Experimental Physics – Medical Physics

# Preliminary research on machine learning for X-ray CT calibration in proton therapy

Lourival Beltrão Martins Jr.
Medical Physics Master Program
Departamento de Física e Astronomia, Faculdade de Ciências, Universidade do Porto
2021

**Orientadores**

Dr. Chiara Gianoli, Experimental Physics – Medical Physics, Ludwig Maxilimians Universität - University of Munich

**Coorientadores**

Prof. Dr. Carla Rosa, FCUP, Departamento de Física e Astronomia, Faculdade de Ciências, Universidade do Porto

U. PORTO

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Todas as correções determinadas
pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, ______/______/________

Universidade do Porto

Masters Thesis

# Preliminary research on machine learning for X-ray CT calibration in proton therapy

Author:
Lourival Beltrão Martins Júnior

Supervisor:
Dr. Chiara Gianoli

Co-supervisor:
Prof. Dr. Carla Carmelo Rosa

The thesis was entirely accomplished at the Department of Experimental Physics – Medical Physics in the Faculty for Physics of the Ludwig-Maximilians-Universität München - LMU.
The thesis submitted in fulfilment of the requirements
for the degree of MSc. Medical Physics

at the

Faculdade de Ciências da Universidade do Porto
Departamento de Física e Astronomia

July 24, 2021

“ So, where are the others? ”

Enrico Fermi

# Acknowledgements

Since the beginning of this experience, I have relied on the support of countless people and institutions, which helped me achieve this moment and to whom I am very grateful.

Thanks to the Chair of Experimental Physics - Medical Physics, Ludwig Maxilimians Universität - University of Munich, Prof. Dr. Katia Parodi, for the opportunity to accomplish the entire research in this laboratory.

To Doctor Chiara Gianoli, advisor of the dissertation, I thank you for monitoring, supporting, enthusiasm, and sharing valuable knowledge for this work. I miei sinceri ringraziamenti, grazie!

To Doctor Sofia Isabel Silva and Prof. Doctor Carla Rosa, co-advisors of the dissertation, I thank you for monitoring and sharing valuable knowledge for this work.

I thank Mrs. Tânia Bravo Silva and Doctor José Eugênio Silva, amazing people that we had the privilege of meeting when we arrived in Portugal. Their support to our family was invaluable. Your gestures will never be forgotten.

I am grateful to my colleague Hugo Felipe Freitas, a young and intelligent scientist who demonstrates great potential in the academy.

I also thank my classmates Maria Clara, Rafaela Inácio, Catarina Pinto, Vasco Abreu, and Monique Witt for the fun moments during the course.

Finally, I deeply thank my wife, Rose Anna Furghestti, and our daughter Belly. Without you, this adventure would not be possible!

UNIVERSIDADE DO PORTO

# Abstract

Faculdade de Ciências da Universidade do Porto

Departamento de Física e Astronomia

MSc. Medical Physics

Preliminary research on machine learning for X-ray CT calibration in proton therapy

by Lourival Beltrão Martins Júnior

Proton therapy is an advanced form of radiation therapy that is increasingly used worldwide. Proton therapy is now moving away from static collimators and compensators toward greater use of Intensity Modulated Proton Therapy $IMPT$, a more recent modality. One problem that remains to be satisfactorily resolved is the so-called range uncertainly. The proton stopping power $(SP)$, has been estimated from X-ray CT images $(xCT)$ by converting Hounsfield Units $HU$ to relative proton stopping power $RSP$, expressed with respect to water. In this work, ML methods such as Random Forest, Gradient Boosting, Adaptive Boosting, and XGBOOST, were tested, aiming at developing a methodological framework that handles the patient-specific calibration of the $CT$ image based on proton radiography which aims to estimate the $RSP$ values, based on the WET as measured by the proton radiography, for a clinical $CT$ image of head and neck patient. The results obtained were, the work was limited to regression algorithms with limited processing of the input data, a strong dependence on the correspondence between the training and test datasets, the estimators did not perform similarly to generalizable models. Based on Bagging procedures, Random Forest Regressor presented lower qualitative performance than methods that use Boosting procedures. Among these, the Adaptive Boosting method showed superior qualitative performance concerning the Gradient Boosting Regressor and XGBOOST.

UNIVERSIDADE DO PORTO

# Resumo

Faculdade de Ciências da Universidade do Porto
Departamento de Física e Astronomia

Medical Physics Master Program

Preliminary research on machine learning for X-ray CT calibration in proton therapy

por Lourival Beltrão Martins Júnior

Terapia de protões é uma técnica avançada de radioterapia cada vez mais utilizada mundialmente. Os últimos avanços da técnica resultaram no surgimento de uma nova modalidade, terapia de protões de intensidade modulada $IMPT$, em detrimento da utilização de colimadores e compensadores estáticos. Apesar dos avanços da técnica, problemas como a incerteza associada ao alcance dos protões continuam a persistir. O cálculo do alcance dos protões baseia-se no poder de paragem do protão ($SP$). Este, por sua vez, é estimado através do uso de imagens de tomografia computorizada ($CT$) que são quantificadas em unidades de Hounsfield $HU$ e posteriormente convertidas em poder de paragem relativo $RSP$ (expresso relativamente à água). Neste trabalho, foram testados métodos de machine learning (ML), tais como, Random Forest, Gradient Boosting, Adaptive Boosting e XGBOOST, com o objetivo de desenvolver uma metodologia eficiente que permitisse estimar os valores de $RSP$ através da espessura equivalente de água (WET) obtida de uma radiografia de protões, para uma imagem $CT$ clínica de um paciente de cabeça e pescoço. Os resultados obtidos pautauram-se por algoritmos de regressão com processamento limitado dos dados de entrada e forte dependência da correspondência entre os conjuntos de dados de treinamento e teste, resultando em estimadores com um desempenho pobre quando comparados com os modelos generalizáveis. Procedimentos baseados em Bagging, como é o exemplo do Random Forest Regressor, apresentaram desempenhos qualitativos inferiores quando comparados com procedimentos baseados em Boosting. Dentre estes últimos, o método Adaptive Boosting apresentou desempenho qualitativo superior em relação ao Gradient Boosting Regressor e ao XGBOOST.

# Contents

# List of Figures

# List of Tables

# Glossary

| | |
|---|---|
| ADABOOST | Adaptive Boosting |
| CT | Computed Tomography |
| DNA | Deoxyribonucleic Acid |
| DTR | Decision Tree Regressor |
| GBR | Gradient Boosting Regressor |
| HU | Hounsfield Unit |
| ICRU | International Commission on Radiation Units and Measurements |
| $I_m$ | Mean Excitation energy |
| IMPT | Intensity Modulated Proton Therapy |
| IMRT | Intensity Modulated Radiation Therapy |
| LET | Linear Energy Transfer |
| MC | Monte Carlo |
| MAE | Mean Absolute Error |
| ME | Maximum Error |
| MCS | Multiple Coulomb Scattering |
| MLP | Most Likely Path |
| MSE | Mean Squared Error |
| RF | Random Forest |
| RBE | Relative Biological Effectiveness |
| RSP | Relative Stopping Power |
| SP | Stopping Power |

| | |
|---|---|
| VMAT | Volumetric Modulated Arc Therapy |
| WET | Water Equivalent Thickness |
| WEPL | Water Equivalent Path Length |

This thesis is dedicated to Rose Anna Furghestti, my wife, for her love and support.

# Chapter 1

# Introduction

In oncology, at some point among the various processes that exist to treat the disease, procedures involving the use of radiation in the execution of the treatment will certainly be an option to be considered [1]. Currently, among the treatments widely available commercially, the use of photon and electron beams stands out, these proven to be effective in their role in delivering radiation doses. However, the use of ions, such as protons, carbon ions among others [2], classified in the scientific world as a new branch of radiotherapy, called Particle Therapy, sometimes also called Hadron Therapy, it presents itself as a potentially more accurate[3] option in the process of delivering the absorbed dose. Within these possibilities, we consider the widely spread in clinical radiation oncology use of protons, Proton Therapy.

The suggestion to use high energy protons for radiotherapy treatment was made by the physicist Robert R. Wilson, founder of the Fermi National Accelerator Laboratory (Fermilab) in its original publication in the year 1946 [4]. Therefore, a long path of research, technological as well methodological, has been developed in academic circles to the use of cancer therapy.

Proton therapy is now moving away from static collimators and compensators toward greater use of Intensity Modulated Proton Therapy *IMPT*, a more recent modality in which a pencil beam [3] while its energy and intensity are modulated according to the treatment program, which can include treatment from multiple directions [5]. *IMPT* not only allows proton therapy to be applied to tumors that could not be accessed via the older proton-therapy treatment methods but also allows better control of the dose distribution and, therefore, greater success in sparing critical organs from harmful radiation dose. Proton therapy has to be evaluated not just in terms of the success of the irradiation in degenerating the tumor but most importantly in terms of how well the effects of the radiation on the rest of the body are minimized [5].

Even with the advent of $IMPT$ the field of proton therapy has almost certainly not yet reached its full potential. One problem that remains to be satisfactorily resolved is the so-called range uncertainly [5]. For proton therapy to be successful, the range of the therapy protons in the tissue of the patient must be accurately known in advance to plan the treatment and then machine the compensator or to program the $IMPT$.

Up to now, the average energy dissipated by ionization in a medium per unit path length, the proton stopping power ($SP$), has been estimated from X-ray CT images ($xCT$) by converting the X-ray absorption (linear attenuation coefficient) measured in Hounsfield Units $HU$ to relative proton stopping power $RSP$, the stopping power expressed with respect to water [5]. The main interest is to build a curve which relates $HU$ and $RSP$ directly. Therefore, as a result we obtain a so called calibration curve between $HU$ and $RSP$.

There are different methodologies to obtaining $RSP$ values. One well recognized is called Stoichiometric calibration [6]. Stoichiometric calibration is based on a data table [7] which contains mean anatomic composition values from human tissues. It's worth to highlight that the values tabulated are different relating to real tissue. To strengthen this argument, it is a fact that persons present different percentages of chemical composition a different phase of life (e.g, a young patient, from the percentage composition point of view, shows different values than middle age or even an old patient). Furthermore, the composition of a given type of tissue may depend on its location in the body (e.g, pelvic kidney).

An alternate methodology to theorize the $RSP$ values was proposed by Bethe-Bloch's formulation [8, 9] to obtain those values from quantities relating to the medium and physics parameters from the beam used. One of the most important parameters used into their arguments is defined as mean excitation energy ($I_m$) from a specific substance or a medium that presents slightly different values depending on the publication for which it was generated [10]. After the ICRU Report 37 [11] work was completed, new values of 80.0, 81.8, and 77.0 eV were reported. The difference in the $I_m$ values between 75 and 80 eV results in 0.8–1.2% differences in the $SP$ in the energy range of 10–250 MeV, which implies the same impact on absorbed doses [10]. Consequently, different levels of accuracy are present and influence on the range calculations for the proton beam.

Besides the problem with these different approaches, X-rays interact very differently in material compared with protons, resulting in relations between $HU$ and $RSP$ that are not unique and can therefore be ambiguous [12]. If, on the one hand, the calculation of the $RSP$ has inaccurate aspects, on the other hand, the CT number is a scale that depends on the electronic and electrotechnical control relating to the manufacturer of the CT device. CT number directly relates to the scanner spectrum emitted for each CT device, originally built and installed. Therefore, an empirical performance between $HU$ and a tissue structure is established for each specific device.

Artifacts in X-ray images can produce additional errors in calibration curve of $HU$ to $RSP$. Uncertainties up to 11% in the head [13] have been reported. More recent work predicts typical errors of 1.8% and 1.1% for bone and soft tissue respectively [14], although the presence of higher density materials, and the resulting beam-hardening artifacts, can result in larger errors for specific cases, depending on the position of the high density sample within the body and the size of the body. The use of Monte Carlo simulations can be a significant aid in reducing these uncertainties [3].

Thereby, errors inherent relating $HU$ and $RSP$ values also influencing various other aspects that compose the entire planning process for the dose delivery, since patient positioning and immobilization, passing through critical intermediate steps, such as target volume definition and healthy tissue delineation, and finally, the plan approval. These aspects have a considerable influence over the target structure's margin choices and the final dose absorbed calculation.

This work is aimed at developing a methodological framework that handles the patient-specific calibration of the $CT$ image based on proton radiography which aims to estimate the $RSP$ values, based on the water equivalent thickness $WET$ as measured by the proton radiography, for a clinical $CT$ image of head and neck patient, using for that some common methods that stand out when using Machine Learning, such as Random Forest, Gradient Boosting, Adaptive Boosting, and XGBOOST.

# Chapter 2

# Proton Therapy

> ”It must have occurred to many people that the particles now become of considerable therapeutic interest”.
>
> Dr. Robert R. Wilson

Protons have very interesting characteristics for the optimization of the radiation dose calculation process. These characteristics allow a greater conformation in the distribution of the radiation dose in relation to the target to be irradiated and in the same way, save healthy tissues. The physical interaction properties of protons provide a more pronounced biological effect. As a proton passes through the medium, it decelerates sequentially. The energy it deposits per unit of distance (called linear energy transfer, or $LET$) increases until all of its energy runs out and stops abruptly [15]. Thus, in a uniform medium, for example, a water phantom, a monoenergetic beam of protons leads to the formation of the characteristic Bragg curve. The Bragg peak is a pronounced peak on the Bragg curve which plots the energy loss of ionizing radiation during the proton travel, Fig. 2.1.

Since protons are much heavier than electrons, Coulomb's interactions with electrons do not deviate them considerably from their original path. However, Coulomb scattering of the nuclei, although occurring much less frequently, leads to wider angle scattering and contributes to substantial lateral scattering of proton beams. This leads to an increased transfer of energy from the proton beam, especially when the protons decelerate near the end of their range. Proton interactions with nuclei occur with even less probability and mainly at higher energies and lead to wide-angle scattering and the production of secondary particles, including neutrons [15].

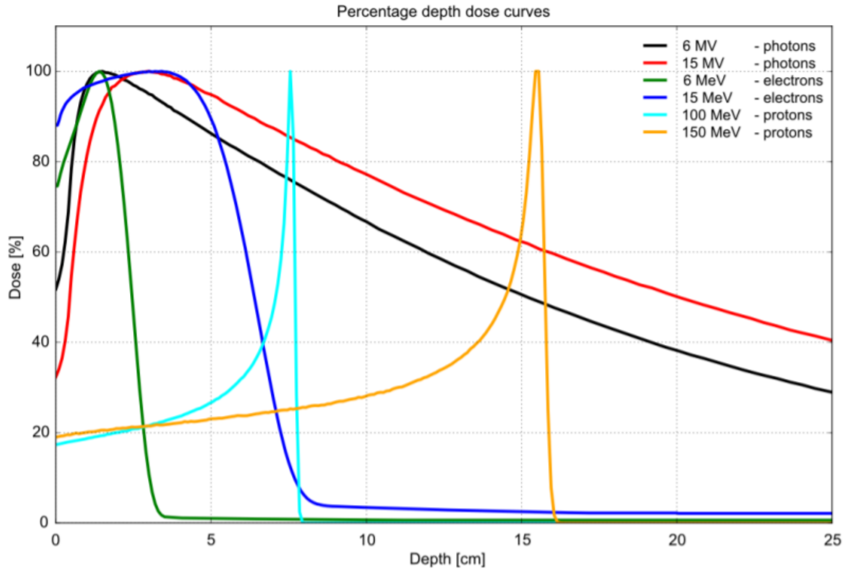


Figure 2.1: Percentage depth dose curves in a water phantom for different energies and beams modalities. The measured electron and photon depth dose curves correspond to a $10x10cm^2$ field delivered using a Varian Truebeam accelerator, the MC - calculated (Geant4) proton integral depth dose curves correspond to a pencil beam [16]

.

## 2.1 A brief physical theory resume

### 2.1.1 Stopping Power

Stopping Power $S(E)$ is defined as the energy loss of charged particle per unit length. The kinetic energy of protons is the energy source for the ionization, excitation, and nuclear interactions. The proton loses its energy while moving through matter and stops when the kinetic energy is lost. The energy loss of protons is caused by the interactions occurring in the matter, mainly by collisional interactions. The matter has the ability to stop protons, which depends on its characteristics, such as the chemical nature of the compounds and physical density[17].

$$S(E) = -\frac{dE}{dx} \tag{2.1}$$

where E and x are the kinetic energy and the path length of the charged particle, respectively. S(E) is composed of two parts, stopping power by collisional $S_{col}$ and radiative process $S_{rad}$ [17].

### 2.1.2 Relative Stopping Power

Conveniently, the stopping power of tissues is commonly considered relative to water $RSP$ since this physical quantity is almost independent on the energy for the range used in proton therapy [18].

$$RSP = \frac{S\left(E\right)}{S_{water}\left(E\right)} \tag{2.2}$$

### 2.1.3 Bethe-Bloch Equation

The theory of the energy loss of proton was developed by Hans Bethe and Felix Bloch around 1933. They considered the proton–electron collision with quantum mechanics and relativity [17]. Bethe–Bloch equation is generally expressed as

$$-\frac{dE}{dx} = 4\pi N \left(\frac{e^2}{4\pi\varepsilon_0}\right)^2 \frac{z^2 Z}{m_c}\frac{1}{v^2}\left[ln\frac{2m_e v^2}{I} - \ln\left(1-\beta^2\right) - \beta^2 - \frac{\delta}{2} - \frac{Z}{C}\right][J/m] \tag{2.3}$$

$$-\frac{dE}{dx} = 4\pi N\frac{z^2 Z e^4}{m_c}\frac{1}{v^2}\left[ln\frac{2m_e v^2}{I} - \ln\left(1-\beta^2\right) - \beta^2 - \frac{\delta}{2} - \frac{Z}{C}\right][MeV/m] \tag{2.4}$$

where N is the electron density of the material, $m_e$ is the mass of the electron, z is the charge of the incident particle, Z is the atomic number of the material, I is the mean excitation potential of the material, $\delta$ is the density correction, and C is the shell correction. I is commonly set to 75 eV for liquid water, as shown in ICRU Report 37 [11] and ICRU Report 49 [19]. N can be written as $N_A\rho/M$, where $N_A$ is the Avogadro's constant, $\rho$ is the mass density, and M is the molar mass of the material. From the eq. 2.4, it can be seen that dE/dx is proportional to the density and atomic number of the material, as well as the charge of the particle but independent of the particle mass. Thus, the protons and deuterons have the same $dE/dx$ in a particular material. $dE/dx$ significantly changes with the velocity $v$ of the particle [17]. At a high velocity, where $\beta$ is close to 1, the $dE/dx$ of any particle is almost constant, and has almost the same value for all particles of unit charge. Due to the proportionality between N and $\rho$, the electron density $N_A\rho Z/A$ has a direct influence on $dE/dx$. This is a key aspect when planning a treatment based on proton therapy.

The proton slows down as it penetrates the matter and loses its kinetic energy. $dE/dx$ increases with the depth until the proton comes to rest. As a result, a peak of $dE/dx$, called Bragg peak, is formed. The width of the peak is smaller than that of the depth dose distribution of the photon beam, which we are familiar with. The shrinkage of energy straggling explains it [17].

It is important to note that in proton planning on the treatment planning system, the obtaining $RSP$ values are accomplished from Bethe and Bloch formalism application Eq. (2.3), first suggested by Schneider at al [6],

$$RSP = \rho_{e,t} \frac{\ln\left[\frac{2m_e c^2 \beta^2}{I_t\left(1-\beta^2\right)}\right] - \beta^2}{\ln\left[\frac{2m_e c^2 \beta^2}{I_{water}\left(1-\beta^2\right)}\right] - \beta^2} \tag{2.5}$$

where $\rho_{e,t}$ is the relative electron density of the tissue, $m_e$ is the mass of the electron, $\beta$ is the relative speed of the incident protons, $I_t$ is the mean excitation energy of the tissue, and $I_{water}$ is the mean excitation energy of water [18].

### 2.1.4 Linear Energy Transfer

Most of the lost kinetic energy of the proton is locally transferred to matter while the rest escape as delta and gamma rays. Since LET is defined in a very limited vicinity along the track, the energy carried away by $\delta$-*rays*, photons, and neutrons is not taken into account [17].

The LET is defined in ICRU report 85 [20] as

$$L_\Delta = -\frac{dE_\Delta}{dx} \tag{2.6}$$

where $dE_\Delta$ is the mean energy loss through interactions with electrons while crossing a distance dx, excluding electrons with kinetic energy in excess of $\Delta$. $L_\Delta$ is called restricted energy transfer, where $\Delta$ is referred to as an energy cutoff, since the energy transferred to electrons with the kinetic energy greater than $\Delta$ is not considered. The LET largely depends on the value of $\Delta$. When $\Delta$ is 100 eV, it is expressed as $L_{100}$. For a low $\Delta$, the energetic recoil electrons originated in the collisions of delta rays ($\delta$-*rays*) with the matter, are excluded, and only energy transferred to local matter is taken into account. $L_\infty$, the unrestricted energy transfer, refers the LET without energy cutoff and involves all the electronic interactions including $\delta$-*rays*. Thus, $L_\infty$ is equal to the electronic stopping power [17].

### 2.1.5 Range

The range R traveled by a charged particle, with a certain amount of energy in a certain material medium, corresponds to the expected value of a path's length that follows until it stops (discounting the thermal movement). This amount cannot be measured experimentally unless some kind of 3D image of the particle track is available. Thus, trying to find a way to make a possible record, an amount that can properly express this quantity is an average value, $R_{mf}$, of a path, defined as the expected value of the maximum depth of penetration of the particle in its initial direction, which can be determined experimentally. The concepts of R and $R_{mf}$ are illustrated in Fig. 2.2; both quantities are non-stochastic and are usually stated in units of mass/area (e.g. $gcm^{-2}$ ); in general, they exclude the effects of nuclear interactions [21].

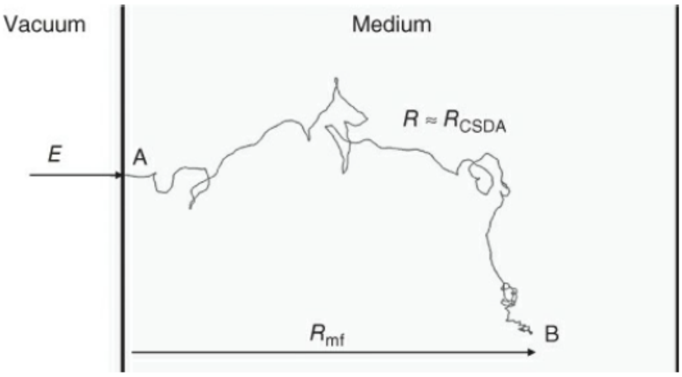


Figure 2.2: Illustration of the concepts of range, R, and maximum depth of penetration or mean forward range, $R_{mf}$, for an electron. R is the average distance along the deviated path from the point of entry A to the stopping point B. Note that $R_{mf}$ is not necessarily the depth of the terminal point B, as the electron can be backscattered. The electron track has been generated using the Monte Carlo system PENELOPE [22] for the simulation of a 1 MeV electron in lead. For a heavy charged particle, both quantities are very similar due to its reduced multiple scattering (MCS) [21].

## 2.2 Some important Imaging Concepts

### 2.2.1 Hounsfield Unit

A *CT* scanner reconstructs the value of linear attenuation coefficients $\mu$ at each pixel within a cross section. Different *CT* scanners, however, have different x-ray tubes, which in turn have different effective energies. Thus, the exact same object will produce different numerical values of $\mu$ on different scanners. Worse, since the x-ray tube on a busy *CT* scanner may need to be replaced about once every year, the same *CT* scanner will produce a different scan of the same object in successive years. This is clearly not a desirable situation [23]. In order to compare data from different scanners, which may have different x-ray sources and hence different effective energies, $CT$ numbers are computed from the measured $\mu$ coefficients at each pixel. The $CT$ number is defined as

$$CT_{number} = 1,000 \times \frac{\mu - \mu_{\text{water}}}{\mu_{\text{water}}} \tag{2.7}$$

and is expressed in HU. Clearly, $CT_{number} = 0$ HU for water, and since $\mu = 0$ in air, we find that $CT_{number} =$ - 1000 HU for air [23]. The largest CT numbers typically found naturally in the body are for bone, where $CT_{number} \approx 1000$ HU for average bone, although CT numbers can surpass $CT_{number} \approx 3000$ HU for metal and contrast agents. Usually, CT numbers are rounded or truncated to the nearest integer, they are typically reproducible to about ± 2 HU between scans and across scanners [23].

### 2.2.2 Imaging Reconstruction

The basic $CT$ measurement is a line integral of the effective linear attenuation coefficient within a cross section. But line integrals are not what is desired; it is required a tomographic image of $\mu$, or equivalently its $CT$ number, over the entire cross section. Therefore, the important step is to reconstruct the image of $\mu$ from a collection of line integrals. For this propose we use the case of $CT$ image reconstruction from projections for parallel-ray geometry [23].

Let x and y be rectilinear coordinates in the plane. A line $L$ in the plane is given by

$$L(l, \theta) = x \cos\theta + y \sin\theta = l, \tag{2.8}$$

where $l$ is the lateral position of the line and $\theta$ is the angle of a unit normal to the line, as shown in Fig. 2.3. The line integral of function $f(x, y)$ is given by

$$g(l, \theta) = \int_{-\infty}^{\infty} f(x(s), y(s)) ds, \tag{2.9}$$

where

$$x(s) = l \cos\theta - s \sin\theta, \tag{2.10}$$

$$y(s) = l \sin\theta + s \cos\theta. \tag{2.11}$$

This is a parametric form of the line integral; an alternate expression is given by

$$g(l,\theta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - l)dxdy \tag{2.12}$$

Here, the shifting property of the 1-D impulse function $\delta(.)$[see item 2.6 reference [23]] causes the integrand to be zero everywhere except on the line $L(l,\theta)$. The integral acts on the impulse function by integrating the values of $f(x,y)$ only along the line; hence, it takes a line integral.

For a fixed $\theta$, $g(l,\theta)$ is called a projection; for all l and $\theta$, $g(l,\theta)$ is called the 2-D Radon transform of $f(x,y)$. The relationship of a projection to the object $f(x,y)$ is shown in Fig. 2.3. If we make the identifications

$$f(x,y) = \mu(x,y;\bar{E}), \tag{2.13}$$

$$g(l,\theta) = -ln(\frac{I_d}{I_0}) \tag{2.14}$$

It has seen that this mathematical abstraction exactly characterizes the CT measurement situation [23]. In what follows, It is assumed that $g(l,\theta)$ correspond to the measurements and $f(x,y)$ corresponds to the underlying unknown function or object that it is desirable to reconstruct. Notice that the definition of a projection corresponds to a collection of line integrals for parallel lines. Hence, these are called parallel-ray projections, and they correspond in geometry to first generation CT scanners only [23].

An image of $g(l,\theta)$ with l and $\theta$ as rectilinear coordinates is called a sinogram. It is a pictorial representation of the Radon transform of $f(x,y)$ and represents the data that are necessary to reconstruct $f(x,y)$.

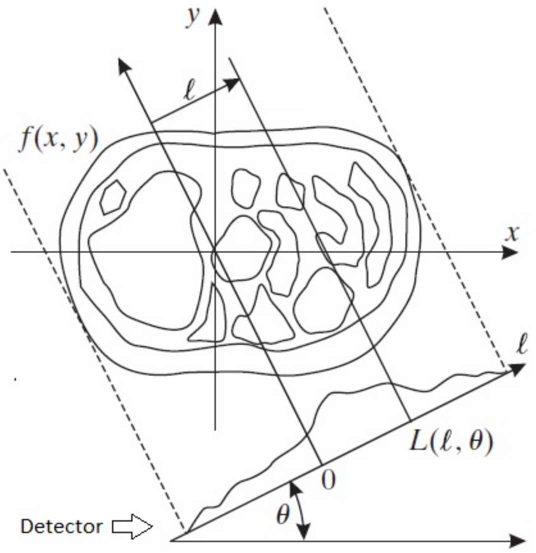


Figure 2.3: Represents the geometry of lines and projections. Image from page 199, reference [23]

## 2.3 Proton Imaging

In 1963, Cormack [24] proposed the concept of proton radiography *pRad* and proton computed tomography *pCT*. The potential development of these tools had some advantages over conventional X-ray computed tomography *xCT*. However, a very important barrier in the image formation process was found, the low spatial resolution in *pCT*. This is because the multiple scattering events refer to the interaction with the nucleus's Coulomb fields in the absorbing material, resulting in many small-angle deflections in the proton trajectory. [25].

The proton imaging device is not implemented in the clinical routine yet. Due to increasing proton therapy center enterprises, there is a lot of research at the academic level to make it available as soon as important technical challenges are overcome.

### 2.3.1 Water Equivalent Thickness

Water equivalent thickness closely mimics the properties of human tissues in terms of energy loss, multiple coulomb scattering, and nuclear interactions. As such, water is a recommended phantom material for dose and range measurements and reference material for reporting corresponding calculated quantities [19]. For example, it is a common and convenient clinical practice to specify the penetrating power of a proton beam by its range in water [19]. In this way, range losses in various beam-line components and the patient may be easily added or subtracted from one another in a physically consistent and intuitive way. Viewed another way, it is also convenient to specify the range-absorbing power of various objects in the beam path, e.g. beam transmission monitors and immobilization devices, by their equivalent thickness as if they were made of water [26].

WET[*], Eq. (2.15), is often used to characterize the beam penetration range, Fig. 2.4. For treatment sites with nearby critical structures, e.g. an optic nerve, the range of the planned and delivered beams must agree within a few millimeters. To accomplish it, treatment planning systems are commonly configured with the WET values of all items not included in the planning CT images, such as components (as mentioned) in the treatment head, immobilization devices not present during the CT scan, or a treatment couch [26]. Similarly, to determine the measurement geometry for patient-specific clinical quality assurance measurements, the WET of measurement instruments and possibly other devices must be determined [26]. Thus, it is important to have methods to calculate and measure WET [26].

$$WET = -\int_{E_{out}}^{E_{in}} \frac{dE}{SP\left(I_w, E\right)} dx \tag{2.15}$$

Regarding image reconstruction, based on pencil beams, the tomographic domain is parameterized in $(x, y, z)$ with z representing the rotational axis of the imaging system. For each z, the projection lines are defined as $\rho = x\cos\theta + y\sin\theta$, where $\theta$ is the projection angle, while the integration lines are defined as $\delta = y\cos\theta + x\sin\theta$, and thus perpendicular to the projection line [12]. The pencil beam position is therefore identified by $(\rho, z)$ in the radiographic domain (i.e., the sinogram), which is therefore a discrete form parameterized in $(\rho, z, \theta)$ [12].

$$WET(\rho, z, \theta) = \varepsilon \sum_{\delta=1}^{D} WET(\rho\cos\theta - \delta\sin\theta, \delta\cos\theta + \rho\sin\theta, z) = \varepsilon \sum_{\delta=1}^{D} WET(\delta, \rho, z, \theta) \tag{2.16}$$

Where $\delta = \{1, ..., D\}$ is the discretization index of the integration line. Both radiographic and tomographic domains are discretized by the pencil beam binning $\varepsilon$, which represents the voxel size of the tomographic domain and the pixel size of the radiographic domain, respectively [12]. The rationale of the thesis is found within proton imaging, with particular reference to proton radiography.

