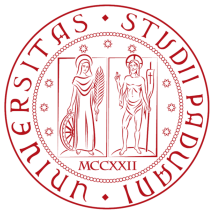


UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia

Corso di Laurea Triennale in Fisica

Tesi di Laurea

Inverse Beta Decay events selection in JUNO using Machine Learning algorithms

Relatore

prof. Alberto Garfagnini

Correlatori

dott. XXXXXX

dott. XXXXXX

Laureando

Candidate

Badge Number

Anno Accademico YYYY/YYYY

Discussion date

Abstract

The Jiangmen Underground Neutrino Observatory (JUNO) will be the largest liquid scintillator-based neutrino detectors in the World, for the next decade. Thanks to its large active mass (20 kt) and state of the art performances (3% effective energy resolution at 1 MeV), it will be able to perform important measurements in neutrino physics. The proposed thesis will study the application of different Machine Learning inspired algorithms for the discrimination of signal events (interactions of anti-neutrinos coming from the nearby nuclear power plants) from background events.

Contents

Contents **ii**

1 Introduction **1**
1.1 Neutrinos Oscillation 1
1.2 The JUNO detector 3
1.3 JUNO signal and background 5

2 Frameworks **11**
2.1 Introduction to Machine Learning 11
2.2 Decision Tree 11
2.3 Neural Networks 13

3 Analysis **17**
3.1 Datasets 17
3.2 Feature creation 18
3.3 Models 21
3.4 Conclusion 27

References **29**

Chapter 1

Introduction

The Jiangmen Underground Neutrino Observatory (JUNO), currently under construction in southern China, is a large liquid scintillator neutrino detector. It is designed to detect electron antineutrino interactions produced by nearby Nuclear Power Plants (NPP) through the inverse beta decay reaction. The primary objective of this experiment is to determine the neutrino mass hierarchy, thereby addressing the Neutrino Mass Ordering (NMO) problem.

The field of neutrino physics has entered a new era of precision following the measurement of the third lepton mixing angle, the so-called reactor angle θ_{13}. This has had a significant impact on models of neutrino mass and mixing. The JUNO experiment, with its excellent energy resolution and large fiducial volume, is expected to make significant contributions to this field.

This leads us to the theory of neutrino oscillation, a quantum mechanical phenomenon whereby a neutrino created with a specific lepton flavor can later be measured to have a different flavor. The oscillation is quantified in terms of parameters that the JUNO experiment aims to measure with high precision.

1.1 Neutrinos Oscillation

The Standard Model of elementary particle interactions provides an accurate description of strong, weak, and electromagnetic interactions, but it treats these interactions as distinct and unrelated. Within this framework, neutrinos are assumed to be massless, but this assumption has been called into question by physicists. Neutrino oscillations, are a potential indication of neutrino mass.
The term "neutrino oscillations" refers to this phenomena and it involve the conversion of a neutrino of a particular flavor to another as it propagates through space.

Each known flavor eigenstate, $(\nu_e, \nu_\mu, \nu_\tau)$, linked to three respective charged leptons (e, μ, τ) via the charged current interactions can be considered a complex combination of neutrino mass eigenstates as follow:

$$\begin{pmatrix} v_e \\ v_\mu \\ v_\tau \end{pmatrix} = U_{\text{PMNS}} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}$$

in wich ν_i are the three mass eigensates, that have 3 masses m_i $(i = 1, 2, 3)$, which are non-degenerate, with $m_i \neq m_j$ for $i \neq j$.

The matrix U_{PMNS}, called Pontecorvo-Maki-Nakagawa-Sakata (PMNS) matrix, is composed of three rotation matrices, R_{23}, R_{13}, and R_{12}, each corresponding to a different mixing angle, θ_{23}, θ_{13}, and θ_{12}, respectively and a parameter δ_{CP} called the Dirac CP-violating phase. For this case, the Majorana CP phases are $\eta_i (i = 1, 2)$, which are only physically possible if neutrinos are Majorana-type particles and do not participate in neutrino oscillations. Therefore, U can be expressed as:

$$
\begin{aligned}
U_{\text{PMNS}} &= \\
&= \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{23} & s_{23} \\ 0 & -s_{23} & c_{23} \end{pmatrix} \begin{pmatrix} c_{13} & 0 & s_{13}\mathrm{e}^{-\mathrm{i}\delta_{CP}} \\ 0 & 1 & 0 \\ -s_{13}\mathrm{e}^{\mathrm{i}\delta_{CP}} & 0 & c_{13} \end{pmatrix} \begin{pmatrix} c_{12} & s_{12} & 0 \\ -s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathrm{e}^{\mathrm{i}\eta_1} & 0 & 0 \\ 0 & \mathrm{e}^{\mathrm{i}\eta_2} & 0 \\ 0 & 0 & 1 \end{pmatrix}
\end{aligned}
$$

where $s_{ij} \equiv \sin\theta_{ij}, c_{ij} \equiv \cos\theta_{ij}$.

The theoretical framework for neutrino oscillations involves the calculation of the oscillation probability as a function of the distance traveled by the neutrino, the neutrino mixing matrix, and the difference in squared masses between the three neutrino mass states, $\Delta m^2_{ij} = m^2_i - m^2_j$ for $i, j = 1, 2, 3, i > j$. Specifically, two nuclear power reactors 53 km away from the detector, which mostly produce anti-electron neutrinos $\bar{\nu}_e$ with energy below 10 MeV, are the principal sources of neutrinos for the JUNO experiment. So, it is necessary for the JUNO experiment to calculate the survival probability $P(\bar{\nu}_e \to \bar{\nu}_e)$ of electron antineutrinos.

$$
P(\bar{\nu}_e \to \bar{\nu}_e) = 1 - \sin^2 2\theta_{12} c^4_{13} \sin^2\left(\frac{\Delta m^2_{21} L}{4\mathcal{E}}\right) - \sin^2 2\theta_{13}\left[c^2_{12}\sin^2\left(\frac{\Delta m^2_{31} L}{4\mathcal{E}}\right) + s^2_{12}\sin^2\left(\frac{\Delta m^2_{32} L}{4\mathcal{E}}\right)\right]
$$

where $\mathcal{E}$ is the neutrino energy, L the travelled distance and $\Delta m^2_{ij} \equiv m^2_i - m^2_j$.
Past experiments have already given estimates for $\Delta m^2_{21}, |\Delta m^2_{31}|$ and the 3 mixing angles.

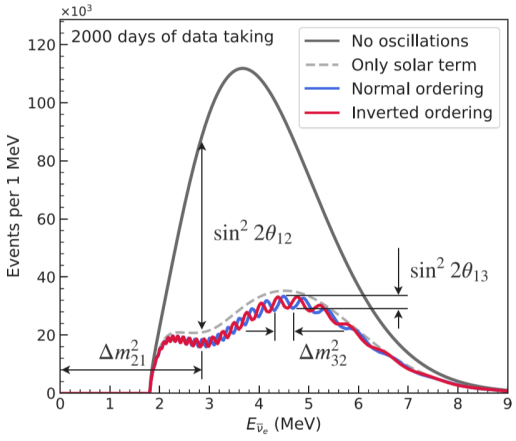


Figure 1.1: JUNO's reactor antineutrino energy spectrum is shown with and without the effect of neutrino oscillation. The gray dashed curve includes only the term in the disappearance probability modulated by $sin^2(2\theta_{12})$, while the blue and red curves use the full oscillation probability for normal and inverted mass orderings. Spectral features driven by oscillation parameters are illustrated, highlighting the rich information available in JUNO's high-resolution measurement of the oscillated spectrum.

