Politecnico di Torino

Master's degree in Petroleum and Mining Engineering

Electromagnetic Exploration Data Transformation into Geoelectrical Models through Clustering and Rescaling Apparent Resistivity Curves

Riccardo Valeri

Supervisors:
Prof. Laura Valentina Socco
Dr. Oscar Ivan Calderon Hernandez

Politecnico di Torino
Academic year 2023/2024

Abstract

The Magnetotelluric (MT) method is a passive geophysical technique used to study the Earth's subsurface, by measuring variations in natural electromagnetic fields. In fact, from this geophysical method is possible to measure electric and magnetic field of the earth due to natural telluric currents, and from these measurements an apparent resistivity curve as a function of frequency is obtained. The apparent resistivity of a multi-layered subsurface reflects at each depth the impact of a sequence of layers with varying resistivities above the measurement point.

Through inversion processes is possible to estimate the resistivity distribution in the subsurface starting from apparent resistivity curves, but these processes are characterized by a series of limitations: non-uniqueness of solutions, non-linearity, etc.

This Thesis is a prosecution of the work done by *Calderon Hernandez O., 2023 [15]* , who described an alternative method to inversion, for rescaling 1D MT apparent resistivity data directly into layered subsurface resistivity models, using a relationship called depth/pseudo-depth rescaling function. To obtain the rescaling function a resistivity model relevant to the data is needed, and if the rescaling function is known for a specific sounding it can be used to rescale other sounding provided that the data are “similar enough” to those used to estimate the rescaling function.

The added value of this study lies in applying clustering algorithms (the three employed are k-means, CURE, and OPTICS) to the apparent resistivity data, trying to reduce the number of rescaling functions needed (and therefore the number of resistivity models required through inversion processes) for the rescaling process, using one rescaling function for each cluster.
In particular, clustering algorithms were employed in a multidimensional space, testing as clustering criteria, combinations of data patterns, related to a physical meaning, that describe the apparent resistivity curves.

In the over 30 tests performed on randomly generated synthetic data, various iterations were carried out on Phyton scripts to determine the most effective clustering algorithm, and the best combination of parameters. The clustering algorithms and the parameters were evaluated by computing the error between the obtained rescaled models and the true models.

The findings on synthetic data showed the benefit of clustering, in fact, it emerged that knowing less than 5% of resistivity models in a dataset, and applying the rescaling process using one depth/pseudo-depth rescaling function per cluster, yields very low errors (avg. error <5%) compared to rescaling all the data using a random model without considering clustering processes (avg. error >20%).

Once the conditions were defined, this model was also tested on a real dataset (COPROD2) consisting of 35 apparent resistivity data. After clustering the data and rescaling them using a single 'reference' depth/pseudo-depth rescaling function per cluster, the results confirmed, with minimum computational costs, the overall trend and the presence of the conductive bodies expected from the inverted models.

Acknowledgements

Vorrei partire ringraziando il Politecnico per questi 7 anni di amore e odio. Troppe sono le immagini che ho in testa ed è impossibile ricordare tutto ma grazie per tutte le sofferenze talvolta ricompensate, per le amicizie, le esperienze e soprattutto per quello che mi ha dato sia dal lato didattico che da quello umano.
Grazie quindi alla Prof. Socco per l'opportunità di questo lavoro di tesi, per ciò che ho appreso da lei in questi mesi e grazie soprattutto per avermi fatto appassionare al mondo della geofisica.
Grazie al mio supervisor, Oscar. Lo ringrazio soprattutto per la pazienza con cui mi ha spiegato le cose, con la calma con cui mi ha fatto notare gli errori, e soprattutto per la sua idea senza la quale tale progetto non avrebbe mai avuto luogo.

Grazie a mio padre e mia madre, sempre sin dal primo giorno del primo anno di questa esperienza a credere in me, essere a conoscenza delle mie capacità e spronarmi, nonostante le premesse non fossero tanto rosee. Grazie per questo, e per il loro aiuto sia morale che economico.
Grazie al resto della mia famiglia, i miei 3 cugini, i miei zii, mia nonna Valentina (e gli altri da lassù), mia zia Elvira, e i parenti tutti. Grazie per tutto il supporto.

Grazie a Giulia, per essere entrata nella mia vita da più di tre anni ed aver deciso insieme a me da più di due, di intraprendere questa strada impervia che è la nostra relazione. Grazie perché i km di distanza che ci separano nei momenti in cui siamo lontani, non si sono mai fatto sentire e anzi mi sei stata sempre vicino nel bene o nel male. Grazie per comprendermi e sopportarmi ogni singolo giorno.

Grazie a Nicolò, Enrico, Silvia e Tomas per essere quelli che mi conoscono da di più, per avermi sopportato e spronato in tutti questi anni ognuno a suo modo. Grazie soprattutto per la vostra costanza e per aver creato insieme un rapporto indissolubile di sana amicizia che spero continui nel tempo.

Grazie a Giovanni, dapprima conoscente, a collega, ad amico ed infine coinquilino. Impossibile racchiudere tutto quello che abbiamo passato ma grazie per questi 4 anni, dalla scelta della magistrale assieme, al continuo vedersi solo per videochiamata (do something jovani), alle feste rosselli, alle partite, alla convivenza. Grazie davvero.

Grazie ad Antonio, per essere stato una bella sorpresa avvenuta in un modo completamente casuale ma che ha portato a una bella amicizia. Grazie per esserci stato in questo ultimo periodo e per tutti i momenti ludici assieme.

Grazie a Marco, Davide, Peppe, Alessio, Mattia per essere stati per 4 anni, chi prima, chi dopo la mia casa a Torino. Rosselli.
Grazie ad Alessandro, Emanuele, Vincenzo e Ric, casa futili sempre il punto di riferimento di tante serate tra una pagliacciata e l'altra.
Grazie al gruppo Fantaprank e tutte le altre amicizie di Torino, nuove e vecchie nessuno escluso.

Grazie ai ragazzi del gruppo Estate, per come in questi anni siamo stati in grado di ricreare un gruppo che non esisteva, per tutte le estati, le vacanze e le esperienze passate insieme.
Grazie di tutto. Riccardo.

Inscription

A chi c'è sempre stato,
da vicino o da lontano,
prima o dopo,
in un modo o nell'altro,
Grazie.

Index

List of Figures

List of Tables

List of Equations

Chapter 1

1. Introduction

In geophysics, resistivity is a key factor for studying the subsurface properties of the Earth. Different materials (such as rocks, minerals, soils, and fluids) have distinct resistivity values. By measuring resistivity variations underground, geophysicists can infer valuable information about the composition, structure, and presence of subsurface features like groundwater, mineral deposits, oil, gas, and geological formations. In terms of values, as shown in Fig.1, higher resistivities may correspond to rocks such as granite, basalt, or quartz, while lower resistivities could indicate the presence of shales or clays.

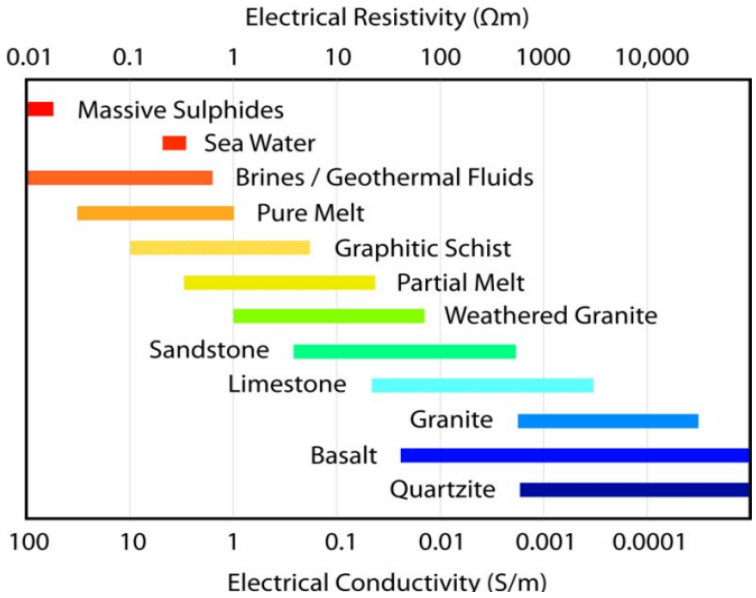


Figure 1 - Electrical resistivity (Ωm) and conductivity (S/m) of different rocks and formations [1]

Since the early 20th century, electromagnetic methods (EM) have been extensively employed for subsurface exploration, aiming to map the electromagnetic characteristics of subsurface materials. In recent years, EM originally designed for deeper purposes like mining, hydrocarbon exploration, or crustal studies have been adapted for shallow applications such as environmental studies or geotechnical investigations (*Pellerin, 2002*) [2].

The most commonly used techniques for obtaining underground resistivity models, starting from geophysical data, are inversion processes. By utilizing data acquired through diverse EM resistivity measurement methods such as Magnetotelluric (MT), Electrical Resistivity Tomography (ERT), and other geophysical surveys, inversion processes aim to reconstruct resistivity models of the subsurface, by achieving the best possible match (minimum misfit) between observed and simulated resistivity data.

However, inversion processes are characterized by some limitations:

- → Non-uniqueness**:** There are multiple geological models that can explain the observed data equally well, leading to non-unique solutions.
- → Non-linearity: Many underlying geological processes may be nonlinear, making data inversion challenging and requiring the use of complex iterative algorithms.
- → Computational complexity: Some inversion algorithms may require significant computational resources and long computation times, especially when dealing with large datasets or complex models.

In recent years, there has been ongoing research into new techniques aimed at directly estimating geological models of the subsurface, starting from the measurements and bypassing the need for an inversion process. For instance, *Florio (2018)*[3] introduced a method to determine the depth of the basement by utilizing gravity or pseudo-gravity measurements, employing a linear iterative rescaling approach. Another example, is the one proposed by *Socco et al. (2017)* [4]*,* a methodology that directly extract time average body waves velocity models of the subsurface starting from DCs of surface waves by applying a relationship between the surface waves wavelength and the investigation depth of the time-average velocity model.

Based on these examples, the main idea of this Thesis is to find an alternative method to inversion processes, minimizing their contribution in obtaining underground resistivity models, starting from MT data.

Indeed, this study will describe a methodology, that includes:

- → Rescaling apparent resistivity data into resistivity models using a relationship between the depth of the models and the pseudo-depth of the apparent data (depth/pseudo-depth rescaling function), and
- → Clustering apparent resistivity data based on mathematical parameters, in order to rescale an entire dataset into resistivity models using a limited number of rescaling functions (one per cluster), and thus reduce the needed known resistivity models, which have to be provided from inversion processes.

The layout of the thesis is structured as follows:

- → Chapter 2: In Chapter 2, the physics underlying the Magnetotelluric method will be described. It starts from the Maxwell/Heltmoltz equations and progresses to the formula for apparent resistivity for MT and that of pseudo-depth (through a data transformation of the frequency called the Niblett-Bostick method). Subsequently, a study developed by *Calderon Hernandez et al. (2023) [15]*, which explains the theory of rescaling resistivity data into

models without the need for inversion, will be presented, and the results obtained from a synthetic test to understand the validity and limitations of the research will be shown.

→ Chapter 3: Subsequently, in Chapter 3, the theory of clustering will be explained in detail. It includes a general explanation of the three major categories and then, in particular, a description of the three clustering algorithms (one per category) that have been used in the project.

→ Chapter 4: Chapter 4 will discuss the methodology employed in the development of the project. This will be explained through synthetic data examples, encompassing the generation of synthetic 1D models, the computing of apparent resistivity data, the explanation of the clustering criteria and the clustering methodology, and the cross-rescaling of the clustered data to assess the quality of clustering results, investigating the best combination of criteria and the optimal algorithm.

→ Chapter 5: In Chapter 5, all the main final results derived from the application of this methodology will be presented. These include the final outcomes, so the rescaled models, obtained from two synthetic datasets (200 and 1000 data), and in the end, those obtained from a real dataset. For the real ones, a comparison will also be made with models produced by an inversion to highlight the potential and validity of the model for future studies.

→ In Chapter 6, all the conclusions and findings from this thesis work will be summarized.

Chapter 2

2. Rescaling MT Apparent Resistivity Data

This chapter aims to go deeply into resistivity as a central concept in geophysics. It will explore first the fundamental principles of the apparent resistivity and the Magnetotelluric (MT) method employed for its measurement. Furthermore, a research to obtain geoelectrical models of the subsurface from apparent resistivity curves, through the application of a depth/pseudo-depth rescaling function, will be outlined. As will be explained, this function is the relationship between the depth of the cumulative resistance and the depth of the apparent resistance. In the end to reduce the number of rescaling functions needed, clustering techniques will be introduced, and how clustering could be implemented to improve this work will be discussed.

2.1 Apparent Resistivity and MT

The apparent resistivity (in a multi-layer scenario) reflects, at each point in the subsurface, the impact of a sequence of layers with varying resistivities above the measurement point. It can be directly calculated from the ground-acquired data and, "*since apparent resistivity is a normalizing procedure with physical significance*" [5], it is possible to formulate new definitions based on the methodology employed for data acquisition.

The apparent resistivity curves used in this study are derived from the Magnetotelluric (MT) method. The MT method is a passive electromagnetic (EM) exploration technique, that records orthogonal components of both electric and magnetic fields on the Earth's surface. [6]
As illustrated in Fig.2, above the 1 Hz threshold, MT fields largely originate from thunderstorms across the globe, emitting fields irradiated across large distances by lightning. Below the 1 Hz range, the majority of the signal arises from current systems within the magnetosphere induced by solar activity.
The necessary instrumentation for conducting these measurements includes magnetometers designed for the relevant frequency range, sets of electrodes positioned at appropriate intervals to detect variations in the electric field, as well as amplifiers, filters, and digital recording and processing systems that enable the signals to be captured and analysed. The method allows to explore various depths, from 0 to tens of km, without the need for artificial power sources and with minimal environmental impact. [7]

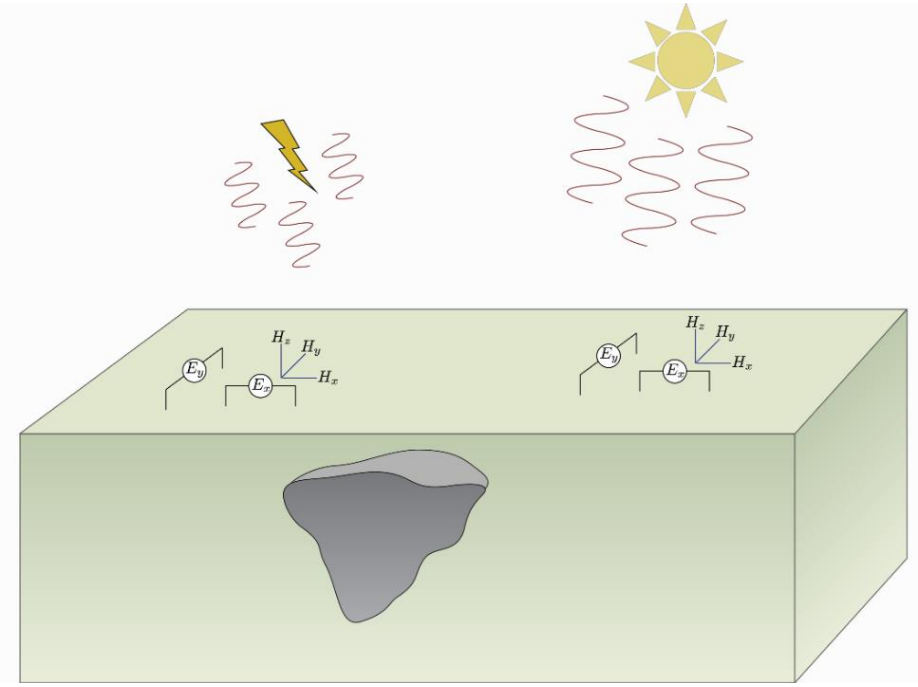


Figure 2 - MT method, natural sources of EM fields: high-frequency signals (>1 Hz) from thunderstorms, or low-frequency (<1 Hz) from solar wind activity. [6]

To obtain an equation for apparent resistivity in the Magnetotelluric method, we have to start with Maxwell's theory, considering the propagation of plane waves (like those produced by solar activity in MT) within a medium.

