Supressing the Muon Background in the ORCA10 Detector

Bosman, Jesse

Supressing the Muon Background in the ORCA10 Detector

THESIS

submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
PHYSICS

Author :	J. Bosman
Student ID :	s2025396
Supervisor :	Dr. Dorothea Samtleben
Second corrector :	Prof. Koenraad Schalm

Leiden, The Netherlands, October 11, 2024

Supressing the Muon Background in the ORCA10 Detector

J. Bosman

Huygens-Kamerlingh Onnes Laboratory, Leiden University
P.O. Box 9500, 2300 RA Leiden, The Netherlands

October 11, 2024

Abstract

This thesis focuses on the task of separating detector events caused by atmospheric neutrinos from those caused by atmospheric muons. Performance on this task is analysed using simulated data of these events as they are detected in the KM3NeT/ORCA10 detector setup. We present a new procedure for training the Machine Learning (ML) classifiers that handle this separation task. This most notably includes separating the data into track- and shower- like events, and training separate classifiers on these subsets of data. We show a significant improvement in the resulting neutrino signal when compared to the current classification procedure.

Chapter 1

Introduction

Neutrino research plays a crucial role in advancing our understanding of the universe, particularly in areas such as particle physics, astrophysics, and cosmology. Neutrinos, nearly massless subatomic particles, interact weakly with matter. This makes them notoriously difficult to detect. Yet studying their properties, such as neutrino oscillations, is essential for answering fundamental questions about the nature of matter and the structure of the Standard Model. One of the key questions in neutrino physics is the mass ordering of neutrinos (NMO), a topic that the KM3NeT's Oscillation Research with Cosmics in the Abyss (ORCA) detector aims to address.

The ORCA detector, located at the bottom of the Mediterranean Sea, is designed to measure atmospheric neutrinos. These neutrinos are produced when cosmic rays collide with the Earth's atmosphere, and their oscillations provide insights into the neutrino mass hierarchy. However, a significant challenge in such measurements comes from the background signals generated by atmospheric muons, which are also produced in cosmic ray interactions. These muons mimic the signatures of neutrinos, making it difficult to distinguish between them. Because the frequency of muon events is several orders of magnitude higher than that of neutrino events, getting a strong and clean neutrino signal is non-trivial.

This thesis therefore focuses on the suppression of the muon background in ORCA10, a phase of the ORCA detector comprising 10 of the foreseen 115 Detection Units (DUs). The task of separating muon events from neutrino events is critical for the success of the ORCA experiment, as the presence of muons introduces significant noise into the neutrino signal on which analysis of the oscillation properties is performed.

The core methods for reconstructing particle interactions in the detector are well-established. This work contributes by refining the data se-

lection procedure and machine learning techniques used, and thereby enhancing the strength and purity of the neutrino signal. The analysis presented here uses a new procedure where the detector data is split into two subsets with different properties. This allows the classification models to improve their performance on these subsets. This improvement is essential for the precise measurement of neutrino occurrences and, ultimately, for determining the neutrino mass hierarchy and refining the oscillation parameters.

In chapter 2 we will first explore the physics of neutrinos, particularly atmospheric neutrinos, and their interactions. Chapter 3 will provide a more detailed description of the ORCA detector and its operational principles. Chapter 4 will cover the machine learning techniques used in this work to improve the separation of muon events from neutrino events. Chapter 5 includes the data processing and analysis steps used for this work. Chapter 6 will explain the experimental methods employed to gain our final results. In chapter 7 the results of these methods will be shown. Then the results will be discussed and a conclusion on this work will be provided in chapter 8. Lastly, chapter 9 finishes with an outlook on future research possibilities.

Chapter 2

Neutrino Physics

Neutrinos are extremely light particles that are difficult to detect because they only weakly interact with matter. Neutrinos are often separated into groups based on their production process and location. Examples are solar neutrinos and neutrinos from nuclear decays. The neutrinos of interest to this work, and in the appropriate energy range for the KM3NeT detectors, are separable into two groups. The first group are cosmic neutrinos, produced in extra-galactic sources. The second group are atmospheric neutrinos, neutrinos formed when cosmic rays hit the earth's atmosphere. Atmospheric neutrinos are the target of neutrino oscillation research, and therefore by extension also of this work. This chapter will provide an introduction to neutrinos, covering the basics and properties of interest in the context of this project. First we shall briefly go over the origin and relevance of neutrino research in physics. Then atmospheric neutrinos will be discussed in more depth, as they are the neutrinos targeted by ORCA. The chapter closes with a section on the oscillation properties of neutrinos.

2.1 Relevance

Exploration of neutrino properties has already produced significant scientific progress. The experimental evidence for neutrino oscillations solved an open problem in the detection ratio between two different neutrino flavours, muon-neutrinos (ν_μ) and electron-neutrinos (ν_e) in the atmospheric flux [25]. It also explained the deficit of ν_e in the solar neutrino flux [11, 20]. For their work on neutrino physics, Takaaki Kajita and Arthur B. McDonald were rewarded the 2015 Nobel Prize in physics. The evidence for neutrino oscillations indicated that neutrinos have non-zero mass (see sec.

2.5). This showed the incompleteness of the Standard Model, which postulated neutrinos to have zero mass. Therefore theoretical extensions of the Standard Model such as Seesaw mechanisms [31] were proposed. The existence of *sterile neutrinos* such as postulated by the Seesaw mechanism are a possible candidate for dark matter [12], the nature of which is one of the greatest open questions in (astro)physics. Currently, the main question surrounding neutrino oscillations, and the question KM3NeT/ORCA aims to solve, is that of the neutrino mass ordering (see sec. 2.5). Determining the correct mass ordering would narrow down which mechanisms (such as Seesaw mechanisms) are responsible for neutrinos' mass. This additionally has impact on the viability of *leptogenesis* [19][23] as an explanation of the matter/anti-matter asymmetry in the universe.

Detectors with capabilities to detect neutrinos reliably and with high accuracy in direction and energy, enable researchers to look very deep into the universe, and far into the past. Neutrinos' small interaction cross-section make them the ideal particle for such purposes, as they pass mostly unhindered through gas and other matter in their path. And because they are not deflected by magnetic fields, they point back straight to their origin.

2.2 Standard Model

Neutrinos are part of a theory called the Standard Model. The Standard Model describes three fundamental forces of nature: the weakforce, the strongforce, and the electromagnetic force. It also describes the elementary particles that exist in the Universe. Neutrinos fall under a subcategory of elementary particles called leptons. Leptons are particles that have spin $\frac{1}{2}$, and only interact through the electromagnetic- and weakforce. There are three generations of leptons, giving rise to three types of neutrinos. Each generation consists of two leptons. One of them carries charge, such as the electron e, and one of them is neutral, such as ν_e. These two together (with their respective anti-particles) constitute the 1st generation of leptons. The 2nd and 3rd generations with the muon-neutrino ν_μ and tau-neutrino ν_τ respectively, are structured in the same way. These three different types of neutrinos are referred to as the different neutrino *flavours*.

The Standard Model does not cover the effects of gravity. Neither does it cover the quantum effects that produce the neutrino oscillations covered later on in this chapter. It is therefore known that despite its large number of successful predictions, such as the existence of the W, Z and Higgs Bosons, the Standard Model provides an incomplete description of

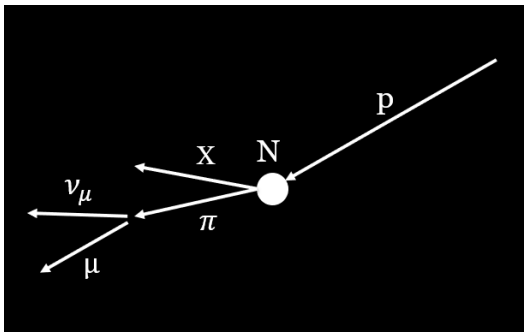


Figure 2.1: *Schematic representation of a possible production channel for atmospheric muon-neutrinos (ν_μ) and muons (μ). Here a cosmic ray consisting of a proton (P) collides with the nucleus (N) of an atom in earth's atmosphere. This collision produces a pion (π), and other particles of lesser interest here denoted with X. The resulting pion is unstable, and decays into a muon and a muon-neutrino.*

elementary particles.

2.3 Atmospheric Neutrinos

2.3.1 Production

The formation of atmospheric neutrinos is caused by cosmic rays, which consist mostly of protons, colliding with nuclei in the earth's atmosphere. These collisions produce π and in lesser quantity K mesons [29]. Mesons are another group of sub-atomic particles. These mesons are unstable and decay into neutrinos and other particles, some of which subsequently decay into neutrinos. A possible and common production channel is shown in figure 2.1.

2.3.2 Flux

To study neutrino oscillations, we naturally need values for the expected neutrino flux for each neutrino flavour. These flux values depend on many variables. The strength of the earth's magnetic field as a factor of detection site longitude, air temperature and density, and the angle of the incoming neutrinos are a couple of examples. Increasing the accuracy of the flux values is an ongoing effort. This work uses the values as found by Honda et al. [26] for the Frejus site.

2.4 Interaction Cross-section

Because neutrinos interact only through the weakforce, they have an extremely small *interaction cross-section*. This interaction cross-section is a quantifier for the probability of particles to interact with the matter they pass through. It depends on the energy of the neutrino, and of the type of interaction of that specific neutrino. The two different interaction groups are Neutral Current and Charged Current, enacted through Z and W bosons respectively [36].

2.5 Oscillations

2.5.1 Discovery

Oscillation between the different neutrino flavours was first theorized to occur in analogy to the oscillation of other particles, called Kaons. While those oscillations were well established, it took some time before experimental indications of neutrino oscillations were found. Among the first of these indications were the deficit of ν_e in the solar neutrino flux measured in the Homestake experiment, and the discrepancy between observed and expected 'muon-like' detection signals in the Kamiokande detector [20][25]. It was not until 10 years later that neutrino oscillations was concluded to be the cause for this deficit, based on new data with higher statistics from the Super-Kamiokande detector [22][28].

2.5.2 Flavour Mass

The three different neutrino flavours can be described as superpositions of three mass eigenstates of neutrinos (see fig. 2.2). This is what causes the oscillations between the different flavours as a function of time, and it can be described as the evolution between quantum eigenstates of the neutrino. It follows that the three mass eigenstate neutrinos can also be written as superpositions of the neutrino flavours, as this constitutes merely a change of basis.

$$\nu_\alpha = \sum_{i=1,2,3} U^*_{\alpha i} \nu_i \tag{2.1}$$

$$\nu_i = \sum_{\alpha=e,\mu,\tau} U_{\alpha i} \nu_\alpha \tag{2.2}$$

Where ν_i indicates the i-th neutrino mass eigenstate and $U_{\alpha i}$ is the mixing coefficient between the different states. Here * indicates the complex

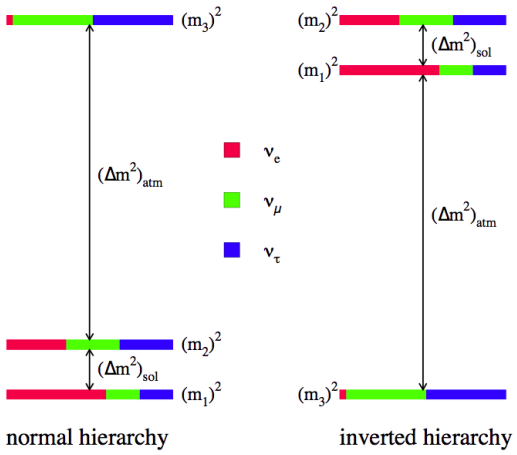


Figure 2.2: *A depiction of how the neutrino mass eigenstates are composites of the neutrino flavours. Figure is taken from Cahn et al. (2013) [13]. Here the term hierarchy is used in stead of ordering.*

conjugate of the coefficient. The ordering of the neutrino eigenmasses by increasing mass, the *Neutrino Mass Ordering (NMO)* is still and open question. There are two options. Normal Ordering (NO): $m_1 < m_2 < m_3$, and Inverse Ordering (IO): $m_3 < m_1 < m_2$. Solving this question is one of the main goals of the ORCA detector.

2.5.3 2-Flavour Approximation

To better understand neutrino oscillations, this section shall work out an example under the assumption that there are only 2 flavours of neutrinos. This simplifies the process and the resulting equations. The 3-flavour case is a natural extension of this example. We shall closely follow the example given by Nauta (2022) [35].

Assume the existence of two flavour states ν_e and ν_μ, and two mass eigenstates ν_1 and ν_2. The unitary matrix U describing the *mixing* between these states is then the 2d rotation matrix.

$$\begin{pmatrix}\nu_e\\ \nu_\mu\end{pmatrix} = \begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}\nu_1\\ \nu_2\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}\nu_1\\ \nu_2\end{pmatrix} = \begin{pmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}\nu_e\\ \nu_\mu\end{pmatrix} \tag{2.3}$$

The solution for the time-dependant evolution of ν_μ is then given by equation 2.4.

$$|\nu_\mu(t)\rangle = -\sin\theta e^{-iE_1 t}|\nu_1\rangle + \cos\theta e^{-iE_2 t}|\nu_2\rangle \tag{2.4}$$

Using equation 2.3 to substitute out the mass eigenstates we can then write

the probability of ν_μ oscillating to ν_e as shown in equation 2.5.

$$\begin{aligned} P(\nu_\mu \to \nu_e)(t) = |\langle \nu_e | \nu_\mu \rangle|^2 = \\ |\cos\theta \sin\theta (e^{-iE_2 t} - e^{-iE_1 t})|^2 = \sin^2(2\theta) \sin^2(\frac{E_2 - E_1}{2} t) \end{aligned} \tag{2.5}$$

As neutrinos have extremely low mass, atmospheric neutrinos are always in the ultra-relativistic regime. Because of this, we can use $x \approx t$ and $m \ll E$ to rewrite equation 2.5.

$$P(\nu_\mu \to \nu_e) = \sin^2(2\theta) \sin^2(\frac{\Delta m^2_{21} L}{4E}) \tag{2.6}$$

Now it becomes clear how the mass squared difference between neutrino flavours and the mixing angle θ influence the oscillation between two neutrino flavours.

2.5.4 3-Flavours

When extending to the full 3-flavour case, the unitary matrix now describing the mixing between the flavour and mass states is called the Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix [34]. This is often shown in its decomposed version as shown in equation 2.7.

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13} e^{-i\delta_{CP}} \\ 0 & 1 & 0 \\ -s_{13} e^{-i\delta_{CP}} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

For legibility, here $s_{ij} = \sin\theta_{ij}$ and $c_{ij} = \cos\theta_{ij}$. δ_{CP} is called the CP-violating phase, which when non-zero creates a difference between neutrino- and anti-neutrino interactions. Increasing the accuracy with which these parameters and the mass squared differences are known is the goal of neutrino oscillation research

2.5.5 Matter Effects

Up and until now we have treated the oscillations as if they occur in vacuum. However for oscillation research the main interest is in *upgoing* neutrinos, neutrinos that have passed through the earth on their way to the detector. This has multiple reasons. *Downgoing* atmospheric neutrinos that reach the detector on their way down from the atmosphere have not travelled distances long enough to produce significant oscillations. Also

Version of October 11, 2024– Created October 11, 2024 - 09:35

important are *matter effects*. These effects describe how the potential felt by neutrinos as they move through matter changes the effective mass and effective mixing angles, thus altering the oscillation probabilities. The potential effects all neutrinos with NC interactions the same, and thus its effects are not measurable. However the potential produces measurable effects through CC interactions. Combining measurements of the matter effect with high precision measurements of oscillation parameters is how the full ORCA detector will aim to solve the NMO problem [36].

Chapter 3

Detector

3.1 KM3NeT

KM3NeT is an international collaboration of research institutes and universities that is implementing deep-sea neutrino telescopes/detectors in the Mediterranean sea. The full KM3NeT architecture consists of two seperate detectors: Astroparticle Research with Cosmics in the Abyss (ARCA) and Oscillation Research with Cosmics in the Abyss (ORCA). ARCA will be used to study the energy spectrum, energy and flavour composition of neutrinos from (extra) galactic sources [5]. ORCA focusses on the oscillation of atmospheric neutrinos. A key difference between ARCA and ORCA is the energy range they are optimized to detect. For ORCA this is in the range 0-100GeV, while ARCA is optimized for energies in the TeV range. This is also what makes ORCA different from other neutrino detectors such as IceCube [24]. The goal of ORCA is to refine oscillation parameter values and determine the correct NMO, by analysing the fluxes of *upgoing* neutrinos [6].

3.2 ORCA

The ORCA detector is situated at 42° 48′ N 06° 02′ E at a depth of 2450 m, about 40 km offshore from Toulon, France [7]. It is placed at this depth to shield it from noise. Fully finished, it will consist of 115 Detection Units (DUs). A DU is a string consisting of 18 Digital Optical Modules (DOMs) [17]. For reference, see figure 3.1. The DOMs detect photons from neutrino collisions and the resulting particles (see sec. 3.3). This is done through the 31 photo-multiplier tubes (PMTs) housed in each DOM [7].

Figure 3.1: *A visual representation of a DU (left) and a DOM (right). This image was taken from the letter of intent for KM3NeT2.0 [7].*

The vertical spacing between each DOM is about 9m starting about 40m from the seabed, and the horizontal spacing between each DU is about 20m [7]. Deployment of the detector happens in phases, with multiple DU deployment runs per phase. At the moment of writing this work, 23 DUs have been deployed. However, this work will focus on the detector setup with 10 DUs dubbed ORCA10. This is because the latest iteration of data processing is available for ORCA10. Therefore, improvements made on ORCA10 simulation data may already lead to improvements to the oscillation analyses being done with ORCA10. Potential improvements the classification process for ORCA10 can also be transferred to newer data.

