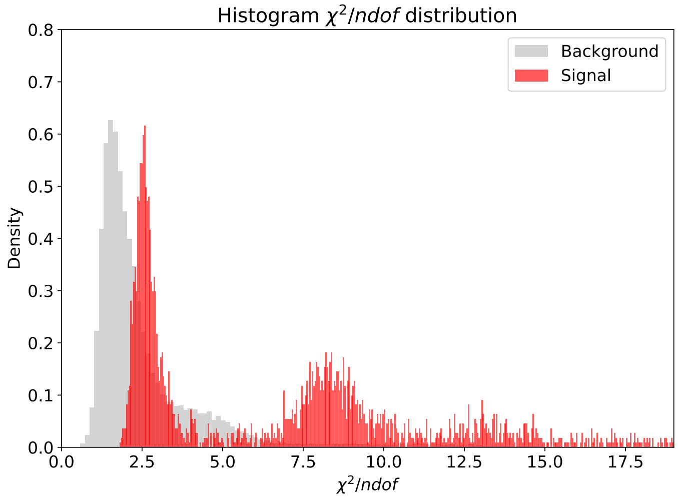

Figure 9: Reduced χ^2 distributions for all light and background pulses.

To take a better look at the reduced χ^2 the counts in the histogram above are normalized in a way that the integral over the distributions is equal to one for both the backgrounds and also for the signals. Statistically, the goal is to have $\chi^2/ndof = 1$. It gives information about how good the fit actually worked. As it can be seen, in figure 9, the values for the reduced χ^2 have a large spread. Since in this thesis the parameters play the main role for the analysis, it was decided to make a cut. In particular, the analysis for the discrimination of signal and background in section 4 and the background analysis in section 5 considers only the timelines and therefore the parameters for which $\chi^2/ndof < 6$ is valid.

One goal in this thesis is the classification of signal and background events. By looking at the histograms in figure 8, it is noticeable which parameters could be used to this distinction. For instance, the trigger time t_0, the amplitude A and the constant c look quite similar to the background, just with less counts due to the unequal amount of timelines for light and background. However, for the rise- and decay time τ_{rise} & τ_{decay} and for the pulse integral it is visible that the signal distribution of these parameters are distinguishable from the background distribution. This first observation is used and discussed more in detail in section 4.

3.2 Principal Component Analysis

The Principal Component Analysis is a widely used technique to preprocess data [18]. Since one of the main goals in this thesis is the classification of signal and background pulses, applying Principal Component Analysis to the data can help increasing the success of classification tasks. In the first section the mathematical procedure of Principal Component Analysis is briefly explained. The main reference for section 3.2.1 from which all information are taken is [18] in which the mathematical discussions of Principal Component Analysis can be found in more detail. In the second section the Principal Component Analysis is applied to the parameters and the results are discussed.

3.2.1 Theory

The Principal Component Analysis (PCA) is a mathematical method. It can be used to decrease the dimensionality of a large dataset with minimum information loss, which makes it easier to interpret the data. This will be used in section 4 & 5. Next to the reduction of dimensionality, PCA is also useful to improve the results in classification tasks. Since the principal components can be seen as vectors along the axes with the highest covariances, it might help classifiers to obtain better results. An example for that is shown in figure 10.

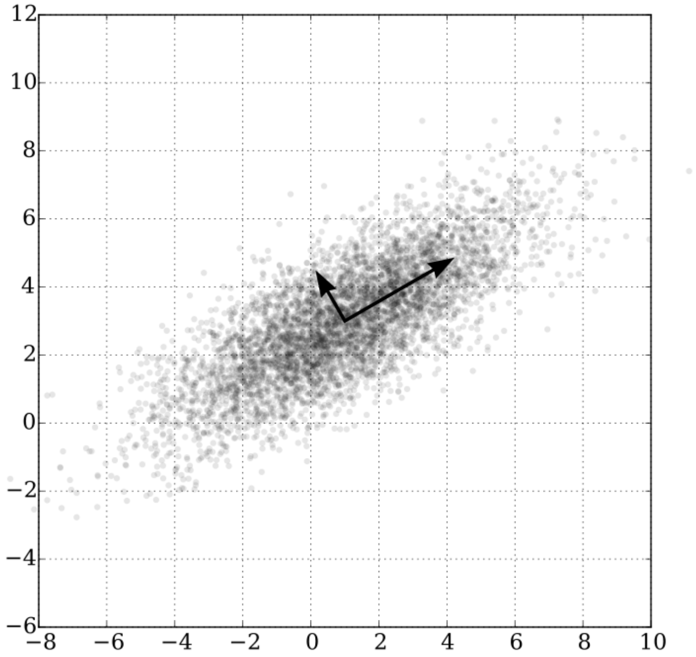

Figure 10: Illustration of principal components, which are represented by the black arrows. Figure from [19].

In general, a dataset can be described as a matrix X with the dimension $n \times p$, where n is the number of rows and p the number of columns. The columns of the matrix are noted as x which represent column vectors. The goal of the PCA technique is to find a linear combination of all vectors x which are the columns of X where the covariance matrix $\Sigma = \frac{1}{p} X X^T$ [20] is maximized. We introduce a vector a in which the entries are constant values and their dimension is the same as the dimension of x. With aid of the vector a the linear combinations can be written as $\sum_{i=1}^{p} a_i x_i = Xa$, for which the variance can be computed by $var(Xa) = a^T \Sigma a$. For the purpose of maximizing the covariance matrix Σ, it is made use of the method of Lagrange-multiplier (for further information, see [18]). The maximizing problem results in the equation $\Sigma a = \lambda a$ where λ is the Lagrange multiplier. It can be seen that a describes the eigenvector of the matrix S with λ as its eigenvalues. For the PCA the largest eigenvalues λ_k are interesting, since they represent the highest variances. The principle components as an outcome of the PCA are the linear combinations of the data matrix X multiplied with the corresponding eigenvector a_k from the largest eigenvalues λ_k. These linear combinations or principle components produce the eigenspace with dimension p, which can be chosen beforehand.

3.2.2 Principal Component Analysis of parameters

With the purpose to make the data more suitable for the further work, the PCA is applied to the parameters. Besides the parameters that are coming from the model function $V(t)$, the pulse integral I and the reduced χ^2 are appended to the matrix of parameters. As a result, the matrix has seven columns: $t_0,\ \tau_{rise},\ \tau_{decay},\ A,\ c,\ \chi^2/ndof\ \&\ I$.

In advance, the parameters are getting standardized. This means that all fit parameters belonging to one variable from the fit are processed in a way, so that the mean value becomes zero and the standard deviation one. To implement that, the *StandardScaler* module from scikit-learn [21] is used. The standardization is a useful tool, especially with regards to machine learning applications. By using this, the machine learning algorithms may provide results that are more meaningful. After the standardization, the PCA is applied to the new matrix containing the standardized parameters. The PCA is implemented by the PCA funktion from scikit-learn. The eigenspace of the principle components has the dimension seven, since all seven parameters have been used for the PCA.

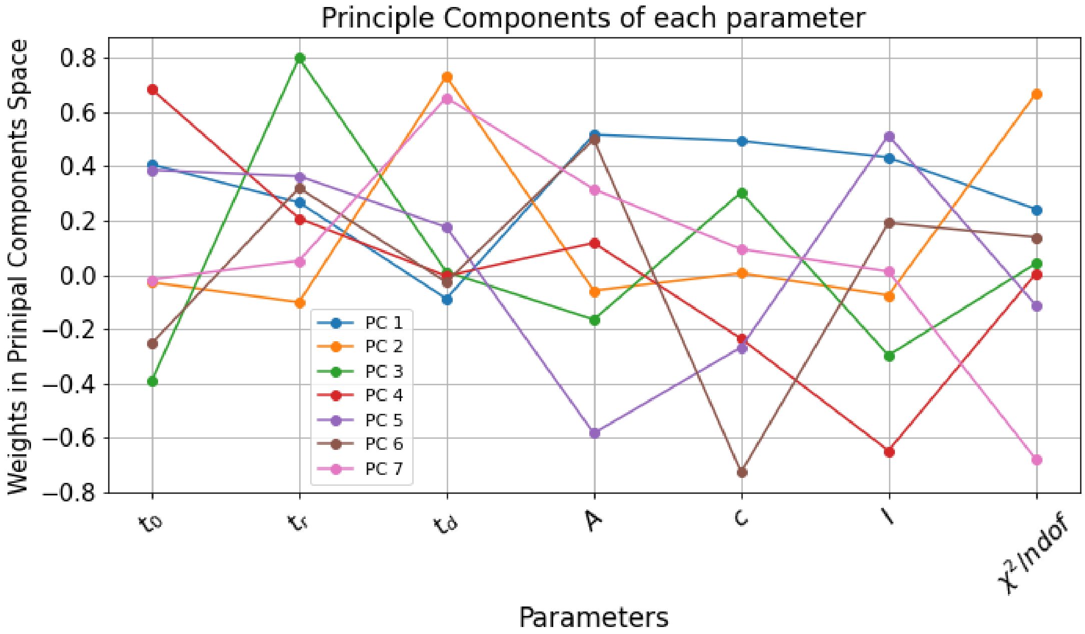

Figure 11: Illustration of the principle components. Each line corresponds to one component in the eigenspace of the principle components. This plot shows the weights in every principle component in its eigenspace for every parameter.

