Events Pattern Recognition and Image Reconstruction in Compton Camera for Proton Therapy Monitoring

Doctoral Thesis
by

Majid Kazemi Kozani

A thesis submitted in fulfilment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Physics

Supervisor

Professor, Andrzej Magiera

Kraków
November 2021

Oświadczenie

Ja niżej podpisany Majid Kazemi Kozani (nr indeksu: 1142425) doktorant Wydziału Fizyki, Astronomii i Informatyki Stosowanej Uniwersytetu Jagiellońskiego oświadczam, że przedłożona przeze mnie rozprawa doktorska pt. *„Events Pattern Recognition and Image Reconstruction in Compton Camera for Proton Therapy Monitoring"* jest oryginalna i przedstawia wyniki badań wykonanych przeze mnie osobiście, pod kierunkiem prof. dr. hab. Andrzej Magiera. Pracę napisałem samodzielnie.

Oświadczam, że moja rozprawa doktorska została opracowana zgodnie z Ustawą o prawie autorskim i prawach pokrewnych z dnia 4 lutego 1994 r. (Dziennik Ustaw 1994 nr 24 poz. 83 wraz z późniejszymi zmianami).

Jestem świadom, że niezgodność niniejszego oświadczenia z prawdą ujawniona w dowolnym czasie, niezależnie od skutków prawnych wynikających z ww. ustawy, może spowodować unieważnienie stopnia nabytego na podstawie tej rozprawy.

Kraków, dnia ....

(podpis doktoranta)

Abstract

Hadron therapy is an acknowledged technique for cancer tumor treatment based on electromagnetic interaction of ions with matter. Unlike conventional radiotherapy, charged particles, such as protons and heavy ions deposit the maximum energy (Bragg peak) at the end of their trajectory as they penetrate matter. This property allows for the effective destruction of the tumor during the treatment. Most of the energy is deposited in the tumor with a significantly reduced dose to the surrounding healthy tissue. This advantage made hadron therapy a better therapeutic option compared to conventional radiotherapy. However, due to inter- and intra-fractional anatomical changes of the human body, it is necessary to vary the proton and heavy ion beam energies by applying relatively large safety margins in treatment plans. Therefore, the development of online monitoring during hadron therapy is one of the most important challenges, which may result in a reduction of the safety margins, leading to more effective treatment. Since all primary protons are stopped within the tissue, the only way to control dose deposition is to detect secondary particles, such as prompt gamma emitted immediately after nuclear reactions take place. The good correlation between the prompt gamma emission and the range of the incident proton beam made prompt gamma detection as one of the most promising options for hadron therapy monitoring.

The aim of the SiFi-CC project is to develop an online method for monitoring dose distribution in proton therapy. The method under development is based on the detection of prompt gamma radiation emitted from a patient's body during irradiation. For this purpose, a prototype of a Compton camera is constructed, made of scintillating fibers of heavy materials. This work used a predetermined design of the detector and advanced Monte-Carlo simulations of detector responses made with the use of Geant4 software.

My research, as a member of SiFi-CC research group, was concentrated on the development of programming frameworks for machine learning and image reconstruction. In order to obtain a detailed insight into the expected response of the SiFi-CC detector, it is crucial to implement such software frameworks already in the initial phase.

The most important part of the dissertation was devoted to a software framework design for the classification of pseudo-data generated by the Geant4 simulations. The detection system response simulations were performed for a spot-scanning with a 180 MeV proton beam impinging on the PMMA phantom. The gamma radiation emitted from the excited nuclei interacted with the material of the proposed SiFi-CC detector, providing information about the coordinates of the interaction and the deposited energy. Machine learning software based on the TMVA multivariate data analysis toolkit developed at CERN was used to analyze the obtained pseudo-data. Given the probabilities of the different gamma radiation interaction processes in the detector, combinations of interaction positions and deposited energies are assigned either to Compton scattering or to background. This study investigated the performance of three different machine learning models, including Boosted Decision Tree (BDT), Multilayer Perceptron (MLP) Neural Network and k-Nearest Neighbors (k-NN). It turned out that the BDT classifier excels in signal and background

separation compared to the other two models. Therefore, in the analysis phase, the BDT classifier was used to identify Compton scattering events. The performance stability and robustness of the BDT classifier were studied during the analysis phase using well-known evaluation metrics, such as recall, efficiency and purity.

The second part of the dissertation concerns the reconstruction of the prompt gamma emission profile, containing information on the deposited dose distribution in a proton therapy treatment. For this purpose, software was prepared based on the algorithm of List-Mode Maximum Likelihood Estimation Maximization (LM-MLEM). Then, the LM-MLEM algorithm was used to reconstruct the gamma source position. This allowed to determine the distal edge of the Bragg peak for the proton beam stopped within the phantom.

Thanks to the use of machine learning, a very good selection of Compton scattering events in relation to the background events was obtained. As a result, a very good agreement was achieved between the position of the reconstructed distal edge and the position of this edge obtained in the simulations. Finally, it has been shown that it is possible to determine the position of the distal edge with a resolution of 3.5 mm FWHM. The results of this work show that the use of the SiFi-CC prototype is a promising approach to determine the location of the distal edge of the Bragg peak. The methods developed in this work also allow to optimize the configuration of the Compton camera prototype, which may allow for even better determination of the location of the distal edge of the Bragg peak. In addition, the developed software can also be applied to real data measured with the SiFi-CC detector. This should allow the proton therapy to be accurately monitored, leading to a reduction in side effects.

Dedicated to my wife and parents.

Acknowledgments

I would like to thank Aleksandra Wrońska, without whom this project would have never come to fruition. Andrzej Magiera for his patience, guidance, and insight over the past four years. Katarzyna Rusiecka and Jonas Kasper, who have provided their generous insight into my project development.

I also would like to thank my Father and Mother who have successfully injected in me the excitement in sciences, and always enormously support me in my life. I am also grateful to both of my Brothers for all the nice moments we have spent together and strongly motivating me to achieve my goals.

And, most importantly I would like to express my deep gratitude to my beloved Wife Maryam who is my inexhaustible source of support, inspiration, motivation, and love.

This work was supported by the Polish National Science Centre (grant number 2017/26/E/ST2/00618) and the financial resources for research or development work and related tasks for the development of doctoral students, DSC and MNS grants with numbers of K/DSC/004987 and N17/MNW/000009, respectively.

Contents

Glossary

AUROC Area Under Receiver Operating Characteristic.

BDT Boosted Decision Tree.

BDTG Gradient Boosted Decision Tree.

FOV Field Of View.

IDD Inter-Detector Distance.

k-NN k-Nearest Neighbour.

LM-MLEM List Mode Maximum Likelihood Expectation Maximization.

MLEM Maximum Likelihood Expectation Maximization.

MLP Multilayer Perceptron.

PG Prompt Gamma.

PSF Point Spread Function.

RE Recoil Electron.

ROC Receiver Operating Characteristic.

SiFi-CC SiPMs and scintillating Fiber-based Compton Camera.

SP Scattered Photon.

SSD Source-Scatterer Distance.

List of Figures

List of Tables

Chapter 1

Introduction

Based on the Global Cancer Observatory's report [1], cancer is the second most cause of death in Europe, with more than 1.9 million death records within 2020 in Europe. Moreover, more than 4 million new cases has been registered in Europe up to now. Because of these all facts, cancer treatment has been an ongoing research field for decades.

Around 50% of all cancer patients receive radiation treatment as one of the well-known methods of treating cancer tumors [2]. In traditional radiation therapy, a beam of high energy photons is used to kill the tumors cells [3]. In this method, the photon's energy falls exponentially as it goes to the deeper organs. Therefore, the tumors will not receive the dose effectively and a large volume of healthy tissues is irradiated. On the contrary, ion radiation therapy is a different approach where protons or heavy ions are used rather than photons to maximize the radiation dose to cancer tumors while minimizing the exposure to other tissues [4]. The most important advantage in ion radiation therapy is that they release most of their energy at a certain point called Bragg peak [5]. This allows us to precisely kill the tumor cells with minimum radiation dose exposure to surrounding tissues. Since a small deviation in locating the Bragg peak may have a significant effect on the tissues surrounding the tumor, it is crucial to locate the Bragg peak position accurately [6].

Currently the only technique which is used in clinical application is positron emission tomography (PET) imaging [7]. This approach is based on the detection of photons generated by the annihilation of delayed positrons emitted from fragments

such as ^{11}C or ^{15}O produced during the therapy. There are two strategies including in-beam and offline PET measurements applied in clinical centers [8]. However, in both cases, the reconstructed image quality of positron activity distribution is degraded because of the low effective activity (much lower than in standard PET examination), and the washout effect caused by metabolism [9]. Therefore, online dose and range monitoring of ion beams during the treatment fraction is not possible with the current technology [10]. Another promising approach based on detection of prompt gamma (PG)s emitted in nuclear reactions of ions with the atomic nuclei of tissue is under investigation. As they are emitted almost immediately (within times below ns), their distribution is not affected by physiological processes (there is no washout effect). Therefore, it could provide an online monitoring for dose distribution during hadron therapy.

The Silicon Photo Multiplier (SiPM) and scintillating Fiber-based Compton Camera (SiFi-CC) is a concept for an online monitoring tool for the Bragg peak position. It is a joint collaboration effort between working groups from the Jagiellonian University in Poland and RWTH Aachen University in Germany. The collaboration is developing the SiFi-CC detection system to reconstruct the positions of PG emissions during proton therapy, leading to measure Bragg peak location precisely [11, 12].

Along with the Compton interactions, many other gamma interactions occur within the SiFi-CC detector. This dissertation aims to introduce a machine learning model to identify the Compton events among many gamma interactions from simulated data of the SiFi-CC. Then, an image reconstruction algorithm based on the list-mode Maximum Likelihood Expectation Maximization (LM-MLEM) is used to reconstruct the events selected by the model, locating consequently the Bragg peak distal falloff position. This study could help assess the performance and optimize the feasible geometric configuration of SiFi-CC allowing for more precise monitoring of proton therapy. These frameworks can be also used in the analysis of real data measured during the treatment.

The dissertation consists of four chapters which I introduce them shortly here. In *Theoretical background* chapter, a literature review about the history of ion beam

therapy and the physics behind the Compton effect and its capabilities in ion therapy monitoring are introduced. Finally, a full description about image reconstruction and machine learning frameworks used in this project is provided. In *Materials and Methods* chapter, I introduce the detection setup simulation, the development of machine learning, and the image reconstruction frameworks implemented in this work. Later, the results of my work are presented in *Results* chapter. Finally, in the last chapter (*Discussion and Conclusions*), I discuss results obtained via the proposed methods and the feasibility of the proposed Compton camera in clinical applications.

"if I have seen further it is by standing on the shoulders of Giants."

Isaac Newton, 1675

Chapter 2

Theoretical background

This chapter introduces the theoretical background needed for this thesis. A brief introduction of the ion beam therapy history, the Compton effect physical aspects, and the simulation framework is presented. Finally, an in-depth introduction about the machine learning and image reconstruction frameworks is presented.

2.1 Principles of ion beam therapy

Ion beam therapy for treating cancer patients is widely considered a superior form of radiotherapy compared to conventional treatments using high energy X-rays. This is due to the characteristic dose deposition induced by protons, which exhibits a sharp peak near the stopping point, the so-called Bragg peak, resulting in a decrease of the volume of healthy tissues irradiated to intermediate and low doses [13, 14]. Figure 2-1 illustrates that carbon ions and protons deposit a maximum of energy at the end of their path within the matter called the Bragg peak, while the dose deposited by photons decreases exponentially. Another advantage of ion beam therapy compared to the conventional radiotherapy is to deliver a homogeneous dose to the tumor with a largely reduced dose to the surrounding tissue using only a few irradiation fields (see Figure 2-2). We observe indeed a large reduction in the total dose to the healthy tissues and organs at risk.

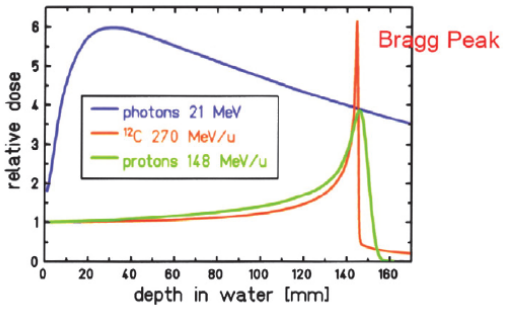


Figure 2-1: Depth dose distribution for photons, protons and carbon ions in water [15].

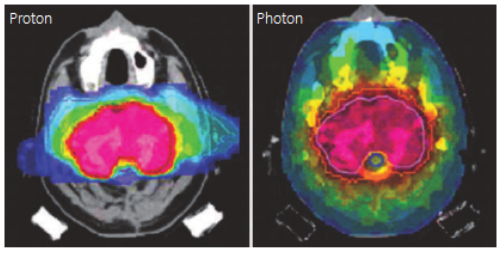


Figure 2-2: Comparison of treatment plans for a large target volume in the base of the skull. Left: Plan for proton beam (two fields). Right: Plan for photon therapy (nine fields) [16].

Nevertheless, ion therapy requires more precise application in the therapeutic stage due to the presence of the Bragg peak. It means that a small deviation can have a significant destructive impact, especially in the treatment of tumors located in critical parts of the body, such as the brain. Minor patient positioning errors, patient anatomical changes, and translation of computed tomography (CT) to water equivalent units may introduce uncertainties between the treatment plan and the actually applied dose distribution [17]. Safety margins ranging from a few millimeters up to over a centimeter depending the tumor's location are applied during ion therapy. To reduce safety margins and destroy tumor precisely, it is very important to verify the Bragg peak position, preferably even during treatment, to be sure that the dose is delivered as planned [18]. Therefore, the online monitoring tools are needed to control the deposited dose distribution during ion therapy treatments [14].

Several approaches have been implemented for in vivo monitoring of proton therapy dose delivery and proton beam range verification. Positron emission tomography (PET) is currently the only method used for dose verification, which involves the detection of 511 keV gamma rays resulting from positron emission decay of proton-induced radioactive nuclides such as ^{11}C, ^{13}N, and ^{15}O [7]. However, this approach is suffering from the low quality of the reconstructed activity images due to low effective activity within the patient body and the washout effect caused by physiological processes. Another promising candidate for dose monitoring is detection of PGs emitted in nuclear reactions of protons with tissue's atomic nuclei [17]. It has

been observed that the number of PG is much larger than the number of emissions resulting from PET isotope decay [19]. Furthermore, the absence of washout effects in PG measurement [20] and well correlation between the proton range and PG distribution [21–24] are other important advantages which make measuring PGs as a viable method for clinical application.

2.2 Compton Effect

The Compton effect is an incoherent scattering of a photon on an electron. The photon with an incident energy E_0 is scattered at an angle θ and its energy E_1 after scattering is calculated as follows:

$$E_1 = \frac{E_0}{1 + \frac{E_0}{m_e c^2}(1 - \cos\theta)}, \tag{2.1}$$

where $m_e c^2$ is the electron rest energy. The probability distribution of the Compton scattering polar angle θ of the photon on a free electron is described by the Klein-Nishina cross section given in eq. (2.2).

$$\frac{d\sigma_e}{d\Omega} = r_0^2 \frac{1 + \cos^2\theta}{2(1 + \alpha(1 - \cos\theta))^2} \left(1 + \frac{\alpha^2(1 - \cos\theta)^2}{[1 + \alpha(1 - \cos\theta)](1 + \cos^2\theta)}\right), \tag{2.2}$$

where $\alpha = \frac{E_0}{m_e c^2}$ and r_0 is the classical electron radius given by $r_0 = \frac{e^2}{4\pi\varepsilon_0 m_e c^2}$, e is the elementary charge, ε_0 is the vacuum permittivity and $d\Omega$ is an infinitesimal solid angle. As it is shown in Figure 2-3, at lower energy, the forward scattering and back-scattering probabilities are comparable. On the contrary, as the energy increases, the probability of forward scattering is higher.

From eq. (2.1) and eq. (2.2), it is assumed that the struck electron is unbound and at rest. However, the electron binding energy can be included in the differential cross-section. In this way, the Klein-Nishina formula is multiplied by function D called the incoherent scattering function as a correction for the Klein-Nishina cross section. This correction function D depends on the transferred momentum to the electron after Compton scattering, and the atomic number of target.

The function D decreases the Compton scattering cross section at small scattering angles, and increases it at large angles. The presence of the struck electron's momentum introduces an uncertainty in the energy spectrum of the scattered photons. Therefore, for a given scattering angle, the value of the scattered photon energy E_1 is not unique, leading to the Doppler broadening effect which is an interesting research topic in the electronic structure of atoms, molecules and solids.

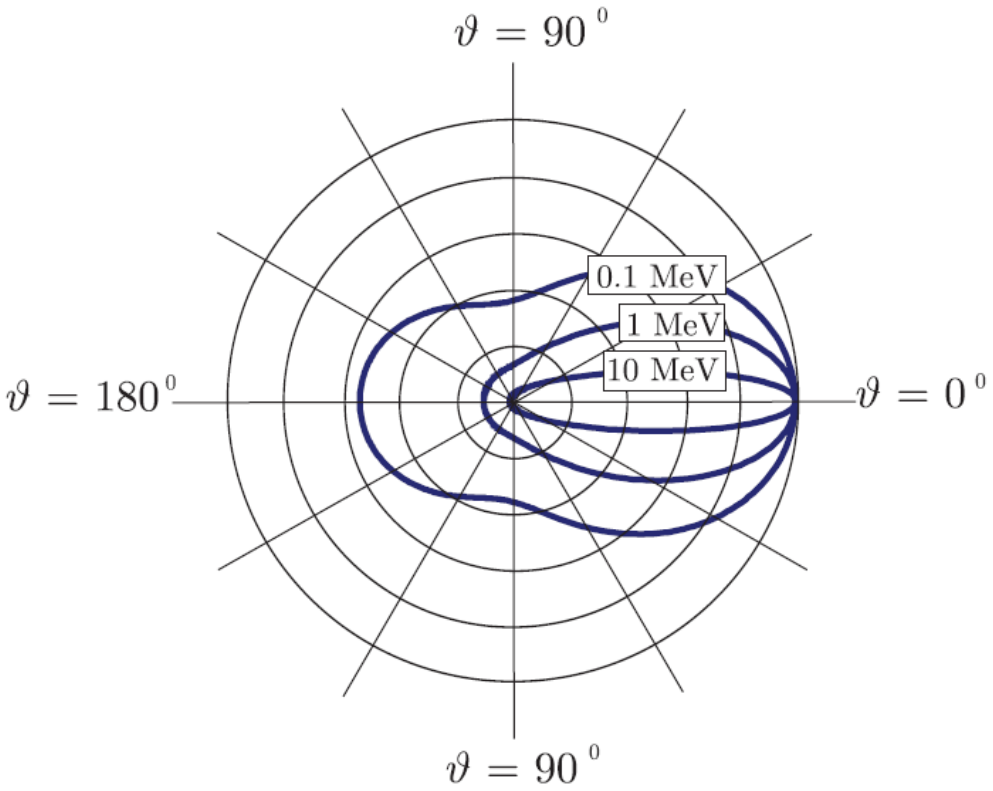


Figure 2-3: Klein-Nishina cross-section as a function of the Compton scattering angle θ for three different photon energies. The higher the energy is, the smaller the average Compton scatter angle is [25].

2.3 Compton Camera

Compton camera generally consists of two detection modules. A PG interacts via a Compton effect in the first module "the scatterer", and subsequent interaction of the scattered photon occurs in the second module "the absorber". According to eq. (2.1),

the deposited energies in the scatterer E_{scat} and in the absorber E_{abs} are calculated as follows:

$$E_{\text{scat}} = E_0 - E_1, \tag{2.3}$$

$$E_{\text{abs}} = E_0 - E_{\text{scat}} = E_1. \tag{2.4}$$

The eq. (2.4) is valid only when the scattered photon is fully absorbed in the absorber. Using the deposited energies and the positions of the interactions make it possible to reconstruct the cone containing its incident trajectory accurately (see Figure 2-4). The reconstructed cone's apex is the interaction points in the scatterer. The cone's axis is formed by the interaction points in the scatterer and in the absorber. The source of the photon is limited to the cone's surface determined by the scattering angle θ. Finally, the intersection of several of these cones surfaces enables reconstruction of PG emission distribution.

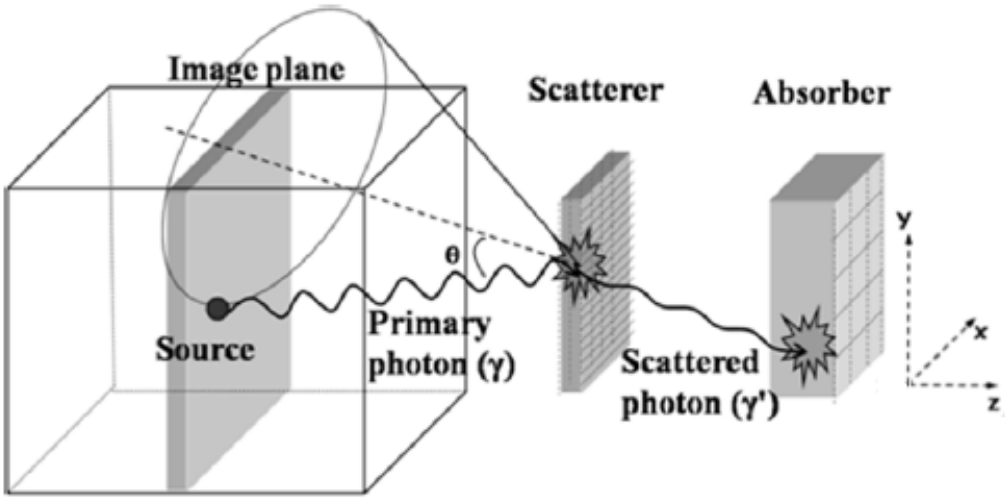


Figure 2-4: Principle of a Compton camera [26], see text for more details.

2.4 Geant4 Simulation

Geant4 is a popular toolkit, which was developed through an international collaboration. It allows us to simulate the passage of particles through matter by using a

Monte-Carlo method and is used in the field of ion beam therapy for the simulation of both proton and carbon beams. It is a common practice in experimental science to design and simulate complex detectors for studying their behaviors and properties [27–29].

The proposed Compton camera design has been implemented and simulated with Geant4 in order to have a detailed insight into the response expected from our prototype SiFi-CC detector. Moreover, it is possible to simulate and study different possible geometries, building materials, and arrangements before actually building them [11, 29]. As a result, different setups of the SiFi-CC prototype are analyzed and optimized faster and at a lower cost. The Geant4 simulations have been performed as a part of another Ph.D. project [30].

2.5 Image Reconstruction

Several different image reconstruction methods have been investigated since the Compton camera was proposed for the gamma imaging of sources in medicine [31]. There are two main approaches including analytical and iterative algorithms [32–35].

The analytical methods aim to find an analytical solution for the detection model. Then, the solution is discretized and often solved using algorithms such as filtered back-projection and linograms. Two significant limitations of this approach refer to the fact that many imaging systems can not be reliably modeled, and the solution may be too difficult to be derived analytically.

In the iterative methods, the detection model is firstly discretized. Then, the solution is provided using iterating algorithms. In this approach, a greater number of imaging devices can be modeled. However, the uniqueness and the exactness of the solution is lost. Moreover, an iterative process can be very computationally intensive.

The main challenge for both kind of approaches is the computational power needed. Here, we first present an overview of the back-projection method and then describe the List Mode Maximum Likelihood Expectation Maximization (LM-MLEM) algorithm used for Compton imaging.

2.5.1 Back-Projection

The back-projection of Compton cones calculated from the coincidence events is the simplest analytical image reconstruction method to the Compton imaging problem. Assuming the complete energy deposition in scatterer and absorber due to Compton scattering (see eq. (2.4)), one can reconstruct a cone surface on which the source location lies.

A simple back-projection reconstruction method consists of the segmentation of the source space into voxels and the determination of the number of cones intersecting each voxel in image space volume. To simplify the complexity of the intersections' computation of a cone surface with a 3D segmented image space, we use the method adapted from [36] for the back-projection of cones on a 2D segmented image plane. The 3D image reconstruction can be obtained by applying the method to each individual image plane. The plane-cone intersection problem can be solved analytically. The intersection of the Compton cone and the image volume is solved by:

$$\cos\theta = \frac{(\overrightarrow{r} - \overrightarrow{r_s}) \cdot (\overrightarrow{r_s} - \overrightarrow{r_a})}{|\overrightarrow{r} - \overrightarrow{r_s}| \cdot |\overrightarrow{r_s} - \overrightarrow{r_a}|}. \tag{2.5}$$

where the $\overrightarrow{r}$ vector corresponds to all possible positions of the photon source, $\overrightarrow{r_s}$ vector is the Compton cone vertex in the scatterer, and $\overrightarrow{r_a}$ vector is the interaction position of the scattered photon in the absorber. The $(\overrightarrow{r_s} - \overrightarrow{r_a})$ vector forms the cone axis, and $\overrightarrow{n} = \frac{(\overrightarrow{r_s} - \overrightarrow{r_a})}{|\overrightarrow{r_s} - \overrightarrow{r_a}|}$ is its axis direction vector. Therefore, the final form of eq. (2.5) will be given by:

$$\cos^2\theta \cdot |\overrightarrow{r} - \overrightarrow{r_s}|^2 = (\overrightarrow{n} \cdot (\overrightarrow{r} - \overrightarrow{r_s}))^2. \tag{2.6}$$

Solution of eq. (2.6) for each event define a cone surface on which all possible photon source points lie. The image volume is divided into many slices, then the intersection of the cone with each slice can be calculated. The intersection of a cone with a plane is an ellipse. Therefore, three cases are possible as shown in figure Figure 2-5 : i) the conic section is completely inside the image plane, ii) the conic

section is completely outside the image plane, iii) conic section arcs are inside the image plane.