In particular the starting point are the vector Heltmoltz equations for the electric (E) and magnetic (H) fields (Eq. 1 and 2), that are constructed from the Faraday's law and the Ampere-Maxwell one. [8]

$$\nabla^2\mathbf{E} + \mathrm{k}^2\mathbf{E} = 0 \tag{1}$$

$$\nabla^2\mathbf{H} + \mathrm{k}^2\mathbf{H} = 0 \tag{2}$$

where k is the wavenumber (m^{-1}). These equations are simplified according to *D. J. Griffiths, 1999* [9], which explains that the EM fields induced by plane waves are transversed to the direction of propagation, so xy is the polarisation plane. For these conditions a generical solution can be obtained for both E and H (Eq. 3 and 4),:

$$\mathbf{E} = \mathbf{E_0^-}\mathrm{e}^{\mathrm{i}(\mathrm{kz}-\omega\mathrm{t})} + \mathbf{E_0^+}\mathrm{e}^{-\mathrm{i}(\mathrm{kz}+\omega\mathrm{t})} \tag{3}$$

$$\mathbf{H} = \mathbf{H_0^-}\mathrm{e}^{\mathrm{i}(\mathrm{kz}-\omega\mathrm{t})} + \mathbf{H_0^+}\mathrm{e}^{-\mathrm{i}(\mathrm{kz}+\omega\mathrm{t})} \tag{4}$$

From this solutions, applying boundary conditions (such as $\mathrm{E}(\mathrm{z}\to-\infty,\omega)=0$ and $\mathrm{E}(\mathrm{z}=0,\omega)=\mathrm{E}_0$) and defining the wavenumber as:

$$\mathrm{k} = \alpha - \mathrm{i}\beta \tag{5}$$

Is possible to derive these solutions for E and H (Eq. 6 and 7):

$$\mathbf{E} = \mathbf{E_0}\mathrm{e}^{\beta z}\mathrm{e}^{\mathrm{i}(\alpha\mathrm{z}-\omega\mathrm{t})} \tag{6}$$

$$\mathbf{H} = \mathbf{H_0} e^{\beta z} e^{i(\alpha z - \omega t)} \quad (7)$$

In both of the previous expressions, the second part characterizes a simple harmonic motion with its own phase shift, where the real part of the wavenumber α, "*determines the wavelength and the velocity of the plane wave"*[8]. Instead the first one, includes the attenuation factor β of the electromagnetic wave. [10] Through this factor, it is possible to define the depth at which the wave will be attenuated by a factor 1/e. This distance is called the skin depth and its quasi-static approximation ($\epsilon\omega << \sigma$, 1/e=0.369) is defined as follows:

$$\delta = \frac{1}{\beta} = \sqrt{\frac{2}{\omega\mu\sigma}} \quad (8)$$

where $\sigma\ (\mathrm{S/m})$ is the conductivity and $\mu\ (\mathrm{H/m})$ is the permeability.

Then, from Eq. 6 and 7, by applying a quasi-static approximation, and knowing that the surface impedance Z_{xy} (Ω) is defined as the ratio between the horizontal electric field, E_x, and the orthogonal horizontal magnetic field H_y[8]:

$$Z_{xy} = \frac{\mathbf{E_x}}{\mathbf{H_y}} \quad (9)$$

the apparent resistivity ρ_{app} (Ω·m) for MT is defined as the ratio between the square of the surface impedance (in absolute value), and the product between the angular frequency ω (s^{-1}) and the magnetic constant μ_0 ($4\pi\times10^{-7}$) (H/m), that is the permeability of free space [11][12]:

$$\rho_{App} = \frac{|Z_{xy}|^2}{\omega\mu_0} \quad (10)$$

The measuring domain for which apparent resistivity data are acquired is frequency, but starting from Eq. 8 and using the approach known as Niblett transformation (1/e=0.5), which directly estimates a point in depth for a given measured apparent resistivity data as function of the measuring domain, is possible to move into depth domain using the concept of *pseudo-depth* $\bar{z}(f)$, that is defined as [13]:

$$\bar{z} = \sqrt{\frac{\rho_{App}(f)}{\omega\mu_0}} \quad (11)$$

The simplest approach for MT surveying is to invert individual sounding using a 1D layered resistivity model as reference of the subsurface resistivity distribution. Even though 2D and 3D inversion approaches are available (like the work conducted by *Jones, A., G., 1993* [14]), 1D inversion is often used to quickly generate initial models for further 2D/3D refinements. In those cases where the subsurface can be reasonably described as a layered system and the sounding spatial density is low, 1D surveys might be an acceptable approximation. 1D inversion is highly non linear and many methods have been envisaged in the past decades to transform the apparent resistivity data in proxy of 1D models without inverting the data.

One of these concerns the rescaling of apparent resistivity data into geoelectrical models without the need for inversion, and it will be described below.

2.2 Data rescaling into models

Following the research of *Calderon Hernandez et al. (2023) [15]* for MT data, it is possible "*to transform the data measured from a resistivity survey directly into a model, developing a rescaling function, using a 1D cumulative resistance model*".

Instead of considering a layered medium and trying to derive local parameters (resistivity and layer thickness) through inversion, this method, aiming to find a relationship between measured data and models, considers cumulative resistivity models that can be associated with apparent resistivity data (since they are also cumulative). This conversion process of layered resistivity models into cumulative resistivity models is done using the concept of *equivalent layers*, that is, the method of converting multiple horizontal layers into a single effective layer that exhibits a uniform resistivity value[15]. It is explained by the following equation:

$$\rho_{eq} = \sqrt{\frac{T}{S}} \tag{12}$$

where T ($\Omega \cdot m^2$) is the *transverse unit resistance* and S (Ω^{-1}) is known as *longitudinal unit conductance,* defined by:

$$T = \rho_1 z_1 + \rho_2 z_2 + \cdots + \rho_n z_n = \sum_{i=1}^{n} \rho_i z_i \tag{13}$$

$$S = \frac{z_1}{\rho_1} + \frac{z_2}{\rho_2} + \ldots + \frac{z_n}{\rho_n} = \sum_{i=1}^{n} \frac{z_i}{\rho_i} \tag{14}$$

By using Eq.12 a cumulative resistivity model is obtained from a layered one. In Fig. 3 are shown, the acquired apparent resistivity data (in green) as a function of pseudo-depth, the stratified model obtained through inversion (in blue), and the cumulative model obtained from the layered one (in orange).

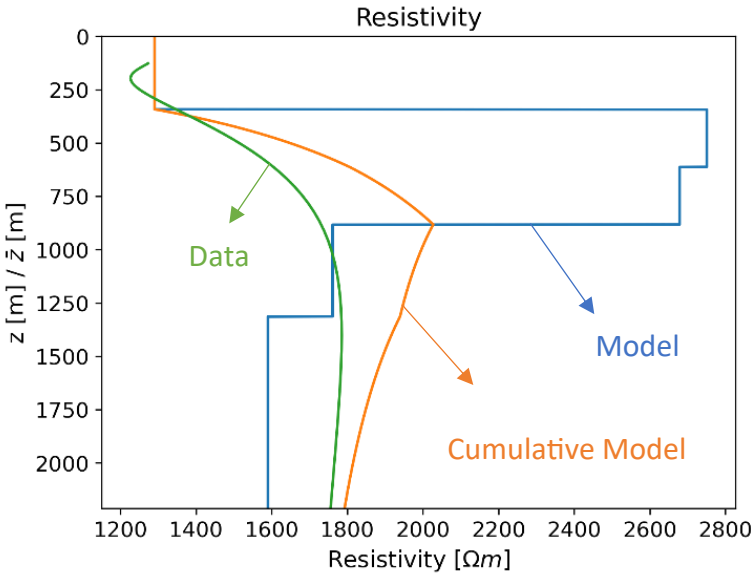


Figure 3 - Comparison between cumulative models (orange) and measured data (green) in resistivity domain. In blue are also shown the 1D layered models from which the cumulative are computed. Models are displayed as functions of depth z(f) while data of pseudo-depth $\bar{z}(f)$.

From a physical point of view, these cumulative resistivity models incorporate the cumulative effect of all the layers above a given point in depth, showing the trend of resistivity similarly to how it is done by geophysical measurements, instead of the typical layered models used to describe the subsurface [15].

At this point, to find a function that allows transforming the data into models, it is necessary to establish a relationship between the depth at which the models (cumulative) are defined and the pseudo-depth of the apparent data. However, this cannot be done in the resistivity domain because one point in resistivity can be related to more than one point in depth (non-unique relationship). For this purpose, both the model and the data are transformed into the cumulative resistance domain, ensuring that the uniqueness condition is met (whereby each cumulative resistance value corresponds to one and only one pair of depth/pseudo-depth values). Furthermore, as illustrated in Fig. 3, in the resistivity domain the two trends (of the cumulative model and measured data) follow a similar path (being the cumulative resistivity model closely aligned with the physical phenomena), but discrepancies are significant.

However, it is found that in resistance domain discrepancies between cumulative model and data are lower (Fig. 4). The transition from the cumulative resistivity data (Ω·m) to the cumulative resistance ones (Ω·m^2) necessitates an integration process:

$$R(z) = \int_0^z \rho\ (z)dz \tag{15}$$

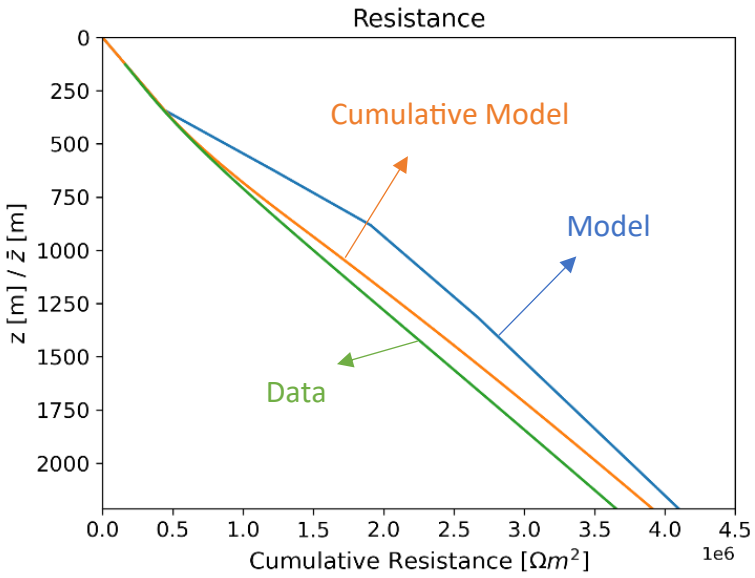


Figure 4 - Comparison between cumulative models (orange) and measured data (green) in resistance domain. As illustrated in resistance domain discrepancies between data and cumulative models are much lower than in resistivity domain. In blue are also shown the layered models from which the cumulative are computed. Models are displayed as functions of depth z(f) while data of pseudo-depth z̄(f).

In the resistance domain, the gap between depth and pseudo-depth is a Δz for a given value of resistance. In the left plot of Fig. 5, two Δz examples are depicted for two different cumulative resistance values. These Δz are defined, for each value of cumulative resistance, by:

$$\Delta z(R) = z(R) - z(R_{app}) \qquad \text{when } R \approx R_{app} \tag{16}$$

By approximating the Δz function to a polynomial regression, a rescaling function that enables the transformation of data into models in the resistance domain is retrieved (Fig.5 right).

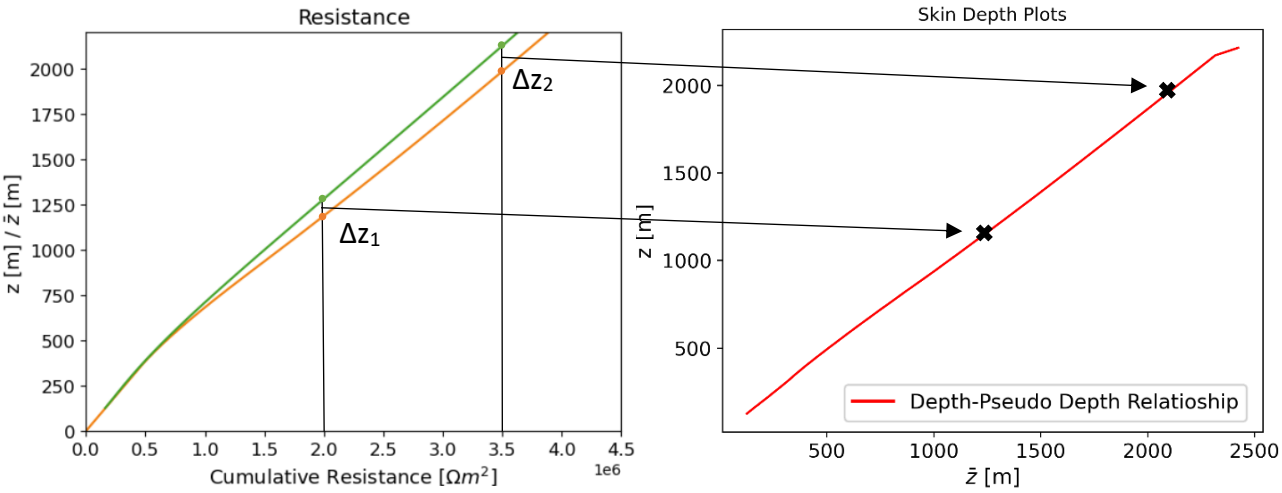


Figure 5 - Depth/pseudo-depth rescaling function (right). The relationship is obtained between the depth of the model cumulative resistance and the pseudo-depth of the data, for a given fixed value of resistance. In the left plot are shown two examples (Δz_1 and Δz_2, respectively for 2 and 3.5 $\Omega \cdot m^2$) that explain how the relationship is generated

By applying the depth/pseudo-depth rescaling function to the apparent cumulative resistance data, it is possible to retrieve cumulative models defined as rescaled models. After that resistance data have been rescaled, next step is to apply a numerical derivative to pass from resistance to resistivity domain, obtaining rescaled apparent resistivity cumulative models (Fig. 6 left).
Last step is to pass from cumulative to layered models using the inverse formula of Eq.12, to retrieve 1D layered resistivity models (Fig. 6 right)[15].