3.3 Detection Principles

This section will discuss the radiation processes that enable the detection of our events of interest and give rise to the different detection signatures. We shall also brief discuss the data taking procedure for the detector.

3.3.1 Bremsstrahlung

Bremstrahlung is radiation caused by charged particles as they are deflected by the charged particles in the medium they are travelling through [36]. It is the (de)acceleration caused by this deflection that produces the radiation.

3.3.2 Cherenkov Light

The source of light which enables event detection is a phenomenon called Cherenkov radiation. Cherenkov radiation occurs when a charged particle moves through a medium faster than a wavefront of light could [27]. The change in electromagnetic field caused by the passing of this particle causes excitations in the medium, which in turn causes photons to be released. Due to the particle's speed, the wavefronts of light interfere to produce what is called the *Cherenkov cone*. This cone of light moves away from the particle's track under the characteristic *Cherenkov angle* θ_c. The value of this angle is determined by the refractive index n of the medium and the speed of the particle. For all events of interest for oscillation analysis, Cherenkov light is the dominant source of hits in the detector [35].

3.3.3 Data Taking

The DOMs and PMTs are constantly *online*, i.e. continuously recording data. However, due to the significant presence of background noise, discussed further in section 3.5, the number of hits registered by the PMTs is extremely high. Transferring, storing, and analyzing this vast amount of data would be an overwhelming task. To address this, filters are applied directly at the detector site. These filters ensure that only hits occurring within a small time window, and with a sufficient number of hits across different DOMs that can be causally connected within that window, are transmitted for further analysis.

3.4 Signatures

This section will explain the two distinct event signatures observed in the detector. Before going into the processes behind these signatures, we shall briefly discuss a characteristic important for all events relevant for oscillation analysis.

3.4.1 Zenith

An important property for particles as they arrive at detectors is the zenith angle θ_z. The zenith angle is defined as the angle between the direction of the neutrino and the line straight up from the detector. Using ORCA convention, upgoing events have $\cos\theta_z < 0$. Because on the scale of neutrino oscillation lengths neutrino production can be approximated to be at one average height, θ_z parameterizes the length the neutrino has travelled before reaching the detector. This is naturally important for oscillation analysis. Selecting only events reconstructed as upgoing is also a crucial part of filtering out the muon background noise that will be further discussed in section 3.5.

3.4.2 Shower

Shower events are caused by all NC, and CC $\nu_e(\overline{\nu}_e)$ and $\nu_\tau(\overline{\nu}_\tau)$ (with a branching rate of 83%) neutrino interactions. In these events, a high fraction of the neutrino's energy goes into the production of lower energy particles. These particles may in turn decay into more particles of even lower energy. This process continues until the leftover energy is insufficient for particle creation. Each of the charged particles created in this shower, if

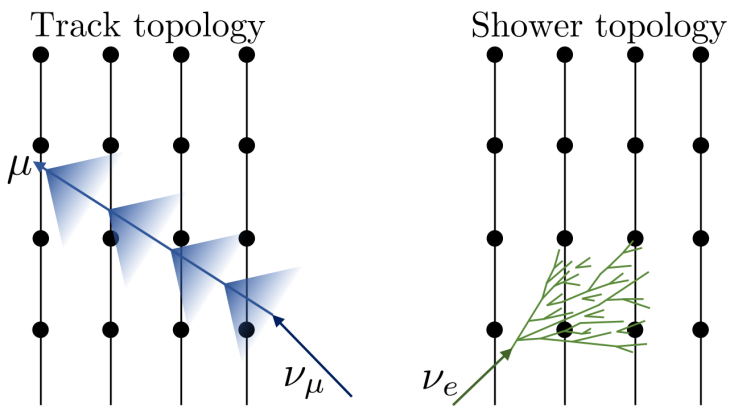


Figure 3.2: *A figure showing the characteristic track and shower signatures in the OCRCA detector, with the dots and strings representing DOMs and DUs respectively. Image taken from Nauta (2022) [35].*

they have sufficient energy, may in turn produce Cherenkov radiation. For what are called *hadronic* showers, which are caused by NC interactions, the process described above covers the full process.

However for *electromagnetic* showers caused by CC interactions there is an extra effect. The highly energetic electrons and positrons that are formed not only produce Cherenkov radiation, but also give rise to high energy bremsstrahlung. These high-energy photons may through *pair-production* result in more electron/positron pairs. These in turn may produce more bremsstrahlung, again giving rise to a shower [36].

Due to the rapid loss and dissipation of energy in the shower processes, the showers produced by atmospheric neutrinos are very concentrated.

3.4.3 Track

Track events are caused by CC $\nu_\mu(\overline{\nu}_\mu)$ and $\nu_\tau(\overline{\nu}_\tau)$ (with a branching rate of 17%) interactions [7]. This is caused by the μ produced in these interactions. As the rest mass of muons is roughly 200 times that of electrons, their deceleration through interaction with the water happens at a slower rate. Because of this, the bremstrahlung produced by these muons does not have enough energy to give rise to pair-production of electrons/positrons. Their mass is conversely still low enough, that with a lifetime of about 2.2 microseconds they are sufficiently stable to traverse the full detector volume without decaying [16]. Due to these factors, these events can be very well approximated by only taking account the Cherenkov cone produced by the muon.

3.5 Background Noise

There are also a variety of processes that produce noise that may be detected. Among these are *dark noise*, bioluminescence and radioactive potassium-40 (K-40) decay [35]. Dark noise is thermal noise produced by the PMTs themselves. Bioluminescence is the light produced by certain organisms in the sea, and K-40 noise is due to potassium present in the seawater. These noise signals grouped together are also called optical noise. Optical noise is filtered out or removed from the data with relative ease. Therefore we will largely ignore it in this work and focus on the main source of background noise in the data: atmospheric muons.

Muons produced in the atmosphere by processes discussed in section 2.3 constitute almost all of the events that pass the initial filters and are stored. This is because muons that reach the detector cause a track event that is very hard to distinguish from track events caused by the muons resulting from CC $\nu_\mu(\overline{\nu}_\mu)$ interactions inside of or near the detector. Luckily, most of these muons can be filtered out because the lifetime of muons with about 2.2 μ is too short for them to be able to pass through the earth. They also interact too strongly with the matter in the earth. Therefore there can be no truly upgoing atmospheric muons that reach the detector. Thus selecting only events that are reconstructed to be upgoing removes most of the muons. Still, due to imperfect reconstruction algorithms and the sheer quantity of atmospheric muons, atmospheric muons that are incorrectly constructed as upgoing still constitute about 98% of the signal after filtering and general selection cuts (see sec. 5.2.1). To further separate muons from neutrinos, machine learning classification models are used. Improving this classification process is the goal of this work. More on the classifiers used and the results will be shown in chapters 4 and 7 respectively.

3.6 Reconstruction Algorithms

When a detection event occurs in the detector, the raw detection data is used by reconstruction algorithms to reconstruct information about the event and the potential neutrino. This information includes but is not limited to the position of the collision and the direction and energy of the neutrino. Because the shower and track event signatures are so different, two different reconstruction algorithms are used. This section will explain the main principles behind both of these algorithms. For a more complete overview of the reconstruction algorithms please refer to O'Fearraigh

Version of October 11, 2024– Created October 11, 2024 - 09:35

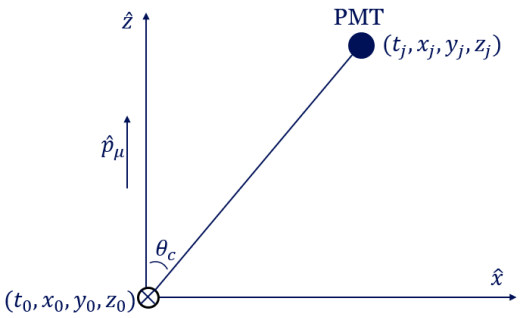


Figure 3.3: *A schematic drawing of the coordinate system used for the track reconstruction algorithm. The z-axis is chosen such that it coincides with the direction of the muon.*

(2024) [36].

3.6.1 Track

The first step of track reconstruction is determining the direction of the track. This is done with maximum likelihood fits based on PMT hits. This is done under the assumption that a clear Cherenkov cone is produced by the track. Then the time residual between the hit time and the estimated time of arrival of the Cherenkov cone of a track candidate can be determined. Given the system as shown in figure 3.3, the expected time of arrival $\hat{t}_j$ of light at PMT j is given by equation 3.1.

$$\hat{t}_j = t_0 + \frac{z_j}{c} - \frac{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}\tan\theta_c}{c} \tag{3.1}$$

Because maximum likelihood fits are often plagued by local minima, multiple slightly different fit algorithms are applied consecutively, each bootstrapping onto the results of the previous fit. The first *prefit* roughly scans the space of causally connected hits for a couple of first track candidates by minimizing the time residuals over consecutive hits using a least squares fit. These track candidates are then improved upon with each consequent fit iteration. The last direction fit performed by the *Gandalf* algorithm produces the final, most likely fit.

The hits within a pre-defined radius from this track fit are then back-projected along the muon track. This allows for the reconstruction of the start and end positions of the track, and thus also the tracklength. The start of the track is also called the vertex. To estimate the energy, predetermined photo-electron distribution tables are used that describe the

expected number of hits in a PMT as a result of the Cherenkov light from a track or shower event. Combined with the PMT hit probabilities calculated during the direction fit, this allows for a likelihood fit of the muon energy at the start of the track.

3.6.2 Shower

Shower reconstruction also occurs in separate consecutive steps. Because the energy range of interest for ORCA is relatively low, the photon-emission and electron/positron pair production cycle described in section 3.4.2 is short-lived. The resulting shortness of the shower makes it difficult to precisely determine the direction of the neutrino. To remedy this, the vertex is reconstructed first. Again hits that are causally connected are selected to eliminate background noise. The photons caused by the event are assumed to spread along a spherical wavefront. Under this assumption, equation 3.2 should hold for PMT j.

$$(\hat{t}'_j - t'_0)^2 = (x_j - x_0)^2 + (y_j - y_0)^2 + (z_j - z_0)^2 \tag{3.2}$$

Where $t' = nct$ with n the refraction index of seawater. By minimising the difference between the left and right hand side of this equation across all selected PMT hits, the vertex can be fitted. This process happens in iterations of increasing accuracy, similar to the track direction, ending in a maximum likelihood fit. These fits also use tables for photo-electron distributions like the ones mentioned in the previous section, now following the point-source assumption. Using the hit probabilities from the vertex likelihood fit and these photo-electron distribution tables a prefit of the event's energy is made. With all previous vertex and energy fits as input, along with electro-magnetic shower photo-electron distribution tables, a prefit of the shower direction is then made. Finally a last maximum likelihood fit is applied to both the energy and direction of the event, using all previous fits and information as input.

3.7 Simulations

To verify observations made with the detector, simulations of events are necessary. This section will explain the chain in which the simulated data is created, and discuss the weighting of the simulated data.

3.7.1 Simulation Chain

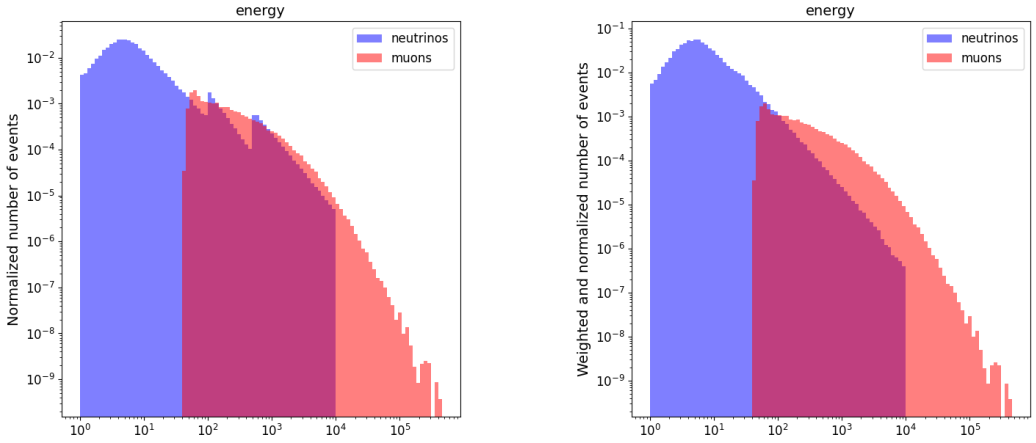
This is done in a chain of different simulation software packages. This chain starts by defining the model of the detector. For this work the ORCA10 version of the detector is used. Next, events are generated. For neutrinos these events are generated using gSeaGen [9], and for muons MUPAGE [14] is used. Both use Monte Carlo (MC) techniques to simulate events. For the muon events only the muon itself and its path are simulated, as it is assumed that the muon will not decay near the detector. For neutrinos, the particles produced in the interaction events are simulated as well, as these may produce light of their own. In either case, only events that are close enough to the instrumented volume of the detector to trigger hits are simulated.

From this point on the simulation process is the same for muons and neutrinos. The next step is simulating the light that would be produced by the particles in the event. Two packages are used for this process. JSirene is a light-weight software that uses pre-computed PDF tables of the arrival time of light. To verify the correctness of JSirene a Geant-4 [8] based package called KM3 is used. KM3 simulates the full propagation of individual photons and is thus much computationally heavier. Both JSirene and KM3 take into account real PMT responses.

The trigger algorithms, JTriggerEfficiency and JTriggerEfficiencyRBR, and the reconstruction algorithms as described in the previous chapter, are now applied to create the data used for this work. These algorithms are exactly the same as those that are applied to real data.

3.7.2 Weights

To get sufficient statistics for robust analysis, and to ensure the training data for the classification models sufficiently covers all possible types of events, events are simulated with a different energy dependence than that of real data. This ensures that for example neutrinos with high energy ($E > TeV$) are still represented in the data while their occurrence is rare. This is compensated by giving a weight to each event. This weighting process is automatically handled by the simulation software. This weight is called w_2 in the simulation software, and this nomenclature will be used in this work as well. Figure 3.4 shows how the weighting of neutrinos impacts the energy distribution of events.



(a) ***(b)***

Figure 3.4: *The (a) unweighted and (b) weighted distributions of the true energy of events in the shower-like training dataset. The weighted distribution also takes into account neutrino oscillations, as will be discussed further in section 5.3.1.*

Chapter 4

Classification

As mentioned before, the goal of this work is to separate events with signals from neutrino interactions from events with signals from atmospheric muons in ORCA10 data. This chapter will discuss the different classification algorithms used for this work. As all of these algorithms are build on *decision trees*, we will start there.

4.1 Decision Trees

Decision trees are among the most commonly used classification algorithms. This is due partly to their relative simplicity and to their explainability. Different than for other ML techniques, it is easy to follow the reasoning of a decision tree by looking at its structure. A decision tree starts with a root node. Each node in a decision tree is split into two, based on a split value for a feature of the dataset. The order in which the features are selected and the split values are chosen to maximally separate the different groups in the data from each other. There are different possible measures to quantify this separation. Common examples are *Gini impurity* and *log loss*. The log loss term for a single data instance in a dataset with k different classes is shown in equation 4.1 [2], and is closely related to entropy of information.

$$L = -\sum_{k} p_k \log p_k \tag{4.1}$$

Here k represents the index of the the different classes in the data. p_k is the estimated probability of the instance to belong to class k. The branches that result from this initial split can then grow nodes and split again. This iterative process continues until all leaves of the tree, meaning all groups of

data that will not be split again, are classified correctly. This also immediately showcases one of the dangers of using decision trees: they are prone to overfit on the data. Therefore it is common to set a maximum depth and/or minimum amount of datapoints in a leaf to combat this. Decision trees are also limited because they perform poorly on data with complex non-linear relations, such as is often the case in the physics setting.

4.2 Ensembles

A way to combat the disadvantages of decision trees is to combine the output of multiple different trees to produce the final prediction. These types of combined models are called *ensembles* in general, and *decision forests* in the context of trees. While individual trees are very sensitive to overfitting and variations, and thus have what is often called high *variance*, they have low systematic error or *bias*. By averaging out the variance over multiple trees, the final answer will have both low variance and low bias. Because of the higher dimensionality of the resulting decision space, decision forests are also capable of learning the complex non-linear relations in data. There are two main methods for building decision forests, known as *bagging* and *boosting*.

4.2.1 Bagging

Bagging is the strategy of training all of the trees in the ensemble in parallel on different subsets of the data. This increases the variance of each individual tree, and increases the variations between trees. Due to these variations the decision forest itself becomes robust to overfitting, and averaging over the entire forest removes the variance. Because each tree is initialised separately and trained on different data subsets, these types of ensembles are often called Random Decision Forests.

4.2.2 Boosting

When the boosting method is used, trees are trained sequentially. This allows each new tree to be trained specifically to improve upon the previous tree(s')'s performance. Through this, a boosted decision forest also lowers the total bias of the prediction, while still keeping variance low. This method is most beneficial when the underlying relations in the data are complex and non-linear. It also makes boosted decision forests better at handling rare events in the dataset. These factors combine to make

boosted decision forests a natural choice for the classifier type in this work. Following convention, tree ensembles using boosting are called Boosted Decision Trees (BDTs).