In figure 11 the different principle components are shown. Each colored line represents one principal component. From this, it is observable how the parameters in the principle components space behave. On the y-axis the weights of the parameters in the eigenspace of the principle components are illustrated. The weights may help giving a look into the importance of the parameters, when doing classification tasks as in section 4 & 5. In particular, for the rise and decay time the weights seem to be for all principle components in one direction and for one of them it takes a large value, where in comparison reduced χ^2 is spread in both directions at large values. This may lead to the conclusion to focus on the rise and decay time, but in fact the pulse integral and decay time are the most promising parameters regarding the classification tasks in section 4 & 5.

4 Signal & background discrimination with machine learning

In this section machine learning algorithms are used to discriminate signal and background events. At first, the section 4.1 introduces the used algorithms theoretically, which are used to implement the classification tasks. The algorithms applied to the data are routines provided by scikit-learn [21]. The results of each classifier are presented and discussed in 4.2 & 4.3.

4.1 Theory of used algorithms

Machine learning (ML) is a widely used technique for the purpose of analyzing data. In this thesis it is applied to the best-fit parameters found in section 3 to make a distinction between signal and background pulses. This means that the ML algorithms used for the discrimination serve the purpose to do a classification task.
The underlying principle of the used machine learning algorithms is to train the classifier models with a certain proportion of the best-fit parameters. The used training data is labeled in a way that the algorithm knows which data belongs to which class of pulses. By that, the model learns the properties of the different classes, which are to be distinguished at the end. After the training, the classifier can use its experience for the evaluation of unknown data to make a distinction between the different classes. This sort of learning principle is called supervised learning. Typically, 80% of the data is used for training and 20% is used to evaluate and test the model after the learning procedure.

The scikit-learn module provides multiple classifier models. For the discrimination of signal and background in this thesis two of them are chosen: the Random Forest Classifier and the K-Nearest Neighbor Classifier.

- **Random Forest**
  Random forests are ensemble methods provided by scikit-learn [21]. It is based on decision trees. In particular, a random forest is built of a certain number of randomized decision trees which are uncorrelated [22].

  Decision trees are another method based on supervised learning. The algorithm can also be used for classification tasks. Every decision tree starts with a root node and is built with consecutive nodes. In each node, the data is split under a certain condition. This procedure repeats until the node is pure. A node is called pure, when all data in the node is from one single class. During the learning process of a decision tree it categorizes the parameters in a way, that the true labels of the samples are equal to each other [21]. Considering the light and background pulses it would mean that all light pulses and all background pulses are put into two different groups. From the training the decision tree is built and it consists of nodes where the decisions are made. In the leafs of a decision tree the outcome of the classification can be found. The depth of a tree can be chosen in advance. In each node the algorithm verifies if the input is lower or higher than a specific threshold value, that is learned by the model during the training process. By doing that, the test data can be evaluated. To make the clearest decision at a node, decision trees calculate the impurity [22], for which scikit-learn offers the Gini-impurity or the Shannon-information gain. In this work the Gini-impurity is used as a criteria for the classification tasks. Consider the number of outcomes K of the classifier, i.e. signal or background, which is in this case $K = 2$. The Gini-impurity is calculated with the following formula [22]:

  $$G = \sum_{k=1}^{K} p_{mk} \cdot (1 - p_{mk}) \tag{4}$$

  where m is the index for the node and p_{mk} describes the probability for each outcome. By minimizing the Gini-impurity it is made sure that the decision tree works in an efficient way.

Consider the training data to be a matrix of n lines and p columns. In this thesis, each column p describes one parameter obtained from the fit and the number of lines n contain the parameter values for the different pulses. The first step in the random forest algorithm is called bootstrapping. Bootstrapping is a technique where from all available lines in the training data the same amount of lines are picked randomly [22]. It is also possible that the algorithm picks the same line multiple times. By applying this technique a new sample of training data is generated. Since the underlying principle of random forests are decision trees, this new sample are given as an input to decision trees. Another randomly chosen feature is the use of parameters in the nodes of the decision tree. The process of bootstrapping and building new decision trees is repeated many times. In particular, the random forest is developed by repeating this procedure. After the training, the outcome of the test data is evaluated by the majority of votes in all leafs of the decision trees [22].

- **K-Nearest Neighbor**

  The K-Nearest Neighbour method is similar to the random forest and very common for classification. In contrast to random forests, it is not a model which is fit to data. In fact, the algorithm remembers the properties of the training data which are used on the test data. For that reason, the K-nearest neighbor algorithm is called memory-based [22].

  As the training data is labeled, it can be assumed that the data given into the K-nearest neighbor algorithm is described by N pairs $(x_1, y_1)...(x_N, y_N)$, where N is the total number of pulses [22]. In general, x contains the data and y is the label belonging to the data, as it is commonly used in supervised learning. In the context of this thesis, x is multi dimensional and can not be described by a single column vector. In K-nearest neighbor the label y_i of each x_i, where i describes the i-th entry of N-pairs is remembered by the algorithm. To do a classification the algorithm takes the new data x_0 and computes the distance to the neighbors, where the number of neighbors K can be chosen in advance. In this work the Euclidian distance is used as a distance measurement [22]:

$$d_i = \|x_i - x_0\| = \sqrt{(x_i - x_0)^2} \tag{5}$$

  In particular, this is done for all neighbors K. The distances d_i are calculated for the unlabeled test data x_0. The label of each distance d_i is the same as the label of the corresponding training sample x_i. For the classification the distances are sorted by length. The number of neighbors K decides how many of the closest neighbors are considered for the classification. The outcome for the test data is the label with the most votes based on the number K of closest neighbors.

4.2 Results of random forest

The data used in this algorithm contains the following parameters coming out of the fitting procedure: $t_0, \tau_{rise}, \tau_{decay}, A, \chi^2/ndof$ & I. First the natural logarithm was applied on all parameters. Since the constant c contains values that are negative, it was not considered as a parameter for random forests.

Since the results in a machine learning algorithm are desired to be as accurate as possible, it is necessary to do more preprocessing to optimize the training process and the outcome of the algorithm as well. Typically the data is split into training and test data, where the training data contains 80% of the data and the test is done with 20% of the data. In reality it is risky, since one training data set and one test data set could give a great accuracy in its combination due to overfitting of the parameters. In fact, the goal is to train a model, by which accurate results can be obtained with slightly different test data sets. For that, performing a cross validation is an appropriate approach. There are several techniques to perform a cross validation. In this work the k-fold cross validation technique is implemented. In fact, the k-fold cross validation method is enhanced to the so called stratified-k-fold cross validation. Stratifying data is necessary, since the amount of signal and background data is highly unequal. As already mentioned, the total data contains 2627 light pulses and 39580 background pulses. Without stratifying, it could be possible that the test data does not contain any signal data. In that scenario the algorithm would be missing the goal of the classification in general. The stratifying makes sure that the training and test data includes the same fraction of signal and background data. The k-fold technique describes the procedure of splitting the data into k equal pieces in which the training and test is performed. A basic illustration of stratified-k-fold can be seen in figure 12.

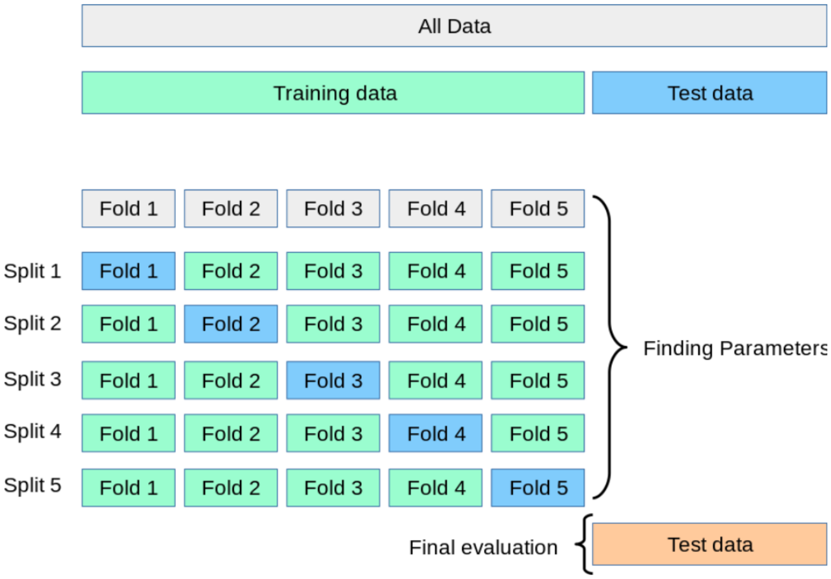

Figure 12: Procedure of K-fold cross validation. First the split of training and test data is done. The cross validation procedure splits the training data k times where in each split the training and test data are evaluated. It is common to determine the best hyper parameters from each split for the final evaluation. Figure from [21].