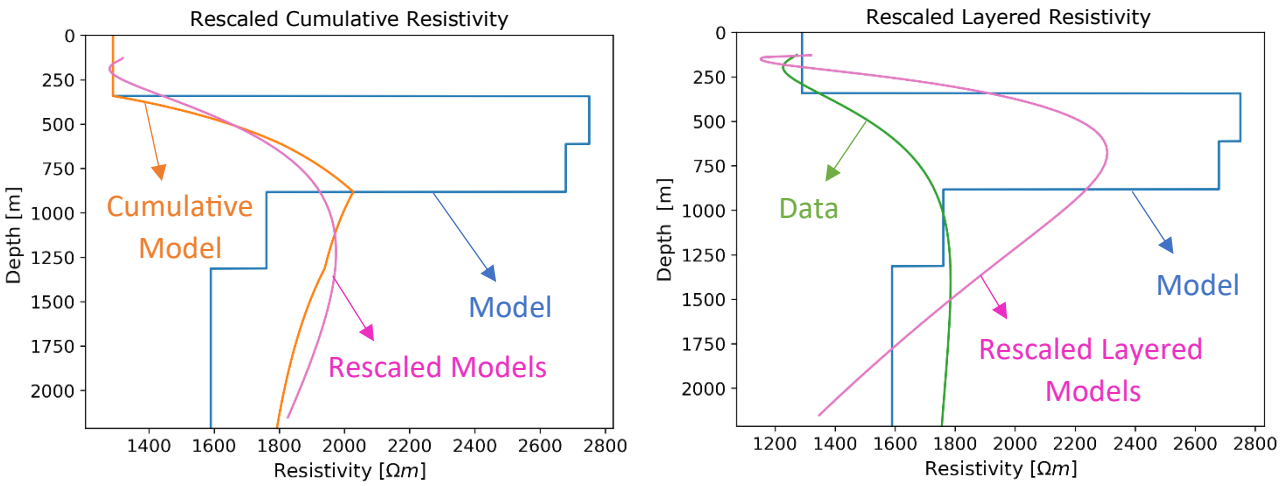


Figure 6 - Cumulative (left plot) and layered (right plot) rescaled resistivity models obtained by applying the depth/pseudo-depth rescaling function to measured data.

As visible in the right plot of Fig.6, the layered 1D resistivity model (blue line) and the layered resistivity model obtained by rescaling apparent resistivity data (pink curve) have a similar trend, since the rescaled layered models obtained are smoother versions of the 1D stratified models.

2.2.1 Applicability and limits of the rescaling method

Once the depth/pseudo-depth rescaling function approach is described, the next aim is make this method feasible by using the rescaling function obtained for one model to rescale an entire dataset. This means having a dataset where, for each apparent resistivity curve, there is a corresponding resistivity model, then the depth-pseudo-depth rescaling function is derived from one 'reference' selected model, and this rescaling function is applied to all experimental curves to retrieve the whole set of models without inversion.
To do so, a test was performed simulating 20 synthetic MT data obtained from randomly generated stratified models. To convert 1D resistivity models to MT apparent resistivity data, the Python routine "*empymod*" forward modelling (whose functioning will be explained in Chapter 4) by *Werthmüller, 2017* [16] was used. In Fig.7 both 1D layered models generated and apparent data simulated are displayed with same colours.

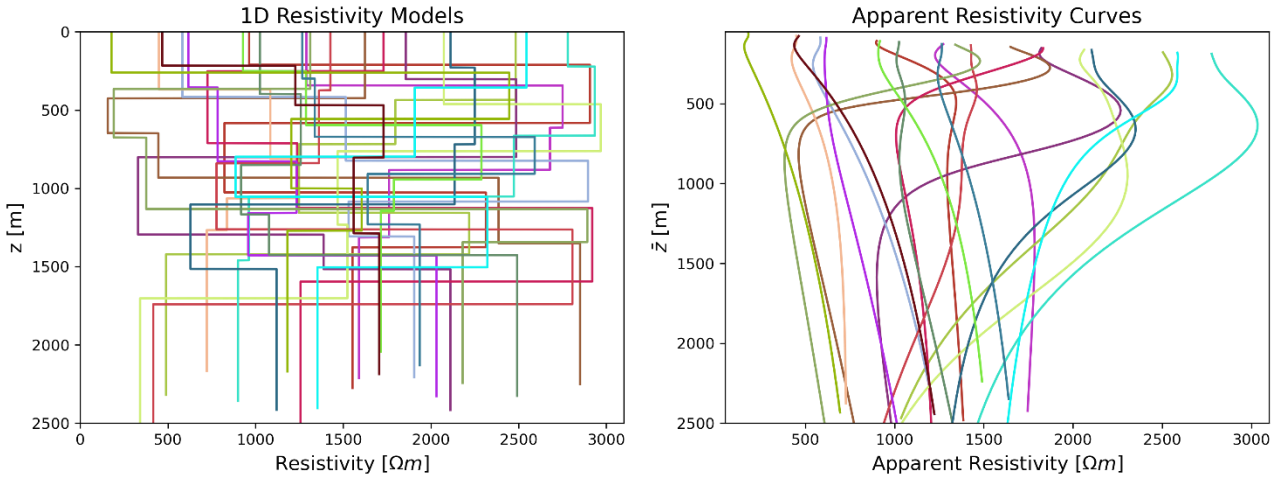


Figure 7 - 20 apparent resistivity curves dataset (right) defined as a function of pseudo-depth, obtained by empymod routine from the 20 randomly generated models (left).

To conduct an extensive test, after generating the 1D resistivity models, a depth/pseudo-depth rescaling function for each model/data was obtained. After that multiple iterations were performed, considering in each iteration a single rescaling function as the 'reference' one of the dataset, and using it to rescale the entire dataset into models.
In Fig. 8 are represented the depth/pseudo-depth rescaling functions obtained for each data/model and used one at a time to rescale the data.

To test the efficiency of the method, since the data are synthetic and therefore the generated models are true, the rescaled models obtained at each iteration were compared with the true ones. This was done by calculating the error between true models and those estimated by the method for

each of the rescaling functions. This error is determined using a Python function *(error_depth)* that calculates (in the cumulative resistance domain) the difference between the rescaled models and the true ones at each depth, and divides it by the value of the true model, producing a measure of error that is then converted into a percentage.

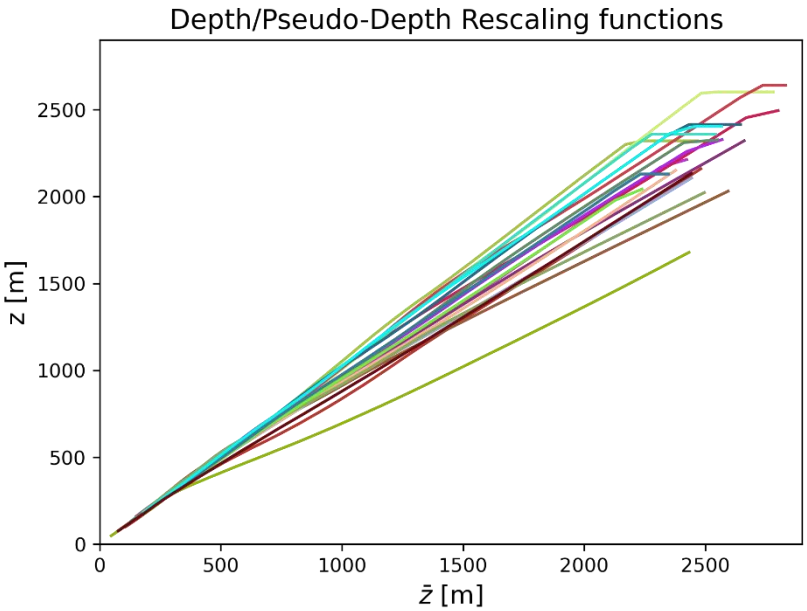


Figure 8 - Set of 20 depth/pseudo-depth rescaling functions, related to the data/models of Fig.7.

At this point, it is possible to represent the errors found in the form of box-plots (Fig. 9), where each box represents the errors produced by each rescaling function, it is delimited by the limit errors, and in each of them, is represented the average error value produced by the rescaling with an orange horizontal line.

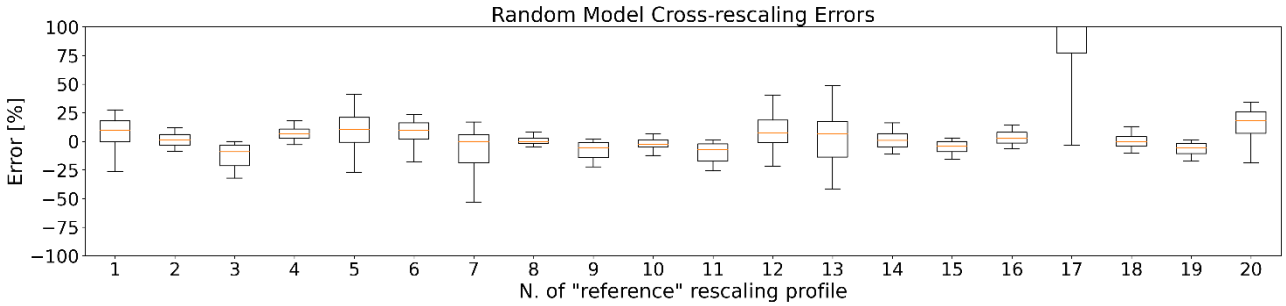


Figure 9 - Error box plot computed by cross-rescaling the whole dataset using a single random depth/pseudo-depth rescaling function. Avg. error is around 18%, with a peak value of 250% (model 17).

As visible from the graph, in a small dataset consisting of 20 curves (Fig. 7), huge errors can be reached, touching 250% (using model 17 as a 'reference model' to rescale the whole dataset), with an average error nearing 18%. From this conducted test, the limitations of the rescaling method emerged, indeed obtaining good results depends on the selection of the reference model used to obtain the reference depth/pseudo-depth rescaling function and then rescale the entire dataset.

As shown in Fig. 9, choosing models like No. 8 or No. 15 yields better results in terms of error (average error around 1/2%), but choosing others like No. 13 or No. 17 leads to significantly worse results with average errors above 20%. Moreover, considering that these errors were obtained with only 20 curves, huge errors are expected from a larger dataset.

To this end, the added value of this Thesis is to find a method to use, considering a large dataset of apparent resistivity curves, only a small amount (since only with one is not feasible) of depth/pseudo-depth rescaling functions to rescale the whole dataset. This means that a large amount of apparent resistivity curves could be rescaled also knowing a limited number of models (obtained a priori or through inversion).

Clustering apparent resistivity curves, in fact, could represents the solution to this problem, trying to use one single 'reference' depth/pseudo-depth rescaling function for each single cluster. Clustering methods are explained in detail in the next chapter.

Chapter 3

3. Clustering methods

The primary focus of this study revolves around *clustering* apparent resistivity curves based on their mathematical properties (such as maxima, minima, gradients etc. that will be explained in the following chapter), to employ a single 'reference' rescaling function for depth/pseudo-depth (where depth represents model depth and pseudo-depth corresponds to the data) for each cluster. This approach is pursued to use these set of functions to rescale the apparent resistivity data acquired from MT surveys, and to obtain geoelectrical models of the subsurface for an entire dataset while having access to a limited number of models.

The use of a single function per cluster is strategic, as it helps to reduce the requisite number of models as happens in real-life scenarios. In these cases, available models, can be obtained through inversion processes, but their limitations are one of the reasons for which this work was designed.

Clustering is a set of techniques that organize data into distinct groups called clusters. These clusters are formed by grouping together data elements that have similar characteristics or features of interest (which can be defined as similarities), with each other than with elements in different clusters [17]. Based on this definition, high within-cluster similarity and low inter-cluster similarity are criteria to assess the quality of the clustering. Dissimilarities and similarities are evaluated by considering the feature values that describe the objects, frequently utilizing distance metrics. [18]

Clustering is applied to many fields: from machine learning to biology, from social sciences to finance and to geophysical data, facilitating resource exploration.
There are three main categories of clustering algorithms, and each of them is characterized by different criteria to define a cluster and its own advantages or disadvantages. These categories are:

- → Partitional Clustering
- → Hierarchical Clustering
- → Density-Based Clustering

Among these categories, the most effective clustering method for the dataset in question should be determined by considering aspects such as the nature of the clusters, dataset attributes, the presence of outliers, and the quantity of data points.

These three categories will be analysed in detail in the following sections. For each of them, the methodology used during the testing phase of the project will be outlined.

3.1 Partitional Clustering

Partitional clustering is the most widely used category of clustering, as it can be applied to various types of data across different domains, such as multidimensional data, which includes cases like ours where various mathematical parameters are considered for creating clusters of apparent resistivity data, with each parameter adding an extra dimension.

In this category data items are divided into distinct groups that do not overlap, because by definition "*no object can be a member of more than one cluster, and every cluster must have at least one object*". [17] So, given a set of unlabelled data, a label is assigned to each data (Fig.10) and the total number of labels must be decided by the user. In fact, a constraint of this category is that, the number of clusters 'k' (set a priori), must be found by the algorithm through an iterative process based on the distance to a centroid. Algorithms that belong to that category are non-deterministic so they could give different results with same data. An example and also the most used clustering algorithm is K-means.

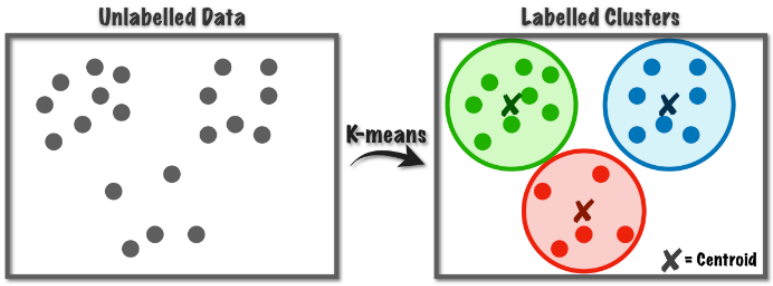


Figure 10 - Partitional Clustering example, k-means application where 3 different labels are applied to unlabelled data and one centroid is defined for each cluster.[19]

3.1.1 K-means

The K-means algorithm starts with the user specifying the number 'k' of clusters to identify. A centroid is defined for each of these clusters, initially positioned randomly within the dataset, and each data point is then assigned to the closest centroid. [20] After that, the algorithm automatically calculates the SSE (Sum of the Squared Euclidean distance) summing the contributions of each point to its closest centroid. The SSE between two points (x and y) in a n-dimensional space is defined as:

$$\mathrm{SSE}(x,y) = \sum_{i=1}^{n} (x_i - y_i)^2 \tag{17}$$

This distance is a measure of an error and have to be minimized through several automatic iteration steps by the algorithm. At each step, centroids are moved randomly in the dataset trying to obtain at each iteration a lower SSE value, recalculating it with Eq.17 till 'centroids convergence' that happens for the lowest value of SSE, and, achieving this condition final clusters are obtained.[17] The entire algorithm overview is displayed in Fig.11.