4.2.3 Gradient Boosting

Gradient Boosting is a generalization of the boosting method, which formulates the training process to have a cost function, and optimizes the building of each consecutive tree through gradient descent of that cost function [21].

4.3 Model Options

Multiple different BDT software packages were used for this work. The GradientBoostingClassifier (GBC) from Scikit-Learn [38] is a popular choice. Important for our research is that GBC allows the user to inspect the importance of features in the decision making process. Because of the large amount of features used, this narrows down the features selected for further analysis (see sec. 5.3). However it is also very computationally expensive for large datasets, as it enumerates all split candidates. This makes it a poor choice for use with hyperparameter optimization. That is why the HistGradientBoostingClassifier (HGBC) from Scikit-Learn was also considered. This uses essentially the same algorithm, but is much faster as it uses histograms to bin the split candidates. However, it loses the availability of feature importance. Therefore it was chosen to not use HGBC.

The main BDT algorithm used for this work is the eXtreme Gradient Boosting Classifier (XGBC) implementation form XGBoost [15]. In doing so we follow earlier work done with ORCA6 data [18]. This algorithm boasts improved scalability and computational efficiency for large datasets, and has been used in the winning solutions of many machine learning challenges. It has the feature importance variable available, and also by default uses histograms of binned data to approximate the split points.

4.4 Optimization

For many machine learning problems, the tuning of the hyperparameters of a chosen model has significant impact on the model's performance. As

the goal of this work is to optimize classification performance, it is crucial to take the optimization of hyperparameters into account. There are many different packages available for hyperparameter optimization. This work uses the optuna package [10] due to its ease of use across different machine learning packages (Scikit-Learn/XGBoost). While multiple sampling methods are available, for this work the sampler based on Tree-Parzen Estimators (TPE) is chosen.

Tree-Parzen Estimation is a Bayesian based method. It works by fitting the previously tested hyperparameter values with their corresponding score to a distribution. The next set of hyperparameters is selected by selecting the values in the hyperparameter search space that have the highest probability of resulting in a better score [39].

Chapter 5

Data

The data used for this work was extracted from the v9.0 version of ORCA10 MC simulations as described in section 3.7. The raw output files of these simulations are then summarized in what are called dst files. To extract the features used to train the classification models, and to clean the data, the dst files are put through a pre-processing pipeline. The pre-processing steps are shown in detail in appendix A. A complete list of all of the training variables and their importance as found by the XGBC basemodels for track- and shower-like data can be found in appendix B. There an explanation of the most important features is also already given. For the sake of legibility, the prefixes 'T.feat_Neutrino2020.' and 'T.sum_' will be dropped from features names when discussing them.

This chapter will first introduce the concept of *scores*. Then the different data selections used in this work will be shown. It finishes with an analysis of these data selections, and a preliminary examination of the features that are most important for classification.

5.1 Scores

For this work it is important to have the estimated probability for each event to be a muon, a track event, or noise, as produced by the current standard classification method [18, 32, 37]. This process includes applying three separate BDTs to the data. Before training the track/shower and neutrino/muon classifiers, a cut is made that leaves only events that are reconstructed as upgoing. In both cases, the training sample consists of 10^5 datapoints for each class. For the track/shower classifier, $\nu_\mu(\overline{\nu_\mu})$ and $\nu_e(\overline{\nu}_e)$ CC events represent the track and shower classes, respectively. For

the neutrino/muon classifier, an extra cut was made based on the fitted likelihood of the event based on the track reconstruction algorithm. Here, only $\nu_\mu(\overline{\nu}_\mu)$ CC ($\approx$ 70%) and $\nu_e(\overline{\nu}_e)$ CC ($\approx$ 30%) events were used to represent the neutrino class in training. The probabilities for an event to be in one of the classes as found by these classifiers, is called the *score*. The track- noise- and muonscores are currently produced using the ParamPID code [33].

5.2 Selection Cuts

To reduce the full dataset to only the subsets of the data that are of interest to oscillation analysis, selection cuts are applied. We follow the selection cuts as described on the KM3NeT internal wikipage [4]. Note that in these selection cuts, the muonscore is mentioned. These muonscore based cuts are not applied to the data, as for training and testing our new classifiers we assume we do not yet have an available muon score. However the cut values described here will be used to determine the benchmark figures of merit later on by which we compare performance. This section will explain the three selection cuts.

5.2.1 General Selection Cuts

The first selection is called the general selection cut. This cut in effect selects all of the data that is of interest in oscillation analysis, without further subdividing it. The cut criteria are:

- rectype_JShower == 4000. This ensures that all of the selected events are successfully reconstructed by the shower reconstruction algorithm.
- rectype_JGandalf == 4000. This ensures that all of the selected events are successfully reconstructed by the track reconstruction algorithm.
- anti_sparks_DOM == 1 (Dist. to closest DOM >3m & maximum-ToT_triggerhit<160). These cuts ensures that light originating from close to the DOMS and that trigger the DOMS for more that 160 nanoseconds are removed. In real data, these types of signals are predominantly produced by electrical discharges or *sparks* originating in the DOMs themselves.
- cos_zenith_recoJGandalf <0. This selects only events that are reconstructed by the Gandalf track reconstruction algorithm as going up.

Version of October 11, 2024– Created October 11, 2024 - 09:35

- noisescore $\leq$ 2e-5. This removes nearly all of the noise from the data while removing a minimal amount of neutrinos.

- muonscore $\leq$ 2.9e-3. This cut aims to remove most of the muons from the signal, while removing the least amount of neutrinos.

Besides these general selection criteria being applied to all of the data, there are three other subsets of selection cuts that are made in the eventual analyses. This is done because it improves the overall fit quality on the oscillation parameters.

5.2.2 Track Selection Cuts

The track selections cuts select for events that are most likely tracks. We will from here on out refer to this data as the track-like data. This is not to be confused with data consisting of only true track events. While for training and test purposes there is one set of track-like data, as no cuts based on muonscore are made yet, this data set is further subdivided into high and low purity tracks for analysis purposes.

Low Purity Tracks

The events that pass these criteria are most likely tracks, but the signal might consist for a few percent of muons (see table 7.1).

- general selection cuts.

- trackscore $>$ 0.56. Selects events that are likely track-like.

- 2 (GeV) $\leq$energy_recoJEnergy $<$ 50 (GeV). Selects the events within the energy range of interest for tracks, as reconstructed by Gandalf.

High Purity Tracks

This selection aims to remove all muons in the signal.

- low purity tracks cuts.

- muonscore $< 1.5e-4$.

Version of October 11, 2024– Created October 11, 2024 - 09:35

5.2.3 Shower Selection Cuts

- general selection cuts.
- trackscore ≥ 0.56.
- 2 (GeV) $\leq$ energy_recoJShower < 1000 (GeV). Selects the events within the energy range of interest for showers, as reconstructed by the shower algorithm.
- cos_zenith_recoJShower < 0. Selects events that are upgoing according to the shower reconstruction.

5.3 Data Analysis

5.3.1 Number of Events

When analysing the data, it is important to take into account the weight of each simulated event. The correct weight of an event can be split into two parts. First, the expected distribution of events per the atmospheric neutrino flux as described in sec. 2.3 needs to be taken into account. This is the w_2 weight, as explained in section 3.7.2. The next step is taking into account the oscillation of neutrinos. As explored in sec. 2.5, upwards going neutrinos (may) oscillate during their path through the earth. The correct weight w_{osc} for a neutrino of flavour β arriving at the detector is shown in equation 5.1.

$$w_{osc} = w_2 \sum_{\alpha=e,\mu,\tau} [f_\alpha(E, \cos\theta_z) P_{\alpha\to\beta}(\alpha, \beta, E, \cos\theta_z)] \frac{t_{exp}}{N_{gen}} \tag{5.1}$$

Where f_α is the atmospheric flux for neutrino flavour α with energy E and arriving with zenith θ_z, according to the flux values determined by Honda et al. [26]. $P_{\alpha\to i}$ is the probability of neutrino of flavour α to oscillate to flavour β, which also depends on the energy and the zenith. t_{exp} is the amount of detector exposure time that simulation run represents, and N_{gen} is the amount of these specific types of particles that were simulated during the simulation run.

The data that is left per particle type is shown in tables 5.1 and 5.2 for the shower- and track-like datasets respectively. Here the data is already shown split into separate test and training datasets, with a 0.2:0.8 test:train split ratio. This is done to allow for a test of the models on data that they

Version of October 11, 2024– Created October 11, 2024 - 09:35

type	interaction	# datapoints train	# datapoints test	# real events train	# real events test	exposure (yr)
muon		47375	11844	46113.2	11547.1	0.4
anumu	NC	30457	7516	20.1	5.1	0.4
numu	NC	42971	10883	64.3	16.7	0.4
anue	CC	25190	6298	84.7	20.5	0.4
nue	CC	43387	10847	216.	54.3	0.4
anutau	CC	5995	1499	18.4	5.0	0.4
nutau	CC	6945	1736	50.8	10.6	0.4
numu	CC	39706	9786	307.	76.6	0.4
anumu	CC	14146	3635	119.4	30.7	0.4

Table 5.1: *An overview of the data per particle type after the shower selection cuts. Exposure shows how much real data-taking time the simulation data used for this work equates to.*

have not seen during hyperparameter optimization. To keep the distribution of particles in the test and training datasets as close as possible to the true distribution, the splitting was stratified using the particle types. This is imperfect, as weights within particle types can vary significantly. That is why undersampling of anutau CC events with about a factor 3 can be observed in the track-like test dataset.

5.3.2 Feature Importance

To get a first sense of which features are important in classifying the data, we use the feature importance results from the XGBC basemodel.

It is clear by looking at figures 5.2 and 5.1 that while the decision process differs between the track- and shower-like data, there are a couple of features that are important for both. Examples of this are parameters that provide hit information such as the average z-position of the hits and the difference in amount of hits used in reconstruction of the best upgoing and downgoing tracks. It also seems that for classification on track-like data, the top 5 features really dominate the classification in terms of importance. Whereas on the shower-like data, the importances are more spread out. Note that this analysis is in no way conclusive. Different initiations of the XGBC, different hyperparameters, and different splits on the data may all impact the order and importance of the features. 1-dimensional histograms are also not capable of capturing complex relations between fea-

type	interaction	# datapoints train	test	# real events train	test	exposure (yr)
muon		497546	124386	484459.9	121095.1	0.4
anumu	NC	7377	1741	3.5	0.8	0.4
numu	NC	9831	2435	11.5	2.8	0.4
anue	CC	5583	1396	11.5	3.	0.4
nue	CC	8718	2180	28.6	7.0	0.4
anutau	CC	1946	487	5.2	0.4	0.4
nutau	CC	2067	516	12.6	3.5	0.4
anumu	CC	43352	10941	362.1	91.9	0.4
numu	CC	95896	23887	719.1	181.1	0.4

Table 5.2: *An overview of the data per particle type after pre-processing and the track selection cuts. Exposure shows how much real data-taking time the simulation data used for this work equates to.*

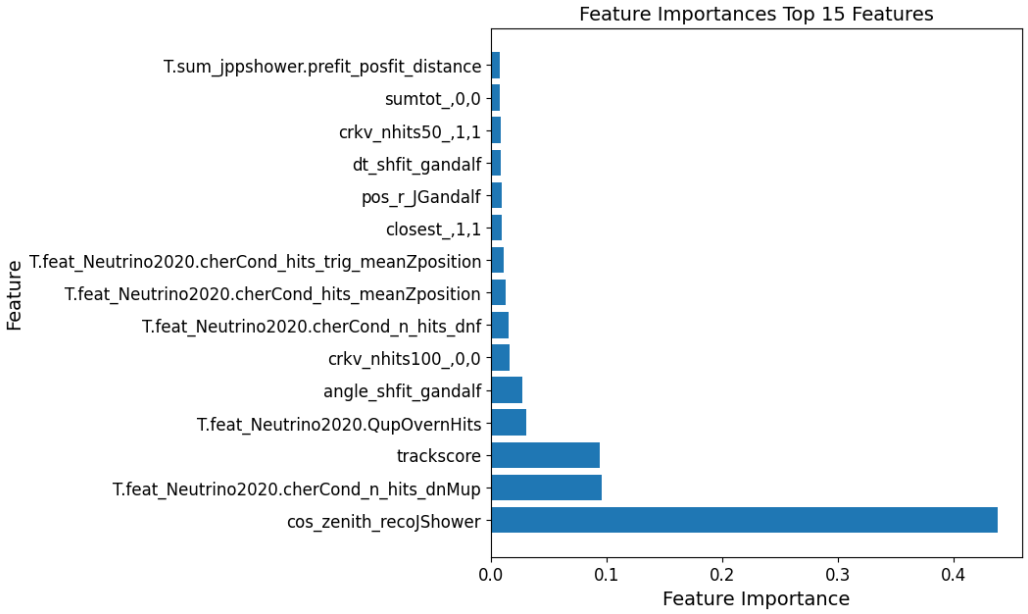


Figure 5.1: *The top 15 most important features as determined by the basemodel XGBC when trained on the track-like training dataset.*

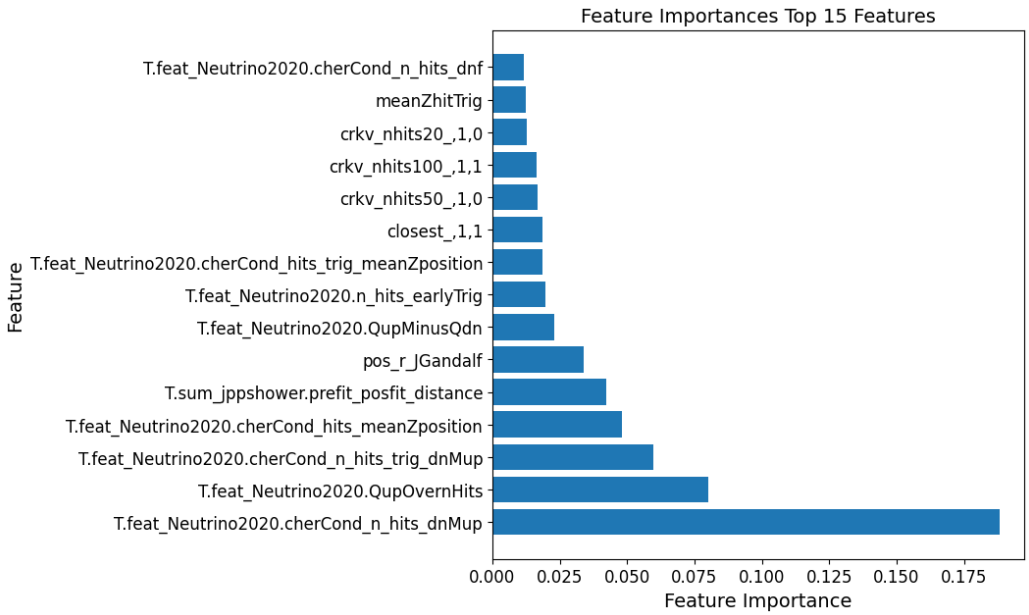


Figure 5.2: *The top 15 most important features as determined by the basemodel XGBC when trained on the shower-like training dataset.*

tures that might give features importance in a higher dimensional phase-space of features. However, these figures provide a starting point to further analyse some of the features that are at least of reasonable importance for classification. Features with relatively large importance are expected to be relatively consistent in importance, and are likely to show a more explainable impact on classification. In the next section, we shall therefore take a closer look at the top 5 features for both shower- and track-like data.

5.3.3 Feature Distributions

The histograms presented in this section are weighted according to the event weights. They show the distribution of muons and neutrinos according to a certain feature. To be able to better represent the difference in distribution without taking into account the number of events, the distributions are normalized.

Track-like Data

First we shall explain what the features represent.

- cos_zenith_recoJShower: the cosine of the zenith angle of the particle

as reconstructed by the JShower algorithm.

- cherCond_n_hits: the amount of hits that fit the Cherenkov cone hypothesis for the best fitted downgoing track minus those for the best fitted upgoing track. The conditions for a hit to fit the Cherenkov cone hypothesis are: a difference between expected and actual hit time < 15 ns, a distance between the PMT and the track < 100 m, and angle between PMT orientation and photon direction < 90 degrees.

- trackscore: the probability of the event to be a track event (see sec. 5.1).

- QupOvernHits: the quality of the best fitted upgoing track divided by the amount of hits used in the fit. This indicates the goodness of the upgoing track fit.

- angle_shfit_gandalf: the angle between the best fitting shower and track reconstruction directions.

Figure 5.3a shows that the shower reconstruction algorithm (rightly) reconstructs nearly all of the muons in the track-like data as upgoing. The starkly different distribution make it clear why this is such an important feature. Figure 5.3b shows that the hits for muons on average fit about as wel to an upgoing as a downgoing Cherenkov cone. Clearly this makes it difficult for the reconstruction algorithms to determine the direction of the track for muons. It also is clear that if the feature value is larger than about 50, the event is almost always caused by a neutrino. As for the trackscore in figure 5.3c, it seems that a large majority of the neutrinos in this cut have a very distinct track-like signature. This is not true for muons, although they should produce track-like signatures. This again indicates the indistinct signature produced by these muons. This is affirmed by QupOvernHits which is worse for muons than it is for neutrinos, as seen in figure 5.3d. The on average large difference in direction between the JShower and gandalf reconstructions in figure 5.3e also hints at badly definable muon events. It is also very much in line with figure 5.3a.