It is provided by the module scikit-learn and in this thesis the number of folds k equals 5. In practice, all data is split into 80% as training data and 20% as test data under stratifying. The training data is then split, where in the split 20% of the whole training data set is used as the test data. This is repeated 5 times under the condition that the 20% of test data is changing in each fold. It is suitable to perform a grid search to find the best hyper parameters for the final evaluation of the test data.

Before training and evaluating the model in the stratified-k-fold for each group of data, the data is standardized and PCA is applied on the data. The model is trained and in each fold the accuracy of the test data is calculated. The accuracy P is calculated by dividing the number of true prediction by the total number of predictions. The mean accuracy of the 5 folds is the following:

$$P_{acc-kfold} = 99.6413\% \pm 0.0967\%$$

In addition to the stratified-k-fold cross validation a grid search is implemented to find the best hyper parameters. In figure 12 this is denoted as "Finding parameters". The features used for the random forest are the number of trees, the maximum depth of each tree and the pruning parameter α. The α is a real-valued parameter that prunes the trees in a way that the cost-complexity function $R_\alpha(T)$ of a tree T is minimized [21] [22] (see [22] for deeper mathematical formulation). Basically, the pruning parameter cuts of leafs from trees in the random forest under the condition to minimize the cost-complexity and avoids overfitting in that way.
For the grid search, the used features are searched over a certain range that can be chosen in advance:

- Number of trees: $30 - 80$ trees with stepsize of 10 trees
- Maximum depth of trees: $5 - 15$ leafs with stepsize of 1 leaf
- Pruning parameter: $5 \cdot 10^{-5} - 9 \cdot 10^{-5}$ with stepsize 10^{-5}

After doing the grid search, the best hyper parameters found by the search are used for the final test. As a condition for the determination of the best hyper parameters, the accuracy is used. For the number of trees the best found value is 30 trees, the depth of the trees is 8 and the pruning parameter α is $5 \cdot 10^{-5}$.

The outcome of classifier tasks can be visualized with help of a so-called confusion matrix, as it can be seen in figure 13. The y-axis corresponds to the true label of the data where the x-axis is the predicted label of the test data set. As it can be seen, the majority of the test data was predicted correctly. The accuracy of the random forest algorithm used on the test data is:

$$P_{acc-test} = 99.8826\%$$

This speaks in favor of assuming that the model worked successfully. The discrimination of signal and background events by using the random forest algorithm provides highly precise predictions.

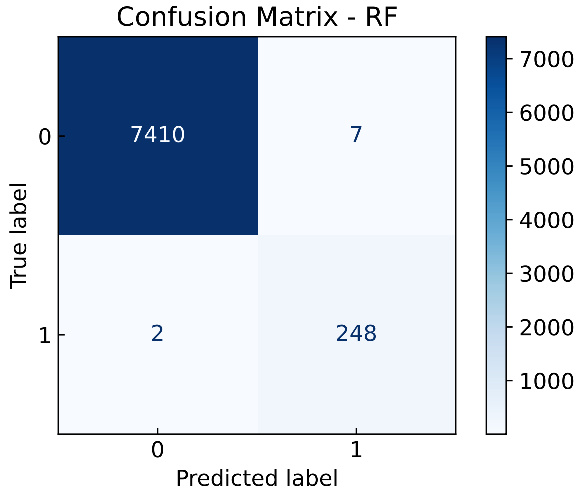

Figure 13: Confusion matrix of random forest, in which the classification with its labels is illustrated.

Especially, the increase of the accuracy compared to the mean accuracy shows that the grid search and the choice of the best parameters has a significant influence on the evaluation of the model.

From the entries in the confusion matrix, more properties can be calculated. Interesting is the False-Positive-Rate (FPR) to see how many wrong predictions are made of signal events. For that the number of false positives, in this case 7, is divided by the total length of time in day, which amounts 21 days. Since the amount of data in the test split is 20% of the total data, the 21 days are multiplied with 0.2. The result is:

$$\text{FPR} = 1.67\ d^{-1} = 1.85 \cdot 10^{-5}\text{Hz}$$

The true positive rate (TPR) or the precision of signal events can be calculated by dividing the correct predicted signal events by the total signal events in the test data. For this, the result is the following:

$$\text{TPR} = 97.25\%$$

In order to evaluate the false positive rate FPR, it is necessary to calculate the statistical significance. It can be calculated with [23]:

$$S = 2\left(\sqrt{\epsilon_d \epsilon_a n_s T + n_b T} - \sqrt{n_b T}\right) \tag{6}$$

The factors in equation 6 are explained in the following [23]:

- Detection efficiency $\epsilon_d = 0.5$
- Efficiency to classify signal events correctly $\epsilon_a = \text{TPR} = 0.9725$
- Signal rate $n_s = 2 \cdot 10^{-5}\text{Hz}$ [24]
- Observation time $T = 517.91\text{h}$
- Rate of mis-identified background events $n_b = \text{FPR} = 1.85 \cdot 10^{-5}\text{Hz}$

With the factors described above, the significance S can be computed:

$$S = 2.76$$

Since the expectation of axions converting to photons which can be detected then is at 1 conversion per day, the FPR seems to be rather high. This means that 1.67 wrong light pulses are predicted for each day. The acceptance threshold for the statistical significance is chosen to be $S \geq 5$. Therefore the false positive rate is to high. In order to obtain a significance of 5, the false positive rate must be about $\sim 10^{-6}$.

In figure 14 the test data with its prediction is shown. Since the PCA was applied on the data, the principal components span a six-dimensional space. Therefore the first two components are shown in the plot. It can be seen that the trained model is capable of discriminating between signal and background events since the scatter points of each group are close to each other.

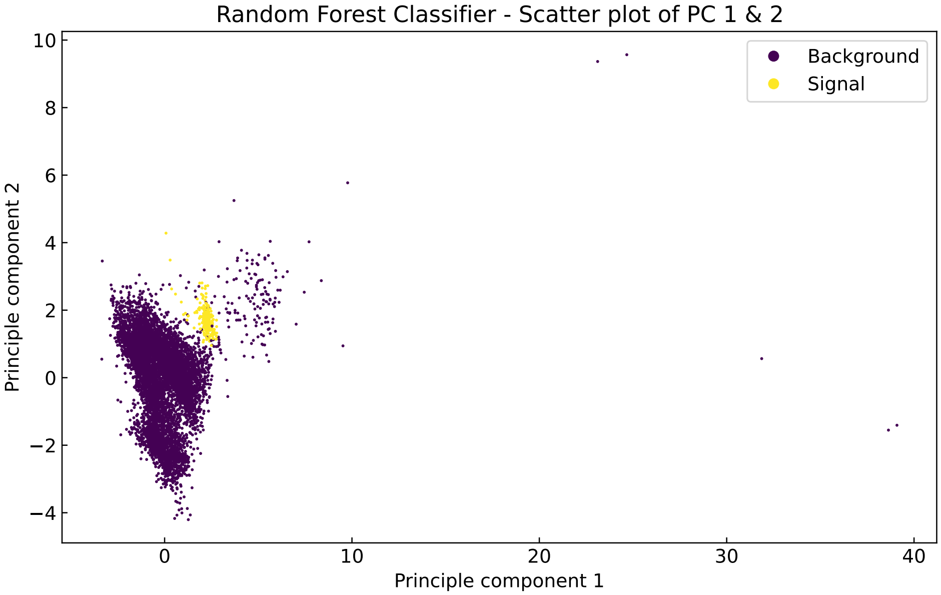

Figure 14: Scatter plot of the first two principle components in which the light and background events are classified, which are coming out of evaluating the test data after training the random forest model.

4.3 Results of K-nearest-neighbour

The procedure for the classification with the K-nearest-neighbor algorithm is the same as for random forests. First the logarithm is applied on all data. The stratified-k-fold is made and a grid search for the best hyper parameters is performed. Since the most important feature for K-nearest-neighbor algorithms are the number of neighbors K, the grid search was focused on finding the best accuracy for it. The range of the search of the best value for the number of neighbors is the following:

- Number of neighbors: 5-15 with step size of 1

The cross validation provides the accuracy for each fold again. The mean accuracy calculated from all folds is the following result:

$$P_{acc-kfold} = 99.5370\% \pm 0.1327\%$$

From the cross validation, the highest obtained accuracy is for $K = 6$. In figure 15 the confusion matrix is shown. The result speaks for an accurate prediction of signal and background events. The accuracy of the final test data amounts:

$$P_{acc-test} = 99.8174\%$$

Analogously, as it was done for random forests, the FPR and TPR can be calculated for this model as well:

$$\text{FPR} = 1.19\ d^{-1} = 1.38 \cdot 10^{-5}\text{Hz}$$

$$\text{TPR} = 97.97\%$$

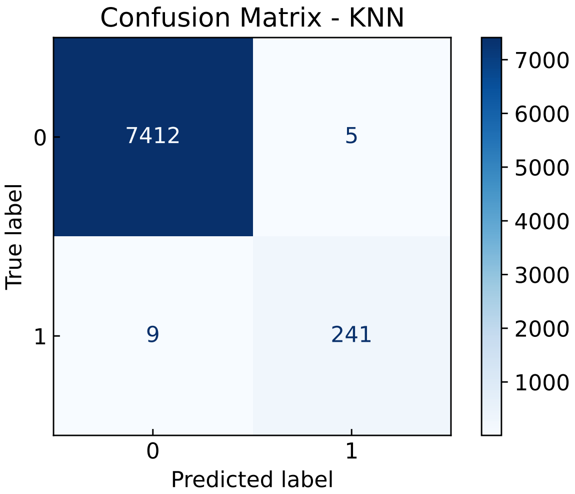

Figure 15: Confusion matrix for the K-nearest-neighbor model. The classification of the light and background data with its labels is illustrated.