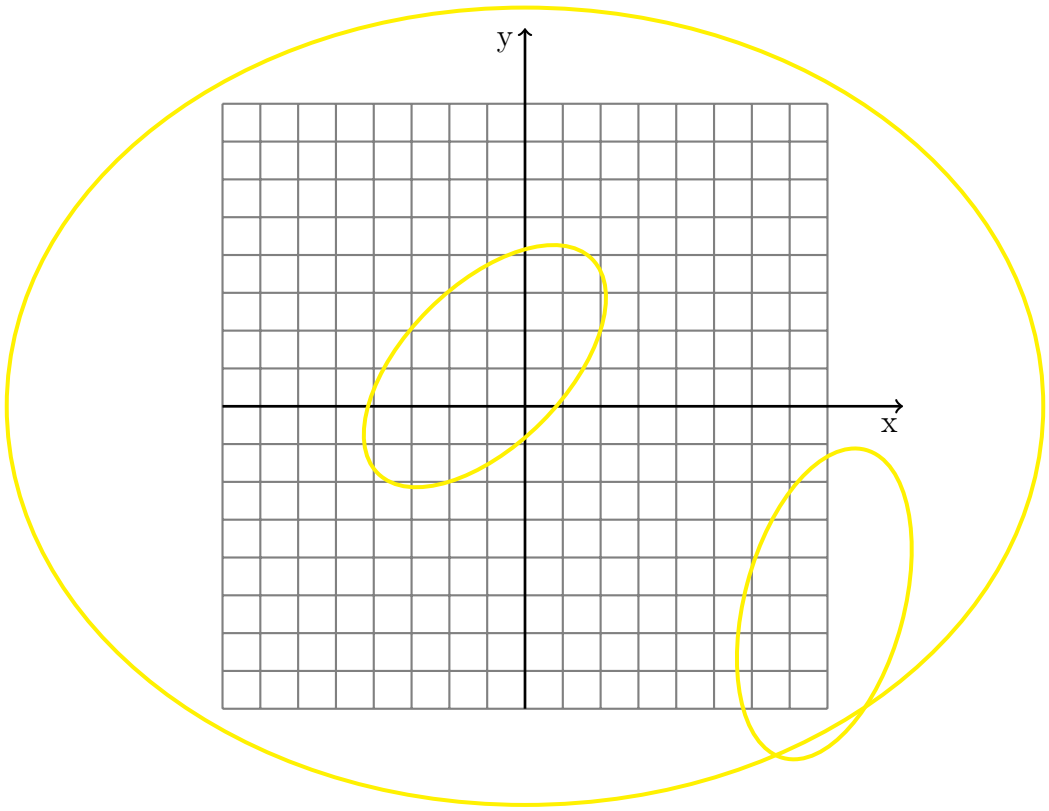


Figure 2-5: Illustration of the three possible cases of conic section intersecting the image plane. The conic section can be inside the image plane, partially included in or outside of the image plane.

In this method, the intersections of the ellipses are computed with the the edges of the segmented image plane, depending on whether they are on a vertical or a horizontal border. Next, the pixels containing the intersections are searched for and the length of the track in each pixel is calculated. In this way, an estimate of the reconstructed conic section is obtained (see Figure 2-6). This process continues until the last event is processed. Therefore, the back-projections of all events are weighted by segment length and these weights are summed up in each pixel to show the probability of the source position distribution. In section 3.2.2, it will be explained in more detail.

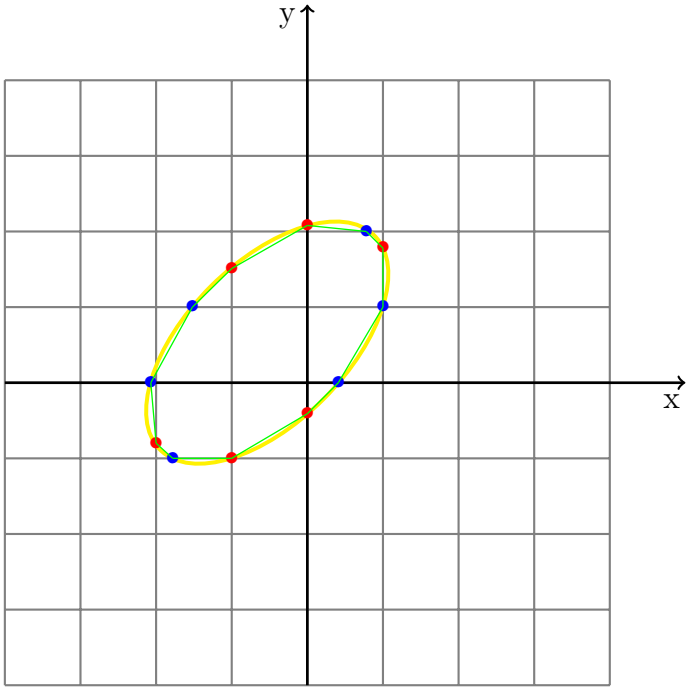


Figure 2-6: Depiction of an example of a 2D tracked conic section (exaggerated view). Here, the xy-plane is intersected by a Compton cone. The blue and red dots represent the intersection points between the grid and the conic section on the horizontal and vertical borders, respectively. The green lines are the approximated arc length in each intersected pixel of the image plane.

2.5.2 LM-MLEM

Most of the iterative image reconstruction methods are based on the Maximum Likelihood Expectation Maximization (MLEM) [36–38]. In MLEM method, the system matrix of a Compton camera consists of all possible combinations of bins in scatterer and absorber which are equal to the total number of detector bins. Therefore, for a Compton camera with a large number of bins, the MLEM needs an unrealistic large computer memory and it will be time-consuming. Due to this fact, the list-mode MLEM (LM-MLEM) method was proposed and implemented for Compton camera by Wilderman [36, 39]. It provides a computational advantage in comparison with the binned MLEM approach [36, 40–42].

Here, the number of system matrix elements depends on the considered field of view (FOV) dimensions and the number of detected coincidence events. In other words, the system matrix is a matrix with rows of total number of detected events and columns of the total number of FOV dimensions. The LM-MLEM begins with an

initial approximation of the source distribution I_j^0 obtained from the segment length weighted back-projection. Then, it can be subsequently improved by applying the recursive equation as follows:

$$I_j^{n+1} = I_j^n . C_j^n \tag{2.7}$$

$$C_j^n = \frac{1}{s_j} \sum_{i=1}^{N_{events}} \frac{t_{ij}}{\sum_{k=1}^{M_i} t_{ik} I_k}. \tag{2.8}$$

where N_{events} is the total number of detected events, s_j is the detection sensitivity (i.e. the probability for FOV voxel j which an event can be successfully detected) which compensates for the loss of emitted photons in distant voxels. t_{ij} refers to the element of the Compton camera system matrix and represents the transition probability (i.e. segment length weight in voxel j for the detected event i). M_i is the number of voxels in FOV intersected by the back-projected cone of event i. As it is given in the eq. (2.8), the inner sum belongs to M_i image voxels intersected by each back-projected cone (i.e. the forward projection of an image estimate I onto the detector) and definitely is different from one detected event to another one. For each iteration, the outer sum is over the N_{events} detected events. In other words, in each iteration, the back-projection of data is weighted with the forward projected data (the inner sum), leading consequently to an update correction for the image estimate [43]. Therefore, the method provides us with a boosted image estimate after n iterations using the multiplicative correction image C_j^n as depicted in Figure 2-7.

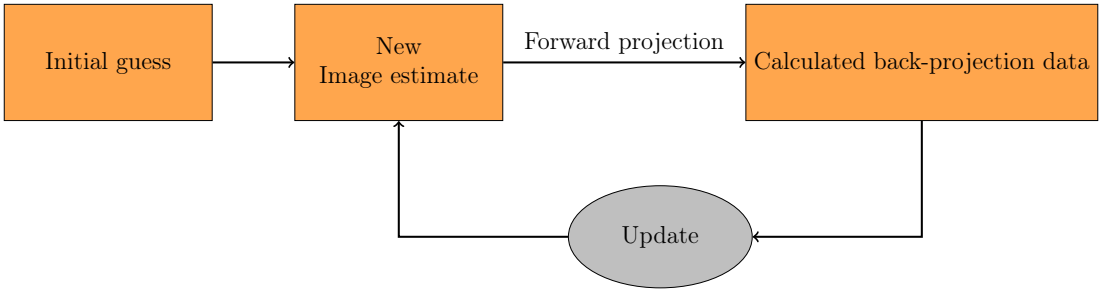


Figure 2-7: Illustration of the LM-MLEM algorithm.

2.6 Machine Learning

The concept of machine learning comes from this question: could a computer automatically learn the rules on its own to carry out a specific task?

This question familiarizes us with a new programming paradigm. Unlike the classical programming in which a programmer should define a set of rules over the data to get the desired answers, the machine learning system can be trained by giving it the data sample and the expected answers. The machine learning model learns the rules needed without being explicitly programmed (see Figure 2-8). The trained model can be applied to new data to reproduce original results [44, 45].

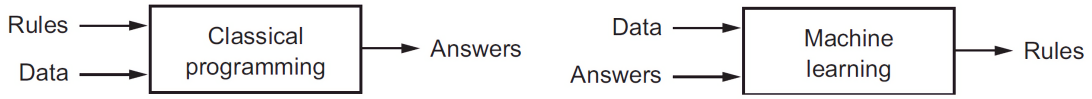


Figure 2-8: The difference between classical programming approach in which rules be explicitly defined by a user (left), and a machine learning model which produces the rules from a training data sample and the expected answers (right) [44].

Generally, for training a machine learning model on a specific task, five components are required as follows [46].

1. Input data samples. These data samples can be in any form, like simple tabular data for patients, or the kinematic information of all secondaries detected in proton therapy.

2. Expected output. These output records are associated with each input data sample. For example, the expected output for dose range monitoring in proton therapy might be the event's label like PG detected (signals) or other secondary (background) information.

3. Loss function. It measures the model's ability to predict the right output and how good its performance is.

4. Optimizer. The model will update itself based on the data it sees and its loss function.

5. Metrics to monitor during training and testing. They display the models' capabilities in a classification task especially when comparing different models' performances.

The first three items are the main to train a model and the other two are optional to tune and control the model. There are other types of machine learning models that do not need the expected output for training called unsupervised learning. They mainly find interesting transformations and important features of the input data for data analysis and visualization. Dimensionality reduction and clustering are well-known categories of unsupervised learning. More information could be found in [44]. In the following, the description of some different machine learning models which were used in this project are listed.

2.6.1 Boosted Decision Tree

A decision tree is a binary tree structure which classifies the input data sample as depicted in Figure 2-9. The main advantage for using a decision tree is that it is easy to follow and interpret. The questions about the input data with responses of (yes/no) are taken on one single variable (feature) repeatedly until a stop criterion like: depth of tree to grow, number of features to build a given tree and etc. is met [47]. In other words, a decision tree takes a set of input features and splits the input data recursively based on those features that are eventually classified as signal or background events, depending on the majority of training events that end up in the final leaf node. The decision trees suffer from inconsistency in statistical fluctuations when training input data sample. It causes the whole tree structure to be changed below the node with that problem. In other words, if two or more features exhibit similar separation power, a fluctuation in the training sample may cause the tree to grow and split on one of features, while each of those features could have been selected in case of no fluctuation. Therefore, the whole tree structure will change below this node, resulting in a different classifier response.

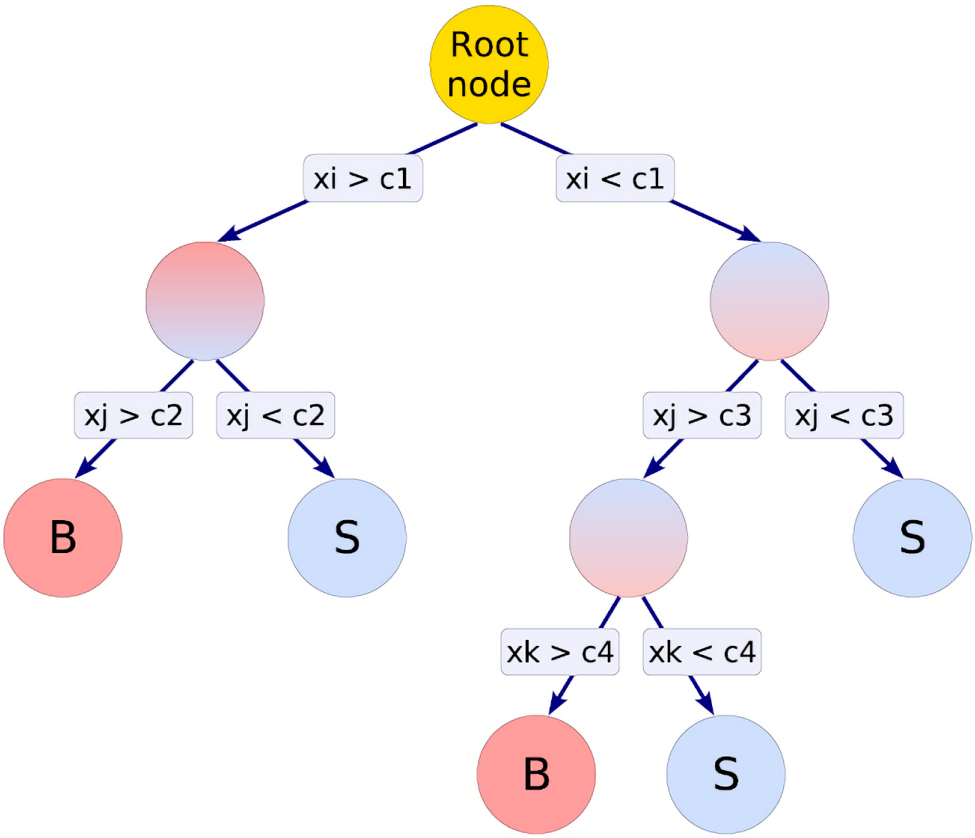


Figure 2-9: Schematic view of a decision tree. The starting point is the root node. Then a binary structure sequence splits using the discriminating features x_i, x_j, x_k applied to the input data. Each split uses the feature giving the best separation between signal and background when the node is being cut on. The same feature may be used several times, although others might not be used at all. The leaf nodes at the bottom end of the tree are labeled "S" for signal and "B" for background depending on the majority of events in the respective nodes [47].

To overcome such problem, a forest of decision trees can be constructed. All trees in the forest are derived from the same training sample, and then boosting i.e. a method which modifies the events' weights in the sample is applied which is recognized as the boosted decision tree (BDT) classifier. In such a way, for input features X, each tree's output $H_t(X)$ is given a weight w_t relative to its accuracy. The weighted sum output is:

$$Y'(X) = \sum_t w_t H_t(X). \tag{2.9}$$

where $Y'(X)$ is the ensemble true output. Then, the boosting procedure is employed to minimize the objective function $O(X)$:

$$O(X) = \sum_i L(Y_i, Y_i') + \sum_t \Omega(w_t). \tag{2.10}$$

where $L(Y_i, Y_i')$ is the loss function which shows the deviation between the true and the model prediction of the ith sample and $\Omega(w_t)$ is the regularization function that penalizes the complexity of the tth tree which can be defined as the number of proportional tree leaves and guard against overtraining [48, 49]. The $\Omega(w_t)$ function is defined by the available hyperparameters depending on the type of BDT classifier.

2.6.2 Artificial Neural Networks

An Artificial Neural Network is a collection of interconnected neurons (perceptrons) arranged in layers, with each neuron applying a linear data transformation to a given set of input data via weighting it. Then, it applies a non-linear transformation using *activation function* g to the linear transformation outcome eq. (2.11). During the training phase, a set of values for the weights (w) of all layers is found in the network. Therefore, a map is drawn from a space of input features X onto their associated output Y'.

$$Y' = g(wX + b). \tag{2.11}$$

where b is the bias term. The reason of applying the non-linear function to a neuron's output is to support the network in learning complex patterns. Otherwise, a neural network is a stack of linear transformations which eventually can be combined to a single linear transformation.

Neuron response function

The neuron response function ρ consists of two functions including *synapse function* κ and *neuron activation function* g, so that $\rho = \kappa \circ g$. The synapse function has the following forms: the sum, sum of square, and sum of absolute values of all available neurons in the network. The most frequently used neuron activation functions

for a single neuron output are *sigmoid* and *tanh* given by eqs. (2.12) and (2.13) respectively [50].

$$g(\hat{y}) = \frac{1}{1+e^{-\hat{y}}}. \tag{2.12}$$

$$g(\hat{y}) = \frac{e^{\hat{y}} - e^{-\hat{y}}}{e^{\hat{y}} + e^{-\hat{y}}}. \tag{2.13}$$

The modification of activation functions and their robustness and limitations are still a hot research area in machine learning [51, 52].

Multilayer Perceptron

There are several popular architectures of artificial neural networks. The simplest form is a Multilayer Perceptron (MLP). This architecture reduces the complexity of the neural network by arranging the neurons in layers and only allowing direct connections from a given layer to the following layer (see Figure 2-10). The first layer of a MLP is the input layer, the last one is the output layer, and all others are hidden layers.

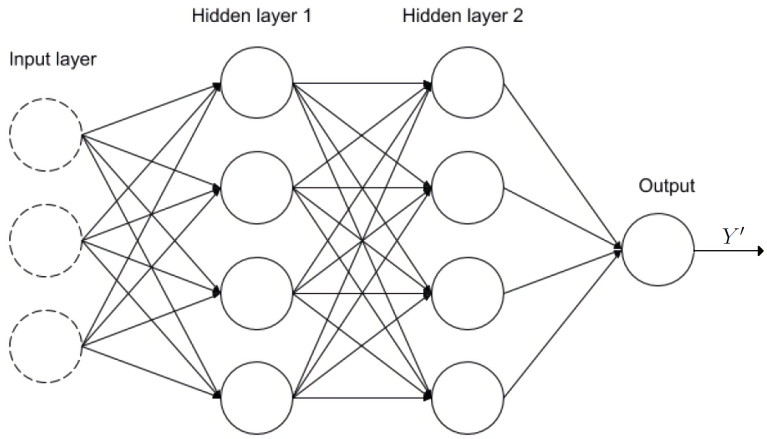


Figure 2-10: Multilayer Perceptron (MLP) with two hidden layers and a single neuron output layer.

For a classification problem with n_{var} input variables, the input layer consists of n_{var} neurons that hold the input values, $X_1, ..., X_{n_{var}}$, and one neuron in the output layer that holds the output variable Y'. For training each event e, the neural network output Y'_e is computed and compared to the true output $Y_e \in \{1, 0\}$ (in

classification 1 for signal events and 0 for background events). The loss function is applied to measure how far the predictions of the network are from what is expected (true target), defined by

$$L = \sum_{e=1}^{N} L_e = \sum_{e=1}^{N} \frac{1}{2}(Y'_e - Y_e)^2. \tag{2.14}$$

Eventually, this loss score, which shows how well the network has performed on the data sample, is computed. Moreover, this score is used as a feedback indicator for the optimizer to adjust the weights and make network outputs as close as the true targets (see Figure 2-11). The most common algorithm for adjusting the weights that optimize the neural network performance is the *Back propagation* algorithm. In this sense, the neural network computes the gradient of the loss score with respect to the network's weights. Then the optimizer updates a random set of weights $\mathbf{w}^\rho$ in the opposite direction from the gradient. The weights are updated by a certain positive factor called *learning rate* η (see eq. (2.15)). It is one of the available hyperparameters in the MLP model which should be tuned by users to avoid overtraining.

$$\mathbf{w}^{\rho+1} = \mathbf{w}^\rho - \eta \nabla_{\mathbf{w}} L. \tag{2.15}$$

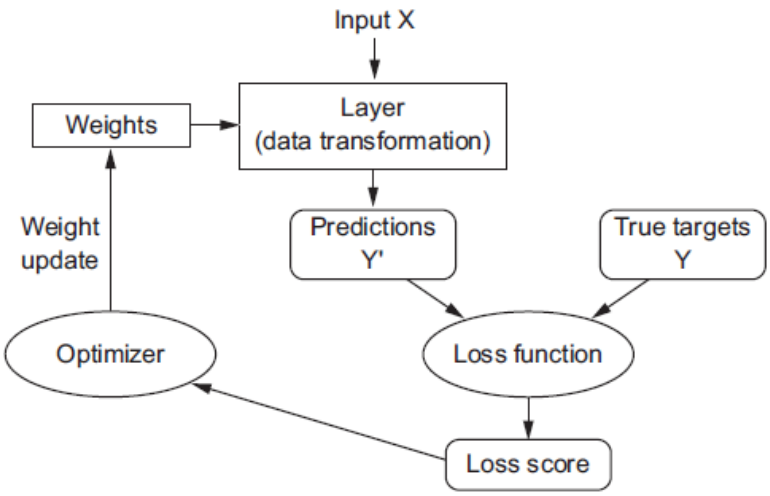


Figure 2-11: Illustration of an artificial neural network training phase, see text for more details [44].

2.6.3 k-Nearest Neighbour

The k-Nearest Neighbour (k-NN) algorithm does not need to build a predictive model from a training data set. Indeed, there is no actual training phase to make a prediction [53]. The k-NN algorithm captures the idea of similarity (sometimes called distance, proximity, or closeness) [54]. The k-NN classifier searches for k events from a training data set that are closest to an observed (test) event [47]. The distance is thereby measured using a metric function called the *Euclidean distance* given by

$$R = \left(\sum_{i=1}^{n_{var}} |x_i - y_i|^2 \right)^{\frac{1}{2}}. \tag{2.16}$$

where n_{var} is the number of input features used for the classification, x_i are the coordinates (features) of an event from a data sample and y_i are the variables of an observed (test) event. Figure 2-12 shows a case study of event classification with the k-NN algorithm.

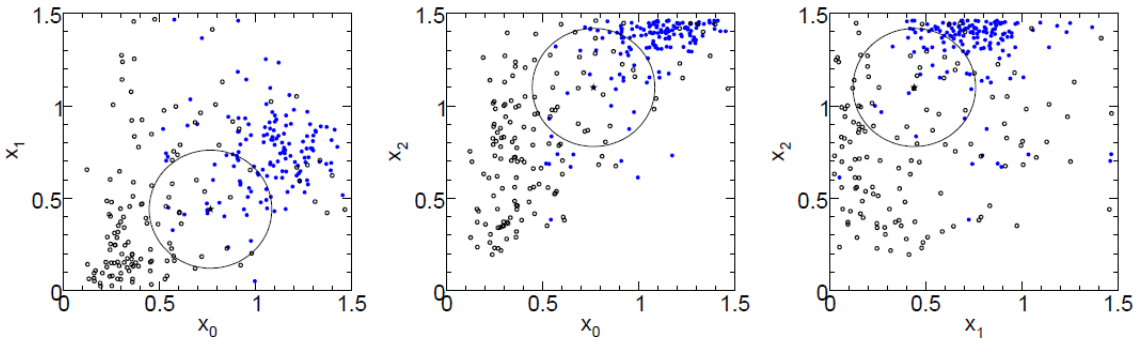


Figure 2-12: The k-NN algorithm for a case study with three discriminating input features. The three projections of the two-dimensional coordinate planes are drawn. The blue (open) circles are the signal (background) events. The k-NN algorithm searches for 20 nearest points in the nearest neighborhood (the bigger circle) of the query event, shown as a star. The nearest neighborhood counts 13 signal and 7 background data points so that the query event may be classified as a signal [47].

The values of k determines the behavior of the probability density function which does not represent its local behavior for large values of k. However, small values of k cause statistical fluctuations in the probability density estimate. Therefore, the

optimal value of k should be obtained by the trial and error method [47, 54, 55]. The relative probability that the test event is of signal type is given by

$$P_S = \frac{k_S}{k_S + k_B} = \frac{k_S}{k}. \tag{2.17}$$

where $k_{S(B)}$ is the number of signal (background) events in the training sample. When the input features have different units, each feature can contribute to the Euclidean metric depending on its distribution width. However, it can be compensated by a scaling factor $(1/w_i)$ applied to input feature i. w_i is the width of the x_i distribution for the combined sample of signal and background events. Then, the *Euclidean distance* is rescaled and given by

$$R_{rescaled} = \left(\sum_{i=1}^{n_{var}} \frac{1}{w_i^2} |x_i - y_i|^2 \right)^{\frac{1}{2}}. \tag{2.18}$$

Nevertheless, the k-NN's main disadvantage refers to becoming significantly slower as the number of input features increases.

Chapter 3

Materials and Methods

A novel design for a Compton camera is proposed by our group. The SiFi-CC project is a joint collaboration effort of two research groups from the Jagiellonian University and the RWTH Aachen University [12].

This chapter first presents the design of SiFi-CC detection system, the process of geometry optimization, and the simulation of the detector response. Finally, a detailed explanation of the machine learning approach used for the Compton camera imaging is presented.

3.1 SiFi-CC Detector Design

The SiFi-CC consists of two modules, the scatterer and the absorber. In the proposed design, both the scatterer and the absorber consists of thin, long fibers made of high-density inorganic scintillating material. The dimensions of each fiber are 1 mm×1 mm×100 mm. Each module has many layers of aligned fibers (see Figure 3-1). For the readout, the silicon photo-multipliers (SiPM) are coupled to the both ends of each fiber. Presently, the SiFi-CC prototype consisting of 4 layers, 16 fibers per layer, is under investigation [12].

In order to achieve high gamma detection efficiency in a few MeV energy range, a detection volume should be made of material characterized by high density and large effective atomic number. The material also should show good timing properties (fast decay time), thereby decreasing background from random coincidences.

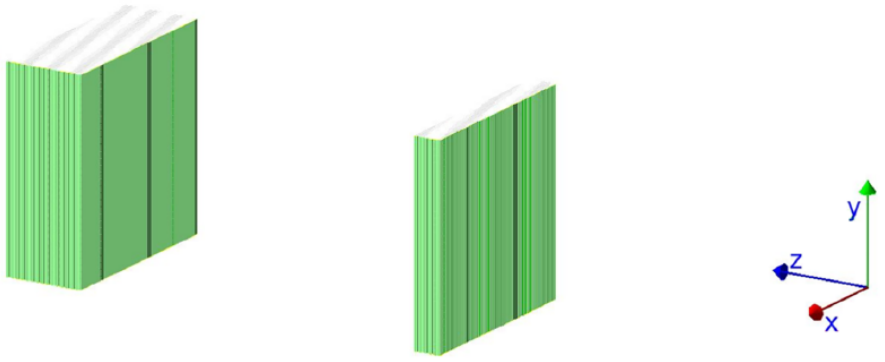


Figure 3-1: The proposed Compton camera setup. Each module consists of stacked fiber units arranged in layers [11].