[*] Some authors draw a distinction between WET and water equivalent path length (WEPL). This thesis uses just WET

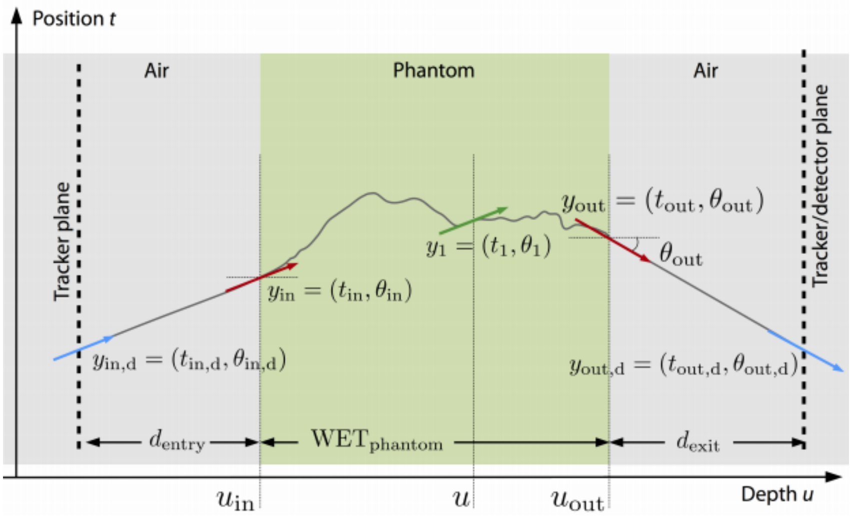

Position $t$
Air
Phantom
Air
Tracker plane
Tracker/detector plane
$y_{\text{out}} = (t_{\text{out}}, \theta_{\text{out}})$
$y_1 = (t_1, \theta_1)$
$\theta_{\text{out}}$
$y_{\text{in}} = (t_{\text{in}}, \theta_{\text{in}})$
$y_{\text{in,d}} = (t_{\text{in,d}}, \theta_{\text{in,d}})$
$y_{\text{out,d}} = (t_{\text{out,d}}, \theta_{\text{out,d}})$
$d_{\text{entry}}$
$\text{WET}_{\text{phantom}}$
$d_{\text{exit}}$
$u_{\text{in}}$
$u$
$u_{\text{out}}$
Depth $u$

Figure 2.4: Detection scheme and scattering geometry of proton inside a phantom. Coloured arrows indicate the position and propagation direction of a proton. The solid "zigzag" line is to exemplify a single proton trajectory. The dashed vertical lines summarise pairs of tracker detectors each. Page 5 from [27].

### 2.3.2 Proton Image Detector Configuration

the residual energy after that proton has traversed the object of interest. (With MC simulations, we have access to the energy loss along the trajectory but not in reality)

Proton imaging consists of measuring the residual energy after that proton has traversed the object of interest (with MC simulations, we have access to the energy loss along the trajectory but not in reality), and then using this information to calculate the *WET* of the proton trajectory. Due to the weak energy dependence of the *RSP* in an arbitrary medium, information on the mean residual ion range gained at higher initial beam energies than the therapeutic ones can still be used for indirect verification of the *HU-RSP* calibration curve integrated all along the path in the patient [28]. Besides, the radiographic measurement can also convey information on the actual distribution of ion beam residual ranges, showing a natural broadening around the mean value due to the initial beam momentum spread, the statistical fluctuations of the energy loss as well as the spread (due to small lateral deflections) of multiple ion paths in the traversed tissue yet sharing the same stopping place. Whereas Gaussian distributions can approximate the first two processes, the latter is clearly influenced by the different stopping properties of the medium encountered along the different ion paths. Thus, this broadening also referred to as range dilution, provides a quantitative measure of tissue inhomogeneities exploited, e.g., at the stage of treatment planning for selection of beam incidence angles more robust against ion range uncertainties [29].

Over half a century ago, during a seminar on image reconstruction by line integrals, Cormack suggested a solution which has been encouraging the development of Proton Radiography [24]. It has a series of potential clinical applications such as; patient set-up and passive range verification [30]; patient specific calibration of *CT-HU* numbers to *RSP* [12, 30, 31]; potential improvement of patient treatment planning [25] in addition to its potential use for tumor tracking, active range verification and adaptive proton therapy treatment [32]. During the last decades, the technological development of proton imaging detectors defined two main types of detector configurations, namely list-mode and integration-mode.

#### 2.3.2.1 Single particle tracking systems (List-mode)

In this configuration, proton radiography relies on proton trackers placed before and after the patient and on a detector to measure the residual energy. With the list-mode detector configuration, and assuming that a straight-line path in proton radiography leads to poor spatial resolution [33], it is necessary to develop a probabilistic model for protons' movement trajectory in uniform water. Thus this estimate is given by the maximum likely path - MPL [25], which is based on the inclusion of the Fermi-Eyges MCS model in the Bayes formalism [34–37]. By tracking protons individually, their trajectories inside the patient may be estimated and linked to their measured energy loss, Fig. 2.5.

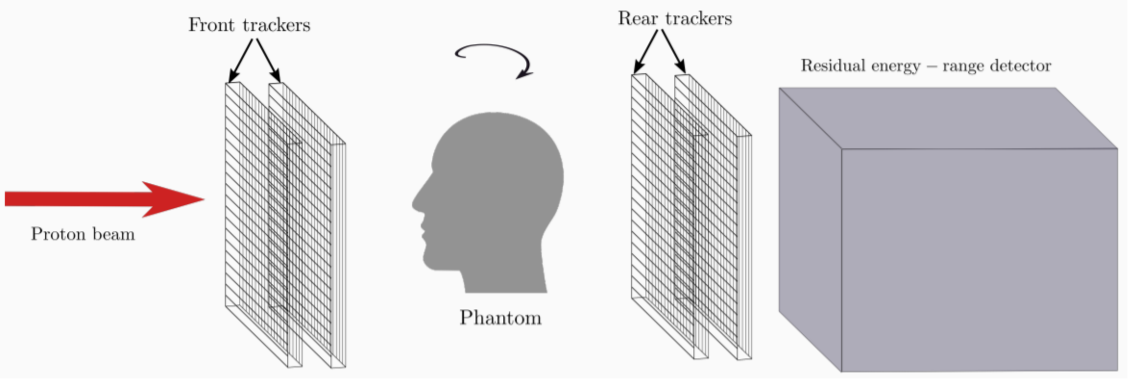


Figure 2.5: List-mode configuration [38].

#### 2.3.2.2 Integration-mode

In the integration-mode detector configuration, the *WET* of the proton trajectory is instead stored as a histogram for the proton ensemble of each pencil beam (i.e., the WET histogram of the WET components) [37]. In integration-mode set-ups, an ensemble of protons traverses the phantom and is captured in the detector, but differently from tracking protons individually, the averaging is of physical nature and essentially takes place already during image acquisition [27].

## 2.4 Calibration of CT Numbers for RSP

### 2.4.1 Calibration by tissue equivalent

During the semi-empirical process of the calibration curve definition, between $HU$ and $RSP$, uncertainty in the order of up to 5% may occur [12].

The most direct way to achieve the conversion of $CT$ Number into $RSP$ aims the use of tissue-equivalent, Fig. 2.6, with intend to obtaining an empirical relation of correspondence. Follows some procedures involved in it:

$(i)$ From the tissue surrogates, the $CT$ number is experimentally measured in a CT scanner.

$(ii)$ The corresponding $RSP$ values of the tissue surrogates can be measured directly by the proton beam or theoretically calculated from Eq. (2.5).

$(iii)$ By a simple linear interpolation of the correspondences of those points, we can obtain a plot $HU$ vs $RSP$. See Fig. 2.7.

This method presents an important disadvantage. The $HU$-$RSP$ relationship can result non-bijective* [6].

The $CT$ number average of each inserted tissue substitute in the head and body phantom should be used to reduce the beam-hardening effect [39]. Surrogates manufacturers design them so that they exhibit the same attenuation characteristics for a given energy range and so that they show similar physical properties such as mass and density. Then, because the surrogate's atomic compositions differ from those of real tissues, their attenuation characteristics cannot be identical for the entire energy range used in radiation therapy, particularly for low energy x-rays and high energy protons [18]. Examples of common non-human tissue materials are listed in tables 2.1 and 2.2.

*When there is not a one-to-one correspondence between those sets

| Tissue substitute Material | 80 KVp | 100 kVp | 120 kVp | 140 kVp |
|---|---|---|---|---|
| LN-300 Lung | -683.0 | 692.4 | -684.9 | 689.3 |
| LN-450 Lung | -565.1 | -574.51 | -568.2 | -571.6 |
| AP6 Adipose | -104.7 | -98.2 | -90.0 | -86.2 |
| BR-12 Breast | -46.3 | -46.5 | -44.1 | -42.6 |
| CT Solid Water | 6.2 | -0.5 | -0.8 | -2.7 |
| BRN-SR2 Brain | 14.6 | 20.3 | 25.6 | 29.6 |
| LV1 Liver | 87.1 | 82.5 | 79.2 | 78.9 |
| IB Inner Bone | 283.6 | 230.3 | 196.4 | 176.5 |
| B200 Bone Mineral | 308.8 | 255.0 | 218.7 | 198.4 |
| CB2 - 30% CaCO3 | 575.1 | 494.3 | 438.4 | 407.6 |
| CB2 - 50% CaCO3 | 1057.6 | 900.9 | 790.0 | 727.6 |
| SB3 Cortical Bone | 1602.0 | 1360.6 | 1188.6 | 1092.0 |

Table 2.1: CT numbers of rod materials as measured with a GE $CT/i^{TM}$ scanner. The CT numbers given for the materials used in the Gammex 467 Tissue Characterization Phantom [40] are the result of measurements made with a specific scanner and sample holding fixture. Data obtained from other scanners may vary due to differences in the filtration and reconstruction algorithms used.

| Tissue substitute Material | Relative electron density | Relative stopping power |
|---|---|---|
| LN-300 Lung | 0.283±0.014 | 0.282±0.014 |
| LN-450 Lung | 0.433±0.009 | 0.432±0.009 |
| AP6 Adipose | 0.910±0.021 | 0.917±0.022 |
| BR-12 Breast | 0.962±0.008 | 0.968±0.006 |
| CT Solid Water | 0.990±0.002 | 0.991±0.001 |
| Water | 1.000 | 1.000 |
| BRN-SR2 Brain | 1.041±0.010 | 1.054±0.012 |
| LV1 Liver | 1.057±0.016 | 1.060±0.018 |
| IB Inner Bone | 1.087±0.014 | 1.074±0.016 |
| B200 Bone Mineral | 1.099±0.009 | 1.085±0.010 |
| CB2 - 30% CaCO3 | 1.276±0.003 | 1.260±0.005 |
| CB2 - 50% CaCO3 | 1.464±0.004 | 1.423±0.007 |
| SB3 Cortical Bone | 1.694±0.002 | 1.626±0.005 |

Table 2.2: The RSP values were calculated for 208 MeV proton energy. List of tissue substitute inserts in the RMI 467 [41] tissue characterization phantom and the associated ranges of relative electron densities and relative stopping powers for the five phantoms.

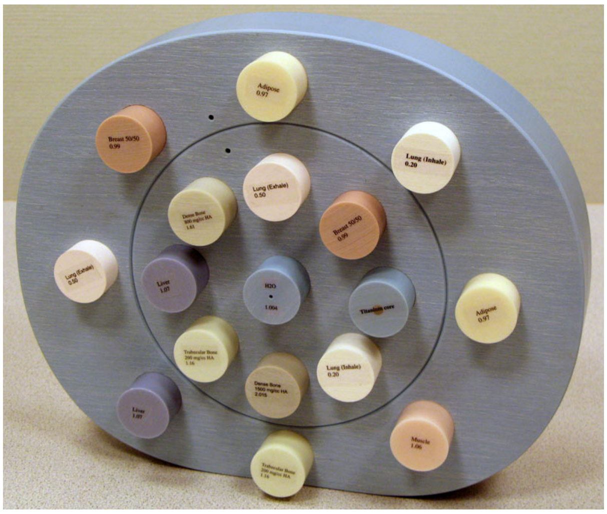


Figure 2.6: Electronic density phantom, model 062M, made by CIRS Tissue Simulation & Phanton Technology.

The relation between *HU* and *RSP*, according to Table 2.3, was obtained by a linear interpolation*. Particularly, the *xCT* is converted to the *pCT* relying on a clinical-like calibration curve, such as the one used at Heidelberg Ion-Beam Therapy - HIT for treatment planning of head and neck patients [42].

| Points | HU | RSP |
|---|---|---|
| 1 | -1024 | 0.0001 |
| 2 | -1000 | 0.003 |
| 3 | -200 | 0.8 |
| 4 | -115 | 0.94 |
| 5 | -40 | 0.995 |
| 6 | 0 | 1 |
| 7 | 78 | 1.075 |
| 8 | 204 | 1.11 |
| 9 | 1595 | 1.7 |
| 10 | 1974 | 1.778 |
| 11 | 3096 | 2.077 |

Table 2.3: *HU* and *RSP* Values

*from https://docs.scipy.org/doc/scipy/reference/tutorial/interpolate.html

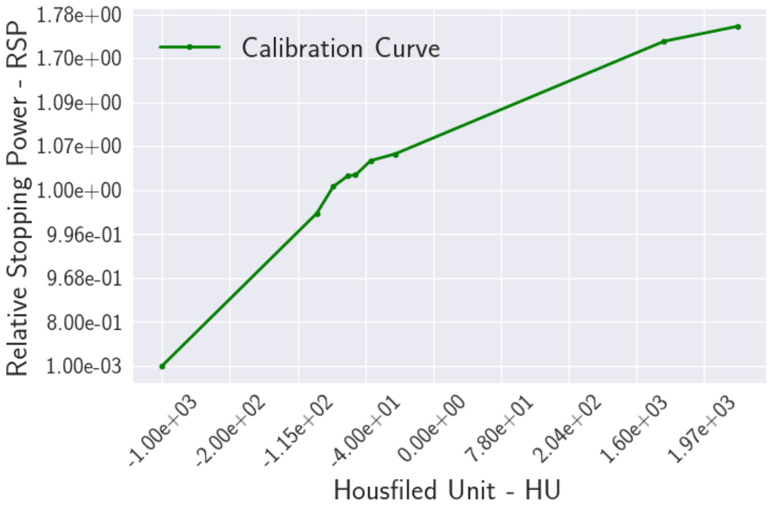


Figure 2.7: *CT* number [HU] versus RSP, calibration curve used for this thesis, 120-kV *CT* scanner, tissue equivalent method.

### 2.4.2 Stoichiometric calibration

Although the stoichiometric calibration overcomes the approximation of the elemental composition for biological tissue with tissue equivalent inserts, tissue-specific and patient-specific variations of elemental compositions are responsible for up to 3% of remaining inaccuracies in the calibration curve [12, 43, 44]. This method is very interesting and was proposed [6], as was cited, to overcome the disadvantage of the calibration by tissue-equivalent materials. Follows some procedures involved in it:

$(i)$ Use linear regression to fit the measured *CT* numbers of the tissue substitutes Eq. (2.17) to determine the coefficients, A, B, and C, which characterize the cross-sections for the photoelectric effect, coherent scattering, and incoherent scattering, respectively, of the *kV x-ray* beam interacting with the phantom and the insert [39]:

$$HU = \rho_e^{rel}(A\tilde{Z}^{3.62} + B\hat{Z}^{1.86} + C) \tag{2.17}$$

where $\rho_e^{rel}$ is the electron density relative to water; $\tilde{Z} = [\Sigma\lambda_i Z_i^{3.62}]^{1/3.62}$, $\hat{Z} = [\Sigma\lambda_i Z_i^{1.86}]^{1/1.86}$, $\lambda_i$, is the fraction of number of electrons per unit volume for element $i$; and $Z_i$ is the atomic number of element $i$ [39].

$(ii)$ Compute the CT numbers of various reference tissues as given in the international Commission on Radiological Protection *ICRP*, Report 23 [45] by using Eq. (2.17).

$(iiii)$ Compute the $RSP$ for the same $ICRP$, mentioned in item $(ii)$ by using an approximation to the Beth-Block expression, Eq. (2.5) [6, 39].

$(iv)$ Plot a calibration curve with appropriate fit through the data points, $CT$ numbers vs $RSP$.

# Chapter 3

# Machine Learning

“All models are wrong; some models are useful”.

Dr. George Box

Early applications of machine learning in radiation oncology focused on predicting normal tissue toxicity, but its application has since branched into almost every part of the field, including tumor response modeling, radiation physics quality assurance, contouring and treatment planning, image-guided radiotherapy, respiratory motion management, among others [46].

For the successful application of machine learning in general and in medical physics and radiation oncology in particular, one first need to properly characterize the nature of problem, in terms of the input data and the desired outputs. Secondly, despite the robustness of machine learning to noise, a good model cannot substitute for bad data, keeping in mind that models are primarily built on approximations. Additionally, this has been stated as the GIGO principle, (”garbage in, garbage out”) as shown in Fig. 3.1 [47] . Thirdly, the model needs to generalize the observed data into unseen data [46].

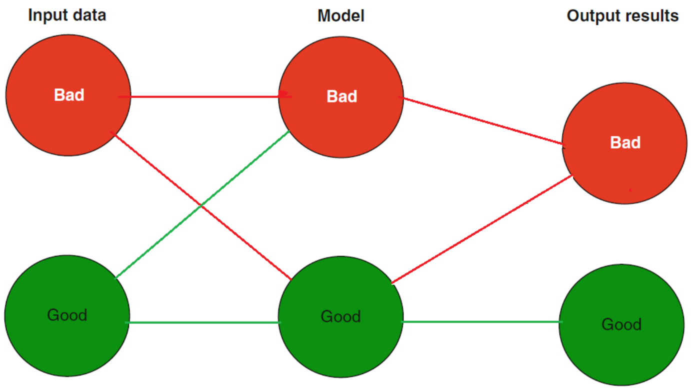


Figure 3.1: GIGO paradigm. Learners cannot be better than the data.

Machine learning is a method that aims to develop computer algorithms that can mimic human intelligence by introducing propositions from neuroscience, probability and statistics, computer science, information theory, and philosophy with successful applications in computer vision, robotics, and medicine, among others. The development and application of machine learning have undergone a significant surge in recent years due to the exponential growth and data science availability. Machine learning techniques are an important tool to understand different data in many fields, including radiation oncology [46].

## 3.1 Unsupervised Learning

Unsupervised learning is a class of ML techniques used to find patterns in data. The data given to unsupervised algorithms is not labelled, which means only the input variables are given with no corresponding output variables. In unsupervised learning, the algorithms are left to discover relevant structures in the data on their own.

## 3.2 Reinforcement Learning

It is a method of learning by interacting with an environment. A reinforcement learning procedure learns from the consequences of its actions rather than from being explicitly taught. It selects its actions based on its past experiences and new choices, essentially with trial and error learning. The reinforcement signal that the reinforcement learning agent receives is a numerical reward that encodes the success of an action's outcome. The agent seeks to learn to select actions that maximize the accumulated reward over time.[*]

[*] Reinforcement Learning concept getting from http://www.scholarpedia.org/article/Reinforcement_learning

## 3.3 Supervised Learning

Supervised learning, also known as supervised machine learning, is a subcategory of machine learning and artificial intelligence. It is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights through a reinforcement learning process, which ensures that the model has been fitted appropriately. Supervised learning helps organizations solve for a variety of real-world problems.[*]

Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function until the error has been sufficiently minimized. Loss function or cost function is a function that maps an event or values of one or more variables into a real number intuitively representing some ”cost” associated with the event [48].

Supervised learning can be separated into two types of problems when data mining, classification, and regression.

### 3.3.1 Classification

Classification algorithms are predictive calculations used to assign data to preset categories by analyzing sets of training data.

### 3.3.2 Regression

Linear regression is a method for modeling the relationship between one or more independent variables and a dependent variable when the data suggests a linear relationship between the variables. In the simplest case, linear regression models the relationship between two scalar values, the input variable $x$, and the output variable $y$. The model assumes that $y$ is a linear function or a weighted sum of the input variable [49].

$$y = f(x) \tag{3.1}$$

Or stated with the coefficients $b_0$ and $b_1$.

$$y = b_0 + b_1 \times x_1 \tag{3.2}$$

[*] Supervising Learning concept getting from https://www.ibm.com/cloud/learn/supervised-learning

The method can also be used to model an output variable given multiple input variables $x_1, x_2, \ldots, x_i$, called multivariate linear regression

$$y = b_0 + (b_1 \times x_1) + (b_2 \times x_2) + \ldots + (b_i \times x_i) \tag{3.3}$$

The objective of creating a linear regression model is to find the values for the coefficient $b$ that minimize the error in the prediction of the output variable $y$.

Suppose we have 4 observations of the output variable $y$ and the 3 input variables $x_1, x_2, x_3$. Putting

$$\mathrm{X} = \begin{pmatrix} 1 & x_{1,1} & x_{1,2} & x_{1,3} \\ 1 & x_{2,1} & x_{2,2} & x_{2,3} \\ 1 & x_{3,1} & x_{3,2} & x_{3,3} \\ 1 & x_{4,1} & x_{4,2} & x_{4,3} \end{pmatrix} \tag{3.4}$$

$$b = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} \tag{3.5}$$

$$y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} \tag{3.6}$$

where $X$ is the matrix of the input data, $y$ is the vector of output variables for each row in $X$, and $b$ is the vector of coefficients. we can construct the interpolation system

$$y = \mathrm{X} \cdot b \tag{3.7}$$

Therefore, the problem becomes a system of linear equations where the $b$ vector values are unknown. That interpolation system is referred to as over-determined because there are more equations than there are unknowns, i.e. each coefficient is used on each row of data. It is impossible to solve because there are multiple inconsistent solutions, e.g. multiple possible values for the coefficients. Further, all solutions will have some error because there is no line that will pass nearly through all points, therefore the approach to solving the equations must be able to handle that. A problem like that can be framed as the minimization of squared error, called least squares, and can be recast in the language of linear algebra, called linear least squares. This is typically achieved by finding a solution where the values for $b$ in the model minimize the squared error. These are called linear least squares [49].

$$||X \cdot b - y||^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} X_{i,j} \cdot \left(b_j - y_i\right)^2 \tag{3.8}$$

This formulation has a unique solution as long as the input columns are independent (e.g. uncorrelated). We cannot always get the error $e = b - Ax$ down to zero. When $e$ is zero, $x$ is an exact solution to $Ax = b$. When the length of $e$ is as small as possible, $\hat{x}$ is a least squares solution [49].

Linear least squares problems can be solved efficiently on computers using matrix operations such as matrix factorization. Least squares is most known for its role in the solution to linear regression models, but also plays a wider role in a range of machine learning algorithms.

## 3.4 Ensembles Methods

The ensemble method consists of combinations of simpler models and low predictive power to result in a stronger model with greater accuracy. Within the Ensemble model class, there are two techniques commonly known as Boosting and Bagging [50].

### 3.4.1 Boosting and Bagging

Boosting, initially named Hypothesis Boosting, consists on the idea of filtering or weighting the data that is used to train our team of weak learners[*], so that each new learner gives more weight or is only trained with observations that have been poorly classified by the previous learners.

Bagging predictors is a method for generating multiple versions of a predictor[†] and using these to get an aggregated predictor. The aggregation averages over the versions when predicting a numerical outcome and does a plurality vote when predicting a class[‡].

[*] A 'weak learner' is any machine learning algorithm (for regression/classification) that provides an accuracy slightly better than random guessing.

[†] A unitary method that makes isolated predictions

[‡] referring to the classification carried out by "weak learners."

The multiple versions are formed by making bootstrap* replicates of the learning set and using these as new learning sets. Tests on real and simulated datasets using classification and regression trees and subset selection in linear regression show that bagging can give substantial gains in accuracy. [51]

Likewise, the Bagging methodology, the Boosting method, is also trained by individual samples, however, the combination method is not the preliminaries results, but a weighting of the performance of each model [52]. Within the class of Boosting models, the Gradient Boosting was chosen to be used in this work.

### 3.4.2 Random Forest - RF

Random Forest is one of the most popular and most powerful machine learning algorithms. Random forest is just a combination of many simple decision trees. For instance, we train M different trees on different subsets of the data, chosen randomly with replacement, and then compute the ensemble

$$f\left(X\right) = \sum_{m=1}^{M} \frac{1}{M} f_m\left(X\right) \tag{3.9}$$

where $f_m$ is the $m$'th tree. It is a type of ensemble machine learning algorithm called Bootstrap Aggregation or bagging [51].

### 3.4.3 Adaptive Boosting - ADABOOST

The ADABOOST algorithm, introduced in 1995 by Freund and Schapire [53], solved many of the practical difficulties of the earlier boosting algorithms. The algorithm takes as input a training set $(x_1, y_1), \ldots, (x_m, y_m)$ where each $x_i$ belongs to some domain or instance space X, and each $y_i$ is in some label set $Y$. ADABOOST calls a given weak or base learning algorithm repeatedly in a series of rounds $t = 1, \ldots, T$. One of the main ideas of the algorithm is to maintain a distribution or set of weights over the training set. The weight of this distribution on training example $i$ on round $t$ is denoted $D_t(i)$. Initially, all weights are set equally, but on each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set.

The weak learner's job is to find a weak hypothesis $h_t : X \rightarrow \{-a, +a\}$ appropriate for the distribution $D_i$. The goodness of a weak hypothesis is measured by its *error*

*A bootstrap sample is a sample obtained by sampling with replacement and may be repeated.

$$\epsilon_t = Pr_{i \sim D_i}\left[h_t(x_i) \neq y_i\right] = \sum_{i:h_i(x_i) \neq y_i} D_i(i). \tag{3.10}$$

The error is measured with respect to the distribution $D_i$ on which the weak learner was trained. In practice, the weak learner may be an algorithm that can use the weights $D_i$ on the training examples. Alternatively, when this is not possible, a subset of the training examples can be sampled according to $D_i$, and these (unweighted) resampled examples can be used to train the weak learner.

For $t = 1, \ldots, T$:

$(i)$ Train weak learner using distribution $D_i$

$(ii)$ Get weak hypothesis $h_t : X \rightarrow \{-a, +a\}$ with error $Pr_{i \sim D_i}\left[h_t(x_i) \neq y_i\right]$

$(iii)$ Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$

$(iv)$ Update:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_i} & \text{if } h_t(x_i) \neq y_i \\ e^{\alpha_i} & \text{if } h_t(x_i) = y_i \end{cases}$$

$$= \frac{D_i(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$$

where $Z_t$ is a normalization factor (chosen so that $D_{t+1}$ will be a distribution)

$(iv)$ Output the final hypothesis:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

#### 3.4.3.1 Parameters for ADABOOST

From the Sklearn library regarding AdaBoost method, follows some parameters in highlight:

$(i)$ $base\_estimator$ : The base estimator from which the boosted ensemble is built. If None, then the base estimator is $DecisionTreeRegressor$* initialized with $max\_depth = 3$ [55].

$(ii)$ $n\_estimators$ : The maximum number of estimators at which boosting is terminated. In case of perfect fit, the learning procedure is stopped early. [55].

$(iii)$ $learning\_rate$ : Learning rate shrinks the contribution of each regressor by $learning\_rate$. There is a trade-off between $learning\_rate$ and $n\_estimators$ [55].

*Randomize the internal decisions of the learning algorithm [54]

$(iv)$ loss : The loss function to use when updating the weights after each boosting iteration [55].

$(v)$ *random_state* : Controls the random seed given at each *base_estimator* at each boosting iteration. Thus, it is only used when *base_estimator* exposes a *random_state*. In addition, it controls the bootstrap of the weights used to train the *base_estimator* at each boosting iteration [55].

### 3.4.4 Gradient Boosting - GB

The GB is a well-known method and it has been used to solving classification and regression problems [52]. This method consists of an arrangement of combinations of previous models still considered unsatisfactory, manipulated iteratively to provide a robust model.

In view of internal calculations that aim to adjust the GB method, the main objective is to minimize a defined cost function. In this application of the GB method, it was decided to use cost function using least squares.

$$L = \sum (y_i - \hat{y}_i)^2 \tag{3.11}$$

Where, $y_i = i$th is a value of a response variable, $\hat{y}_i = i$th is a value for a predicted variable, $L(y_i, \hat{y}_i)$ is here called cost function.

With the iterations of the GB method, the goal is that the cost function is minimized to zero. Using the descendent gradient function for prediction, it is possible to estimate the values that satisfy the minimum cost function, according to:

$$\hat{y}_i = \hat{y}_i + \phi \cdot \delta \sum \frac{(y_i - \hat{y}_i)^2}{\delta \hat{y}_i} \tag{3.12}$$

become

$$\hat{y}_i = \hat{y}_i - \phi \cdot 2 \cdot \sum (y_i - \hat{y}_i)^2 \tag{3.13}$$

where $\phi$ is the learning rate of the algorithm and $\Sigma(y_i - \hat{y}_i)^2$ represents the sum of residuals.

Due to act directly on the estimators during each iteration, $\phi$ has an important role in trying to reduce a possible over-fitting behavior, thus contributing to building a model with generalization capacity.

Once defined the main mathematical functions of the GB method, cost function, and descent gradient, it follows a rationale that describes its operation [56].

$(i)$ Iteration number $m = 1$ up to M.

$(ii)$ Cost function definition

$(iii)$ Residuals minimization $\hat{h}_m(x_i) = \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}^{(m-1)}(x)}$

$(iv)$ Adjusting the previous function with the residuals found up to iteration M. $\hat{f}_m = \hat{f}_{m-1}(x) + \phi_m \hat{h}_m(x)$

### 3.4.4.1 Parameters for GBR

Another essential aspect in the application of the GB method is the choice of some parameters, among which we highlight:

$(i)$ loss : loss function to be optimized. [57]

$(ii)$ *learning_rate* : learning rate shrinks the contribution of each tree by *learning_rate*. There is a trade-off between *learning_rate* and *n_estimators* [57].

$(iii)$ *n_estimators* : The number of boosting stages to perform. Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance. [57]

$(iv)$ *min_samples_split* : The minimum number of samples required to split an internal node. [57]

$(v)$ *min_samples_leaf* : The minimum number of samples required to be at a leaf node. A split point at any depth will only be considered if it leaves at least min_samples_leaf training samples in each of the left and right branches. This may have the effect of smoothing the model, especially in regression [57].

$(vi)$ *max_depth* : maximum depth of the individual regression estimators. The maximum depth limits the number of nodes in the tree. Tune this parameter for best performance; the best value depends on the interaction of the input variables. [57]

## 3.4.5 eXtreme Gradient Boosting - XGBOOST

XGBoost is short for eXtreme Gradient Boosting, was introduced in 2014 [58]. The eXtreme refers to speed enhancements such as parallel computing and cache awareness that makes XGBoost approximately 10 times faster than traditional Gradient Boosting. In addition, XGBoost includes a unique split-finding algorithm to optimize trees, along with built-in regularization that reduces overfitting.

As mentioned in section 3.4.4, GB divides the optimization problem into two parts by first determining the direction of the step and then optimizing the step length. Different from GB, XGBoost tries to determine the step directly by solving

$$\frac{\partial L\left(y, f^{(m-1)}(x) + f_m(x)\right)}{\partial f_m(x)} = 0 \tag{3.14}$$

for each $x$ in the dataset. By doing second-order Taylor expansion of the loss function around the current estimate $f^{(m-1)}(x)$, we get

$$L\left(y, f^{(m-1)}(x) + f_m(x)\right) \approx L\left(y, f^{(m-1)}(x)\right) + g_m(x) f_m(x) + \frac{1}{2} h_m(x) f_m(x)^2 \tag{3.15}$$

where $g_m(x)$ is the gradient, same as the one in GB, and $h_m(x)$ is the Hessian[*] (second order derivative) at the current estimate:

$$h_m(x) = \frac{\partial^2 L\left(y, f^{(m-1)}(x)\right)}{\partial f(x)^2}_{f(x)=f^{(m-1)}(x)} \tag{3.16}$$

Then the loss function can be rewritten as

$$L(f_m) \approx \sum_{i=1}^{n} \left[ g_m(x_i) f_m(x_i) + \frac{1}{2} h_m(x_i) f_m(x_i)^2 \right] + const \tag{3.17}$$

$$\propto \sum_{j=1}^{T_m} \sum_{i \in R_{jm}} \left[ g_m(x_i) w_{jm} + \frac{1}{2} h_m(x_i) w_{jm}^2 \right] \tag{3.18}$$

Letting $G_{jm}$ represent the sum of gradient in region $j$ and $H_{jm}$ be qual to the sum of hessian in region $j$, the equation can be rewritten as

$$L(f_m) \propto \sum_{j=1}^{T_m} \left[ G_{jm} w_{jm} + \frac{1}{2} H_{jm}(x_i) w_{jm}^2 \right] \tag{3.19}$$

With the fixed learned structure, for each region, it is straightforward to determine the optimal weight:

$$w_{jm} = -\frac{G_{jm}}{H_{jm}}, j = 1, \cdots, T_m \tag{3.20}$$

Plugging it back to the loss function, we get

[*] Hessian matrix definition from https://www.sciencedirect.com/topics/mathematics/hessian-matrix

$$L(f_m) \propto -\frac{1}{2}\sum_{j=1}^{T_m}\frac{G_{jm}^2}{H_{jm}} \tag{3.21}$$

This is the structure score for a tree [58]. The smaller the score is, the better the structure is. This, for each split to make, the proxy gain is defined as

$$Gain = \frac{1}{2}\left[\frac{G_{jmL}^2}{H_{jmL}} + \frac{G_{jmR}^2}{H_{jmR}} - \frac{G_{jm}^2}{H_{jm}}\right] \tag{3.22}$$

$$Gain = \frac{1}{2}\left[\frac{G_{jmL}^2}{H_{jmL}} + \frac{G_{jmR}^2}{H_{jmR}} - \frac{(G_{jmL}+G_{jmR})^2}{H_{jmL}+H_{jmR}}\right] \tag{3.23}$$

all deductions above did not consider regularization.