The K-nearest neighbor algorithm provides a better result for the false positive rate, but in fact it is still rather high in comparison to the expectation of the axion-photon conversion which could be detected.

The significance for the K-nearest neighbor algorithm is:

$$S = 3.12$$

Due to the lower false positive rate the statistical significance is slightly higher compared to the significance obtained by the random forest algorithm. However, it is still lower than 5 and therefore the model should be rejected.

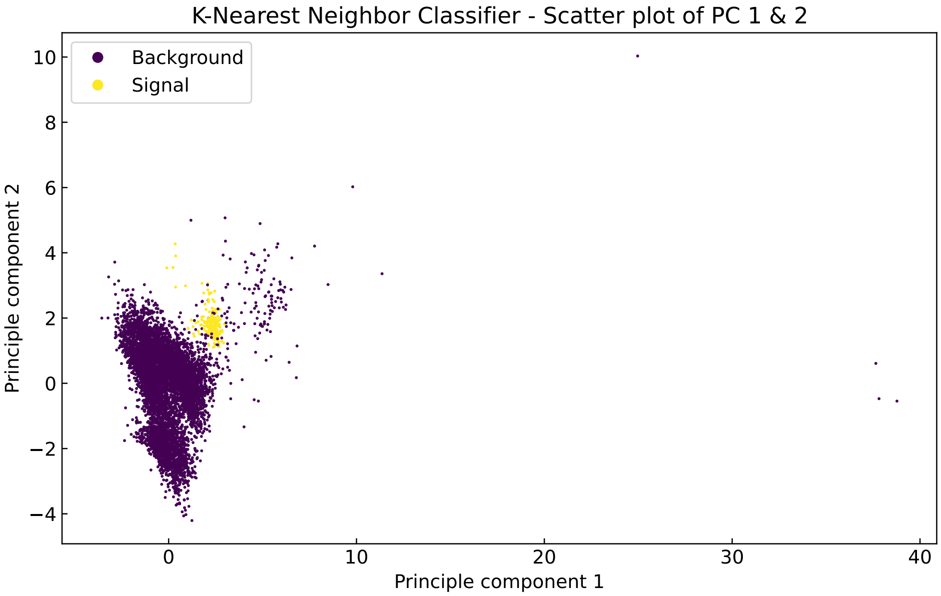

Figure 16: Scatter plot of the first two principle components in which the light and background events are classified, which are coming out of evaluating the test data after training the K-nearest-neighbor model.

In figure 16 the first two principle components are shown again. It can be seen again, that the data points in the plot are very close to each other and the K-nearest neighbor algorithm provides a possibility to discriminate the two different events. Since the plots of random forest and K-nearest neighbor are built up on the same way and the first two principle components are shown in each scatter plot, they look similar to each other. In both of the plots in figure 14 & 16 there are outliers of background data which have a higher value for the first principle component. The number of outliers in the total data could be higher, since the test data contains only 20% of the total data. That could be an indicator for different background events as it will be discussed in the next section.

5 Investigation of background

This section investigates the intrinsic background recorded with the TES. To analyze the background another machine learning algorithm is used. The theoretical background of the algorithm is explained in section 5.1. The section 5.2 presents the results obtained by using the algorithm on the parameters of the intrinsic background.

5.1 Theory of Gaussian Mixture Model

The Gaussian Mixture Model is a clustering algorithm. It is not a usual classification task where a model is trained under supervised learning. The clustering algorithm belongs to the unsupervised learning models, which means that the data given into the algorithm is not labeled with certain outcomes. The unsupervised algorithm is useful in this case, since it is not known in advance how many background sources contribute to the background events.

The base of the Gaussian Mixture Models is the multi-dimensional Gaussian distribution, which is described by the following [22]:

$$f(x) = \sum_{m=1}^{M} \alpha_m \cdot \Phi(x; \mu_m, \Sigma_m) \tag{7}$$

The function Φ describes the Gaussian density function depending on the data x, the multi-dimensional μ_m and the covariance matrix Σ_m. The multi-dimensional Gaussian density function can be written in the following way [25]:

$$\Phi(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{-1}} \cdot \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right) \tag{8}$$

Here, the mean has the dimension $\mu \in \mathbb{R}^n$ and the vector x, which contains the data, can be written as $x = (x_1, ..., x_n)$. The covariance matrix is defined in the following equation [25]:

$$\Sigma = \mathrm{E}[xx^T] - \mu\mu^T \tag{9}$$

where $\mathrm{E}[xx^T]$ is the expectation value of the product. In equation 7 the number M is the total number of clusters. The factor α_m is a weight factor, analogously to the probability and describes the mixing proportion of the different clusters [22]. Therefore, the mixing proportions for all clusters M sum up to 1.

The fitting procedure of the data to the Gaussian Mixture Model is implemented by the method of Maximum-Likelihood. In this method the Likelihood function

$$L(\theta) = \prod_{i=1}^{N} f(\theta, x_i) \tag{10}$$

is maximized, from where the parameter θ can be obtained. In fact, the maximization of the Likelihood function in Gaussian Mixture models is rather difficult [22]. In particular, using the logarithm on the Likelihood function provides the Log-Likelihood function which can be written as the following:

$$l(\theta) = \log L(\theta) = \sum_{i=1}^{N} \log f(\theta, x) \tag{11}$$

The summation over the logarithms for each cluster makes the differentiation and therefore the maximization complicated [22]. To solve this particular problem, the Gaussian Mixture Model uses an algorithm called the Expectation-Maximization algorithm (EM-algorithm) [22]. Basically, the EM-algorithm provides different random Gaussian distributions over the data and computes for each data point the probability to make a decision in which gaussian distribution the data point could belong and after this, the parameters are getting optimized to maximize the Likelihood function [21]. By doing that, the different clusters belonging to the different Gaussian distributions evolve. A mathematical explanation in detail can be found in [22].

5.2 Background analysis

In addition to the discrimination of signal and background pulses, it is also important to understand where the background originates from. As already explained in section 3.1, the background data in this thesis contains the intrinsic background which includes radioactive material in the detector itself or cosmic rays that are depositing energy to the TES. In order to interpret the background events better, the Gaussian Mixture Model (GMM) is applied to achieve a clustering of the events. In advance preprocessing is applied to the data. In particular all data points with a greater reduced χ^2 than 6 are not considered. The parameters used for the algorithm are the same as for the classification in section 4.

In figure 17 the decay time and pulse integral are illustrated after the fit of GMM to the parameters. Each cluster is represented by a color, and the plus sign marks the mean value of the clusters found by the algorithm. It can be seen that the data points can be split in sections by eye. This will be investigated in detail in the following.

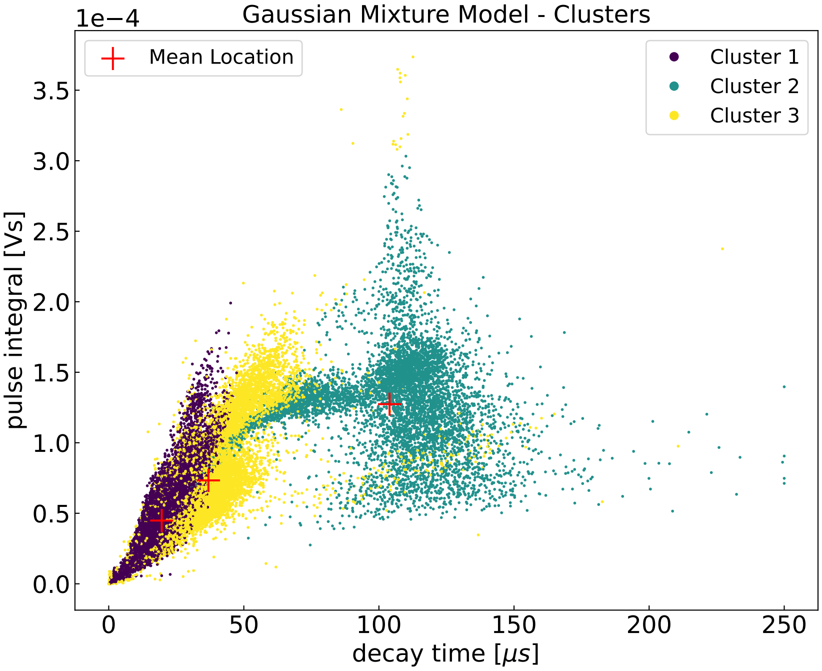

Figure 17: Clusters as a result of the Gaussian Mixture Model. As an example, the decay time and pulse integral is shown. The mean location of each Gaussian distribution is marked. The different colors represent the clusters.