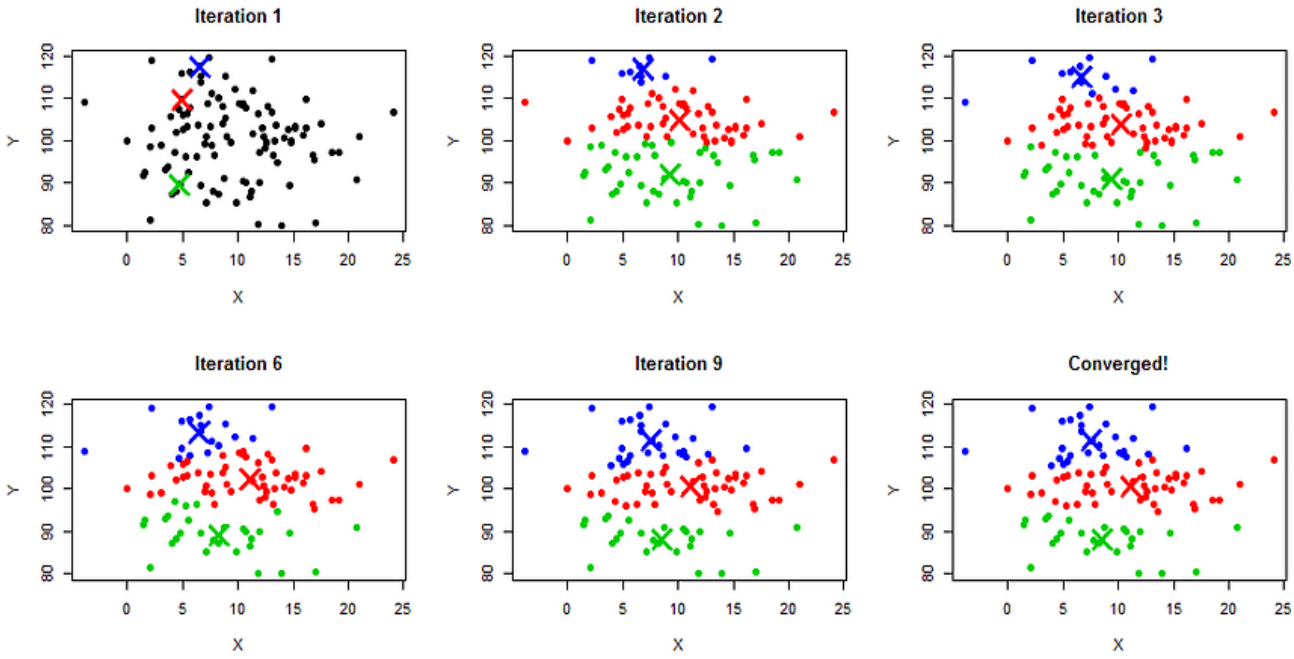


Figure 11 - K-means algorithm overview applied on a dataset formed by 50 items, 9 iteration steps from the first random centroids allocation to centroids convergence. The algorithm moves the centroids and recalculates the SSE at each step.[22]

Advantages of K-Means:

→ Simplicity and speed**:** K-Means is simple, easy to implement, and computationally efficient. It works well even with spherical clusters.

→ Versatility**:** It is versatile and can be applied to a wide range of data types and clustering problems.

→ Guaranteed solution**:** K-Means always guarantees convergence to a solution ('centroids convergence'), also if data items are totally different and so tough to be clustered together.

Disadvantages of K-Means:

→ Sensitivity to initial centroids**:** It is highly sensitive to the initial random placement of centroids by the algorithm (that cannot be modified by the user) which might result in different outcomes for different initializations (non-deterministic).

→ Dependence on 'k'**:** The choice of the number of clusters (k) can significantly impact the clustering result. Selecting an incorrect 'k' value might lead to poor clustering.

→ Vulnerability to outliers**:** Outliers in the dataset can influence the initial position of centroids, leading to suboptimal clusters.

→ Inability to handle complex data**:** K-Means struggles with non-linear clusters, clusters of different sizes, and varying densities. It assumes clusters to be spherical, which might not reflect the actual data distribution accurately.

3.1.2 Determination of the best 'k'

After understanding how k-means works and its pros and cons, it is important to focus on the number of clusters to set before running the algorithm. There are two practical methods that could be performed in series to evaluate the best 'k' for the analysed dataset:

→ Elbow method: In this method the SSE will be measured for an increasing k, so, multiple iterations of the k-means algorithm, incrementing 'k' with each run, and recalculating the SSE at each step, have to be performed. During the process, as the number of clusters increases, the SSE will decrease. However, there will be a specific point on the SSE vs number of clusters curve, known as the "elbow point," where the curve starts to bend (as illustrated in Fig.12, where at k=3 the gradient of the curve changes noticeably). This point represents a balance between SSE and the optimal number of clusters, denoted as 'k'. The value of 'k' corresponding to this point is selected as the optimal number of clusters. [17]

Figure 12 - Elbow method applied to a dataset formed by 10 items, the 'elbow point' is found for k=3 [13]

→ Silhouette coefficient: An alternative methodology to the elbow is the determination of the silhouette coefficient. It is an average value of the entire dataset, calculated based on two principles; 1) how close a point is to other points in the same cluster (cohesion) and 2) how far it is to points that belong to the others (separation). It is defined for a generic point 'i' as:

$$S(i) = \frac{b(i)-a(i)}{\max [a(i),b(i)]} \tag{18}$$

where 'a' corresponds to the mean distance between 'i' and all other data points within the same cluster, and 'b' the mean distance between 'i' and all the other data points that belong to other clusters. It goes from -1 (worst value) and 1 (best value), and it is recalculated, as the elbow, incrementing the number of clusters, and the optimal value of 'k' is found at the highest silhouette coefficient [17][23]. In Fig.13, there is an example applied to the same dataset of Fig.12 that shows the same result of the elbow method, k=3.

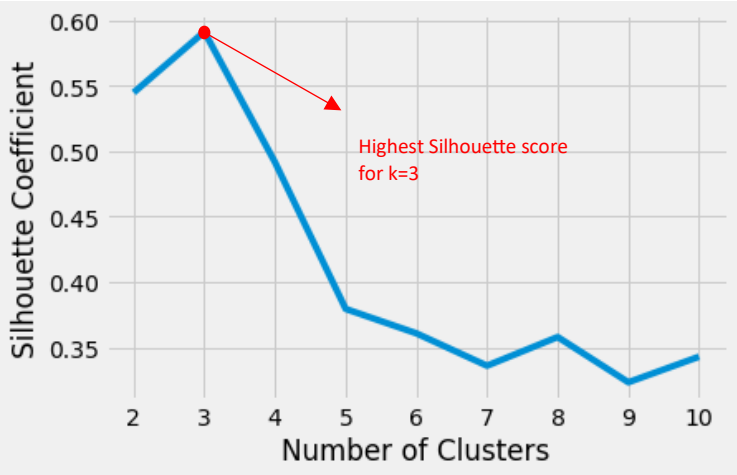


Figure 13 - Silhouette coefficient calculated for the same dataset of elbow method, as expected, the highest score is obtained for k=3 [13]

3.2 Hierarchical Clustering

Hierarchical Clustering is considered one of the most suitable clustering algorithms for handling large datasets, particularly when dealing with hundreds or thousands of data points. It excels over Partitional Clustering in outlier detection, identifying data that do not conform to any specific cluster and are incorrectly assigned to one.

In this second category, cluster assignments are determined by constructing a Hierarchical structure or tree-like representation of the data, also known as 'dendrogram', for which the X-axis represents individual objects that remain distinct, while the Y-axis indicates the distance at which clusters combine. There are two approaches as illustrated in Fig.14: *Agglomerative* (or bottom-up) and *Divisive* (or top-down). In the agglomerative approach, each data point starts as its own cluster and gradually merges at each step until a single cluster is formed, while in the divisive approach, there is the opposite process starting with a single large cluster and dividing it at each step. [21][24] The user must also specify the desired number of clusters 'k' to obtain, so the level at which cut horizontally the 'dendrogram'. An example of this category is CURE.

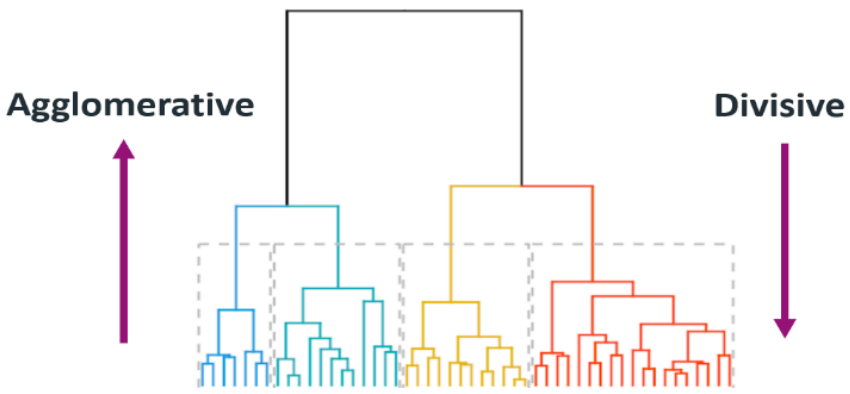


Figure 14 - Hierarchical Clustering, representation of the "dendrogram". Agglomerative approach from the bottom to the top and Divisive approach from the top to the bottom. Dashed lines represent the 'k' number of clusters at which cut the dendrogram (in this case k=4) [25]

3.2.1 CURE

CURE stands for Clustering Using REpresentatives and it is a Hierarchical Clustering algorithm that uses partitioning. The first step of CURE is to pick random samples (partitioning) in the dataset, and form first initial clusters. At this point, a number 'c' of scattered points (as dispersed as possible) are selected for each cluster to represent them. These points are then shrunk towards the centroid (of the cluster for which they are representatives) of a fraction '*α*' (usually 20%). [22]
The distance between each representative point and the corresponding centroid is defined as '*dmin'*, and this parameter is used for the hierarchical merging. Indeed, each data point that does not already belong to a cluster and, has a d<dmin with 'd' (distance from a representative point), is merged into the cluster. Following that methodology, two separated clusters could be merged if their related representative points have d<dmin. At each merging step, new points are selected as representative points, and the 'dmin process' is repeated. [27][28]

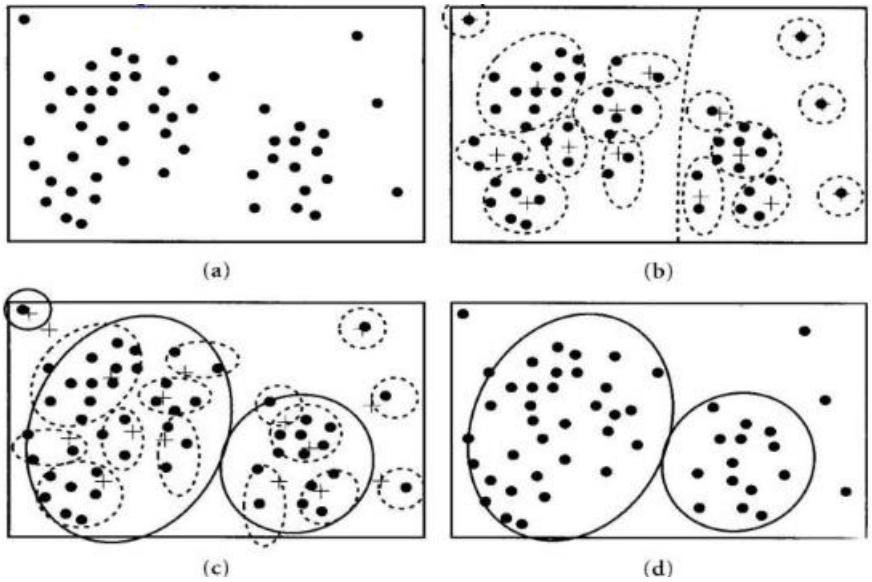


Figure 15 - CURE algorithm overview, (a) random sample of data, (b) Data are partitioned and partially clustered (dashed lines), representative points are marked with a '+', (c) The partial clusters are merged following the 'dmin' process and at each merging step, rep. points are shrunk towards the centroid of a fraction 'α', (d) final clusters and outliers obtained.[26]

In Fig.15 are shown the 4 phases of CURE, the initial samples picking, the selection of the representative points (with the '+'), the shrinking, and in the end, the hierarchical merging and the final clusters with the outliers.

Advantages of CURE:

→ Cluster Shape Variability**:** CURE excels in identifying clusters of various shapes, including non-spherical ones, unlike algorithms like K-Means which are limited to spherical clusters.

→ Suitability for Large Datasets**:** Specifically designed for large datasets, CURE efficiently handles and clusters substantial amounts of data.

→ Outlier Robustness**:** It exhibits low sensitivity to outliers and possesses the ability to effectively identify and handle outliers, which is an added advantage compared to K-Means.

Disadvantages of CURE:

- → Initial sample size and representative points**:** The performance of CURE can be highly influenced by the initial selection of sample size and representative points.
- → Dependence on 'k': Similar to K-Means, the selection of the number of clusters ('k') is crucial and can significantly impact the clustering outcome.

3.3 Density-Based Clustering

In this third class, clusters are determined by considering the density of data points in a region. These groups are identified where there are significant concentrations of data points divided by areas with lower density (in which outliers are found) [17]. A notable contrast in comparison to the first two categories is that the user must not specify the clusters target 'k' but there are other two parameters to define, 'ε' and 'MinPts', to investigate the density of data points:

- → '*ε*' is defined as the radius (or maximum distance) of neighbourhood in which identify neighbour points, while
- → '*MinPts*' is the minimum number of points required to define a 'core point', if they are present in its neighbourhood *ε.*

The user can choose the MinPts value, starting from a minimum of 3 and increasing it based on the dataset size, while, the optimal *ε* distance can be determined in many different ways, one is by investigating the average distances of each point to its MinPts (set by the user) nearest neighbours.[29] In this category, points are divided in classes.

- → A point p is a '*core point*' if there is a number of points (considering p itself) higher or equal to MinPts within a distance ε from it (point A in Fig.16), on the other hand,
- → q is a '*reachable point*' if it is not a core point, but it is within a distance ε from a core point p (points B and C).
- → In the end, points that are not core points and not even reachable from any core points are '*outliers*' (point N)

The most important rule of this class is that only core points can reach other points and not vice-versa. Two examples of density-based clustering are, the most widely known DBSCAN ,or its upgraded version that will be analysed here (and used during testing): OPTICS.