Finally, the high granularity of the detector ensures a good gamma detection position resolution in two directions and a large rate capability. Different heavy inorganic scintillating materials like LYSO:Ce, LuAG:Ce, and recently developed GAGG:Ce were investigated. LYSO:Ce was chosen because of its very good performance in terms of light output, energy and time resolution, and also its widely availability compared to other inorganic scintillators [56].

The SiFi-CC uses the Compton effect's unique characteristics to determine the Compton cone. In an ideal case, a photon interacts via a Compton effect with an electron in the scatterer. Then the scattered photon traverses to the absorber and interacts there via photoelectric effect (see Figure 2-4). The energy deposits and positions of both interactions are measured and with such information, the Compton cone is determined. Finally, it is possible to find the origin of PG from the intersection of all the computed Compton cones (see section 2.2) [11, 56, 57]. However, such clear situation occurs very rarely. Therefore, it is necessary to apply the advanced methods of events classification described in section 4.2.

3.2 Optimization of SiFi-CC Geometry

The feasibility of using PGs for range verification of proton beams depends greatly on the design of a highly efficient Compton camera [13]. Therefore, an optimization of the proposed detector design by means of Monte Carlo simulations is required.

3.2.1 A Simple Compton Camera

In the initial stage of the optimization, the response of the detection setup to a point source in a realistic situation called the point spread function (PSF) [58] should be studied. To achieve this aim, the influence of geometrical parameters of a simple detection setup on the PSF was investigated by the simple simulation [59]. In this study, the scatterer and the absorber were represented by two planes. This approach neglected also the granularity of the detector and the properties of the scintillating material. Simulations were performed for a point-like, isotropic and mono-energetic 4.44 MeV (i.e. the most prominent PG emission energy correlated to the absorbed dose in proton therapy [24]) gamma source emitting photons into the half-space toward the detection planes. Each gamma emitted into the acceptance of the respective (scatterer) module was forced to undergo a single Compton scattering in the scatterer followed by a photoelectric effect in the absorber (see Figure 3-2).

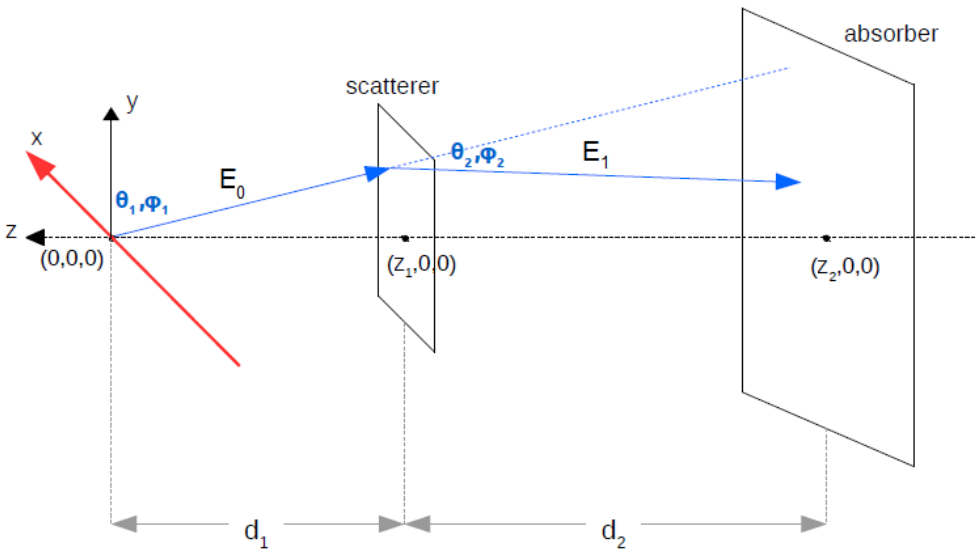


Figure 3-2: A simple detection setup used for geometry optimization. The crossing points of the photon track with the given planes are calculated analytically. The incident photon with energy of $E_0 = 4.44$ MeV interacts with the scatterer plane and the scattered photon with energy of E_1 is absorbed in the absorber plane.

Positions of interactions were calculated analytically using the intersection points of the respective detection planes and the track of the incident photon [59]. The

polar scattering angle of the photon that undergoes a Compton effect in the scatterer was randomly selected with weights according to the Klein-Nishina formula (see eq. (2.2)). Figure 3-3 shows the Klein-Nishina cross-section as a function of the Compton scattering angle θ for different energies of incident photons. The azimuthal angles were also randomly selected with a homogeneous distribution from the interval $[0,2\pi]$.

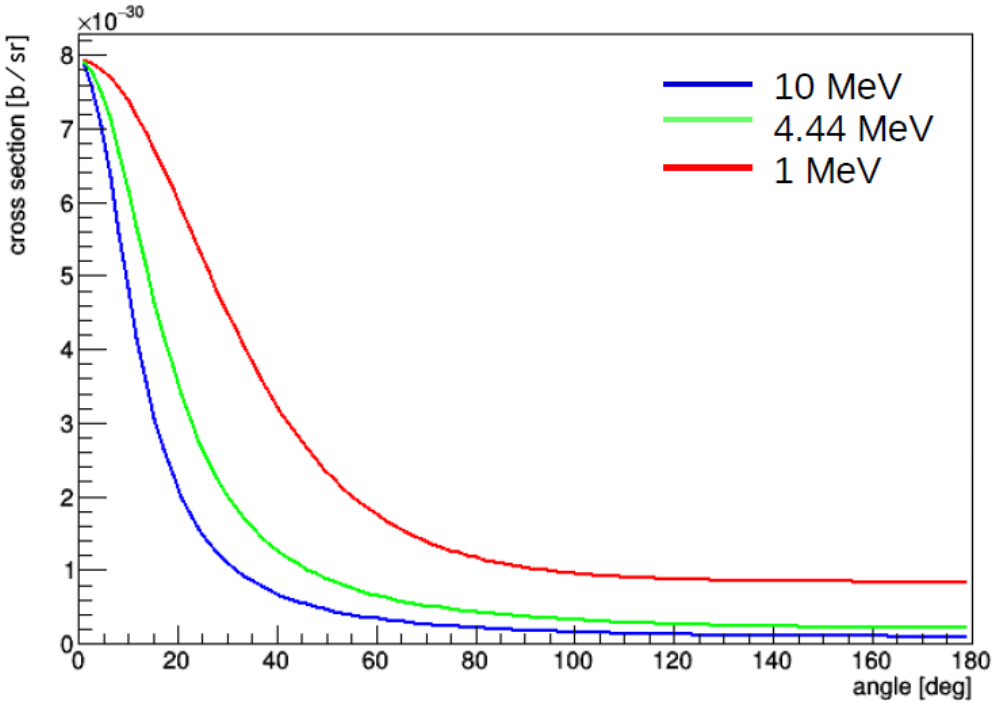


Figure 3-3: The plot of Klein-Nishina cross-section as a function of the scattering angle θ for photons with 1, 4.44 and 10 MeV energies [59].

Then, the energy E_1 of the scattered photon is calculated using eq. (2.1). Assuming that $E_0 = 4.44\,\text{MeV}$, the energy depositions in the scatterer E_{scat} and in the absorber E_{abs} can be calculated. Since the simulation only aims for the optimization of the PSF, other types of interactions, including multiple Compton scattering, were neglected. Therefore, the data obtained in this simulation provided an idealistic situation neglecting detection efficiency, physical phenomena contributing to the potential background, random coincidences, etc. However, it allowed for a preliminary optimization of the geometry of the Compton camera and development of image reconstruction algorithms.

3.2.2 Image Reconstruction

For simplicity, 2D image reconstruction method was used. Provided that the image plane is perpendicular to the z axis and located at $z = z_0$, the intersection of the cone and FOV is calculated by the following equation:

$$[n_x(x-x_s)+n_y(y-y_s)+n_z(z-z_s)]^2 = \cos^2\theta[(x-x_s)^2+(y-y_s)^2+(z-z_s)^2]. \quad (3.1)$$

where (x,y,z) are the Cartesian coordinates of points on cone surface, (n_x,n_y,n_z) is a unit vector of the Compton cone axis, (x_s,y_s,z_s) is the apex of the Compton cone in the scatterer and θ is its half opening angle.

For each event, once the Compton cone parameters are determined (apex, axis and aperture), the source position is reconstructed by intersecting the Compton cone with the image plane which is located at $z = 0$. In such a way, the eq. (3.1) will convert to the ellipse equation:

$$[n_x(x-x_s)+n_y(y-y_s)-n_z(z_s)]^2 = \cos^2\theta[(x-x_s)^2+(y-y_s)^2+z_s^2]. \quad (3.2)$$

The first step in this method is to compute the intersections of the ellipses with the the edges of the segmented image plane using the eq. (3.2) depending on whether they are on a vertical border (x is known) or a horizontal one (y is known) as shown in Figure 2-6. Knowing that all pixels of the image plane were numbered beforehand, a bit value (here, 0.001 mm) is added to and subtracted from the intersection point to find out which two pixel numbers the point of intersection belongs to. Then, the pixels containing the intersection points with the same number are searched for and the length of track in the pixel of interest is calculated. As the pixel's size is small (generally, 1 mm), the arc in each pixel is approximated by straight section. It is repeated for all pixels intersected to make an estimate of the reconstructed conic section (see Figure 2-6). This process continues for all detected events. In such a way, the back-projections of all events are weighted by segment length in each intersected pixel and consequently these weights are summed up for each pixel showing the source position probability.

Subsequently, the LM-MLEM was implemented for the back-projected data using eq. (2.7). When applying back-projection, the system matrix of LM-MLEM was being calculated and stored based on the same sample of events used for image reconstruction at the same time. Each element of the system matrix represents a transition probability as described in section 2.5.2. For image optimization, the system matrix was read and used for the iterations of LM-MLEM. To reduce the cost in CPU time and memory, the sensitivity map was assumed uniform in each pixel for the results presented in this thesis.

In order to achieve a realistic detector response, energy and position smearing were applied. The energy resolution as a function of deposited energy E was determined based on the Geant4 simulation of a $1 \times 1 \times 100\,\mathrm{mm}^3$ LuAG:Ce fiber [56] and parameterized with the following formula:

$$\frac{\sigma_E}{E} = P_0 + \frac{P_1}{E^{1/2}} + \frac{P_2}{E^{3/2}}. \tag{3.3}$$

The fitting results of the eq. (3.3) for the energy E expressed in MeV is shown in the Figure 3-4.

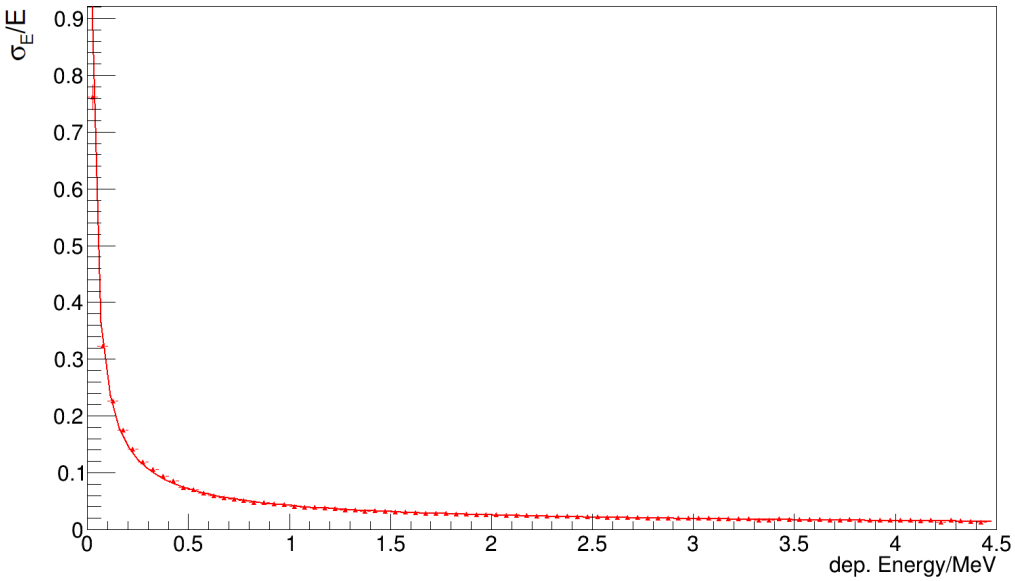


Figure 3-4: The energy resolution as a function of energy deposit for a 10 cm long LuAG(Ce) fiber with a square cross-section ($1 \times 1\,\mathrm{mm}^2$).

The following parameters of the equation were obtained: $P_0 = 0.0026(3)$, $P_1 = 0.0276(6)\,\mathrm{MeV}^{1/2}$, and $P_2 = 0.0176(3)\,\mathrm{MeV}^{3/2}$. The positions in the directions perpendicular to the fiber axis (here, x and z axes) were smeared with a uniform distribution with the width of the fiber size (1 mm), and along the fiber with a Gaussian distribution with $\sigma = 4\,\mathrm{mm}$. This value is almost the same as the expected average value of σ along the fiber for 4.4 MeV gammas [56].

The geometrical parameters of the detector including, the inter-detector distance (IDD), the source-scatterer distance (SSD), areas of the scatterer and the absorber as well as the position of the source in the FOV were investigated. Simulations with different values of those parameters were performed in order to find the optimal geometrical detection setup. Subsequently, the reconstructed images were evaluated and the resulting σ values of the PSF were compared. The results obtained in this detector geometry study are the starting point for a more detailed simulation data and allow for further optimization (see section 3.3).

3.3 Simulation Data

Based on the optimization of the detector surface areas and distances discussed in previous section, the SiFi-CC detector response was simulated [11, 30] in Geant4 version 10.4.p02. Figure 3-1 shows the proposed, simulated detection setup. This detector is composed of the scatterer and the absorber consisting of $1 \times 1 \times 100\,\mathrm{mm}^3$ fibers made of the LYSO scintillator. The stacked fibers are arranged into layers in which every second layer is shifted by half a fiber. For the readout, the SiPMs are coupled to the both ends of each fiber. The scatterer has 10 layers of fibers with each layer consisting of 76 fibers. Its dimensions are 12.7 mm thickness (z direction), 100 mm height (y direction), and 98.8 mm width (x direction). The absorber has 30 layers of fibers. The dimensions of the absorber module are 38.7 mm thickness, 100 mm height, and 98.8 mm width. An extensive description of the fiber units and their configuration can be found in [11, 56].

The module surfaces are parallel to the xy-plane (see Figure 3-1) and are centered on the expected position of the Bragg peak which is localized at space coordinates

(0, 0, 0). The SSD and IDD are set to the achieved optimum value of 200 mm (see section 4.1).

The predefined physics list QGSP_BIC_HP_EMZ was used [60]. A single beam spot was simulated with a count rate of 3×10^8 protons per beam spot and a delivery time of 10 ms [11]. The PMMA target was chosen which is recommended by IAEA [61] as a water substitute and is commonly used in radiation dosimetry. A 180 MeV proton beam interacting with a PMMA target was a source of gamma particles emitted in nuclear reactions. A Gaussian distribution with $\sigma_E = 0.2$ MeV was considered for the beam energy. The energy spectrum of PG along the proton beam axis is shown in Figure 3-5.

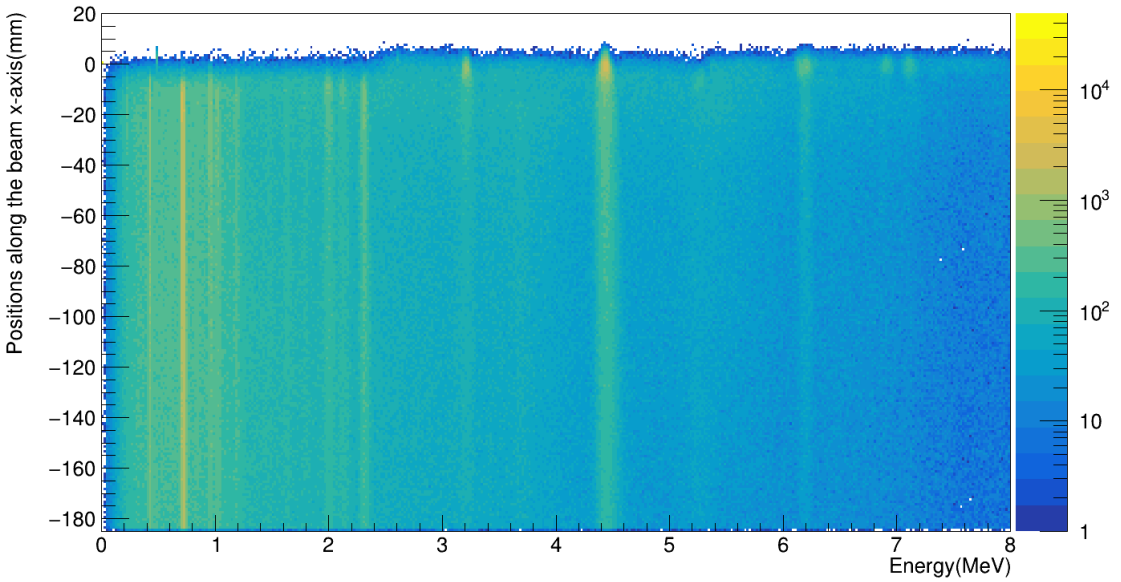


Figure 3-5: Energy spectrum of prompt gamma rays along the beam axis produced during the irradiation of the PMMA phantom by a 180 MeV proton beam.

The beam spot size at the target entrance perpendicular to the beam axis (y, z directions) was selected 2.5 mm standard deviation, typical values for a clinical beam [62]. More details about the SiFi-CC performance and event selection process were described in [11].

The Geant4 simulation provides information on real gamma interaction positions and energy deposits as well as the corresponding SiFi-CC response. All simulation information was stored in a ROOT [63] file and contains the following information:

- Emission position of the gamma photon and its energy and direction.
- Location of the Compton scattering and the energies of the recoil electron (RE) in the first interaction and the scattered photon (SP) after the first interaction.
- All subsequent interactions for the RE and the SP after first Compton scattering in terms of type and location.
- List of reconstructed clusters from the deposited energies within the SiFi-CC modules. The reconstructed information of each cluster contains the number of excited fibers (multiplicity), cluster's location and deposited energy, and the corresponding uncertainties. The explanation of how the uncertainties were calculated can be found in [56].

The information of the RE and the SP after first Compton scattering is used as a golden standard to preprocess the reconstructed clusters; identifying Compton event (signal) from other gamma interactions (background) and preparing the input data for the training phase. More detail can be found in the next section.

3.4 Machine Learning

The software framework used for classification of registered events originated by various processes of PG interaction in the detector is based on the ROOT CERN [63] toolkit for multivariate data analysis, TMVA version 4.3.0 [47]. It is not only a collection of multivariate methods, but also it is a common interface to different methods for classification and regression problems. All multivariate techniques in TMVA belong to the family of supervised learning algorithms.

3.4.1 Training Data

A machine learning model is trained over a data set to make predictions about those data that it has not seen before. Therefore, the training data set is needed in the machine learning model to recognize certain types of patterns. In supervised learning, we have two variable definitions including, features and targets in terms

of the input data sample. The features are a set of discriminating variables (in our case hits positions and deposited energies from detector response) which a machine learning model use to make the predictions. The targets are the outputs predicted from the model (such as Compton events known as positive targets or non-Compton known as negative targets). Using these features and targets, a machine learning model would map an input data to a desired output value (i.e. the positive targets).

The training data usually go through a preprocessing phase to make them suitable for training machine learning models. In the following, it is explained how the data preprocessing is done and how the features and targets will be selected. The source of data is the Geant4 simulation discussed in section 3.3, and the results after preprocessing phase are training data that can be directly fed into different machine learning models.

Data Preprocessing

Data preprocessing refers to the technique of cleaning and organizing the data for training machine learning models. The first step is to build a learning dataset of the simulated events filtered with interactions that yield at least 1 cluster hit in each of the two modules of the SiFi-CC. The reason is that to reconstruct a Compton event, a minimum of 2 cluster hits is required. One cluster would be for the RE in the scatterer, which usually corresponds to the Compton event's position, and the second would be for the SP interacting in the absorber. In this study, different event classes identified by the number of their cluster hits, exist. The events with at least 2 cluster hits only in one of the modules are not suitable for a Compton cone reconstruction but could be used in other event processing [64, 65] which is behind this study.

The first half of all statistics simulated by Geant4 goes for signal/background classification before passing to the training phase. The whole signal/background classification procedure for Geant4 simulated data is shown in Figure 3-6. Moreover, the full description of the stages of this approach is listed as follows. The stages were numbered with respect to the presented classification flowchart.

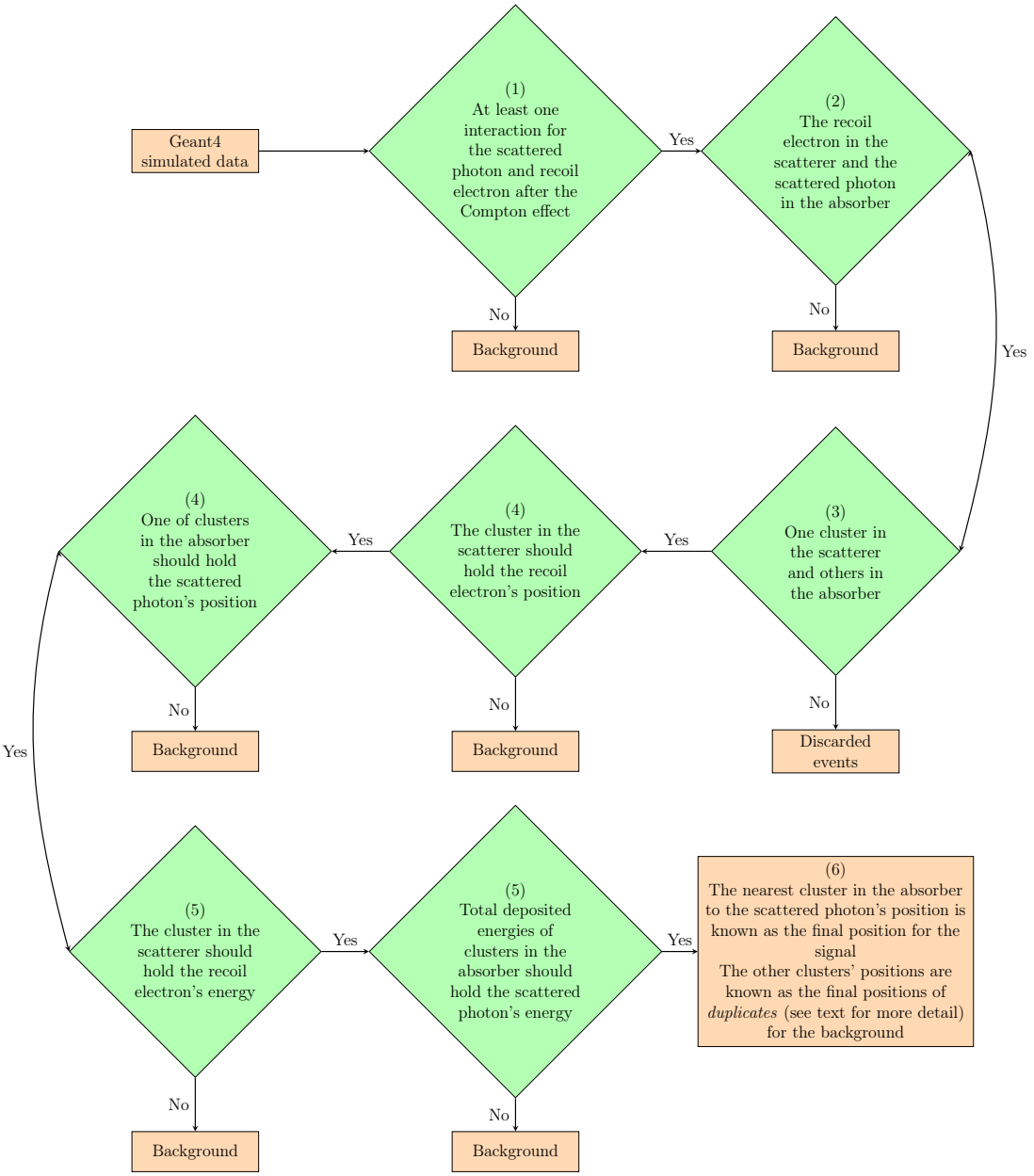


Figure 3-6: A flowchart of the signal/background classification of simulated data by Geant4 before passing to the training phase.

1. It is checked if each primary photon interacts via Compton effect, in this way, both the RE in the scatterer and the SP in the absorber should go through at least one interaction after the Compton scattering. Otherwise, that event belongs to background event and it deposits its energy via different types of interactions (non-Compton events).

   Note: To increase the background statistics, we introduce events called *fake events*. In this way, for each detected non-Compton event which has at least

1 cluster hit in each module, all possible cluster hit pairs are considered as separate events. For example, a non-Compton event class with 4 cluster hits whose 1 cluster is in the scatterer and 3 clusters are in the absorber, results in three separate events. In fact, one of these events should exist and the other two are the *fake events*. The only difference among these three separate events is the final position in the absorber.

2. For each Compton event, it is checked if the RE is in the scatterer and the SP is in the absorber. Otherwise, that event belongs to different possibilities of interactions e.g. Compton back-scattering (background category).
   Note: Here, we also included *fake events* as described above for each Compton back-scattering event.

3. The event classes up to 5 cluster hits whose 1 cluster hit is in the scatterer and others are in the absorber, are passed for further investigation.
   Note: The main reason of this choice refers to low contribution of Compton events with more than 1 cluster hit in the scatterer (only about 8 %, see Figure 3-7).