The number of clusters chosen for the algorithm is 3. To get an accurate number of clusters the 'elbow-method' is a common used technique in clustering algorithms [21]. For that the Akaike information criterion (AIC) is used, which can be computed with the following formula:

$$\text{AIC} = -2 \cdot l(\theta) + 2 \cdot d \tag{12}$$

where $l(\theta)$ is the log-Likelihood function and d is the number of parameters used. The AIC is computed then for a range of numbers for the clusters and the elbow represents the point in which the sharp kink is visible. This can be seen in figure 18. Unfortunately, the fit of the data to GMM does not provide a proper kink, which makes it difficult to chose a certain number of clusters.

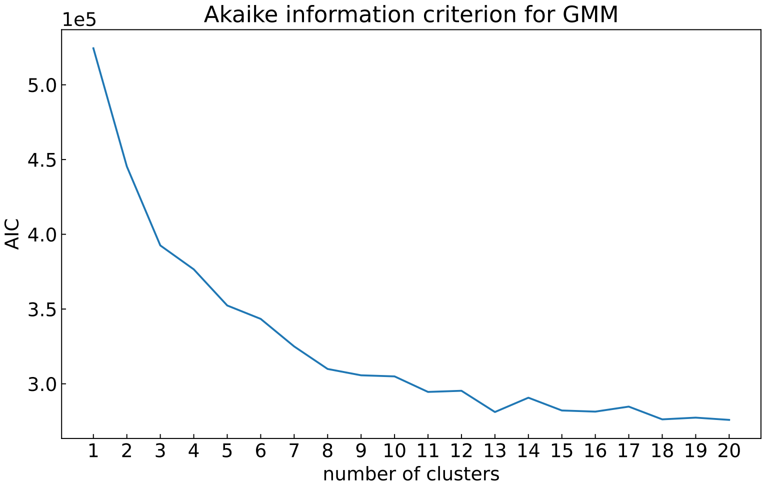

Figure 18: The values of the Akaike information criterion calculated for the number of clusters.

Since the Gaussian Mixture Model is based on the parameters of a Gaussian distribution as means and covariances, they can be observed in the histograms in figure 8. In particular the histogram of the decay time τ_{decay} is used to plot Gaussian distributions over the histogram. This can be seen in figure 19. The histograms and the Gaussian density function are normalized in a way, that the integral is 1.

The Gaussian distributions for 5 and 6 clusters seem to be arbitrary. By looking at the histograms for 3 and 4 clusters, they seem quite reasonably since the clusters can be seen in the histograms as well. But for 5 and 6 clusters it is becoming more difficult to interpret and identify the clusters by observing the distribution of the decay time.

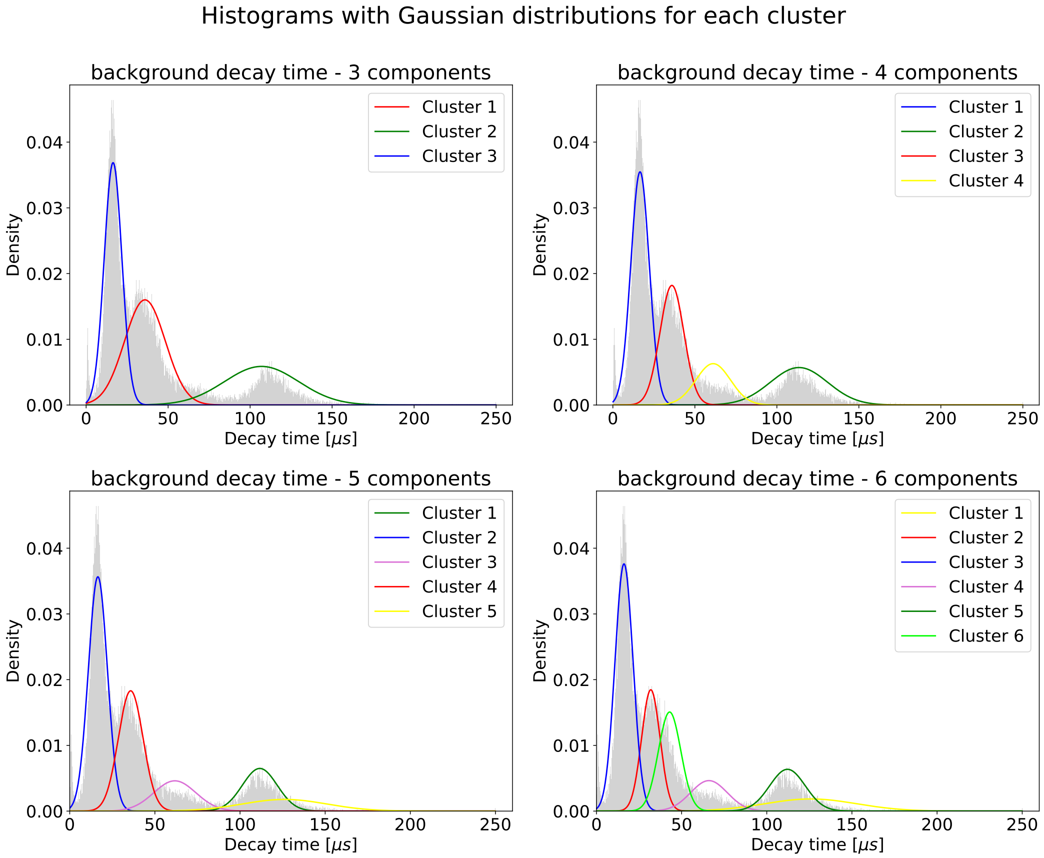

Figure 19: The density functions of the Gaussian distributions coming out of the Gaussian Mixture Model are plotted over the histograms of the decay time. By that, the different clusters can be identified.

To understand the clusters better, it is useful to take a look at the time dependence and the evolving process of the clusters. Since the AIC plot did not deliver a certain number of clusters for which the model is the most appropriate, different numbers of clusters are investigated. More precisely, the numbers of clusters from 4-8 are explored.

The evolving procedure starts with fitting 500 data points to the GMM. These 500 data points are getting clustered by the algorithm. This is repeated where in the next step $500 + 500$ data points are fitted. In total, this is done until all of the data is included. To interpret the evolving over the 21 days of observation the time is noted for each step of 500 data points. Some examples of the evolving process can be seen on the next pages in figure 20, 21 & 22. Unfortunately, the colors of the clusters do not match in the evolving procedure. The reason is, that the fit of the GMM labels the clusters automatically. Therefore the colors for all clusters change in each fitting process. Nevertheless, it can be seen in comparison of the figures 20, 21 & 22 that the big cluster in the middle at around 120μs appears in the middle of the observation time (see fig. 21). Figure 20 shows the state of the clusters at the beginning of the observations and figure 22 shows the final state.