XGBoost provides a variety of regularization to improve generalization performance. Taking regularization into consideration, it can rewrite the loss function as

$$L(f_m) \propto \sum_{j=1}^{T_m}\left[G_{jm}w_{jm} + \frac{1}{2}H_{jm}w_{jm}^2\right] + \gamma T_m + \frac{1}{2}\lambda\sum_{j=1}^{T_m}w_{jm}^2 + \alpha\sum_{j=1}^{T_m}|w_{jm}| \tag{3.24}$$

$$= \sum_{j=1}^{T_m}\left[G_{jm}w_{jm} + \frac{1}{2}(H_{jm}+\lambda)w_{jm}^2 + \alpha|w_{jm}|\right] + \gamma T_m \tag{3.25}$$

where $\gamma$ is the penalization term on the number of terminal nodes, $\alpha$ and $\lambda$ are for $L_1$ and $L_2$ regularization, respectively. The optimal weight for each region $j$ is calculated as:

$$w_{jm} = \begin{cases} -\frac{G_{jm}+\alpha}{H_{jm}+\lambda} & G_{jm} \leq -\alpha \\ -\frac{G_{jm}-\alpha}{H_{jm}+\lambda} & G_{jm} \geq \alpha \\ 0 & \text{else} \end{cases} \tag{3.26}$$

The gain of each split is defined correspondingly:

$$Gain = \frac{1}{2}\left[\frac{T_\alpha G_{jmL}^2}{H_{jmL}+\lambda} + \frac{T_\alpha G_{jmR}^2}{H_{jmR}+\lambda} - \frac{T_\alpha G_{jm}^2}{H_{jm}+\lambda}\right] - \gamma \tag{3.27}$$

$$T_\alpha(G) = \begin{cases} G+\alpha & G \leq -\alpha \\ G-\alpha & G \geq \alpha \\ 0 & \text{else} \end{cases} \tag{3.28}$$

#### 3.4.5.1 Parameters for XGBOOST

Follows some parameters to fit the model:

$(i)$ n_estimators : It controls the maximum number of iterations.*

$(ii)$ max_depth : Maximum depth of a tree. Increasing this value will make the model more complex and more likely to over-fit. ZERO is only accepted in loss-guided growing policy when tree_method is set as hist or gpu_hist and it indicates no limit on depth.*

$(iii)$ min_child_weight : Minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than min_child_weight, then the building process will give up further partitioning. In linear regression task, this simply corresponds to minimum number of instances needed to be in each node. The larger min_child_weight is, the more conservative the algorithm will be.*

$(iv)$ gamma : Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be.*

$(v)$ subsample : Subsample ratio of the training instances. Setting it to 0.5 means that XGBoost would randomly sample half of the training data prior to growing trees, and this will prevent overfitting. Subsampling will occur once in every boosting iteration.*

$(vi)$ scale_pos_weight : Control the balance of positive and negative weights, useful for unbalanced classes. A typical value to consider: sum(negative instances) / sum(positive instances).*

$(vii)$ seed : Random number seed.*

## 3.5 Metrics

Many expressions translate into analyzing the loss function in linear regression processes. In all cases, the accounting will take place through the predicted values, here called $\hat{y}_i$ and reference values, $y_i$.

### 3.5.1 Maximum Error - ME

The *max_error* function computes the maximum residual error, a metric that captures the worst case error between the predicted value and the true value. In a perfectly fitted single output regression model, *max_error* would be ZERO on the training set and though this

*https://xgboost.readthedocs.io/en/latest/parameter.html#general-parameters would be highly unlikely in the real world, this metric shows the extent of error that the model had when it was fitted [57].

$$ME = max\left(|y_i - \hat{y}_i|\right) \tag{3.29}$$

### 3.5.2 Mean Absolute Error - MAE

The *mean_absolute_error* function computes mean absolute error, a risk metric corresponding to the expected value of the absolute error loss [57].

$$MAE = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} |y_i - \hat{y}_i| \tag{3.30}$$

### 3.5.3 Mean Squared Error - MSE

The *mean_squared_error* function computes mean square error, a risk metric corresponding to the expected value of the squared (quadratic) error or loss [57].

$$MSE = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \left(y_i - \hat{y}_i\right)^2 \tag{3.31}$$

### 3.5.4 $R^2$ score

The $r^2$*_score* function computes the coefficient of determination, usually denoted as $R^2$ [57].

It represents the proportion of variance (of $y$) that has been explained by the independent variables in the model. It provides an indication of goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance [57].

As such variance is data-set dependent, $R^2$ may not be meaningfully comparable across different data-sets. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of $y$, disregarding the input features, would get a $R^2$ score of 0.0 [57].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}_i\right)^2} \tag{3.32}$$

where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{n}\epsilon_i^2$. $r^2$*_score* calculates unadjusted $R^2$ without correcting for bias in sample variance of $y$ [57].

# Chapter 4

# Materials and Methods

> "Since I am neither a neurologist nor a psychiatrist, but a mathematician, the work that follows requires some explanation and justification."
>
> Dr. John von Neumann

## 4.1 Dataset

### 4.1.1 Monte Carlo simulations

The Monte Carlo method is a statistical technique which is capable of simulating a mathematical or physical experiment on a computer. In mathematics, it can provide the expectation value of functions and evaluate integrals; in science and engineering, it is capable of simulating complex problems which are comprised of various random processes with known or assumed probability density functions [59].

Monte Carlo codes offer superior performance compared to analytical algorithms and for that reason it is considered the gold standard with regard to the simulation process according to the literature.

#### 4.1.1.1 Monte Carlo simulation of proton radiographies

Proton radiographies were generated using the Monte Carlo approach. Standard processes included energy loss and straggling, multiple Coulomb scattering, and elastic-inelastic ion interactions.

Using the FLUKA Monte Carlo simulation technique, an ideal single-particle tracking detector was used as single ions trajectories data (list-mode type) theoretically calculated the paths taken by each proton, in each voxel, see Fig. 2.4, for a set of $xCT$ image slices of an head and neck patient. The entire collection of information calculated was stored in a matrix called $sys\_matrix$.

The $sys\_matrix$, represented by $A$, and the ground truth values $RSP$ represented by the uni-dimensional vector $x$, relates as follows

$$A \cdot x = b \tag{4.1}$$

where $A$, is a bi-dimensional matrix $[n, m]$, $n$ is the number of image voxels (embracing 7 slices along the axial dimension of the image sufficient to envelop the proton scattering), where each slice has 314 pixels by 314 pixels and $m$ corresponds to the number of protons present in the forward projection; $b$, in turn, is a uni-dimensional vector and represents the $WET$, Eq. (2.15), obtained from proton radiography simulated, and it has been considered as a measured quantity in this research. The $WET$ matches the integral $RSP$ along the traversed object of interest [12].

Both equations (2.15) and (4.1) yield the WET crossed for each proton. The WET extracted from each equation will differ only by the precision of the $RSP$ calibration curve [60]. Both equations can be used as part of a cost function such as

$$argmin\left(||Ax - b||_2^2\right) \tag{4.2}$$

The problem is solved using a non-negative linear-solver with $\mathbf{x} \geqslant 0$ [60].

This thesis uses an alternative attempt, machine learning, to get to the same results regarding Eq. (4.2), avoiding the intrinsically inaccurate forward-projection model. Also, regarding a constraint set based on the consideration that the calibrated x-ray CT's inaccuracy is in the order of 3-5% [12], this research tries to achieve the performance corresponding to a residual inaccuracy equal or inferior to 1% referring to the absolute difference between the ground truth used in the calibration curve and the corresponding predicted values.

#### 4.1.1.2 Retrieve the accurate calibration curve from Machine Learning

As mentioned in the last paragraph of the 4.1.1.1 section, to obtain solutions for Eq. 4.2, not through analytical strategies, but through data learning, some optimization processes have been applied. Therefore, based on a desirable calibration curve to be achieved and therefore designated as a target, the problem is configured as a supervised learning scenario.

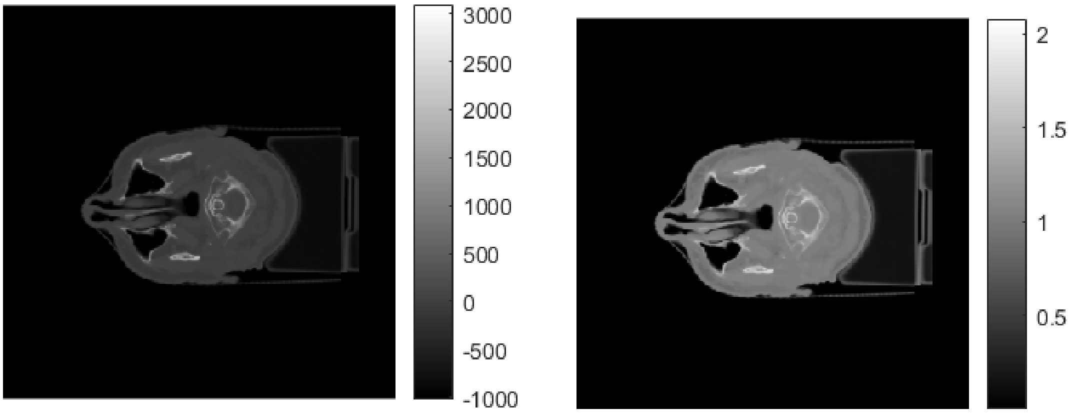
Random Forests, Adaptive Boost, Gradient Descending, and eXtreme Gradient Boosting sets have been used.

## 4.2 Dataset description

### 4.2.1 Reference Images

From the acquisition of a series of $xCT$ images, containing 70 slices, referring to the head and neck region, of which the slices comprised in the range 31 to 70; therefore 39 slices compose the area of interest of this research, acquired under the respective angles of projection [1, 21, 41, 61, 81, 91, 101, 121, 141, 161], have been used as a database.

All pixel values from $xCT$, Fig. 4.1a, through a interpolation approach using the method described in section 2.4.1, Calibration by tissue equivalent, were manipulated and resulting in the object $pCT$ Fig. 4.1b.



(a) From slice 31, $xCT$ express in $HU$.

(b) correspondent image express in $RSP$.

Figure 4.1: Both images showed in (a) $xCT$ and (b) $pCT$ parameters.

### 4.2.2 Pre-processing data

From the HU values, RSP values, the interpolation between them, and the $sys\_matrix$, a series of matrices manipulation was performed in order to obtain a set of input data. Thus, some important functions have been developed and will be presented in the following sections.

#### 4.2.2.1 Histogram calibration function

From the image set $xCT$, where there are many different values of $HU$, it was necessary to structure these values in intervals compatible with those present in Table 2.3. For this, the *Histogram_calibration* function was designed, see algorithm 1.

The *Histogram_calibration* function's main objective is to classify, order, and store all values referring to the $HU$, present in all pixels of the CT images, in 11 different and well-defined intervals of the calibration curve. All pixels were also properly located using indexing features.

Algorithm 1: Histogram Calibration Function

```
Input: Calibration matrix: $C_{11x2}$;
Input: Image matrix;
$index\_structure \leftarrow cell(length(\mathrm{C}), 1)$;
$counts \leftarrow zeros\_values(length(\mathrm{C}))$;
$graylevel \leftarrow image$                    ▷ graylevel receive an image argument
$graylevel \leftarrow graylevel(:)$                 ▷ transform image in a vetor
for $i \leftarrow 1$ to length($C$) do
    if $i = 1$ then
        $high\_bound \leftarrow C[1] + \frac{C[2]}{2}$;
        $auxiliar \leftarrow$ find_by_condition($graylevel < high\_bound$)
    else if $i =$ length($C$) then
        $low\_bound \leftarrow C[end - 1] + \frac{C[end]}{2}$;
        $auxiliar \leftarrow$ find_by_condition($graylevel \geq low\_bound$)
    else
        $lb \leftarrow C[i - 1] + \frac{C[i]}{2}$;
        $hb \leftarrow C[i] + \frac{C[i + 1]}{2}$;
        $auxiliar \leftarrow$
          find_by_condition($graylevel \geq low\_bound \wedge graylevel < high\_bound$);
    end
    $index\_structure \leftarrow auxiliar$;
    $counts[i] \leftarrow$ number of elements from $auxiliar$;
end
```

#### 4.2.2.2 Deviation Calibration function

Once in possession of the *pCT* object and in order to insert a slight disturbance in the *RSP* values, pretending to simulate another image set, in turn *pCT_inaccurate*, a new function was designed, the *Deviation_calibration* function, see algorithm 2. Thus, a relative and random error of up to 5% was applied, referring to each *RSP* value in the first *pCT* image set.

This new *pCT_inaccurate* object, containing a small noise in the *RSP* values and therefore called *RSP_inaccurate*, was also properly organized, classified, and ordered in 11 different intervals. The ordering of the *RSP_inaccurate* values within the 11 intervals previously mentioned is due to maintaining the calibration curve's monotonicity property under construction.

Algorithm 2: Deviate Calibration Function.

```
Input: $CT2RSP$: HU to RSP;
Input: $M$: Deviation Magnification ;
Input: $\vec{i}$: Index_CT2SPR;
while true do
    $C \leftarrow CT2RSP$                                        ▷ $C$ equal CT2RSP
    $C^* \leftarrow CT2RSP$                                      ▷ $C^*$ equal CT2RSP inaccurate
    for $k \leftarrow 1 : length(\vec{i})$ do
        $C^*(i(k)) \leftarrow C(i(k)) + (-M + 2M * \mathsf{rand}(1))$;
        if $\vec{i}(k) \neq 1$ then
            while not $\mathsf{issorted}(C^*)$ do
                $C^*(\vec{i}(k)) \leftarrow C(\vec{i}(k)) \times (1 + (-M + 2M * \mathsf{rand}(1)))$;
            end
        end
    end
    if $\mathsf{issorted}(C^*)$ then
        print: "Calibration curve is monotonic."
    else
        "Calibration curve is NOT monotonic."
    end
end
```

#### 4.2.2.3 Data manipulation

To adapt the parameters used in Eq. 4.2, it was necessary to relate the *pCT_inaccurate*, *sys_matrix* objects and the target values *RSP*. However, *pCT_inaccurate* and *sys_matrix* have different dimensions, so an auxiliary matrix from the *pCT_inaccurate* object was created with same dimensions of *sys_matrix*. This auxiliary matrix is a fraction of the original *pCT_inaccurate* matrix. It has only 7 slices, one central to be analyzed and 3 neighboring slices, in the front and the back, sufficient to record the proton trajectories when scattering.

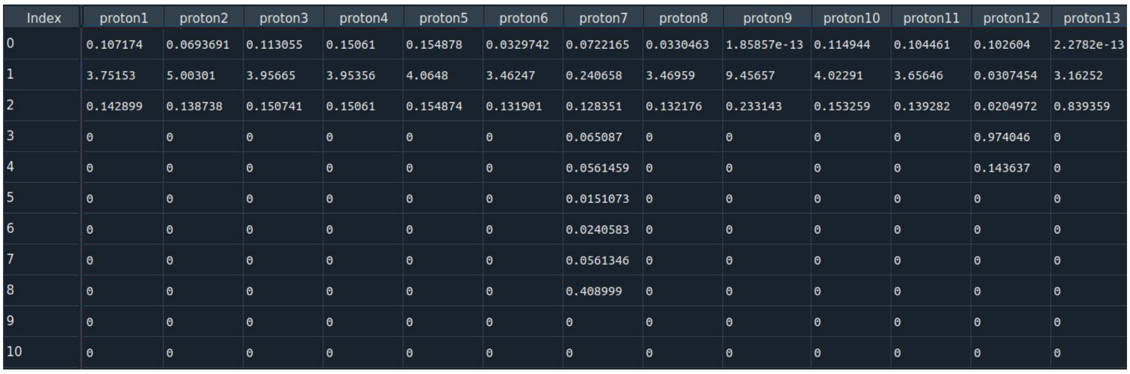
Thus, once the *pCT_inaccurate* values have been associated with the corresponding pixels within the *sys_matrix*, they have been sum out and divided by each WET value corresponding to each proton, see in Algorithm 3. Bearing in mind that each proton can pass through different or even similar pixels with values of *RSP_inaccurate*. Also, it is worth mentioning that each WET value, in turn, is related to the integral path referring to the energy loss of each proton respective, see Eq. 2.16, therefore, new values for recording the paths taken for each proton associated with the *pCT_inaccurate* values have been generated.

Algorithm 3: Input Datset

```
Input: Slices;
Input: Angles;
HCF ←Histogram Calibration function( );
aux ← 7;
WET ← $Sinogram$;                                  ▷ RSP_inaccurate.: RSP with noise
RSP_inacc ←RSP_inaccurate;
iCT_inc ←pCT_inaccurate;
counts ←counts_ict_inaccurate;
input_dataset ← []
for $s$ in Slices do
    for $a$ in Angles do
        plan ← iCT_inc(:, $Slices(s)$ − round($\frac{aux}{2}$) : $Slices(s)$ + round($\frac{aux}{2}$), :));
        [$idx\_iCT\_inc, counts$] ← $HCF$(RSP_inacc, $plan$)     ▷ idx_iCT_inc.: is the index from pixel
        input_data ← zeros($length$(RSP_inacc), $size$(sys_matrix, 2));
        for $i$ ← 1 : $length(RSP\_inacc)$ do
            path_length = sum(sys_matrix(idx_iCT_inc(i)));
            input_data($i$,:) = find_by_division($path\_length, WET$);
        end
        input_dataset ← Concatenate($input\_dataset, input\_data$);
    end
end
```

#### 4.2.2.4 Input dataset

Fig. 4.2 shows a matrix with 11 rows and $n$ columns. The number of lines refers to the different types of the material medium through which protons travel. Each value represents the sum of $WET$ for all voxels of a given type of material medium, as shown in the algorithm 3. Each column represents a proton. The total number of columns, depending on which or how many projections were used. Therefore, input data is a matrix of dimension $[11, n]$, and all data are continuous values.



| Index | proton1 | proton2 | proton3 | proton4 | proton5 | proton6 | proton7 | proton8 | proton9 | proton10 | proton11 | proton12 | proton13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.107174 | 0.0693691 | 0.113055 | 0.15061 | 0.154878 | 0.0329742 | 0.0722165 | 0.0330463 | 1.85857e-13 | 0.114944 | 0.104461 | 0.102604 | 2.2782e-13 |
| 1 | 3.75153 | 5.00301 | 3.95665 | 3.95356 | 4.0648 | 3.46247 | 0.240658 | 3.46959 | 9.45657 | 4.02291 | 3.65646 | 0.0307454 | 3.16252 |
| 2 | 0.142899 | 0.138738 | 0.150741 | 0.15061 | 0.154874 | 0.131901 | 0.128351 | 0.132176 | 0.233143 | 0.153259 | 0.139282 | 0.0204972 | 0.839359 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0.065087 | 0 | 0 | 0 | 0 | 0.974046 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0561459 | 0 | 0 | 0 | 0 | 0.143637 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0151073 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0240583 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0561346 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0.408999 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 4.2: Input dataset. The lines represent the sum of $WET$ for all voxels of a given type of material medium. Each column represents a proton. In the figure, the number of columns is limited due to a large number of protons.

#### 4.2.2.5 Target

The final output to be predicted is the $RSP$ values. These values are present in a vector with 11 values (Table. 2.3) also called the ground truth values.

## 4.3 Model

From Sklearn website[57], the following methods were implemented: eXtreme [61], Gradient Boost [62], Adaptive Boost[55] and Random Forest [63]

### 4.3.1 Model adjustment

#### 4.3.1.1 Use of seed to generate random numbers

Seed determines the sequence of pseudo-random numbers that will be generated. Random function in various programming languages is actually a pseudo-random function. The process of generating (pseudo) random numbers has two phases. 1. Setting seed 2. Generating random numbers, for instance $x_0, x_1, x_2, x_3, \cdots$. Therefore, a standard sequence of numbers was imposed on all models used in this research.

#### 4.3.1.2 Parameters choice

From the GridSearchCV [64] function, which refers to a hyperparameter adjustment process to determine the ideal values for a given model, since the performance of a model significantly depends on the value of the hyperparameters, a stress parameter procedure was performed by selecting the best parameters from a range of values, according to the table 4.1.

| Parameters | RF | ADABOOST | GBR | XGBOOST |
|---|---|---|---|---|
| n_estimators | [300, 400, 500] | [300, 400, 500] | [300, 400, 500] | 300 |
| max_depth | [1, 2, 4] | – | [1, 2, 4] | – |
| learning_rate | – | [0.001, 0.01, 0.1] | [0.001, 0.01, 0.1] | 0.01 |
| min_samples_split | 1 | – | – | – |
| min_samples_leaf | [1, 2, 4] | – | [1, 2, 4] | – |
| random_state | 0 | 0 | 0 | – |
| loss | – | linear | least square | – |
| min_child_weight | – | – | – | 1 |
| gama | – | – | – | 0 |
| subsample | – | – | – | 0.8 |
| scale_pos_weight | – | – | – | 1 |
| n_jobs | – | – | – | 4 |
| seed | – | – | – | 0 |

Table 4.1: Parameter ranges to define which parameters will be used to adjust each method.

Due to the size of the input data set, this task can imply a lot of computational execution time. But in general, this practice works very well. For this research, it should be considered that for each slice or combination of them, associated with the choice of one or more projections, it can represent numerous possibilities of models, thus, we restrict the values of the hyperparameters used in this thesis to the values of the table 4.2, which are reasonable for hyperparameters. The exception was for the XGBOOST method, for which the hyperparameters have been defined based on the previous choice of the other methods.

| Parameters | RF | ADABOOST | GBR | XGBOOST |
|---|---|---|---|---|
| n_estimators | 300 | 300 | 300 | 300 |
| max_depth | 4 | – | 4 | 4 |
| learning_rate | – | 0.01 | 0.01 | 0.01 |
| min_samples_split | 3 | – | – | – |
| min_samples_leaf | 2 | – | 2 | – |
| random_state | 0 | 0 | 0 | – |
| loss | – | linear | least square | – |
| min_child_weight | – | – | – | 1 |
| gama | – | – | – | 0 |
| subsample | – | – | – | 0.8 |
| scale_pos_weight | – | – | – | 1 |
| n_jobs | – | – | – | 4 |
| seed | – | – | – | 0 |

Table 4.2: Parameters used to define each method

## 4.4 Training and testing groups

A straightforward method for evaluating a machine learning algorithm's performance is to use different sets of training and test data. An original dataset is obtained and divided into two parts. Depending on several factors, such as sample size, dataset specifications, among other aspects, the division is made according to the modeling's intent. The first part is used to fit model and the second one to make predictions. To split the input dataset, the *train_test_split*[*] function from the sklean library has been used.

### 4.4.1 Correspondence between Training and testing groups

As a first approach, the training and testing groups were not subdivided. In this step, the *train_test_split* function was not applied. Therefore, in this way, the attempt is certainly exposed to the problem of overfitting and should be taken with caution.

In this specific item, to obtain the two corresponding training and test groups, taking care to clarify that they are not coincident, noise values were added only in the testing group. For the added noise, the function *random.random_sample*[†] was used from the Numpy[†] library. Thus, the noise has been generated and represents a continuous uniform set of values comprised in a characteristic interval for each observation in the testing group, according to $Unif[a,b), b > a$, thus $(b-a) \cdot random\_sample(dataset\ input) + a$, where $a$ and $b$ are equal to the noise value.

[*] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

[†] https://numpy.org/doc/stable/reference/random/generated/numpy.random.random_sample.html

To assess the correspondence between the model's response to its fitting, a noise of 0% (and it's worth noticing that by adding 0% noise the training and test data are equivalents), 1%, and 2% was added to the training group. However, the prediction was made by matching the test group with the original training group without the noise added previously.

### 4.4.2 Alternating proton columns

For all ML methods, a 50% split was performed on the input data, with no change in the proton columns' order, to obtain alternation in the proton columns when structuring the training and test groups.

### 4.4.3 Random training and testing subsets

In order to split the input dataset into random train and test subsets, the *train_test_split*[4] function from the sklearn library has been used. Thus, the following steps, see algorithm 4, have been implemented to adapt the input dataset.

Algorithm 4: Training and Testing Splitting

```
ytrain, ytest ←RSP_original, RSP_original;
test ←test_percentage;
if test is None then
    xtrain ← input_dataset;
    xtest ← input_dataset;
else
    input_dataset ← Transpose(input_dataset);
    aux ← Zeros(length(input_dataset));
    xtrain, xtest, _, _ ← Train_Test_Split(input_dataset, aux, test_size= test);
    xtrain ← Transpose(input_dataset);
    xtest ← Transpose(xtest);
    saux ← Shape[1](xtest);
    xtest ← Resize(xtest, length(xtrain));
    xtest[:, saux :]← Fill_Value(xtest);
end
```

In this approach, the testing group assumed a proportion of 5, 10, 25, and 50% concerning the training group. For the sake of matching the sizes between the training and test groups, the latter was filled with columns containing ZERO values.

## 4.5 Neighboring slices analysis

### 4.5.1 From a central slice to neighboring slices

From a central slice taken as input dataset, slices located in its neighborhood were applied as test groups. Due to the combination of projections and slices, which presents different amounts for numbers of protons, causing a size difference in both groups, a strategy has been applied to circumvent this situation. Thus, ZERO values or NAN values have been inserted in the group with the fewer proton columns.

### 4.5.2 From the neighborhood to the missing central slice

The similar analysis described in the section 4.5.1, however, this time reversing the roles, from slices in the neighborhood, taken as an input dataset, a certain missing central slice was used as a test group. Again, the situation of differences between the sizes of both groups is present. Thus, ZERO values or NAN values were inserted in the group with fewer proton columns.

# Chapter 5

# Results and Discussion

> ”There are two possible outcomes: if the result confirms the hypothesis, then you’ve made a measurement. If the result is contrary to the hypothesis, then you’ve made a discovery”.
>
> Dr. Enrico Fermi

In the graphs below, the nominal values of ground truth are shown on the x-axis. The curves do not have a mathematical relationship between the axes.

## 5.1 Analysis of correspondence between training and testing data.

Fig. 5.1, Fig. 5.2, and Fig. 5.3, there is the analysis of correspondence between the training and test groups. The analysis is relevant since a dataset has been used as a basis for both groups’ composition. In this specific research, the models have been tested based on datasets derived from training data. Thus, it is important to highlight that the input dataset’s structuring probably provides over-fitting effects. Hence, the test group presents a slightly subtle change pattern. In this way, the insertion of a slight noise value in the test group reasonably well represents the condition described. In the table 5.1, the values of $MSE$ are shown for all methods as a function of the percentage of noise inserted in each model fitting. Except for Random Forest and $GBR$, the other methods showed high consistency in the training and test groups’ correspondence.

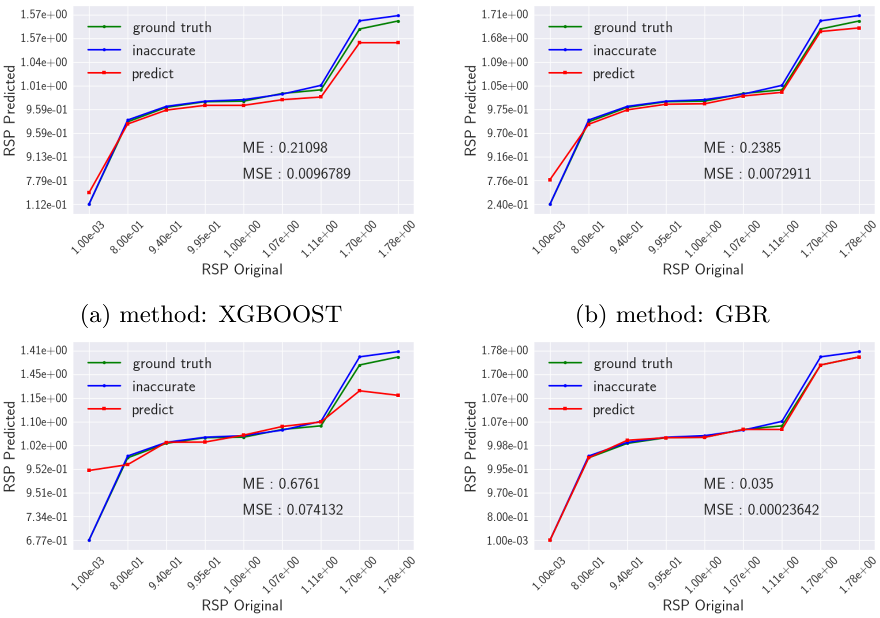
| Metric | RF | ADABOOST | GBR | XGBOOST | Noise (%) |
|---|---|---|---|---|---|
| MSE | 0.07413247 | 0.00023642 | 0.0072911 | 0.0096789 | 0 |
| MSE | 0.07843091 | 0.00025741 | 0.0104724 | 0.0096789 | 1 |
| MSE | 0.07843985 | 0.00027239 | 0.0105104 | 0.0096789 | 2 |

Table 5.1: *MSE* values for different methods as a function of the percentage of noise inserted in the test group only to analyze the training and test groups' correspondence.

| Metric | RF | ADABOOST | GBR | XGBOOST | Noise(%) |
|---|---|---|---|---|---|
| ME | 0.6761 | 0.0350 | 0.2385 | 0.2109 | 0 |
| ME | 0.6988 | 0.0350 | 0.2914 | 0.2109 | 1 |
| ME | 0.6988 | 0.0350 | 0.2914 | 0.2109 | 2 |

Table 5.2: *ME* values for different methods as a function of the percentage of noise inserted in the test group only to analyze the training and test groups' correspondence.

### 5.1.1 A random noise of 0% was inserted in the training group



(a) method: XGBOOST

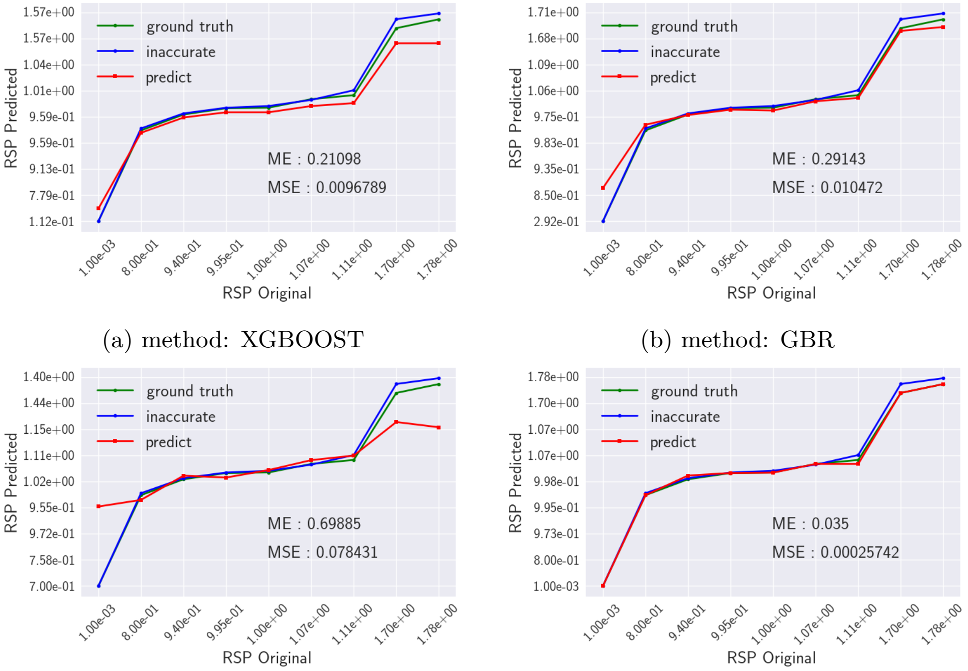
(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.1: Prediction performed with slice 45 and projection 91 to the noise of 0%, corresponds to the equivalence between training and testing datasets. Green line: ground truth, blue line: inaccurate, and red line: prediction.