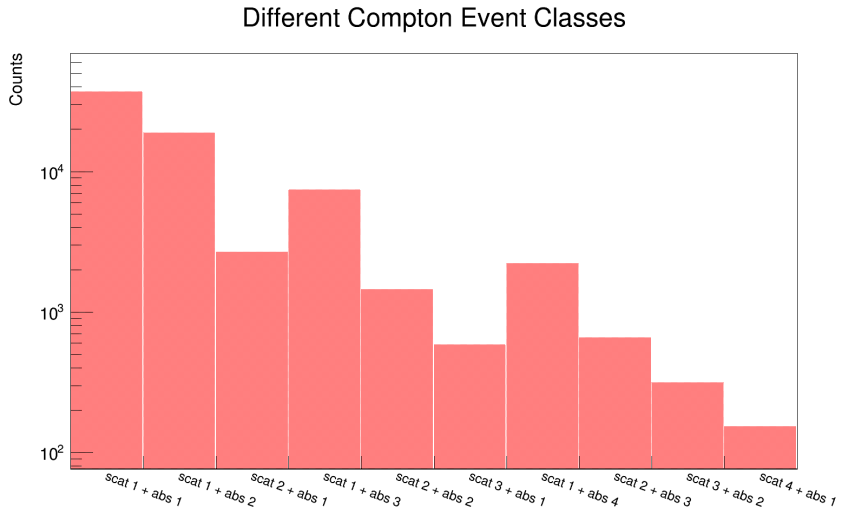


Figure 3-7: The Compton event classes for the first half of all statistics. Each Compton event class consists of different number of cluster hit combinations. For example, the label (scat 3 + abs 2) refers to the event whose 3 cluster hits are in the scatterer and 2 cluster hits are in the absorber. As it is shown, the contribution of the Compton events with more than 1 cluster in the scatterer is very less for all event classes. Therefore, they are discarded from further investigation.

4. It is checked if the absolute difference between the cluster's position and the position of the RE in the scatterer is within the given uncertainties (2.6 mm along x and z axes and 10 mm along y axis [11], for more detail see section 4.2.1). Moreover, it is also checked for clusters' and the SP's in the absorber and at least one of them should hold the position of the SP (the position uncertainties are the same used for cluster's position in the absorber). If one of the mentioned conditions is not met, the event is called as bad Compton events and belongs to the background category.
   Note: Here, we also included *fake events* for each bad Compton event.

5. The absolute difference between the cluster's deposited energy and the RE's energy in the scatterer should be within the given uncertainty (12% of the RE's energy [11], for more detail see section 4.2.1). It is also checked if the absolute difference between the total deposited energies in the clusters and the SP's energy in the absorber is less than the 12% of the SP's energy (see section 4.2.1). Again, if one of the mentioned conditions is not met, the event is called as bad Compton events and belongs to the background category.
   Note: Here, the *fake events* were included for each bad Compton event.

6. Finally, the nearest cluster's position to the position of the SP is selected as the final position for the Compton event.
   Note: In order to increase the statistics, we introduce events called *duplicates* (opposite pairs), which are made of the cluster's position in the scatterer and those clusters' positions farther away from the SP's position in the absorber compared to the one's for the Compton event. Such an event is classified as a background event.
   Note: The total deposited energy in the absorber is the same for Compton events and *duplicates*.

The records of all event types matching the above criteria are 258484 events which form the basis for the training data. It should be also mentioned that those *fake events* and *duplicates* will be removed at the end of analysis. Later, the removal process is explained in more details (see section 4.3.4).

Features

Although the simulated data contains much information (see section 3.3), the features are the variables of the reconstructed clusters only. They are the inputs for each machine learning model which are processed. Each recorded event gives us information about its cluster hit positions and the corresponding deposited energies.

Although it seems that using all reconstructed cluster information would be useful as features in Compton event identification, finding suitable features is a real challenge to achieve the best performance of a model in the training phase. Hence, we introduced a new parameter called the cosine of internal scattering angle term (shown in Figure 3-8) as another feature.

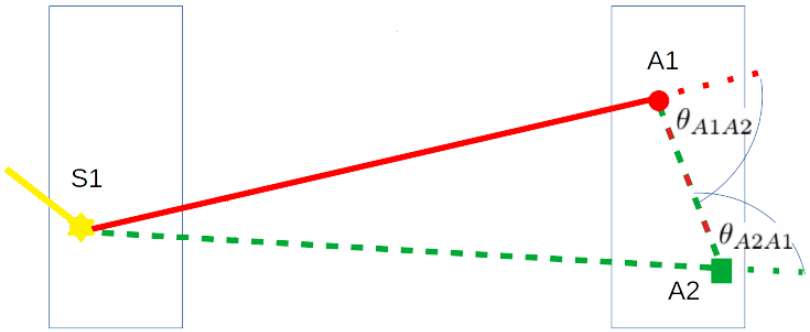


Figure 3-8: The internal scattering angles are shown for a reconstructed event with 3 cluster hits as a feature for training this event class. After first interaction of PG in the scatterer (S1), the scattered photon interacts in two positions in the absorber at points A1 and A2 for which the interaction sequence is unknown in the real situation. The internal scattering angle would be useful in Compton event identification. See text for more details.

We used Compton scattering properties defined by kinematics and Klein-Nishina cross section (see section 2.2) for identification of the scattering sequence. As the scattering angle is larger, the scattered photon has smaller energy and then the Compton cross section for smaller energy becomes larger and vice versa (see Figure 3-3).

Figure 3-9 also shows the integral probability of Compton scattering as a function of scattering angle. It is illustrated that a photon whose energy is smaller has higher Compton scattering probability in comparison with a higher energy photon.

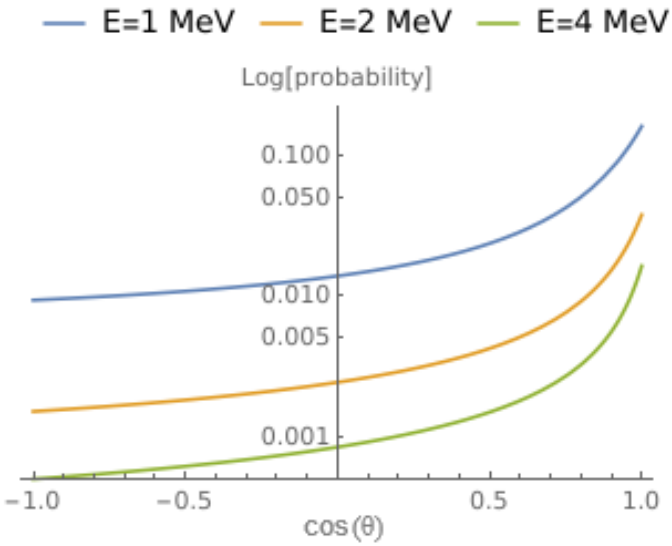


Figure 3-9: The illustration of the integral probability of Compton interaction for three different photon's energies of 1, 2 and 4 MeV as a function of scattering angle.

Therefore, this physical principle can also be very useful to discriminate Compton events from non-Compton events. This angular term combines all positions into one parameter, makes the model less complex, and helps select Compton events in case of more than 1 cluster hit in the absorber.

On the one hand, the more variables are used, more information about the event can be obtained and the model should gain greater separating power theoretically, but on the other hand, practically we want to reduce the number of variables used to train the model while retaining its performance as much as possible.

The main reason refers to the fact that we use Monte Carlo simulated data to train different classification models and Monte Carlo simulations are not always perfectly reliable for all variables. Moreover, the less variables are used, the less human effort is needed for the variables' validity checks. Finally, it leads to reduction in systematic errors and training time.

Therefore, the correlation among these variables is firstly studied before they are recognized as features in the training phase. The more knowledge about the capabilities of variables in Compton event identification, the less complex model.

Targets

The targets are Compton events that can be correctly reconstructed from the SiFi-CC. As it was shown in Figure 3-6, if an event meets the conditions successfully, it will be labeled as a Compton event (positive target). The rest of the events, including the Compton events that do not match these criteria (bad Compton events) and events coming from other interactions of PG with the detector (non-Compton events) are marked as negative targets. The objective of each presented model is to correctly identify the positive target events reflecting the ideal Compton events. The target of a single event includes the RE and SP locations and energies, the corresponding clusters' positions and deposited energies, and the index of event type whether signal or background events. All information is saved in two separate categories of signal/background for each event class and prepared for the training in TMVA.

Data Splitting and Overtraining

The training data is generally split into two subsequent data sets, namely, the *train* and *test* data sets. The simple train/test split is a technique for evaluating and monitoring the performance of a machine learning algorithm. During the training phase, the data set is used to train the model and fit it to the available data with known inputs and outputs. Then, the test data set is used to evaluate the model's performance on unseen data samples and how well the model will work in practice. In this study, we split the data set into two halves for training and testing. This choice is one of the common train/test split percentages [66, 67], especially when having enough number of events. Therefore, the first half of all statistics simulated by Geant4 was used as the training sample and the second half was considered as the test data sample.

While training a model, the overtraining may occur in which the trained model starts to show the statistical fluctuations in the data set leading to generality loss in the model. To avoid overtraining, tuning of the hyper-parameters available in each model was performed as one of the most effective solutions.

In this study, a useful method called *k-fold* cross-validation [68] was used to train the data set. It can help us monitor the model's performance, especially when having a high risk of overtraining. It generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split [69]. In standard *k-fold* cross-validation, the data set is partitioned into k subsets, called folds. Then, the algorithm is trained on $k-1$ folds while the remaining fold is used as the validation set. It is repeated until each fold is used as the validation set exactly once. Using the validation set is important, especially when trying to find the best hyper-parameter values for each model. More details about the hyper-parameters tuning of each model and *k-fold* cross-validation can be found in section 4.2.4.

3.4.2 Analysis Phase

After selecting the best classifier among all present classifiers based on the performance evaluation, its weights containing the training results (i.e. classifier's response to signal/background events) are stored for each event class.

In the analysis phase, the weights are loaded along with the second half of the data sample as an unknown input (test data set). The event loop is then run and for each event, the classifier value is computed according to the weights from the training phase.

Later, to separate Compton events from background events, one can either apply cuts on the Receiver Operating Characteristic (ROC) curve for each event class or apply fitting methods [47] to the classifier output, finding the best optimal cuts for each event class. It is explained in more detail in section 4.3.2.

Chapter 4

Results

This chapter firstly focuses on optimization of the detector geometry for the proposed setup presented on Figure 3-1. A detailed illustration of the hyper-parameters tuning and training different machine learning models is presented. Then, the performance of the best selected model in Compton event identification are evaluated. Finally, the LM-MLEM reconstructed images of PG distal edge position distributions from the selected model predictions are presented. The results presented in the fist part have been published in [11]. Some part of the results was presented on IEEE Nuclear Science Symposium and Medical Imaging Conference [70].

4.1 SiFi-CC Design Optimization

In this section, the results of the geometry study optimization of the simulated simple detection setup (see Figure 3-2) is presented. This study allows for a preliminary optimization of the proposed SiFi-CC detection setup. The PSF results were obtained for 10^5 reconstructed events emitted from a 4.44 MeV point-like gamma source. A more detailed description of the simple detection setup is presented in section 3.2.1.

Then, the LM-MLEM image reconstruction was performed. All of the presented results (standard deviation values of σ_x) refers to the direction along the proton beam. The LM-MLEM convergence criterion was the relative error of σ_x values for each two successive iterations. The LM-MLEM iteration continued till this relative

error was less than 1%. In most cases, the LM-MLEM was terminated after 20 iterations, since there was no significant improvement.

4.1.1 Influence of inter-detector distance and source-scatterer distance on the detector response

The resulting σ_x values of the PSF as a function of the inter-detector distance (IDD) and the source-scatterer distance (SSD) are presented in Figure 4-1. It can be seen that σ_x value improves with increasing IDD and deteriorates when SSD increases. Based on the presented results and taking into account the minimal possible distance from the patient, the optimal value of the SSD was set to 200 mm. In case of the IDD, there is no significant improvement in the spatial resolution and the image quality after 200 mm. Moreover, it should be mentioned that the detection efficiency would drop [13] and cause degradation of the image quality with larger values of SSD and IDD in realistic conditions.

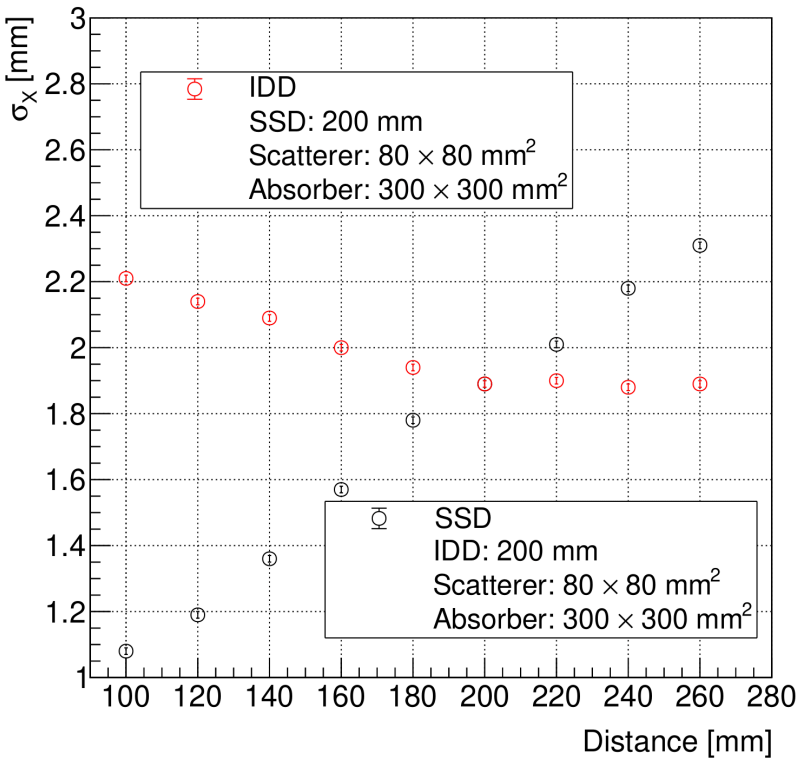


Figure 4-1: The σ_x values of the PSF along the proton beam axis for different IDD and different SSD (figure from [11]).

4.1.2 Influence of the scatterer and absorber size on the detector response

Figure 4-2 shows the influence of the size of the scatterer and the absorber (assumed both have a square shape) on σ_x values of the PSF for one of the simulated configurations. It can be observed that while increasing the size of the respective detector modules, the spatial resolution deteriorates. This dependency is, however, much weaker than in the case of distance variations in the detection setup. This is due to the fact that, for a larger size of the absorber, more low-energetic photons which scattered at large angles impinge on the detector edge. Such photons introduce larger uncertainties in the image reconstruction process compared with high-energetic photons scattered at smaller angles.

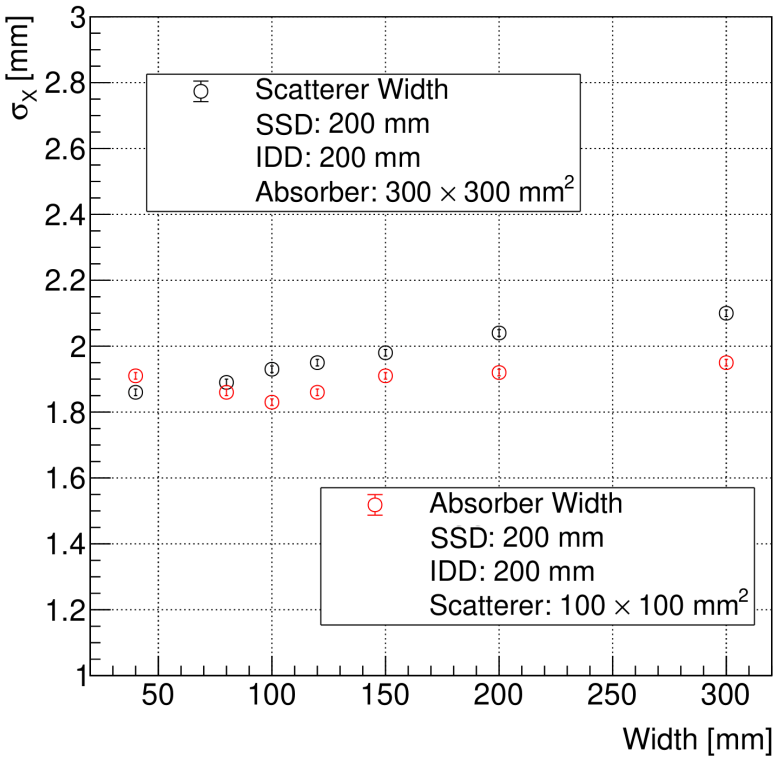


Figure 4-2: The σ_x of the PSF for different widths of scatterer and absorber. The details of the simulated setup for the study of respective parameters are listed in the figure. The optimal width values of the scatterer and the absorber are 100 mm (figure from [11]).

4.1.3 Influence of the lateral position of the source in the FOV on the detector response

A series of simulations with the gamma source placed at different lateral positions with respect to the detector axis allowed to determine its FOV. Figure 4-3 shows that σ_x values gradually increases when the source is moved away from the centre of the scatterer along the x axis and it dramatically deteriorates for farther distances.

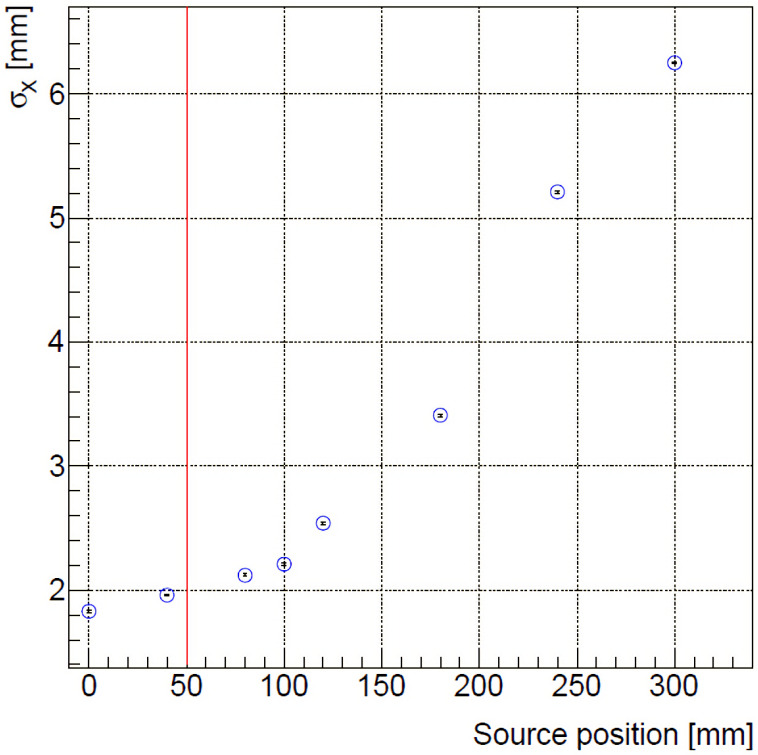


Figure 4-3: Influence of the lateral source position in the field of view of the camera on σ_x of the PSF. The red line indicates the edge of the scatterer (figure from [11]).

Additionally, Figure 4-4 shows the fraction of events reconstructed for different source positions. As it can be seen, the fraction of reconstructed events decreases when the source is moved away from the centre of the detector. The image quality degradation refers to the fact that only photons emitted at certain angles from a non-central source are registered. The farther away from the detector centre, the source is placed, the less fraction of photons are registered. This is due to the smaller solid angle coverage of the detector with respect to the source, leading to

missing some information about the source distribution. Moreover, the registration of interactions in both detector modules requires scattering at large angles, which has a lower probability.

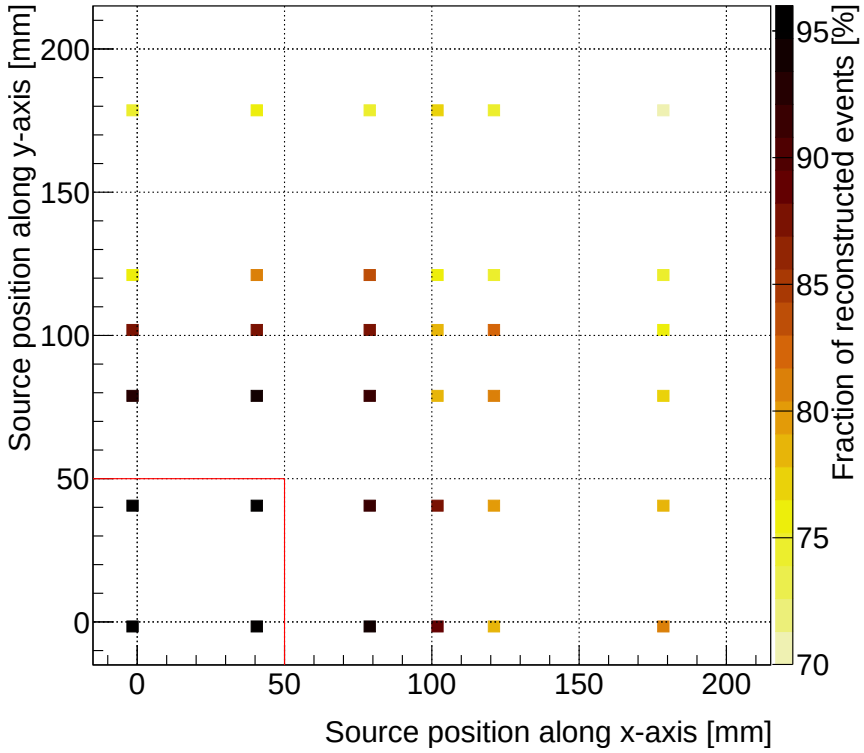


Figure 4-4: Two-dimensional profile of the fraction of events reconstructed for given source positions in the geometrical simulations. One quarter of the field of view was investigated because of symmetrical acceptance with respect to the detector center. Red lines indicate boundaries of the scatterer. After 100 mm source displacement from the detector center, the fraction decreases drastically (figure from [11]).

4.1.4 Design Guidelines

From the presented results, several design guidelines emerge for the proposed SiFi-CC detector.

- First, it appears that SSD is a sensitive parameter. Its choice can have a significant effect on the spatial resolution of the detector. It should be as small as possible in order to maximise the detector efficiency and to minimise its spatial resolution (for a source located at the centre of the detector FOV).

The choice of SSD is also limited by the patient's comfort and by the tumour depth. Therefore, the optimal value of the SSD was determined to be 200 mm.

- Second, we know that the choice of IDD is a trade-off between a good spatial resolution and a high efficiency. Therefore, the optimal value of the IDD was determined to be 200 mm. This finding is also in agreement with other research [71–73].

- Third, as it is found, σ_x values of the PSF behaviour as function of the scatterer and absorber widths is less significant compared to the IDD and SSD influences on resolution of the detector. However, it is important that the area of both modules should be large enough to detect a sufficient number of the Compton events. Therefore, it was determined that the optimal width for the scatterer and the absorber is 100 mm. As discussed further in section 4.1.3, these detector dimensions would also provide a sufficient FOV.

- Finally, as expected, when the source is moved farther away from the detector center, the spatial resolution of the detector would deteriorate. The fraction of reconstructed events also decreases when the distance between the source and center of the detector exceeds 100 mm. This is due to lower statistics resulting from a smaller geometrical acceptance. It shows that the obtained FOV is relatively large compared to the size of the proposed detector and certainly sufficient for PG imaging applications.

The final setup of the proposed SiFi-CC in this study (see section 3.3) was chosen not only following these guidelines, but also the technical aspects of the design, such as its compactness, a reasonable distance between the detector and patient, and the practicability of the LYSO fiber production.

4.2 SiFi-CC Machine Learning

One of the main goal of this thesis is to develop a machine learning framework taking the event data registered in the SiFi-CC as an input, then identify the Compton events. Three different models available in TMVA including BDT, MLP, and k-NN

compete and their performances are compared using ROC curves. Later, the best one will be selected for further investigation.

4.2.1 Target Variables

Identified Compton events used as targets in the training models should meet the following criteria:

- The distances between the x and z positions of the recoil electron (RE) and the scattered photon (SP) after first Compton scattering and their corresponding reconstructed cluster hits in the scatterer and the absorber are less than the uncertainty of 2.6 mm corresponding to the width of two fibers.
- The distance between the y position of the RE and the SP after first Compton scattering and their corresponding reconstructed cluster hits in the scatterer and the absorber is less than the uncertainty of 10 mm corresponding to the resolution achieved with the cluster position reconstruction [11].
- The difference between the RE's and the SP's energies after first Compton scattering and the reconstructed clusters' energies are less than 12% of the RE's and SP's energies. It comes from the resolution achieved with the cluster energy reconstruction [11].

Note that in case of more than one reconstructed cluster hit in the absorber, the one matching the above position criteria and the nearest to the SP is selected as deposition position in the absorber.

4.2.2 Variables Correlations

A model can be trained more precisely only if the suitable variables are chosen as features. The cluster hit positions and deposited energies can potentially contribute as features in the training phase. However, training with such variables does not always yield a good performance [74]. Therefore, before going through the training phase, the correlation among available variables should be understood. Linear correlation coefficient is a simple measure of shared information content among variables.

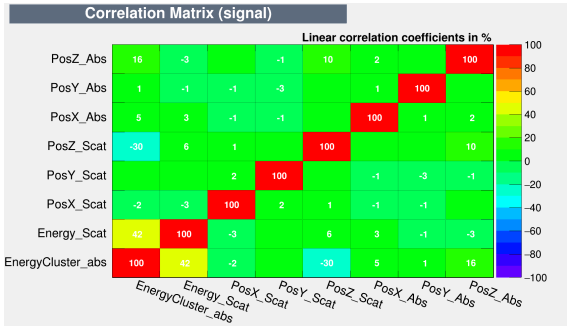
The TMVA framework readily provides a matrix of linear correlation coefficients for the variables used for each event class. As discussed in section 3.4.1, each event class whose exactly 1 cluster hit is in the scatterer and the other cluster hits are in the absorber were studied in further analysis. The linear correlation coefficient matrix among feature variables of such event classes is shown in Figure 4-5.