JUNO's primary objective is to refine these results, particularly to ascertain the sign of Δm^2_{31}, which will distinguish between two potential scenarios:

- *Normal Ordering (NO)*, where $|\Delta m^2_{31}| = |\Delta m^2_{32}| + |\Delta m^2_{21}|$ and the mass hierarchy is $m_1 < m_2 < m_3$,
- *Inverted Ordering (IO)*, where $|\Delta m^2_{31}| = |\Delta m^2_{32}| - |\Delta m^2_{21}|$ and the mass hierarchy is $m_3 < m_1 < m_2$.

The sign of Δm^2_{31} subtly alters the plot of 1.1. However, it remains uncertain whether the ν_3 neutrino mass eigenstate is heavier or lighter than the ν_1 and ν_2 mass eigenstates.

1.2 The JUNO detector

Nestled beneath the Dashi hill in Jinji town, Southern China, the Jiangmen Underground Neutrino Observatory (JUNO) is an ongoing experiment. Its placement 43 km southwest of Kaiping city was strategically chosen to significantly reduce background noise from cosmic rays due to its underground location. JUNO is anticipated to detect a plethora of antineutrinos, predominantly originating from the Taishan and Yangjiang nuclear power plants (NPPs). These power plants are approximately 52.5 km away from the JUNO detector and together, they have a combined nominal thermal power of 26.6 GW_{th}. The detector's design has been meticulously optimized for the highest sensitivity to the ordering of neutrino masses.

A schematic illustration of JUNO is presented in Fig.1.2.

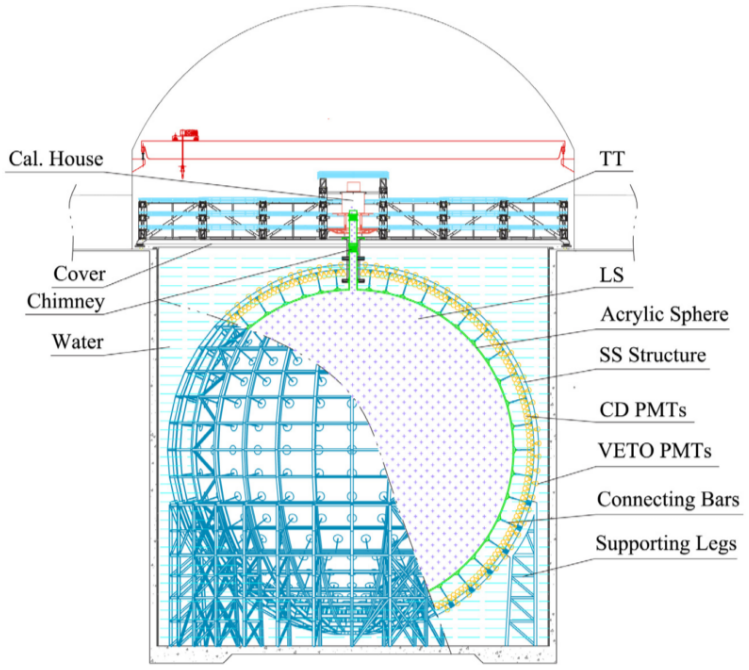


Figure 1.2: Schematic view of the JUNO experiment

Furthermore, the JUNO experiment deploys a specialized compact detector named TAO. Situated approximately 30 meters from one of the Taishan reactors, TAO serves to measure

the unoscillated reactor antineutrino spectrum shape precisely. The data collected by TAO is intended to provide a crucial data-driven input to refine the spectra from the other reactor cores.
The core of the JUNO detector, the **Central Detector (CD)**, is complemented by a water **Cherenkov detector** and a **Top Tracker (TT)**. Notably, the CD, designed as a compact, non-segmented detector, boasts an effective energy resolution of $\sigma_E/E = 3\%/\sqrt{E(MeV)}$, a testament to the advantage of opting for a compact design over a segmented one.

The CD contains a 20 kton liquid scintillator (LS), safely housed within a spherical acrylic vessel and submerged in a water pool. The pool, with a diameter of 43.5 m and a height of 44 m, provides an adequate buffer to shield the LS from the radioactive influence of the surrounding rock.

The vessel is supported by a stainless steel (SS) structure through connecting bars. Additional CD PMTs are mounted on the inner surface of this structure, which also hosts compensation coils designed to mitigate the Earth's magnetic field and thereby minimize its impact on the photoelectron collection efficiency of the PMTs.

Above the water pool resides the Top Tracker, an assembly of a plastic scintillator array, meticulously arranged to measure muon tracks accurately. The CD is connected to the external environment through a chimney, which facilitates calibration operations. Located above this chimney is the Calibration House, equipped with special radioactivity shielding and a muon detector, playing a crucial role in the overall experimental setup.

1.3 JUNO signal and background

1.3.1 Signal

The JUNO experiment primarily draws its sources from the Taishan and Yangjiang Nuclear Power Plants (NPPs), which house two and six cores respectively. In addition to these, the Daya Bay reactor complex contributes to the antineutrino flux. The reactor power, baselines, and anticipated Inverse Beta Decay (IBD) rates from the Taishan, Yangjiang, and Daya Bay reactor cores are detailed in Table 1.1.

Reactor	Power [GW_{th}]	Baseline [Km]	IBD Rate [day^{-1}]
Taishan	9.2	52.71	15.1
Yangjiang	17.4	52.46	29.0
Daya Bay	17.4	215	3

Table 1.1: Information on nuclear reactors

JUNO employs a Liquid Scintillator (LS) primarily composed of Linear Alkyl-Benzene (LAB), known for its transparency, high flash points, robust light yield, and low chemical reactivity. The LS, with a density of 0.859 g/mL, is further enhanced with 3 g/L of 2,5-diphenyloxazole (PPO) as the fluor, and 15 mg/L of p-bis-(o-methylstyryl)-benzene (bis-MSB) as the wavelength shifter. The scintillator is doped with a small amount of gadolinium, increasing its sensitivity to antineutrinos via the inverse beta decay (IBD) process.

This process is initiated when an antineutrino interacts with a proton in the liquid scintillator, producing a positron and a neutron. It can be described by the following reaction:

$$\overline{\nu}_e + p \rightarrow e^+ + n \tag{1.1}$$

IBD is characterized by a comparatively low threshold of 1.8 MeV, a substantial cross section, and it can be readily differentiated from the background due to its delayed γ signature.

The positron, carrying the majority of the antineutrino's initial energy, deposits this energy in the scintillator through ionization. This energy deposition, coupled with the positron's subsequent annihilation typically into two 0.511 MeV photons, forms the prompt signal, characterized as follow: $e^+ + e^- \rightarrow 2\gamma$. The energy deposited by the positron directly correlates with the antineutrino energy, providing a precise measure critical for neutrino oscillation studies.

Following the prompt signal, the neutron is captured primarily on hydrogen (approximately 99% of the time) after an average delay of about 220 µs. This capture event releases a single 2.2 MeV photon, creating the delayed signal. Occasionally, the neutron is captured on carbon (around 1% of the time), resulting in a gamma-ray signal with a total energy of 4.9 MeV. The process is described as follows:

$$n + {}^1\mathrm{H} \rightarrow {}^2\mathrm{H}^* \rightarrow {}^2\mathrm{H} + \gamma \tag{1.2}$$

Despite carrying only a small fraction of the initial antineutrino energy, typically from zero to a few tens of keV, neutron recoils are considered in the calculations due to JUNO's exceptional energy resolution.

The light output from these events is detected by the photomultiplier tubes (PMTs), sensitive detectors that convert light into an electrical signal. They operate based on the photoelectric effect and subsequent electron multiplication. The signals from all the PMTs are then combined to reconstruct the position and energy of the original neutrino interaction. This technique allows JUNO to measure the energy of the incoming neutrino to high precision, which is crucial for studying neutrino oscillation.