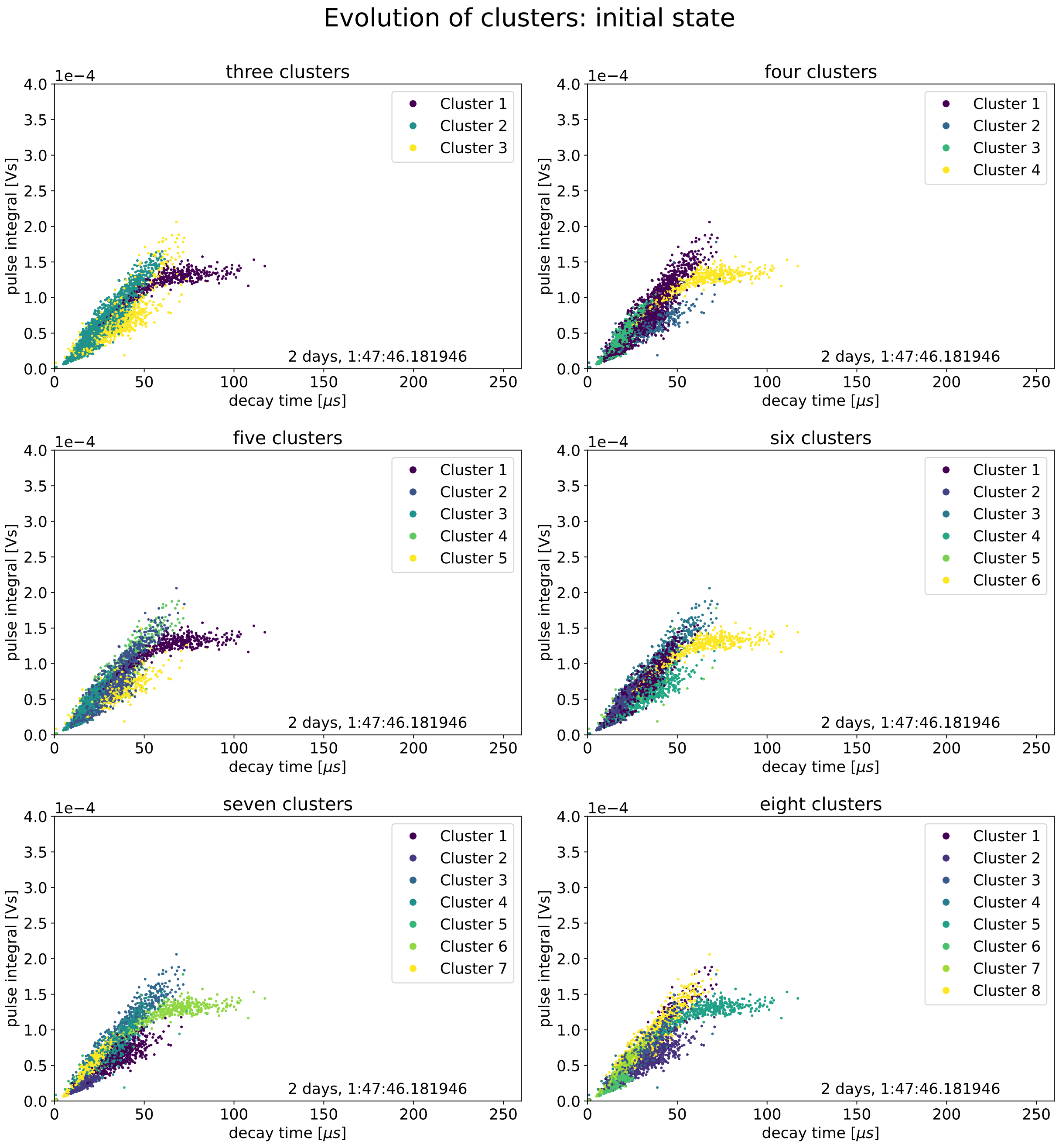

Figure 20: This figure shows the beginning state of the clusters after an observation time around 2 days.

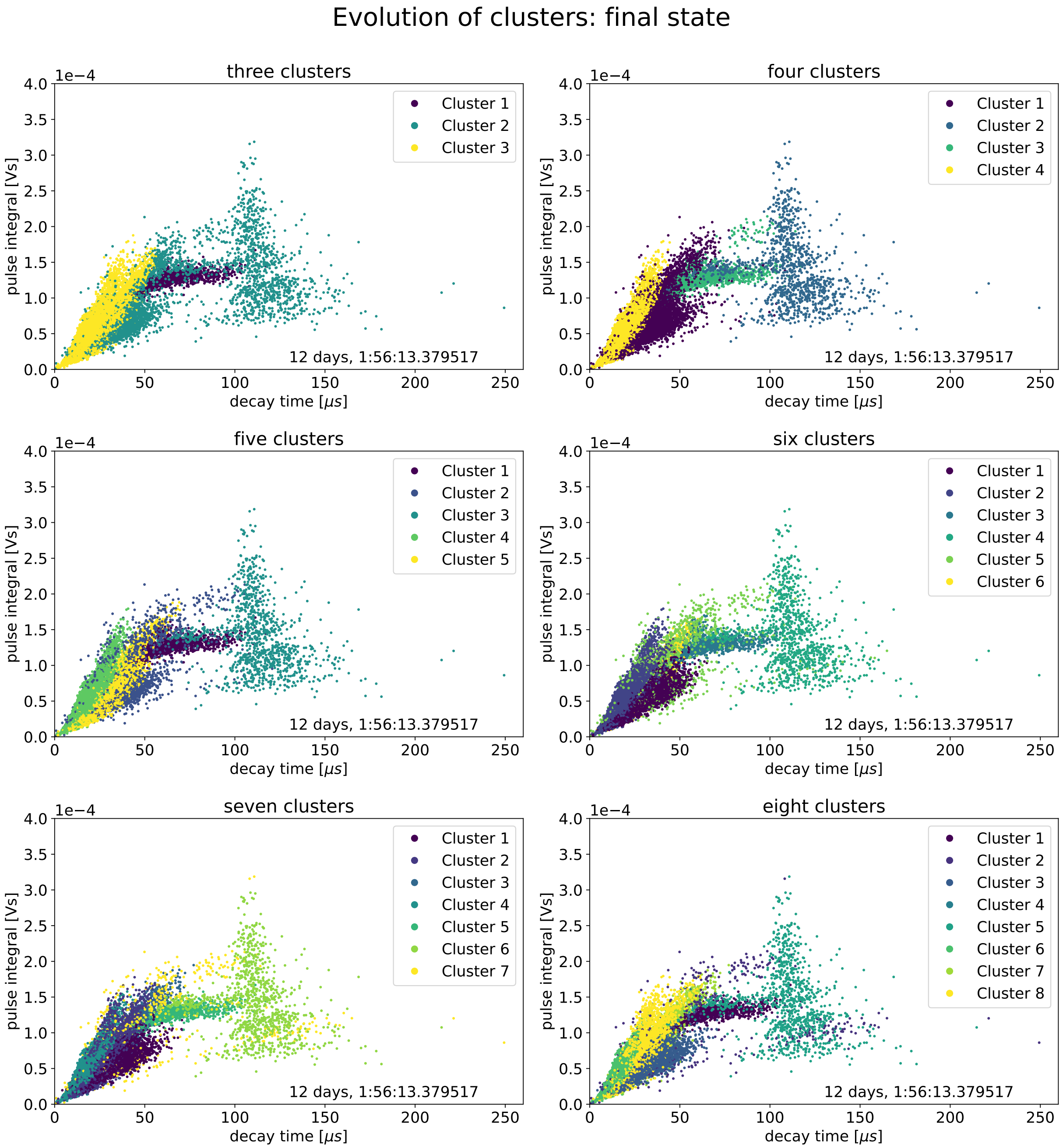

Figure 21: In this figure the appearance of the new cluster is illustrated. After an observation time of around 11 days the cluster in the middle can be seen. The plots above present the state after around 12 days.

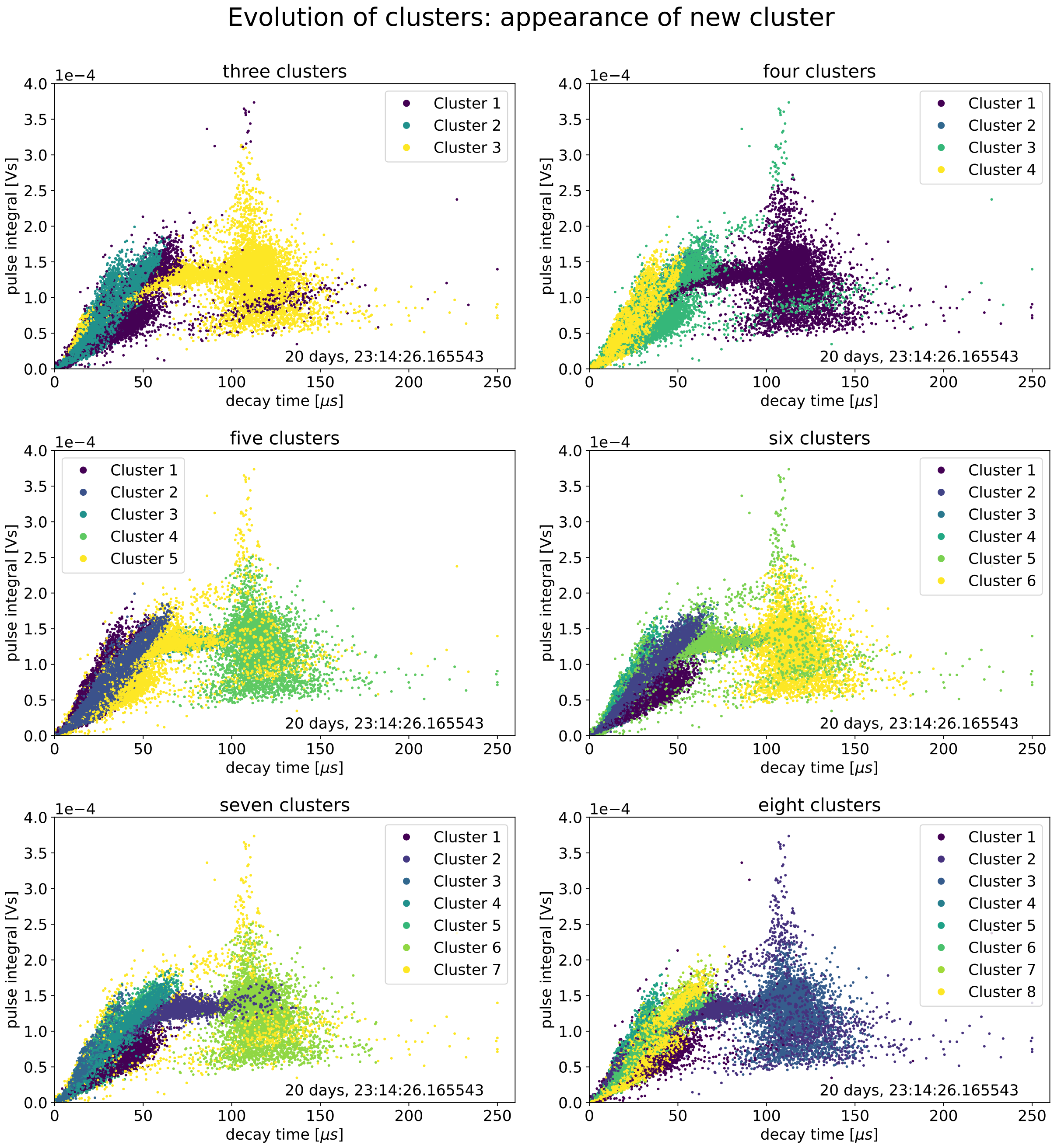

Figure 22: This figure shows the final state of the clusters at an observation time of around 21 days.