The poor fit quality and indistinct up/down cherenkov cone signatures are most likely due to these muon events being detected at the edge of the detector volume, as can be seen in figure 5.4. More figures supporting this claim both for the track- and shower-like datasets can be found in appendix D.

Shower-like Data

For the shower-like data we shall also first explain what the features represent.

- cherCond_n_hits_trig_dnMup: this feature is very similar to chercond_n_hits_dnMup, and the two are highly correlated. The only difference is that whereas cherCond_n_hits_dnMup takes into account all hits that fit the criteria, n_hits_trig_dnMup only takes into account the hits that pass the initial trigger filtering algorithms.
- cherCond_hits_meanZPosition: the mean z-coordinate of hits that fit the Cherenkov cone hypothesis.

For cherCond_n_hits(_trig)_dnMup, and QupOvernHits the same explanation for the distributions (figures 5.5a, 5.5c, 5.5b) holds as for the track-like data. However the value ranges here are ab it larger. This is likely because the reconstructed energy range allowed for the shower-like data is larger than that for track-like data, going up to 1000 and 100 GeV respectively. Higher energy events give off more light and thus usually result in more hits. Figure 5.5d supports the claim that the muons is this dataset are mostly detected at the edges of the detector volume. Whereas the neutrinos are quite uniformly spread out across the z-axis of the detector. Lastly, figure 5.5e shows that the distance between the vertex fitted in the prefit and the final reconstruction vertex is also much larger for muons than for neutrinos. The extreme range on the distance is an artefact of extremely poorly fitted events.

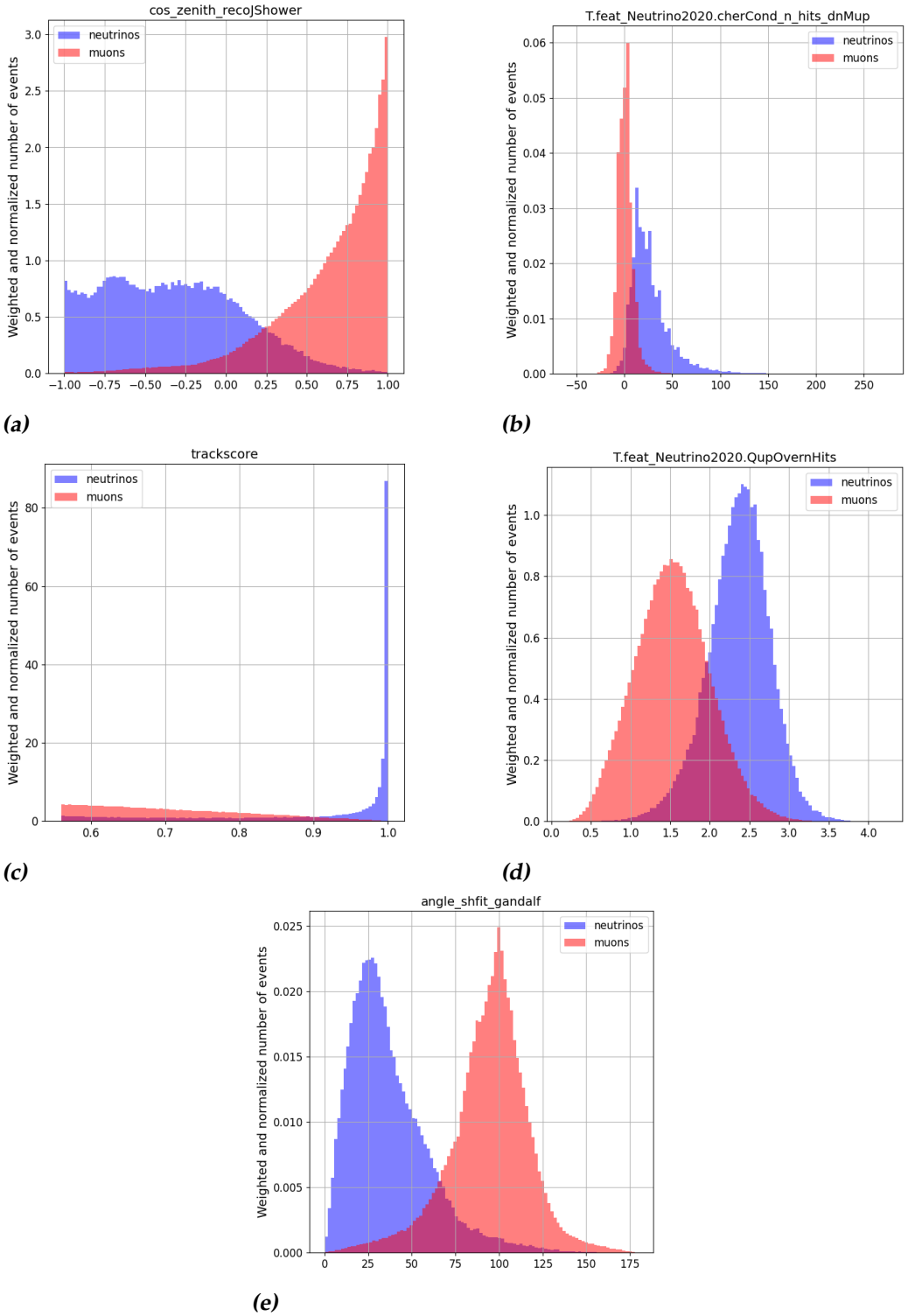
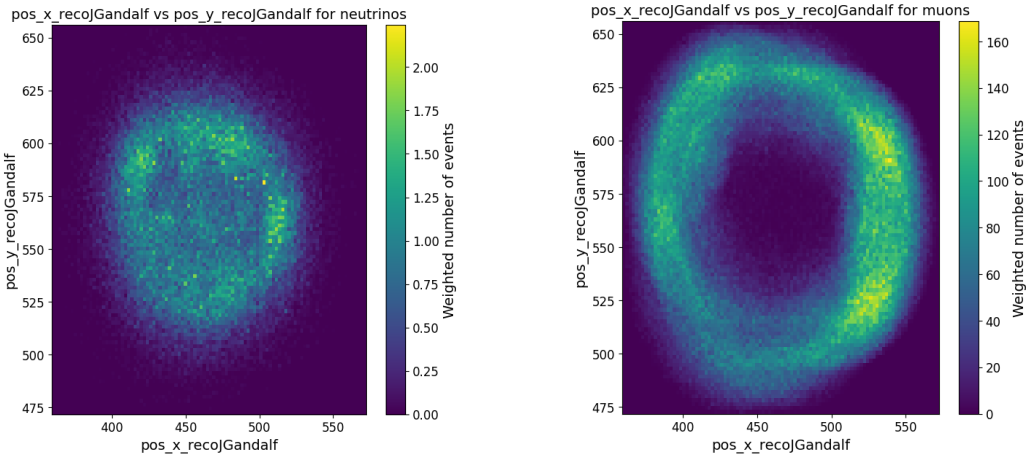
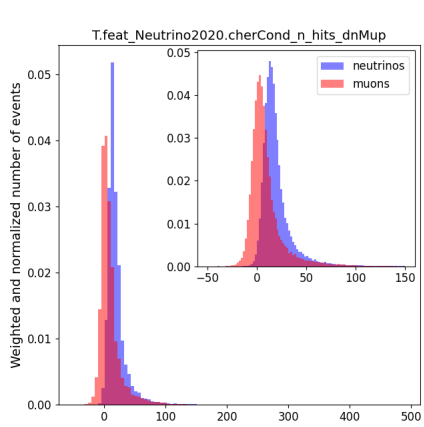


Figure 5.3: *Feature distributions for top 5 most important features for track-like data classification, with seperate distributions for muon and neutrino data.*

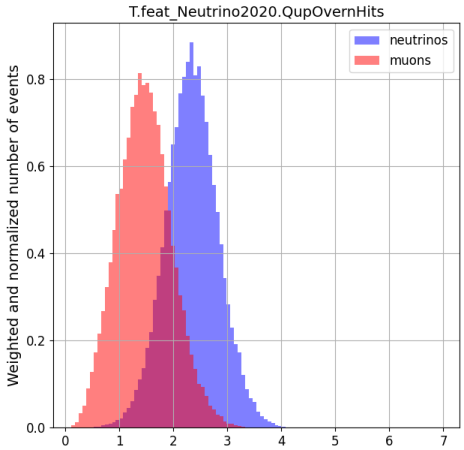


(a) ***(b)***

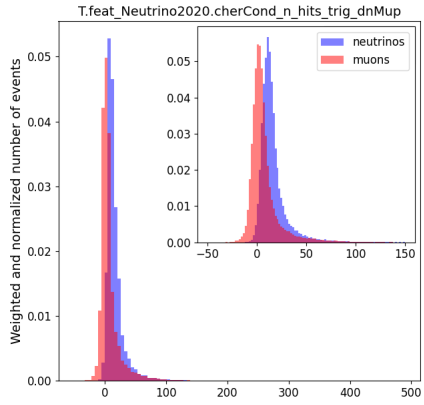
Figure 5.4: *2-dimensional distributions of the x- and y-positions of the track-like events as reconstructed by the track algorithm.*



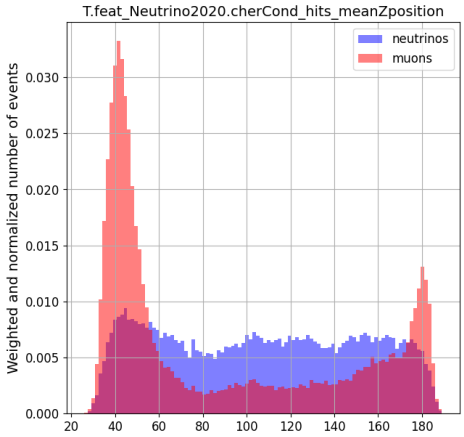
(a)



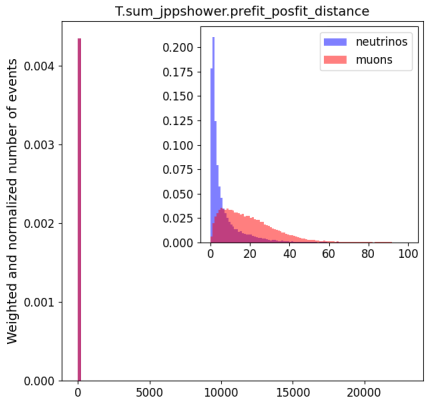
(b)



(c)



(d)



(e)

Figure 5.5: *Feature distributions for top 5 most important features for shower-like data classification, with seperate distributions for muon and neutrino data..*

Chapter 6

Methods

This chapter describes the experimental methods used for the results in this work. First the different performance targets used are explained. Then the base model and its optimization are discussed.

6.1 Data Subsets

For this work, an approach different from the current workflow (see sec. 5.1) for training and testing a classifier to distinguish muons from neutrinos are used. Because the different selection cuts eventually used in oscillation analysis are known, it makes sense to use these to create the separate shower and track classification datasets. Because the signatures are different, the optimal decision classification process is likely also different. This is affirmed by the analysis in section 5.3. Creating these subsets before classification necessitates training the noise and track/shower classifiers first. This is entirely possible, even without changing anything from the currently used approach.

6.2 Targets

6.2.1 Contamination and Efficiency

The muon contamination C_μ we define as the percentage of the leftover signal that consists of muons. For the neutrino efficiency E_ν, the percentage is determined by the ratio of neutrinos that are left after applying the score threshold(s), compared to the full amount of neutrinos in the dataset. For both C_μ and E_ν, the event weights are taken into account.

6.2.2 Optimization Targets

The optimization targets are determined by looking at the muon contamination and neutrino efficiency at the 'old' muonscore thresholds. This is done separately for the test and train subsets for both the track- and shower-like datasets. For the track-like data, these targets are also determined separately for the different Low Purity (LP) and High Purity (HP) track muonscore thresholds.

The threshold values for the new models are then chosen such that the contamination values achieved on the track LP/HP and shower training sets match those of the 'old' muonscore. This is done by looping over threshold values assorted on an increasing log scale. When the C_μ measured at a threshold exceeds the optimization targets, the previous threshold is taken as the threshold with which to produce the classification targets. In this way, the neutrino efficiency at the set muon contamination targets can used as the score value to be optimized. Note that as the threshold previous to exceeding the muon contamination target is used, the performance of the model is slightly underestimated.

For track-like data, the LP and HP performance needs to be optimized at the same time. To achieve this the track optimization target is a combination of the neutrino efficiencies at the LP and HP thresholds. Because increasing E_ν for the HP dataset is most important, this is given a larger weight in the combined score. The final score is reported in the following equation 6.1.

$$score_{track} = 0.1E_\nu^{LP} + 0.9E_\nu^{HP} \tag{6.1}$$

6.2.3 Classification Targets

The final targets for classification are slightly different from the optimization targets. After determining the optimization targets, the target C_μ for the different data subsets are known. Because the muonscore thresholds are adapted for each different model to match these contamination values, C_μ is omitted from the classification targets. Instead, we zoom in on E_ν. To get a more in depth analysis, the neutrino efficiency is determined per particle type and interaction combination. Through this it becomes visible how difficult the different neutrinos and interactions are to distinguish from muons.

6.3 Models

6.3.1 Basemodels

The two basemodels used for the results in this work are the default implementations of the XGBC [3] and GBC [1] for reasons explained earlier in section 4.3. They are the same for both track- and shower-like data. In all cases, the random seed is set to 42. This allows for reproducible results. For the GBC, the amount of estimators (trees) used is changed from the default 100 to 101. This removes unwanted behaviour of the muonscore in edge cases.

6.3.2 Tuning

While base model results are produced for both XGBC and GBC, only the XGBC models are optimized. This is due both to the much larger computational cost of the GBC models, and the superior performance of the XGBC basemodels (see sec. 6.3). The hyperparameter tuning process occurs in trials. In each trial, a 5-Fold cross-validation loop is applied to the training data. This means that a model with the new trial hyperparameters is initialized and trained 5 times, on 5 different training and validation subsets. When the loop is done the optimization target values across the full training data are determined. In this way the optimization target values are averaged, which increases stability in the optimization process. The hyperparameters and the allowed value ranges that were used during this procedure can be found in appendix C. The optimal hyperparameter values for the optimized models can also be found there. For both track- and shower-like data, 500 trials were completed to arrive at the final model.

6.4 Test and Train

As discussed earlier in this report, the full data per event type (track/shower) has been split into separate train and test datasets using stratification on the particle type. Naturally the final check on model performance is produced by training the model on the full training dataset and testing it on the test data. These are the results that will be indicated by 'test'. Due to the 5-Fold CV procedure employed during optimization, results are also available for the model on the training dataset. These will be indicated with 'train'. For the results in this work, both of these will be included. This enables us to verify whether the hyperparameters of the models have

been overfitted to the training data set. If this is not the case, the train results are also valuable results, as the train dataset is larger and follows the distribution of true data better.

Chapter 7

Results

Here the results from our experiments are presented. The discussion about the results and their interpretation will take place in the chapter 8.

7.1 Optimization Targets

From table 7.1 it is clear that for HP tracks, there should be no muons left in the data. For LP tracks contamination of around 2% is accepted.