As can be seen, there is no strong relationship between the cluster hit positions and deposited energies. Therefore, we added the derived variable based on physical information (Compton effect), *angular distribution* term along with these 8 variables in the training to obtain better performance, especially in case of event classes with more than 2 cluster hits. We first perform the training on all available features and compare the three machine learning models' performances using ROC curves.

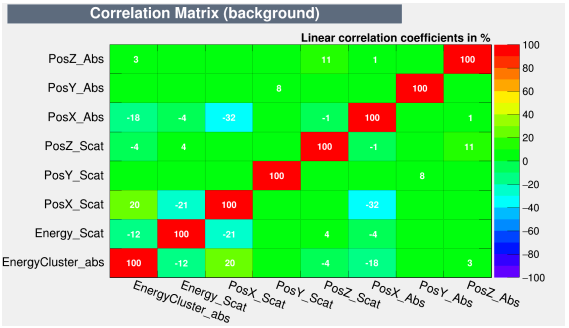
As discussed in section 3.4.1, we are also interested in using only those feature variables which can be useful in the training rather than randomly distributed cluster hit positions. Therefore, after selecting the best model, we train it with another feature list including only the deposited energy of the selected cluster hit in the scatterer and the absorber, and the Compton effect *angular distribution* term for each existing event. In such a way, we evaluate the trained model's performance using two different feature lists to select a better machine learning approach for further analysis in the image reconstruction.

4.2.3 Machine Learning Models

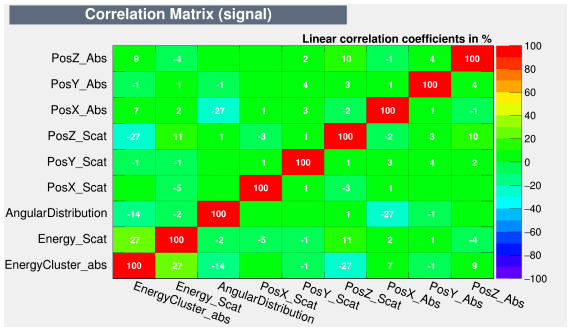
Based on the properties of different machine learning models presented in [47], we selected three models which fit more properly to our case of study. A coarse assessment of each model's capability is shown in Table 4.1. As can be seen, all these models have good performance in dealing with problems which exhibits any type of correlations among the input feature variables. Moreover, the models have slightly better robustness in favor of avoidance of overtraining and their robustness can be enhanced by adjusting the available hyperparameters (i.e., parameters whose values are used to control the learning process [75, 76]) of each model.



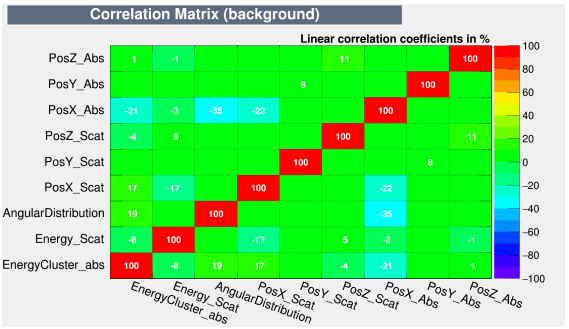
2 cluster hits (signal)



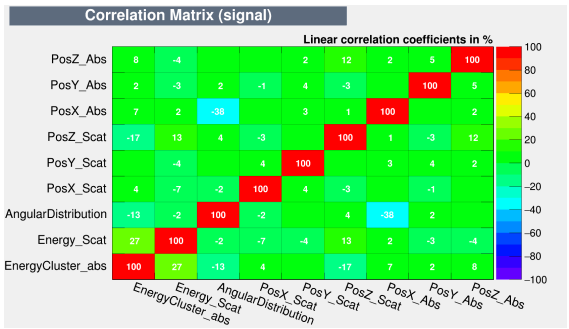
2 cluster hits (background)



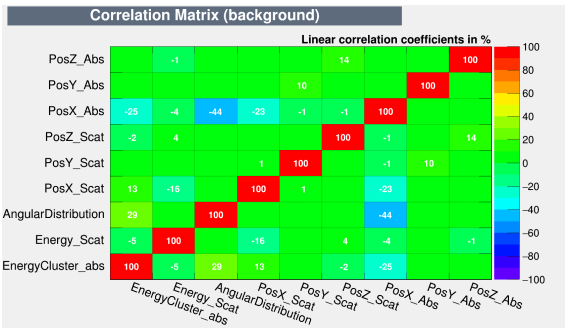
3 cluster hits (signal)



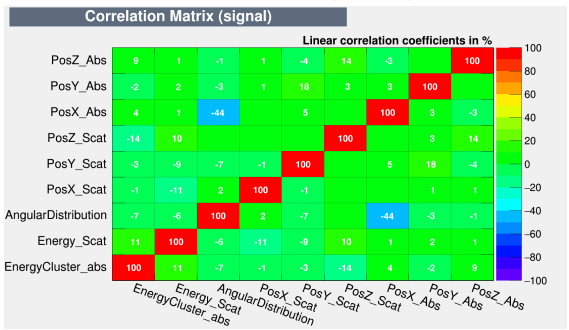
3 cluster hits (background)



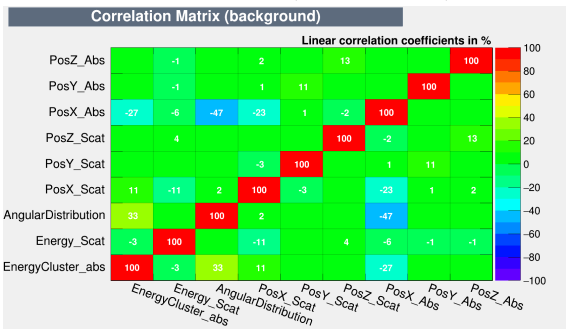
4 cluster hits (signal)



4 cluster hits (background)



5 cluster hits (signal)



5 cluster hits (background)

Figure 4-5: The correlation coefficient matrices of all available variables for each event class, generated by TMVA framework. Different event classes are presented in rows. The signal events are in the left column and the background events are in the right column.

Properties	Criteria	BDT	MLP	k-NN
Performance	No or linear correlations	++	++	+
	Non-linear correlations	++	++	++
Speed	Training	+	+	++
	Response	+	++	-
Robustness	Overtraining	+	+	+
	Weak variables	++	+	-
Transparency		-	-	+

Table 4.1: Assessment of three models' properties. The symbols stand for the attributes "good" (++), "fair" (+) and "bad" (-). More details could be found in [47].

4.2.4 Hyperparameters Tuning

Hyperparameters tuning is an integral part of the model training, especially for avoiding overtraining. Usually, it is time consuming to reach the proper trained model in classification problems. The recommendations available in TMVA tutorial and other research works [47, 55, 77, 78] provided a starting point for us to tune the hyperparameters of each model. The final tuned version of each model was obtained after several trial and errors.

To efficiently take a measure against overtraining, a *k-fold* cross-validation was applied to tune each model's hyperparameters. There is no formal rule to choose the k value. A poorly chosen value for k may result in a high variance changing a lot based on the data used to fit the model, or a high bias leading to an overestimate of the skill of the model. In this study, k was fixed to 10, based on the experimental results of the model skill estimate showing low bias and modest variance [69, 79, 80].

Moreover, the total area under the ROC curve (AUROC) was used as a well-known representation of the separating performance for different models trained on a particular data set. Note that the validation data set is named as the test data sample in TMVA.

BDT classifier

The most important hyperparameters for the BDT training in this study are presented in Table 4.2.

Hyperparameter	Value	Explanation
NVariables	9 (8*)	Number of variables used to train the BDT
NTrees	850	Number of decision trees in the forest
MinNodeSize	5%	Minimum percentage (lower bound) of training samples required in a leaf node
MaxDepth	3 (4^{+})	The maximum depth which a single decision tree can grow
BoostType	AdaBoost	Boosting type for assigning weights to each tree in the forest
nCuts	-1	Number of grid points across the variable range used to find an optimal cut for a new node

Table 4.2: Hyperparameters of the BDT model. The (*) sign points out the 8 feature variables used in case of 2 cluster hits event class. The ($^{+}$) sign points out that only in case of event class with 5 cluster hits, the MaxDepth is set to 4.

The first parameter investigated is NTrees. Increasing the number of trees is expected to yield a stabilization of the BDT output distribution with respect to statistical fluctuations and make the distribution more smooth. However, higher values of NTrees can not improve BDT performance unless lead to a time-consuming training. Table 4.3 shows the BDT performance saturation with higher number of trees for different event classes. Therefore, a shallow BDT with 850 trees was chosen because of its well enough AUROC value for each event class.

No. of Cluster hits in Event Class	Int ROC (300)	Int ROC (850)	Int ROC (2000)
2	0.880	0.885	0.886
3	0.839	0.843	0.841
4	0.842	0.845	0.842
5	0.831	0.832	0.832

Table 4.3: The ROC curve integral with different number of trees for all event classes. The number of trees were indicated in the brackets. A performance saturation with increase in trees' number can be seen.

Among the hyperparameters, the main two model's parameters in preventing a single decision tree from overtraining are MaxDepth and MinNodeSize. The theoretical maximum depth which a decision tree can grow is one less than the number of training samples, but it leads to overtraining. Therefore, the tree's depth can be adjusted by the MaxDepth hyperparameter. Moreover, the decision tree nodes can be expanded until all leaves contain less than the minimum percentages of train-

ing samples which is determined by MinNodeSize. These two hyperparameters are strongly related to one another. When regularizing the BDT, it was found that larger values of MinNodeSize and lower values of MaxDepth (simpler tree) can reduce the overtraining possibility. As the MaxDepth is the coarsest parameter, followed by MinNodeSize, such that one would start by finding a reasonable range for this value first. For the given data set both the choices MaxDepth values of 3 and 4 greatly reduce overtraining without compromising performance. The MinNodeSize value of 5% performs best in dealing with remaining overtraining. In the BDT training, the node splitting criterion is always a cut on a single variable selected by the model (see section 2.6.1). The nCuts optimizes the cut values by scanning over the selected variable range. In this study, the nCuts was set to -1. Therefore, the model automatically finds the best step size across the feature variable range to split a node into two new nodes.

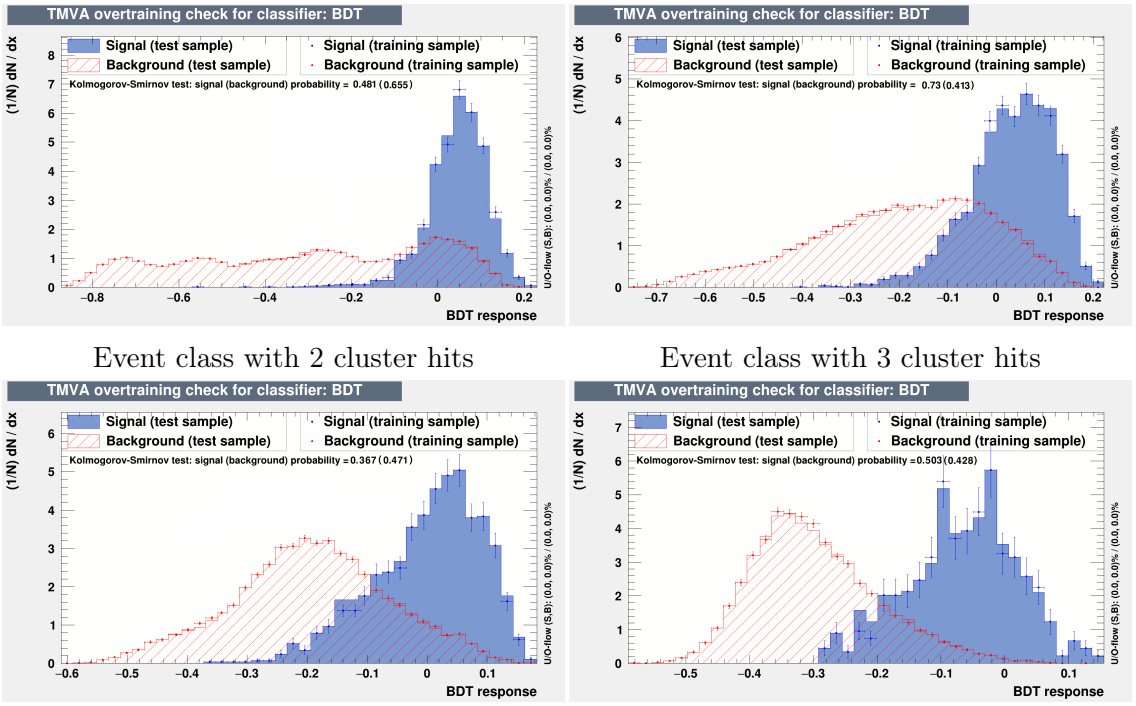
To check overtraining, the TMVA output provides a signal efficiency comparison for different specific background efficiencies from the training and test data samples (see Table 4.4). It turns out that there is a good agreement between the signal efficiency from the test and training samples. The maximum absolute difference was obtained in the case of event class with 5 cluster hits at 1% background efficiency. Also, it is observed that the more background rejection, the more compatible signal efficiencies from training and test samples. Therefore, the overtraining was avoided by tuning the hyperparameters.

Signal Efficiency: from test sample (from training sample)			
No. of Clusters in Event Class	@B = 0.01	@B = 0.1	@B = 0.3
2	0.087 (0.086)	0.538 (0.543)	0.951 (0.952)
3	0.086 (0.093)	0.475 (0.482)	0.828 (0.836)
4	0.104 (0.107)	0.527 (0.538)	0.835 (0.838)
5	0.168 (0.180)	0.589 (0.600)	0.812 (0.817)

Table 4.4: The comparison of signal efficiency obtained from test sample and training sample at different background efficiency @B.

Another useful test which clarifies how much training a model is far away from overtraining is *Kolmogorov-Smirnov test* [81, 82]. In TMVA, the *Kolmogorov-Smirnov*

test is applied between the training and the test output probability distribution for signal and background separately. In case the two distributions are compatible coming from the same parent distribution, a random value between 0 and 1 should be obtained. Otherwise it might be an indication of overtraining. The values very close to 0 and 1 are also not too good because they indicate that the statistical fluctuation is too small and again the distributions are not similar. Figure 4-6 shows the *Kolmogorov-Smirnov test* plots for the final configuration of BDT classifier for each event class.



Event class with 2 cluster hits | Event class with 3 cluster hits

Event class with 4 cluster hits | Event class with 5 cluster hits

Figure 4-6: The overtraining check using the Kolmogorov-Smirnov test for BDT model.

As it can be seen, the training and test data samples of all event classes are similar. Therefore, the model with the current configuration for each event class is kept safe from overtraining.

MLP classifier

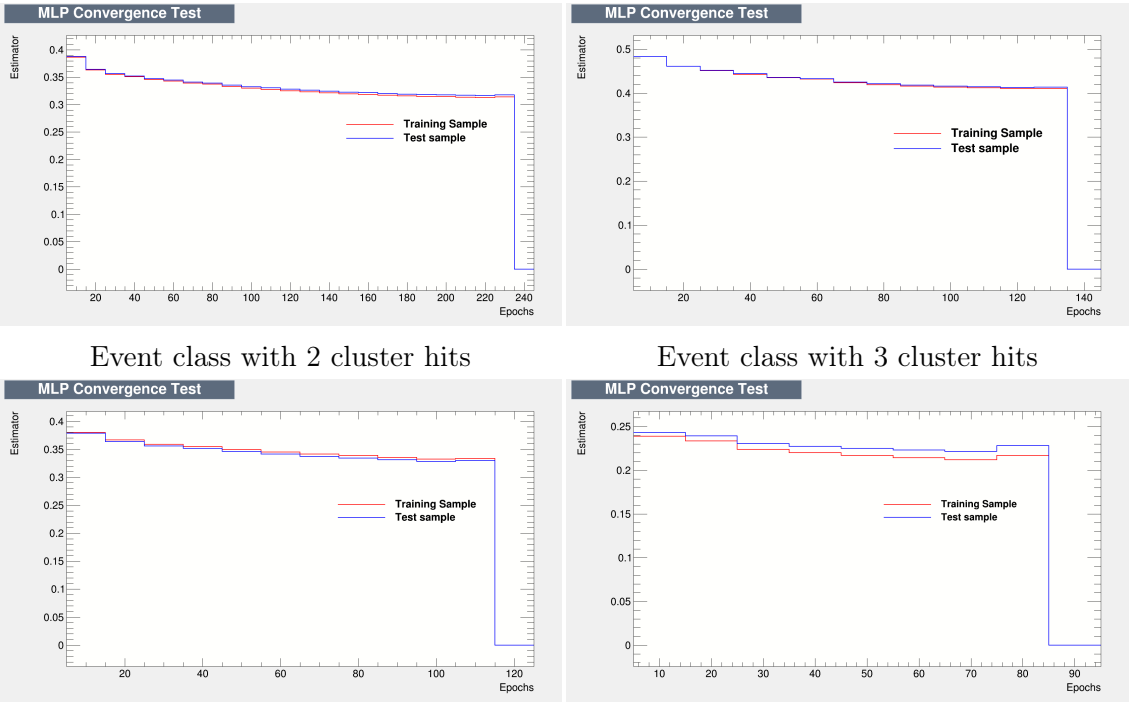
A short description of some important hyperparameters for the MLP model is shown in Table 4.5.

Hyperparameters	Description
NCycles	Number of training cycles (epochs)
HiddenLayers	Specification of hidden layers architecture
NeuronType	Neuron activation function type
EstimatorType	Loss function type
LearningRate	Learning rate parameter
TestRate	Test for overtraining performed at each i^{th} epoch
ConvergenceTests	Monitoring the number of steps required for convergence

Table 4.5: Hyperparameters of the MLP model.

In the first step of the hyperparameter tuning, one should start with a guess about the number of training epochs. Since it is not generally known beforehand, how many epochs are necessary to achieve a sufficiently good MLP training. In TMVA, it is possible to activate a convergence test by setting ConvergenceTests parameter to a value above 0. This value denotes the number of subsequent convergence tests which shows no improvement of the estimator (i.e. loss function) as an indicator of the completed training. The convergence tests and overtraining tests are performed simultaneously. The frequency of these tests can be set by the parameter TestRate. Figure 4-7 represents the estimator (loss function) versus the number of epochs for each event class on the final tuned hyperparameters. It is shown that more than the required number of epochs will not improve the configured neural network, leading to overtraining.

The choice of hidden layer architecture is one of the most important hyperparameters in the training. Although these layers do not directly interact with the external environment, they influence significantly the final output. In this stage, the number of layers and the number of neurons presented in each layer were determined. Several shallow and deeper parallel layers implementations were done for each event class. It was shown that a shallow neural network can be sufficient to achieve a reasonable performance of MLP model. However, the choice of number of neurons is more challenging. Too few neurons in the hidden layers will result in undertraining, so the signals are not detected adequately in a complicated data set.



Event class with 2 cluster hits | Event class with 3 cluster hits

Event class with 4 cluster hits | Event class with 5 cluster hits

Figure 4-7: The MLP convergence test for each event class.

On the other hand, too many neurons in the hidden layers will lead to over-training along with an increase in training the network. As a starting point, different number of neurons in each hidden layer were tested based on a few rules of thumb [47, 80]. First, the number of hidden neurons should be between the size of the input layer and the size of the output layer. Second, the number of neurons should be 2/3 the size of the input layer, plus the size of the output layer. Third, the number of neurons should be less than twice the size of the input layer.

The final neural network architecture for each event class was selected based on the quality of performance of each model after several trial and errors. Figure 4-8 shows the MLP architecture of the event class with 4 cluster hits. In this study, within each neuron, after calculating the linear transformation of the input data using the internal weights, a nonlinear activation function was applied to obtain the neuron's output. The NeuronType parameter was set to the desired activation function. Two activation functions including *sigmoid* and *Tanh* were used and tested in the hidden fully connected layers for different configurations. However, they both have achieved close performance to each other.

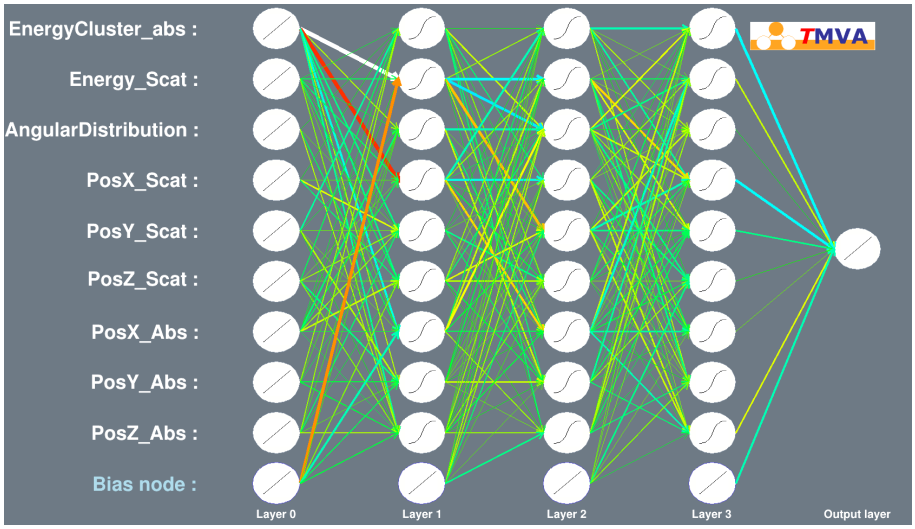


Figure 4-8: The MLP architecture for the event class with 4 cluster hits. The first layer is the input layer, the last one is the output layer, and the three middle layers are hidden layers. In case of this event class, the input layer consists of neurons that hold the number of features used in the training, the number of neurons in hidden layers is also equal to the number of features, and one neuron in the output layer that holds the estimator output variable, showing the signal or background.

Therefore, the *sigmoid* function was selected as an activation function of hidden layers for further study. Moreover, the activation function used for the output neuron is linear (see Figure 4-8). Another crucial hyperparameter is the learning rate for network training. It defines the step size used to update the weights of the network neurons (see eq. (2.15)). A large learning rate increases the network weights' fluctuations and, consequently, increases the loss score. A small learning rate makes very insignificant updates and prevents convergence. When training a neural network, a compromise between the convergence and obtaining a loss score as low as possible should be reached (see Table 4.6). The loss function of the event type MLP classifier, determining if the event is Compton or not, is computed as mean squared error (see eq. (2.14)). Moreover, the Bayesian extension of MLP can be used to avoid overtraining. While it leads to an increase in computation time. This extension is enabled with the parameter UseRegulator in TMVA. By adding a new term, the network loss function $L(\mathbf{w})$ will be:

$$L'(\mathbf{w}) = L(\mathbf{w}) + \alpha|\mathbf{w}|^2, \tag{4.1}$$

where the additional term is proportional to the squared norm of the weight ensemble **w** of the network and the parameter α controls the level of model complexity and TMVA automatically selects the best value of α. This extension penalizes large weights to mitigate overtraining. Therefore, during the training phase, the network learns to identify the Compton events by optimizing the loss function. The final configuration of the tuned hyperparameters for training the MLP model is illustrated in Table 4.6.

No. of Cluster hits in Event Class	Hyperparameters for each MLP model				
	No. of Epochs	Hidden Layers	Learning Rate	Test Rate	Convergence
2	155	N, N, N+1	0.003	10	1
3	155	N, N, N	0.003		
4	115	N, N, N	0.005		
5	85	N, N+1	0.02		

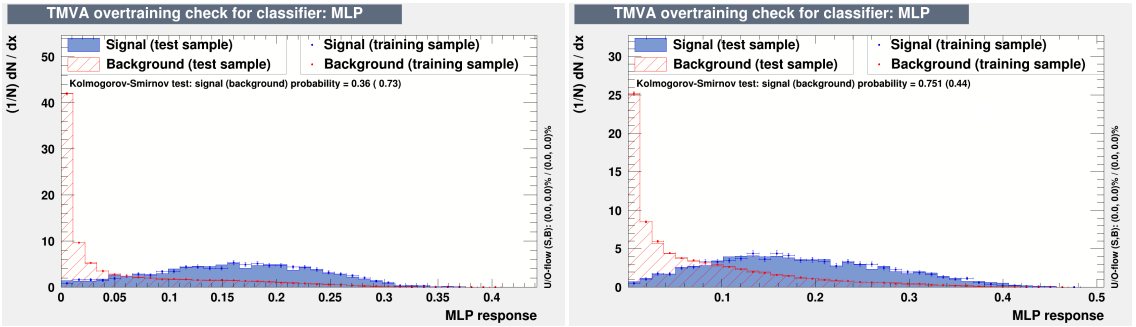
Table 4.6: The final configuration of the MLP model. The number N indacates the number of input variables (features) as neurons in each hidden layers. Also, the repetition of N shows the number of hidden layers used for each event class.

The overtraining test also showed that the models were significantly kept away from harmful overtraining effects (see Table 4.7).

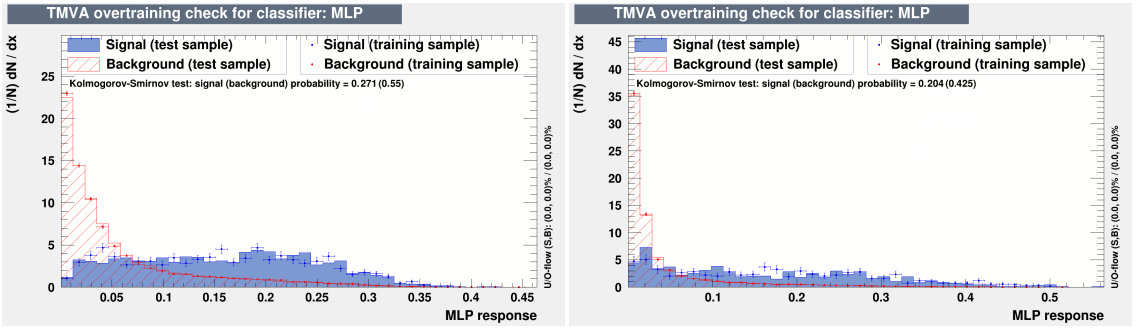
Signal Efficiency: from test sample (from training sample)			
No. of Clusters in Event Class	@B = 0.01	@B = 0.1	@B = 0.3
2	0.065 (0.066)	0.469 (0.475)	0.915 (0.913)
3	0.054 (0.051)	0.423 (0.426)	0.825 (0.831)
4	0.081 (0.083)	0.509 (0.495)	0.826 (0.824)
5	0.116 (0.111)	0.549 (0.563)	0.828 (0.833)

Table 4.7: The comparison of signal efficiency obtained from test sample and training sample in training the MLP model.