The beginning state can be seen in figure 20. The data collected over 2 days can be clustered in $3-4$ clusters. This amount of data to more clusters seems to be arbitrary, since it can also be described by a lower number of clusters properly. The event occurred at day 12 is shown in figure 21. It is difficult to make a statement about the exact origin of this event, but this will be discussed in the following. With the appearance of the new cluster, which can be seen in figure 21, the number of clusters needed has to increase to at least $5-6$, as it can be seen in figure 21. The final state of the clusters is shown in figure 22, where the shapes of the clusters do not differ much from figure 21. For that reason, the needed number of clusters should stay at $5-6$.

In order to understand the clusters better, the mean timelines are calculated. For this, each member of every cluster was identified and was traced back to its initial timeline, where the parameters are coming from. Then the mean was calculated from every entry in the timelines. The mean timelines are shown in figure 23

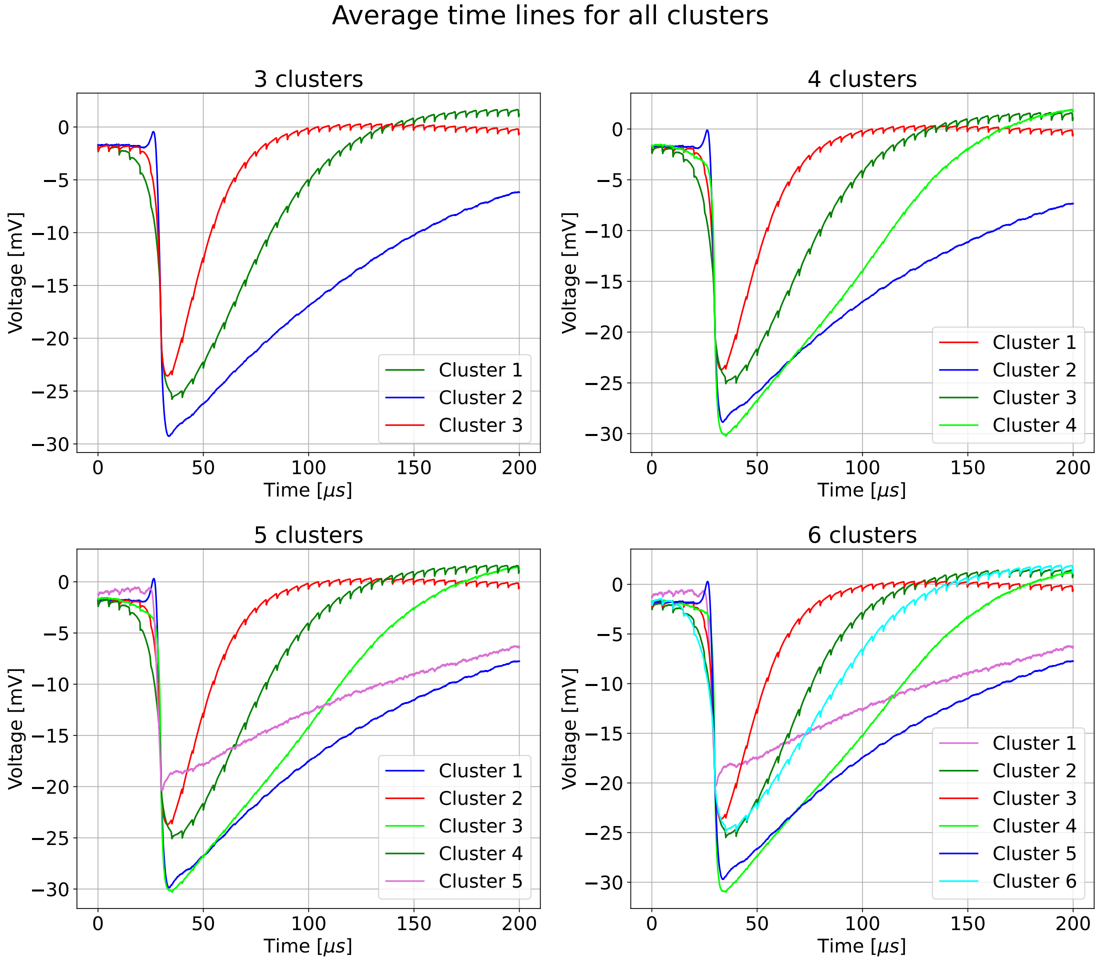

Figure 23: The mean timelines for each cluster are calculated and plotted, after fitting the data to the Gaussian Mixture Model with different numbers of clusters.

In the figure above it is recognizable that each mean timeline for the clusters has different properties. Since the data used for the clustering is only the background data, it is quite interesting that the a few of the mean timelines in the clusters look similar to light pulses. The typical background pulse that could be expected has a high decay time and could be compared with the blue mean timeline in figure 23. The red mean timeline in the figure above looks rather like a light pulse.

It could be assumed that this comes from fluorescence, emitted by the material around the TES. An approach to explain that in more detail could be to compute the energy deposition to the TES and compare it to the energy deposition from the used light source. The background energy deposition seems to be in the same magnitude as the used laser.
The observation of the cluster appearing after 11 days could lead to the idea that the mean timelines could differ as well. First, all background data has been split in two halves where the first half contains the background events until 11 days and the second half the last 10 days of events. The result can be seen in figure 24.

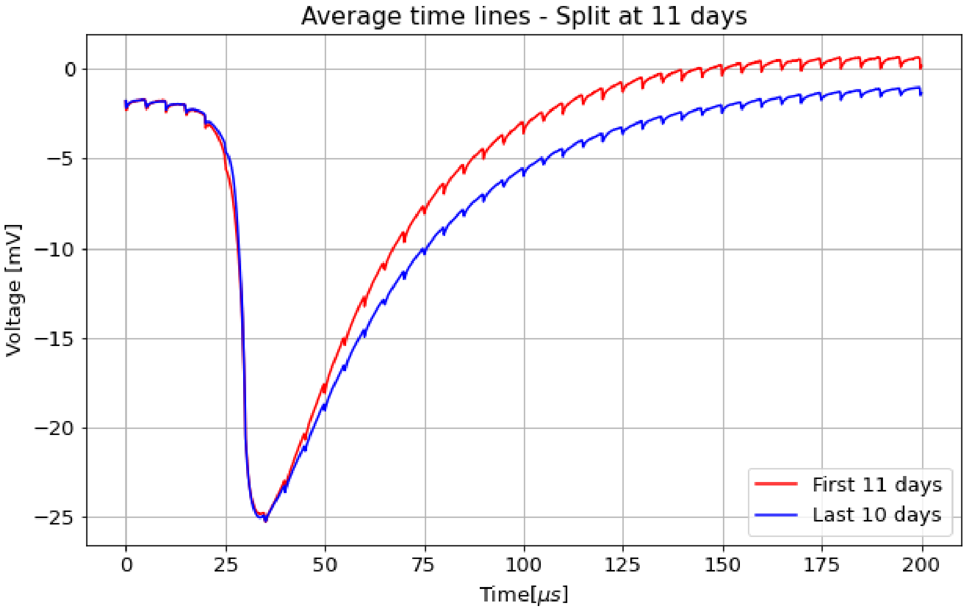

Figure 24: In this figure the mean timeline of all detected events in the first half of observation time and second half of observation time are shown.

It can be seen clearly that there is a difference between the two mean timelines in figure 24. In particular, the decay time differs. For the last 10 days of data the decay time is greater than for the first 11 days of observations and therefore the pulse integral is higher. However, regarding the cluster development it is not really useful. The shapes of the two mean pulses are to similar to each other to make a statement about the properties of the cluster evolving after 11 days of observation time. With the purpose of analyzing the evolving procedure the methods used for figure 23 and figure 24 are combined. Instead of looking at the mean timelines of each cluster as it is done in figure 23, the data is split again exactly as it was done for figure 24. This can be seen in figure 25 where the plots on the left show the mean timelines of each cluster until 11 days and the plots on the right after 11 days.

For 3 and 4 clusters coming from the GMM no significant difference in shape is observable. The only difference is the amplitude. It seems to become lower for the data after 11 days and by that the pulse integral decreases. By looking at 5 and 6 clusters it can be seen that some clusters are appearing after 11 days. More precisely, for 5 clusters the clusters number 4 and 5 appear after 11 days and for 6 clusters there are 2 new clusters which appear after 11 days as well: cluster 1 and 6. Due to that observation, one can say as a first statement that the number of clusters given to the algorithm should be more than 4. By that the analysis of the background events can be more accurate. As a second point, the fact that some clusters appear after 11 days which have a different shape leads to the assumption that there has been a significant change in the background. This could have happened by electronic noise at the SQUIDs. But it is also possible that the appearance of the cluster is human made. As an example one could think of electronic devices and its electromagnetic fields which affect the TES.