DBSCAN (Density-Based Spatial Clustering Application with Noise) is the most used algorithm for density-based clustering, and it is based on all the previously explained parameters. In simple terms, the algorithm works by examining the *ε*-neighborhood of a point called p. If it determines that this point is a core point, based on user-defined parameters, it assigns the point to a cluster. The algorithm then moves on to other points. If another core point is found and grouped in the same cluster as the first one, the *ε*-neighborhood of that cluster expands to include all points reachable from the core points in that cluster. The process continues until all points are analyzed, resulting in identified clusters and detected outliers. [30][31]

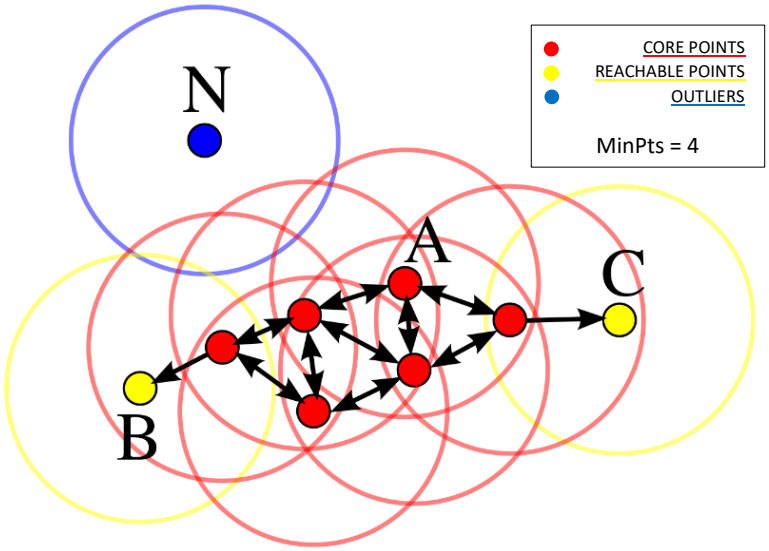


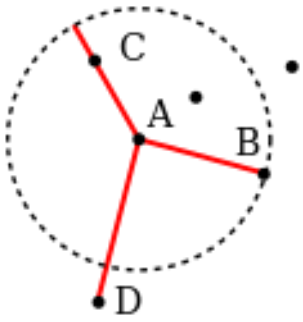
Figure 16 - Density-based clustering, classes of points: Core points (red), reachable points (yellow) and outliers (blue). Point A and the other red points are classified as core points because they have a number of points higher or equal then MinPts=4 in their distance ε. Points B and C are reachable points because they are included in the ε of a core point, while N is an outlier because is not part of both precedent classes. [32]

3.3.1 OPTICS

OPTICS, which stands for Ordering Points To Identify the Clustering Structure, is an enhanced and more comprehensive version of its predecessor, DBSCAN. The significant advantage of this algorithm lies in the fact that the ε does not need to be manually chosen by the user, instead, it is dynamically determined by the algorithm. Specifically, each point is associated with a different ε value based on the user-defined MinPts parameter. This feature addresses the major limitation of DBSCAN, which assumes a fixed ε value for investigating cluster densities and so that completely different clusters have same densities, whereas OPTICS dynamically adjusts ε for each point based on the specified minPts value and so it can handle different density clusters.[33]
Moreover, OPTICS introduces the definition of two types of distances: core distance and reachability distance.

- → The '*core distance*' is the minimum distance at which a point p can be considered a core point. If the point is not a core point, this parameter remains undefined.
- → The '*reachability distance*' is the minimum distance at which a point q can be considered reachable from a core point p. This value cannot be lower than the core distance and remains undefined (for the point q with respect to the point p), if for example, the point p is not a core point.

In Fig. 17 is shown an example to better understand the two definitions stated here above. The MinPts is set equal to 4 and so the core distance for the point A is equal to d(A,B) being B the 4th point to satisfy this condition. On the other hand, analysing reachability distances, the one between A and D is just the distance between the two points, while, the reachability distance between A and C is equal to the core distance because the distance between two points is lower than the core one.



MinPts = 4

Core distance = d(A,B)

Reachability distance (A,D) = d(A,D)

Reachability distance (A,C) < Core distance →
Reachability distance (A,C) = Core distance (=d(A,B))

Figure 17 - Core and reachability distances. Core distance for the core point A is equal to d(A,B); Reachability distance between (A,D) is equal to the distance between the two points, while, the one between (A,C) is equal to core distance (d(A,B) [30]

The ultimate goal of OPTICS is to construct reachability plots from which clusters can be extracted. These graphs are generated through multiple iterations by rearranging the reachability distances of all the points of the dataset with respect to core points. The algorithm begins with the first iteration by randomly selecting a point p, it analyses its ε-neighborhood and, if this point qualifies as a core point based on the chosen MinPts value, it calculates its core distance. Subsequently, it computes all reachability distances of other points relative to it (which are included in a list), following the rules explained earlier and initially setting its reachability distance as undefined.
If, however, this point is not a core point, the algorithm moves on to another random point q.
In the subsequent iterations, OPTICS automatically selects the first core point from the list of reachability distances obtained in the previous iteration. It then recalculates the reachability distances of other points from this core point. If distances shorter than those in the first iteration are found, these values are updated with the new ones. The algorithm continues until all points have been investigated, and after this, reachability plots are generated rearranging data points based on their reachability distances, thus forming valleys.

After that, clusters are extracted from reachability plots using the X_i method, an automatic threshold that detect the steepness of the valleys formed in reachability plots. In Fig. 18 left is displayed an example of 3 valleys found by applying the reachability distance methods of OPTICS on a dataset composed by 60 items, and on the right are illustrated the 3 extracted clusters [34].

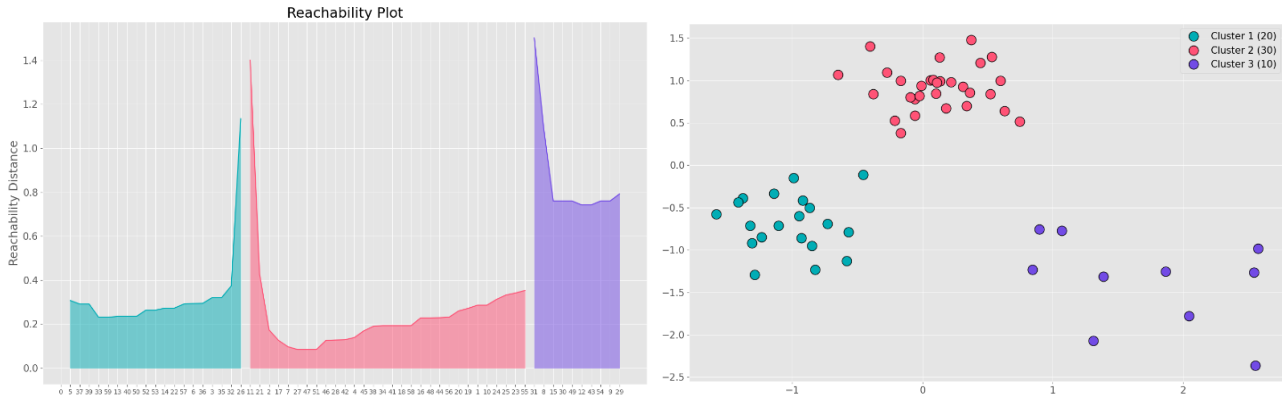


Figure 18 - Reachability plot (left) composed by 3 valleys detected by OPTICS algorithm and the three corresponding clusters (right) extracted with the X_i threshold.[30]

Advantages of OPTICS clustering:

→ Parameter-less approach**:** OPTICS is a parameter-less algorithm, eliminating the need to specify the number of clusters 'k' and the '*ε-neighbourhood*'

→ Robust handling of noise and outliers**:** It effectively identifies core points and non-core points, enabling robust clustering even in the presence of noise and outliers.

→ Handling varying densities and shapes**:** In addition to DBSCAN, it can extract clusters of varying densities and shapes, making it versatile for different cluster structures.

Disadvantages of OPTICS clustering:

→ Computational expense and slowness**:** Can be computationally expensive and slow, particularly for large datasets, impacting processing time and, it requires more memory compared to DBSCAN. [35]

After analysing in detail the clustering techniques and the three selected examples for each of them that were used in the testing phase, the next chapter will focus on the methodology underlying this project. Specifically, it will delve into how these algorithms work and what their limitations are when dealing with apparent resistivity data.

Chapter 4

4. Methodology

This chapter will discuss the methodology employed in the development of the project. This will be explained through synthetic data examples, encompassing the generation of synthetic 1D models, the computing of apparent resistivity data, the explanation of the clustering criteria and the clustering methodology, and the cross-rescaling of the clustered data to assess the quality of clustering results, investigating the best combination of criteria and the optimal algorithm.

4.1 Synthetic 1D model and apparent resistivity curves simulation

The methodology begins with the simulation of synthetic 1D resistivity models, which were used to develop the method before applying it to real data (an example of which will be discussed in the results chapter). To generate these models, and in the whole project, Python language was used, and its Integrated Development Environment (IDE) employed were specifically JupyterLab and PyCharm. In approximately 30 tests conducted with synthetic datasets, the initial 1D models were randomly generated, starting from 20 models, to 50 (this test will be used as a sample in this chapter to explain the method), then ranging up to 200, and finally creating a set of 1000 models, the results of which will be discussed in the next chapter. The algorithm was allowed to simulate the models, and the only constraints imposed were the intervals to be respected for the three key values: frequency, depth, and resistivity. These values are listed in Table 1.

frequency	depth	resistivity
100 samples in range [1, 10^4] Hz	n. of layers = 4 + 1 (halfspace)	5 values in range [100,3000] Ω·m
	thickness of each layer in range [200,500] m	

Table 1 - Frequency, depth and resistivity interval values to generate random 1D resistivity models.

After defining the ranges of values for each parameter, the algorithm proceeds with the simulation of 1D resistivity models. In Fig. 19, is depicted an example of 50 synthetic layered resistivity models.

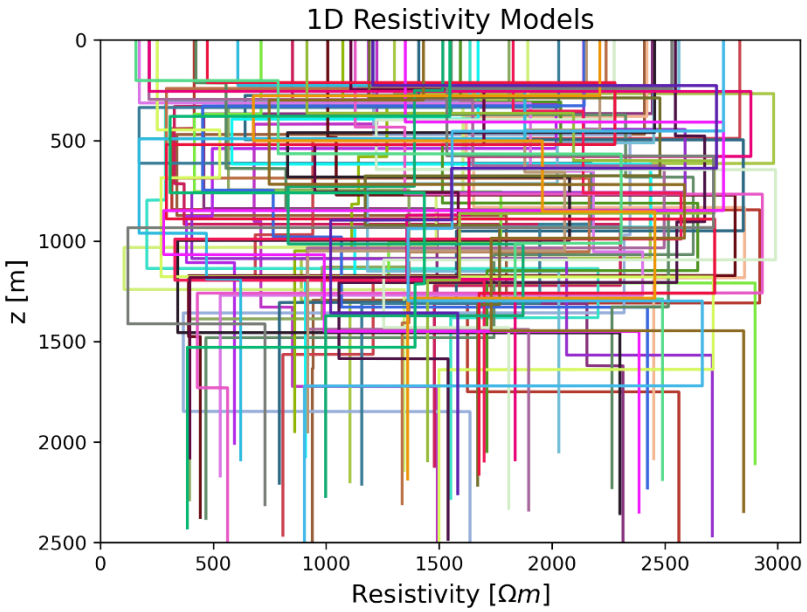


Figure 19 - 50 1D resistivity models randomly generated and defined as a function of depth.

To convert 1D resistivity models to MT apparent resistivity data, the Python routine "*empymod*" forward modelling by *Werthmüller (2017)* [16] was used. This routine takes input data such as:

- → Source position used to simulate plane waves, which is positioned at a distance of 100,000 [km] from the receivers, to emulate the natural currents of the sun.
- → Model Depth values generated, so layer boundaries
- → Model Resistivity values generated, both air and subsurface resistivities
- → Frequency values generated

Through these input data, the routine automatically calculates the values of the horizontal electric field (E_x), and orthogonal magnetic field (H_y) measured. These values are then inserted into Eq.9 to measure the impedance value Z_{xy} (Ω), which is subsequently used in Eq.10 to determine the apparent resistivity values.

At this point, the apparent resistivity curves could be represented as a function of frequency, but to compare them with resistivity models, a domain change is necessary. Through the Niblett-Bostick transformation (Eq.11), pseudo-depth values for each curve are obtained.

In Fig.20 are illustrated apparent resistivity curves obtained with empymod routine from the 50 randomly generated models of Fig.19. As shown, they are depicted with same colours of the models from which are computed.

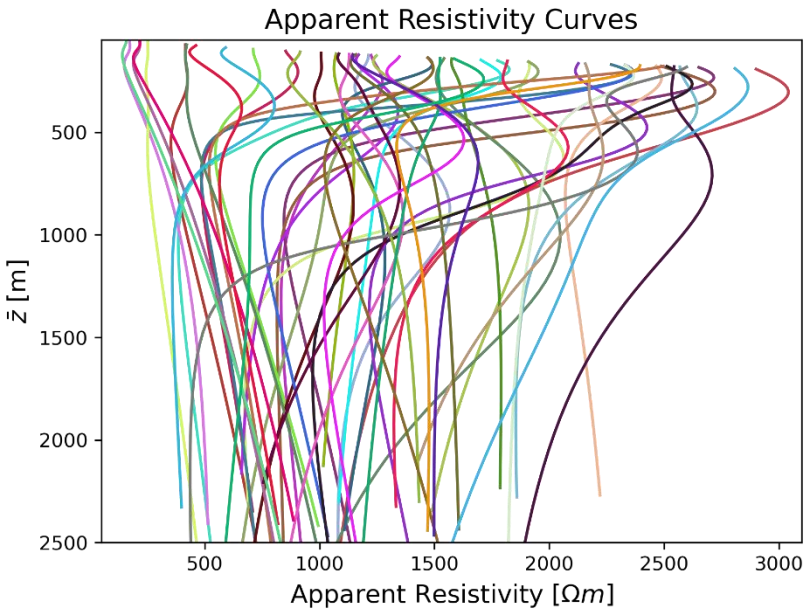


Figure 20 - 50 apparent resistivity curves dataset, obtained by empymod routine (from the 50 randomly generated models of Fig.19) and defined as a function of pseudo-depth

4.2 Clustering criteria and methodology

Once the apparent resistivity curves have been obtained, the next step is to divide them into clusters considering mathematical parameters as clustering criteria, so requiring that curves within the same cluster have similar parameters (or a combination of them). The aim of clustering, is to obtain, in the end, a depth/pseudo-depth relationship for each cluster able to rescale all the data cluster into models with the lowest possible error.

The idea for initializing the clustering algorithms is to create a multi-dimensional input matrix that contains a set of mathematical features/geometrical parameters (linked to a physical meaning) for each apparent resistivity curve. Each parameter added to the matrix will effectively constitute an additional dimension, in fact the three tested algorithms, k-means, CURE, and OPTICS, will have to work to form clusters of apparent resistivity data operating in a multi-dimensional space. They will group, data that have the most similar mathematical parameter values or combination of such parameters.
Several tests and implementation of clustering algorithms were conducted, starting from using all parameters together and gradually reducing their number, trying to identify the most significant ones that ensure the best clusters. This means finding the ultimate combination of clustering parameters, and the best algorithm, that ensures the grouping of depth/pseudo-depth relationships that are most similar to each other into the same clusters, and so the minimum error in the cross-rescaling and direct transformation of data into models.
The different parameters (or clustering criteria) considered, and their physical meanings, are listed hereinafter.