As it can be seen, the maximum absolute difference between the signal efficiency from test and training samples happens for event classes with 4 and 5 cluster hits at 90% background rejection. Moreover, *Kolmogorov-Smirnov test* plots are another evidence of avoiding overtraining for the final MLP configuration. Figure 4-9 indicates the similarity between training and test data samples for signal/background events in the training phase.



Event class with 2 cluster hits Event class with 3 cluster hits



Event class with 4 cluster hits Event class with 5 cluster hits

Figure 4-9: The overtraining check using the Kolmogorov-Smirnov test for the MLP model.

k-NN classifier

The final configuration of the trained k-NN model for each event class is illustrated in Table 4.8. The number of k-nearest neighbours around a query event plays the most important role in training the k-NN model. As a general rule, the smaller values of k neighbors are used, the more the model is subject to undertraining, leading to statistical fluctuations in the probability density function. Conversely, as the value of k increases, the predictions become more and more stable and precise (up to a certain k value). For large k values, the model is unable to generalize well on observations it has not yet seen. Based on the performance quality (AUROC) of each model, nkNN parameter (i.e., the number of k-nearest neighbours) was set after several trial and errors.

As discussed in section 2.6.3, since the input variables (features) have different units, an inverse weight of each feature was applied on the Euclidean metric to obtain rescaled *Euclidean distance* (see eq. (2.18)).

No. of Clusters in Event Class	Hyperparameters		
	nkNN	ScaleFrac	UseKernel
2	80	0.4	True
3	100	0.6	
4	70		
5	60		

Table 4.8: The final configuration of the k-NN model for each event class. See the text for more details.

The term ScaleFrac indicates the fraction of events used to compute the feature distribution width. For each event class, different event fractions were tested. The results show that higher values than the selected fraction for each event class can not improve the model's performance. Moreover, TMVA provides the UseKernel term (a polynomial kernel) for training the k-NN model to mitigate statistical fluctuations in the training data set. Adding this weight function (eq. (4.2)) can reduce the high variance of the k-NN response.

$$W(x) = \begin{cases} (1 - |x|^3)^3 & \text{if } |x| < 1, \\ 0 & \text{otherwise,} \end{cases} \tag{4.2}$$

Therefore, the weighted signal (background) events and then the weighted probability for the test event of signal (background) can be obtained through

$$W_{S(B)} = \sum_{i=1}^{k_{S(B)}} W\left(\frac{R_i}{R_k}\right), \tag{4.3}$$

$$P_{S(B)} = \frac{W_{S(B)}}{W_S + W_B}, \tag{4.4}$$

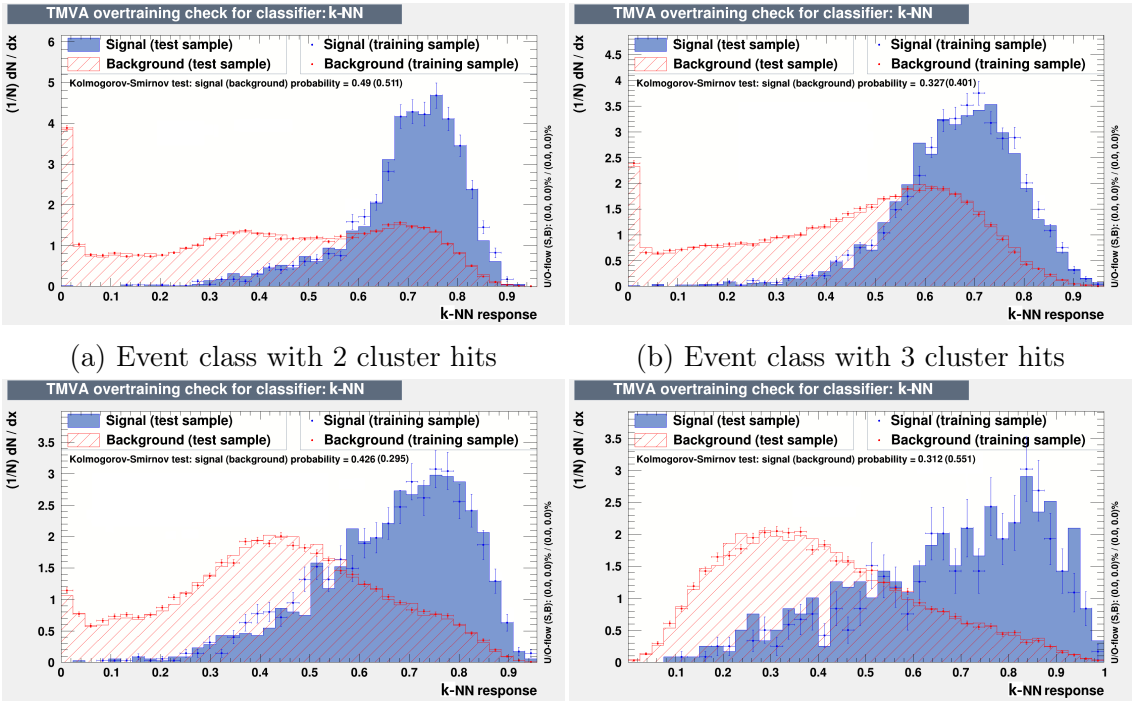
where $k_{S(B)}$ is the number of signal (background) events in the neighbourhood. R_i is the distance between the test event and the i_{th} neighbour and R_k is the rescaled *Euclidean distance* obtained from eq. (2.18) for all k-nearest neighbours. The eq. (2.17) will be then converted to eq. (4.4). To control model overtraining, we can check the TMVA output in which the signal efficiency of the training and test data samples at different background efficiencies are compared (see Table 4.9). It is shown that there is a good agreement between the signal efficiency from the test and training samples.

The maximum absolute difference of 0.009 was obtained in the case of event class with 5 cluster hits at 1% background efficiency. Therefore, the overtraining effects were greatly mitigated by tuning the hyperparameters.

Signal Efficiency: from test sample (from training sample)			
No. of Clusters in Event Class	@B = 0.01	@B = 0.1	@B = 0.3
2	0.058 (0.060)	0.455 (0.454)	0.876 (0.875)
3	0.065 (0.069)	0.409 (0.416)	0.766 (0.773)
4	0.076 (0.071)	0.489 (0.481)	0.811 (0.811)
5	0.127 (0.118)	0.515 (0.521)	0.783 (0.785)

Table 4.9: The comparison of signal efficiency obtained from test sample and training sample at different background efficiency @B.

Figure 4-10 shows *Kolmogorov-Smirnov test* plots indicating a good similarity among signal (background) events from training and test samples for all event classes, and demonstrating overtraining prevention.



(a) Event class with 2 cluster hits

(b) Event class with 3 cluster hits

(c) Event class with 4 cluster hits

(d) Event class with 5 cluster hits

Figure 4-10: The overtraining check using the Kolmogorov-Smirnov test for k-NN model.

4.2.5 Classifiers' Performances Evaluation

As mentioned earlier in section 4.2.4, AUROC curve was used as each model's performance evaluation when comparing their separation power on the training data sample. In fact, ROC curve shows how much background is rejected at each possible point of signal efficiency. Rather than making a single cut on each model's ROC curve, we used the full spectrum of the models' output score in this study. Therefore, the AUROC calculation provides a better representation of the separating performance of each trained model. This is the primary reason why we decided to use AUROC as a useful benchmark for evaluating the models' performance in this investigation, but other performance benchmarks are also possible [83–85]. Figure 4-11 shows the ROC curves of the presented models for each event class.

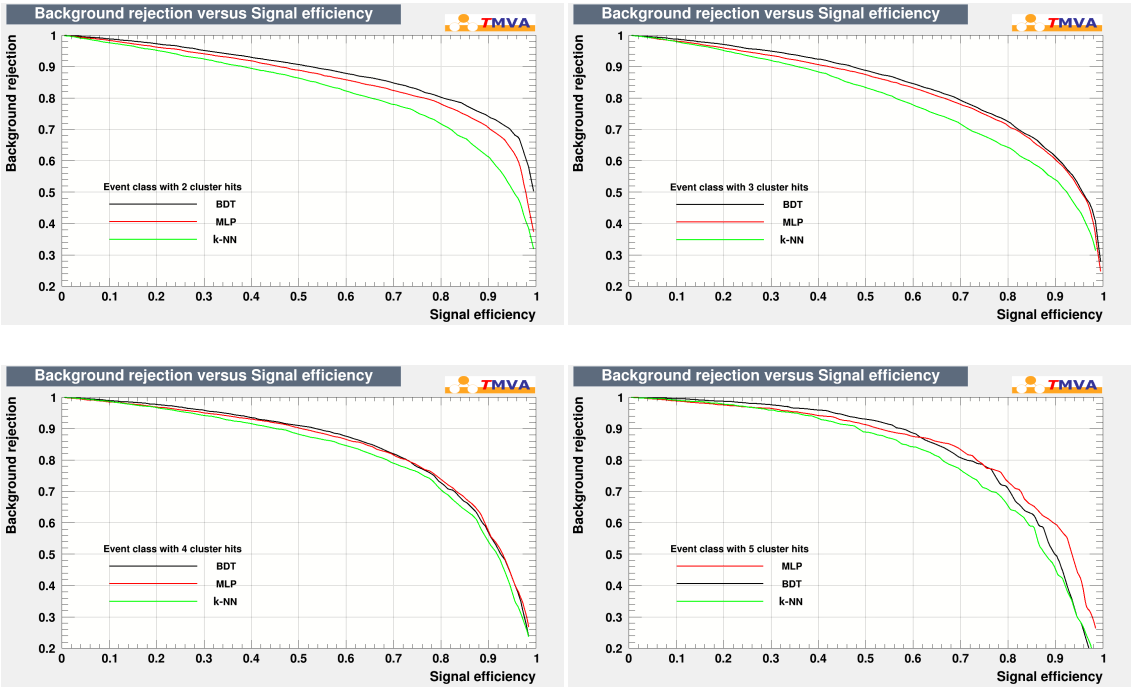


Figure 4-11: The comparison of ROC curves among all trained models for each event class. The higher (more convex) the curve, the better the model performs. Top row indicates event class with 2 cluster hits (left) and event class with 3 cluster hits (right). Bottom row represents event class with 4 cluster hits (left) and event class with 5 cluster hits (right).

As shown in the plot, TMVA ranks the models in order of their performance. In most cases, it seems that BDT model outperforms the other two, although both

BDT and MLP models represent almost the same separation power of the training. The AUROC values shown in Table 4.10 allows for the best model selection.

No. of Clusters in Event Class	The ROC Curve Integral		
	BDT	MLP	k-NN
2	0.885	0.863	0.851
3	0.843	0.832	0.811
4	0.845	0.837	0.826
5	0.832	0.842	0.815

Table 4.10: The AUROC values for different trained classifiers.

As can be seen, the large values of AUROC confirm that the training was completed successfully with the chosen hyperparameters for each model during the training phase. However, the BDT model achieves slightly higher separating performance on the data set except in the case of the event class with 5 cluster hits. Moreover, in the case of the event class with 2 cluster hits in which we are not able to benefit from the *angular distribution* term, the BDT classifier has shown stronger separating power (around 2% higher than MLP model). Therefore, it was selected as the best candidate in the following study.

4.2.6 BDT Classifiers Training

The influence of features' selection on the best classifier's performance was studied. We trained the model with only a few number of features mentioned in section 4.2.2, then compared its performance with the model's performance when trained with all available features. The ROC curves of the trained BDT models with two different number of feature variables (see section 3.4.1) on the first half of data sample are shown in Figure 4-12.

Intuitively, it can be expected that using the complete number of features in BDT training leads to higher separating power compared to when training the model with only a few features. However, it is also shown that the difference between AUROC values obtained from training BDT model with two different numbers of features decreases for event classes with higher number of cluster hits.

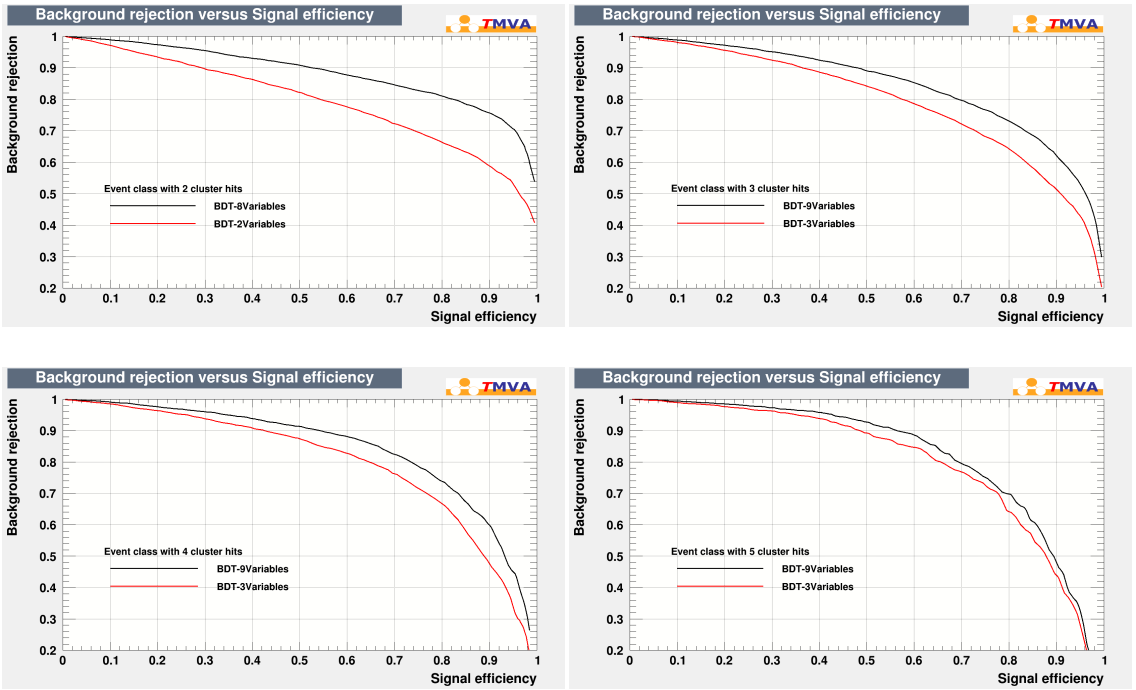


Figure 4-12: The comparison of ROC curves of the trained BDT models using different numbers of features for each event class. Top row indicates event class with 2 cluster hits (left) and event class with 3 cluster hits (right). Bottom row represents event class with 4 cluster hits (left) and event class with 5 cluster hits (right).

Table 4.11 provides a closer insight into the BDT models' power in signal/background classification.

The ROC Curve Integral		
No. of Clusters in Event Class	9 (8^+) Features	3 (2^+) Features
2	0.885	0.797
3	0.843	0.790
4	0.845	0.800
5	0.832	0.803

Table 4.11: The AUROC values of the BDT models trained using two different number of features. The ($^+$) sign points out the number of features in the case of event class with 2 cluster hits which are 8 and 2 respectively. 9 (8^+) features are all available variables. 3 (2^+) features are the deposited energy of the selected cluster hit in the scatterer and the absorber, and the *angular distribution* term only in the case of the event classes with higher than 2 cluster hits.

From Figure 4-12 and Table 4.11, it is deduced that BDT training with all features would be potentially a merit to reject most background events, especially in the case of the event classes with lower number of cluster hits. However, in the analysis phase, the performances of both trained BDT models using two different feature lists were assessed. As discussed in section 4.2.2, we want to know how well this model trained with only a few number of features, can discriminate quantitatively the Compton events from background events in the analysis phase and how it can affect the determination of the distal falloff of the Bragg peak in the image reconstruction stage.

4.3 Analysis Phase and Evaluation

This section evaluates the performance of the trained BDT model with two lists of features mentioned in Table 4.11 according to the evaluation metrics defined in the next section. Finally, the LM-MLEM reconstructed images from the model's predictions are presented and compared.

As mentioned earlier in section 3.4.1, the first half of all statistics simulated by Geant4 went into the training phase. So, the second half of statistics is used in the analysis phase as the test data set in which the BDT classifier's capability in signal/background classification on unseen data set will be assessed. The records available in the analysis phase contain 260663 events as the test data. The number of Compton and background events in the test data set before event selection by the trained BDT is 14297 and 246366, respectively. Moreover, the total number of Compton events in the Geant4 simulated data is 91787.

The significant difference between the number Compton events in the test data set and the simulated data refers to the fact that we identified Compton events up to 5 cluster hits for the analysis phase however, the contribution of the Compton events with higher number of cluster hits (up to 14 cluster hits) are available in the simulated data [11].

Also when preprocessing the data for the analysis phase, most number of Compton events which did not meet the position and energy uncertainties (see sec-

tion 3.4.1) went to the background category, however, they were recognized as the Compton events in the simulated data.

4.3.1 Evaluation Metrics

The metrics used to evaluate the predictions for the SiFi-CC are the recall, efficiency, and the purity. The analysis and improvements carried out throughout the course of this thesis focused on improving these metrics. Before going through the metrics' definitions, it should be mentioned that the Compton events predicted by the model consist of correctly classified events (Compton events) and not correctly classified events (background).

Recall

Recall measures the ratio between the number of correctly classified Compton events and the number of Compton events in the analysis phase before event selection by the model [67].

$$Recall = \frac{No.\ correctly\ classified\ Compton\ events}{No.\ Compton\ events\ before\ event\ selection} \tag{4.5}$$

Efficiency

Efficiency measures the ratio of the number of correctly classified Compton events to the total number of Compton events within the Geant4 simulated data.

$$Efficiency = \frac{No.\ correctly\ classified\ Compton\ events}{No.\ Compton\ events\ in\ simulated\ data} \tag{4.6}$$

Purity

Purity measures the ratio between the number of correctly classified Compton events and the total number of predicted Compton events by the model.

$$Purity = \frac{No.\ correctly\ classified\ Compton\ events}{No.\ predicted\ Compton\ events\ by\ model} \tag{4.7}$$

4.3.2 Cut Optimization

To discriminate Compton events from background events, we preferred to implement the Genetic Algorithm fitter [47] on the model output as the cut optimizer for this aim rather than applying a cut on a certain point of the ROC curve obtained from the training phase for each event class.

The reason refers to the preliminary results showing that it finds better optimal cuts to achieve higher recall, efficiency and purity. In the sense of signal/background discrimination, several configurations of this fitter were tested and the final parameters list is shown in Table 4.12.

Parameters	Value	Description
PopSize	100	The number of population at each generation of the Genetic Algorithm
Steps	30	The number of steps for convergence
Cycles	3	Independent cycles for evaluating the fitness

Table 4.12: The final configuration of the Genetic Algorithm.

4.3.3 Energy Regression

From the simulation, we know that the total deposited energy called *energy sum* in the SiFi-CC detector does not represent the primary energy of incident PG in most cases. This is due to the fact that photons might interact only a few times and leave the SiFi-CC detector. The lack of information about the total deposited energy may result in a worse reconstructed distal falloff position distribution. Therefore, the *energy sum* correction is challenging especially when reconstructing PG distal falloff positions more accurately.

Figure 4-13 shows this deficiency in case of one category of background events (bad Compton events i.e. those events which do not meet either position or energy uncertainties mentioned in section 4.2.1).

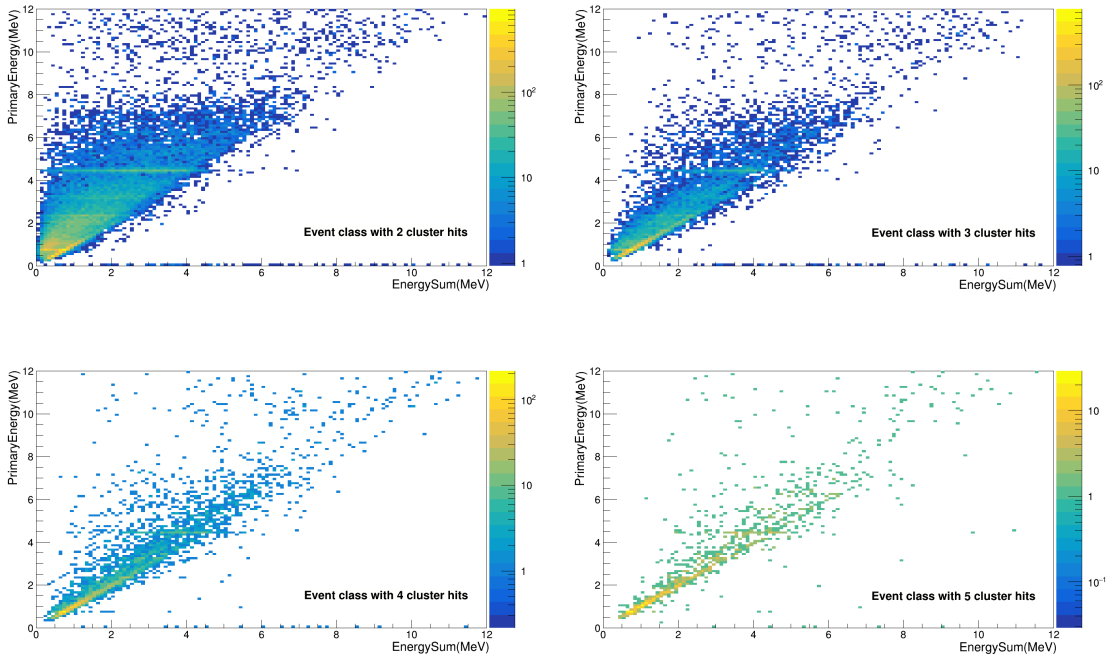


Figure 4-13: The relationship between the primary energy of PG and energy sum of background (bad Compton events) for each event class from the first half of data sample. Top row indicates event class with 2 cluster hits (left) and event class with 3 cluster hits (right). Bottom row represents event class with 4 cluster hits (left) and event class with 5 cluster hits (right).

As can be seen, the deposited energy of bad Compton events were not collected properly because such events escape from the SiFi-CC detector after a few interactions in most cases.

Although we can see that as the number of cluster hits increases, the total deposited energy has less deviation from the PG primary energy. Still, there is a broad energy deviation even in event class with 5 cluster hits leading to a worse reconstructed distal falloff position distribution at the end.

To solve this problem, we decided to perform an energy regression to recover the total deposited energy called *recovered energy sum* instead of *energy sum* for each event detected in SiFi-CC. In such a way, we introduced the appropriate weights for each energy bin in order to correct *energy sum* of each event.

We know that the total deposited energy of Compton events can greatly reflect the primary energy of PG. Figure 4-14 represents a linear relationship between the deposited energy of Compton events and their primary energy of PG for each event

class. Therefore, the primary energies of the Compton events were used as the target variable in the energy regression to produce weights as more precisely as possible.

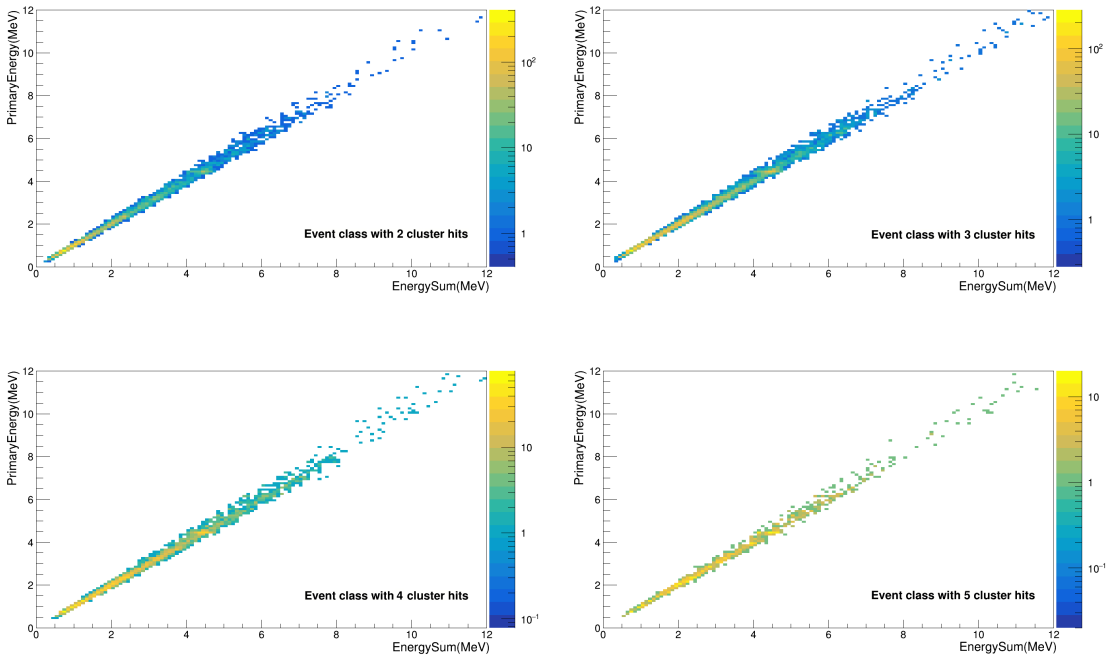


Figure 4-14: The relationship between the PG primary energy and energy sum of Compton events for each event class from the first half of data sample. Top row indicates event class with 2 cluster hits (left) and event class with 3 cluster hits (right). Bottom row represents event class with 4 cluster hits (left) and event class with 5 cluster hits (right).