1.3.2 Background

The design and composition of the scintillator in the JUNO experiment are meticulously optimized to minimize background noise from various radiation sources, such as cosmic rays and natural radioactivity. Despite these efforts, several types of background signals are inevitably produced in the detector. For the purpose of analysis, we focus primarily on the three most significant contributors:

Radiogenic Backgrounds

Radiogenic backgrounds in the JUNO experiment primarily originate from the radioactive decay of isotopes such as ^{238}U, ^{232}Th, and ^{40}K. These isotopes are naturally present in the materials comprising the JUNO detector, including acrylic used for the detector walls, the metal structure supporting the detector, PMT glass, the gas during early filling phases, and the surrounding water. They are also found in the surrounding rock. These isotopes undergo radioactive decay, emitting various forms of radiation. The decay of ^{238}U and ^{232}Th occurs through decay chains, where each isotope successively decays into different isotopes, releasing radiation in the process. The emitted radiation includes alpha particles, beta particles, and gamma rays. As for ^{40}K, it undergoes beta decay to ^{40}Ca or electron capture to ^{40}Ar, with a small fraction (0.001%) resulting in the emission of a gamma ray. These radiogenic backgrounds need to be carefully accounted for and minimized to accurately detect reactor antineutrinos in the JUNO experiment.

These radiogenic backgrounds can potentially mimic the signal from inverse beta decay (IBD) in several ways:

1. **Beta decays and electron captures**: These processes result in the emission of electrons or positrons, which can produce a scintillation signal similar to the prompt signal from IBD.

2. **Gamma rays**: High-energy gamma rays can Compton scatter in the detector, producing electrons with enough energy to mimic the prompt signal from IBD. In addition, gamma rays can produce electron-positron pairs in the detector, which can mimic both the prompt and delayed signals from IBD.

3. **Neutrons**: Some decays in the ^{238}U and ^{232}Th chains emit neutrons, which can be captured on protons in the detector, mimicking the delayed signal from IBD.

Cosmogenic Backgrounds

Cosmogenic backgrounds in JUNO primarily result from the interaction of cosmic rays, particularly high-energy muons ($\mathcal{O}$(GeV)), with the detector materials. These interactions lead to the production of fast neutrons and unstable isotopes through the process of spallation in which a high-energy particle strikes a target atom, causing it to emit smaller particles such as neutrons and unstable isotopes. Specifically, these muons interact with the detector materials, resulting in the production of isotopes like ^{9}Li, ^{8}He and ^{11}C, which are unstable and subsequently decay, contributing to additional background events.

These fast neutrons and unstable isotopes, produced from the interactions of muons with the detector materials, can generate signals that mimic an inverse beta decay (IBD) event. Specifically, there are two distinct signals to consider.

The first, known as the prompt signal, is generated by an electron. The energy and momentum of this electron can make it appear like a positron, the particle that would be expected in an IBD event. The second, known as the delayed signal, is generated by a neutron. This neutron can be captured by a proton in the detector, producing a signal identical to what would be expected from the neutron in an IBD event.

Other source of $\overline{\nu}_e$

Other sources of antineutrinos also contribute to the background. Those are geoneutrinos, atmospheric neutrinos, and reactor antineutrinos:

Geoneutrinos are antineutrinos produced by natural radioactivity within the Earth, primarily below 2.5 MeV in antineutrino energy. Natural radioactivity exists in materials present in the Earth's crust and mantle, such as U, Th,and ^{40}K. These materials undergo radioactive decays, generating antineutrinos as decay products, that produce IBD signals.

Atmospheric neutrinos are generated by interactions of cosmic rays with the Earth's atmosphere. When high-energy cosmic rays collide with the atmosphere, they produce a cascade of particles, including muons and neutrinos. The muons generated in these interactions can decay, producing antineutrinos.

Reactor antineutrinos are an artificial source of antineutrinos. Besides the reactors that are used to generate the signal to be analyzed, there are various other reactors that contribute to the total event count. Given the vast number of nuclear reactors worldwide, the collective signal from these reactors becomes significant.

It's beneficial to categorize the aforementioned background sources into two distinct groups:

- *Accidental Background*: This category includes background events that result from the coincidence of two independent events, typically of radiogenic origin. These events primarily influence the low-energy region of the spectrum. A portion of the cosmogenic background also falls into this category. The goal of this thesis work is to significantly reduce these accidental background events, a topic that will be discussed in detail in the following sections
- *Correlated Backgrounds*: These backgrounds originate from a single physics process and produce both a prompt and a delayed signal. Significant correlated backgrounds include cosmogenic Li/He and fast neutrons. Among all the radiogenic processes, only one correlated background requires consideration: the C(α,n)^{16}O, decay that produces an alpha particle (prompt signal) and a neutron that is captured as delayed, exactly like an IBD, occurring within the liquid scintillator.

Here a viasualization sumary of all the bacgrounds contributions:

Following the comprehensive discussion of all background events in the JUNO experiment, it becomes clear that due to the significant presence of various types of background events, efforts are being made to reduce their contribution in every possible way. Several strategies have been employed to mitigate these background signals. Methods include the use of shielding materials to block external radiation, careful selection and treatment of detector materials to

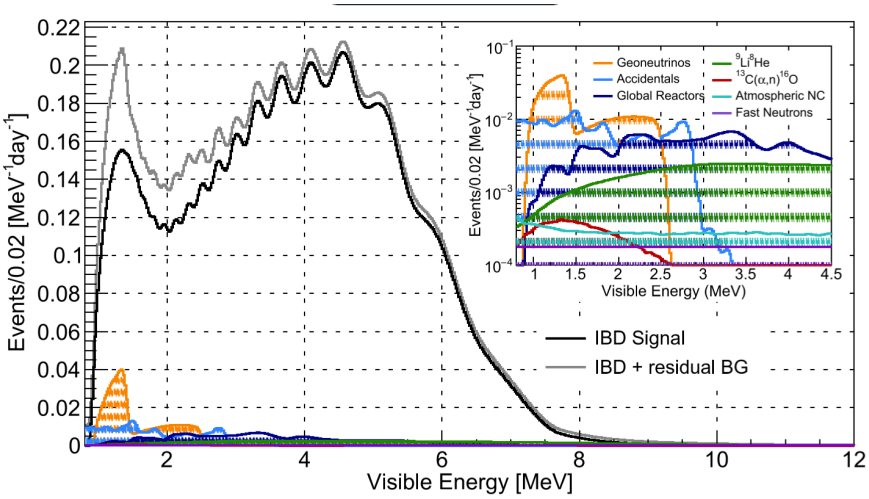


Figure 1.3: Visible energy spectrum with (grey) and without (black) backgrounds is that which is anticipated for JUNO. The predicted backgrounds, which make up around 7% of the whole sample of IBD candidates and are primarily confined below, are shown in the inset as spectra. $\approx$ 3 MeV

minimize internal radioactivity, and sophisticated data analysis techniques to identify and reject background events.

However, it's important to note that a large portion of the accidental background events are the only ones where significant reduction can be achieved. These are the events that occur randomly and independently, and their reduction requires a different approach compared to correlated backgrounds. The focus of this thesis is precisely on these accidental background events, exploring strategies and techniques to further minimize their impact on the experiment. This is a crucial aspect of the experiment's success, as reducing these events can significantly improve the sensitivity and accuracy of the neutrino measurements.

Chapter 2

Frameworks

2.1 Introduction to Machine Learning

Machine learning is a powerful tool that can be used to identify patterns in complex datasets. In the context of particle physics, machine learning algorithms can be used to detect signals from background noise in large datasets generated by detectors. In particular, for the detection of IBD signals from background, machine learning algorithms can be used to identify patterns in the data that are indicative of an IBD event, and to distinguish these signals from the background noise explained above.

2.1.1 Supervised Learning

Supervised learning, a cornerstone of machine learning, operates on the premise of training an algorithm using a labelled dataset. This training ensures that each input aligns with a correct output, with the overarching aim to develop a function that can link input data to output data accurately. A primary task stemming from supervised learning is binary classification, which is responsible for classifying elements of a dataset into one of two possible categories based on inherent features. In the context of this thesis, the emphasis is on applying the binary classification task to the JUNO experiment. The task is to distinguish whether a specific event is an Inverse Beta Decay (IBD) event or a background event.