4.2.1 Clustering parameters

The investigated clustering parameters (or criteria) are divided into 3 main categories:

→ Resistivity values based parameters
→ Gradient based parameters
→ Overall trend/tendency of the curve based parameter

1. **Resistivity parameters:** In this first category, the 8 parameters are mainly related to the numerical value of apparent resistivity, and corresponding pseudo-depth, for the different feature of interest:

→ Average resistivity and average pseudo-depth**:** This set of parameters provides a comprehensive perspective on the entire curve, reflecting the average values of resistivity for the geological materials involved.
The average resistivity not only captures the overall behaviour in terms of resistivity, but also aids in categorizing the curve within a specific field: resistive (higher avg. resistivity values), conductive (lower avg. resistivity values), or neutral (values between the first two fields).
Regarding average pseudo-depth, it represents the middle point of the maximum investigation depth, so, the mean thickness of the whole geological body that was studied.

Fig. 21 shows the representation of the average pseudo-depth/average resistivity point for the profile n.5 of the dataset under examination (Fig.20).

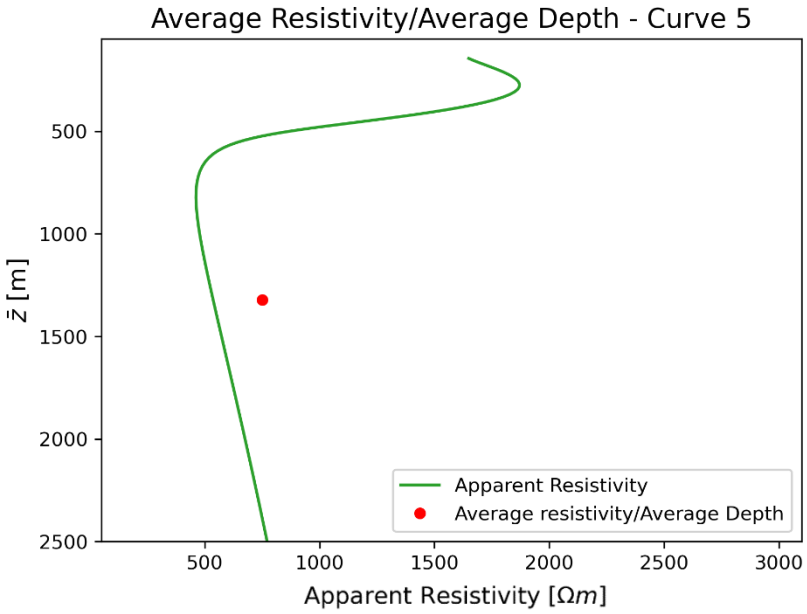


Figure 21 - Average resistivity/Average pseudo-depth point for the profile n.5 of the 50 apparent resistivity curves examined

→ Initial resistivity and final resistivity: This pair of parameters represents the initial and final values of the resistivity curves. Physically, it symbolizes the resistivity value of the first layer (Initial resistivity), and a resistivity value that takes into account the cumulative effect of all layers (Final resistivity).

→ The highest local maximum, the lowest local minimum and corresponding pseudo-depths: this is a set of 4 values, two related to resistivity and the other two associated with depth. Mathematically, they correspond to inflection points (first derivative of the curve = 0).
The highest local maximum is defined as the highest point among local maxima in the apparent resistivity curve, and the corresponding depth, is the depth at which this maximum occurs. A local maximum is a point in the curve at which the value is greater than the values of nearby points, but not necessarily the highest within the entire curve and it occurs when the first derivative of the curve in this point is equal to 0 and the second one is lower than 0. Physically, it might represent a layer, or a formation, with very high resistivity, such as rocks or compacted soil, that are located between formations characterized by lower resistivity.
Like the local maximum, the lowest local minimum values represent the lowest point among local minima in the resistivity curve, and the corresponding depth, is the depth at which this minimum occurs. In contrast to the local maximum point, a local minimum is a point in the curve at which the value is lower than the values of surrounding points, but not necessarily the lowest within the entire curve and it occurs when the first derivative of the curve in this point is equal to 0 and the second one is higher than 0. It could indicate areas with very low resistivity, such as more permeable soil or water-bearing formations, located between zones of higher resistivity.

In the case where local maxima or minima are not found in the entire apparent resistivity curve, such as a profile that is either only increasing or only decreasing, fictitious values (e.g., 1) are assigned to the highest local maximum or the lowest local minimum (and also to their corresponding pseudo-depths) of the curve, replacing the values of these parameters. This approach informs the algorithm that no maxima or minima are present in the curve, indicating that resistivity increases or decreases linearly with depth. In other words, there are no layers with higher resistivity between layers with lower resistivity and vice versa.

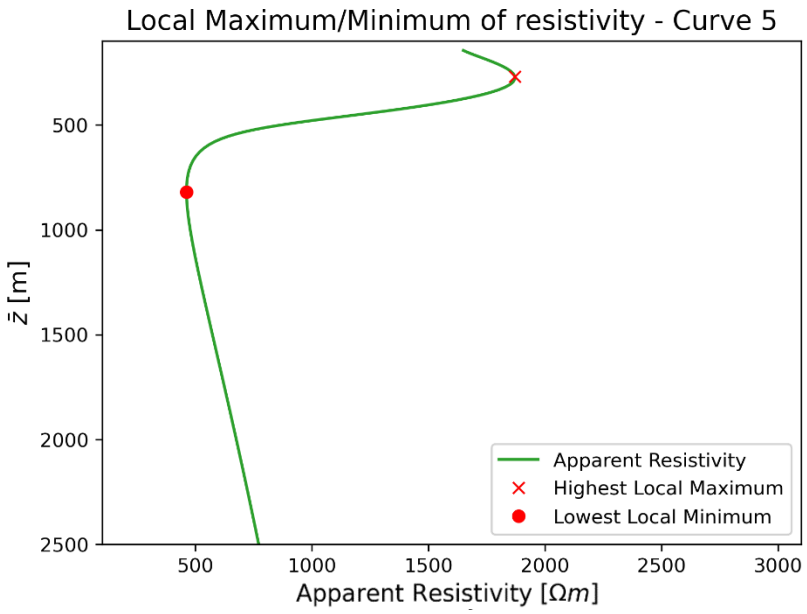


Figure 22 - Highest local maximum and lowest local minimum of the profile n. 5 of the dataset

In Fig. 22 is illustrated one of the 50 apparent resistivity curves examined, more specifically the profile number 5, and on this curve are depicted the highest local maximum, and the lowest local minimum point.

2. **Gradient parameters:** In this category, parameters are related to the different gradients that make up the apparent resistivity curves. Each gradient between two points of the curve is obtained by:

$$\nabla(1,2) = \frac{\rho_2 - \rho_1}{\bar{z}_2 - \bar{z}_1} \tag{19}$$

Each transition from one gradient to another, for simplicity, is represented by the shift from a positive gradient to a negative one, that from a mathematical point of view corresponds to the first derivative of the curve equal to 0. Initially, all gradients between the points of the curve are calculated. If a change in sign occurs (f ' =0), the gradient from the first point to the last with the same sign is calculated and considered as the first gradient. Subsequently, another gradient is considered, starting from the point of the sign change to the last with the same sign, and so on until the last gradient found. This means that two gradients of the same sign but different shape will be treated as a single gradient, calculating the gradient between the first point of the first gradient and the last of the second one. The two parameters considered are:

→ Number of gradients: They correspond to the number of sign changes in gradients of the entire curve + 1, so to the number of zeros of the derivative + 1 (or also the number of local maxima/minima detected in the curve +1). Physically, each shift from a positive to a negative gradient is a detectable transitions between a resistant to a conductor layer (or vice versa). So, they represent the number of significant alterations in the subsurface geological structure or material properties.

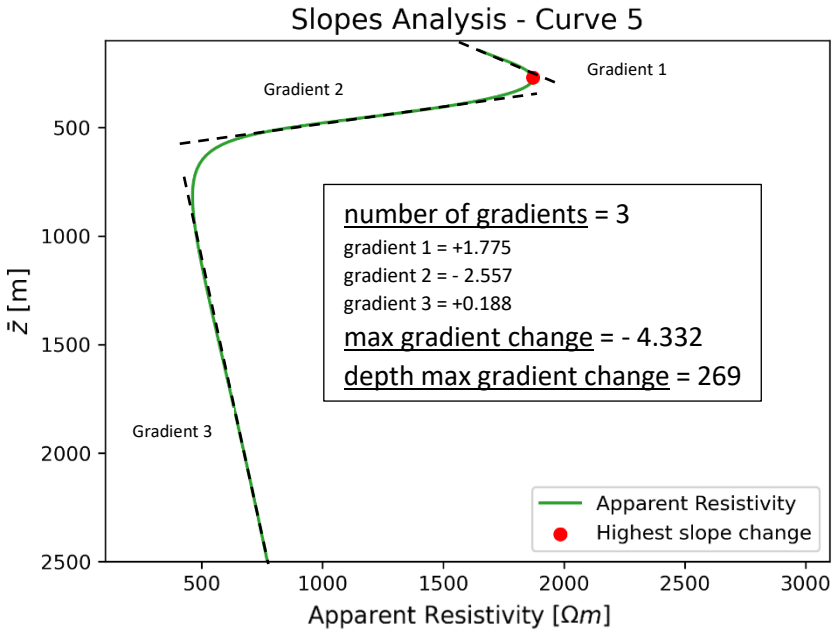


Figure 23 - Values of the 3 identified gradients of the profile n. 5 of the dataset and highest gradient

→ Highest gradient change and corresponding pseudo-depth: These values correspond to the most significant variation in gradient sign (f ' =0) across the entire curve. Mathematically, it corresponds to the point of minimum distance between two inflection points. By indicating the highest shift from a positive gradient to a negative one, these parameters symbolize the most abrupt transition from a resistant to a conductor layer (or vice versa). The highest gradient change is evaluated by considering the highest absolute value, by calculating the

differences between the contiguous gradients along the curve. Instead the pseudo-depth correspond to the depth at which this transition happens, so to the point in the curve at which the 1st gradient ends and the 2nd one starts.

Also with these parameters, if no sign change values of the gradient are found, fictitious values are assigned instead of the highest gradient change value and the corresponding pseudo-depth.
Fig. 23 displays the same curve as Fig. 22, but with added information on the number of gradient, their values and the highest gradient change identified in that profile (red dot).

3. **Overall tendency of the curve parameter:** In this last category of clustering parameters, a factor has been introduced to take into account the overall trend of the curves, considering all behaviours of different segments. This parameter is:

→ Ratio between total area and length of the curve: This parameter is a ratio between two factors. The first one is an integral, so an area calculated between each apparent resistivity curve and a straight line drawn at its mean value. The integral computation is performed by summing two components: the first is a positive area corresponding to the region on the right side of the line and to the left of the curve, while the second is a negative area formed by the portion to the left of the line and to the right of the curve.
The second factor is the effective length of the apparent resistivity curve, or "length of the arc", measured by approximating the curve to a set of segments connecting the points from which it is composed, and then summing the contribution of each segment. This parameter has been chosen to differentiate and avoid clustering of curves with similar total area values, but exhibiting completely different characteristics (e.g., 4 gradients vs only one).
The ratio between these two values is considered as a clustering parameter. Therefore, for each curve, a factor is determined, whose sign corresponds to the general tendency of the curve, considering the individual contributions of all the layers. It can be resistive if the sign is positive or conductive if it is negative. Furthermore, the magnitude of this parameter symbolizes the trend, differentiating more resistive/conductive curves, from less resistive/conductive ones.

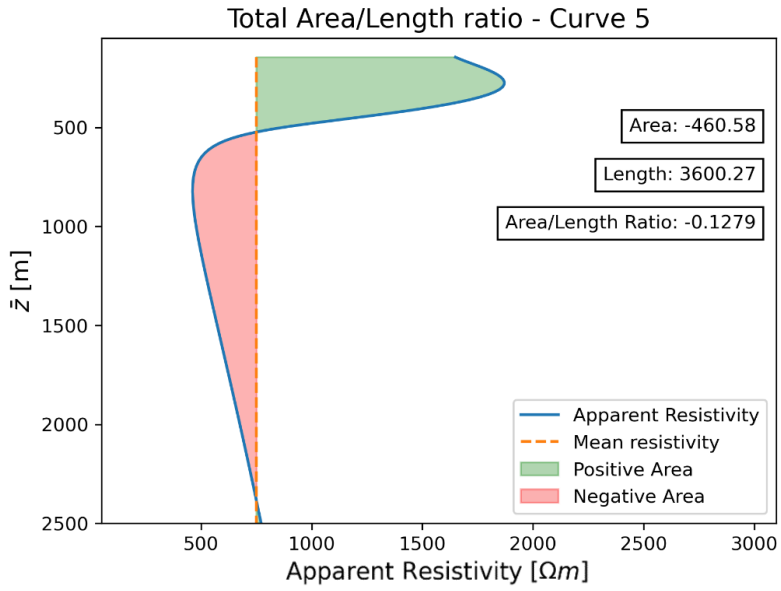


Figure 24 - Total area/length ratio calculation of the profile n.5 of the dataset. Computed results are shown in a table on the right. The negative sign and the magnitude of the ratio confirm a clear conductive overall tendency of the curve.

In Fig. 24, the calculation of the ratio between the total area and the length of the curve is shown for profile n.5 under analysis. As expected, the sign of the ratio is negative, and the magnitude is quite low, describing a low conductive trend.
Once these parameters are determined by the algorithm, before being inserted into the input matrix of clustering algorithms, they are normalized. This process involves applying to each parameter (X) calculated for each apparent resistivity curve, this relationship:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (20)$$

where X_{min} and X_{max} are the lowest and the highest value obtained for the analysed parameter. By doing so, values ranging from 0 to 1 are obtained, and the algorithm treats them all equally. This ensures that parameters with higher magnitudes, such as average resistivity values, do not have a greater contribution than parameters with lower magnitudes, such as the number of gradients.

4.2.2 Parameters reduction

Not all the parameters contribute equally to the purpose of this project, i.e., some parameters are more efficient than others. To establish the combination of these parameters that ensures the best solution (lowest possible error between rescaled models using a rescaling function per cluster and true ones), about 30 tests were conducted using k-means as the reference algorithm. Throughout the various tests (the score of which will be shown in the results chapter, section 5.1), we started by considering all 12 parameters in the input matrix of the clustering algorithm, and then various combinations were tested, gradually reducing the number of parameters. The overall result, which will be anticipated here to also explain the choice of the best algorithm, is that the 8 'resistivity parameters' proved to be more efficient in terms of results compared to the other two categories. In particular, the first 4 explained criteria:

- → Average resistivity
- → Average pseudo-depth
- → Initial resistivity
- → Initial pseudo-depth

were selected as the ultimate combination of parameters for the input matrix of the 3 clustering algorithms.

4.2.3 Selection of the best algorithm

After the definitive set of criteria for clustering resistivity data was established, k-means, CURE, and OPTICS were tested to determine the best algorithm and thus obtain the final results.

1. **K-means**

Starting with k-means and always considering the dataset shown in Fig. 20 as the case study, the first objective is to determine the desired number of clusters. To this end, the two methods presented in section 3.1.2, Elbow and Silhouette, are employed. In Fig. 25, the results of both

methods are shown, and following this logic, in this case the optimal number of clusters is 4, at which the elbow curve starts to bend and the Silhouette Coefficient has the highest value.