7.2 Models

Our results show a distinct pattern in models' performances. The 'old' and GBC basemodel performances are on average very similar. There are deviations between the different test and train splits, and between different particle types within these splits, where the best between the two of them varies. The differences are largest on E_ν^{HP} for track-like data in table 7.3. These large differences are possibly due to the imperfect split method

$C_\mu^{shower}(\%)$		$C_\mu^{HP}(\%)$		$C_\mu^{LP}(\%)$	
test	train	test	train	Test	train
2.08	1.42	0.0	0.0	2.66	1.70

Table 7.1: *The muon contamination values established by the selection criteria described in sec. 5.2 and using the 'old' muonscore. These are the targets used for the optimization procedure.*

Version of October 11, 2024– Created October 11, 2024 - 09:35

type	interaction	$\%_\nu$	$E_\nu^{HP}(\%)$				$E_\nu^{LP}(\%)$			
			old	gbc	xgdc	*xgdc**	old	gbc	xgdc	*xgdc**
anumu	NC	0.3	21.7	20.6	44.5	**50.6**	50.6	51.3	**64.5**	**64.5**
numu	NC	1.	22.4	21.0	45.	**48.7**	48.5	50.0	62.8	**64.9**
anue	CC	1.	32.3	28.1	56.4	**58.8**	64.9	57.4	72.6	**73.7**
nue	CC	2.5	35.7	30.8	57.6	**63.1**	66.4	62.1	74.7	**77.1**
anutau	CC	0.4	64.2	67.9	92.5	**97.8**	86.1	90.1	**99.1**	98.1
nutau	CC	1.1	50.2	54.9	76.0	**81.9**	81.8	84.2	**91.2**	87.3
anumu	CC	31.4	57.0	61.8	83.3	**86.1**	85.4	84.3	91.3	**92.2**
numu	CC	62.3	55.7	60.0	81.6	**84.5**	83.4	83.	90.1	**90.8**

Table 7.2: *Neutrino efficiency for different particle types and interactions as achieved on the track-like training data for the different models. $\%_\nu$ shows what percentage of the total neutrino signal in the dataset consists of each of the different type-interaction combinations. Old indicates the performance of the current classification methods (see sec. 5.1). xgdc indicates the basemodel XGDC, and $xgdc^*$ indicates the XGDC after 500 steps of hyperparameter tuning (see sec. 6.3). The best performance is made bold.*

type	interaction	$\%_\nu$	$E_\nu^{HP}(\%)$				$E_\nu^{LP}(\%)$			
			old	gbc	xgdc	*xgdc**	old	gbc	xgdc	*xgdc**
anumu	NC	0.3	30.5	34.7	**56.1**	45.9	56.1	53.4	67.8	**71.5**
numu	NC	1.	21.2	33.5	**57.9**	46.6	53.	54.2	71.1	**73.3**
anue	CC	1.0	27.1	32.9	**56.1**	46.6	56.4	52.0	67.3	**71.1**
nue	CC	2.4	31.9	40.8	**64.7**	53.4	62.1	59.5	75.3	**75.9**
anutau	CC	0.1	45.7	86.2	**92.8**	**92.8**	92.8	92.8	**100.**	**100.**
nutau	CC	1.2	60.9	60.1	**86.7**	81.2	83.0	85.9	90.8	**96.4**
anumu	CC	31.6	56.5	72.2	**87.8**	83.3	84.5	83.7	92.4	**92.6**
numu	CC	62.4	55.7	70.8	**86.7**	81.9	83.7	83.2	91.7	**92.1**

Table 7.3: *Neutrino efficiency for different particle types and interactions as achieved on the track-like test data for the different models. $\%_\nu$ shows what percentage of the total neutrino signal in the dataset consists of each of the different type-interaction combinations. Old indicates the performance of the current classification methods (see sec. 5.1). xgdc indicates the basemodel XGDC, and $xgdc^*$ indicates the XGDC after 500 steps of hyperparameter tuning (see sec. 6.3). The best performance is made bold.*

type	interaction	$\%_\nu$	$E_\nu(\%)$			
			old	gbc	xgdc	*xgdc**
anumu	NC	2.3	69.0	68.5	**83.1**	82.9
numu	NC	7.3	69.	67.5	81.8	**82.1**
anue	CC	9.6	77.5	70.9	**97.0**	87.3
nue	CC	24.5	79.3	71.8	88.1	**88.4**
anutau	CC	2.1	90.	83.9	**96.2**	94.2
nutau	CC	5.8	86.2	83.1	**92.9**	92.5
anumu	CC	13.6	69.7	61.1	79.6	**79.9**
numu	CC	34.9	67.8	60.7	78.9	**79.2**

Table 7.4: *Neutrino efficiency for different particle types and interactions on shower-like training data for the different models. $\%_\nu$ shows what percentage of the total neutrino signal in the dataset consists of each of the different type-interaction combinations. Old indicates the performance of the current classification methods (see sec. 5.1). xgdc indicates the basemodel XGDC, and $xgdc^*$ indicates the XGDC after 500 steps of hyperparameter tuning (see sec. 6.3). The best performance is made bold.*

type	interaction	$\%_\nu$	$E_\nu(\%)$			
			old	gbc	xgdc	*xgdc**
anumu	NC	2.3	69.8	72.4	85.1	**86.6**
numu	NC	7.6	67.3	69.6	83.7	**85.6**
anue	CC	9.4	76.6	73.0	87.9	**89.8**
nue	CC	24.7	78.6	75.	89.6	**90.5**
anutau	CC	2.3	90.6	84.1	97.4	**97.5**
nutau	CC	4.8	79.0	80.9	88.9	**90.6**
anumu	CC	14.	69.8	64.7	82.2	**84.4**
numu	CC	34.9	68.3	64.3	80.3	**82.1**

Table 7.5: *Neutrino efficiency for different particle types and interactions on the shower-like test data for the different models. $\%_\nu$ shows what percentage of the total neutrino signal in the dataset consists of each of the different type-interaction combinations. Old indicates the performance of the current classification methods (see sec. 5.1). gbc indicates the basemodel GBC, xgdc indicates the basemodel XGDC and $xgdc^*$ indicates the XGDC after 500 steps of hyperparameter tuning (see sec. 6.3). The best performance is made bold.*

Version of October 11, 2024– Created October 11, 2024 - 09:35

(see sec. 5.3.1), which are exacerbated through the even smaller selection of events that pass the HP selection criteria. Still, performance on the LP selection criteria is again very similar.

Across all results, the XGDC models outperform the other two models. Often with as many as 15 percentage points. This is a very significant improvement over the old model.

When comparing the basemodel and tuned XGDC, there seems to be a difference between the shower- and track-like datasets. On the track-like data, the optimization procedure appears to have overfit the training data. On the training data the HP efficiencies are highest across the board for the tuned model, but on the test data the basemodel is at least as good for all event types. However the LP efficiencies are highest on the test data for the tuned model. This indicates that the balancing between the HP and LP scores in the optimization target as defined in equation 6.1 is not optimal.

On shower-like data, the base model seems to slightly outperform the tuned model on the training data. However the optimized model then more strongly outperforms the base model on the test data. This is almost the opposite of what one would expect given the results on the train data.

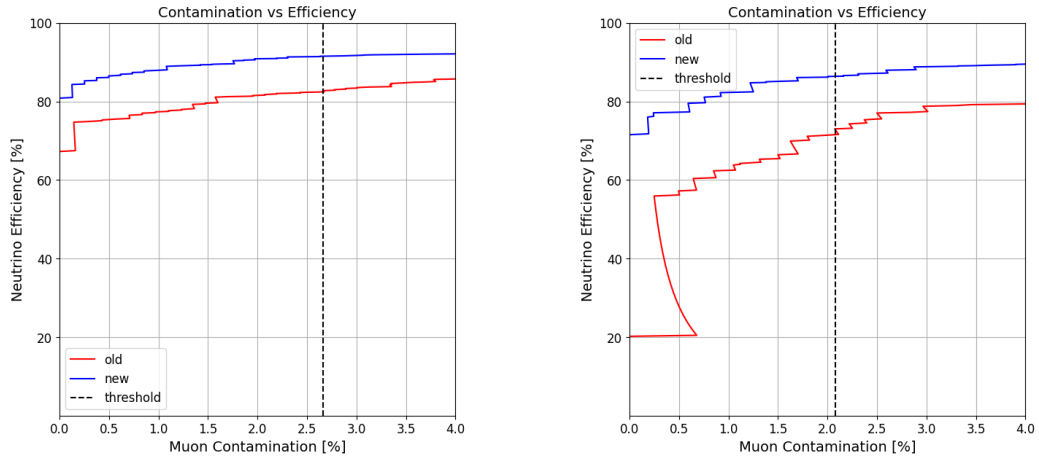
It appears that the margins for optimizing the XGDC model are very small, and that optimizing is non-trivial for this dataset. The XGDC basemodel is clearly already a very strong choice without any tuning, making tuning in this way almost absolute and inconsistent.

Our analysis also reveals a clear pattern in the relative difficulty of the different particles and interaction types, which is consistent across the different models. For track-like data, $\overline{\nu}_\tau$ CC is quite consistently the easiest to classify. $\overline{\nu}_\mu$, ν_μ and ν_τ CC are also consistently easier. ν_μ NC, $\overline{\nu}_\mu$ NC and $\overline{\nu}_e$ CC seem to be consistently difficult for the classifier.

For shower-like data, the pattern is similar but slightly different. Again, $\overline{\nu}_\tau$ CC is consistently the easiest. ν_τ, ν_e and $\overline{\nu}_e$ CC are consistently easy. ν_μ CC, ν_μ NC and $\overline{\nu}_\mu$ CC are on average difficult.

The most notable difference in relative difficulties between the track- and shower-like data are that ν_μ and $\overline{\nu_\mu}$ CC were easy for the track classifiers but difficult for the shower classifiers. Conversely $\overline{\nu_e}$ CC and to a lesser extent ν_e CC are easy for the shower classifiers but relatively difficult for the track classifiers.

Figure 7.1 shows that the tuned XGBC model is not just better than the current classification model at the specific optimization target points. The improvements are consistent for all muon contamination targets.



(a) ***(b)***

Figure 7.1: *The neutrino efficiency E_ν plotted as a function of the muon contamination C_μ for (a) the track-like and (b) the shower-like data. The label 'new' here represents the tuned XGBC model and 'old' represents the performance of the current classification models. The dotted threshold shows the muon contamination set as the target by the current models. For the track data, this is the threshold for the Low Purity events. For more information refer to section 6.2. The events' weights are taken into account.*

Chapter 8

Discussion

The big jump in performance of the XGDC models at first glance seems almost too good to be true, as if there is something wrong with the data selections or the the training features used. However, the fact that the results of the GBC basemodel are in line with 'old' model, and often enough worse, provides some assurance that the method itself is not faulty.

As the current procedure that produces the old performance also uses a XGBC classifier, the difference in performance is likely largely due to the different training data selection procedure. That would mean that the procedure as described in this work, which zooms into the target region of oscillation analysis data and splits the data into shower and track subsets, is very promising and should perhaps become the standard. The different feature importances as shown in section 5.3.2 seem to support the claim that splitting the data into track- and shower-like subsets, allows classifiers to focus on different feature phase-spaces. Seeing the difference between the track and shower signature, it makes sense that the optimal way to distinguish neutrinos from muons is different between the two subsets.

What could also be making a contribution is the amount of features used for classification. In previous work with ORCA 6 data, only 23 features were used in training the muon/neutrino classifiers [18]. In contrast, the basemodels presented in this work use 132 features. The tuned XGBC shower and track models use 74 and 99 features respectively. Assuming the current classification method also used less features, it could be that there is significant gain in performance possible by using more features. This is supported by the fact the optimization procedure selects a large amount of features for both the shower and track models.

Another contribution to the difference in performance could be the different sample sizes for the data. The size of the training datasets used for

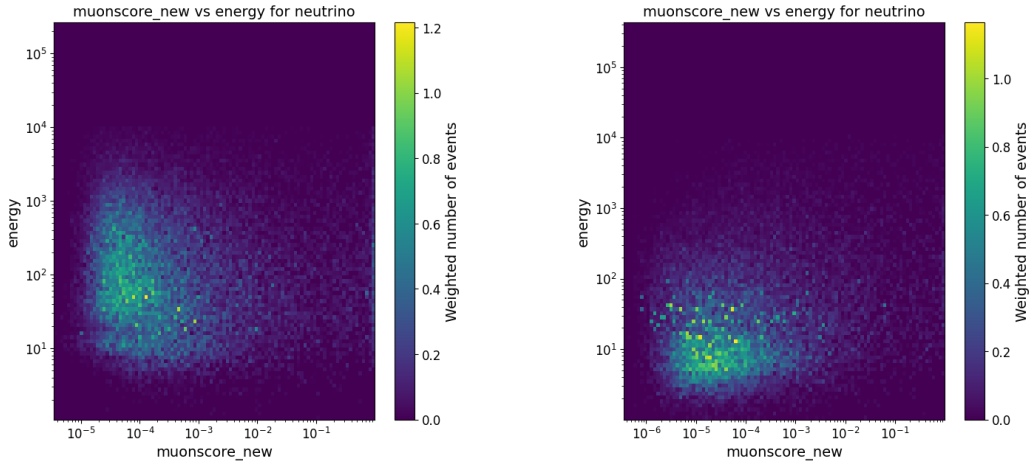
this work are about an order of magnitude larger than those used during the old procedure.

The relatively poor performance of the GBC on the shower-like data shows that this method of splitting the data may have a greater benefit for the track-like data than for the shower-like data. This makes sense, as the muons' signatures are in essence track-like. Training a classifier on classification between track-like neutrino events and muons may allow the classifier to learn more subtle differences. Whereas a classifier that is trained on track-like and shower-like events at the same time may learn to reject muons partly based on their track-like signature, which would decrease the performance of track-like data and boost performance of shower-like data.

In terms of the relative difficulties of the different neutrino and interaction types, it is interesting to see that there is some shift in which particles are relatively difficult/easy for the classifiers between track- and shower-like data. This pattern seems linked to the true signatures of the interactions. In the track-like data, its seems that it is easier to distinguish events that are true track events from muon events. These are $\nu_\mu(\overline{\nu}_\mu)$ and in a percentage of cases $\nu_\tau(\overline{\nu}_\tau)$ CC events. Likewise, on shower-like data the events that are true shower events are easier to distinguish from muon events. These are all NC, $\nu_e(\overline{\nu}_e)$ CC and (most) $\nu_\tau(\overline{\nu}_\tau)$ CC interaction events. This could be because true shower events that end up in the track selection are likely poorly defined and reconstructed events. The same holds for true track events in the shower selection. This shows that the classifiers in general have the most difficulty with poorly defined events, which coincides with our hypothesis that the muons that pass the selection criteria must inherently be poorly defined and reconstructed. This also further supports the claim that the training data selection procedure followed in this work has benefit.

Even taking into account the mechanism described above, the high performance on $\nu_\tau(\overline{\nu}_\tau)$ events in both track- and shower-like data selections is remarkable. A hypothesis would be that this is because of a direct correlation between classification difficulty and energy. Higher energy events give produce more light, and thus it is likely their signatures are better defined. Because of the high rest mass of τ particles, the energy threshold for their interactions is much higher. However as shown in figure 8.1, there does not seem to be a clear correlation between neutrino energy and the muonscore predicted by the tuned XGBC model.

In conclusion, this work has analysed some of the most important features in classifying muons and neutrinos in track and shower selection samples. It has attempted to explain why these features are important



(a) ***(b)***

Figure 8.1: *2-dimensional distribution plots of the muonscore produced by the tuned XGBC models on the (a) track-like and (b) shower-like test data sets and the true energy for neutrinos.*

in the decision process. After that, we have shown the performances of different models. From these results, it is clear that a tuned XGBC model combined with our training data selection procedure greatly outperforms the model and data selection procedure currently used. This shows promise in improving the quality of fit for the oscillation parameters to ORCA data.

Chapter 9

Outlook

The results in this work provide many promising avenues for further research. A first step could be to exactly follow the training data selection and sampling procedure as followed for the 'old' score. This would allow for a better analysis of the exact effect of the different classification algorithms. Performing an ablation study, one could even pinpoint which parts of the full algorithm are have the most impact on model performance.

In the same light, it would be interesting to also analyse the performance of a HGBC model, both base and tuned versions. By comparing these results to that of the XGBC, more information on the significance of certain algorithm components could be gained. It would also allow for a more in depth analysis of the effect of hyperparameter tuning.

Additionally, a more in depth analysis of the impact of the amount of features provided to the model and which features to include, could lead to a more well defined and optimized set of features to use use for classification purposes.

There are specific aspects of the 'old' training data selection procedure that were likely chosen because they were found to be empirically better. Adding components of the old procedure to the selection procedure used for this work, such as using using only $\nu_\mu(\overline{\nu}_\mu)$ and $\nu_e(\overline{\nu}_e)$ CC events to represent the neutrino class, could further improve performance on the shower-like events.

The XGBC model allows the user to assign individual weight to events in the training dataset. This would remove the need to use stratification of any kind to balance the training and test data, and may also improve performance.

Building on the strength of the XGBC models shown in this work, they

could be applied for classification of neutrino flavours. Or in the classification of neutrinos and muons in the downgoing signal. Classification of downgoing neutrinos may lead to improved values for the expected neutrino fluxes. Combined with improved overall neutrino classification leading to a stronger neutrino signal, this would improve the fit of the oscillation parameters.

For further analysis of model performance, the performances could be determined on data in different energy and zenith bins. As the data is also binned in this way for the oscillation parameter fits, this would provide further insight into the quality of the neutrino signal relevant to the fit.

In terms of data analysis, it would be interesting to look at the distributions of important features per particle and interaction type. This would allow further and more detailed analysis of why certain neutrino types are harder to separate from muons than others.

In this work, the track and noise scores have been passed to the BDTs as features. For track classification, the track score was a strongly influential feature. By removing this score, one could analyse the impact of stacking the results from sequential classifiers that focus on different properties in the data.

Version of October 11, 2024– Created October 11, 2024 - 09:35

Appendix A

Data Pre-processing

A.1 PID_2_DST

To extract the features used as input for the machine learning models in this work, an adapted version of the PID_2_DST git code [30] was used. The changes to the code added a larger amount of features to extract from the dst files, to attempt to get as much information as possible for the classification models. In this process, two standard ORCA pre-cuts on feature values were made. These cuts are based on the likelyhood with which the events are reconstructed by the track reconstruction algorithm, and the amount of PMT hits that were triggered by the event. The cuts then are: JG_lik>40 & N_trig_hit $\geq$ 15. These pre-cuts are in place to remove events that are very poorly defined, which would otherwise increase memory usage without being useful for eventual analyses. This also already reduces the optical noise to negligible levels [18].

A.2 Handling Duplicates

Due to the way the features are extracted in this process, attempting to get the maximum amount of features, there may be duplicate features in the data. This is due to the different naming conventions that are present in the feature extraction file. Therefore a check is performed to match features that have exactly the same values. A list of the different names is composed. The feature with the preferred alias is then kept, while the other features are removed.

A.3 Handling NaNs

There are different ways of handling missing values or *NaNs* in machine learning applications. One option is to impute the missing values. This means that the missing values are set to a certain value based on the rest of the data. This can mean being set to the min/max/mean of the column, or being predicted by a regressor based on the rest of the features. Imputing can be solid option if the amount of features with NaN values is relatively small. However even in such a case, the reason for the NaN in the data can still make imputing a good or bad choice.