### 5.1.2 A random noise of up to 1% was inserted in the training group



(a) method: XGBOOST

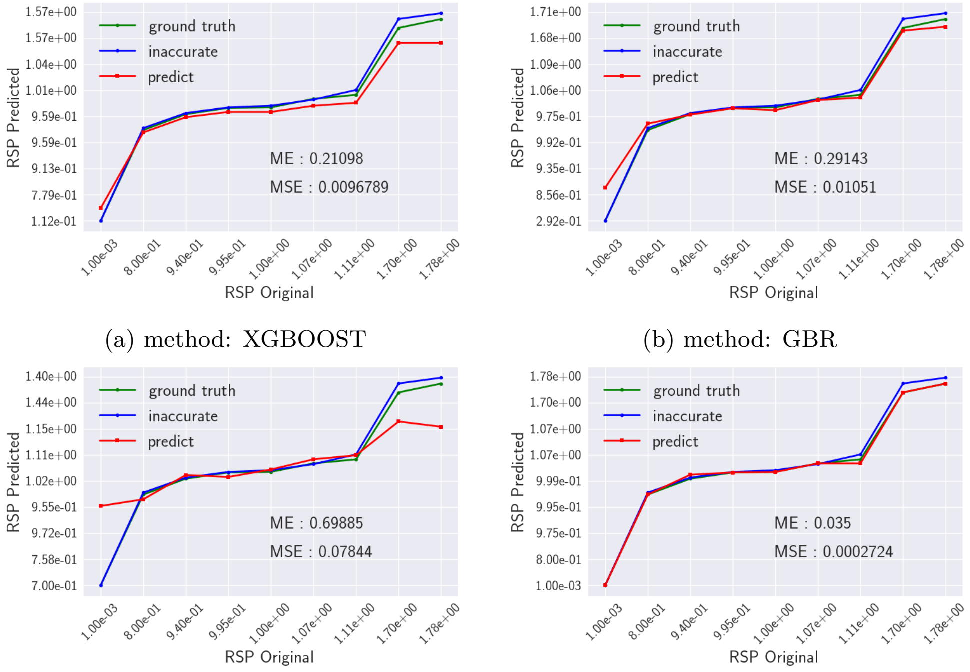
(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.2: Prediction performed with slice 45 and projection 91 to the noise of 1%, corresponds to the equivalence between training and testing datasets. Green line: ground truth, blue line: inaccurate, and red line: prediction.

### 5.1.3 A random noise of up to 2% was inserted in the training group



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.3: Prediction performed with slice 45 and projection 91 to the noise of 2%, corresponds to the equivalence between training and testing datasets. Green line: ground truth, blue line: inaccurate, and red line: prediction.

It is worth mentioning the performance of the ADABOOST method as seen in figures 5.1, 5.2, and 5.3, item (d). This method combines the predictions from short one level decision trees, decision stumps, although other algorithms can also be used. Thus, the decision stump algorithm is used to find many weak models and correct their predictions by making use of them. Exploring the peculiarities of the intrinsic processes present in this specific method is interesting to expand its potential, see more results in Appendix A.

## 5.2 Analysis of the proton radiographies with respect to the calibration curve points.

The calibration curve was originally built with 11 *RSP* values. These values, in turn, are related to the trajectories of individual protons along their paths. Each point of the calibration curve can be therefore differently represented, depending on the proton paths in the radiography. These pieces of information have been properly classified and make up the input dataset, as seen previously, in section 4.2.2.4.

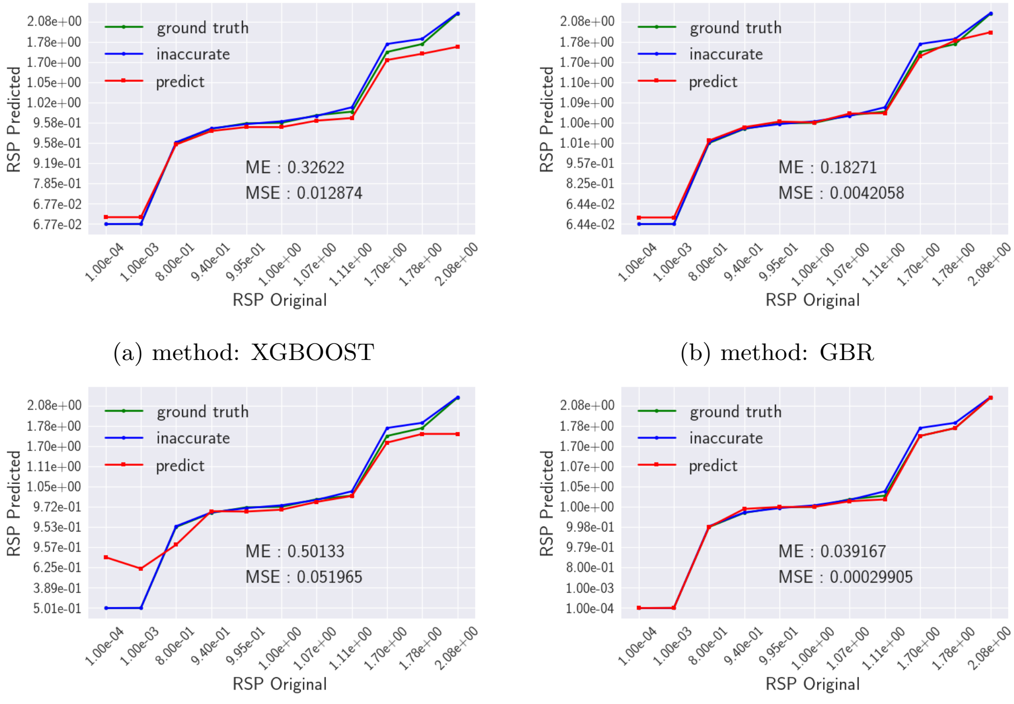
In table 5.3, some information regarding the input dataset is shown. The Points column refers to the classification of the distinctive intervals that correspond to 11 $RSP$ values, in ascending order, which translates into a monotonic behavior of the calibration curve. The column of Voxel counts shows all voxels' accounting, from am image under analysis, which content corresponds to the record of the path taken by any protons. The Zero count column refers to the counting of the number of voxels that, in turn, do not record any information about the proton trajectory. Finally, in the last two columns, the statistical indicators of mean and variance are presented, referring to the distance traveled by each proton in each voxel traveled.

Two important aspects stand out: a variance value present in the proton trajectory records in the region with a low $RSP$ value and the high record of null proton trajectory values for the region with the highest $RSP$ value. The very high mean and variance values concerning its neighbors influence the models' adjustment and directly affect the predicted $RSP$ values. On the other hand, the high records of null values for proton trajectories (ZERO values to compose the models' adjustment) represent intrinsic limitation to perform a prediction adequately. In Fig. 5.4, except for the $ADABOOST$ method, the described effects are clearly observed.

The study showed that under these conditions of the input dataset, the two extreme points of the calibration curve represent two situations that can hinder a model's development. On the one hand, the excess of information without relevance, see Table 5.3, data for point 1; on the other hand, the excess of null values, see table 5.3, data for point 11. Therefore, in order to avoid such influence on the fitting model, it was chosen to work with only 9 $RSP$ values, thus excluding the curves' two ends. Fig. 5.5 follows a direct application of this condition. As can be seen in Table 5.5, the exclusion of the extremes of the calibration curve significantly reduced the MSE and the MAE for all methods.

| Points | Voxel counts | Zero counts | mean $[mm]$ | variance $[mm^2]$ |
|---|---|---|---|---|
| 1 | 522789 | 1 | 0.184667 | 0.098563 |
| 2 | 41274 | 139 | 0.915489 | 3.198606 |
| 3 | 18526 | 145 | 0.208456 | 0.079450 |
| 4 | 12956 | 26729 | 0.093301 | 0.016467 |
| 5 | 11620 | 24549 | 0.82659 | 0.006170 |
| 6 | 35028 | 26542 | 0.194093 | 0.042830 |
| 7 | 16930 | 31580 | 0.093902 | 0.007171 |
| 8 | 23982 | 36911 | 0.140895 | 0.022673 |
| 9 | 6973 | 52257 | 0.043725 | 0.003823 |
| 10 | 94 | 112587 | 0.000364 | 0.000006 |
| 11 | 0 | 115934 | 0.0 | 0.0 |

Table 5.3: Occurrence values have been calculated from all radiographies and projection.



(a) method: XGBOOST

(b) method: GBR

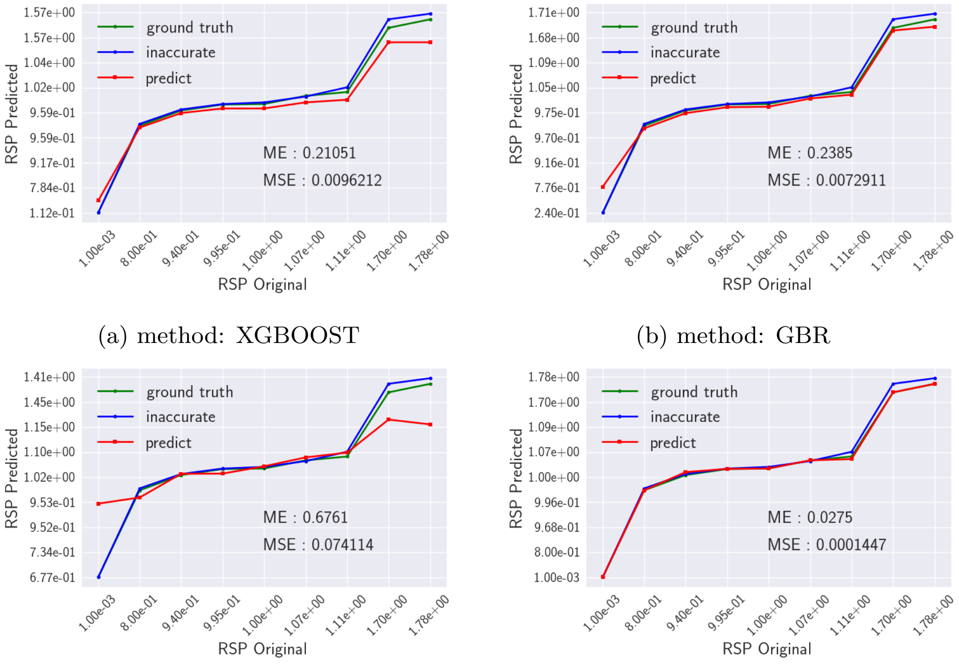
(c) method: RF

(d) method: ADABOOST

Figure 5.4: Slice 45, Projection 91 - Plots taking into account 11 RSP values. ME = Maximum Error, MSE = Mean Square Error. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.32621 | 0.20817 | 0.31034 | 0.96701 |
| GBR | 0.18146 | 0.29856 | 0.43005 | 0.98972 |
| RF | 0.55465 | 0.35467 | 0.27745 | 0.84750 |
| ADABOOST | 0.05000 | 0.19803 | 0.00250 | 0.99891 |

Table 5.4: Table referring to Fig. 5.4. Metrics related to each method.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.5: Slice 45, Projection 91 - Plots taking into account 9 RSP values. ME = Maximum Error, MSE = Mean Square Error. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23850 | 0.00729 | 0.05300 | 0.96929 |
| RF | 0.67609 | 0.07411 | 0.16690 | 0.68785 |
| ADABOOST | 0.02750 | 0.00014 | 0.00578 | 0.99939 |

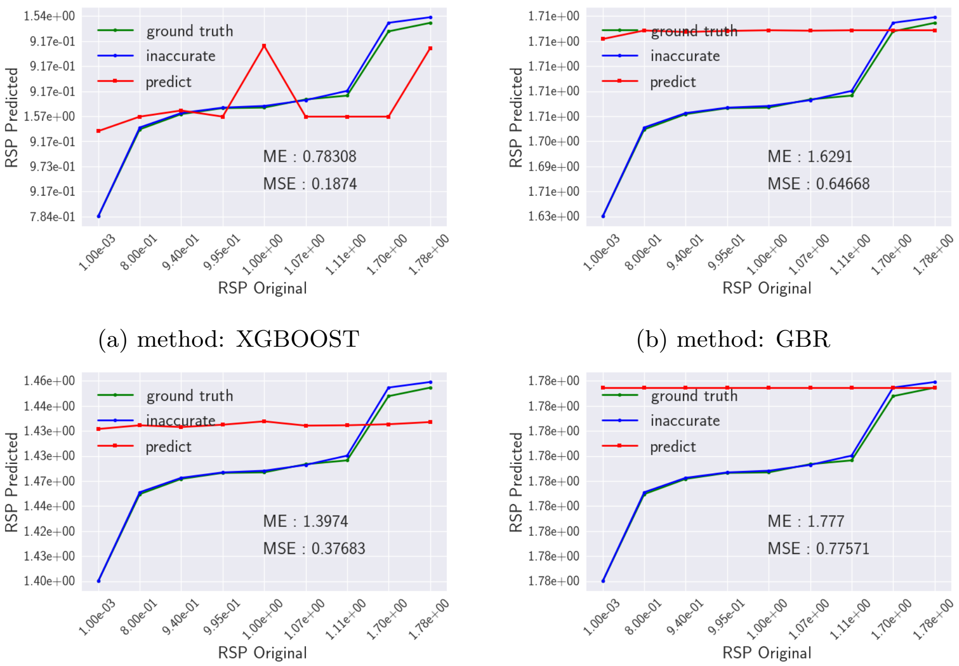
Table 5.5: Table referring to Fig. 5.5. Metrics related to each method.

### 5.2.1 Random splitting of the training and testing datasets

Figs. 5.6, 5.7, 5.8 and 5.9 show the results of applying the *train_test_split* function to the data set. Each test group was defined in the respective simulations by a value of 5, 10, 25, and 50% concerning the columns of the input data set; see section 4.4.3. With the values of ground truth as a target, this approach generates a different way of organizing data for training and testing. Regardless of the percentage value applied to the test set, none of the options performed well in this configuration. On the other hand, it must be considered that the construction of the data used as an input set comes from a situation that is strongly dependent on the ordering between protons and their trajectory records. Therefore, changing the proton columns' order translates into a significant loss in the performance of the methods used.

On the other hand, when replacing the proton columns with ZERO values by NAN values in the test group, it has been observed that only the *XGBOOST* method was able to deal with the replacement, see Fig. 5.10. The other methods were unable to perform the simulations due to errors in their execution since their algorithms are unable to perform calculations with NAN values.

#### 5.2.1.1 Testing group filling by 5% from the training group columns.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

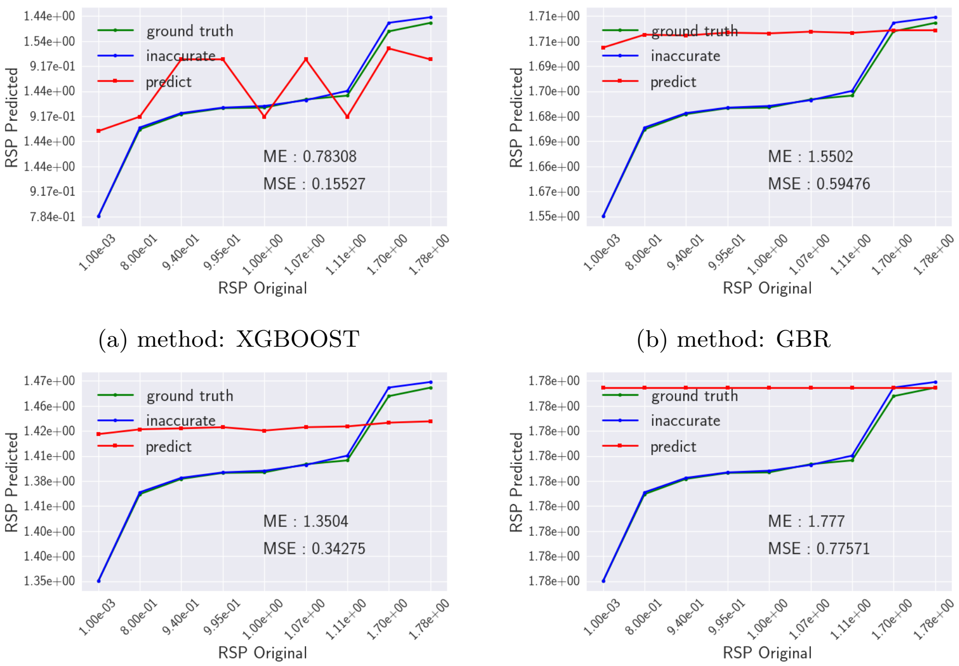
(d) method: ADABOOST

Figure 5.6: Slice 35 Projections 1. Testing group filling by 5% from the training group columns and by 95% with ZERO values columns. ME = Maximum Error, MSE = Mean Square Error. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.86107 | 0.26357 | 0.43846 | -0.11006 |
| GBR | 1.62908 | 0.64667 | 0.66906 | -1.72356 |
| RF | 1.39741 | 0.37682 | 0.51875 | -0.58706 |
| ADABOOST | 1.77700 | 0.77570 | 0.73366 | -2.26697 |

Table 5.6: Table referring to Fig. 5.6. Metrics related to each method.

#### 5.2.1.2 Testing group filling by 10% from the training group columns.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

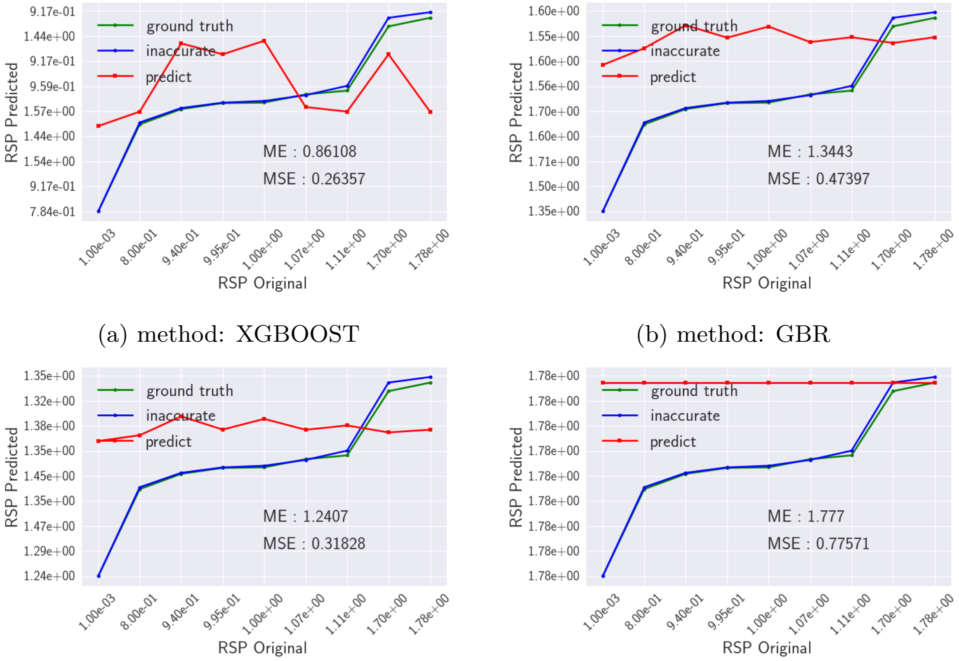
(d) method: ADABOOST

Figure 5.7: Slice 35 Projections 1. Testing group filling by 10% from the training group columns and by 90% with ZERO values columns. ME = Maximum Error, MSE = Mean Square Error. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.78307 | 0.15526 | 0.33191 | 0.34606 |
| GBR | 1.550170 | 0.59475 | 0.64392 | -1.50488 |
| RF | 1.35040 | 0.34275 | 0.49091 | -0.44353 |
| ADABOOST | 1.77700 | 0.77570 | 0.73366 | -2.26697 |

Table 5.7: Table referring to Fig. 5.7. Metrics related to each method.

### 5.2.1.3 Testing group filling by 25% from the training group columns.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

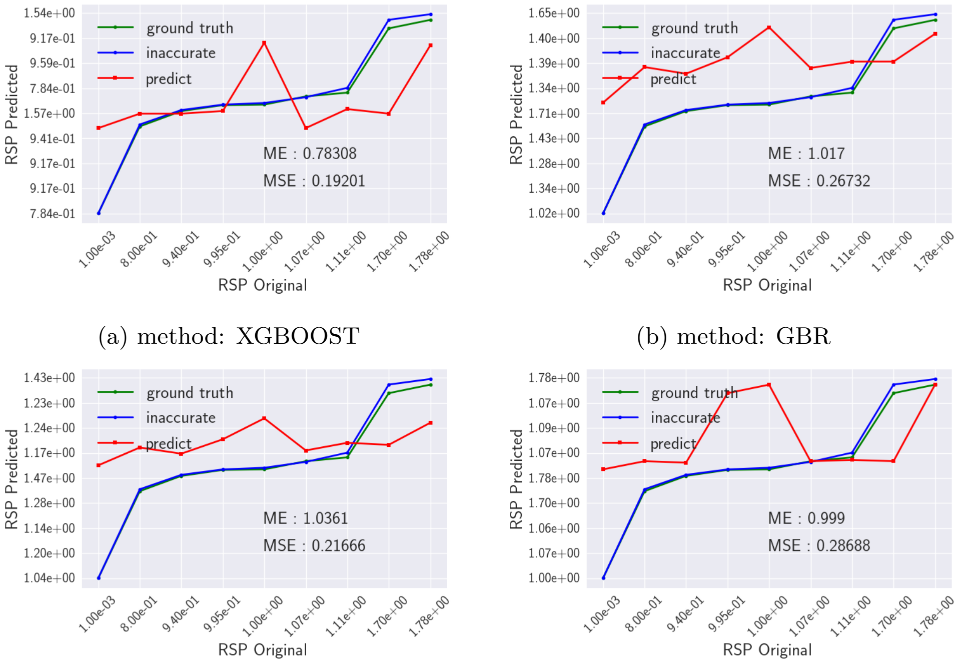
(d) method: ADABOOST

Figure 5.8: Slice 35 Projections 1. Testing group filling by 25% from the training group columns and by 75% with ZERO values columns. ME = Maximum Error, MSE = Mean Square Error. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-squared |
|---|---|---|---|---|
| XGBOOST | 0.86107 | 0.26357 | 0.43846 | -0.11006 |
| GBR | 1.34433 | 0.47396 | 0.60170 | -0.99617 |
| RF | 1.24069 | 0.31827 | 0.49080 | -0.34046 |
| ADABOOST | 1.77700 | 0.77570 | 0.73366 | -2.26697 |

Table 5.8: Table referring to Fig. 5.8. Metrics related to each method.

#### 5.2.1.4 Testing group filling by 50% from the training group columns.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

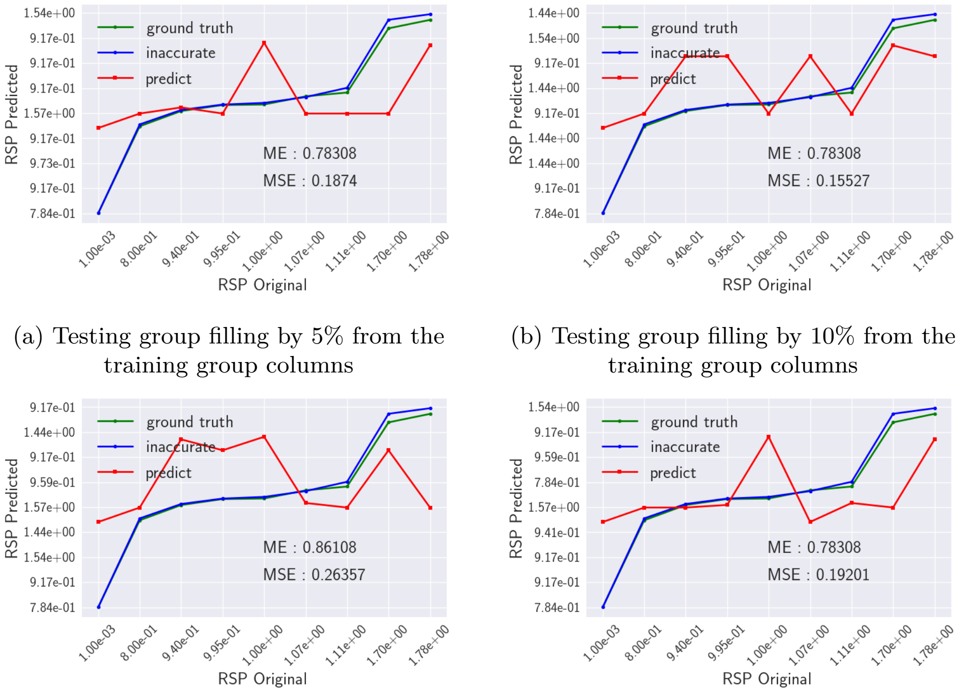
(d) method: ADABOOST

Figure 5.9: Slice 35 Projections 1. Testing group filling by 50% from the training group columns and by 50% with ZERO values columns. ME = Maximum Error, MSE = Mean Square Error. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-squared |
|---|---|---|---|---|
| XGBOOST | 0.78307 | 0.19200 | 0.33380 | 0.19134 |
| GBR | 1.01704 | 0.26732 | 0.44829 | -0.1258 |
| RF | 1.03612 | 0.21665 | 0.38309 | 0.08751 |
| ADABOOST | 0.99900 | 0.28687 | 0.39188 | -0.20822 |

Table 5.9: Table referring to Fig. 5.8. Metrics related to each method.

#### 5.2.1.5 Test group filling with NAN values columns.



(a) Testing group filling by 5% from the training group columns

(b) Testing group filling by 10% from the training group columns

(c) Testing group filling by 25% from the training group columns

(d) Testing group filling by 50% from the training group columns

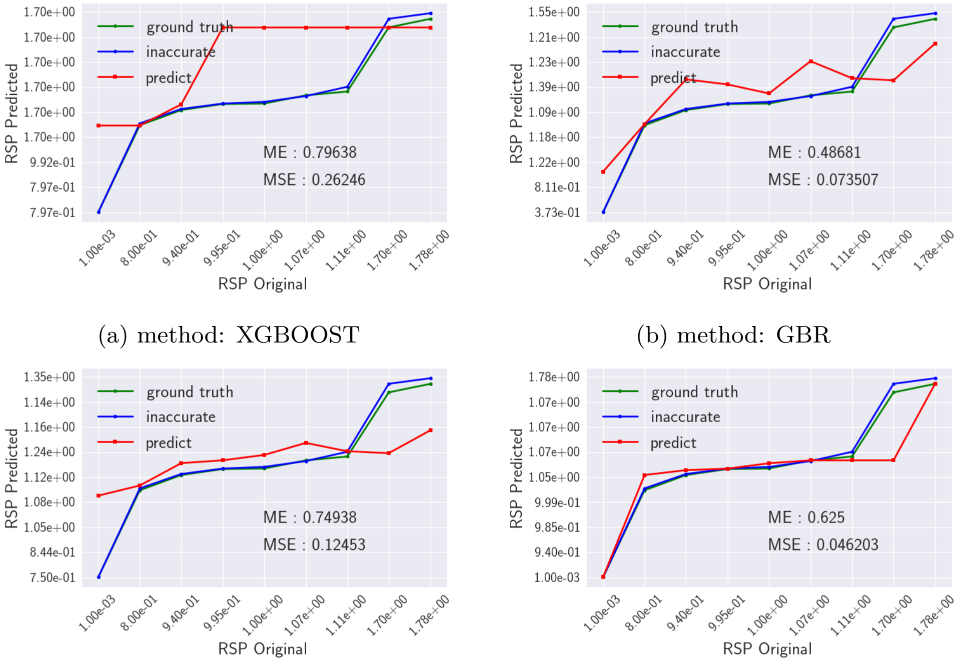
Figure 5.10: Method *XGBOOST*, Slice 31 Projections 1. Items (a)-(d), the test group was filled with 5, 10, 25, and 50% by columns from the training group. To complete those testing groups, they have been filling with 95, 90, 75, and 50% by *NAN* values columns, respectively. ME = Maximum Error, MSE = Mean Square Error. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| (%) Test group from the training group | ME | MSE | MAE | R-squared |
|---|---|---|---|---|
| 5 | 0.78307 | 0.18739 | 0.32748 | 0.21704 |
| 10 | 0.78307 | 0.15526 | 0.33191 | 0.34606 |
| 25 | 0.86107 | 0.26357 | 0.43846 | -0.11006 |
| 50 | 0.78307 | 0.19200 | 0.33380 | 0.19134 |

Table 5.10: Table referring to Fig. 5.10. Metrics related to each method.

### 5.2.2 Fitting model and testing through alternating column

Once again, the results showed unsatisfactorily. This reinforces the fact that the input dataset is strongly dependent on the proton columns' ordering. A new approach has been used, still concerning the input dataset's division into training and testing subgroups. This time, alternating columns, according to the order of odd columns for those used to adjust the model and even for its test. As for the alternation of proton columns, their original positions have been filled by columns with ZERO values due to their absence in each group, see Fig. 5.11.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.11: Slice 50, Projection 81. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.79638 | 0.26246 | 0.39405 | -0.1054 |
| GBR | 0.48681 | 0.07350 | 0.23203 | 0.69041 |
| RF | 0.74938 | 0.12452 | 0.25587 | 0.47553 |
| ADABOOST | 0.62500 | 0.04620 | 0.09967 | 0.80541 |

Table 5.11: Table referring to Fig. 5.11. Metrics related to each method.

## 5.3 Impact from different slices and projections

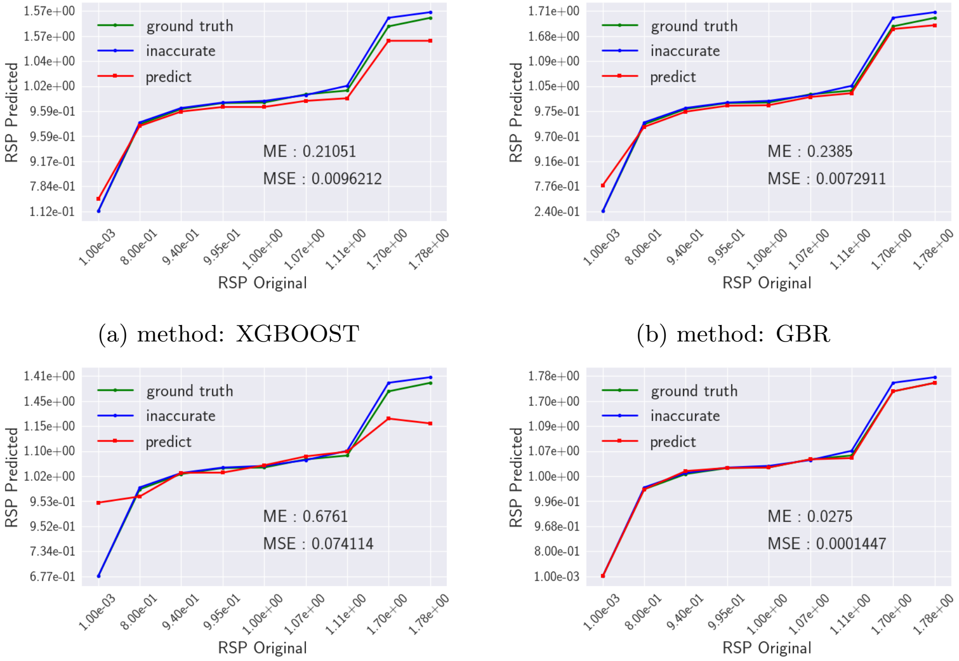
A series of prediction are presented in the following items, carried out from combinations between slices and projections.

Regarding the values of hyperparameters, these contain the data that control the model's training process. There are other combinations of possible hyperparameters that were not used in this work, and that can generate potentially interesting results.