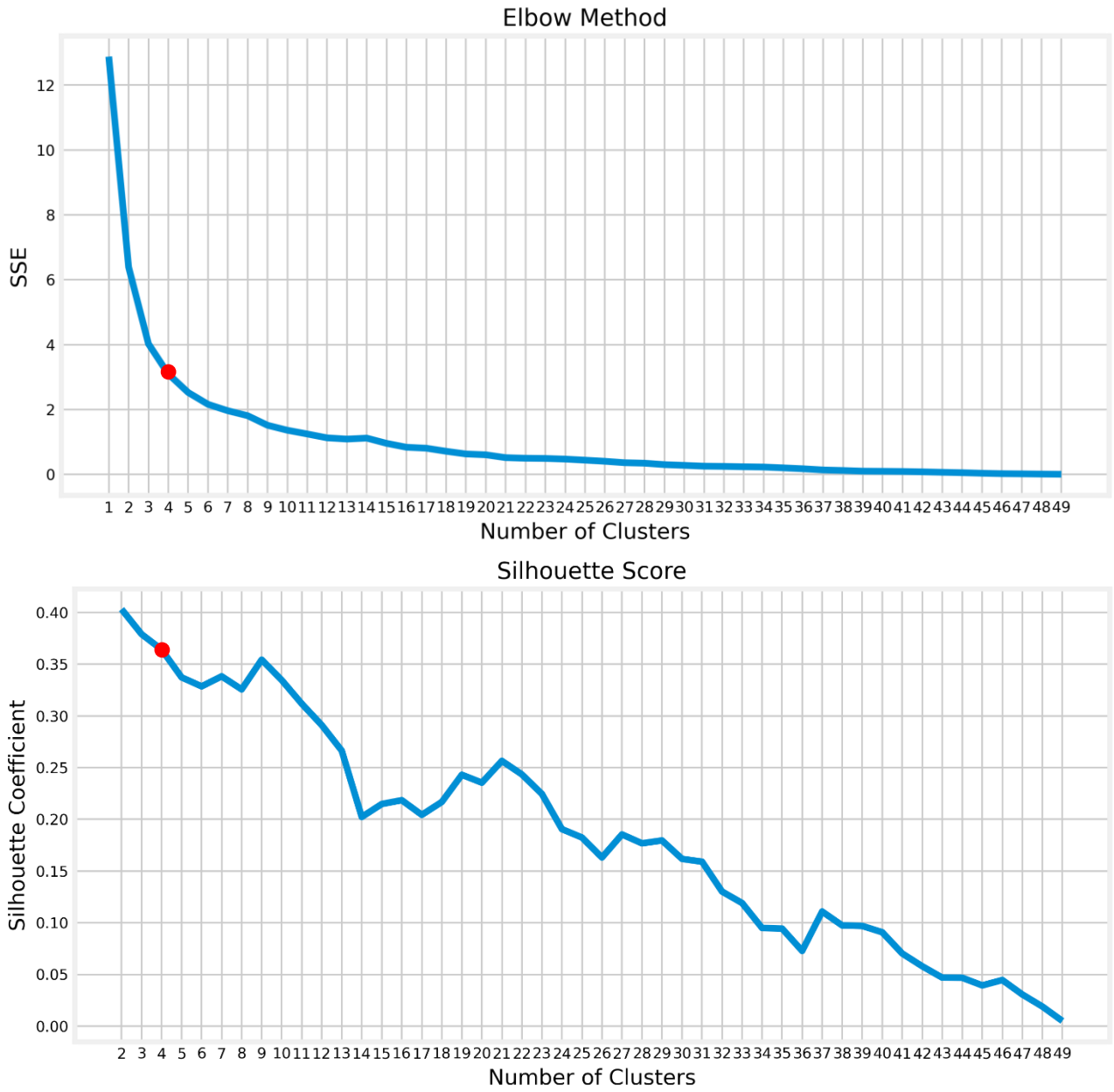


Figure 25 - Elbow (top) and Silhouette Coefficient (bottom) methods for the dataset of Fig.20. They both retrieve 4 as the optimal 'k'.

At this point, with the input matrix determined and the number of clusters established, k-means is run a certain number of times, which can be decided or left by default as in this case (number of iterations = 10), and in a few seconds, it returns the results found for the iteration with the lowest possible SSE. These values include the position of the centroids in space (*kmeans.cluster_centers_*) , the lowest SSE value found for centroid convergence (*kmeans.inertia_*), and in particular, the number of cluster labels for each data point (*kmeans.labels_*), in this case, at each curve of apparent resistivity is assigned 0, 1, 2 or 3. In Fig. 26, apparent resistivity curves of Fig.20, divided into the 4 clusters by k-means, are represented.

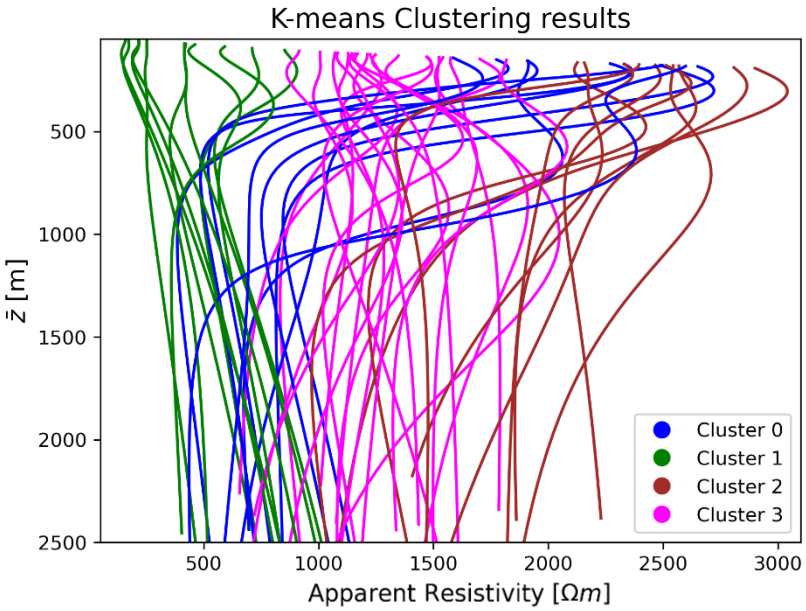


Figure 26 - K-means results for dataset of Fig.20. Apparent resistivity curves are divided in 4 clusters, 0 (blue), 1 (green), 2 (brown) and 3 (magenta)

Fig. 27 shows the curves of apparent resistivity for the 4 clusters, with each individual subplot that illustrate the curves that belong to the cluster.

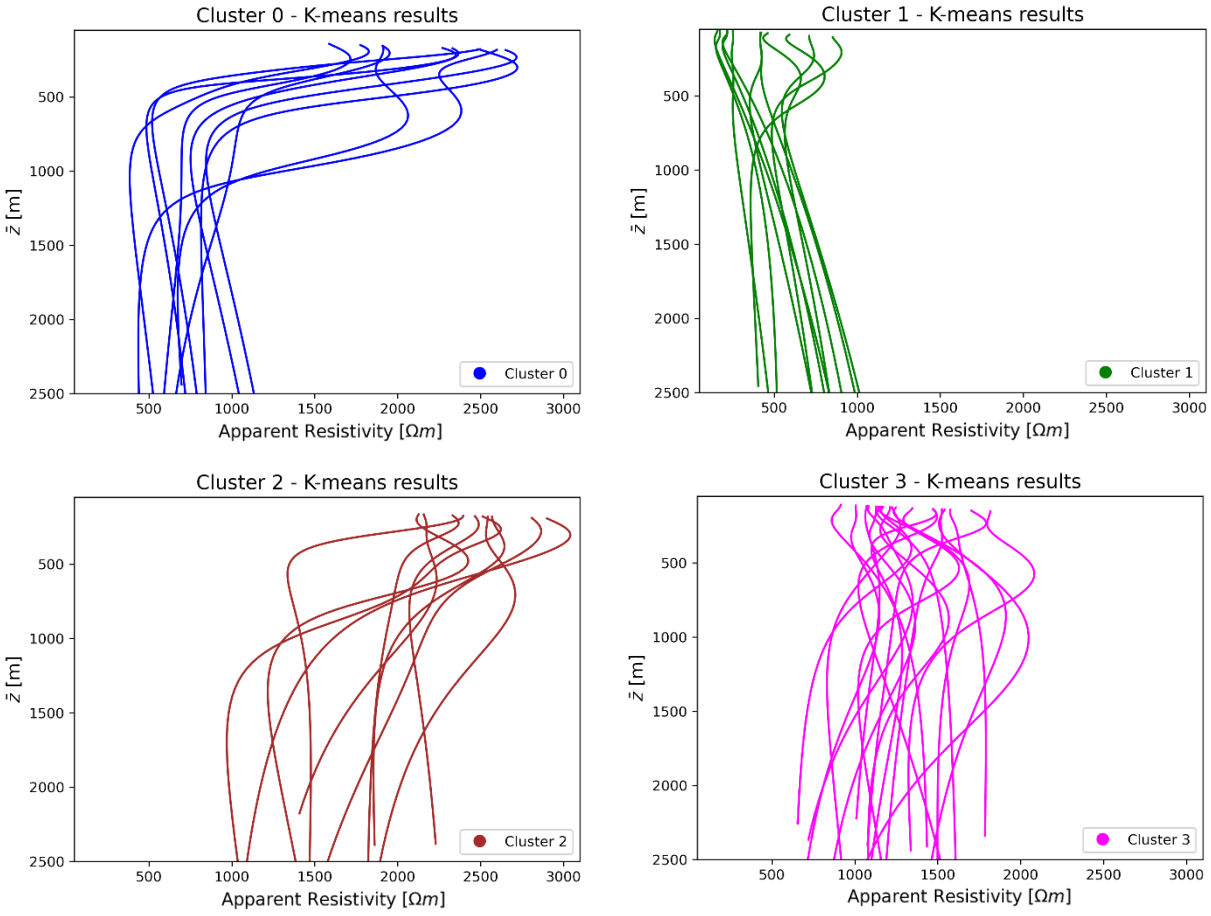


Figure 27 - Same results of Fig.26 but each plot represents the set of curves that belong to a single cluster.

Following this approach, it is also possible to derive, as a test of clustering efficiency, the depth/pseudo-depth relationships belonging to the different clustered curves (Fig. 28). These rescaling functions are obtained by comparing the data and models in the cumulative resistance domain, using the method explained in section 4.2.2. The graphs of the clustered depth/pseudo-depth relationships are an useful check, because achieving a good level of clustering, and thus minimizing the error in rescaling the data into models, requires clustering together rescaling functions that are as similar to each other as possible.

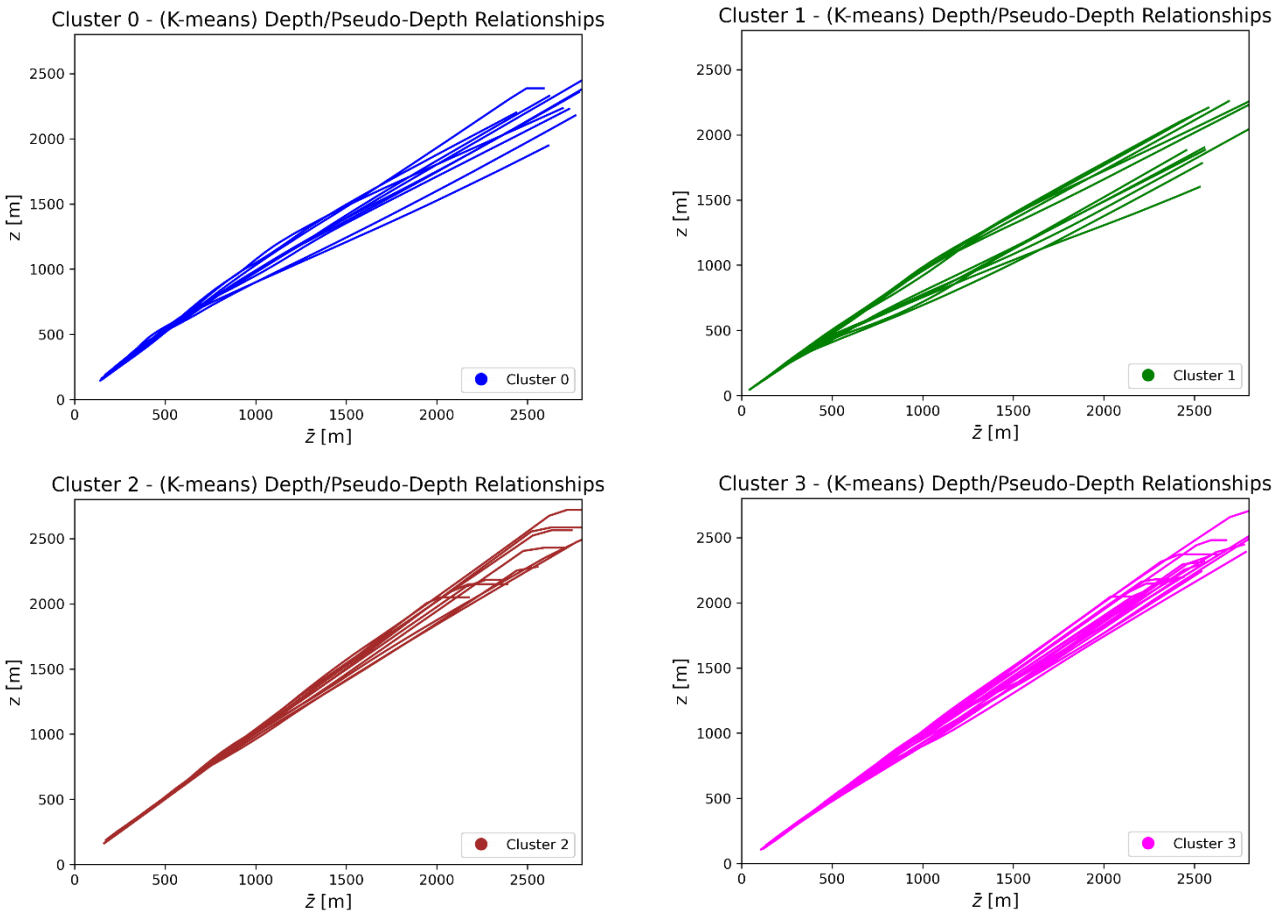


Figure 28 - Depth/Pseudo-depth relationships computed between models and data in the cumulative resistance domain, and divided in clusters based on the k-means clustering of apparent resistivity data.

Analysing the picture above, for clusters 0, 2, and 3, the rescaling functions are similar to each other, while for cluster 1 they appear to be slightly more different. Consequently, a slightly larger error is expected for this cluster.