Two distinct machine learning algorithms, **Gradient Boosting Decision Trees** and **Neural Networks**, are deployed to achieve this.

2.2 Decision Tree

A Decision Tree algorithm, used in supervised machine learning for classification and regression tasks, models the predictive outcome of a target variable based on decision rules inferred from input features. The process involves dissecting the overall dataset into distinct regions, where each region contains data points that are as similar as possible to each other in terms of their target class. Formally, each internal node of the tree represents a decision rule based on an input feature, which bifurcates the data into two child nodes. The decision for splitting the data at each node is determined using a metric known as Information Gain, which in turn is based on the concept of Entropy.

In the context of a binary classification, entropy (H) is a measure of the impurity or disorder within a set (S) of instances. It quantifies the uncertainty involved in predicting the class of a random instance from the set (S). It is mathematically formulated as:

$$H(S) = -p_+ \log_2(p_+) - p_- \log_2(p_-) \tag{2.1}$$

Here, p_+ and p_- denote the proportions of positive and negative instances in the set S, respectively. Entropy attains a maximum value when the set S contains an equal number of positive and negative instances, reflecting the highest uncertainty.

Information Gain (IG) measures the reduction in entropy achieved by partitioning the instances based on a feature (A). It is the difference between the entropy of the set before the split (H(S)) and the weighted sum of the entropies of each subset resulting from the split. It can be formulated as:

$$IG(S, A) = H(S) - \sum_{v \in V(A)} \left(\frac{|S_v|}{|S|} \right) H(S_v) \tag{2.2}$$

where $V(A)$ indicates the set of all possible values of feature A.
In this equation, S_v denotes the subset of instances in S for which the feature A takes on the value v. $|S_v|$ and $|S|$ are the cardinalities of the sets S_v and S, respectively.

The algorithm constructs the tree by recursively applying these splits, each time selecting the feature that results in the maximum information gain. This process continues until a stopping criterion is met, such as reaching a pre-specified maximum depth of the tree or a minimum number of samples per leaf.

While Decision Trees are straightforward and practical models, their ability to decipher complex patterns in data can be limited. This limitation paves the way for a more advanced technique known as Gradient Boosting Decision Trees.

2.2.1 Gradient Boosting Decision Trees

Gradient Boosting is a machine learning algorithm that stems from the concept of boosting, with the application of gradient descent methodology. Its goal is to produce a robust predictive model through the combination of multiple weak learners, typically decision trees.

The primary innovation in Gradient Boosting over classical boosting techniques is its approach to error correction. Instead of modifying the weights of misclassified instances, Gradient Boosting fits each new tree to the residuals (or the negative gradient) of the loss function with respect to the prediction of the existing ensemble of trees. This means each new tree is trained to predict the error of the existing model, thereby iteratively reducing the overall error.

Let's formalize this process:

1. **Initialization**: We begin by initializing our model with a constant value. This is denoted as $F_0(x) = \arg\min_\gamma \sum_{i=1}^{N} L(y_i, \gamma)$, where $L(y, F(x))$ represents the loss function, y represents the true target value, and $F(x)$ is the model's prediction for the input features x. This constant prediction, γ, is chosen to minimize the total loss over all N instances. Thus, our initial model starts with a prediction that globally minimizes the loss.

2. **Computation of Residuals**: Next, we iteratively construct an ensemble of M trees. For each iteration $m = 1$ to M, we calculate the residuals as

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{2.3}$$

for each instance $i = 1, 2, ..., N$. These residuals are essentially the negative gradients (or first derivatives) of the loss function with respect to the model's predictions. They provide a measure of the direction that would decrease the loss function fastest.

3. **Fitting a Decision Tree**: After computing the residuals, we fit a new decision tree, $h_m(x)$, to these residuals. This tree is thus trained to predict the negative gradient of the loss function, using train it using the training set $(x_i, r_{im})_{i=1}^n$. By doing so, it attempts to correct the errors made by the current ensemble model.

4. **Model Update**: The model is then updated by applying the rule

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x) \tag{2.4}$$

Here, ν represents the learning rate, a parameter typically less than 1, which controls the contribution of each tree to the final prediction. This essentially adjusts the previous model's prediction in the direction that most decreases the loss.

5. **Final Model**: The final model's prediction is given by $F_M(x) = F_0(x) + \sum_{m=1}^{M} \nu \cdot h_m(x)$. In the final ensemble model, each decision tree provides a small correction to the predictions of the previous trees, collaboratively reducing the loss function's value and improving the overall model's performance.

An advanced and highly efficient implementation of this method is XGBoost, which introduces several improvements such as regularization terms in the objective function to prevent overfitting, the computation of the second-order gradient for faster convergence, and built-in mechanisms to handle missing values and enable parallel processing.

2.3 Neural Networks

Neural Networks (NNs) are computational models that draw inspiration from the interconnected structure of the human brain. Each individual computational unit, often referred to as an "artificial neuron" or simply "neuron", is designed to mimic the fundamental working mechanism of a biological neuron.

Let's denote the inputs to an artificial neuron as $x = [x_1, x_2, ..., x_n]$, a representation analogous to dendrites in a biological neuron. These inputs are linearly transformed by a set of weights, $w = [w_1, w_2, ..., w_n]$, summed together, and a bias term, b, is added to the result. This operation can be expressed mathematically as:

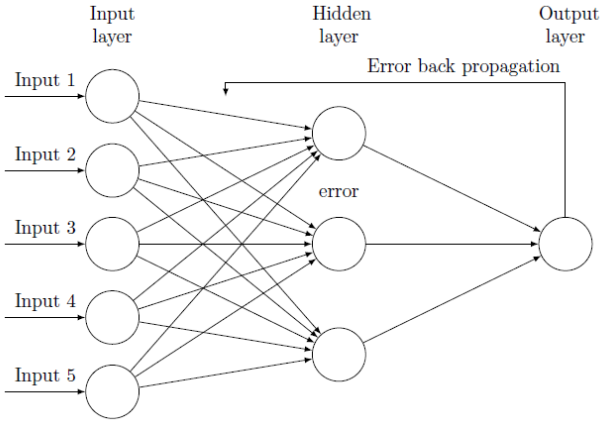
$$z = \sum_{i=1}^{n} w_i x_i + b$$

The calculated value, z, is then passed through an *activation function*, f, to generate the neuron's output, $a = f(z)$.

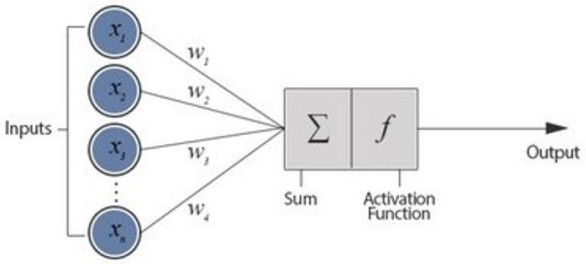
The activation function introduces non-linearity into the model, which is crucial for the network's ability to learn complex patterns. Common choices for f include the sigmoid, hyperbolic tangent, and ReLU (Rectified Linear Unit) functions.

An Artificial Neural Network builds upon the concept of the artificial neuron to form an interconnected assembly of these neurons, structured in layers. An ANN typically comprises an input layer, one or more hidden layers, and an output layer. Each layer may contain one or more neurons, and the layers are fully connected, meaning every neuron in one layer connects with all neurons in the following layer.

The following is a graphical representation of an ANN and a single neuron:



(a) Graphic representation of ANN



(b) Single Neuron representation

For classification problems, the output layer typically uses a softmax function for multi-class problems to output a probability distribution over the classes, or a sigmoid function for binary classification problems to provide the probability of the positive class.