### 5.3.1 Single slice and different Projections

#### 5.3.1.1 One slice and one projection

In Fig. 5.12, a prediction for a single slice and projection is shown.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.12: Slice 45, Projection 91. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23850 | 0.00729 | 0.05300 | 0.96929 |
| RF | 0.67609 | 0.07411 | 0.16690 | 0.68785 |
| ADABOOST | 0.02750 | 0.00014 | 0.00578 | 0.99939 |

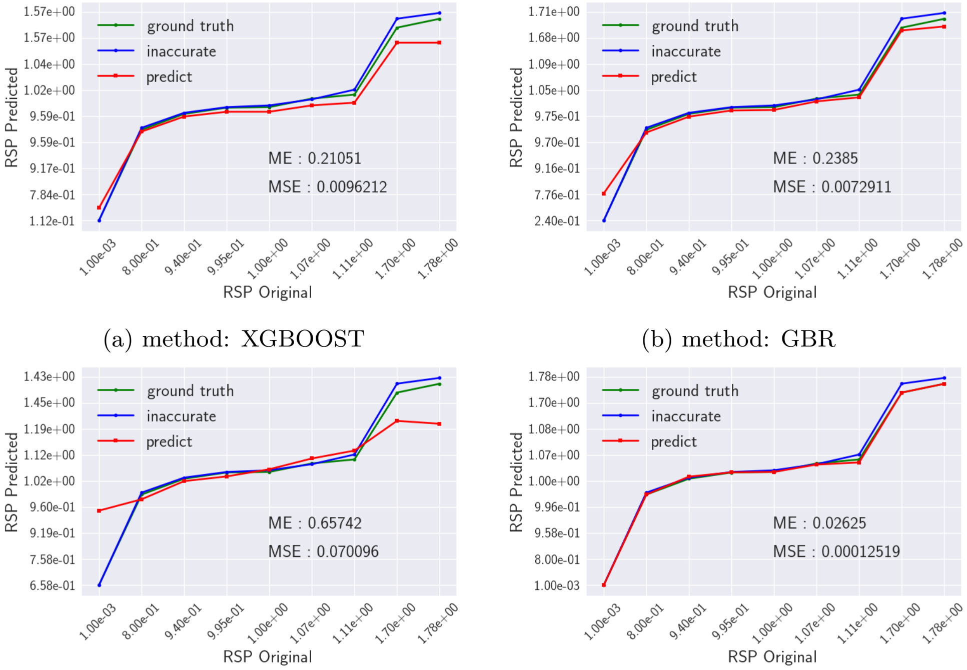
Table 5.12: Slice 45, Projection 91. Data referring to Fig. 5.12. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 2.92 | 0.12 | 0 | 0 | 2.10 | 0 | 0 |

Table 5.13: Slice 45, Projection 91. Individual error percentage values between prediction and ground truth referring to Fig. 5.12d, ADABOOST method. Maximum value 2.92%.

### 5.3.1.2 One slice and two Projections

In Fig. 5.13, a prediction for a single slice and two projections is shown.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.13: Slice 45, Projection 21, 141. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23850 | 0.00729 | 0.05300 | 0.96929 |
| RF | 0.65742 | 0.07009 | 0.16694 | 0.70478 |
| ADABOOST | 0.02625 | 0.00012 | 0.00620 | 0.99947 |

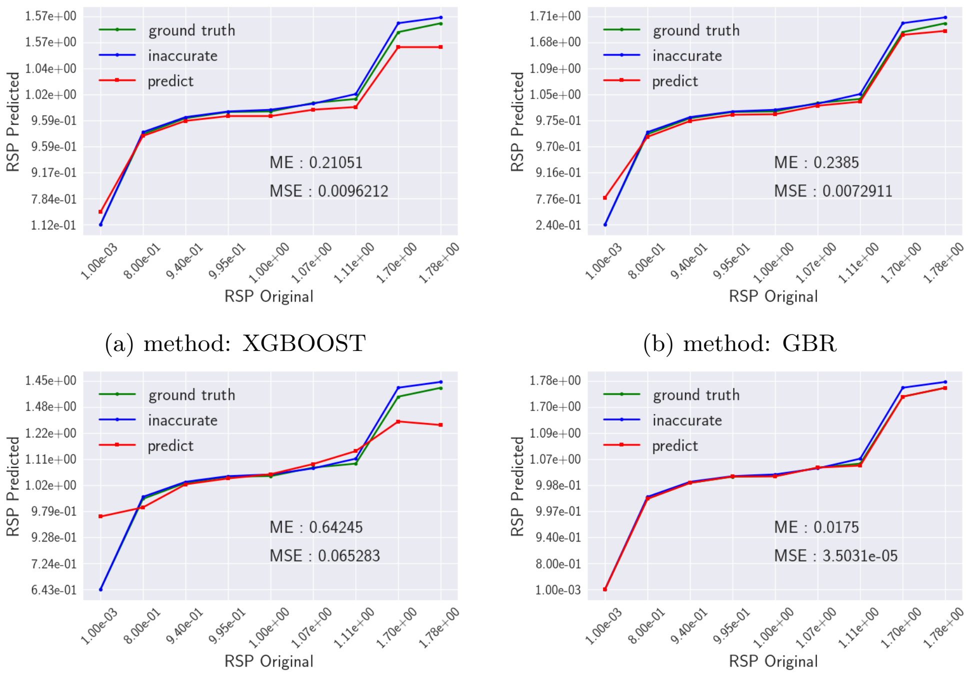
Table 5.14: Slice 45, Projection 21, 141. Data referring to Fig. 5.13. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 1.95 | 0.12 | 0 | 0.93 | 2.36 | 0 | 0 |

Table 5.15: Slice 45, Projection 21, 141. Individual error percentage values between prediction and ground truth referring to Fig. 5.13d, ADABOOST method. Maximum value 2.36%.

### 5.3.1.3 One slice and three Projections

In Fig. 5.14, a prediction for a single slice and three projections is shown.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.14: Slice 45, Projection 1, 91, 161. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23850 | 0.00729 | 0.05300 | 0.96929 |
| RF | 0.64244 | 0.06528 | 0.16121 | 0.72505 |
| ADABOOST | 0.01750 | 0.00000 | 0.00240 | 0.99985 |

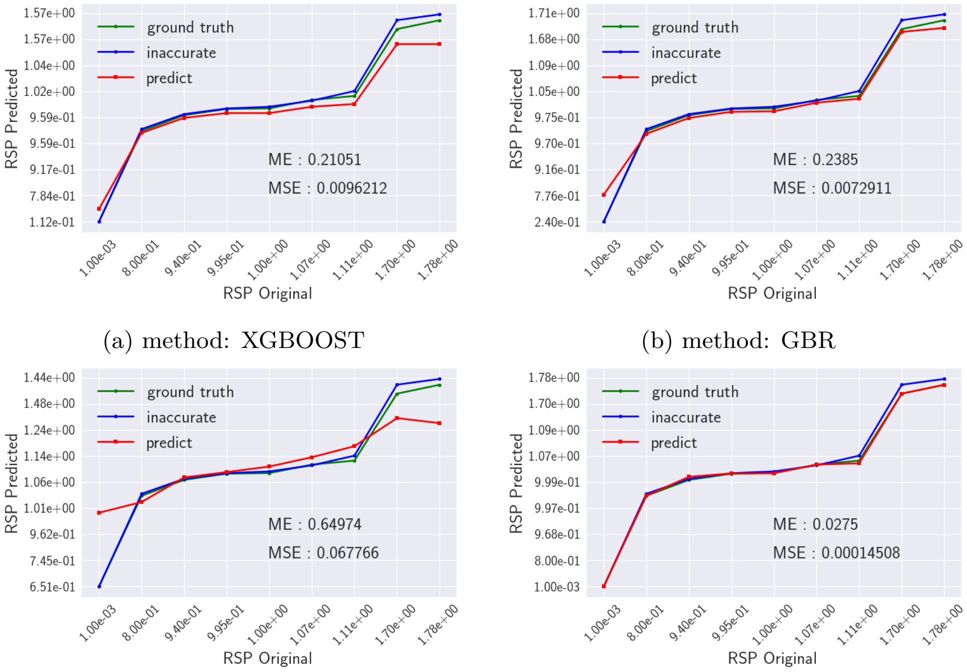
Table 5.16: Slice 45, Projection 1, 91, 161. Data referring to Fig. 5.14. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 0 | 0.16 | 0.25 | 0 | 1.57 | 0 | 0 |

Table 5.17: Slice 45, Projection 1, 91, 161. Individual error percentage values between prediction and ground truth referring to Fig. 5.14d, ADABOOST method. Maximum value 1.57%.

#### 5.3.1.4 One slice and all projections

In Fig. 5.15, a prediction for a single slice and all projections is shown.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.15: Slice 45, all projections. Error. Green line: ground truth, blue line: inaccurate, and red line: prediction.

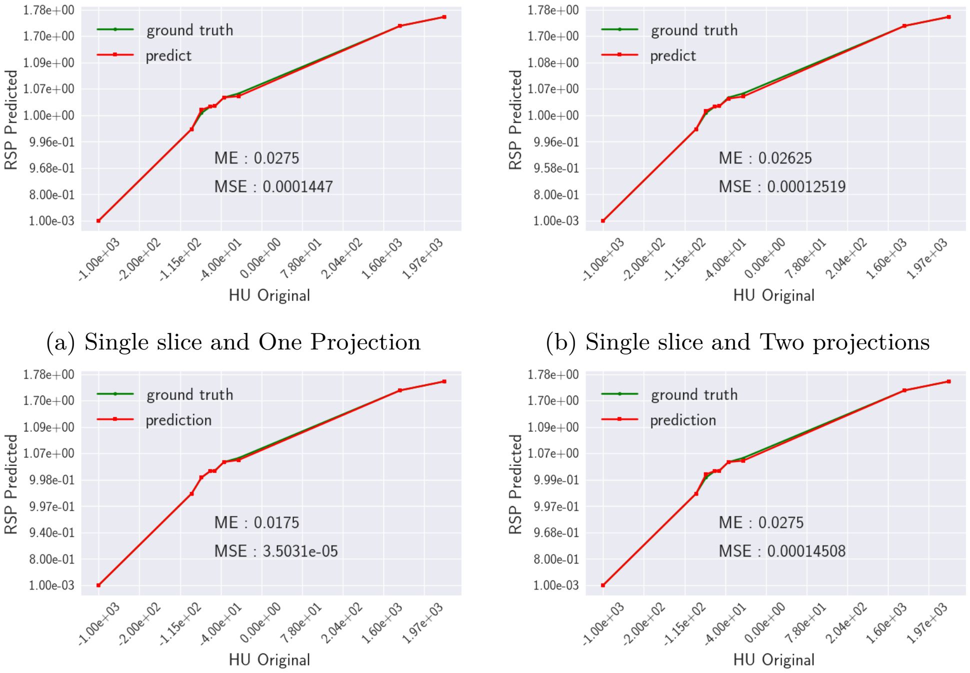
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23850 | 0.00729 | 0.05300 | 0.96929 |
| RF | 0.64973 | 0.06776 | 0.17145 | 0.71459 |
| ADABOOST | 0.02750 | 0.00014 | 0.00598 | 0.99938 |

Table 5.18: Slice 45, all projections. Data referring to Fig. 5.15. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 2.92 | 0.20 | 0.10 | 0 | 2.10 | 0 | 0 |

Table 5.19: Slice 45, all projections. Data referring to 5.15d, ADABOOST method. Maximum value 2.92%.

#### 5.3.1.5 Calibration Curve - Single Slice and different projections



(a) Single slice and One Projection

(b) Single slice and Two projections

(c) slice and Three projections

(d) Single slice and All projections

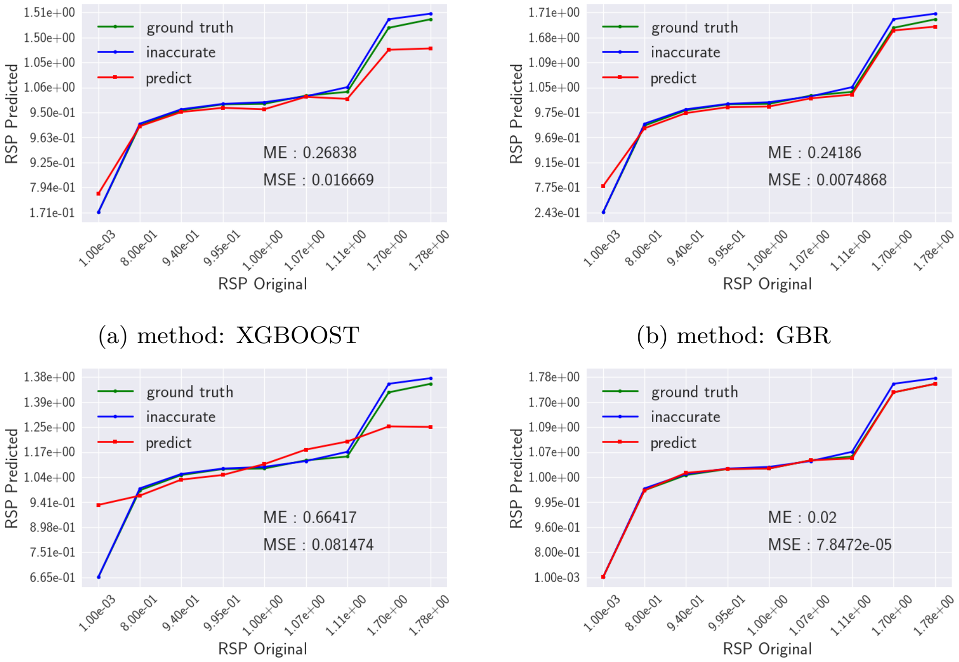
Figure 5.16: Calibration curves for a single slice and different projections concerning to 5.12d, 5.13d, 5.14d, and 5.15d respectively. The ADABOOST method only. Green line: ground truth, red line: predicted.

In Fig. ??, all the graphics refer to the *ADABOOST* method; the other methods performed poorly. Following graphics 5.16 (a)-(d), the individual maximum percentage amount between the predicted values and the ground truth are, respectively, 2.92 %, 2.36 %, 1.57 %, 2.92 %. It is important to point out that from the perspective of a specific patient and a specific proton model, the model adjustment has a very high R-square value, clearly indicating overfitting behavior. This scenario clearly falls in situations where one wishes to study an event that has already happened and thus describe it in order to obtain a previously known result. See more results Appendix B.

### 5.3.2 Different slices and one projection

#### 5.3.2.1 Two slices and one projection

In Fig. 5.17, a prediction for two slices and one projection is shown.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.17: Slice 55, 56 and Projection 101. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26837 | 0.01666 | 0.09106 | 0.92979 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.66417 | 0.08147 | 0.19952 | 0.65686 |
| ADABOOST | 0.02000 | 0.00008 | 0.00416 | 0.99966 |

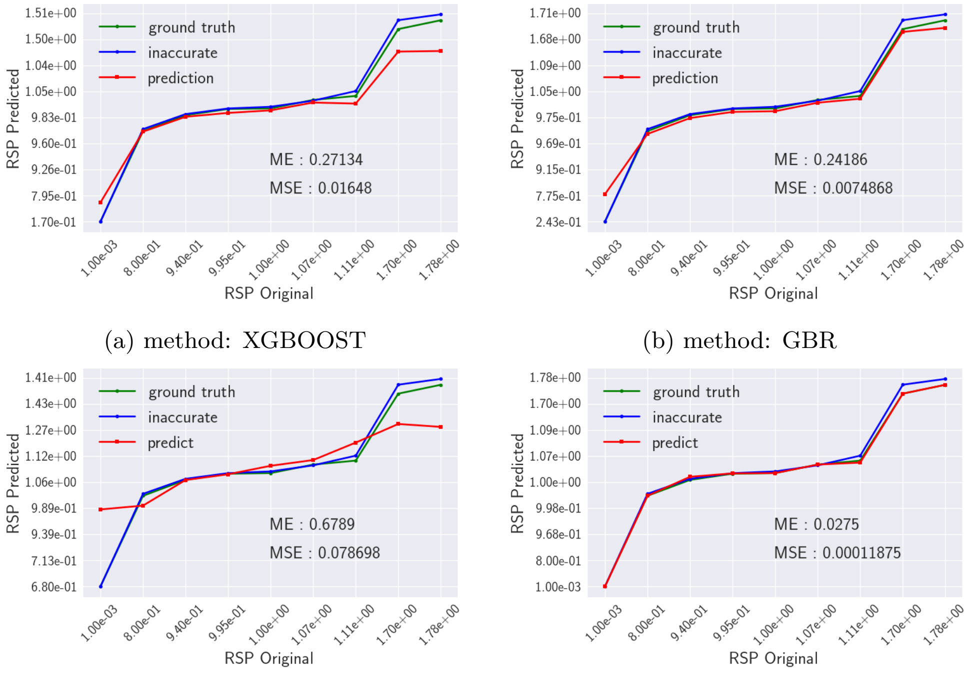
Table 5.20: Table referring to Fig. 5.17. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 2.12 | 0 | 0 | 0 | 1.57 | 0 | 0 |

Table 5.21: Slices 55, 56. Projection 101. Individual error percentage values between prediction and ground truth referring to Fig. 5.17d, ADABOOST method. Maximum value 2.12%.

#### 5.3.2.2 Three slices and one projection

In Fig. 5.18, a prediction for two slices and two projections is shown.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.18: Slice 54, 55, and 56. Projection 101. Green line: ground truth, blue line: inaccurate, and red line: prediction.

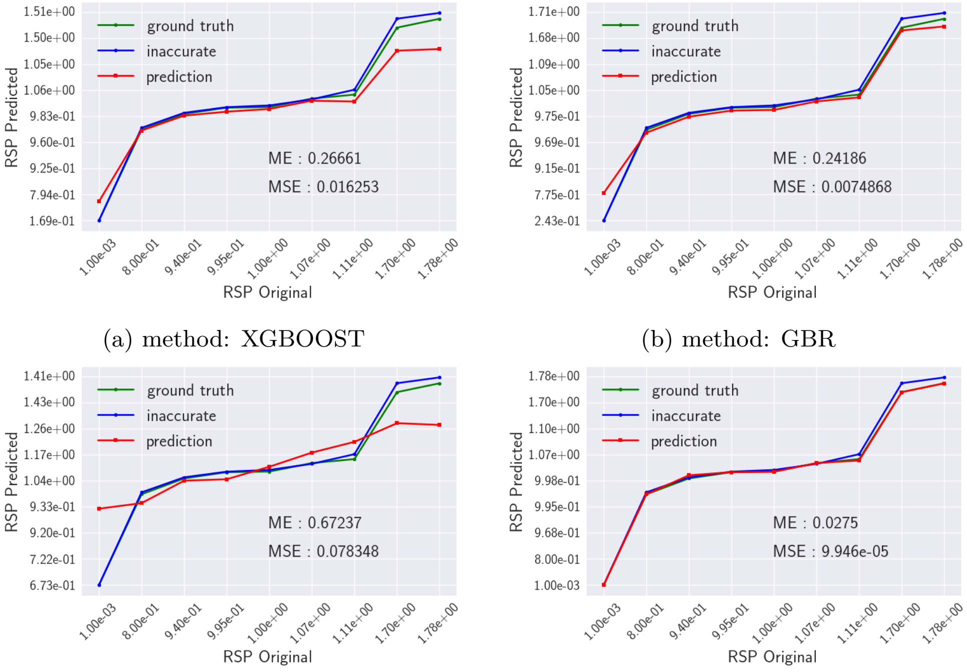
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.27133 | 0.01647 | 0.08889 | 0.93059 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.67889 | 0.07869 | 0.18590 | 0.66855 |
| ADABOOST | 0.02750 | 0.00011 | 0.00527 | 0.99949 |

Table 5.22: Slice 54, 55, and 56. Projection 101. Data referring to Fig. 5.18. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 2.92 | 0.25 | 0 | 0 | 1.57 | 0 | 0 |

Table 5.23: Slices 54, 55, and 56. Projection 101. Individual error percentage values between prediction and ground truth referring to Fig. 5.18d, ADABOOST method. Maximum value 2.92%.

#### 5.3.2.3 Four slices and one projection



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.19: Slices 53, 54, 55, and 56. Projection 101. Green line: ground truth, blue line: inaccurate, and red line: prediction.

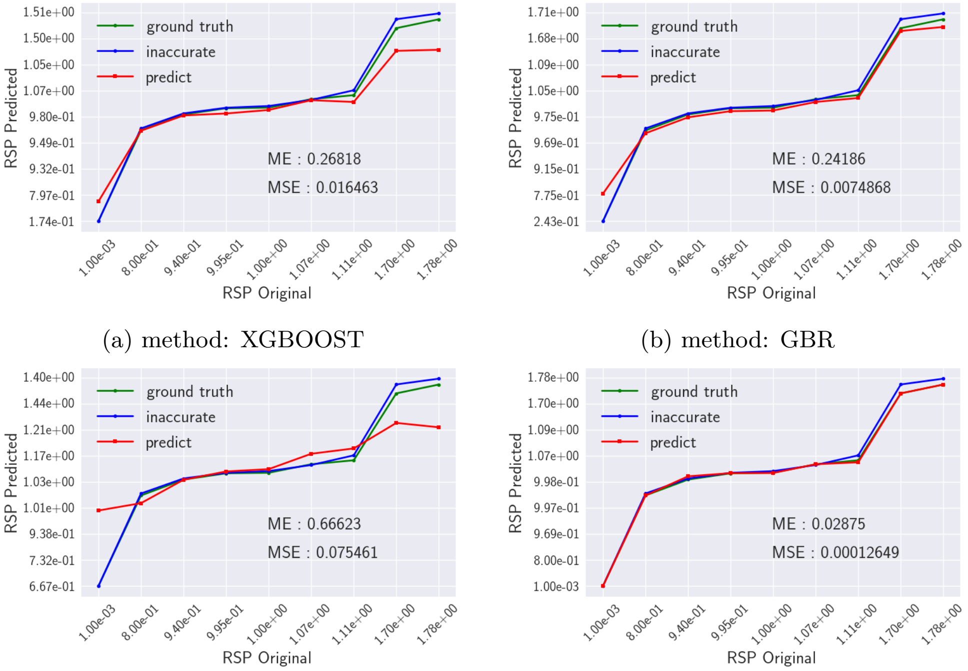
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26661 | 0.01625 | 0.08779 | 0.93154 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.67236 | 0.07834 | 0.19553 | 0.67002 |
| ADABOOST | 0.02750 | 0.00001 | 0.00453 | 0.99958 |

Table 5.24: Slices 53, 54, 55, and 56. Projection 101. Data referring to Fig. 5.19. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 2.92 | 0 | 0.16 | 0 | 1.05 | 0 | 0 |

Table 5.25: Slices 53, 54, 55, and 56. Projection 101. Individual error percentage values between prediction and ground truth referring to Fig. 5.19d, ADABOOST method. Maximum value 2.92%.

#### 5.3.2.4 Five slices and one projection



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.20: Slices 52, 53, 54, 55, and 56. Projection 101. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26818 | 0.01646 | 0.08754 | 0.93066 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.66623 | 0.07546 | 0.17985 | 0.68218 |
| ADABOOST | 0.02875 | 0.00012 | 0.00550 | 0.99946 |

Table 5.26: Slices 52, 53, 54, 55, and 56. Projection 101. Data referring to Fig. 5.20. Metrics related to each method.

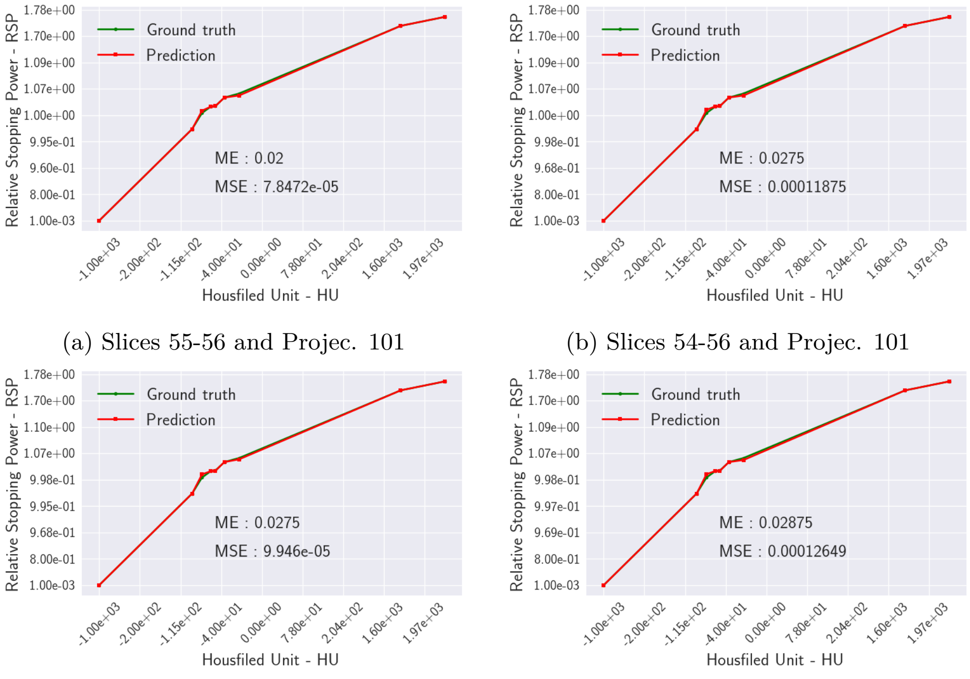
| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 3.05 | 0.17 | 0.16 | 0 | 1.57 | 0 | 0 |

Table 5.27: Slices 52, 53, 54, 55, and 56. Projection 101. Individual error percentage values between prediction and ground truth referring to Fig. 5.20d, ADABOOST method. Maximum value 3.05%.

#### 5.3.2.5 Calibration Curve - Multiple slices and one projection

In Fig. 5.21, once again all the graphics refer to the ADABOOST method; the other methods performed poorly. Following items Fig. 5.21 (a)-(d), the maximum percentage

values between the predictioned values and the ground truth were, respectively, 2.12 %, 2.92 %, 2.92 %, 3.05 %.



(a) Slices 55-56 and Projec. 101

(b) Slices 54-56 and Projec. 101

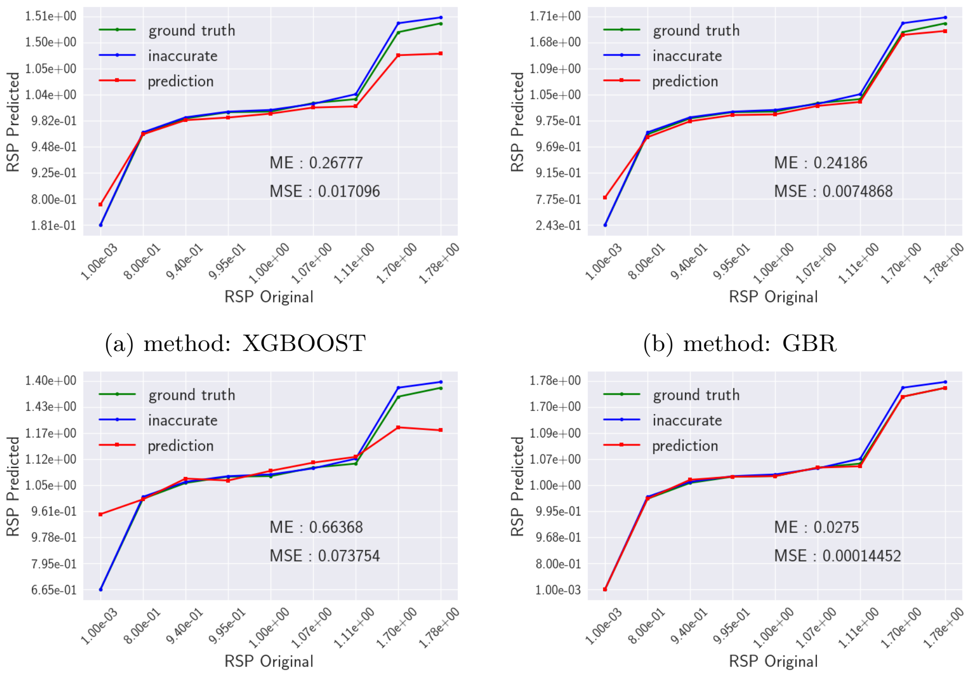
(c) Slices 53-56 and Projec. 101

(d) Slices 52-56 and Projec. 101

Figure 5.21: Calibration curves for a single slice and different projections concerning to 5.17d, 5.18d, 5.19d, and 5.20d respectively. The ADABOOST method only. Green line: ground truth, red line: predicted.

### 5.3.3 Different slices and two projections

#### 5.3.3.1 One slice and two projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.22: Slices 49. Projections 1, 91. Green line: ground truth, blue line: inaccurate, and red line: prediction.

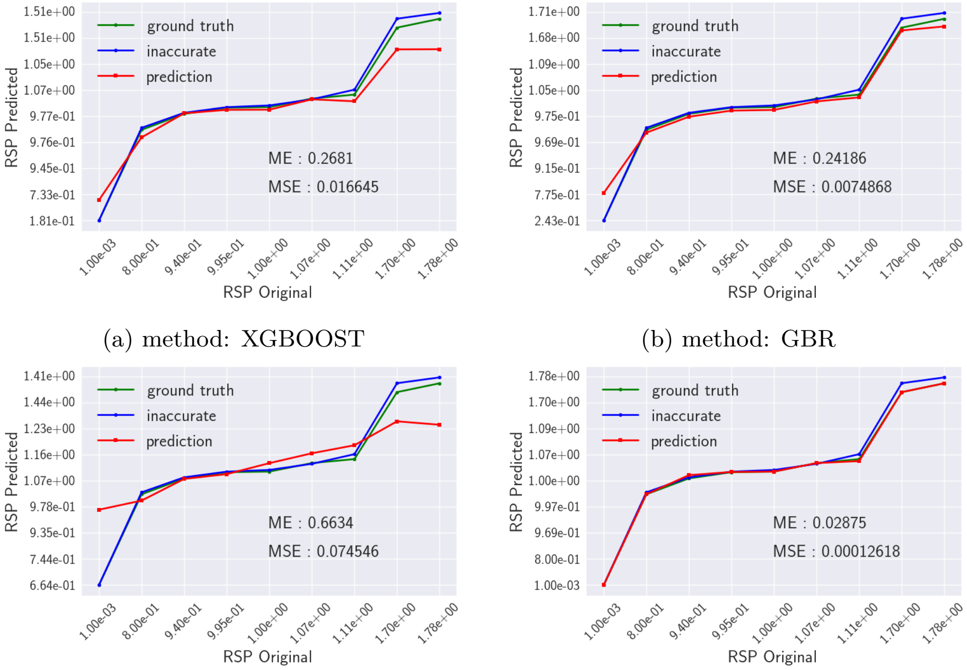
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26818 | 0.01646 | 0.08754 | 0.93066 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.66367 | 0.07375 | 0.17051 | 0.68937 |
| ADABOOST | 0.02750 | 0.00014 | 0.00564 | 0.99939 |

Table 5.28: Slices 49. Projections 1, 91. Data referring to Fig. 5.22. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 2.92 | 0 | 0 | 0 | 2.10 | 0 | 0 |

Table 5.29: Slice 49. Projections 1, 91. Individual error percentage values between prediction and ground truth referring to Fig. 5.22d, ADABOOST method. Maximum value 2.92%.

### 5.3.3.2 Two slices and two projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.23: Slices 48, 49. Projections 1, 91. Green line: ground truth, blue line: inaccurate, and red line: prediction.

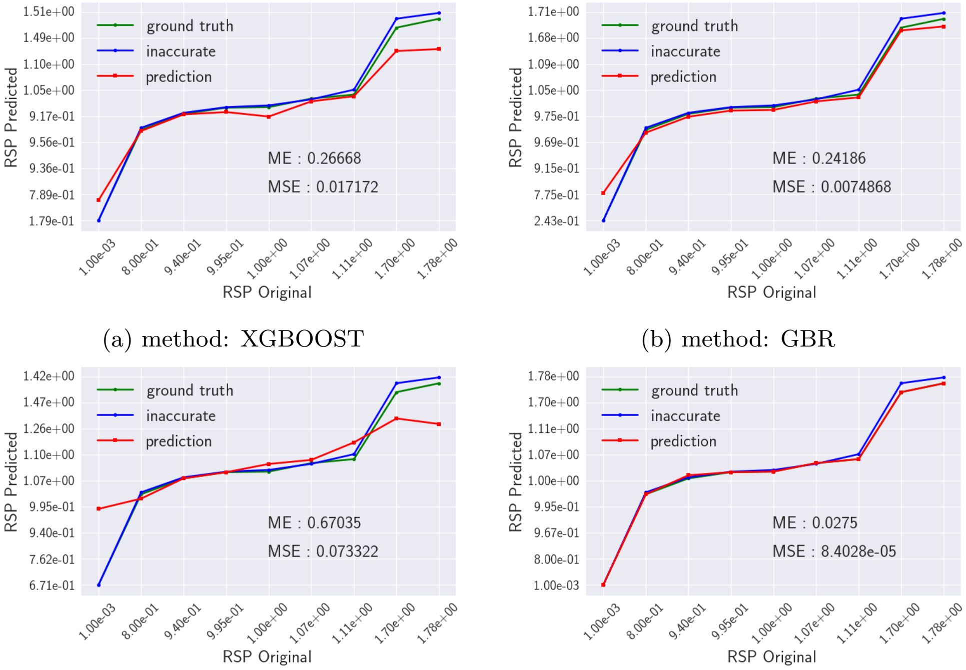
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26810 | 0.01664 | 0.09095 | 0.92989 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.66339 | 0.07454 | 0.18285 | 0.68604 |
| ADABOOST | 0.02750 | 0.00012 | 0.00532 | 0.99946 |

Table 5.30: Slices 48, 49. Projections 1, 91. Data referring to Fig. 5.22. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 3.05 | 0.16 | 0 | 0 | 1.57 | 0 | 0 |

Table 5.31: Slices 48, 49. Projections 1, 91. Individual error percentage values between prediction and ground truth referring to Fig. 5.23d, ADABOOST method. Maximum value 3.05%.