Because of limited time during this project, a more straightforward approach was chosen. NaNs were handled by either dropping columns with 1% or more of missing data or by removing rows with NaNs. Note that in hindsight this step was not necessary, as the machine learning models used in this work all have built in NaN support.

A.4 Handling Infinite Values

In handling (negative) infinite values, we set the value of that feature to times 1.5 the (min) max value of the feature in the full dataset. If the lowest value of a feature with negative infinite values is positive, the infinite value is set to zero. If the highest value of a feature with positive infinite values is negative, we also set the infinite values to 0.

A.5 Feature Renaming

Many of the feature names contain characters such as ”[”, ”]” and ”:” which are not compatible with windows file names. Therefore these characters were removed from the name or replaced by either ”_” or ”,” for ease of transferring figures with the feature names in their name to windows computers.

Version of October 11, 2024– Created October 11, 2024 - 09:35

Appendix B

Feature Importance

Rank	Feature	Importance
1	T.feat_Neutrino2020.cherCond_n_hits_dnMup	0.194
2	T.feat_Neutrino2020.QupOvernHits	0.071
3	T.feat_Neutrino2020.cherCond_n_hits_trig_dnMup	0.056
4	T.feat_Neutrino2020.cherCond_hits_meanZposition	0.045
5	T.sum_jppshower.prefit_posfit_distance	0.042
6	pos_r_JGandalf	0.035
7	closest_,1,1	0.021
8	T.feat_Neutrino2020.n_hits_earlyTrig	0.021
9	T.feat_Neutrino2020.QupMinusQdn	0.020
10	T.feat_Neutrino2020.cherCond_hits_trig_meanZposition	0.019
11	crkv_nhits100_,1,1	0.018
12	crkv_nhits20_,1,0	0.016
13	furthest_,0,1	0.015
14	meanZhitTrig	0.014
15	crkv_nhits50_,1,1	0.013
16	T.feat_Neutrino2020.cherCond_n_hits_dnf	0.012
17	crkv_nhits50_,1,0	0.011
18	T.sum_jppshower.ratio_prefit_fits_near_best	0.011
19	cos_zenith_recoJShower	0.011
20	pos_z_recoJShower	0.010
21	ratio_both_tot_jg	0.01
22	maximumToT_triggerHit	0.01
23	T.feat_Neutrino2020.cherCond_n_doms	0.009
24	noisescore	0.009

Version of October 11, 2024– Created October 11, 2024 - 09:35

25	closest_,0,0	0.009
26	T.feat_Neutrino2020.dClosestApproach	0.009
27	T.feat_Neutrino2020.cherCond_n_hits_trig_upf	0.008
28	T.feat_Neutrino2020.zClosestApproach	0.007
29	crkv_nhits20_,1,2	0.007
30	T.feat_Neutrino2020.gandalf_Qdn	0.007
31	T.feat_Neutrino2020.cherCond_n_doms_trig	0.006
32	dist_JSH_JG_track	0.006
33	T.sum_hits.nlines	0.006
34	T.sum_trig_hits.atot	0.005
35	trigger_mask	0.005
36	energy_recoJEnergy	0.005
37	T.sum_jppshower.mean_tres_selected_hits	0.005
38	pos_z_recoJGandalf	0.005
39	crkv_nhits100_,1,0	0.005
40	closest_,0,1	0.005
41	furthest_,1,0	0.005
42	gandalf_shfit_lik_ratio	0.005
43	T.sum_jppshower.prefit_posfit_dt	0.005
44	E.trks.fitinf_,0,9	0.004
45	ratio_both_100_jg	0.004
46	min_dom_dist	0.004
47	crkv_nhits50_,0,1	0.004
48	sumtot_,0,0	0.004
49	pos_x_recoJGandalf	0.004
50	E.trks.fitinf_,0,2	0.004
51	T.feat_Neutrino2020.cherCond_n_hits_upf	0.004
52	nTriggerHits	0.003
53	energy_recoJShower	0.003
54	ratio_E_jshf_gandalf	0.003
55	furthest_,1,1	0.003
56	closest_,1,2	0.003
57	crkv_nhits100_,0,1	0.003
58	gandalf_shfit_lik_difference	0.003
59	cos_zenith_recoJGandalf	0.003
60	likelihood_JGandalf	0.003
61	pos_y_recoJShower	0.003
62	E.trks.fitinf_,0,5	0.003
63	T.sum_trig_hits.nlines	0.003
64	T.sum_hits.nhits	0.003
65	energy_recoTracklength	0.003

Version of October 11, 2024– Created October 11, 2024 - 09:35

66	crkv_nhits100_,1,2	0.003
67	loglik_jsh	0.003
68	dist_crkv_both_jsh	0.003
69	crkv_nhits_,0,1	0.003
70	log_distance_shfit_gandalf_for_shfit_time	0.003
71	angle_shfit_gandalf	0.003
72	T.sum_trig_hits.ndoms	0.003
73	crkv_nhits_,1,0	0.003
74	sumtot_,0,1	0.003
75	crkv_nhits100_,0,0	0.003
76	crkv_nhits20_,0,0	0.003
77	T.feat_Neutrino2020.cherCond_n_hits_trig_dnf	0.002
78	dist_crkv_both_jg	0.002
79	E.trks.fitinf_,0,6	0.002
80	pos_x_recoJShower	0.002
81	crkv_nhits_,1,2	0.002
82	sumtot_,1,0	0.002
83	crkv_nhits50_,0,0	0.002
84	furthest_,1,2	0.002
85	E.trks.fitinf_,0,16	0.002
86	sumtot_,1,2	0.002
87	E.trks.dir.y_,0	0.002
88	sumtot_,0,2	0.002
89	dt_shfit_gandalf	0.002
90	E.trks.fitinf_,0,14	0.002
91	E.trks.lik_,1	0.002
92	crkv_nhits_,0,0	0.002
93	distance_shfit_gandalf	0.002
94	dist_JG_Jsh_track	0.002
95	sumtot_,1,1	0.002
96	furthest_,0,0	0.002
97	furthest_,0,2	0.002
98	crkv_nhits_,1,1	0.002
99	pos_y_recoJGandalf	0.002
100	shortest_4D_dist_jg_jsh	0.002
101	E.trks.fitinf_,0,7	0.002
102	E.trks.fitinf_,0,15	0.002
103	T.sum_jppshower.n_selected_hits	0.002
104	crkv_nhits50_,1,2	0.002
105	E.trks.fitinf_,0,1	0.002
106	beta0	0.002

107	crkv_nhits100_,0,2	0.002
108	E.trks.dir.x_,1	0.002
109	T.feat_Neutrino2020.gandalf_nHits	0.002
110	crkv_nhits_,0,2	0.002
111	T.sum_hits.ndoms	0.002
112	E.trks.dir.x_,0	0.002
113	E.trks.dir.y_,1	0.002
114	crkv_nhits200_,0,1	0.002
115	E.trks.t_,0	0.002
116	closest_,0,2	0.002
117	crkv_nhits200_,0,0	0.001
118	trackscore	0.001
119	T.sum_trig_hits.tmax	0.001
120	crkv_nhits200_,1,1	0.001
121	crkv_nhits50_,0,2	0.001
122	crkv_nhits200_,1,0	0.001
123	crkv_nhits200_,0,2	0.001
124	E.trks.t_,1	0.000
125	T.feat_Neutrino2020.gandalf_Qup	0.
126	T.sum_trig_hits.tmin	0.
127	E.trks.fitinf_,0,13	0.
128	crkv_nhits200_,1,2	0.
129	crkv_nhits20_,0,1	0.
130	crkv_nhits20_,0,2	0.
131	crkv_nhits20_,1,1	0.
132	loglik_jg	0.

Table B.1: *Feature ranking for the shower-like data*

Rank	Feature	Importance
1	cos_zenith_recoJShower	0.464
2	T.feat_Neutrino2020.cherCond_n_hits_dnMup	0.098
3	trackscore	0.078
4	T.feat_Neutrino2020.QupOvernHits	0.03
5	angle_shfit_gandalf	0.025
6	crkv_nhits100_,0,0	0.014
7	T.feat_Neutrino2020.cherCond_hits_meanZposition	0.014
8	T.feat_Neutrino2020.cherCond_n_hits_dnf	0.014
9	closest_,1,1	0.011

10	T.feat_Neutrino2020.cherCond_hits_trig_meanZposition	0.010
11	pos_r_JGandalf	0.009
12	sumtot_,0,0	0.009
13	crkv_nhits50_,1,1	0.009
14	T.feat_Neutrino2020.cherCond_n_doms	0.007
15	T.feat_Neutrino2020.n_hits_earlyTrig	0.007
16	crkv_nhits50_,0,1	0.006
17	ratio_both_tot_jg	0.006
18	T.sum_jppshower.prefit_posfit_distance	0.006
19	dt_shfit_gandalf	0.005
20	crkv_nhits50_,1,0	0.005
21	T.feat_Neutrino2020.dClosestApproach	0.004
22	closest_,0,0	0.004
23	E.trks.fitinf_,0,2	0.004
24	meanZhitTrig	0.004
25	pos_z_recoJShower	0.004
26	T.feat_Neutrino2020.zClosestApproach	0.004
27	T.feat_Neutrino2020.QupMinusQdn	0.003
28	crkv_nhits20_,1,2	0.003
29	noisescore	0.003
30	maximumToT_triggerHit	0.003
31	crkv_nhits_,1,1	0.003
32	T.feat_Neutrino2020.cherCond_n_hits_trig_upf	0.003
33	furthest_,1,1	0.003
34	likelihood_JGandalf	0.003
35	furthest_,1,0	0.003
36	sumtot_,0,1	0.003
37	crkv_nhits100_,0,2	0.003
38	furthest_,1,2	0.003
39	pos_z_recoJGandalf	0.003
40	T.sum_hits.nlines	0.003
41	cos_zenith_recoJGandalf	0.003
42	distance_shfit_gandalf	0.002
43	T.sum_jppshower.ratio_prefit_fits_near_best	0.002
44	crkv_nhits100_,1,0	0.002
45	crkv_nhits_,1,2	0.002
46	gandalf_shfit_lik_difference	0.002
47	energy_recoJEnergy	0.002
48	dist_crkv_both_jg	0.002
49	crkv_nhits100_,1,2	0.002
50	T.sum_trig_hits.atot	0.002

51	dist_JSH_JG_track	0.002
52	closest_,1,2	0.002
53	T.feat_Neutrino2020.cherCond_n_hits_trig_dnMup	0.002
54	T.feat_Neutrino2020.cherCond_n_doms_trig	0.002
55	T.sum_hits.nhits	0.002
56	pos_x_recoJGandalf	0.002
57	sumtot_,1,2	0.002
58	furthest_,0,1	0.002
59	crkv_nhits20_,1,0	0.002
60	nTriggerHits	0.002
61	E.trks.fitinf_,0,5	0.002
62	loglik_jsh	0.001
63	E.trks.fitinf_,0,9	0.001
64	crkv_nhits_,0,2	0.001
65	crkv_nhits_,1,0	0.001
66	crkv_nhits100_,1,1	0.001
67	trigger_mask	0.001
68	shortest_4D_dist_jg_jsh	0.001
69	min_dom_dist	0.001
70	sumtot_,1,1	0.001
71	sumtot_,1,0	0.001
72	loglik_jg	0.001
73	sumtot_,0,2	0.001
74	ratio_E_jshf_gandalf	0.001
75	T.sum_trig_hits.nlines	0.001
76	T.sum_jppshower.mean_tres_selected_hits	0.001
77	energy_recoJShower	0.001
78	energy_recoTracklength	0.001
79	E.trks.dir.x_,0	0.001
80	E.trks.fitinf_,0,7	0.001
81	gandalf_shfit_lik_ratio	0.001
82	T.sum_trig_hits.ndoms	0.001
83	crkv_nhits_,0,1	0.001
84	E.trks.fitinf_,0,16	0.001
85	T.feat_Neutrino2020.gandalf_Qdn	0.001
86	E.trks.t_,1	0.001
87	ratio_both_100_jg	0.001
88	pos_y_recoJShower	0.001
89	crkv_nhits_,0,0	0.001
90	crkv_nhits50_,0,0	0.001
91	dist_crkv_both_jsh	0.001

92	closest_,0,1	0.001
93	E.trks.lik_,1	0.001
94	E.trks.fitinf_,0,14	0.001
95	furthest_,0,0	0.001
96	pos_x_recoJShower	0.001
97	T.sum_jppshower.prefit_posfit_dt	0.001
98	T.feat_Neutrino2020.gandalf_nHits	0.001
99	crkv_nhits50_,1,2	0.001
100	dist_JG_Jsh_track	0.001
101	log_distance_shfit_gandalf_for_shfit_time	0.001
102	E.trks.fitinf_,0,15	0.001
103	E.trks.fitinf_,0,6	0.001
104	crkv_nhits20_,0,0	0.001
105	T.feat_Neutrino2020.cherCond_n_hits_upf	0.001
106	E.trks.dir.y_,0	0.001
107	T.feat_Neutrino2020.cherCond_n_hits_trig_dnf	0.001
108	crkv_nhits200_,0,1	0.001
109	crkv_nhits100_,0,1	0.001
110	furthest_,0,2	0.001
111	pos_y_recoJGandalf	0.001
112	beta0	0.001
113	E.trks.dir.y_,1	0.001
114	closest_,0,2	0.001
115	T.sum_hits.ndoms	0.001
116	E.trks.fitinf_,0,1	0.001
117	E.trks.dir.x_,1	0.001
118	T.sum_jppshower.n_selected_hits	0.001
119	E.trks.t_,0	0.001
120	crkv_nhits50_,0,2	0.001
121	T.sum_trig_hits.tmax	0.000
122	crkv_nhits200_,1,2	0.000
123	crkv_nhits20_,1,1	0.000
124	crkv_nhits200_,0,0	0.000
125	crkv_nhits20_,0,1	0.000
126	crkv_nhits200_,0,2	0.
127	crkv_nhits200_,1,0	0.
128	E.trks.fitinf_,0,13	0.
129	crkv_nhits200_,1,1	0.
130	crkv_nhits20_,0,2	0.
131	T.sum_trig_hits.tmin	0.
132	T.feat_Neutrino2020.gandalf_Qup	0.

Table B.2: *Feature ranking for the track-like data*

Appendix C

Hyperparameters

The hyperparameters and their ranges used for optimizing the XGBC model are:

- eta: float in range [0.1, 1.0]
- gamma: float in range [0, 0.5]
- max_depth: int in range [3, 15]
- min_child_weight: int in range [1, 10]
- subsample: trial in range [0.3, 1]
- colsample_bytree: float in range [0.2, 1]
- colsample_bylevel: float in range [0.2, 1]
- colsample_bynode: float in range [0.2, 1]
- reg_lambda: [0, 2.5]
- alpha: [0, 2.5]
- scale_pos_weight: float in range [0.2, 5]
- n_features: int in range [5, 110] with steps of 1
- grow_policy: choice [”depthwise”, ”lossguide”]

The hyperparameter values for the tuned XGBC model on the shower data are:

- eta = 0.179366696215447
- gamma = 0.07464467596938863
- max_depth = 15
- min_child_weight = 8
- subsample = 0.8593924117067859
- colsample_bytree = 0.8787332316250065
- colsample_bylevel = 0.8907988230784997
- colsample_bynode = 0.9352157162619897
- reg_lambda = 0.3240813905157929
- alpha = 0.1181044009985062
- scale_pos_weight = 1.9858829960470836
- n_features = 74
- grow_policy = depthwise

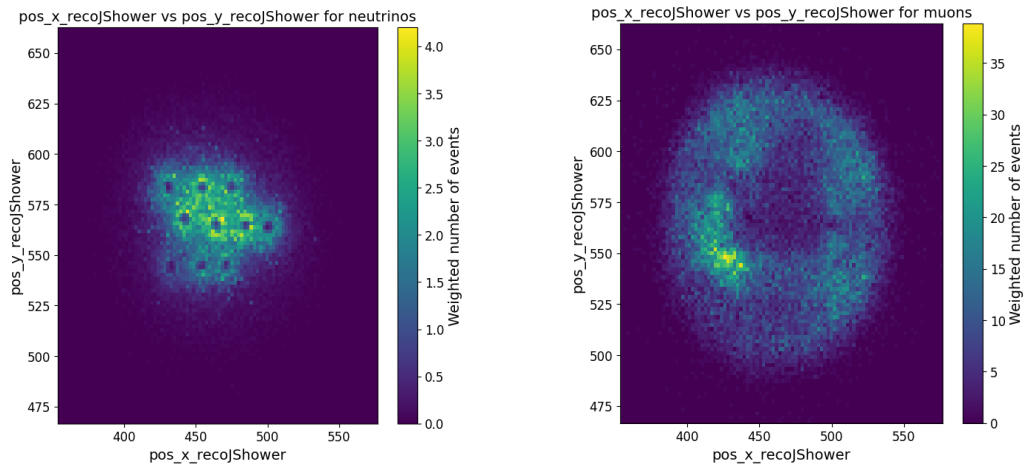
The hyperparameter values for the tuned XGBC model on the track data are:

- eta = 0.18002961104233298
- gamma = 0.32204618525901385
- max_depth = 11
- min_child_weight = 8
- subsample = 0.9953936477317626
- colsample_bytree = 0.884793119891182
- colsample_bylevel = 0.8461815581023293
- colsample_bynode = 0.5148570860053796
- reg_lambda = 0.6960181424058038
- alpha = 2.229285898333837