We implemented a gradient boosted decision tree (BDTG) model only on the Compton events for each event class from the first half of data sample. When working with BDTG, it is also needed to tune some important parameters avoiding overtraining. Given small values of MaxDepth (3-4), BDTG is much less prone to overtraining compared to simple decision trees.

Moreover, its power can be increased by reducing the learning rate using the Shrinkage parameter. It is recommended that a small shrinkage (0.1-0.3) can improve the accuracy of the prediction but higher number of trees is required [47]. After several trial and errors, the final configuration of BDTG model is listed in Table 4.13.

No. of Clusters in Event Class	NTrees	MinNode Size (%)	MaxDepth	Shrinkage	nCuts
2	8000	0.1	4	0.1	40
3	10000	0.2			
4	8000	0.2			
5	8000	0.1			

Table 4.13: Final parameter configuration of the BDTG model.

Applying the produced weights from the Compton events to the *energy sum* of bad Compton events, the linearity between the PG primary energy and *recovered energy sum* is more visible as shown in Figure 4-15.

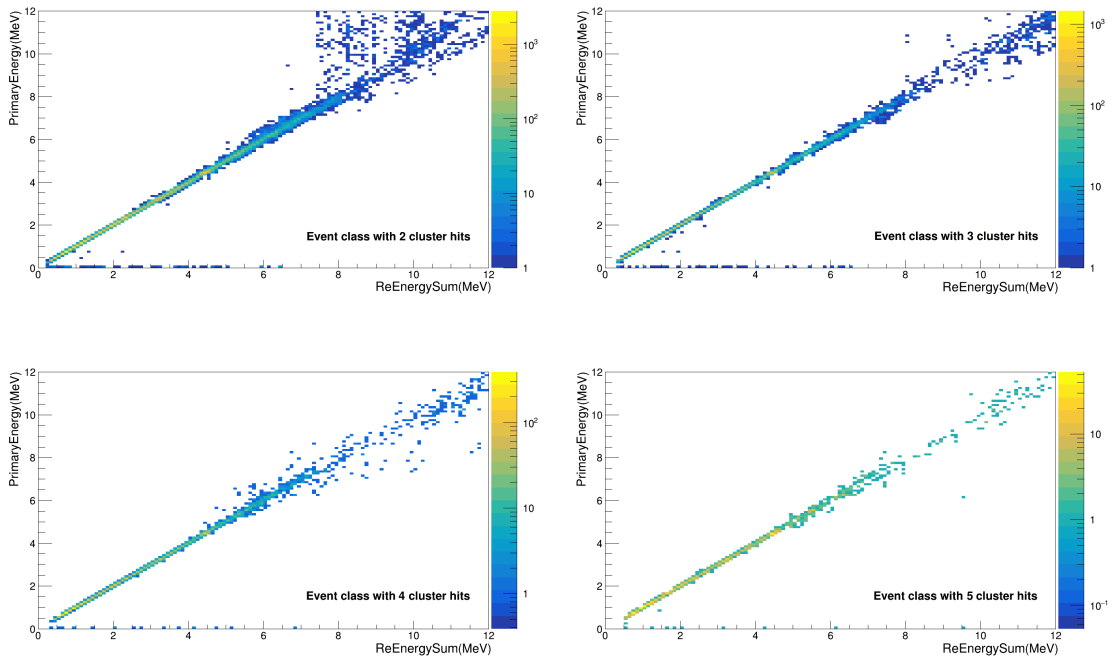


Figure 4-15: The relationship between the primary energy of PG and recovered energy sum of background (bad Compton events) for each event class from the first half of data sample. Top row indicates event class with 2 cluster hits (left) and event class with 3 cluster hits (right). Bottom row represents event class with 4 cluster hits (left) and event class with 5 cluster hits (right).

As can be seen, there is a linear relationship between the PG primary energy and the *recovered energy sum* until the total deposited energy of 7.5 MeV (the worst in the case of 2 cluster hits). As the higher number of cluster hits, the more accurate

the *recovered energy sum*. Moreover, it is shown that the model greatly predicts one of the most prominent energies (4.44 MeV) which is very useful in distal falloff position reconstruction [24].

Therefore, we decided to benefit from the energy regression in the analysis phase. In such a way, the *energy sum* of each Compton event predicted by the model were corrected using the produced weights from the energy regression and used as the *recovered energy sum* in the image reconstruction stage.

4.3.4 Fake Events and Duplicates Exclusion

As mentioned in section 3.4.1, all possible cluster hit pairs (containing also *fake events*) in the case of non-Compton, Compton back-scattering events and bad Compton events and *duplicates* in the case of Compton events were included in both the training and analysis phases. It was done to increase the statistics and assess the ability of models in distinguishing between Compton events and background events in more complicated situations. However, we know that only one of combination of cluster hits in the case of background events such as non-Compton events is representing the real background event, but the others such as *duplicates* in the case of Compton events are the *fake events*. Therefore, we should remove such events when presenting the analysis results.

In this study, we dealt with only those event classes whose 1 cluster hit was in the scatterer and others were in the absorber. After event selection by the model, each background event which has the same cluster hit position in the scatterer as the Compton event's was recognized as *duplicates* and removed from the analysis result. Moreover, as the BDT model provided the classification probability for each event, after event selection by the model in the analysis phase, the probability of background events with the same cluster hit position in the scatterer were compared. Then, the cluster hit pair which has higher probability was chosen as the real background event and others were recognized as *fake events* and removed from the analysis result.

4.3.5 Quantitative Results

The robustness of BDT models in signal/background separation on the unseen data set with two different feature lists are shown in Figure 4-16.

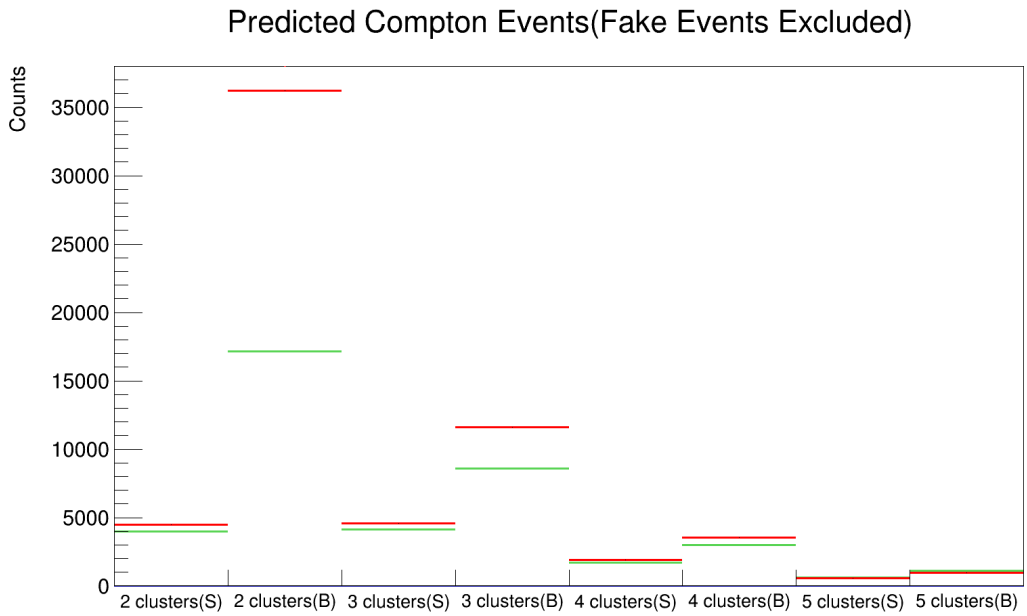


Figure 4-16: The event topology comparison of the predicted Compton events by two BDT models (the fake events were excluded). The red lines represent the number of Compton events when using only a few features (3 variables). The green lines show the number of Compton events in case of using all possible features (9 variables) in the analysis. The letters (S) and (B) refer to the correctly classified Compton events (signal) and the not correctly classified events (background) predicted by two models, respectively.

As expected, the number of background events decreases when including all possible features (9 variables) in the analysis. The most significant decrease happens in the case of event class with 2 cluster hits (by around 50%). In addition, it can be seen that a slight decrease in the number of correctly classified Compton events compared to the case when using only a few features (3 variables).

As the comparison between two BDT models' performance, the Table 4.14 shows the obtained recall, efficiency and purity along with the number of predicted Compton events after applying the optimal cuts in both study cases.

Evaluation Terms	3 (2^+) Features	9 (8^+) Features
Total Number of Predicted Compton Events	63827	40291
Number of correctly classified Compton Events	11520	10448
Background Events	52307	29843
Recall	81%	73%
Efficiency	12.6%	11.4%
Purity	18.0%	25.9%

Table 4.14: Evaluation results of the two BDT models. The ($^+$) sign refers to the number of features in the case of event class with 2 cluster hits which are 2 and 8 respectively.

The higher the recall, efficiency and purity, the better the model performs in signal/background separation. The excellent recall values of 81% and 73% were obtained in both cases of the training model with 3 features and 9 features, respectively.

In the case of using all possible features, the efficiency decreases by 1% while the purity increases by around 8% which is a magnificent improvement in signal/background discrimination. This higher value of purity in the case of training with all possible features indicates a relative increase of 44% in the ratio of correctly classified Compton events, which has an important effect in the reconstruction stage. Therefore, training the model using all possible features can lead to a better performance.

The difference between *recovered energy sum* of predicted Compton events by using two different feature lists and the primary energy of PGs from Geant4 simulation is illustrated in Figure 4-17. It is found that the energy difference in each model's predictions is uniform and centered and the energy regression model worked well even for background events such as non-Compton events. Therefore, the *recovered energy sum* of predicted Compton events were recovered greatly; reflecting the prominent PG energy peaks in both models' predictions. In addition, the contribution of events whose the *recovered energy sum* were not predicted properly (after 7.5 MeV, see section 4.3.3) is low in image reconstruction stage.

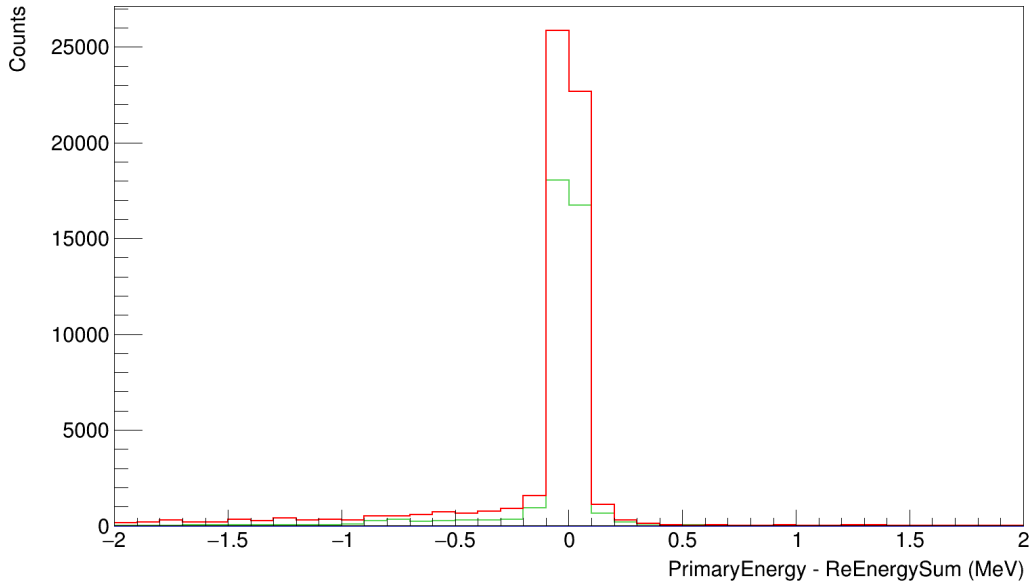


Figure 4-17: The energy difference between the recovered energy sum of predicted Compton events by two BDT models and the PGs from Geant4 simulation (the fake events were excluded). The green curve shows the energy difference between the PGs and predicted Compton events in case of using all possible features (9 variables) in the analysis. The red curve represents the energy difference between PGs and predicted Compton events when training with only a few features (3 variables). In both cases, the energy difference is greatly uniform, however in the case of training with 9 features, there is less tail in the energy difference and it is more centered. This refers to the fact that more number of background events were removed compared to training with only 3 features.

4.3.6 Image Reconstruction Assessment

The distal falloff positions of the models' predictions were reconstructed using the LM-MLEM algorithm, and then refined with Gaussian smoothing filter [86]. The image for the Compton events obtained from the Geant4 simulation were also reconstructed for comparison.

A convergence criterion called *pixel-wise* was applied to LM-MLEM algorithm. The *pixel-wise* was selected due to dealing with 2D profiles of the reconstructed images in the course of this thesis. However, it is also applicable to LM-MLEM reconstruction for 3D profiles of PG deposited dose positions. The steps of *pixel-wise* convergence criterion are as follows.

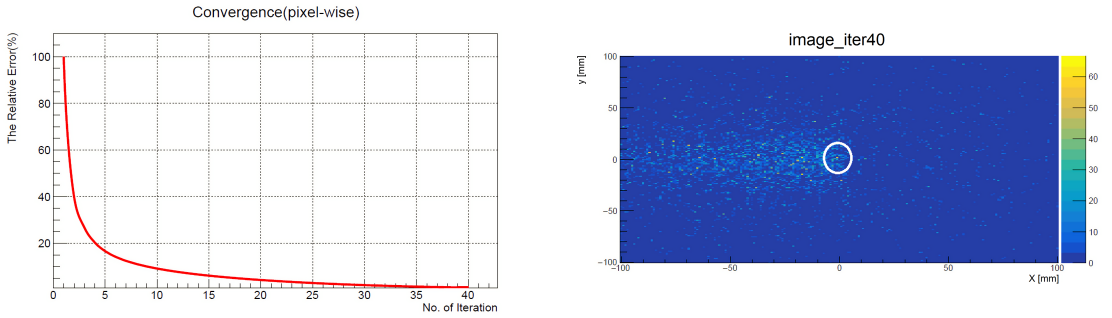


Figure 4-18: The *pixel-wise* convergence rule for the LM-MLEM algorithm. Right: 2D profile of the reconstructed emission position of PGs after applying the *pixel-wise* convergence rule. The white null circle displays the expected distal falloff location. Left: The relative error of pixels content after 40 iterations of LM-MLEM. It is assumed that the relative error is 100% for the first iteration.

- 20 pixels are chosen around the expected distal falloff location.
- Each pixel from iteration (n) is compared with the same pixel in iteration (n+1).
- The maximum difference between 20 pixels content is found.
- The relative error of pixel content between iteration (n+1) and (n) is computed as

$$Relative\ Error = \frac{|Maximum\ intensity\ difference\ between\ pixels|}{|The\ pixel's\ intensity\ in\ iteration\ (n+1)|}. \quad (4.8)$$

- The iteration stops when the relative error is less than 1%.

Figure 4-18 shows the 2D profiles (xy-plane) of the reconstructed emission position of PGs in the case of using a few features (3 variables) after applying the *pixel-wise* convergence criterion. The LM-MLEM stopped after 40 iterations and then the Gaussian smoothing filter with kernel of 3 mm was applied for further study.

To see how successful applying the energy regression to the *energy sum* of the predictions is, one would compare the 2D profiles of the predicted Compton events reconstruction when using *energy sum* and *recovered energy sum* of each predicted Compton event (see Figure 4-19).

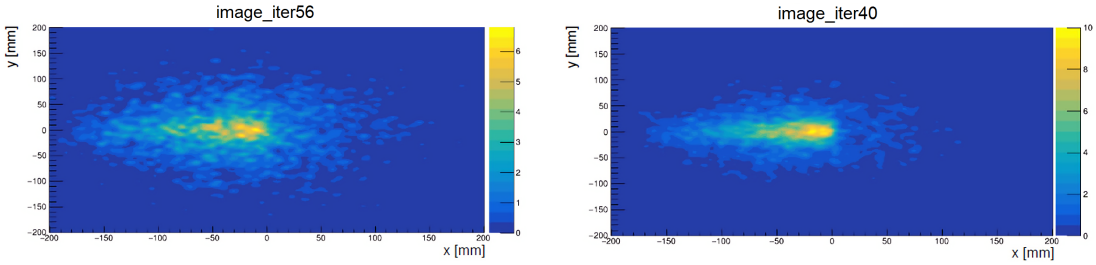


Figure 4-19: The comparison of predicted Compton events reconstruction profiles using *energy sum* (left) and *recovered energy sum* (right) of the predictions. The results of training the BDT model with a few features (3 variables) were used. The 2D profiles were obtained after sufficient iterations and then refined by the Gaussian smoothing with kernel of 3 mm.

It is found that although the predicted Compton events image reconstruction shows a clear peak at the Bragg peak position when using *energy sum*, there is a quite broad activity distribution and even notable activity after the Bragg peak position resulting in inconsistent falloff. Therefore, benefiting from *recovered energy sum*, it is possible to significantly improve the PG reconstruction and greatly reduce the false activity after the Bragg peak.

Nevertheless, the LM-MLEM reconstruction of the result obtained from training BDT model with all possible features (9 variables) represents even more clear falloff distribution with less false activity after the Bragg peak (see Figure 4-20).

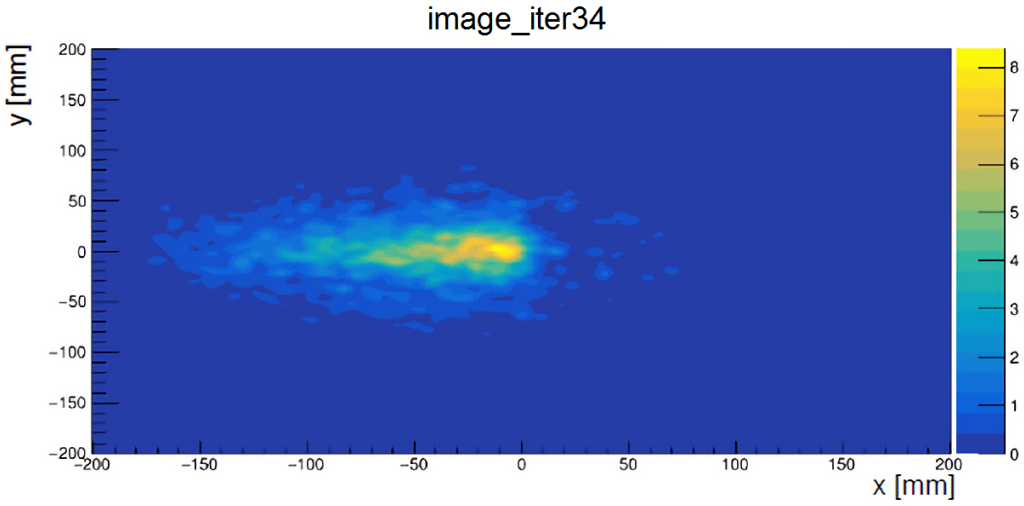


Figure 4-20: The reconstructed position distribution of predicted Compton events obtained from training the BDT using all possible features (9 variables) after 34 LM-MLEM iterations and applying the Gaussian smoothing with kernel of 3 mm.

To see the distal falloff behaviors of the predicted Compton events, the 1D depth-dose profiles of the reconstructed deposited dose for these two BDT models, correctly classified Compton events from the trained BDT model with 9 features, and the Compton events from Geant4 simulation are illustrated in Figure 4-21.

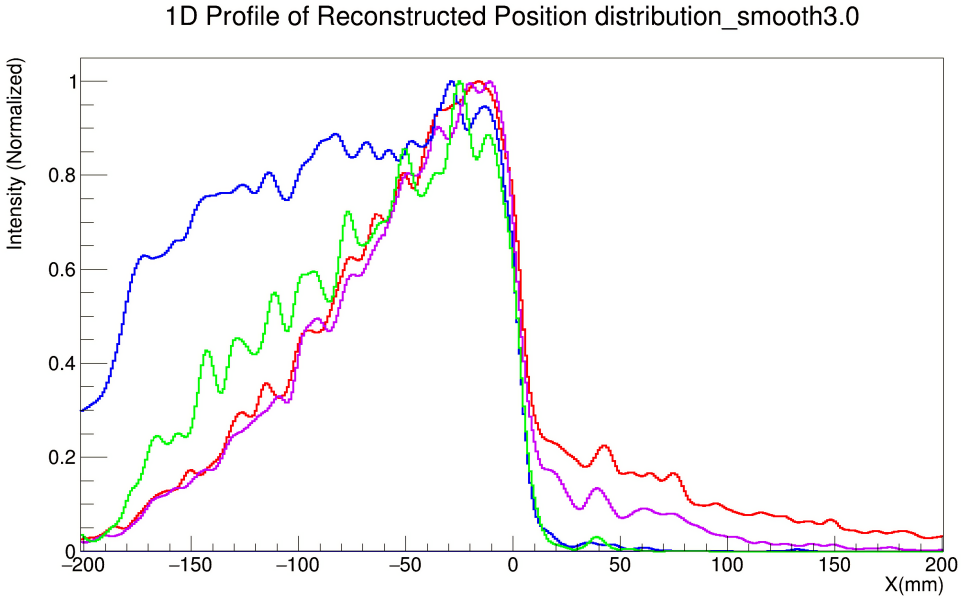


Figure 4-21: The depth-dose profile of predicted Compton reconstructed position along the beam axis (x-axis) for the models, the correctly classified Compton events, and the Compton events from Geant4 simulation. The blue curve shows the reconstructed image of 180000 Compton events from the simulation after 43 iterations. The red curve displays the reconstructed image result from training BDT with a few features (3 variables) after 40 iterations. The violet curve shows the results of training BDT with all possible features after 34 iterations. The green curve shows the reconstructed image of 10448 correctly classified Compton events obtained from training with all feasible features after 38 iterations. All depth-dose profiles were normalized by their maximum intensity value. The Gaussian smoothing with kernel of 3 mm was applied to all reconstructed profiles.

When comparing the 1D depth profiles of the reconstructed predicted Compton events obtained from the BDT models with the Compton events from Geant4 simulation, it is found that there is a very good agreement in the reconstructed falloff positions between them. Their profiles show a steep falloff at the end of the Bragg peak but with a tail in the case of the trained models' outputs. This lack of accuracy is due to not properly predicted Compton events whose *energy sum* were not precisely corrected. This tail is reduced in the high purity BDT model. When reconstructing only the correctly classified Compton events from the BDT

predictions, the produced image is more comparable to the image of the Compton events from Geant4 simulation. They show a very similar falloff behavior, and there is almost no activity after the Bragg peak position (see Figure 4-21). Therefore, as the purity increases, the reconstructed image of the model is more similar to the Compton events' reconstructed profile from the simulation, resulting in a better determination of the Bragg peak falloff position. Moreover, it can be seen that the 1D depth profiles of models' predictions are not overlapped with that of the Compton events from the Geant4 simulation for negative x values. This is due to the detector acceptance which is smaller for PG emitted far away from the detector centre (see section 4.1.3) in the case of the predicted Compton events however, the simulated Compton events are direct information from Geant4 and not affected by the detector response.

Distal Edge Determination Precision

For accurate determination of the PG distal edge, it is needed to obtain sufficient number of reconstructed events which are real coincidence events for a single beam spot in clinical uses. According to our recent article, 5000 events is the number of real coincidence events within SiFi-CC detector including Compton events [11]. To make an estimate of the statistical precision of the falloff position determination, several random subsets from the BDT model output should be selected. The number of events in the random subset is obtained from the multiplication of the ratio of the total number of events after and before event selection in the analysis phase (e.g., 40291/260663, in the case of training with all possible 9 features) and the number of real coincidence events obtained for a single beam spot.

The BDT model with higher purity was used to determine the Bragg peak distal edge. We selected 30 random subsets of the model output. The number of each subset is 773 events based on the results mentioned above. The procedure of distal edge determination for each random subset is as follows.

- The 1D depth profile of PG reconstructed position along the beam axis after sufficient iterations of LM-MLEM (i.e. reaching the convergence criterion) and applying Gaussian smoothing (kernel of 5 mm) was obtained.

Note: there is no sophisticated study on which kernel is the best. Different kernel values were tested and that one which visibly led to less fluctuation in the depth profile was selected as the best.

- In the depth profile, the first bin with the maximum deposited dose (intensity) value was found.
- Next, moving from that bin towards a larger depth, the bin was found where the intensity becomes minimum for the first time after the threshold fraction of half of the maximum intensity [87] (see Figure 4-22).
- A sigmoidal curve fitting method was applied to each PG reconstructed distribution [88].

$$I = \frac{I_{max} - I_{min}}{1 + \exp \frac{(X - X_0)}{dX}} + I_{min}, \tag{4.9}$$

where X_0 is the position of the half value between the maximum and minimum, dX is the width between these data points, I_{max} is the maximum intensity value and I_{min} is the minimum intensity value.

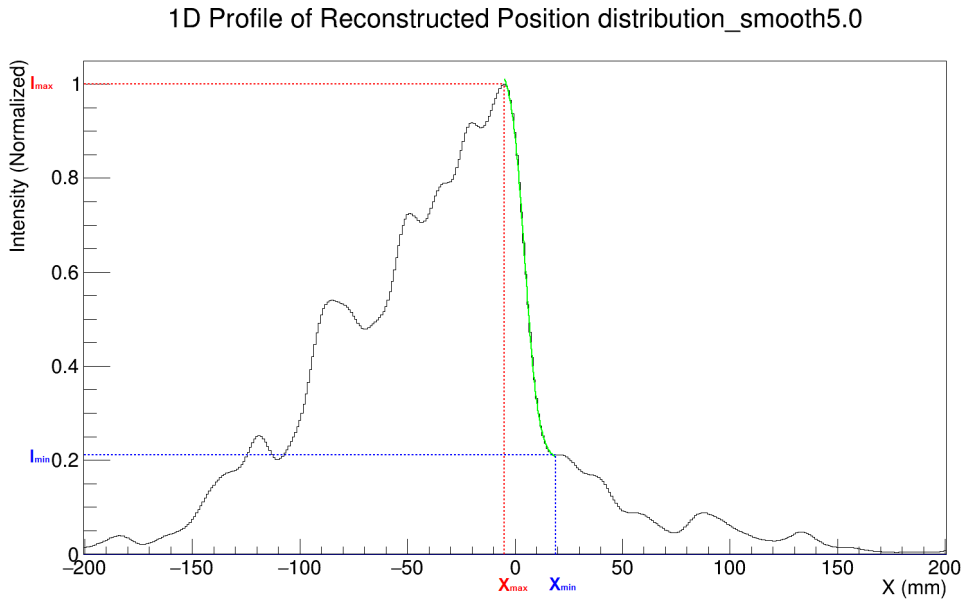


Figure 4-22: 1D depth profile of PG falloff behavior filtered by Gaussian smoothing with kernel of 5 mm and its sigmoidal curve fitting for a random subset of the data. The fitting result for this random subset shows that mean value X_0 and the outlier dX were obtained 2.6 mm and 2.1 mm, respectively.