Training a neural network involves a two-step process: *forward propagation* and *backpropagation*.

In **forward propagation**, the input is passed through the network to generate an output. This output is then compared with the actual target to compute the loss function L.

Backpropagation uses the chain rule of calculus to compute the gradient of L with respect to the network's parameters, which are then used to update the weights and biases:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial a}\frac{\partial a}{\partial z}\frac{\partial z}{\partial w}$$

Here, $\frac{\partial L}{\partial a}$ is the derivative of the loss function with respect to the activation output, $\frac{\partial a}{\partial z}$ is the derivative of the activation function, and $\frac{\partial z}{\partial w}$ is the derivative of the weighted sum with respect to the weights.

Once these gradients are calculated, they are used to update the weights and biases via *gradient descent*, a process that iteratively adjusts the parameters to minimize the loss function:

$$w_{\text{new}} = w_{\text{old}} - \alpha\frac{\partial L}{\partial w}$$

$$b_{\text{new}} = b_{\text{old}} - \alpha\frac{\partial L}{\partial b}$$

In these equations, α is the learning rate, a hyperparameter that determines the size of the steps the algorithm takes down the gradient towards the minimum.

The interconnected structure of ANNs, combined with the ability of backpropagation and gradient descent to effectively adjust the model parameters, allows these networks to learn and represent complex, non-linear relationships in the data.

Chapter 3

Analysis

3.1 Datasets

è iortante dei dire che sono stati utilizzati due datasets. The labeled datasets employed in the ensuing analysis are all products of Monte Carlo simulation, generated via the SNiPER software.

The first of the datasets provided is specifically tailored for the study of Inverse Beta Decay (IBD) events. Each event within this dataset, simulated and injected into the system, is tagged with a unique Simulation Identifier, or SimID. Furthermore, events which trigger a sufficient number of PMTs to be captured by the electronic system are assigned an EventID. This intricate labeling system allows for a clear differentiation between correlated IBD events, which represent actual IBD occurrences, and uncorrelated IBD events.

The second dataset focuses primarily on radioactivity events. Similar to the IBD dataset, it encompasses a large number of simulated events, each reflecting the complex reality of real-world physics phenomena. Additionally, the inherent electronic noise prevalent in actual physical environments is accurately accounted for, ensuring a realistic representation within the simulated context.

In this research undertaking, my central task will focus on a detailed examination and evaluation of the provided datasets. My work will primarily involve not just interpreting the inherent characteristics and peculiarities of the recorded events, but also harnessing these insights to construct comprehensive feature tables. These feature tables, generated from the datasets, will serve as the basis for my subsequent analysis and interpretation, a process which will be elaborated upon in the following sections of this study. The aim is to provide a meaningful understanding of the correlations and implications of these events within the broader context of the JUNO experiment.

Dataset Name	Number of Events	Rates (used in elecsim)
U238@LS	1,000,000 events	3.234 Hz
Th232@LS	1,000,000 events	0.733 Hz
K40@LS	1,000,000 events	0.53 Hz
Pb210@LS	1,000,000 events	17.04 Hz
C14@LS	1,000,000,000 events	3.3e4 Hz
Kr85@LS	1,000,000 events	1.163 Hz
U238@Acrylic	10,000,000 events	98.41 Hz
Th232@Acrylic	10,000,000 events	22.29 Hz
K40@Acrylic	10,000,000 events	161.25 Hz
U238@node/bar	100,000,000 events	2102.36 Hz
Th232@node/bar	100,000,000 events	1428.57 Hz
K40@node/bar	100,000,000 events	344.5 Hz
Co60@node/bar	100,000,000 events	97.5 Hz
U238@PMTGlass	1,000,000,000 events	4.90e6 Hz
Th232@PMTGlass	1,000,000,000 events	8.64e5 Hz
K40@PMTGlass	1,000,000,000 events	4.44e5 Hz
Tl208@PMTGlass	1,000,000,000 events	1.39e5 Hz
Co60@Truss	0	? Hz
Tl208@Truss	0	? Hz
Rn222@WaterRadon	100,000,000 events	90 Hz

Table 3.1: Here Caption

3.2 Feature creation

The development of machine learning models for the detection of Inverse Beta Decay (IBD) events necessitates a systematic and efficient approach to feature engineering. This process begins with the loading of two separate datasets, one for IBDs and one for radioactivity background, each containing a multitude of potential IBD events. The primary objective is to construct a feature table that encapsulates the unique characteristics of these events, providing a robust foundation for subsequent model training.

3.2.1 IBD dataset

As we mentioned earlier, an IBD event is characterized by two distinct signals with different energies, positions, and times. The first, known as the prompt signal, is caused by the annihilation of a positron with an electron in the scintillator liquid. This interaction yields a signal with a characteristic energy. The second, the delayed signal, results from the capture of a neutron by the scintillator liquid. This signal occurs with a significant delay, at a different position, and with a different energy compared to the prompt signal.

To create the feature table, all possible pairs of events within the dataset were considered, without repetition. Each possible combination was ordered temporally, meaning the second event followed the first. This temporal ordering is crucial in feature determination. Given a pair $i - j$, and considering that neutron capture occurs temporally subsequent to electron-positron annihilation, the following features were constructed:

- R_{prompt}: This feature represents the distance of the prompt signal, calculated as the distance from the origin to the point (x_i, y_i, z_i) in the detector space where the prompt signal occurred.

- $R_{delayed}$: Similar to R_{prompt}, this feature represents the distance of the delayed signal, calculated as the distance from the origin to the point (x_j, y_j, z_j) in the detector space where the delayed signal occurred.

- E_{prompt}: This feature represents the energy of the prompt signal. It captures the characteristic energy released during the annihilation of a positron with an electron in the scintillator liquid.

- $E_{delayed}$: This feature represents the energy of the delayed signal. It captures the energy released when a neutron is captured by the scintillator liquid. This capture can occur by hydrogen, resulting in a gamma ray with an energy of approximately 2.2 MeV, or by carbon, resulting in gamma rays with combined energies of about 4.95 MeV to 5.12 MeV.

- Δ_{Time}: This feature represents the time difference between the two events. It captures the temporal delay between the occurrence of the prompt and delayed signals.

- Δ_{Radius}: This feature represents the spatial distance between the two events. It captures the spatial separation between the points in the detector space where the prompt and delayed signals occurred.

These features encapsulate the temporal and spatial differences between the prompt and delayed signals, as well as their respective energies, providing a comprehensive representation of the unique characteristics of IBD events.

Event Labeling

In the context of supervised learning, the process of labeling is crucial as it provides the ground truth against which the performance of the machine learning model is evaluated. In this scenario, each pair of events in the dataset is assigned a label that indicates whether it represents a true Inverse Beta Decay (IBD) event or a background signal (BKG).

The label is a binary value: a label of 1 signifies a true IBD event, while a label of 0 signifies a BKG event. The assignment of these labels is not arbitrary but is guided by a specific criterion based on the simulation identifier (SimID) associated with each event pair.

The SimID is a unique identifier assigned to each simulated event pair during the generation of the dataset. If a pair of events share the same SimID, it means they were generated as part of the same simulation and thus are considered to represent a true IBD event. In this case, they are assigned a label of 1.

Conversely, if a pair of events do not share the same SimID, it means they were generated as part of different simulations. These events are not correlated and thus are considered to represent BKG events. In this case, they are assigned a label of 0.

This labeling strategy based on the SimID ensures a systematic and consistent methodology for event classification.

Efficient Feature Calculation

Given the large size of the dataset and the computational complexity of feature calculation, a parallel computing approach was adopted to enhance efficiency. The feature calculation task was divided into multiple sub-tasks that could be executed simultaneously by different cores of a CPU. This parallelization significantly reduced the total computation time, particularly beneficial when working with large volumes of data.