- scale_pos_weight = 0.6606133472265183
- n_features = 99
- grow_policy = depthwise

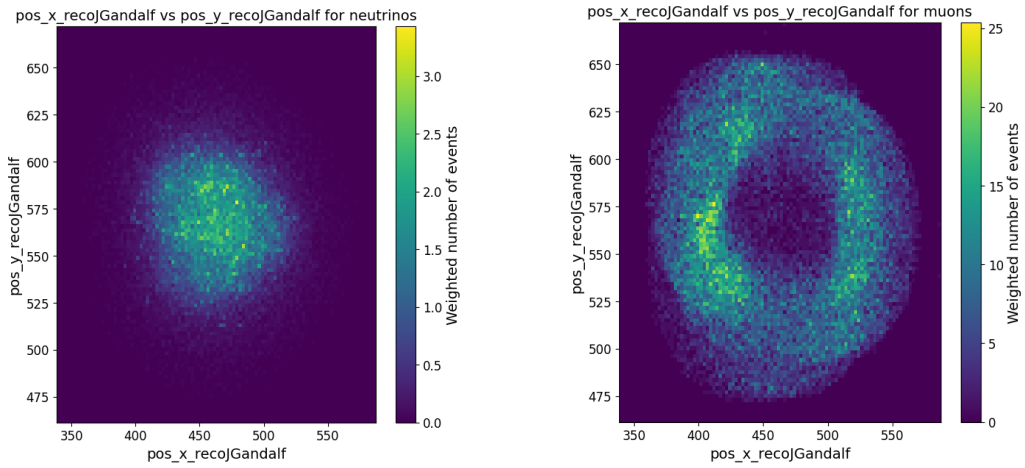
Appendix D

Reconstructed Vertices



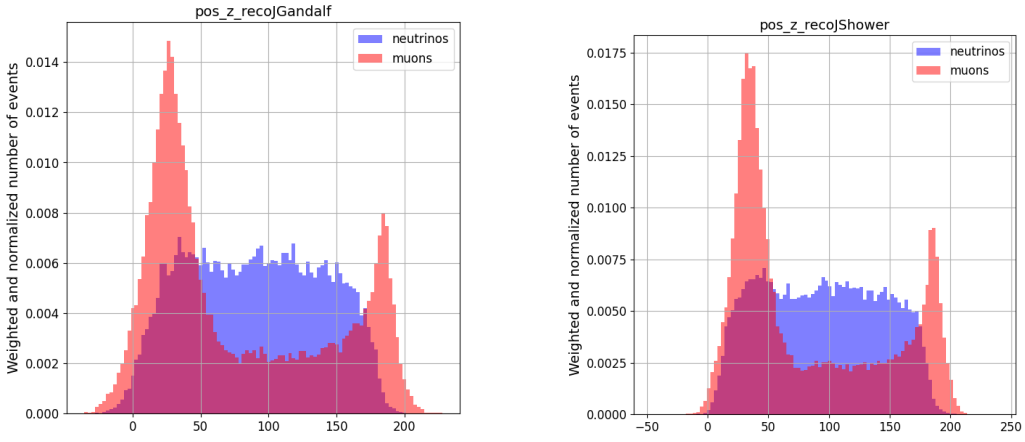
(a) ***(b)***

Figure D.1: *2d-histograms of the x- and y-positions of the events as reconstructed by the shower algorithm for shower-like data.*



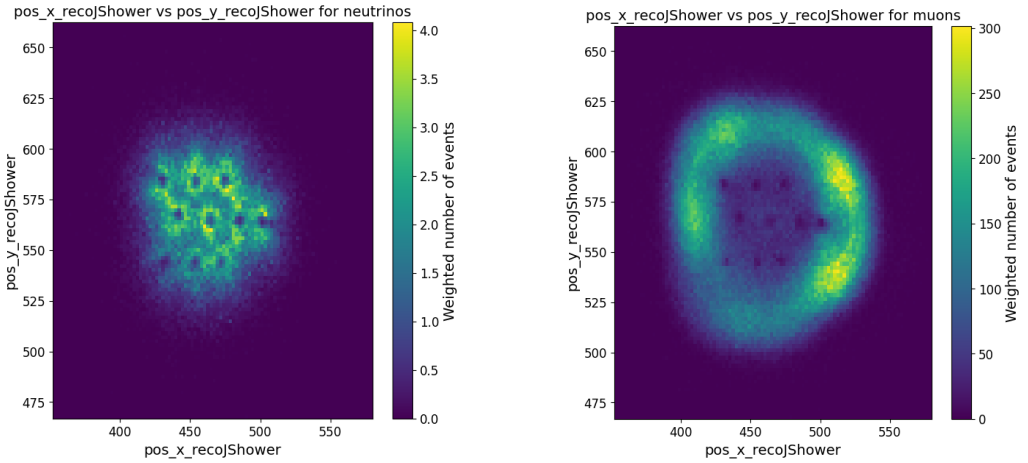
(a) ***(b)***

Figure D.2: *2d-histograms of the x- and y-positions of the events as reconstructed by the track algorithm for shower-like data.*



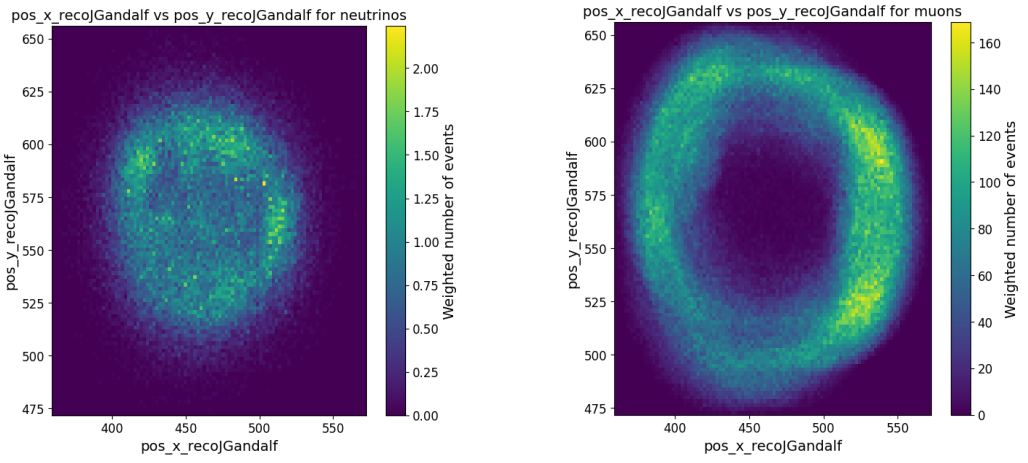
(a) ***(b)***

Figure D.3: *Histograms of the z-position of the events as reconstructed by the track and shower algorithms for shower-like data.*



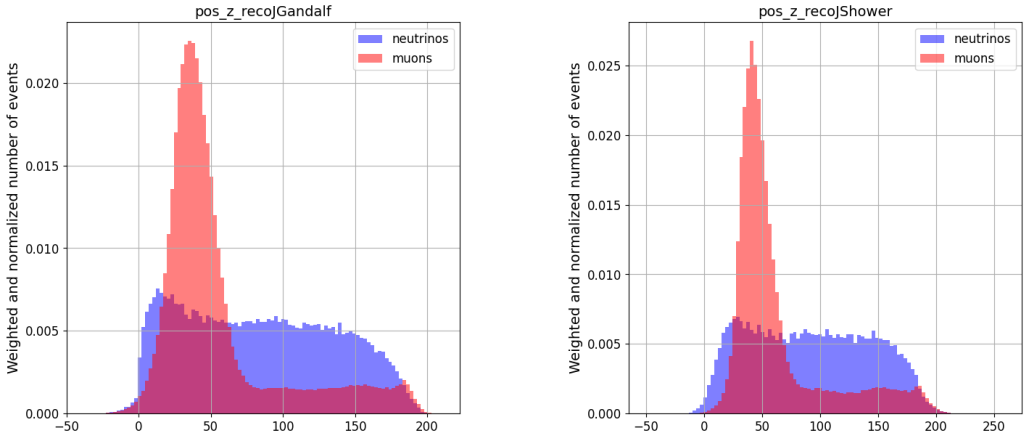
(a) ***(b)***

Figure D.4: *2d-histograms of the x- and y-positions of the events as reconstructed by the shower algorithm for track-like data.*



(a) ***(b)***

Figure D.5: *2d-histograms of the x- and y-positions of the events as reconstructed by the track algorithm for track-like data.*



(a) ***(b)***

Figure D.6: *Histograms of the z-position of the events as reconstructed by the track and shower algorithms for track-like data.*

Bibliography

[1] . URL https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html.

[2] . URL https://scikit-learn.org/stable/modules/generated/sklearn.metrics.log_loss.html.

[3] Parameters - xgboost 2.1.1 documentation. URL https://xgboost.readthedocs.io/en/stable/parameter.html#additional-parameters-for-dart-booster-booster-dart.

[4] Neutrino oscillations/oscillation analysis orca6/10/11. URL https://wiki.km3net.de/index.php/Neutrino_Oscillations/Oscillation_Analysis_ORCA6/10/11.

[5] Nov 2023. URL https://www.km3net.org/research/physics/astronomy-with-arca/.

[6] Sep 2024. URL https://www.km3net.org/research/physics/particle-physics-with-orca/.

[7] S Adrián-Martínez, M Ageron, F Aharonian, S Aiello, A Albert, F Ameli, E Anassontzis, M Andre, G Androulakis, M Anghinolfi, G Anton, M Ardid, T Avgitas, G Barbarino, E Barbarito, B Baret, J Barrios-Martí, B Belhorma, A Belias, E Berbee, A van den Berg, V Bertin, S Beurthey, V van Beveren, N Beverini, S Biagi, A Biagioni, M Billault, M Bondì, R Bormuth, B Bouhadef, G Bourlis, S Bourret, C Boutonnet, M Bouwhuis, C Bozza, R Bruijn, J Brunner, E Buis, J Busto, G Cacopardo, L Caillat, M Calamai, D Calvo, A Capone, L Caramete, S Cecchini, S Celli, C Champion, R Cherkaoui El Moursli, S Cherubini, T Chiarusi, M Circella, L Classen, R Cocimano, J A B

Coelho, A Coleiro, S Colonges, R Coniglione, M Cordelli, A Cosquer, P Coyle, A Creusot, G Cuttone, A DâAmico, G De Bonis, G De Rosa, C De Sio, F Di Capua, I Di Palma, A F Díaz García, C Distefano, C Donzaud, D Dornic, Q Dorosti-Hasankiadeh, E Drakopoulou, D Drouhin, L Drury, M Durocher, T Eberl, S Eichie, D van Eijk, I El Bojaddaini, N El Khayati, D Elsaesser, A Enzenhöfer, F Fassi, P Favali, P Fermani, G Ferrara, C Filippidis, G Frascadore, L A Fusco, T Gal, S Galatà, F Garufi, P Gay, M Gebyehu, V Giordano, N Gizani, R Gracia, K Graf, T Grégoire, G Grella, R Habel, S Hallmann, H van Haren, S Harissopulos, T Heid, A Heijboer, E Heine, S Henry, J J Hernández-Rey, M Hevinga, J Hofestädt, C M F Hugon, G Illuminati, C W James, P Jansweijer, M Jongen, M de Jong, M Kadler, O Kalekin, A Kappes, U F Katz, P Keller, G Kieft, D KieÃling, E N Koffeman, P Kooijman, A Kouchner, V Kulikovskiy, R Lahmann, P Lamare, A Leisos, E Leonora, M Lindsey Clark, A Liolios, C D Llorens Alvarez, D Lo Presti, H Löhner, A Lonardo, M Lotze, S Loucatos, E Maccioni, K Mannheim, A Margiotta, A Marinelli, O MariÅ, C Markou, J A Martínez-Mora, A Martini, R Mele, K W Melis, T Michael, P Migliozzi, E Migneco, P Mijakowski, A Miraglia, C M Mollo, M Mongelli, M Morganti, A Moussa, P Musico, M Musumeci, S Navas, C A Nicolau, I Olcina, C Olivetto, A Orlando, A Papaikonomou, R Papaleo, G E PÄvÄlaÅ, H Peek, C Pellegrino, C Perrina, M Pfutzner, P Piattelli, K Pikounis, G E Poma, V Popa, T Pradier, F Pratolongo, G PÃŒhlhofer, S Pulvirenti, L Quinn, C Racca, F Raffaelli, N Randazzo, P Rapidis, P Razis, D Real, L Resvanis, J Reubelt, G Riccobene, C Rossi, A Rovelli, M Salda na, I Salvadori, D F E Samtleben, A Sánchez García, A Sánchez Losa, M Sanguineti, A Santangelo, D Santonocito, P Sapienza, F Schimmel, J Schmelling, V Sciacca, M Sedita, T Seitz, I Sgura, F Simeone, I Siotis, V Sipala, B Spisso, M Spurio, G Stavropoulos, J Steijger, S M Stellacci, D Stransky, M Taiuti, Y Tayalati, D Tézier, S Theraube, L Thompson, P Timmer, C Tönnis, L Trasatti, A Trovato, A Tsirigotis, S Tzamarias, E Tzamariudaki, B Vallage, V Van Elewyck, J Vermeulen, P Vicini, S Viola, D Vivolo, M Volkert, G Voulgaris, L Wiggers, J Wilms, E de Wolf, K Zachariadou, J D Zornoza, and J Zú niga. Letter of intent for km3net 2.0. *Journal of Physics G: Nuclear and Particle Physics*, 43(8):084001, jun 2016. doi: 10.1088/0954-3899/43/8/084001. URL `https://dx.doi.org/10.1088/0954-3899/43/8/084001`.

[8] Sea Agostinelli, John Allison, K al Amako, John Apostolakis, Henrique Araujo, Pedro Arce, Makoto Asai, D Axen, Swagato Banerjee,

GJNI Barrand, et al. Geant4âa simulation toolkit. *Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 506(3):250–303, 2003.

[9] S. Aiello, A. Albert, S. Alves Garre, Z. Aly, F. Ameli, M. Andre, G. Androulakis, M. Anghinolfi, M. Anguita, G. Anton, M. Ardid, J. Aublin, C. Bagatelas, G. Barbarino, B. Baret, S. Basegmez du Pree, M. Bendahman, E. Berbee, A.M. van den Berg, V. Bertin, S. Biagi, A. Biagioni, M. Bissinger, M. Boettcher, J. Boumaaza, M. Bouta, M. Bouwhuis, C. Bozza, H. BrÃ¢nzaÅ, M. Bruchner, R. Bruijn, J. Brunner, E. Buis, R. Buompane, J. Busto, D. Calvo, A. Capone, V. Carretero, P. Castaldi, S. Celli, M. Chabab, N. Chau, A. Chen, S. Cherubini, V. Chiarella, T. Chiarusi, M. Circella, R. Cocimano, J.A.B. Coelho, A. Coleiro, M. Colomer Molla, R. Coniglione, I. Corredoira, P. Coyle, A. Creusot, G. Cuttone, A. DâOnofrio, R. Dallier, M. De Palma, I. Di Palma, A.F. Díaz, D. Diego-Tortosa, C. Distefano, A. Domi, R. Donà, C. Donzaud, D. Dornic, M. Dörr, D. Drouhin, M. Durocher, T. Eberl, D. van Eijk, I. El Bojaddaini, D. Elsaesser, A. Enzenhöfer, V. Espinosa Roselló, P. Fermani, G. Ferrara, M.D. Filipović, F. Filippini, A. Franco, L.A. Fusco, O. Gabella, T. Gal, A. Garcia Soto, F. Garufi, Y. Gatelet, N. GeiÃelbrecht, L. Gialanella, E. Giorgio, S.R. Gozzini, R. Gracia, K. Graf, D. Grasso, G. Grella, D. Guderian, C. Guidi, S. Hallmann, H. Hamdaoui, H. van Haren, A. Heijboer, A. Hekalo, J.J. Hernández-Rey, J. Hofestädt, F. Huang, W. Idrissi Ibnsalih, G. Illuminati, C.W. James, M. de Jong, P. de Jong, B.J. Jung, M. Kadler, P. Kalaczyński, O. Kalekin, U.F. Katz, N.R. Khan Chowdhury, F. van der Knaap, E.N. Koffeman, P. Kooijman, A. Kouchner, M. Kreter, V. Kulikovskiy, R. Lahmann, G. Larosa, R. Le Breton, O. Leonardi, F. Leone, E. Leonora, G. Levi, M. Lincetto, M. Lindsey Clark, T. Lipreau, A. Lonardo, F. Longhitano, D. Lopez-Coto, L. Maderer, J. Mańczak, K. Mannheim, A. Margiotta, A. Marinelli, C. Markou, L. Martin, J.A. Martínez-Mora, A. Martini, F. Marzaioli, S. Mastroianni, S. Mazzou, K.W. Melis, G. Miele, P. Migliozzi, E. Migneco, P. Mijakowski, L.S. Miranda, Z. Modebadze, C.M. Mollo, M. Morganti, M. Moser, A. Moussa, R. Muller, M. Musumeci, L. Nauta, S. Navas, C.A. Nicolau, B. Ó Fearraigh, M. Organokov, A. Orlando, G. Papalashvili, R. Papaleo, C. Pastore, A.M. Paun, G.E. PÄvÄlaÅ, C. Pellegrino, M. Perrin-Terrin, P. Piattelli, C. Pieterse, K. Pikounis, O. Pisanti, C. Poirè, V. Popa, M. Post, T. Pradier, G. PÃŒhlhofer, S. Pulvirenti, L. Quinn, O. Rabyang, F. Raffaelli, N. Randazzo, A. Rapicavoli, S. Razzaque, D. Real, S. Reck, J. Reubelt, G. Riccobene, M. Richer,

S. Rivoire, A. Rovelli, F. Salesa Greus, D.F.E. Samtleben, A. Sánchez Losa, M. Sanguineti, A. Santangelo, D. Santonocito, P. Sapienza, J. Schnabel, V. Sciacca, J. Seneca, I. Sgura, R. Shanidze, A. Sharma, F. Simeone, A. Sinopoulou, B. Spisso, M. Spurio, D. Stavropoulos, J. Steijger, S.M. Stellacci, M. Taiuti, Y. Tayalati, E. Tenllado, T. Thakore, S. Tingay, E. Tzamariudaki, D. Tzanetatos, V. Van Elewyck, G. Vannoye, G. Vasileiadis, F. Versari, S. Viola, D. Vivolo, G. de Wasseige, J. Wilms, R. Wojaczyński, E. de Wolf, D. Zaborov, S. Zavatarelli, A. Zegarelli, J.D. Zornoza, J. Zúñiga, and N. Zywucka. gseagen: The km3net genie-based code for neutrino telescopes. *Computer Physics Communications*, 256:107477, November 2020. ISSN 0010-4655. doi: 10.1016/j.cpc.2020.107477. URL `http://dx.doi.org/10.1016/j.cpc.2020.107477`.