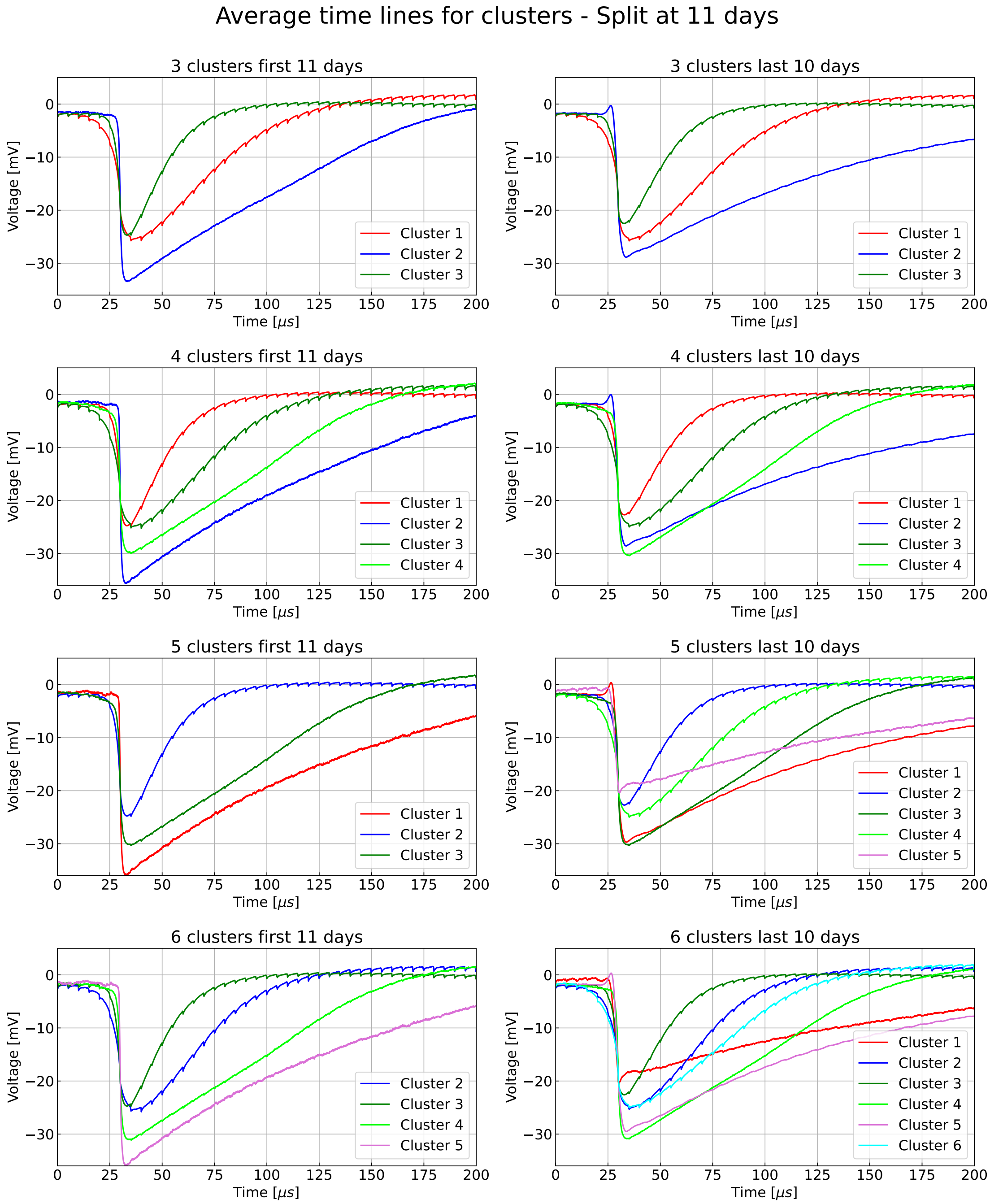

Figure 25: The mean timelines of each cluster achieved by the Gaussian Mixture Model are presented. The left shows the mean timelines for the clusters generated by the first 11 days of observation time and the right presents the clusters generated in the last 10 days of observation.

6 Conclusion and Outlook

In this thesis it can be seen that machine learning provides a solid perspective for the analysis of data. The classification of signal and background events was done at a rather high accuracy, but for the purpose of the ALPS II experiment it requires an improvement. The false positive rate was higher than the expected rate of axions. The statistical significance of both classifiers in this thesis is too low. However, by tuning the algorithms and optimizing the hyper parameters the desired significance level could be obtained. Furthermore, machine learning is a big field with many applications that can be implemented as methods for the analysis. In this work the used algorithms are quite basic. This can surely be expanded to Deep Learning and therefore neural networks, which are also used for classification. By that it could be possible to achieve even better results for the discrimination of signal and background events and also for the significance.

The Gaussian Mixture Model algorithm is an accurate method to analyse the clustering of events. It can deliver information, which can be used to analyse the background events in detail. With this information it is possible to prevent mistakes that could be caused by human interference. Further analysis of the background could provide insights in the environment of the detector and a characterisation of background events could maybe be in much more detail. An example for that is radioactive material in the detector which could be identified. In this work no energies of the background pulses have been calculated or considered, which could be done as a next step and give an insight of the magnitude in which the background events occur.

Furthermore in this work only the intrinsic background was considered in the analysis of the background events. To be more accurate with the analysis it is necessary to analyse the extrinsic background as well, since the extrinsic background includes the intrinsic background. This could help in order to understand the external influences of the laboratory.

References

[1] A. Ringwald. Axions and Axion-Like Particles. *arXiv e-prints*, page arXiv:1407.0546, July 2014.

[2] Katharina-Sophie Isleif and. The any light particle search experiment at DESY. *Moscow University Physics Bulletin*, 77(2):120–125, apr 2022.

[3] Dimitri Bourilkov. Machine and deep learning applications in particle physics. *International Journal of Modern Physics A*, 34(35):1930019, 2019.

[4] Pauline Gagnon. The standard model: A beautiful but flawd theory - https://www.quantumdiaries.org/2014/03/14/the-standard-model-a-beautiful-but-flawed-theory/ - last access: 16.11.2022.

[5] Anson Hook. Tasi lectures on the strong cp problem and axions, 2018.

[6] J.W. Cronin. The discovery of CP violation. EPJ H 36, 487–508 (2012) https://doi.org/10.1140/epjh/e2011-20057-4.

[7] Joerg Jaeckel and Andreas Ringwald. The low-energy frontier of particle physics. *Annual Review of Nuclear and Particle Science*, 60(1):405–437, 2010.

[8] P. G. Harris, C. A. Baker, K. Green, P. Iaydjiev, S. Ivanov, D. J. R. May, J. M. Pendlebury, D. Shiers, K. F. Smith, M. van der Grinten, and P. Geltenbort. New experimental limit on the electric dipole moment of the neutron. *Phys. Rev. Lett.*, 82:904–907, Feb 1999.

[9] Roberto D. Peccei. The strong CP problem and axions. In *Lecture Notes in Physics*, pages 3–17. Springer Berlin Heidelberg, 2008.

[10] Clemens Albrecht, Serena Barbanotti, Heiko Hintz, Kai Jensch, Ronald Klos, Wolfgang Maschmann, Olaf Sawlanski, Matthias Stolper, and Dieter Trines. Straightening of Superconducting HERA Dipoles for the Any-Light-Particle-Search Experiment ALPS II, 2020.

[11] Rikhav Shah, Katharina-Sophie Isleif, Friederike Januschek, Axel Lindner, and Matthias Schott. Characterising a Single-Photon Detector for ALPS II. *Journal of Low Temperature Physics*, pages 1–8, 04 2022.

[12] W. Demtröder. *Experimentalphysik 3: Atome, Moleküle und Festkörper*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2010.

[13] Wolfgang Hansen. Lecture notes on solid state physics, July 2021.

[14] Jan Dreyling-Eschweiler. "A superconducting microcalorimeter for low-flux detection of near-infrared single photons" - PhD Thesis. 2014.

[15] Noemie Alice Chloe Bastidon. *The cryogenic photon detection system for the ALPS II experiment: characterization, optimization and background rejection.* PhD thesis, Hamburg U., 1 2017.

[16] Dietmar Drung. High-tc and low-tc dc squid electronics. *Superconductor Science and Technology*, 16(12):1320, oct 2003.

[17] Hans Dembinski and Piti Ongmongkolkul et al. scikit-hep/iminuit - https://doi.org/10.5281/zenodo.3949207. Dec 2020, last seen: 20.11.2022.

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

[19] Wikipedia contributors. Covariance matrix — Wikipedia, the free encyclopedia, 2022. [Online; accessed 9-December-2022].

[20] Jonathon Shlens. A tutorial on principal component analysis. *CoRR*, abs/1404.1100, 2014.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[23] Manuel Meyer. A first application of machine and deep learning for background rejection in the ALPS II TES detector, poster presentation at the 17th Patras Workshop on Axions, WIMPs and WISPs, 8-12 August, 2022, Mainz, Germany.

[24] Rikhav Shah. ”a TES Detector for ALPS II - Characterising a cryogenic, low-background, low energy single photon detector” - PhD Thesis. 2022.

[25] Chuong B Do. The multivariate gaussian distribution. *Section Notes, Lecture on Machine Learning, CS*, 229, 2008.