To further optimize the computation, a method was implemented to only consider event pairs where the delayed event occurs within a time window of $5 * \tau$ from the prompt event. This approach is based on the fact that the time delay between the prompt and delayed events in Inverse Beta Decay (IBD) typically follows an exponential distribution, a characteristic of radioactive decay processes. While this method significantly reduces the number of potential event pairs, it might exclude about 0.7% of IBD events that occur outside this time window.

While this percentage is relatively small, it's important to consider the potential impact on the analysis results.

3.2.2 Radioactivity dataset

For the radioactivity dataset, the feature calculation was performed in a manner analogous to the IBD dataset. The key difference is that event pairs from the radioactivity dataset are labeled as BKG events, hence assigned a label of 0.

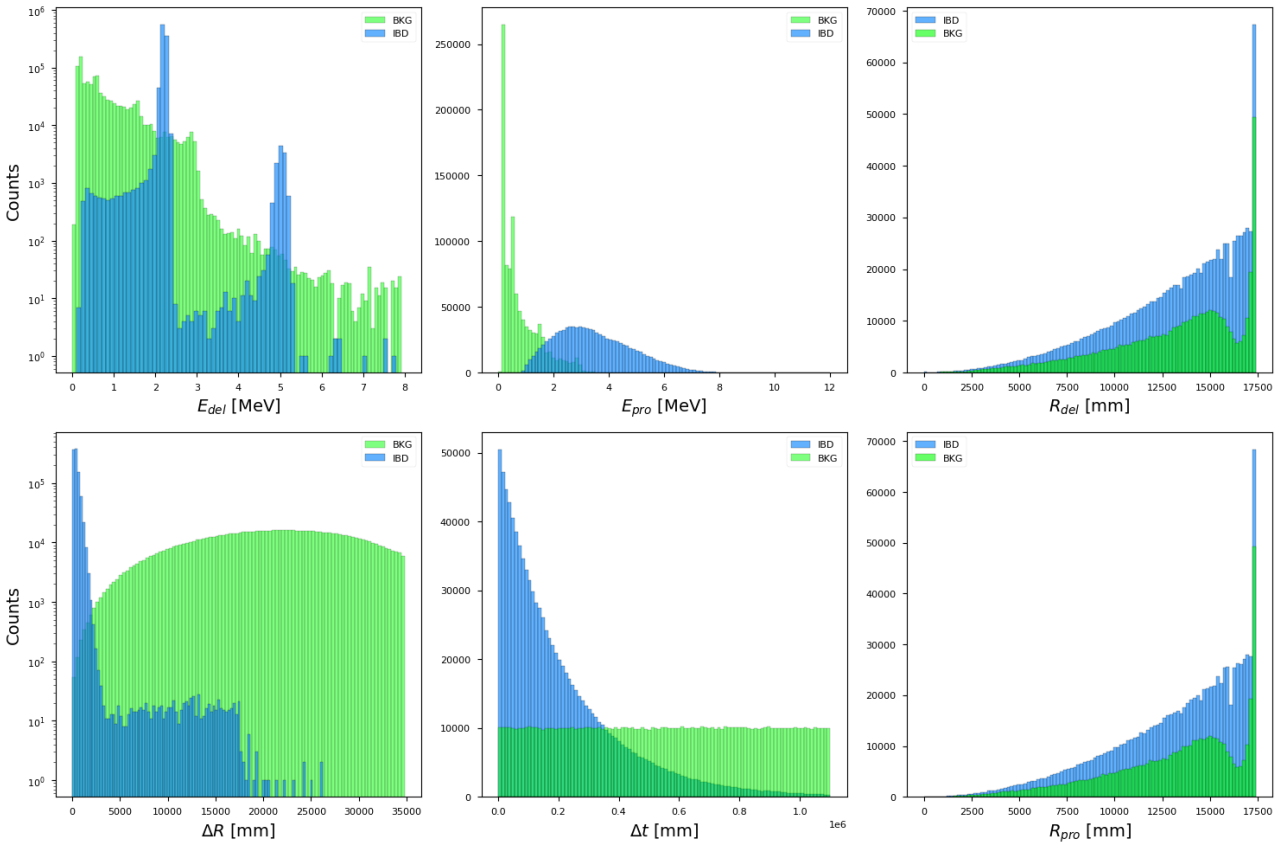


Figure 3.1: Features histograms

In summary, the feature engineering process for IBD event detection involves careful consideration of the unique characteristics of these events, systematic feature construction, and efficient computation strategies. This process provides a robust foundation for the development and training of machine learning models for IBD event detection.

3.3 Models

In the context of the JUNO experiment, a significant part of the effort involves the implementation and optimization of an event selection algorithm. The primary objective of this algorithm is to identify and select Inverse Beta Decay events induced by reactor antineutrinos, which are of paramount importance in the study of neutrino oscillations.

In this chapter, we will present several algorithms for event selection.

The first algorithm is a manual cut-based approach, **Manual Cut**, where specific cuts are defined to select events of interest. This approach involves setting criteria based on the physical characteristics of the events and known background noise sources in the detector. The manual cut algorithm allows for precise control over the selection process and enhances the signal-to-background ratio.

In addition to the manual cut algorithm, other algorithms discussed in this chapter are based on machine learning models, specificaly based on **Boosted Decision Trees** and on **Neural Network**.

By exploring both manual cut and machine learning-based algorithms, we aim to provide a comprehensive understanding of different approaches to event selection, highlighting their strengths and limitations in the context of the JUNO experiment.

3.3.1 Manual Cut

The algorithm is designed to suppress various types of background noise while maintaining high efficiency for true IBD events. The selection criteria, or "cuts", are implemented using Python, and are applied to the Features Tables discussed above. Each cut within the algorithm serves a distinct purpose in the overall event selection process.

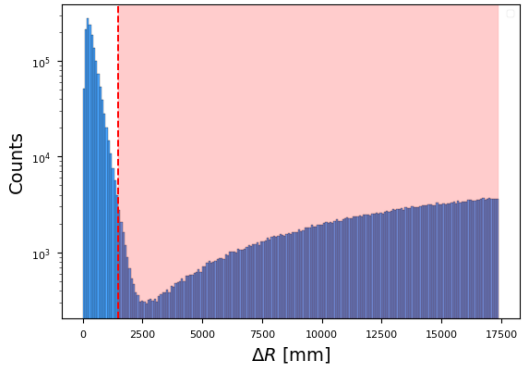
The key components of the event selection algorithm are as follows:

1. **Delta Time (Δt) and Delta Radius (ΔR) cuts**: The first cut is applied on the time delay and the radial distance between the prompt and delayed signals. The criteria are:

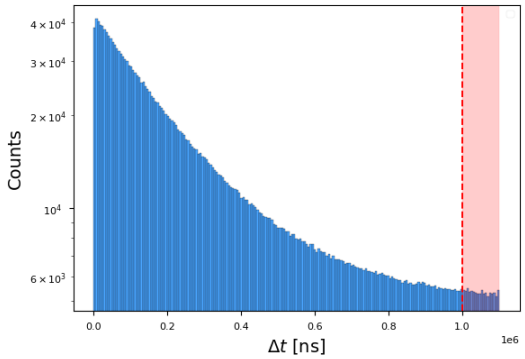
   - Time separation between the prompt and delayed signals should be less than 1.0 ms.
   - Spatial 3D separation should be less than 1.5 m.

   These cuts are designed to reduce accidental background noise, which is the coincidence of two otherwise uncorrelated events, typically of radiogenic origin. The accidental background can be measured with excellent precision and subtracted by off-time window techniques. By imposing a limit on the time and spatial separation between the prompt and delayed signals, the algorithm can effectively distinguish between true IBD events and accidental coincidences.