### 5.3.3.3 Three slices and two projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.24: Slices 47, 48, 49. Projections 1, 91. Green line: ground truth, blue line: inaccurate, and red line: prediction.

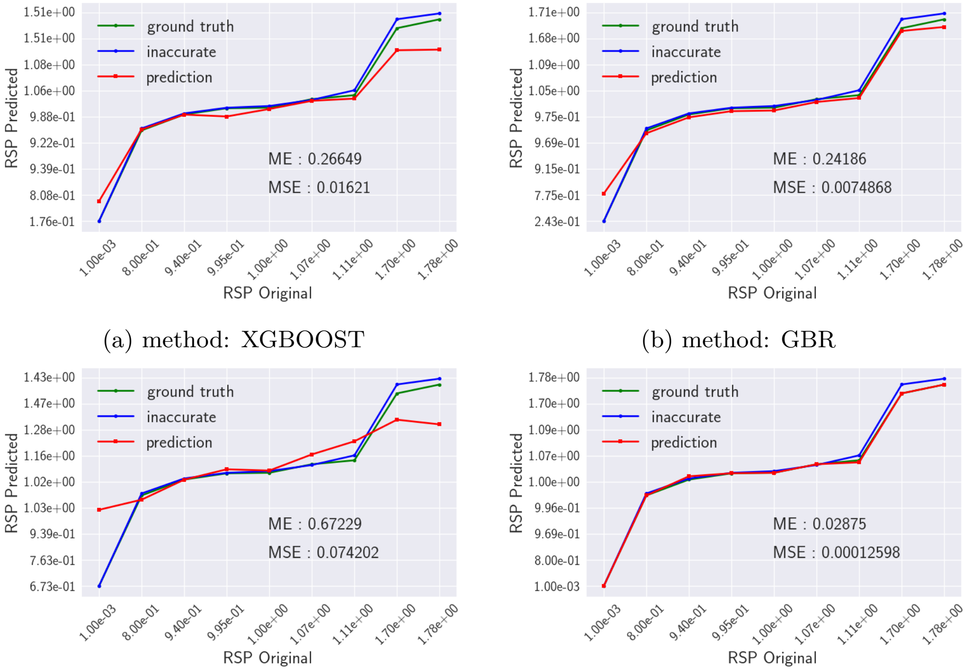
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26667 | 0.01717 | 0.09182 | 0.92767 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.67035 | 0.07332 | 0.17131 | 0.69119 |
| ADABOOST | 0.02749 | 0.00001 | 0.00305 | 0.99964 |

Table 5.32: Slices 47, 48, 49. Projections 1, 91. Data referring to Fig. 5.29. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 2.92 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 5.33: Slices 47, 48, 49. Projections 1, 91. Individual error percentage values between prediction and ground truth referring to Fig. 5.29d, ADABOOST method. Maximum value 2.92%.

#### 5.3.3.4 Four slices and two projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.25: Slices 46, 47, 48, 49. Projections 1, 91. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26648 | 0.01621 | 0.08590 | 0.93172 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.67229 | 0.07420 | 0.17816 | 0.68748 |
| ADABOOST | 0.02875 | 0.00012 | 0.00525 | 0.99946 |

Table 5.34: Slices 46, 47, 48, 49. Projections 1, 91. Data referring to Fig. 5.30. Metrics related to each method.

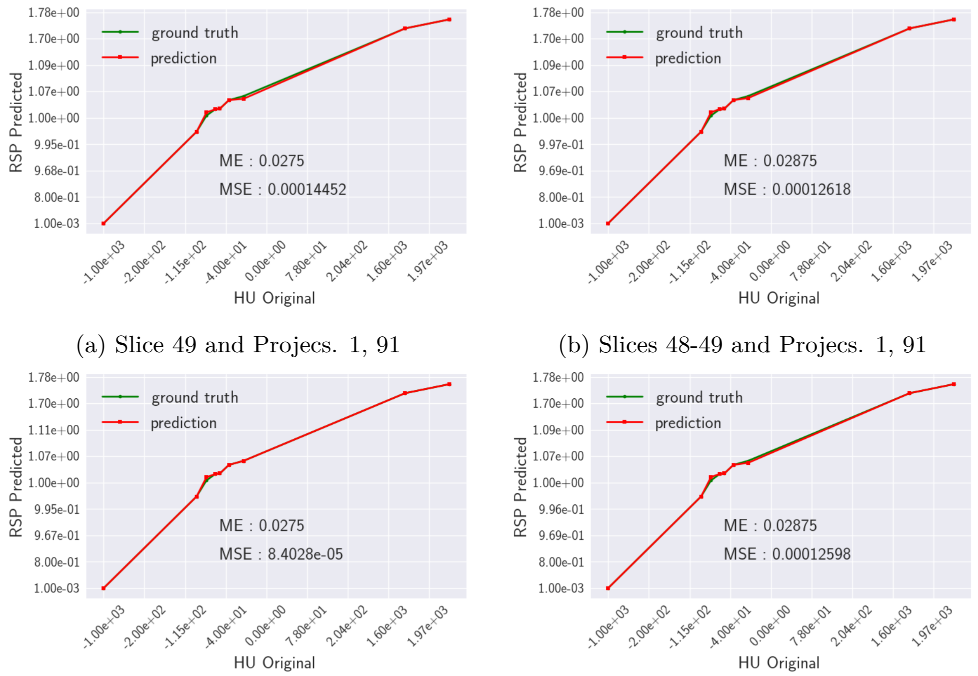
| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 3.05 | 0.10 | 0 | 0 | 1.57 | 0 | 0 |

Table 5.35: Slices 46, 47, 48, 49. Projections 1, 91. Individual error percentage values between prediction and ground truth referring to Fig. 5.30d, ADABOOST method. Maximum value 3.05%.

#### 5.3.3.5 Calibration Curve - Multiple slices and two projection

In Fig. 5.31, once again all the graphics refer to the ADABOOST method; the other methods performed poorly. Following items Fig. 5.31 (a)-(d), the maximum percentage

values between the predicted values and the ground truth were, respectively, 2.92 %, 3.05 %, 2.92 %, 3.05 %.



(a) Slice 49 and Projecs. 1, 91

(b) Slices 48-49 and Projecs. 1, 91

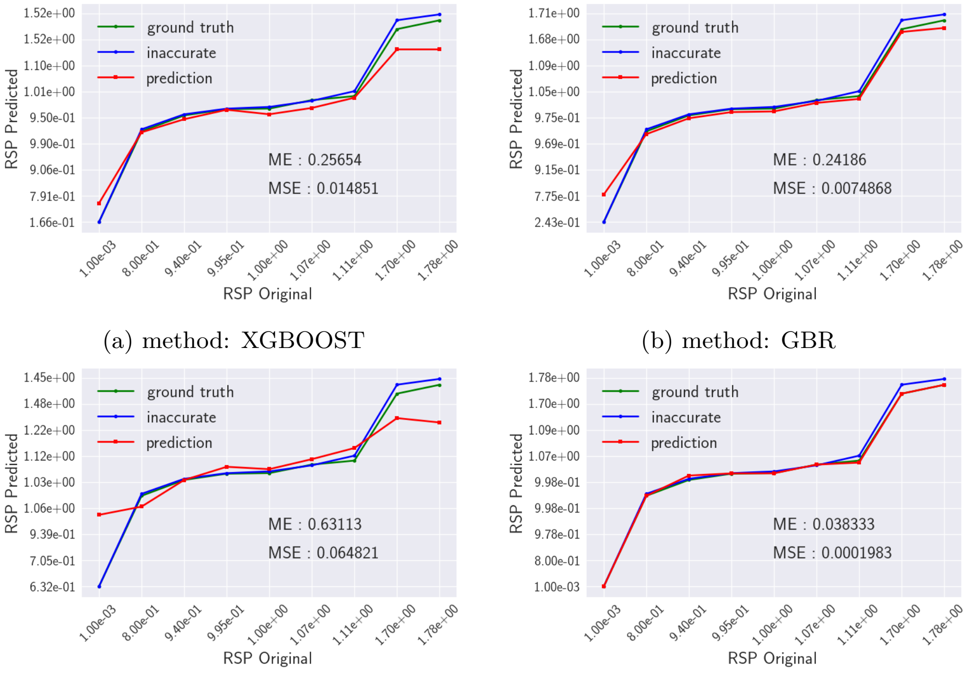
(c) Slices 47-49 and Projecs. 1, 91

(d) Slices 46-49 and Projecs. 1, 91

Figure 5.26: Calibration curves for a single slice and different projections concerning to 5.22d, 5.23d, 5.29d, and 5.30d respectively. The ADABOOST method only. Green line: ground truth, red line: predicted.

## 5.3.4 Different slices and three projections

### 5.3.4.1 Two slices and three projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.27: Slices 41, 42. Projections 41, 121, 161. Green line: ground truth, blue line: inaccurate, and red line: prediction.

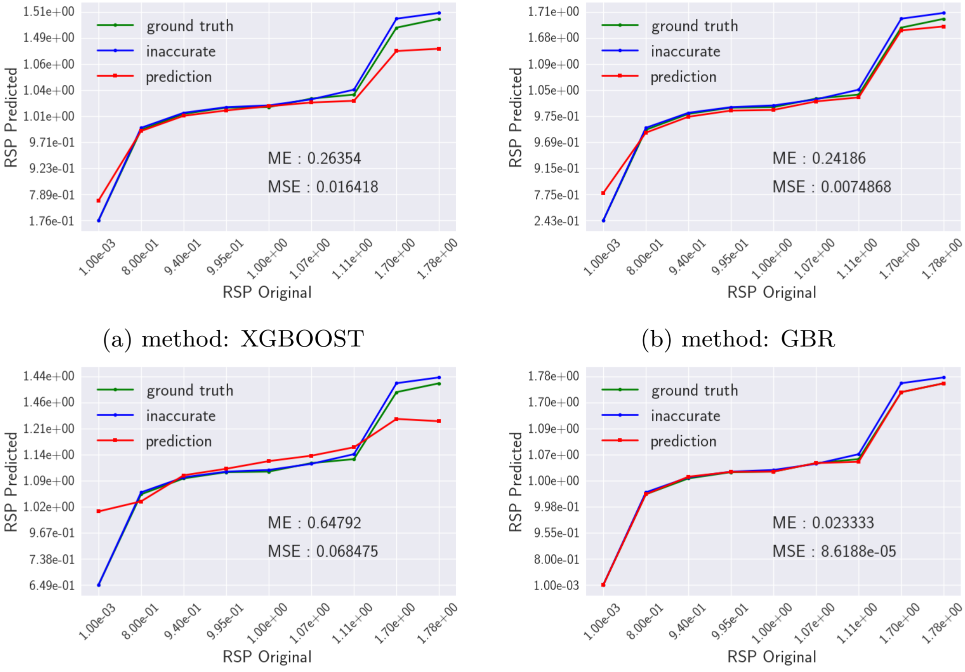
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.25653 | 0.01485 | 0.08678 | 0.93745 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.63112 | 0.06482 | 0.16981 | 0.72699 |
| ADABOOST | 0.03833 | 0.00019 | 0.00666 | 0.99916 |

Table 5.36: Slices 41, 42. Projections 41, 121, 161. Data referring to Fig. 5.27. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 4.07 | 0.25 | 0.16 | 0 | 1.57 | 0 | 0 |

Table 5.37: Slices 41, 42. Projections 41, 121, 161. Individual error percentage values between prediction and ground truth referring to Fig. 5.27d, ADABOOST method. Maximum value 4.07%.

#### 5.3.4.2 Three slices and three projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.28: Slices 40, 41, 42. Projections 41, 121, 161. Green line: ground truth, blue line: inaccurate, and red line: prediction.

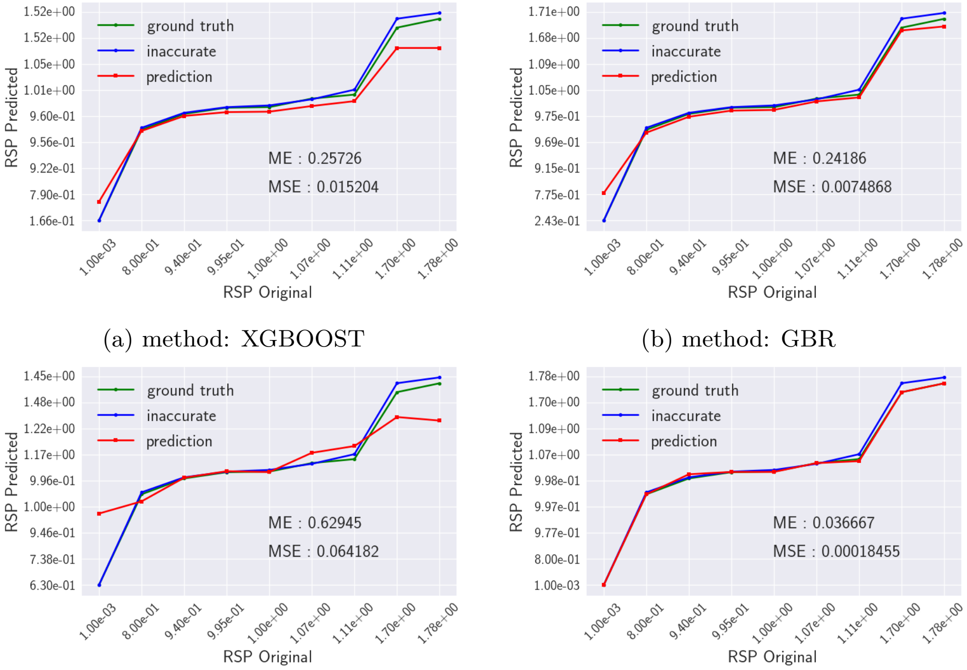
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26354 | 0.01641 | 0.08815 | 0.93085 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.64791 | 0.06847 | 0.17768 | 0.71161 |
| ADABOOST | 0.02333 | 0.00001 | 0.00453 | 0.99963 |

Table 5.38: Slices 40, 41, 42. Projections 41, 121, 161. Data referring to Fig. 5.28. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 1.59 | 0.25 | 0 | 0 | 2.10 | 0 | 0 |

Table 5.39: Slices 40, 41, 42. Projections 41, 121, 161. Individual error percentage values between prediction and ground truth referring to Fig. 5.28d, ADABOOST method. Maximum value 2.10%.

#### 5.3.4.3 Four slices and three projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.29: Slices 39, 40, 41, 42. Projections 41, 121, 161. Green line: ground truth, blue line: inaccurate, and red line: prediction.

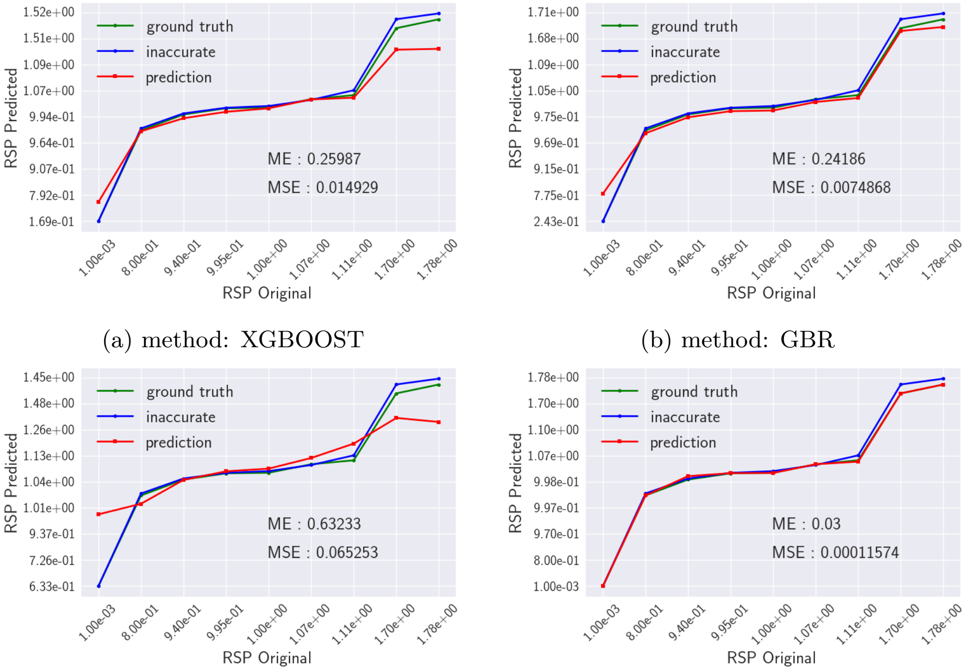
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.25726 | 0.01520 | 0.09248 | 0.93596 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.62945 | 0.06418 | 0.16260 | 0.72969 |
| ADABOOST | 0.03666 | 0.00018 | 0.00651 | 0.99922 |

Table 5.40: Slices 39, 40, 41, 42. Projections 41, 121, 161. Data referring to Fig. 5.29. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 3.9 | 0.20 | 0.25 | 0 | 1.57 | 0 | 0 |

Table 5.41: Slices 39, 40, 41, 42. Projections 41, 121, 1611. Individual error percentage values between prediction and ground truth referring to Fig. 5.29d, ADABOOST method. Maximum value 3.90%.

#### 5.3.4.4 Five slices and three projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.30: Slices 38, 39, 40, 41, 42. Projections 41, 121, 161. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.25986 | 0.01492 | 0.07998 | 0.93172 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.63232 | 0.06525 | 0.16798 | 0.72517 |
| ADABOOST | 0.03000 | 0.00011 | 0.00500 | 0.99951 |

Table 5.42: Slices 38, 39, 40, 41, 42. Projections 41, 121, 161. Data referring to Fig. 5.30. Metrics related to each method.

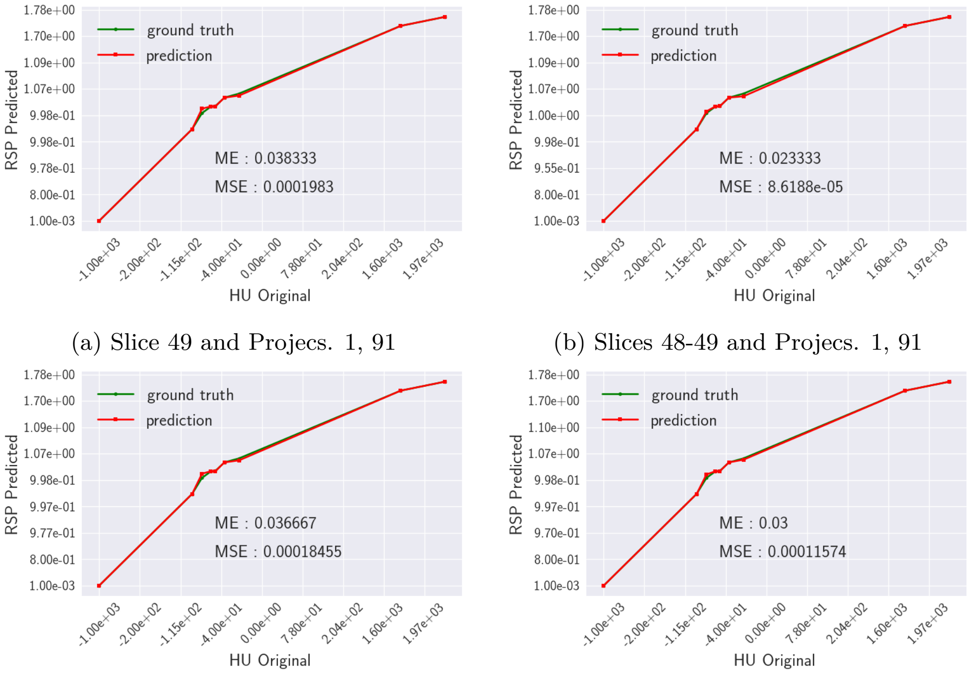
| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 3.19 | 0.16 | 0.16 | 0 | 1.05 | 0 | 0 |

Table 5.43: Slices 38, 39, 40, 41, 42. Projections 41, 121, 161. Individual error percentage values between prediction and ground truth referring to Fig. 5.30d, ADABOOST method. Maximum value 3.19%.

#### 5.3.4.5 Calibration Curve - Multiple slices and three projections

In Fig. 5.31, once again all the graphics refer to the ADABOOST method; the other methods performed poorly. Following items Fig. 5.31 (a)-(d), the maximum percentage

values between the predicted values and the ground truth were, respectively, 4.07 %, 2.10 %, 3.90 %, 3.19 %. See more results in Appendix C.



(a) Slice 49 and Projecs. 1, 91

(b) Slices 48-49 and Projecs. 1, 91

(c) Slices 47-49 and Projecs. 1, 91

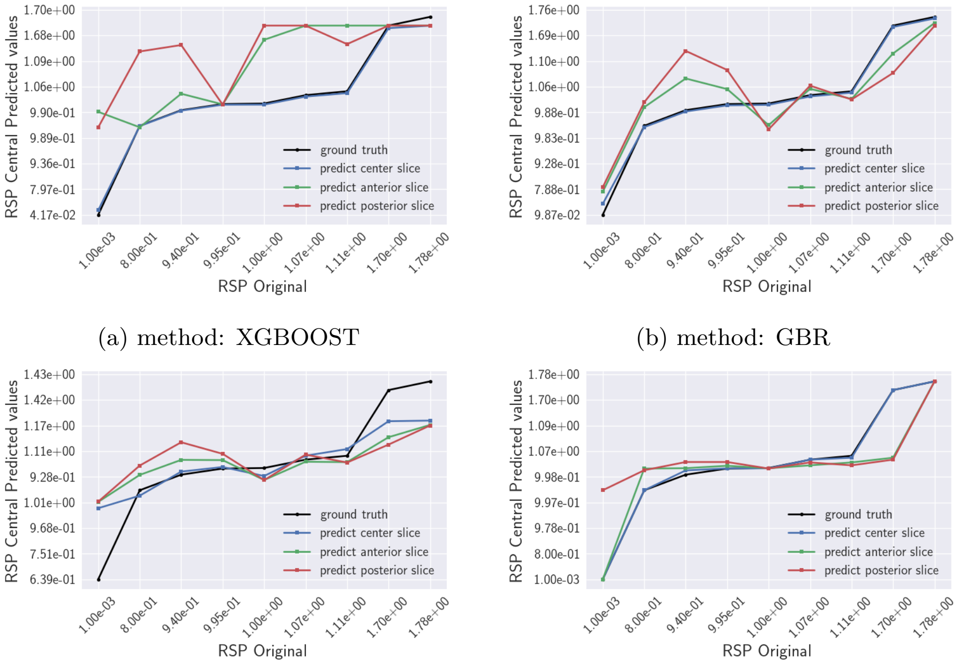
(d) Slices 46-49 and Projecs. 1, 91

Figure 5.31: Calibration curves for a single slice and different projections concerning to 5.22d, 5.23d, 5.29d, and 5.30d respectively. The ADABOOST method only. Green line: ground truth, red line: predicted.

### 5.3.5 Neighboring slices analysis

In the Fig. 5.32, the models have been fitted on slices 45 and 47. The tests have been performed on slice 46. Due to the difference in size between the training and test groups, as previously described, starting from the matrix with the lowest number of columns, this matrix was filled by columns with ZERO values in order to match the number of columns in another corresponding matrix. On the other hand, only with the *XGBOOST* method it was possible to fill in the NAN values columns. The result was similar to filling with ZEROS values, as shown in Fig 5.32(a). Regardless of the parameters used for filling in, the results were not satisfactory. See more results in Appendix D.

#### 5.3.5.1 From central slice to the neighborhood slices



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.32: Fitting model over slice 46. Test model over Slices 45, 47. Projection 1. Dark blue line: ground truth, light blue line: prediction center slice, green line: prediction anterior slice, and red line: prediction posterior slice.

| | ME | | | MSE | | |
|---|---|---|---|---|---|---|
| | anterior | central | posterior | anterior | central | posterior |
| XGBOOST | 0.92427 | 0.07826 | 0.78536 | 0.21645 | 0.00099 | 0.27445 |
| GBR | 0.28480 | 0.09774 | 0.53252 | 0.03124 | 0.00119 | 0.08204 |
| RF | 0.69480 | 0.63801 | 0.69705 | 0.09696 | 0.06917 | 0.11688 |
| ADABOOST | 0.60749 | 0.03833 | 0.79900 | 0.04636 | 0.00019 | 0.12065 |

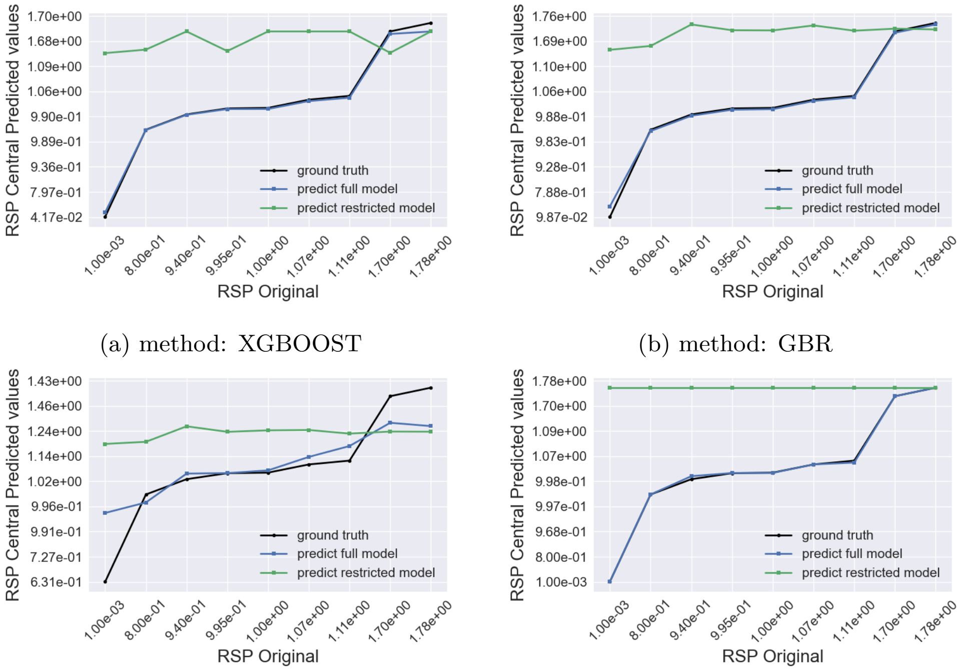
Table 5.44: Table referring to Fig. 5.32. Metrics *ME* and *MSE* are related to each method

| | MAE | | | R-Squared | | |
|---|---|---|---|---|---|---|
| | anterior | central | posterior | anterior | central | posterior |
| XBGOOST | 0.32821 | 0.02158 | 0.42994 | 0.08836 | 0.99582 | -0.15588 |
| GBR | 0.15760 | 0.02172 | 0.24417 | 0.86840 | 0.99496 | 0.65447 |
| RF | 0.22661 | 0.16968 | 0.27150 | 0.59160 | 0.70866 | 0.50771 |
| ADABOOST | 0.11103 | 0.00666 | 0.21027 | 0.80471 | 0.99916 | 0.49186 |

Table 5.45: Table referring to Fig. 5.32. Metrics *MAE* and *R-squared* are related to each method.

#### 5.3.5.2 From neighborhood slices to a missing central slice.

In the Fig. 5.33, the models have been fitted on slices 45, 46, 48, and 49. The tests have been performed on slice 47. Again, the training and test group sizes have been matched from filling by columns with ZERO values. Likewise, only with the *XGBOOST* method was it possible to fill in the NAN values columns. The result was similar to filling with ZEROS values, as shown in Fig 5.33(a). Thus, regardless of the hyperparameters used in the models, and take into account this approach has been made based on a patient-specific and proton-specific model, the results were not satisfactory. See more results in Appendix E.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure 5.33: Fitting model over Slices 45, 46, 48, 49. Test model over missing slice 47. Projection 1. Black line: ground truth, light blue line: prediction full model, green line: prediction restricted model.

| | ME | | MSE | |
|---|---|---|---|---|
| | presented | missed | presented | missed |
| XBGOOST | 0.07826 | 1.50044 | 0.00099 | 0.54603 |
| GBR | 0.09774 | 1.53072 | 0.00119 | 0.60535 |
| RF | 0.63007 | 1.26192 | 0.06789 | 0.30899 |
| ADABOOST | 0.02750 | 1.77700 | 0.00011 | 0.77570 |

Table 5.46: Table referring to Fig. 5.33. Metrics *ME* and *MSE* are related to each method.

| | MAE | | R-squared | |
|---|---|---|---|---|
| | presented | missed | presented | missed |
| XBGOOST | 0.02158 | 0.63409 | 0.99582 | -1.29970 |
| GBR | 0.02172 | 0.65645 | 0.99496 | -1.54951 |
| RF | 0.17482 | 0.47601 | 0.71406 | -0.30138 |
| ADABOOST | 0.00546 | 0.73366 | 0.99949 | -2.26697 |

Table 5.47: Table referring to Fig. 5.33. Metrics *MAE* and *R-squared* are related to each method.

# Chapter 6

# Conclusion and Future Work

> ”Prima di venire qui ero confuso su questo argomento. Dopo aver ascoltato la tua conferenza sono ancora confuso. Ma a un livello superiore”.
>
> Dr. Enrico Fermi

Although machine learning has proven to be a powerful tool in many studies, even using more robust methods, such as Bagging and Boosting, some important dilemmas accompany its implementation, the type of information from the input dataset, therefore the primary purpose of machine learning methods is essential to support decision-making tasks. However, this task is often made impossible either through the limitation of regression algorithms with limited processing of the input data or even by the missing data.

In this initial study, the work was limited to regression algorithms with limited processing of the input data. The study indicates a strong dependence on the correspondence between the training and test datasets, which suggests an obstacle in obtaining a generalizable model. This scenario clearly fits in situations where you want to study an event that has already occurred and, therefore, describe it to obtain a previously known result. In this sense, and as an immediate consequence, the study indicates a high overfitting behavior in any model built under these conditions.

In this study, given the significant dependence on the correspondence between the training and test groups, the random split procedure used to estimate the machine learning algorithms' performance could not be applied due to the structure of the input data set.

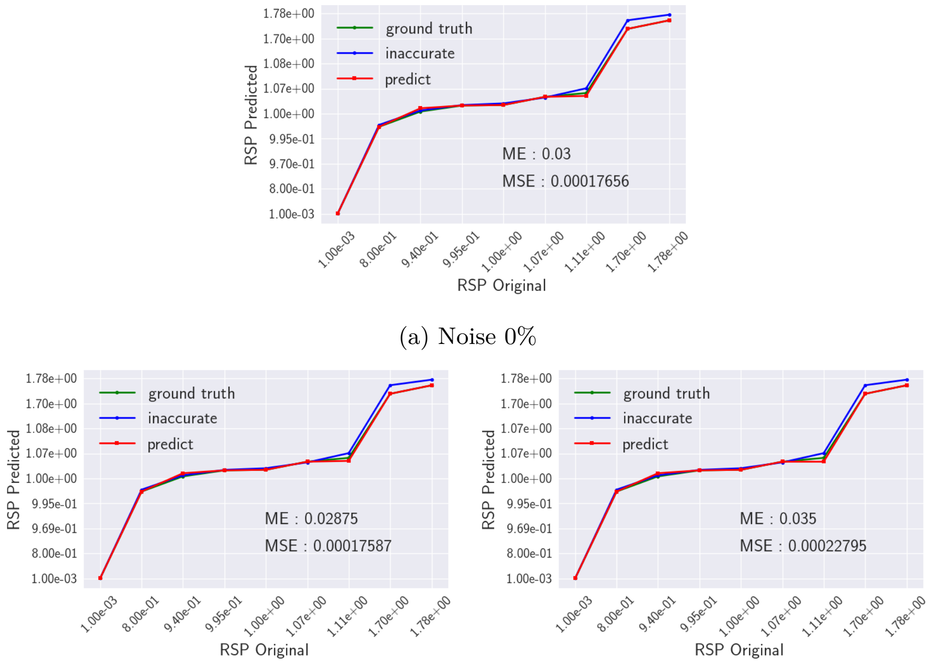
In this study, under this condition of the input dataset, the method based on Bagging procedures, Random Forest Regressor, presented lower qualitative performance than methods that use Boosting procedures. Among these, the Adaptive Boosting method showed superior qualitative performance concerning the Gradient Boosting Regressor and XGBoost.