Figure 4-22 shows the results of the sigmoidal curve fitting with the PG distribution for one random subset. It is expected that the location of the distal dose edge can be determined by the position of the mean value X_0. Moreover, dX shows the interval of the possible distal edge position for each subset. Finally, a distribution of the frequency of the mean values X_0 obtained from the fitting curve to each subset, was fitted by the Gaussian function (see Figure 4-23).

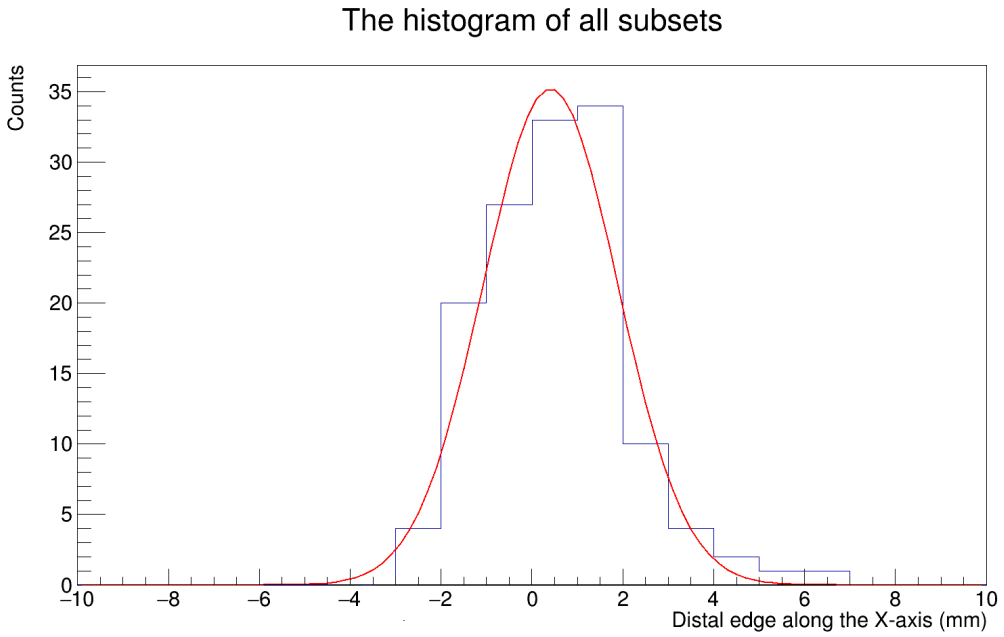


Figure 4-23: The distal dose edge position for a 180 MeV proton beam obtained from 30 random subsets of the BDT model output. The fitting result shows a mean value of 0.41 mm and a standard deviation value of 1.48 mm.

The result shows that PG distal edge position for a 180 MeV proton beam in a PMMA phantom tends to positive values and is located near the normalized Bragg peak position at ($x = 0$) with a good position resolution of 3.5 mm FWHM.

"If you optimize everything, you will always be unhappy."

Donald Knuth

Chapter 5

Discussion and Conclusions

Ion radiation therapy is an attractive alternative for cancer tumor treatment [5]. It employs the physical characteristics of the Bragg peak for better control over the deposited dose location [4]. However, due to the uncertainty in Bragg peak position, some safety margins must be used during the hadron therapy, resulting in some health risks [20]. Consequently, an online monitoring tool is needed to determine the Bragg peak position precisely. The SiFi-CC is a novel design for an online monitoring of dose distribution in proton therapy based on detection of PG radiation emitted from a patient during irradiation. Emitted PGs interact with the modules of the SiFi-CC, and the source position is then computed by reconstructing the Compton events [11, 12].

The goal of my thesis was to perform an event pattern recognition and reconstruction methods in Compton camera imaging for proton therapy monitoring. For this purpose, firstly, I developed a machine learning framework to identify the Compton events among various processes caused by interactions of PGs with SiFi-CC modules. The data set was simulated by Geant4 making it possible to study the response of a SiFi-CC detector. Secondly, I developed an image reconstruction framework based on LM-MLEM algorithm to reconstruct the source position of the predicted Compton events after event selection by the machine learning model, determining the Bragg peak distal dose edge position distribution.

In the first part, the SiFi-CC machine learning classifier is expected to take as input the event data that occurred in the detector, and outputs the event type (sig-

nal/background) and their corresponding positions and energies of the interactions inside the detector. The correlations between the positions and energies of recorded events besides the angular distribution term as features were studied for each event class. It turned out that including all possible features in the training may potentially lead to a better signal/background separation especially in the case of the event class with 2 cluster hits whose each cluster is in one of the detector modules.

The simple train/test split procedure is appropriate when there is a sufficiently large data set. However, in our study, due to relatively low number of events, especially in the case of event classes with 4 and 5 cluster hits, we decided to implement *k-fold* cross-validation method to make up this deficiency and control the overtraining avoidance as much as possible. The BDT, MLP and k-NN classifiers were trained through *k-fold* cross-validation and then their performances were assessed using the ROC curves for each event class. Among all classifiers, BDT was chosen for further analysis because of its robustness in signal/background separation. Besides training the BDT classifier with all possible features, the BDT classifier was also trained with a few features (excluding all randomly distributed cluster hits' positions) to make the model less complex and reduce the systematic error. Finally, the performance results of these two studies were compared.

In the second part, a preliminary study of geometric configuration of the SiFi-CC was performed using the LM-MLEM algorithm before simulating the detector response which is necessary for the machine learning stage as input. The next application of LM-MLEM was done through the reconstruction of the deposited dose of the BDT model's predictions after applying the optimal cuts to locate the distal dose edge. In this study, I used a *pixel-wise* convergence criterion to stop the LM-MLEM iterations of the image reconstruction. However, other convergence rules such as normalized root mean square deviation (NRMSD) and chi-square (χ^2) could be investigated and compared.

Moreover, the final reconstructed images were smeared by Gaussian smoothing to reduce the statistical fluctuations especially around the falloff region; leading to locating the distal edge more precisely. The kernel of Gaussian smoothing was chosen visually by comparing the depth-dose profiles with its different values. A

sophisticated study is needed to select the best value of Gaussian smoothing kernel which is beyond this study.

A 10-fold cross-validation of BDT classifier with all possible features achieved a recall of 73%, an efficiency of 11.4% and a purity of 26%. These results represent a notable improvement compared to the trained BDT model with a few features; reaching a relative increase of 44% in the ratio of correctly classified Compton events. A well enough agreement between the reconstructed deposited energy of the predicted Compton events and the simulated events from Geant4 simulation was obtained. However, in the case of models' predictions, there is still activity tail after the Bragg peak. This is due to less accuracy in total deposited dose prediction.

Finally, a distal dose edge determination study was performed. As a result, a good position resolution of 3.5 mm FWHM was achieved. It is anticipated that higher purity of the model's predictions may lead to a more comparable PG reconstructed position with that of the Compton events from the simulation, resulting in a better position resolution in distal edge determination. Therefore, finding more suitable features based on the physical basics for the model training and more sophisticated machine learning studies are needed to make this aim come true.

The results of this study showed that the SiFi-CC prototype is a promising approach to determine the distal edge position of the Bragg peak. Moreover, it could help assess the performance and optimize the feasible geometric configuration of SiFi-CC detector.

Bibliography

[1] Global cancer observatory: *Europe fact sheet - Cancer.* https://gco.iarc.fr/today/data/factsheets/populations/908-europe-fact-sheets.pdf, 2020.

[2] G. Delaney, S. Jacob, C. Featherstone, and M. Barton. *The role of radiotherapy in cancer treatment.* Cancer, 104:1129–1137, 2005.

[3] R. Baskar, K. A. Lee, R. Yeo, and K. W. Yeoh. *Cancer and radiation therapy: Current advances and future directions.* Int. J. Med. Sci., 9:193–199, 2012.

[4] W. P. Levin, H. Kooy, and J. Loeffler. *Proton beam therapy.* Br. J. Cancer, 93:849–854, 2005.

[5] R. R. Wilson. *Radiological use of fast protons.* Radiology, 47:487–491, 1946.

[6] A. C. Knopf and A. Lomax. *In vivo proton range verification: a review.* Phys. Med. Biol., 58:R131–R160, 2013.

[7] W. Enghardt, P. Crespo, F. Fiedler, R. Hinz, K. Parodi, J. Pawelke, and F. Pönisch. *Charged hadron tumour therapy monitoring by means of PET.* Nucl. Instrum. Methods Phys. Res. A., 525:284–288, 2004.

[8] K. Parodi. *On the feasibility of dose quantification with in-beam PET data in radiotherapy with ^{12}C and proton beams.* PhD thesis, Fakultat Mathematik und Naturwissenschaften der Technischen Universität Dresden, 2004.

[9] M. H. Richard. *Design study of a Compton camera for prompts-gamma imaging during ion beam therapy.* PhD thesis, Université Claude Bernard - Lyon I, 2012.

[10] K. Parodi, T. Bortfeld, and T. Haberer. *Comparison between in-beam and offline positron emission tomography imaging of proton and carbon ion therapeutic irradiation at synchrotron- and cyclotron-based facilities.* Int. J. Radiat. Oncol. Biol. Phys., 71:945–956, 2008.

[11] J. Kasper, K. Rusiecka, R. Hetzel, M. Kazemi Kozani, R. Lalik, A. Magiera, A. Stahl, and A. Wrońska. *The SiFi-CC project - Feasibility study of a scintillation-fiber-based Compton camera for proton therapy monitoring.* Phys. Med., 76:317–325, 2020.

[12] A. Wrońska. *Prompt gamma imaging in proton therapy - status, challenges and developments.* J. Phys. Conf. Ser., 1561:12–21, 2020.

[13] M. H. Richard, M. Dahoumane, D. Dauvergne, M. De Rydt, G. Dedes, N. Freud, J. Krimmer, J. M. Letang, X. Lojacono, V. Maxim, G. Montarou, C. Ray, F. Roellinghoff, E. Testa, and A. H. Walenta. *Design study of the absorber detector of a Compton camera for on-line control in ion beam therapy.* IEEE Trans. Nucl. Sci., 59:1850–1855, 2012.

[14] NuPECC report 2014: *Nuclear physics for medicine.* http://www.nupecc.org/pub/npmed2014.pdf.

[15] E. Fokas, G. Kraft, H. An, and R. Engenhart-Cabillic. *Ion beam radiobiology and cancer: Time to update ourselves.* Biochim. Biophys. Acta, 1796:216–229, 2009.

[16] M. Durante and J. S. Loeffler. *Charged particles in radiation oncology.* Nat. Rev. Clin. Oncol., 7:37–43, 2010.

[17] V. Bom, L. Joulaeizadeh, and F. Beekman. *Real-time prompt gamma monitoring in spot-scanning proton therapy using imaging through a knife-edge-shaped slit.* Phys. Med. Biol., 57:297–308, 2012.

[18] H. Paganetti. *Range uncertainties in proton therapy and the role of Monte Carlo simulations.* Phys. Med. Biol., 57:99–117, 2012.

[19] M. Moteabbed, S. España, and H. Paganetti. *Monte Carlo patient study on the comparison of prompt gamma and PET imaging for range verification in proton therapy.* Phys. Med. Biol., 56:1063–1082, 2011.

[20] A. Knopf, K. Parodi, T. Bortfeld, H. A. Shih, and H. Paganetti. *Systematic analysis of biological and physical limitations of proton beam range verification with offline PET/CT scans.* Phys. Med. Biol., 54:4477–4495, 2009.

[21] C. H. Min, C. H. Kim, M. Y. Youn, and J. W. Kim. *Prompt gamma measurements for locating the dose falloff region in the proton therapy.* Appl. Phys. Lett., 89:1–4, 2006.

[22] M. Pinto, M. Bajard, S. Brons, M. Chevallier, D. Dauvergne, G. Dedes, M. De Rydt, N. Freud, J. Krimmer, C. La Tessa, J. M. Letang, K. Parodi, R. Pleskac, D. Prieels, C. Ray, I. Rinaldi, F. Roellinghoff, D. Schardt, E. Testa, and M. Testa. *Absolute prompt-gamma yield measurements for ion beam therapy monitoring.* Phys. Med. Biol., 60:565–594, 2015.

[23] Joost M. Verburg and Joao Seco. *Proton range verification through prompt gamma-ray spectroscopy.* Phys. Med. Biol., 59:7089–7106, 2014.

[24] L. Kelleter, A. Wrońska, J. Besuglow, A. Konefał, K. Laihem, J. Leidner, A. Magiera, K. Parodi, K. Rusiecka, A. Stahl, and T. Tessonnier. *Spectroscopic study of prompt-gamma emission for range verification in proton therapy.* Phys. Med., 34:7–17, 2017.

[25] A. Zoglauer. *First light for the next generation of Compton and pair telescopes.* PhD thesis, Technische Universität Munchen, 2005.

[26] H. Seo, S. H. Lee, J. H. Jeong, J. H. Lee, C. S. Lee, J. S. Lee, and C. H. Kim. *AID – A novel method for improving the imaging resolution of a table-top Compton camera.* IEEE Trans. Nucl. Sci., 55:2527–2530, 2008.

[27] S. Agostinelli, J. Allison, and K. Amako. *Geant4 — A simulation toolkit.* Nucl. Instrum. Methods Phys. Res. A., 506:250–303, 2003.

[28] J. Allison, K. Amako, J. Apostolakis, and H. Araujo. *Geant4 developments and applications.* IEEE Trans. Nucl. Sci., 53:270–278, 2006.

[29] J. Allison, K. Amako, and J. Apostolakis. *Recent developments in Geant4.* Nucl. Instrum. Methods Phys. Res. A., 835:186–225, 2016.

[30] J. Kasper. *Optimization of the SiFi-CC for online range verification in proton therapy.* PhD thesis, RWTH Aachen University, 2021, in preparation.

[31] R. W. Todd, J. M. Nightingale, and D. B. Everret. *A proposed (gamma) camera.* Nature, 251:132–134, 1974.

[32] P. R. Edholm and G. T. Herman. *Linograms in image reconstruction from projections.* IEEE Trans. Med. Imaging, 6:301–307, 1987.

[33] T. Bortfeld and U. Oelfke. *Fast and exact 2D image reconstruction by means of Chebyshev decomposition and backprojection.* Phys. Med. Biol., 44:1105–1120, 1999.

[34] S. Vandenberghe, Y. DÁsseler, R. Van de Walle, T. Kauppinen, M. Koole, L. Bouwens, K. Van Laere, I. Lemahieu, and R. Dierckx. *Iterative reconstruction algorithms in nuclear medicine.* Comput. Med. Imaging Graph., 25:105–111, 2001.

[35] M. Defrise and G. T. Gullberg. *Image reconstruction.* Phys. Med. Biol., 51:R139–R154, 2006.

[36] S. J. Wilderman, N. H. Clinthorne, J. A. Fessler, and W. L. Rogers. *List-mode maximum likelihood reconstruction of Compton scatter camera images in nuclear medicine.* IEEE Nucl. Sci. Symp. Med. Imaging Conf. Rec., 3:1716–1720, 1998.

[37] T. Hebert, R. Leahy, and M. Singh. *Three-dimensional maximum-likelihood reconstruction for an electronically collimated single-photon-emission imaging system.* J. Opt. Soc. Am. A, 7:1305–1313, 1990.

[38] A. C. Sauve, A. O. Hero III, W. L. Rogers, S. J. Wilderman, and N. H. Clinthorne. *3D image reconstruction for a Compton SPECT camera model.* IEEE Trans. Nucl. Sci., 46:2075–2084, 1999.

[39] S. J. Wilderman, J. A. Fessler, N. H. Clinthorne, J. LeBlanc, and W. L. Rogers. *Improved modeling of system response in list mode EM reconstruction of Compton scatter camera images.* IEEE Trans. Nucl. Sci., 48:111–116, 2001.

[40] X. Lojacono. *Image reconstruction for Compton camera with application to hadrontherapy.* PhD thesis, Universite de Lyon, 2013.

[41] Y. Calderon. *Design, development and modeling of a Compton camera tomographer based on room temperature solid state pixel detector.* PhD thesis, Universidad Autonoma de Barcelona, 2014.

[42] S. Schoene, W. Enghardt, F. Fiedler, C. Golnik, G Pausch, H Rohling, and T. Kormoll. *An image reconstruction framework and camera prototype aimed for Compton imaging for in-vivo dosimetry of therapeutic ion beams.* IEEE Trans Rad. Plasma Med. Sci. (TRPMS), 1:96–107, 2016.

[43] M. Kolstein, G. De Lorenzo, and M. Chmeissani. *Evaluation of list-mode ordered subset expectation maximization image reconstruction for pixelated solid-state Compton gamma camera with large number of channels.* J. Instrum., 9:1–9, 2014.

[44] F. Chollet. *Deep learning with Python.* Manning, United States of America, 2017.

[45] O. Simeone. *A very brief introduction to machine learning with applications to communication systems.* IEEE Trans. Cogn. Commun. Netw., 4:648–664, 2018.

[46] A. Géron. *Hands-on machine learning with Scikit-learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, Inc., 2017.

[47] A. Hoecker, P. Speckmayer, J. Stelzer, J. Therhaag, E. von Toerne, and H. Voss. *TMVA: Toolkit for multivariate data analysis.* PoS, ACAT:040, 2007.

[48] B. Boehmke. *Hands-on machine learning with R.* https://bradleyboehmke.github.io/HOML/gbm.html, 2020.

[49] J. Brownlee. *A gentle introduction to the gradient boosting algorithm for machine learning.* https://machinelearningmastery.com/gentle-introduction-gradient-boosting-algorithm-machine-learning/, 2016.

[50] D. A. Clevert, T. Unterthiner, and S. Hochreiter. *Fast and accurate deep network learning by exponential linear units (ELUs).* ArXiv, 1511.07289, 2016.

[51] P. Kevin and D. K. Kang. *The effect of hyperparameter choice on ReLU and SELU activation function.* Int. J. Adv. Smart Convergence (IJASC), 6:73–79, 2017.

[52] D. Hendrycks and K. Gimpel. *Bridging nonlinearities and stochastic regularizers with Gaussian error linear units.* ArXiv, abs/1606.08415, 2016.

[53] G. Bontempi, M. Birattari, and H. Bersini. *Lazy learning for local modelling and control design.* Int. J. Control, 72:643–658, 1999.

[54] Z. Zhang. *Introduction to machine learning: k-nearest neighbors.* Ann. Transl. Med., 4:1–7, 2016.

[55] O. Harrison. *Machine learning basics with the k-nearest neighbors algorithm.* https://www.towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-\6a6e71d01761, 2018.

[56] A. Wrońska, R. Hetzel, J. Kasper, R. Lalik, A. Magiera, K. Rusiecka, and A. Stahl. *Characterisation of components of a scintillation- fiber-based Compton camera.* Acta. Phys. Pol. B., 15:17–25, 2020.

[57] A. Poitrasson-Rivière, B. A. Maestas, and M. C. Hamel. *Monte Carlo investigation of a high-efficiency, two-plane Compton camera for long-range localization of radioactive materials.* Prog. Nucl. Energy, 81:127–133, 2015.

[58] S. J. Wilderman, W. L. Rogers, G. F. Knoll, and Engdahl J. C. *Monte Carlo calculation of point spread functions of Compton scatter cameras.* IEEE Nucl. Sci. Symp. Med. Imaging Conf. Rec., 1:1538–1542, 1995.

[59] K. Rusiecka. private communication, 2017.

[60] Geant4 collaboration, *Guide for physics lists, release 10.4.* https://geant4-userdoc.web.cern.ch/UsersGuides/InstallationGuide/BackupVersions/V10.4/html/index.html/, 2017.

[61] IAEA, *Absorbed dose determination in external beam radiotherapy.* https://www-pub.iaea.org/MTCD/publications/PDF/TRS398_scr.pdf, 2000.

[62] H. Eickhoff, U. Weinrich, and J. Alonso. *Design criteria for medical accelerators. In: Ion beam therapy: fundamentals, technology, clinical applications.* Springer, Inc., 2012.

[63] R. Brun and F. Rademakers. *ROOT: An object oriented data analysis framework.* Nucl. Instrum. Meth. A, 389:81–86, 1997.

[64] J. Roser, E. Muñoz, and L. Barrientos. *Image reconstruction for a multilayer Compton telescope: an analytical model for three interaction events.* Phys. Med. Biol., 65:1–17, 2020.

[65] E. Draeger, S. Peterson, D. Mackin, H. Chen, S. Beddar, and J. C. Polf. *Feasibility studies of a new event selection method to improve spatial resolution of Compton imaging for medical applications.* IEEE Trans Rad. Plasma Med. Sci. (TRPMS), 1:358–367, 2017.

[66] J. Brownlee. *Train-test split for evaluating machine learning algorithms.* https://www.machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms, 2020.

[67] E. Muñoz, A. Ros, M. Borja-Lloret, J. Barrio, P. Dendooven, J. F. Oliver, I. Ozoemelam, J. Roser, and G. Llosá. *Proton range verifcation with MACACO II Compton camera enhanced by a neural network for event selection.* Sci. Rep., 11:1–12, 2021.

[68] S. Geisser. *The predictive sample reuse method with applications.* J. Am. Stat. Assoc., 70:320–328, 1975.

[69] J. Brownlee. *A gentle introduction to k-fold cross-validation.* https://machinelearningmastery.com/k-fold-cross-validation/, 2018.

[70] M. Kazemi Kozani and A. Magiera. *Machine learning-based event recognition in Compton camera imaging for proton therapy monitoring.* IEEE Nucl. Sci. Symp. Med. Imaging Conf., 2021.

[71] J. C. Polf, D. Mackin, E. Lee, S. Avery, and S. Beddar. *Detecting prompt gamma emission during proton therapy: The effects of detector size and distance from the patient.* Phys. Med. Biol., 59:2325–2340, 2014.

[72] C. Golnik, D. Bemmerer, W. Enghardt, F. Fiedler, F. Hueso-González, G. Pausch, K. Römer, H. Rohling, S. Schöne, L. Wagner, and T. Kormoll. *Tests of a Compton imaging prototype in a monoenergetic 4.44 MeV photon field - a benchmark setup for prompt gamma-ray imaging devices.* J. Instrum., 11:P06009, 2016.

[73] E. Muñoz, J. Barrio, A. Etxebeste, P. G. Ortega, C. Lacasta, J. F. Oliver, C. Solaz, and G. Llosá. *Performance evaluation of MACACO: A multilayer Compton camera.* Phys. Med. Biol., 62:7321–7341, 2017.

[74] A. Zoglauer. *Lessons learned from applying machine learning to the data analysis pipeline of the COSI telescope.* https://data-science.llnl.gov/sites/data_science/files/andreas_zoglauer_lessons_learned_from_applying_machine_learning_to_the_data_analysis_pipeline_of_the_cosi_telescope_0.pdf, 2018.

[75] *Hyperparameter (machine learning).* https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning), 2021.

[76] M. Claesen and B. De Moor. *Hyperparameter search in machine learning.* Meta-heuristics Int. Conf. (MIC), 14:1–5, 2015.

[77] M. Mithrakumar. *How to tune a decision tree?* https://towardsdatascience.com/how-to-tune-a-decision-tree-f03721801680, 2019.

[78] J. Heaton. *The number of hidden layers in neural network.* https://www.heatonresearch.com/2017/06/01/hidden-layers.html, 2017.

[79] M. Kuhn and K. Johnson. *Applied predictive modeling.* Springer, Inc., 2013.

[80] G. James, D. Witten, T. Hastie, and R. Tibshirani. *An introduction to statistical learning: with applications in R.* Springer, Inc., 2013.

[81] NIST/SEMATECH. *E-handbook of statistical methods.* https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm, 2003.

[82] H. Voss. *Introduction to multivariate analysis and TMVA.* https://indico.cern.ch/event/297180/contributions/1655947/attachments/557729/768436/LHCbMVA.pdf, 2010.

[83] P. Gueth, D. Dauvergne, N. Freud, J. M. L´etang, C. Ray, and D. Testa, E.and Sarrut. *Machine learning-based patient specific prompt-gamma dose monitoring in proton therapy.* Phys. Med. Biol., 58:4563–4577, 2013.

[84] R. Vidiyala. *Performance metrics for classification machine learning problems.* https://towardsdatascience.com/performance-metrics-for-classification-machine-learning-problems\-97e7e774a007, 2020.

[85] K. Markham. *Simple guide to confusion matrix terminology.* https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/, 2014.

[86] N. Kohlhase, T. Wegener, M. Schaar, A. Bolke, A. Etxebeste, D. Sarrut, and M. Rafecas. *Capability of MLEM and OE to detect range shifts with a Compton camera in particle therapy.* IEEE Trans. Rad. and Plasma Med. Sci.(TRPMS), 4:233–242, 2019.

[87] A. Morozov, H. Simões, and P. Crespo. *Distal edge determination precision for a multi-slat prompt-gamma camera: a comprehensive simulation and optimization of the detection system.* Phys. Med., 84:85–100, 2021.

[88] C. H. Min, H. R. Lee, C. H. Kim, and S. B. Lee. *Development of array type prompt gamma measurement system.* Med. Phys., 39:2100–2107, 2012.