2. **Energy of the Prompt Signal (E_{pro}) Cut**: The next cut is applied on the energy of the prompt signal, which is the initial signal produced by the antineutrino interaction. The criteria are:



(a) ΔR cut

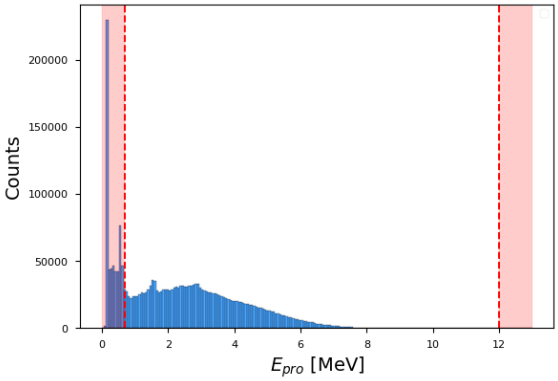


(b) Δt cut

- Energy of the prompt signal should be within the [0.7, 12.0] MeV range.

This cut is based on the expectation that IBD events dominate this energy range. The energy of the prompt signal corresponds to the energy of the positron from the IBD reaction, and the specific range is chosen to maximize the signal-to-background ratio.

Figure 3.2: E_{pro} cut

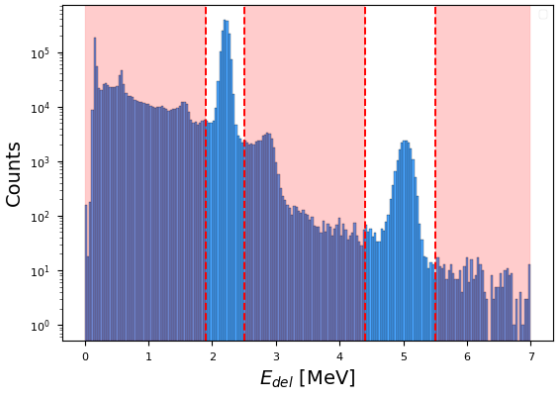


3. **Energy of the Delayed Signal (E_{del}) Cut**: The final cut is applied on the energy of the delayed signal, which is the signal produced by the neutron capture that follows the antineutrino interaction. The criteria are:

   - Energy of the delayed signal should be within the [1.9, 2.5] MeV or [4.4, 5.5] MeV ranges.

These energy selection windows correspond to neutron capture on hydrogen and carbon, respectively. The energy of the delayed signal is characteristic of the neutron capture process, and the specific ranges are chosen to correspond to the expected energies of neutron capture on hydrogen and carbon in the detector.

Figure 3.3: E_{del} cut



An evaluation was conducted to assess the accuracy of the algorithm. In this evaluation, the feature table of true IBD events was utilized to gauge the algorithm's performance. The results

indicated that out of a total of 1,468,385 events, 1,435,115 events were accurately classified as IBD. This achievement translates to an accuracy rate of 97.73%, indicating the algorithm's successful identification of the majority of true IBD events.

Furthermore, the efficiency calculation was performed on the background dataset, yielding the following results: 26.0 out of 993,457 events were mistakenly classified as IBD, resulting in an efficiency rate of 99.997%. This indicates that the algorithm has an excellent capability to distinguish and reject background events, achieving a high level of efficiency in identifying true IBD events. The extremely low misclassification rate of background events as IBD further highlights the algorithm's effectiveness in minimizing false positives and maintaining a high level of purity in the selected IBD events.

A summary table presents the results obtained from the evaluations:

Table 3.2: Summary of Algorithm Performance

Dataset	Total Events	IBD Events Selected	Accuracy (%)
True IBD Events	1,468,385	1,435,115	97.73
Background Events	993,457	26	99.997

3.3.2 XGBoost

In this section, we will delve into the application of the Extreme Gradient Boosting, or XGBoost model, for our event selection task. XGBoost is an advanced and highly efficient implementation of gradient-boosting decision trees, crafted with an emphasis on speed and performance. It uniquely combines the strengths of gradient boosting with the advantages of parallel processing. The result is a powerful, flexible, and fast algorithm that has won favor in a wide range of machine learning tasks and competitions. The speed, performance, and the ability to manage complex patterns of XGBoost make it particularly suited for the event selection in the context of the JUNO experiment. In the following subsections, we will discuss the specifics of how the XGBoost model was developed, optimized, and applied to our task.

The XGBoost model was initialized with several specific hyperparameters to fine-tune the model's performance. These hyperparameters are key factors that determine how the model learns from the data:

- **Number of parallel threads** (*nthread*) : This hyperparameter is set to -1, which means that the model uses the maximum number of threads available on the system. The number of threads refers to the number of operations that can be run in parallel during training, and using more threads can speed up the training process.
- **Random seed** (*seed*) : The seed is set to 1 for reproducibility. The seed value is used to initialize the random number generator, which is part of the stochastic aspect of the gradient boosting algorithm. Using the same seed value ensures that the same sequence of random numbers is generated each time the code is run, which allows the results to be reproduced exactly.
- **Number of estimators** ($n_{estimators}$) : The model was configured with 10,000 estimators. In the context of gradient boosting, an estimator is a single decision tree. The number of

estimators represents the maximum number of decision trees to be built, and it is one of the most important hyperparameters to tune. Having more trees makes the model more complex and can lead to better performance on the training data, but if too many trees are used, the model may overfit the training data and perform poorly on unseen data.

- **Learning rate** ($learning_rate$) : The learning rate was set to 0.05. This hyperparameter controls how much each tree contributes to the final prediction. A smaller learning rate means that each tree has less influence, so more trees are needed to create a model of similar complexity. A smaller learning rate can make the model more robust to overfitting and allow it to generalize better to unseen data.

- **Maximum tree depth** (max_depth) : The maximum depth of each tree was set to 3. The depth of a tree is the length of the longest path from the root to a leaf, and it controls how complex each tree can be. A larger depth allows the model to fit the data more closely, but it can lead to overfitting if the depth is too large.

The chosen hyperparameters strike a balance between computational efficiency and model performance, allowing control over the learning process and model complexity. To optimize the XGBoost model, a Grid Search technique was used. This method systematically evaluated various hyperparameter combinations to identify the optimal configuration that maximizes model accuracy.

Results

In this study, we utilized the XGBoost algorithm to classify true Inverse Beta Decay (IBD) events and measure its performance against a dataset of these events. Out of a total of 1,468,385 events, the XGBoost algorithm managed to accurately identify 1,468,344 IBD events. This result corresponds to an impressive efficiency rate of 99.9972%, highlighting the superior ability of the XGBoost algorithm to correctly identify the majority of true IBD events.

Further evaluation of the algorithm's performance was conducted on the background dataset. Out of 993,457 background events, only 15 were incorrectly classified as IBD events. This achievement translates to an exceptional efficiency rate of 99.9985%. It is clear from these results that the XGBoost algorithm is highly proficient in distinguishing and rejecting background events, achieving a remarkable level of efficiency in identifying true IBD events. The exceptionally low misclassification rate of background events as IBD underlines the effectiveness of the XGBoost algorithm in minimizing false positives in the selected IBD events.

In addition to previous model evaluations, a confusion matrix was calculated. This provided further insight into the precision and effectiveness of the model. The analysis was applied to a test dataset, representing 20% of the initial dataset of approximately one million elements. The detailed results are illustrated in the Table 3.3.

	Predicted: Yes	**Predicted: No**
True: Yes	TP (200485)	FN (6)
True: No	FP (2)	TN (199540)

Table 3.3: Confusion Matrix

	XGBoost
IBD Efficiency	99.9972%
BKG Efficiency	99.9985%

Table 3.4: Performance of the XGBoost.

Interpretation of the model

In the context of our work, we have trained an XGBoost model, which, while powerful and accurate, is also complex due to its ensemble nature. Interpreting such models and understanding the underlying relationships between features and predictions can be quite challenging. To tackle this complexity and offer substantial insights, we adopted **SHAP** (SHapley Additive exPlanations) for model interpretability.