As for future work, in the sense of researching processes restricted to non-linear regression in machine learning, follow some studies that are reasonable to carry out: (*i*) investigating other ways of obtaining instances for composing the input data set taking into account the structuring between observations and attributes, (*ii*) investigating the performance of ensemble boosting methods in more detail, (*iii*) given the characteristics intrinsic to the analyzed event, the development of a specific method in machine learning would be an excellent form of contribution to research.

# Appendix A

# Analysis of correspondence between training and testing data

## A.1 Slice 45. Projections 1, 91



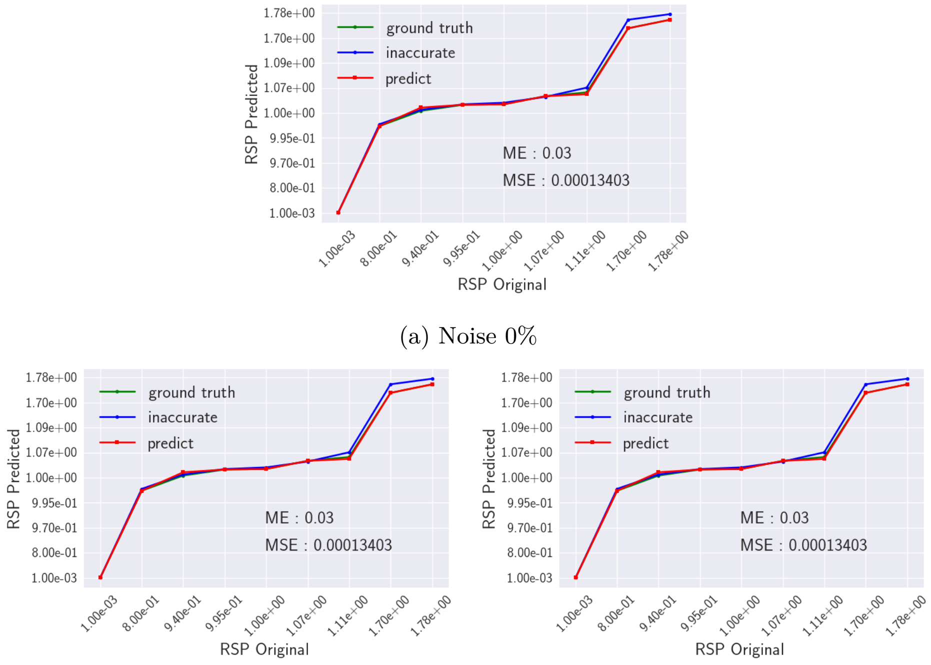
(a) Noise 0%

(b) Noise 1%

(c) Noise 2%

Figure A.1: Prediction performed with slice 45 and projections 1 and 91, corresponds to the equivalence between training and testing datasets. The ADABOOST method only. Green line: ground truth, blue line: inaccurate, and red line: prediction.

## A.2 Slices 44, 45. Projections 21, 121



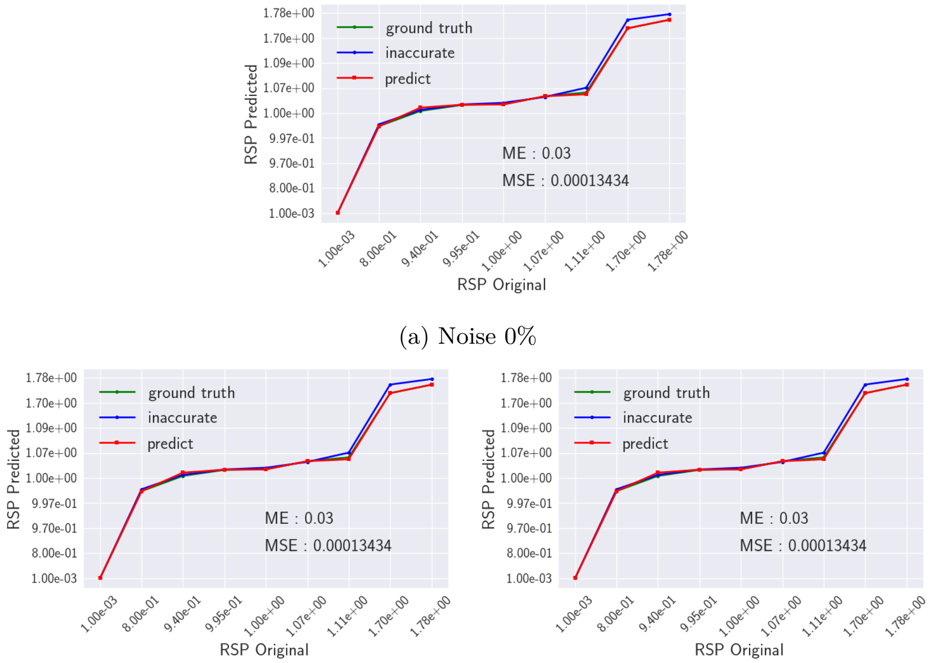
(a) Noise 0%

(b) Noise 1%

(c) Noise 2%

Figure A.2: Prediction performed with slices 44, 45 and projections 21 and 121, corresponds to the equivalence between training and testing datasets. The ADABOOST method only. Green line: ground truth, blue line: inaccurate, and red line: prediction.

## A.3 Slices 35, 36, 37 Projections 1, 61, 101, 161



(a) Noise 0%

(b) Noise 1%

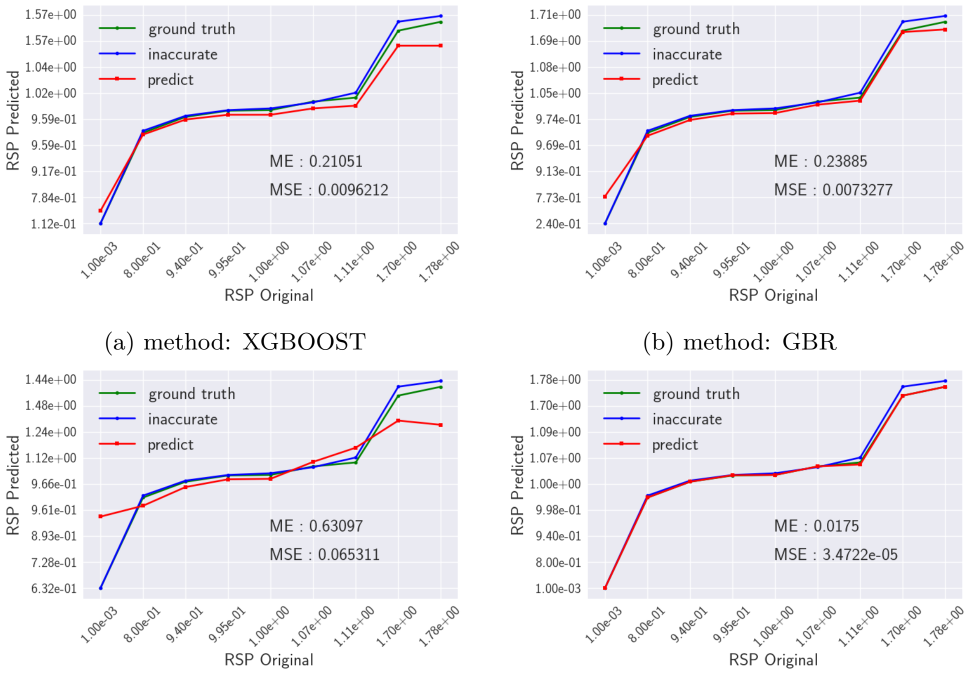
(c) Noise 2%

Figure A.3: Prediction performed with slices 35, 36, 37 Projections 1, 61, 101, 161, corresponds to the equivalence between training and testing datasets. The ADABOOST method only. Green line: ground truth, blue line: inaccurate, and red line: prediction.

# Appendix B

# Single slice and different Projections

## B.1 Slice 31, Projection 91



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure B.1: Slice 31, Projections Slice 31, Projection 91. Green line: ground truth, blue line: inaccurate, and red line: prediction.

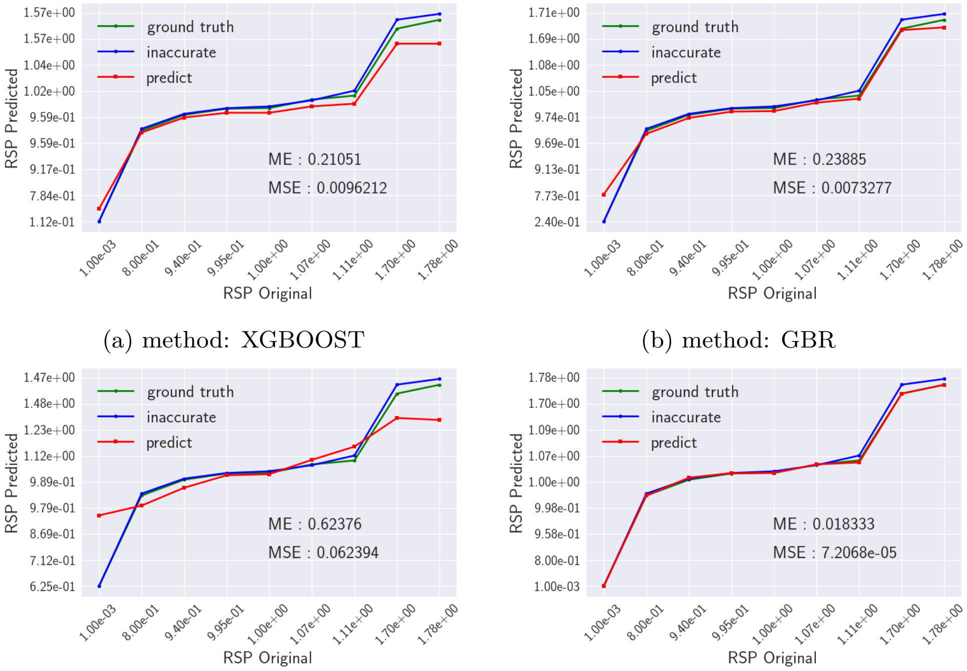
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23884 | 0.00732 | 0.05307 | 0.96913 |
| RF | 0.63096 | 0.06531 | 0.17165 | 0.72493 |
| ADABOOST | 0.01750 | 0.00001 | 0.00222 | 0.99985 |

Table B.1: Slice 31, Projection 91. Data referring to Fig. B.1. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 0 | 0.25 | 0 | 0 | 1.57 | 0 | 0 |

Table B.2: Slice 31, Projection 91. Individual error percentage values between prediction and ground truth referring to Fig. B.5a, ADABOOST method. Maximum value 1.57%.

## B.2 Slice 31, Projections 1, 21



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure B.2: Slice 31, Projections 1, 31. Green line: ground truth, blue line: inaccurate, and red line: prediction.

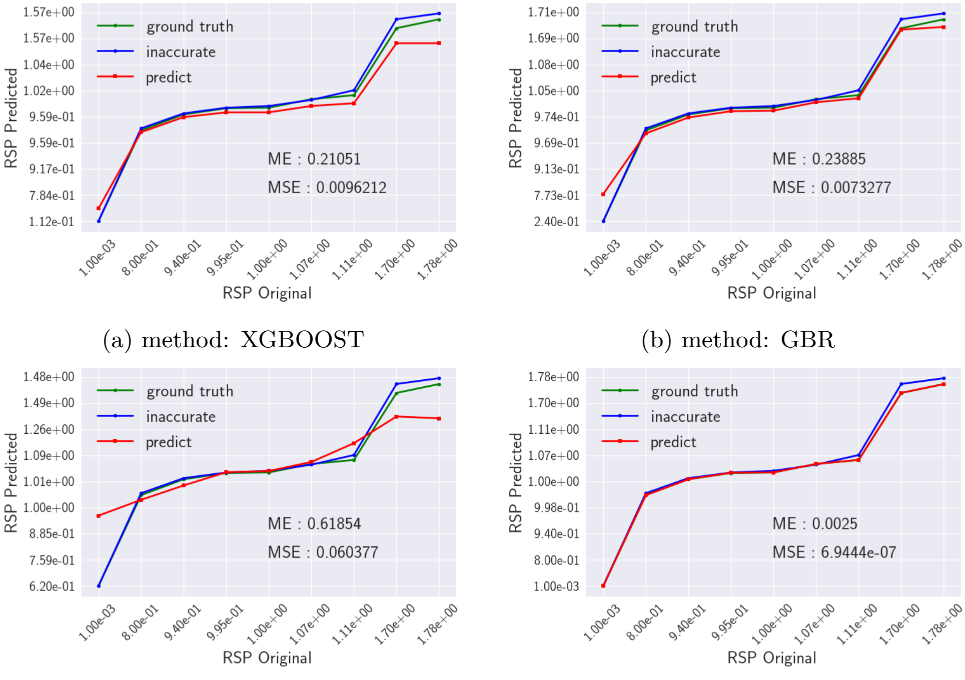
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23884 | 0.00732 | 0.05307 | 0.96913 |
| RF | 0.62376 | 0.06239 | 0.16643 | 0.73721 |
| ADABOOST | 0.01833 | 0.00001 | 0.00425 | 0.99969 |

Table B.3: Slice 31, Projections 1, 21. Data referring to Fig. B.2. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 1.95 | 0.25 | 0 | 0 | 1.57 | 0 | 0 |

Table B.4: Slice 31, Projections 1, 21. Individual error percentage values between prediction and ground truth referring to Fig. B.5b, ADABOOST method. Maximum value 1.95%.

## B.3 Slice 31, Projections 1, 41, 121, 141



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure B.3: Slice 31, Projections 1, 41, 121, 141. Green line: ground truth, blue line: inaccurate, and red line: prediction.

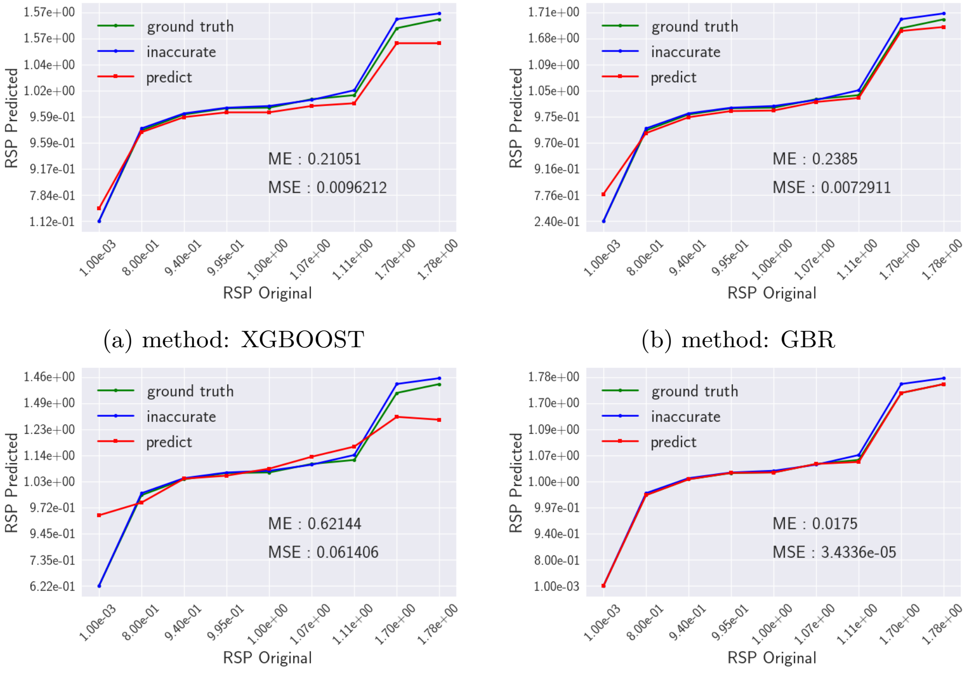
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23884 | 0.00732 | 0.05307 | 0.96913 |
| RF | 0.61854 | 0.06037 | 0.15635 | 0.74571 |
| ADABOOST | 0.00250 | 0.00001 | 0.00027 | 0.99999 |

Table B.5: Slice 31, Projections 1, 41, 121, 141. Data referring to Fig. B.3. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 0 | 0.25 | 0 | 0 | 0 | 0 | 0 |

Table B.6: Slice 31, Projections 1, 41, 121, 141. Individual error percentage values between prediction and ground truth referring to Fig. B.5c, ADABOOST method. Maximum value 0.25%.

## B.4 Slice 31, All projections



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure B.4: Slice 31, All projections. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.21051 | 0.00962 | 0.07789 | 0.95947 |
| GBR | 0.23884 | 0.00729 | 0.05300 | 0.96929 |
| RF | 0.62144 | 0.06140 | 0.16148 | 0.74137 |
| ADABOOST | 0.01750 | 0.00001 | 0.00212 | 0.99985 |

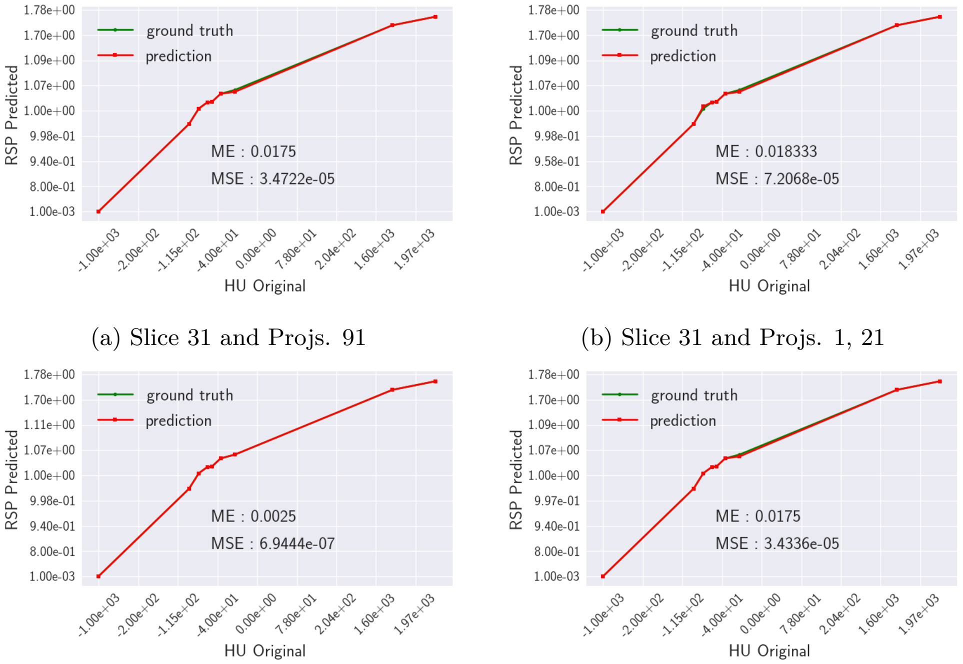
Table B.7: Slice 31, All projections. Data referring to Fig. B.4. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 0 | 0.16 | 0 | 0 | 1.57 | 0 | 0 |

Table B.8: Slice 31, All projections. Individual error percentage values between prediction and ground truth referring to Fig. B.5d, ADABOOST method. Maximum value 1.57%.

## B.5 Calibration Curve - Single slice and different Projections

In Fig. B.5, once again all the graphics refer to the ADABOOST method; the other methods performed poorly. Following items Fig. B.5 (a)-(d), the maximum percentage values between the predictioned values and the ground truth were, respectively, 1.57 %, 1.95 %, 0.25 %, 1.57 %.



(a) Slice 31 and Projs. 91

(b) Slice 31 and Projs. 1, 21

(c) Slice 31 and Projs.1, 41, 121, 141

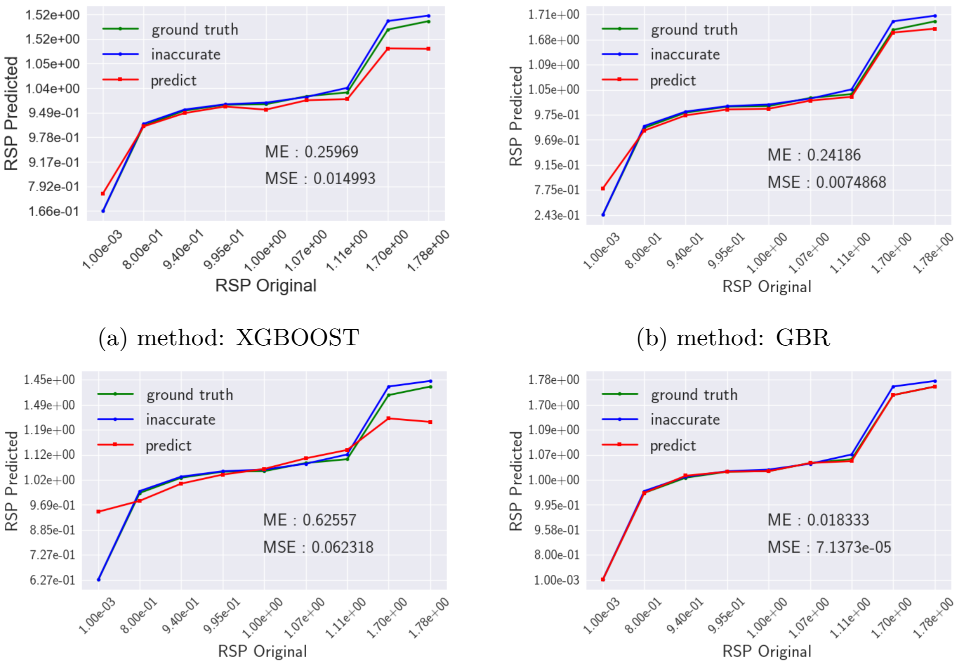
(d) Slice 31 and All Projs.

Figure B.5: Calibration curves for a single slice and different projections concerning to B.5a, B.5b, B.5c, and B.5d respectively. The ADABOOST method only. Green line: ground truth, red line: predicted.

# Appendix C

# Different Slices and different Projections

## C.1 Slices 35, 36, Projection 141



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure C.1: Slice 35, 36 and Projection 141. Green line: ground truth, blue line: inaccurate, and red line: prediction.

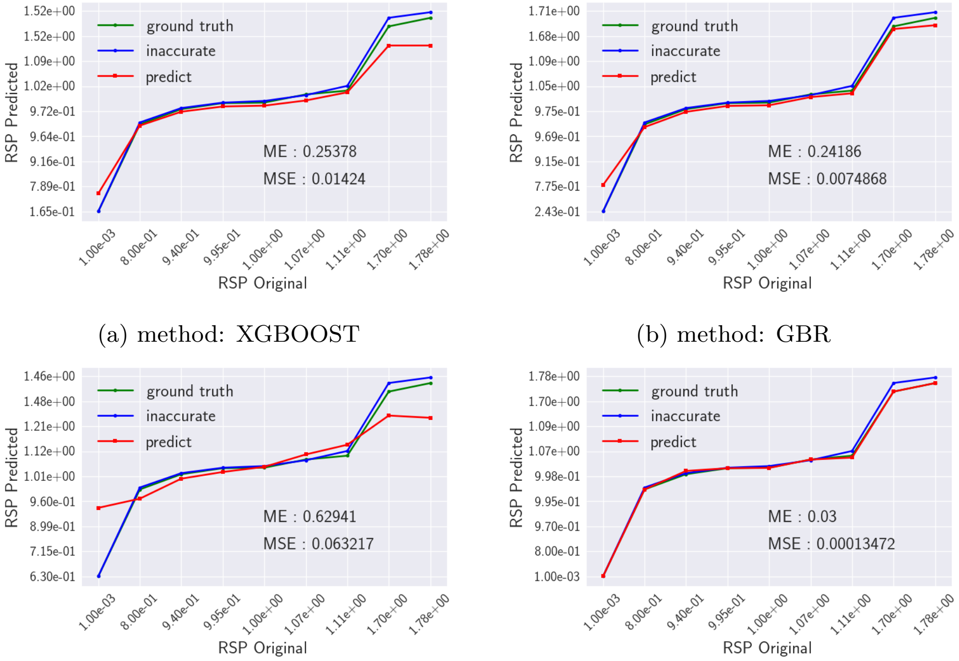
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.25969 | 0.01499 | 0.08901 | 0.93685 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.62557 | 0.06231 | 0.16291 | 0.73753 |
| ADABOOST | 0.01833 | 0.00001 | 0.00398 | 0.99969 |

Table C.1: Table referring to Fig. C.1. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 1.95 | 0 | 0 | 0 | 1.57 | 0 | 0 |

Table C.2: Slice 35, 36 and Projection 141. Individual error percentage values between prediction and ground truth referring to Fig. C.5a, ADABOOST method. Maximum value 1.95%.

## C.2 Slices 37, 38. Projections 41, 141



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure C.2: Slice Slices 37, 38. Projections 41, 141. Green line: ground truth, blue line: inaccurate, and red line: prediction.

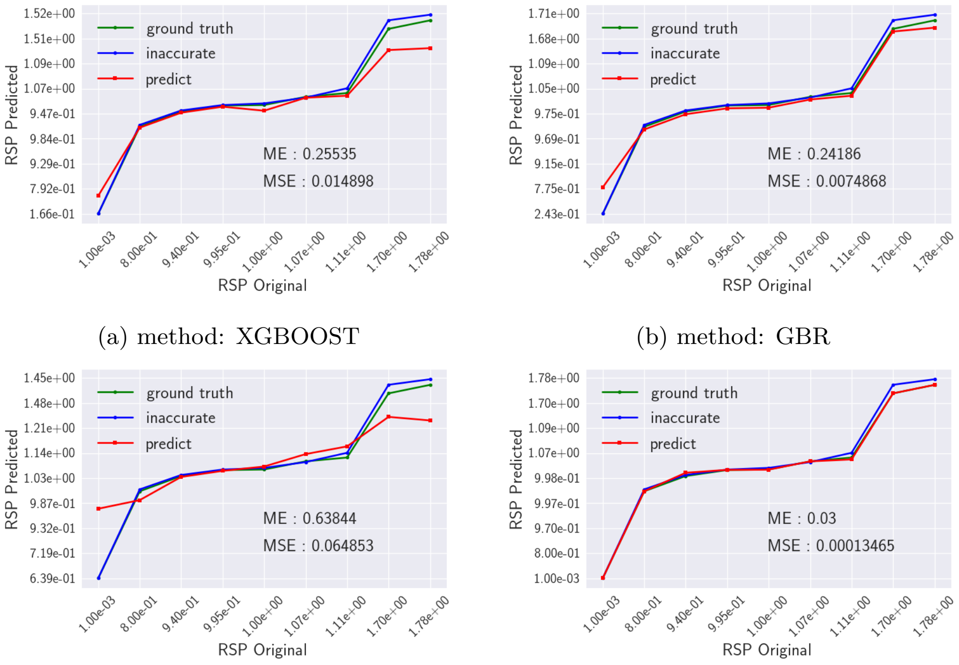
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.25377 | 0.01424 | 0.08444 | 0.94002 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.62941 | 0.06321 | 0.16493 | 0.73375 |
| ADABOOST | 0.03000 | 0.00013 | 0.00555 | 0.99943 |

Table C.3: Slices 37, 38. Projections 41, 141. Data referring to Fig. C.2. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 3.19 | 0 | 0.25 | 0 | 1.57 | 0 | 0 |

Table C.4: Slice 37, 38. Projections 41, 141. Individual error percentage values between prediction and ground truth referring to Fig. C.5b, ADABOOST method. Maximum value 3.19%.

## C.3 Slices 39, 40. Projections 21, 61, 121



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure C.3: Slices 39, 40. Projections 21, 61, 121. Green line: ground truth, blue line: inaccurate, and red line: prediction.

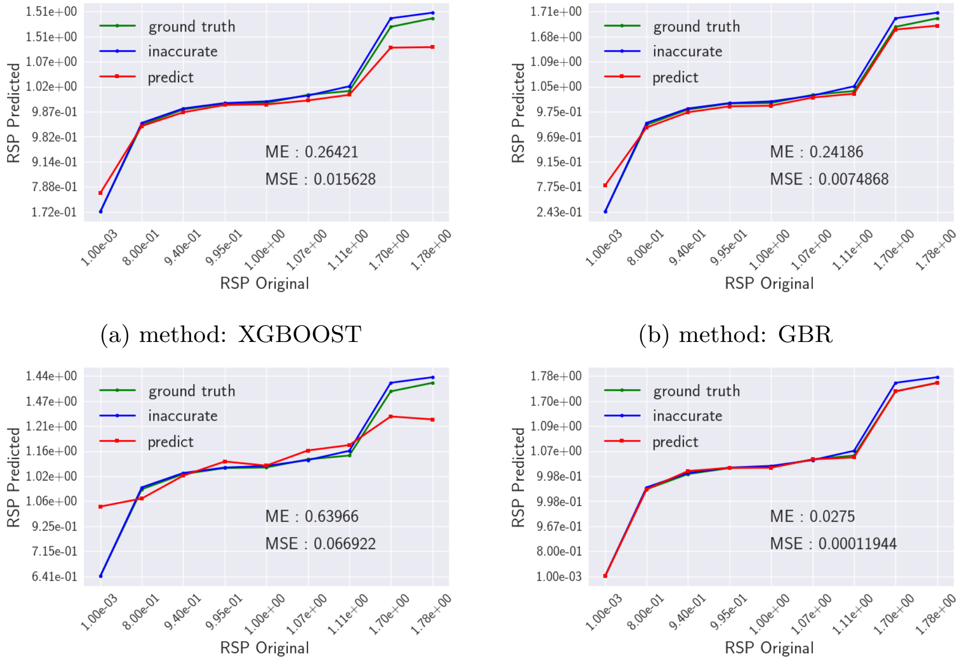
| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.25535 | 0.01489 | 0.08109 | 0.93725 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.63844 | 0.06485 | 0.16389 | 0.72686 |
| ADABOOST | 0.03000 | 0.00013 | 0.00564 | 0.99943 |

Table C.5: Slices 39, 40. Projections 21, 61, 121. Data referring to Fig. C.3. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 3.19 | 0.16 | 0.16 | 0 | 1.57 | 0 | 0 |

Table C.6: Slice 39, 40. Projections 21, 61, 121. Individual error percentage values between prediction and ground truth referring to Fig. C.5c, ADABOOST method. Maximum value 3.19%.

## C.4 Slices 41, 42. Projections 1, 91, 101, 161



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure C.4: Slices 41, 42. Projections 1, 91, 101, 161. Green line: ground truth, blue line: inaccurate, and red line: prediction.

| | ME | MSE | MAE | R-Squared |
|---|---|---|---|---|
| XGBOOST | 0.26421 | 0.01562 | 0.08633 | 0.93418 |
| GBR | 0.24185 | 0.00748 | 0.05374 | 0.96846 |
| RF | 0.63966 | 0.06692 | 0.17345 | 0.71815 |
| ADABOOST | 0.02749 | 0.00011 | 0.00555 | 0.99949 |

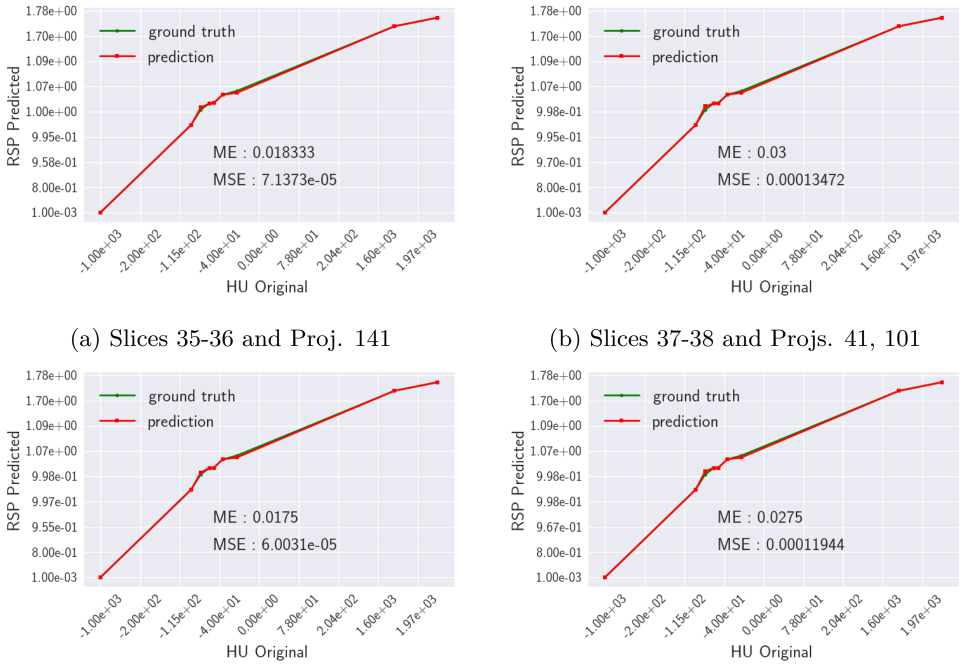
Table C.7: Slices 41, 42. Projections 1, 91, 101, 161. Data referring to Fig. C.4. Metrics related to each method.

| ADABOOST | Point 1 | Point 2 | Point 3 | Point 4 | Point 5 | Point 6 | Point 7 | Point 8 | Point 9 |
|---|---|---|---|---|---|---|---|---|---|
| Error (%) | 0 | 0 | 2.95 | 0.25 | 0.25 | 0 | 1.57 | 0 | 0 |

Table C.8: Slices 41, 42. Projections 1, 91, 101, 161. Individual error percentage values between prediction and ground truth referring to Fig. C.5d, ADABOOST method. Maximum value 2.95%.