[10] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631, 2019.

[11] John N. Bahcall and H. A. Bethe. Solution of the solar-neutrino problem. *Phys. Rev. Lett.*, 65:2233–2235, Oct 1990. doi: 10.1103/PhysRevLett.65.2233. URL `https://link.aps.org/doi/10.1103/PhysRevLett.65.2233`.

[12] A. Boyarsky, M. Drewes, T. Lasserre, S. Mertens, and O. Ruchayskiy. Sterile neutrino dark matter. *Progress in Particle and Nuclear Physics*, 104:1â45, January 2019. ISSN 0146-6410. doi: 10.1016/j.ppnp.2018.07.004. URL `http://dx.doi.org/10.1016/j.ppnp.2018.07.004`.

[13] R. N. Cahn, D. A. Dwyer, S. J. Freedman, W. C. Haxton, R. W. Kadel, Yu. G. Kolomensky, K. B. Luk, P. McDonald, G. D. Orebi Gann, and A. W. P. Poon. White paper: Measuring the neutrino mass hierarchy, 2013. URL `https://arxiv.org/abs/1307.5487`.

[14] G. Carminati, M. Bazzotti, A. Margiotta, and M. Spurio. Atmospheric muons from parametric formulas: a fast generator for neutrino telescopes (mupage). *Computer Physics Communications*, 179(12):915â923, December 2008. ISSN 0010-4655. doi: 10.1016/j.cpc.2008.07.014. URL `http://dx.doi.org/10.1016/j.cpc.2008.07.014`.

[15] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Confer-*

ence on Knowledge Discovery and Data Mining, KDD â16. ACM, August 2016. doi: 10.1145/2939672.2939785. URL http://dx.doi.org/10.1145/2939672.2939785.

[16] D. B. Chitwood, T. I. Banks, M. J. Barnes, S. Battu, R. M. Carey, S. Cheekatmalla, S. M. Clayton, J. Crnkovic, K. M. Crowe, P. T. Debevec, S. Dhamija, W. Earle, A. Gafarov, K. Giovanetti, T. P. Gorringe, F. E. Gray, M. Hance, D. W. Hertzog, M. F. Hare, P. Kammel, B. Kiburg, J. Kunkle, B. Lauss, I. Logashenko, K. R. Lynch, R. McNabb, J. P. Miller, F. Mulhauser, C. J. G. Onderwater, C. S. Özben, Q. Peng, C. C. Polly, S. Rath, B. L. Roberts, V. Tishchenko, G. D. Wait, J. Wasserman, D. M. Webber, P. Winter, and P. A. Żołnierczuk. Improved measurement of the positive-muon lifetime and determination of the fermi constant. *Phys. Rev. Lett.*, 99:032001, Jul 2007. doi: 10.1103/PhysRevLett.99.032001. URL https://link.aps.org/doi/10.1103/PhysRevLett.99.032001.

[17] KM3NeT Collaboration. The km3net multi-pmt optical module, 2022. URL https://arxiv.org/abs/2203.10048.

[18] KM3NeT Collaboration, S. Aiello, A. Albert, A. R. Alhebsi, M. Alshamsi, S. Alves Garre, A. Ambrosone, F. Ameli, M. Andre, E. Androutsou, L. Aphecetche, M. Ardid, S. Ardid, H. Atmani, J. Aublin, F. Badaracco, L. Bailly-Salins, Z. BardaÄovÃ¡, B. Baret, A. Bariego-Quintana, Y. Becherini, M. Bendahman, F. Benfenati, M. Benhassi, M. Bennani, D. M. Benoit, E. Berbee, V. Bertin, C. Beyer, S. Biagi, M. Boettcher, D. Bonanno, J. Boumaaza, M. Bouta, M. Bouwhuis, C. Bozza, R. M. Bozza, H. Brânzas, F. Bretaudeau, M. Breuhaus, R. Bruijn, J. Brunner, R. Bruno, E. Buis, R. Buompane, J. Busto, B. Caiffi, D. Calvo, A. Capone, F. Carenini, V. Carretero, T. Cartraud, P. Castaldi, V. Cecchini, S. Celli, L. Cerisy, M. Chabab, M. Chadolias, A. Chen, S. Cherubini, T. Chiarusi, M. Circella, R. Cocimano, J. A. B. Coelho, A. Coleiro, A. Condorelli, R. Coniglione, P. Coyle, A. Creusot, G. Cuttone, R. Dallier, Y. Darras, A. De Benedittis, B. De Martino, G. De Wasseige, V. Decoene, I. Del Rosso, L. S. Di Mauro, I. Di Palma, A. F. Díaz, D. Diego-Tortosa, C. Distefano, A. Domi, C. Donzaud, D. Dornic, E. Drakopoulou, D. Drouhin, J. G. Ducoin, R. Dvornický, T. Eberl, E. EckerovÃ¡, A. Eddymaoui, T. van Eeden, M. Eff, D. van Eijk, I. El Bojaddaini, S. El Hedri, V. Ellajosyula, A. Enzenhöfer, G. Ferrara, M. D. Filipović, F. Filippini, D. Franciotti, L. A. Fusco, S. Gagliardini, T. Gal, J. García Méndez, A. Garcia Soto, C. Gatius Oliver, N. GeiÃelbrecht, E. Genton, H. Ghad-

dari, L. Gialanella, B. K. Gibson, E. Giorgio, I. Goos, P. Goswami, S. R. Gozzini, R. Gracia, C. Guidi, B. Guillon, M. GutiÃ©rrez, C. Haack, H. van Haren, A. Heijboer, L. Hennig, J. J. Hernández-Rey, W. Idrissi Ibnsalih, G. Illuminati, D. Joly, M. de Jong, P. de Jong, B. J. Jung, G. Kistauri, C. Kopper, A. Kouchner, Y. Y. Kovalev, V. Kueviakoe, V. Kulikovskiy, R. Kvatadze, M. Labalme, R. Lahmann, M. Lamoureux, G. Larosa, C. Lastoria, A. Lazo, S. Le Stum, G. Lehaut, V. Lemaítre, E. Leonora, N. Lessing, G. Levi, M. Lindsey Clark, F. Longhitano, F. Magnani, J. Majumdar, L. Malerba, F. Mamedov, J. MaÅczak, A. Manfreda, M. Marconi, A. Margiotta, A. Marinelli, C. Markou, L. Martin, M. Mastrodicasa, S. Mastroianni, J. Mauro, G. Miele, P. Migliozzi, E. Migneco, M. L. Mitsou, C. M. Mollo, L. Morales-Gallegos, G. Moretti, A. Moussa, I. Mozun Mateo, R. Muller, M. R. Musone, M. Musumeci, P. Napoli, S. Navas, A. Nayerhoda, C. A. Nicolau, B. Nkosi, B. Ó Fearraigh, V. Oliviero, A. Orlando, E. Oukacha, D. Paesani, J. Palacios González, G. Papalashvili, V. Parisi, E. J. Pastor Gomez, A. M. Paun, G. E. Pavalas, I. Pelegris, S. Pe na Martínez, M. Perrin-Terrin, V. Pestel, R. Pestes, P. Piattelli, A. Plavin, C. Poiré, V. Popa, T. Pradier, J. Prado, S. Pulvirenti, C. A. Quiroz-Rangel, U. Rahaman, N. Randazzo, S. Razzaque, I. C. Rea, D. Real, G. Riccobene, J. Robinson, A. Romanov, E. Ros, A. Saina, F. Salesa Greus, D. F. E. Samtleben, A. Sánchez Losa, S. Sanfilippo, M. Sanguineti, D. Santonocito, P. Sapienza, J. Schnabel, J. Schumann, H. M. Schutte, J. Seneca, N. Sennan, B. Setter, I. Sgura, R. Shanidze, A. Sharma, Y. Shitov, F. Simkovic, A. Simonelli, A. Sinopoulou, B. Spisso, M. Spurio, D. Stavropoulos, I. Stekl, S. M. Stellacci, M. Taiuti, Y. Tayalati, H. Thiersen, S. Thoudam, P. de la Torre, I. Tosta e Melo, E. Tragia, B. Trocmé, V. Tsourapis, A. Tudorache, E. Tzamariudaki, A. Ukleja, A. Vacheret, V. Valsecchi, V. Van Elewyck, G. Vannoye, G. Vasileiadis, F. Vazquez de Sola, A. Veutro, S. Viola, D. Vivolo, A. van Vliet, H. Warnhofer, S. Weissbrod, E. de Wolf, I. Yvon, G. Zarpapis, S. Zavatarelli, A. Zegarelli, D. Zito, J. D. Zornoza, J. Zú niga, and N. Zywucka. Measurement of neutrino oscillation parameters with the first six detection units of km3net/orca, 2024. URL `https://arxiv.org/abs/2408.07015`.

[19] Sacha Davidson, Enrico Nardi, and Yosef Nir. Leptogenesis. *Physics Reports*, 466(4â5):105â177, September 2008. ISSN 0370-1573. doi: 10.1016/j.physrep.2008.06.002. URL `http://dx.doi.org/10.1016/j.physrep.2008.06.002`.

[20] Raymond Davis. A review of the homestake solar neutrino experiment. *Progress in Particle and Nuclear Physics*, 32:13–32, 1994. ISSN 0146-6410. doi: https://doi.org/10.1016/0146-6410(94)90004-3. URL https://www.sciencedirect.com/science/article/pii/0146641094900043.

[21] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[22] Y. Fukuda, T. Hayakawa, E. Ichihara, K. Inoue, K. Ishihara, H. Ishino, Y. Itow, T. Kajita, J. Kameda, S. Kasuga, K. Kobayashi, Y. Kobayashi, Y. Koshio, M. Miura, M. Nakahata, S. Nakayama, A. Okada, K. Okumura, N. Sakurai, M. Shiozawa, Y. Suzuki, Y. Takeuchi, Y. Totsuka, S. Yamada, M. Earl, A. Habig, E. Kearns, M. D. Messier, K. Scholberg, J. L. Stone, L. R. Sulak, C. W. Walter, M. Goldhaber, T. Barszczxak, D. Casper, W. Gajewski, P. G. Halverson, J. Hsu, W. R. Kropp, L. R. Price, F. Reines, M. Smy, H. W. Sobel, M. R. Vagins, K. S. Ganezer, W. E. Keig, R. W. Ellsworth, S. Tasaka, J. W. Flanagan, A. Kibayashi, J. G. Learned, S. Matsuno, V. J. Stenger, D. Takemori, T. Ishii, J. Kanzaki, T. Kobayashi, S. Mine, K. Nakamura, K. Nishikawa, Y. Oyama, A. Sakai, M. Sakuda, O. Sasaki, S. Echigo, M. Kohama, A. T. Suzuki, T. J. Haines, E. Blaufuss, B. K. Kim, R. Sanford, R. Svoboda, M. L. Chen, Z. Conner, J. A. Goodman, G. W. Sullivan, J. Hill, C. K. Jung, K. Martens, C. Mauger, C. McGrew, E. Sharkey, B. Viren, C. Yanagisawa, W. Doki, K. Miyano, H. Okazawa, C. Saji, M. Takahata, Y. Nagashima, M. Takita, T. Yamaguchi, M. Yoshida, S. B. Kim, M. Etoh, K. Fujita, A. Hasegawa, T. Hasegawa, S. Hatakeyama, T. Iwamoto, M. Koga, T. Maruyama, H. Ogawa, J. Shirai, A. Suzuki, F. Tsushima, M. Koshiba, M. Nemoto, K. Nishijima, T. Futagami, Y. Hayato, Y. Kanaya, K. Kaneyuki, Y. Watanabe, D. Kielczewska, R. A. Doyle, J. S. George, A. L. Stachyra, L. L. Wai, R. J. Wilkes, and K. K. Young. Evidence for oscillation of atmospheric neutrinos. *Phys. Rev. Lett.*, 81:1562–1567, Aug 1998. doi: 10.1103/PhysRevLett.81.1562. URL https://link.aps.org/doi/10.1103/PhysRevLett.81.1562.

[23] M. Fukugita and T. Yanagida. Baryogenesis Without Grand Unification. *Phys. Lett. B*, 174:45–47, 1986. doi: 10.1016/0370-2693(86)91126-3.

[24] Thomas Gaisser and Francis Halzen. Icecube. *Annual Review of Nuclear and Particle Science*, 64(1):101–123, 2014.

[25] K.S. Hirata, T. Kajita, M. Koshiba, M. Nakahata, S. Ohara, Y. Oyama, N. Sato, A. Suzuki, M. Takita, Y. Totsuka, T. Kifune, T. Suda, K. Nakamura, K. Takahashi, T. Tanimori, K. Miyano, M. Yamada, E.W. Beier, L.R. Feldscher, E.D. Frank, W. Frati, S.B. Kim, A.K. Mann, F.M. Newcomer, R. Van Berg, W. Zhang, and B.G. Cortez. Experimental study of the atmospheric neutrino flux. *Physics Letters B*, 205(2):416–420, 1988. ISSN 0370-2693. doi: https://doi.org/10.1016/0370-2693(88)91690-5. URL https://www.sciencedirect.com/science/article/pii/0370269388916905.

[26] M. Honda, M. Sajjad Athar, T. Kajita, K. Kasahara, and S. Midorikawa. Atmospheric neutrino flux calculation using the nrlmsise-00 atmospheric model. *Physical Review D*, 92(2), July 2015. ISSN 1550-2368. doi: 10.1103/physrevd.92.023004. URL http://dx.doi.org/10.1103/PhysRevD.92.023004.

[27] P. A. Čerenkov. Visible radiation produced by electrons moving in a medium with velocities exceeding that of light. *Phys. Rev.*, 52:378–379, Aug 1937. doi: 10.1103/PhysRev.52.378. URL https://link.aps.org/doi/10.1103/PhysRev.52.378.

[28] Takaaki Kajita. Atmospheric neutrino results from super-kamiokande and kamiokande â evidence for Îœ oscillations. *Nuclear Physics B - Proceedings Supplements*, 77(1â3):123â132, May 1999. ISSN 0920-5632. doi: 10.1016/s0920-5632(99)00407-7. URL http://dx.doi.org/10.1016/S0920-5632(99)00407-7.

[29] Takaaki Kajita. Atmospheric neutrinos and discovery of neutrino oscillations. *Proceedings of the Japan Academy, Series B*, 86(4):303–321, 2010.

[30] A. Lazo. Pid_2_dst. https://git.km3net.de/alazo/pid_2_dst, 2022.

[31] Ernest Ma. Verifiable radiative seesaw mechanism of neutrino mass and dark matter. *Phys. Rev. D*, 73:077301, Apr 2006. doi: 10.1103/PhysRevD.73.077301. URL https://link.aps.org/doi/10.1103/PhysRevD.73.077301.

[32] L. Maderer. Parameter based pid for km3net. *Internal KM3NeT Technote*, 2021.

[33] Lukas Maderer. Parampid. URL https://git.km3net.de/parapid/parampid.

[34] Ziro Maki, Masami Nakagawa, and Shoichi Sakata. Remarks on the Unified Model of Elementary Particles. *Progress of Theoretical Physics*, 28(5):870–880, 11 1962. ISSN 0033-068X. doi: 10.1143/PTP.28.870. URL https://doi.org/10.1143/PTP.28.870.

[35] Lodewijk Nauta. *Quantum Effects on a Planetary Scale - The First Neutrino Oscillation Measurement with KM3NeT/ORCA*. PhD thesis, Nikhef, Amsterdam, 2022.

[36] Brian O Fearraigh. *Following the light - Novel event reconstruction techniques for neutrino oscillation analyses in KM3NeT/ORCA*. PhD thesis, Amsterdam U., 2024.

[37] A. L. Pedrajas. Parameter based pid in orca6 for neutrino2022. *Internal KM3NeT Technote*, 2022.

[38] 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.

[39] Shuhei Watanabe. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance, 2023. URL https://arxiv.org/abs/2304.11127.