SHAP is a tool from game theory that provides an understanding of any machine learning model's output. It links optimal credit distribution with local explanations, making use of the Shapley values from game theory.

In essence, making a prediction is considered a "game" where the "players" are the features. Each feature's Shapley value is its average marginal contribution across all potential coalitions.

We denote the total number of features by N and compute the SHAP value for a specific feature i by averaging its marginal contribution across all possible coalitions. The fundamental process for calculating the SHAP value is described as follows:

a. Consider all 2^N possible subsets of the features.

b. For each subset S that includes feature i, make a prediction using only the features in S, and another prediction excluding feature i.

c. Compute the difference between the two predictions for each subset.

d. Average all these differences. This average is the SHAP value for feature i.

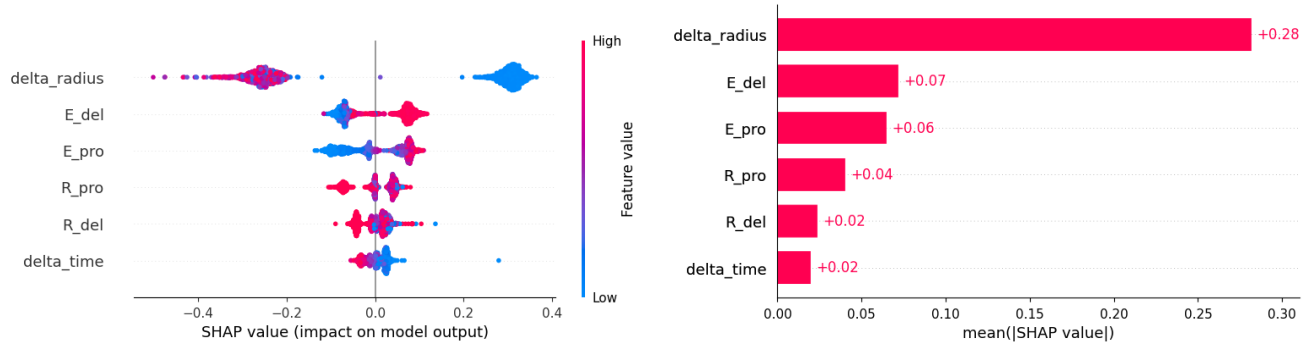
Formally, the Shapley value ϕ_i of a feature i is computed as follows:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \tag{3.1}$$

where N is the set of all features, S ranges over all subsets of N that do not contain feature i, $|S|$ is the cardinality of set S (i.e., the number of features in S), $f(S)$ is the prediction of the model made with feature set S. The term $|S|!(N - |S| - 1)!/N!$ is the weight given to each subset, which corresponds to the number of times each subset will appear when considering all permutations of the features.

Based on the calculation of SHAP values, we can construct visualizations that aid in analyzing and understanding how the model has learned to differentiate between Inverse Beta Decay events and background events, contributing to model interpretability.

The presented graphs depict the importance of each feature used by the algorithm for learning, measured by calculating the mean of the SHAP values. On the left, we see a histogram where the x-axis represents the mean absolute SHAP value for each feature. The first key characteristic of the model is evident here: the feature with the most importance in classification is ΔR. Referring back to the graphs 3.1, it was already observable that ΔR is the feature that separates the IBD class most distinctly from the BKG class. A clear separation is evident from $2500mm$ onwards, where BKG events prevail, while IBD events are more prevalent below $2500mm$.

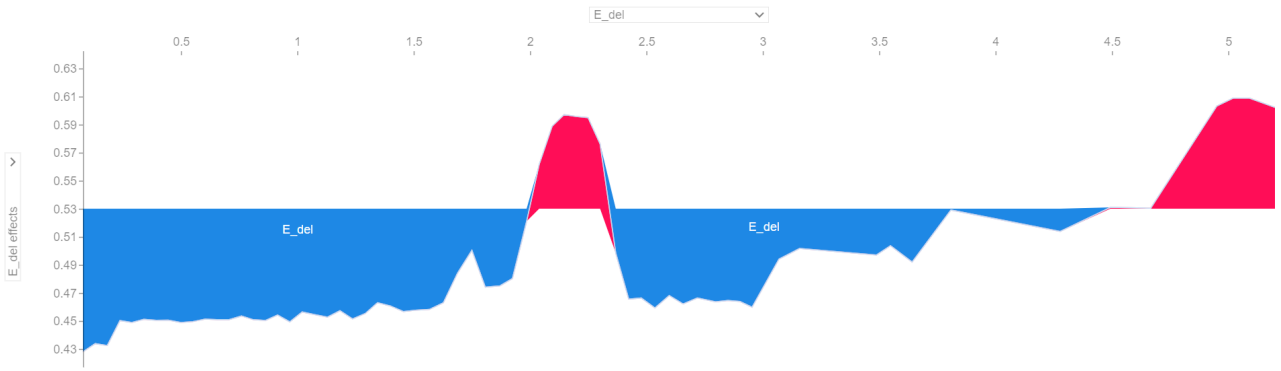


(a) ΔR cut **(b)** Δt cut

The importance of this feature is further highlighted in the right-hand graph, where the x-axis represents the mean absolute SHAP value, and the y-axis represents the various features. Two distinct data clusters for the ΔR feature are clearly visible. For high values of this feature, the algorithm returns a negative SHAP value, which translates into identification as BKG events, as expected. For lower values of the feature, positive SHAP values are returned, corresponding to events correctly identified as IBD. It's also worth noting that the clusters are perfectly separated, indicating that the algorithm is very confident in labeling events based on this feature.

Second in order of importance, with a SHAP value approximately four times smaller than that of ΔR, is "E_{del}", the energy of the delayed event, or electronic capture. Comparing with the feature histogram, it's clear that most BKG data occupy the initial part of the histogram, thus at lower energies, and the algorithm has learned to determine that for lower delayed signal energies, the event is classified as a BKG event. For slightly higher energies, given the presence of characteristic peaks that significantly increase the counts of IBD events, the algorithm learns to correctly determine an IBD event.

Delving deeper into the analysis of this feature, a plot was created where the x-axis represents individual events, and the y-axis represents the effect that each event had on the E_{del} feature. It is observed that for events in the range [1.9, 2.3] MeV and [4.9, 5.1] MeV, the algorithm has learned to perfectly distinguish the characteristic peaks of neutron capture compared to all background (BKG) events.



Regarding the E_{pro} feature, for values in the range [0, 1] MeV, the histogram is predominantly occupied by BKG events, and as seen from the summary plot, these are correctly

identified by the algorithm. However, for prompt signals with energies within the positron-like spectrum, the algorithm identifies these events as IBDs. The features R_{pro}, R_{del}, and Δt do not contribute significantly to the algorithm's ability to discern between the two classes from their distribution, as there are no clear differences between the feature histograms for IBD and BKG.

In conclusion, the SHAP values and the corresponding plots provide a valuable tool for understanding the decision-making process of the model, highlighting the importance of different features in the classification task. The model appears to perform well in distinguishing between IBD and BKG events, particularly using the Δt and E_{del} features. However, further analysis and improvements could be made, especially considering the features that currently contribute less to the classification.

3.3.3 PyThorch

3.4 Conclusion

	Manual Cut Algorithm	**BDT Algorithm**
Radioactivity	Efficiency: 99.9973% Purity: 100%	Efficiency: 99.997684% Purity: 100%
True IBDs	Efficiency: 97.734% Purity:100%	Efficiency: 99.997616% Purity: 100%

References

[Kaj16] Takaaki Kajita. “Nobel Lecture: Discovery of atmospheric neutrino oscillations”. In: *Reviews of Modern Physics* 88.3 (July 2016). DOI: 10.1103/revmodphys.88.030501.