## C.5 Calibration Curve - Different Slices and different Projections

In Fig. C.5, once again all the graphics refer to the ADABOOST method; the other methods performed poorly. Following items Fig. C.5 (a)-(d), the maximum percentage values between the predictioned values and the ground truth were, respectively, 1.95 %, 3.19 %, 3.19 %, 2.95 %.



(a) Slices 35-36 and Proj. 141

(b) Slices 37-38 and Projs. 41, 101

(c) Slices 39-40 and Projs. 21, 61, 121

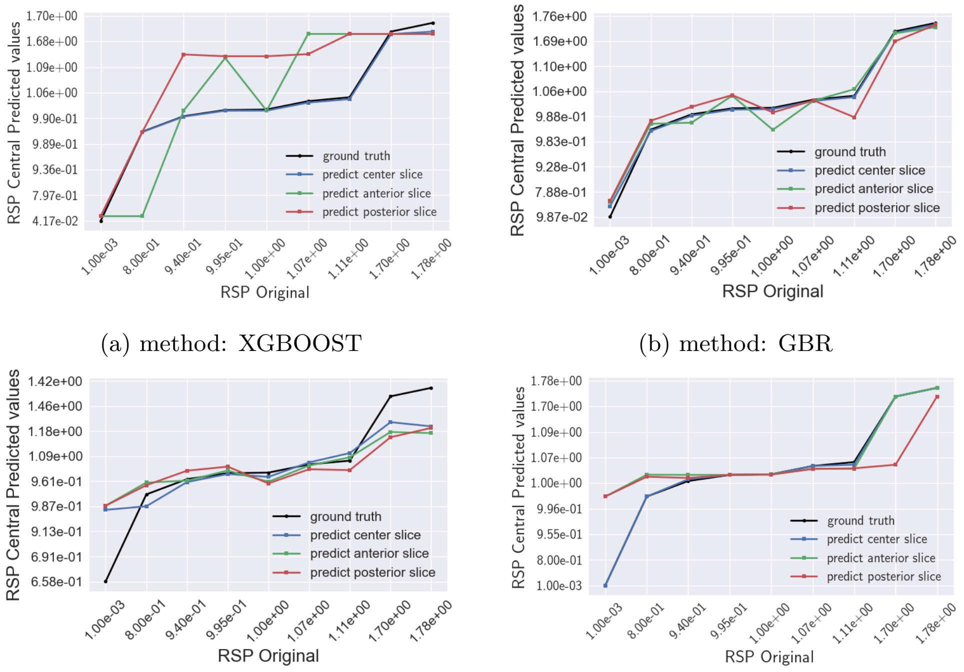
(d) Slices 41-42 and Projs. 1, 91, 101, 161

Figure C.5: Calibration curves for a single slice and different projections concerning to 5.17d, 5.18d, 5.19d, and 5.20d respectively. The ADABOOST method only. Green line: ground truth, red line: predicted.

Appendix D

# From central slice to the neighborhood slices

## D.1 Slices 35, 36, 37. Projection 1.



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure D.1: Fitting model over slice 36. Test model over Slices 35, 37. Projection 1. Dark blue line: ground truth, light blue line: prediction center slice, green line: prediction anterior slice, and red line: prediction posterior slice.

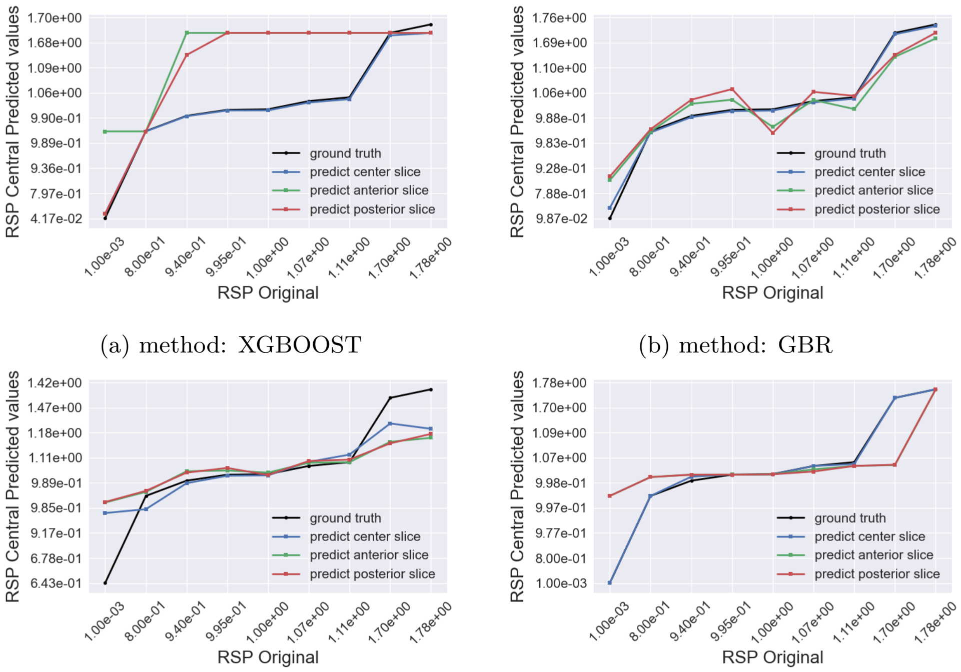
| | ME | | | MSE | | |
|---|---|---|---|---|---|---|
| | anterior | central | posterior | anterior | central | posterior |
| XGBOOST | 0.75829 | 0.07826 | 0.56892 | 0.16629 | 0.00099 | 0.14262 |
| GBR | 0.20171 | 0.09774 | 0.19640 | 0.00964 | 0.00119 | 0.01074 |
| RF | 0.69555 | 0.65738 | 0.69511 | 0.08703 | 0.07027 | 0.08840 |
| ADABOOST | 0.79900 | .023333 | 0.79900 | 0.07603 | 0.00001 | 0.11745 |

Table D.1: Referring to Fig. D.1. Metrics *ME* and *MSE* are related to each method

| | MAE | | | R-Squared | | |
|---|---|---|---|---|---|---|
| | anterior | central | posterior | anterior | central | posterior |
| XBGOOST | 0.29116 | 0.02158 | 0.29674 | 0.29963 | 0.99582 | 0.39930 |
| GBR | 0.07888 | 0.02172 | 0.08582 | 0.95938 | 0.99496 | 0.65447 |
| RF | 0.18974 | 0.16872 | 0.21010 | 0.63343 | 0.07027 | 0.08840 |
| ADABOOST | 0.11103 | 0.00666 | 0.21027 | 0.80471 | 0.70401 | 0.62766 |

Table D.2: Referring to Fig. D.1. Metrics *MAE* and *R-squared* are related to each method.

## D.2 Slices 39, 40, 41. Projection 121



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure D.2: Fitting model over slice 40. Test model over Slices 39, 41. Projection 121. Dark blue line: ground truth, light blue line: prediction center slice, green line: prediction anterior slice, and red line: prediction posterior slice.

| | ME | | | MSE | | |
|---|---|---|---|---|---|---|
| | anterior | central | posterior | anterior | central | posterior |
| XGBOOST | 0.79638 | 0.07826 | 0.70473 | 0.32688 | 0.00099 | 0.22696 |
| GBR | 0.34889 | 0.09774 | 0.38371 | 0.02720 | 0.00119 | 0.03406 |
| RF | 0.73857 | 0.64180 | 0.74296 | 0.10191 | 0.0688 | 0.10086 |
| ADABOOST | 0.79900 | 0.03666 | 0.79900 | 0.11703 | 0.00018 | 0.11681 |

Table D.3: Referring to Fig. D.2. Metrics *ME* and *MSE* are related to each method

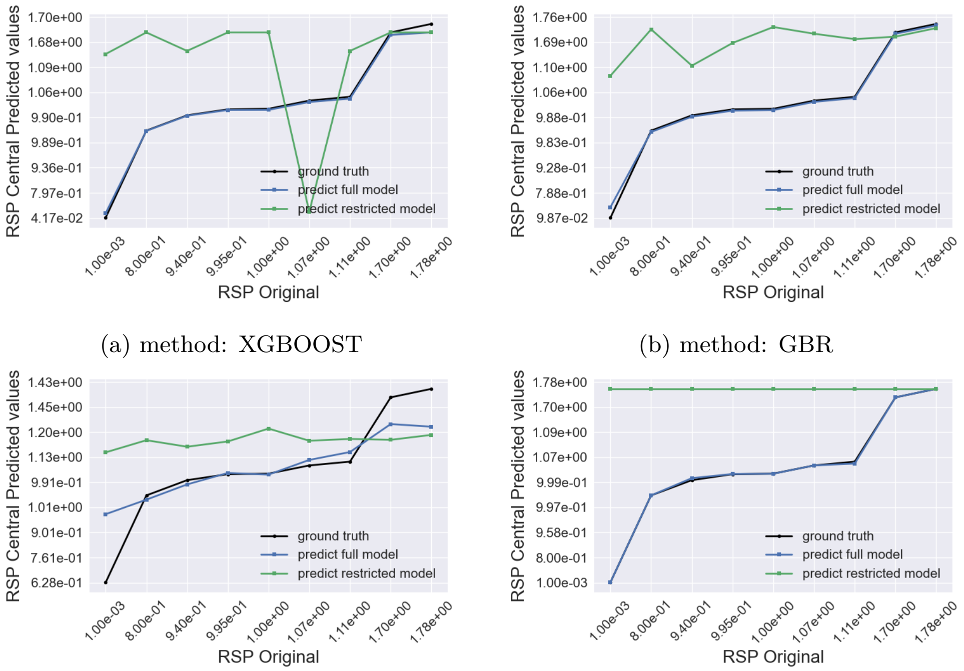
| | MAE | | | R-Squared | | |
|---|---|---|---|---|---|---|
| | anterior | central | posterior | anterior | central | posterior |
| XBGOOST | 0.47291 | 0.02158 | 0.36645 | -0.37670 | 0.99582 | 0.04409 |
| GBR | 0.13141 | 0.02172 | 0.14804 | 0.88541 | 0.99496 | 0.65447 |
| RF | 0.19907 | 0.16785 | 0.20318 | 0.57077 | 0.70994 | 0.57518 |
| ADABOOST | 0.19031 | 0.00648 | 0.19200 | 0.50708 | 0.99922 | 0.50800 |

Table D.4: Referring to Fig. D.2. Metrics *MAE* and *R-squared* are related to each method.

# Appendix E

# From neighborhood slices to a missing central slice

## E.1 44, 46. Projection 1



(a) method: XGBOOST

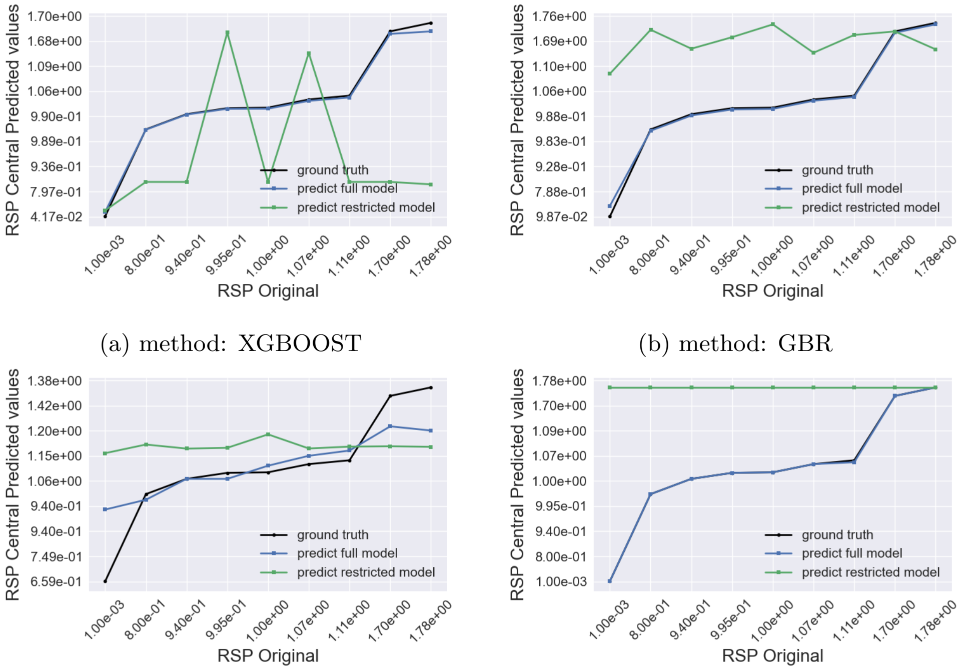
(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure E.1: Fitting model over Slices 44, 46. Test model over missing slice 45. Projection 1. Black line: ground truth, light blue line: prediction full model, green line: prediction restricted model.

## E.2 47, 49. Projection 91



(a) method: XGBOOST

(b) method: GBR

(c) method: RF

(d) method: ADABOOST

Figure E.2: Fitting model over Slices 47, 49. Test model over missing slice 48. Projection 91. Black line: ground truth, light blue line: prediction full model, green line: prediction restricted model.

# Bibliography

[1] R. Baskar, K. A. Lee, R. Yeo, and K.-W. Yeoh, "Cancer and radiation therapy: current advances and future directions," International journal of medical sciences, vol. 9, no. 3, p. 193, 2012. [Online]. Available: https://www.medsci.org/v09p0193.htm [Cited on page 1.]

[2] O. Sokol, E. Scifoni, W. Tinganelli, W. Kraft-Weyrather, J. Wiedemann, A. Maier, D. Boscolo, T. Friedrich, S. Brons, M. Durante, and M. Krämer, "Oxygen beams for therapy: advanced biological treatment planning and experimental verification," Physics in Medicine  Biology, vol. 62, no. 19, pp. 7798–7813, sep 2017. [Online]. Available: https://iopscience.iop.org/article/10.1088/1361-6560/aa88a0 [Cited on page 1.]

[3] H. Paganetti, Proton therapy physics. CRC press, 2018. [Online]. Available: https://www.amazon.com/Therapy-Physics-Medical-Biomedical-Engineering/dp/1138626503 [Cited on pages 1 and 3.]

[4] R. R. Wilson, "Radiological use of fast protons," Radiology, vol. 47, no. 5, pp. 487–491, 1946. [Online]. Available: https://doi.org/10.1148/47.5.487 [Cited on page 1.]

[5] R. P. Johnson, "Review of medical radiography and tomography with proton beams," Reports on Progress in Physics, vol. 81, no. 1, p. 016701, 2017. [Online]. Available: https://doi.org/10.1088/1361-6633/aa8b1d [Cited on pages 1 and 2.]

[6] U. Schneider, E. Pedroni, and A. Lomax, "The calibration of CT Hounsfield units for radiotherapy treatment planning," Physics in Medicine & Biology, vol. 41, no. 1, p. 111, 1996. [Online]. Available: https://iopscience.iop.org/article/10.1088/0031-9155/41/1/009 [Cited on pages 2, 8, 16, 19, and 20.]

[7] M. D. Bethesda, “Icru 1989 tissue substitutes in radiation dosimetry and measurement report 44,” Commission on Radiation Units and Measurements. [Cited on page 2.]

[8] M. Berger, M. Inokuti, H. Andersen, H. Bichsel, D. Powers, S. Seltzer, D. Thwaites, and D. Watt, “Report 49,” Journal of the International Commission on Radiation Units and Measurements, no. 2, pp. NP–NP, 1993. [Cited on page 2.]

[9] J. J. P. R.-E. Tables, “data set.” PROTON RANGE-ENERGY TABLES, 1 keV-10 GeV Energy Loss, Range, Path Length, Time-of-Flight, Straggling, Multiple Scattering, and Nuclear Interaction Probability In Two Parts. Part 1. For 63 Compounds Part 2. For Elements 1< 2 d 92, vol. 27, p. 182, 1982. [Online]. Available: https://doi.org/10.1016/0092-640X(82)90004-3 [Cited on page 2.]

[10] Y. Kumazaki, T. Akagi, T. Yanou, D. Suga, Y. Hishikawa, and T. Teshima, “Determination of the mean excitation energy of water from proton beam ranges,” Radiation Measurements, vol. 42, no. 10, pp. 1683–1691, 2007. [Online]. Available: https://doi.org/10.1016/j.radmeas.2007.10.019 [Cited on page 2.]

[11] M. Berger, M. Inokuti, H. Anderson, H. Bichsel, J. Dennis, D. Powers, S. Seltzer, and J. Turner, “Report 37,” Journal of the International Commission on Radiation Units and Measurements, no. 2, pp. NP–NP, 1984. [Online]. Available: https://doi.org/10.1093/jicru/os19.2.Report37 [Cited on pages 2 and 7.]

[12] C. Gianoli, M. Göppel, S. Meyer, P. Palaniappan, M. Rädler, F. Kamp, C. Belka, M. Riboldi, and K. Parodi, “Patient-specific ct calibration based on ion radiography for different detector configurations in 1h, 4he and 12c ion pencil beam scanning,” Physics in Medicine & Biology, 2020. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/32629442/ [Cited on pages 3, 13, 15, 16, 19, and 36.]

[13] E. L. Alpen, W. Saunders, A. Chatterjee, J. Llacer, G. T. Chen, and J. Scherer, “A comparison of water equivalent thickness measurements: Ct method vs. heavy ion beam technique,” The British journal of radiology, vol. 58, no. 690, pp. 543–548, 1985. [Online]. Available: https://www.birpublications.org/doi/10.1259/0007-1285-58-690-543 [Cited on page 3.]

[14] B. Schaffner and E. Pedroni, “The precision of proton range calculations in proton radiotherapy treatment planning: experimental verification of the relation between ct-hu and proton stopping power,” Physics in Medicine & Biology, vol. 43, no. 6, p.

1579, 1998. [Online]. Available: https://doi.org/10.1088/0031-9155/43/6/016 [Cited on page 3.]

[15] S. J. Frank and X. R. Zhu, Proton Therapy E-Book: Indications, Techniques, and Outcomes. Elsevier Health Sciences, 2020. [Online]. Available: https://www.eu.elsevierhealth.com/proton-therapy-9780323733496.html#description [Cited on page 5.]

[16] D. Henzen, "Monte carlo based treatment planning for modulated electron radiotherapy using a photon multileaf collimator," Ph.D. dissertation, ETH Zurich, 2014. [Online]. Available: https://doi.org/10.1093/jicru/os23.1.Report44 [Cited on pages xv and 6.]

[17] U. Linz, Ion beam therapy: fundamentals, technology, clinical applications. Springer Science & Business Media, 2011. [Online]. Available: https://www.springer.com/gp/book/9783642214134 [Cited on pages 6, 7, and 8.]

[18] S. Deffet, "Proton radiography to reduce range uncertainty in proton therapy," Ph.D. dissertation, UCL-Université Catholique de Louvain, 2018. [Online]. Available: https://dial.uclouvain.be/pr/boreal/object/boreal%3A208431/datastream/PDF_02/view [Cited on pages 7, 8, and 16.]

[19] J. Deasy, "Icru report 49, stopping powers and ranges for protons and alph particles," Medical physics, vol. 21, no. 5, pp. 709–710, 1994. [Online]. Available: https://doi.org/10.1118/1.597176 [Cited on pages 7 and 12.]

[20] D. J. Thomas, "Icru report 85: fundamental quantities and units for ionizing radiation," 2012. [Online]. Available: https://doi.org/10.1093/rpd/ncs077 [Cited on page 8.]

[21] P. Andreo, D. T. Burns, A. E. Nahum, J. Seuntjens, and F. H. Attix, Fundamentals of ionizing radiation dosimetry. John Wiley & Sons, 2017. [Online]. Available: https://www.wiley.com/en-us/Fundamentals+of+Ionizing+Radiation+Dosimetry-p-9783527409211 [Cited on pages xv and 9.]

[22] X. Llovet, C. J. Powell, F. Salvat, and A. Jablonski, "Cross sections for inner-shell ionization by electron impact," Journal of Physical and Chemical Reference Data, vol. 43, no. 1, p. 013102, 2014. [Online]. Available: https://doi.org/10.1063/1.4832851 [Cited on pages xv and 9.]

[23] J. L. Prince and J. M. Links, Medical imaging signals and systems. Pearson Prentice Hall Upper Saddle River, 2006. [Online]. Available: https://www.pearson.com/us/higher-education/program/Prince-Medical-Imaging-Signals-and-Systems-2nd-Edition/PGM60718.html [Cited on pages xv, 10, 11, and 12.]

[24] A. M. Cormack, "Representation of a function by its line integrals, with some radiological applications," Journal of applied physics, vol. 34, no. 9, pp. 2722–2727, 1963. [Online]. Available: https://doi.org/10.1063/1.1729798 [Cited on pages 12 and 15.]

[25] R. Schulte, S. Penfold, J. Tafas, and K. Schubert, "A maximum likelihood proton path formalism for application in proton computed tomography," Medical physics, vol. 35, no. 11, pp. 4849–4856, 2008. [Online]. Available: https://doi.org/10.1118/1.2986139 [Cited on pages 12 and 15.]

[26] W. D. Newhauser and R. Zhang, "The physics of proton therapy," Physics in Medicine & Biology, vol. 60, no. 8, p. R155, 2015. [Online]. Available: https://iopscience.iop.org/article/10.1088/0031-9155/60/8/R155/meta [Cited on page 13.]

[27] N. Krah, F. Khellaf, J. M. Létang, S. Rit, and I. Rinaldi, "A comprehensive theoretical comparison of proton imaging set-ups in terms of spatial resolution," Physics in Medicine & Biology, vol. 63, no. 13, p. 135013, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01909706 [Cited on pages xv, 14, and 16.]

[28] U. Schneider and E. Pedroni, "Proton radiography as a tool for quality control in proton therapy," Medical physics, vol. 22, no. 4, pp. 353–363, 1995. [Online]. Available: https://doi.org/10.1118/1.597470 [Cited on page 14.]

[29] U. Schneider, J. Besserer, P. Pemler, M. Dellert, M. Moosburger, E. Pedroni, and B. Kaser-Hotz, "First proton radiography of an animal patient," Medical physics, vol. 31, no. 5, pp. 1046–1051, 2004. [Online]. Available: https://doi.org/10.1118/1.1690713 [Cited on page 15.]

[30] U. Schneider, P. Pemler, J. Besserer, E. Pedroni, A. Lomax, and B. Kaser-Hotz, "Patient specific optimization of the relation between ct-hounsfield units and proton stopping power with proton radiography," Medical physics, vol. 32, no. 1, pp.

195–199, 2005. [Online]. Available: https://doi.org/10.1118/1.1833041 [Cited on page 15.]

[31] P. Doolan, M. Testa, G. Sharp, E. Bentefour, G. Royle, and H. Lu, "Patient-specific stopping power calibration for proton therapy planning based on single-detector proton radiography," Physics in Medicine & Biology, vol. 60, no. 5, p. 1901, 2015. [Online]. Available: https://iopscience.iop.org/article/10.1088/0031-9155/60/5/1901/meta [Cited on page 15.]

[32] N. Depauw, M. F. Dias, A. Rosenfeld, and J. C. Seco, "Ion radiography as a tool for patient set-up & image guided particle therapy: a monte carlo study," Technology in cancer research & treatment, vol. 13, no. 1, pp. 69–76, 2014. [Online]. Available: https://doi.org/10.7785/tcrt.2012.500357 [Cited on page 15.]

[33] U. Schneider and E. Pedroni, "Multiple coulomb scattering and spatial resolution in proton radiography," Medical physics, vol. 21, no. 11, pp. 1657–1663, 1994. [Online]. Available: https://doi.org/10.1118/1.597212 [Cited on page 15.]

[34] L. Eyges, "Multiple scattering with energy loss," Physical Review, vol. 74, no. 10, p. 1534, 1948. [Online]. Available: https://doi.org/10.1103/PhysRev.74.1534 [Cited on page 15.]

[35] B. Gottschalk, E. W. Cascio, J. Daartz, and M. S. Wagner, "On the nuclear halo of a proton pencil beam stopping in water," Physics in Medicine & Biology, vol. 60, no. 14, p. 5627, 2015. [Online]. Available: https://doi.org/10.1118/1.3264177

[36] B. Erdelyi, "A comprehensive study of the most likely path formalism for proton-computed tomography," Physics in Medicine & Biology, vol. 54, no. 20, p. 6095, 2009. [Online]. Available: https://iopscience.iop.org/article/10.1088/0031-9155/54/20/005

[37] C. Gianoli, S. Meyer, L. Magallanes, C. Paganelli, G. Baroni, and K. Parodi, "Analytical simulator of proton radiography and tomography for different detector configurations," Physica Medica, vol. 59, pp. 92–99, 2019. [Online]. Available: https://doi.org/10.1016/j.ejmp.2019.03.002 [Cited on pages 15 and 16.]

[38] F. Khellaf, "A deconvolution method to improve spatial resolution inproton CT," 6th Annual Loma Linda Workshop, 2020. [Online]. Available: https://ionimaging.org/assets/talks/ws2020llu-ferielkhellaf.pdf [Cited on pages xv and 15.]

[39] B. Schaffner and E. Pedroni, "The precision of proton range calculations in proton radiotherapy treatment planning: experimental verification of the relation between CT-HU and proton stopping power," Physics in Medicine & Biology, vol. 43, no. 6, p. 1579, 1998. [Online]. Available: https://iopscience.iop.org/article/10.1088/0031-9155/43/6/016/pdf#references [Cited on pages 16, 19, and 20.]

[40] J. H. Chris Constantinou and L. DeWerd, "Tissue characterization phantom model 467." [Online]. Available: https://cspmedical.com/content/102-1492_tissue_phantom_user_guide.pdf [Cited on pages xix and 17.]

[41] C.-W. Cheng, L. Zhao, M. Wolanski, Q. Zhao, J. James, K. Dikeman, M. Mills, M. Li, S. P. Srivastava, X. Q. Lu et al., "Comparison of tissue characterization curves for different ct scanners: implication in proton therapy treatment planning," Transl Cancer Res, vol. 1, pp. 236–246, 2012. [Online]. Available: https://www.researchgate.net/publication/253234733 [Cited on pages xix and 17.]

[42] S. Meyer, F. Kamp, T. Tessonnier, A. Mairani, C. Belka, D. J. Carlson, C. Gianoli, and K. Parodi, "Dosimetric accuracy and radiobiological implications of ion computed tomography for proton therapy treatment planning," Physics in Medicine & Biology, vol. 64, no. 12, p. 125008, 2019. [Online]. Available: https://iopscience.iop.org/article/10.1088/1361-6560/ab0fdf/meta [Cited on page 18.]

[43] H. Jiang, J. Seco, and H. Paganetti, "Effects of hounsfield number conversion on ct based proton monte carlo dose calculations," Medical physics, vol. 34, no. 4, pp. 1439–1449, 2007. [Online]. Available: https://doi.org/10.1118/1.2715481 [Cited on page 19.]

[44] M. Yang, X. R. Zhu, P. C. Park, U. Titt, R. Mohan, G. Virshup, J. E. Clayton, and L. Dong, "Comprehensive analysis of proton range uncertainties related to patient stopping-power-ratio estimation using the stoichiometric calibration," Physics in Medicine & Biology, vol. 57, no. 13, p. 4095, 2012. [Online]. Available: https://iopscience.iop.org/article/10.1088/0031-9155/57/13/4095/pdf [Cited on page 19.]

[45] ICRP, "Report of the task group on reference man, icrp publication 23," 1975. [Online]. Available: https://doi.org/10.1080/09553008514551281 [Cited on page 19.]

[46] I. El Naqa, Machine Learning in Radiation Oncology, I. E. N. D. of Oncolog McGill University Montreal Canada, Ed. Springer, 2015. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-18305-3#about [Cited on pages 21 and 22.]

[47] R. L. Tweedie, K. L. Mengersen, and J. A. Eccleston, "Garbage in, garbage out: Can statisticians quantify the effects of poor data?" CHANCE, vol. 7, no. 2, pp. 20–27, 1994. [Online]. Available: https://doi.org/10.1080/09332480.1994.11882492 [Cited on page 21.]

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/about.html#citing-scikit-learn [Cited on page 23.]

[49] J. Brownlee, "Basics of linear algebra for machine learning," 2018. [Online]. Available: https://mariapilot.noblogs.org/files/2020/11/371769141-Jason-Brownlee-Basics-for-Linear-Algebra-for-Machine-Learning-Discover-the-Mathematical-Language-of-Data-in-Python-2018.pdf [Cited on pages 23 and 25.]

[50] M. Forti, "Técnicas de machine learning aplicadas na recuperação de crédito do mercado brasileiro," Ph.D. dissertation, 2018. [Online]. Available: http://bibliotecadigital.fgv.br/dspace/handle/10438/24653 [Cited on page 25.]

[51] L. Breiman, "Bagging predictors," Machine learning, vol. 24, no. 2, pp. 123–140. [Cited on page 26.]

[52] Y. Freund, R. E. Schapire et al., "Experiments with a new boosting algorithm," in icml, vol. 96. Citeseer, 1996, pp. 148–156. [Online]. Available: https://cseweb.ucsd.edu/~yfreund/papers/boostingexperiments.pdf [Cited on pages 26 and 28.]

[53] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of computer and system sciences, vol. 55, no. 1, pp. 119–139, 1997. [Online]. Available: https://doi.org/10.1006/jcss.1997.1504 [Cited on page 26.]

[54] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," Machine learning, vol. 40, no. 2, pp. 139–157, 2000. [Online]. Available: https://link.springer.com/article/10.1023/A:1007607513941 [Cited on page 27.]

[55] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html?highlight=adaboost#sklearn.ensemble.AdaBoostRegressor [Cited on pages 27, 28, and 41.]

[56] J. H. Friedman, "Stochastic gradient boosting," Computational statistics & data analysis, vol. 38, no. 4, pp. 367–378, 2002. [Online]. Available: https://doi.org/10.1016/S0167-9473(01)00065-2 [Cited on page 29.]

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html#sklearn.ensemble.GradientBoostingRegressor [Cited on pages 29, 33, and 41.]

[58] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, pp. 785–794. [Online]. Available: https://www.researchgate.net/publication/310824798_XGBoost_A_Scalable_Tree_Boosting_System [Cited on pages 29 and 31.]

[59] A. Haghighat, Monte Carlo methods for particle transport. Crc Press, 2020. [Online]. Available: https://www.routledge.com/Monte-Carlo-Methods-for-Particle-Transport/Haghighat/p/book/9780367188054 [Cited on page 35.]

[60] C.-A. Collins-Fekete, S. Brousmiche, D. C. Hansen, L. Beaulieu, and J. Seco, "Pre-treatment patient-specific stopping power by combining list-mode proton

radiography and x-ray ct," Physics in Medicine & Biology, vol. 62, no. 17, p. 6836, 2017. [Online]. Available: https://doi.org/10.1088/1361-6560/aa7c42 [Cited on page 36.]

[61] O. C. to XGBOOST, "Xgboost: Machine learning in Python." [Online]. Available: https://xgboost.readthedocs.io/en/latest/python/python_intro.html [Cited on page 41.]

[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html [Cited on page 41.]

[63] ——, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html [Cited on page 41.]

[64] ——, "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html [Cited on page 42.]