The final check to determine the best algorithm, quantify the quality, and assess the added value of clustering the data before rescaling them, is to calculate the percentage error resulting from cross-rescaling the clustered data into models. This error is measured between the rescaled cumulative resistance models obtained using a depth/pseudo-depth per cluster (testing with iterations one rescaling function of each cluster at a time to rescale all cluster data), and the real cumulative resistance models (obtained from the models used to generate the data in the first place). The error is then depicted in graphs with sets of box-plots divided by cluster, where each box of each box plot represents the error produced by each individual rescaling function used to rescale all the cluster

data in each iteration. These box plots have been filtered from some outliers, delimiting the boxes with an upper and a lower limit (maximum and minimum error) defined as ‘ whiskers’, that establish the range of the non-outlier data points [36]. Furthermore, in each box, an orange line is drawn, representing the mean error given by each rescaling function.

In Fig. 29, the error box plots produced by the depth/pseudo-depth rescaling functions of the 4 clusters from Fig. 26 are shown. They represent the results given by k-means on this dataset. In numerical terms, they are particularly positive compared to those obtained on the same dataset but using a random rescaling function to rescale all the data (peak of 400%, Fig.7); indeed, the mean errors are respectively 6.5%, 13%, 2, and 2.4%. As for the whiskers, extreme values are -24% and 67% (due to the presence of the outliers, profile n. 4, 9 and 11 of the cluster 1).

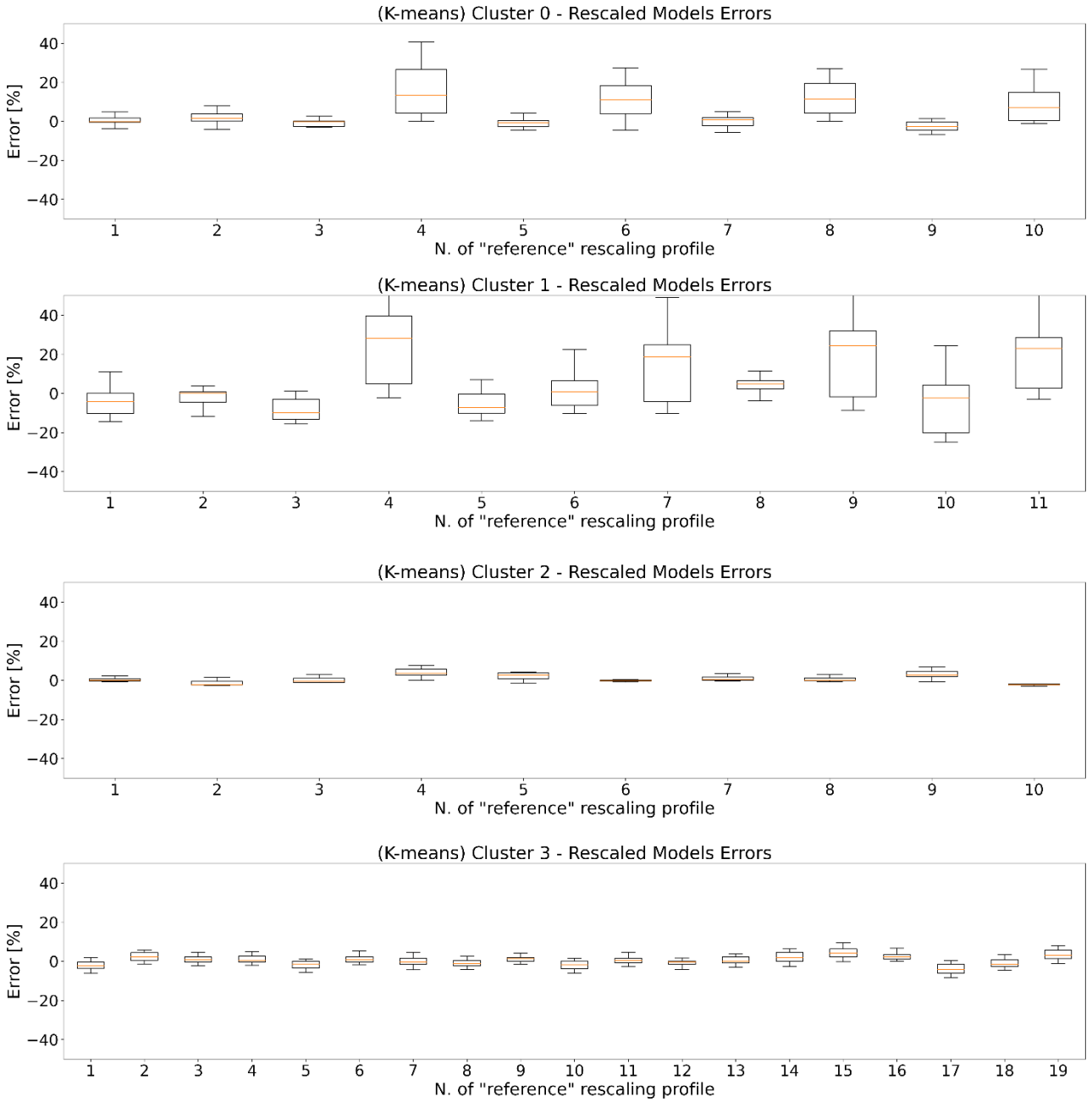


Figure 29 - Error box-plots of the 4 detected clusters by k-means algorithm. Avg. errors found are respectively 6.5%, 13%, 2% and 2.4%. Instead lower and upper limits are -24% and 67% (due to the presence of the outliers of the cluster 1).

2. **CURE**

CURE algorithm, unlike k-means, takes as input not only the clustering parameters matrix and the desired final number of clusters (which is set equal to the number 'k' found for k-means from elbow and silhouette, in the investigated dataset equal to 4), but also the number of representative points for each cluster. This parameter must be set equal to the desired number of clusters, which means that, in the case under consideration, this parameter will also be equal to 4.
After setting these parameters, CURE is run, and it automatically returns the cluster labels for each apparent resistivity curve to the command *cure.get_clusters.* At this point, it is possible to obtain the plot of the curves and the depth/pseudo-depth rescaling functions divided into clusters (Fig.30).

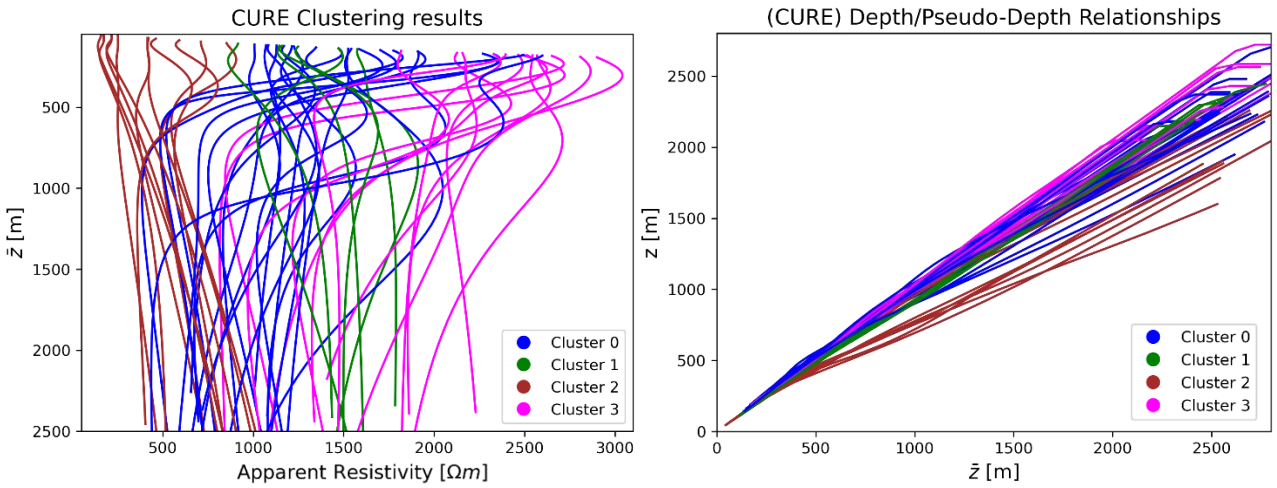
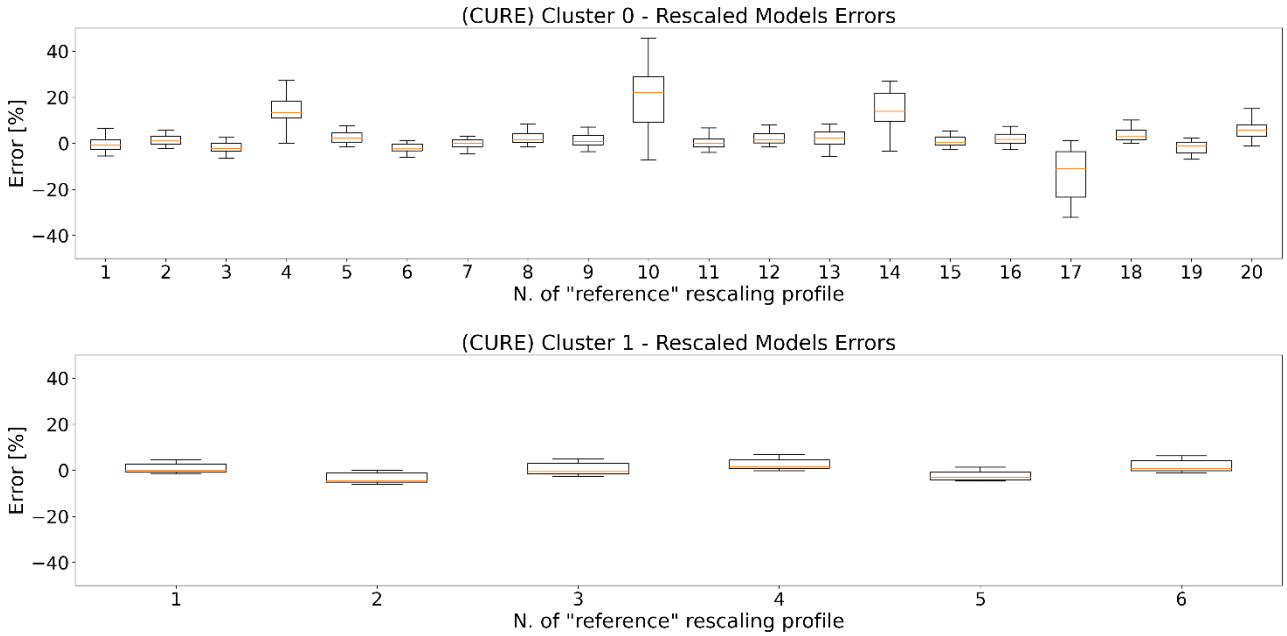


Figure 30 - CURE results for dataset of Fig.20. Apparent resistivity curves on the left, are divided in 4 clusters, 0 (blue), 1 (green), 2 (brown) and 3 (magenta). On the right are shown the corresponding depth/pseudo-depth rescaling functions for each curve, divided in clusters.

As done for the previous algorithm, in this case as well, the error percentages due to rescaling with a depth/pseudo-depth per cluster are measured and represented in box plots.



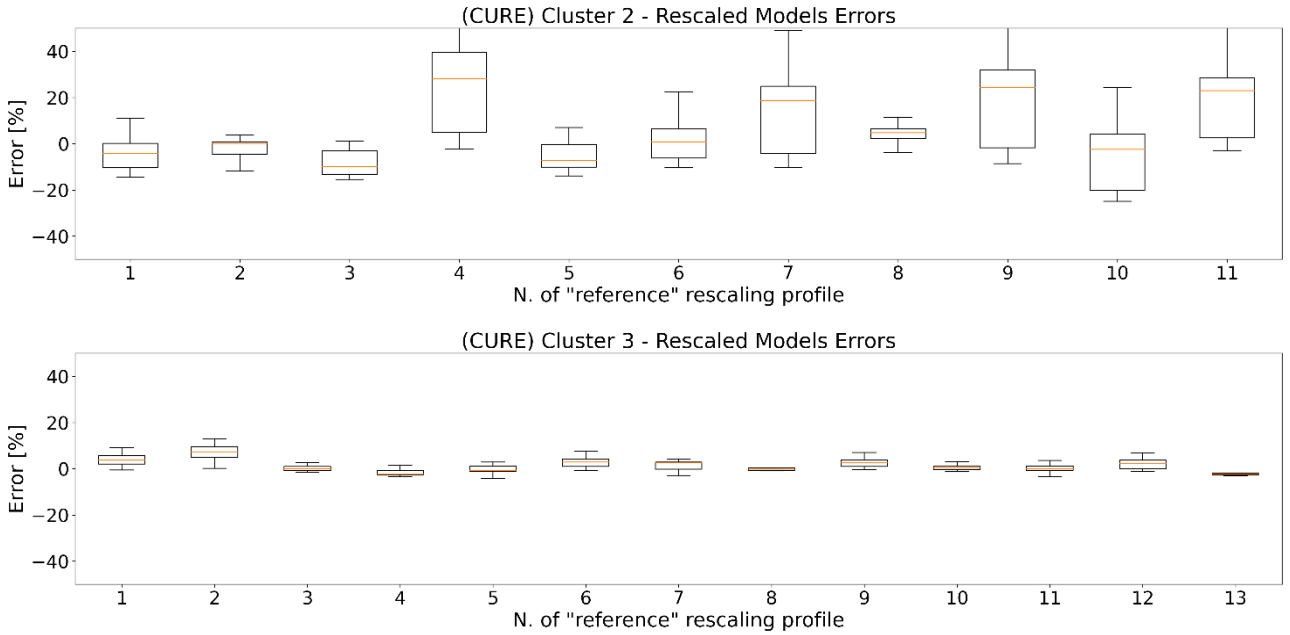


Figure 31 - Error box-plots of the 4 clusters detected by CURE algorithm. Avg. errors found are respectively 5.5%, 13%, 2.6%, and 2.7%. Instead lower and upper limits are -32% and 67%.

In Fig. 31, the error box plots produced by CURE algorithm are shown. Error results are very similar to the k-means one, in fact, the mean errors are respectively 5.5%, 13%, 2.6%, and 2.7%. As for the whiskers, extreme values are -32% (due to outlier 17 of cluster 0) and 67%. The main difference with these set of results and the one with k-means, is the distribution of curves in the clusters (just 6 curves in cluster 1 vs 20 in cluster 0), that represents a problem, in datasets composed by 200 or 1000 apparent resistivity curves like the ones tested in this Thesis and shown in the next chapter.

3. **OPTICS**

The third and final clustering algorithm tested is OPTICS. Unlike the first two, OPTICS does not require the user to specify a fixed number of clusters, but is able to determine it autonomously. The only parameter that is manually set is the minimum number of data points (in this case, curves) required to form a cluster (MinPts). This parameter is selected based on the composition of the dataset, for example in the case at hand, it is set equal to 8.

The particularity of this algorithm lies in the calculation of reachability distances and the composition of reachability plots, as explained in section 3.3.1, based on the steepness of the valleys found. Despite OPTICS working well with smaller datasets (e.g., 20 tested curves), as the number of profiles increases, the algorithm does not perform as expected. In fact, as shown in Fig.32, where the Reachability plot is depicted, OPTICS manages to identify the first two clusters (so it finds 2 valleys in blue and green) but then stops and classifies all the remaining profiles as outliers (black dots). This issue is not solved by increasing or decreasing the MinPts parameter, on the contrary, it leads to even worse results, such as finding only one single cluster or labelling all profiles as outliers.

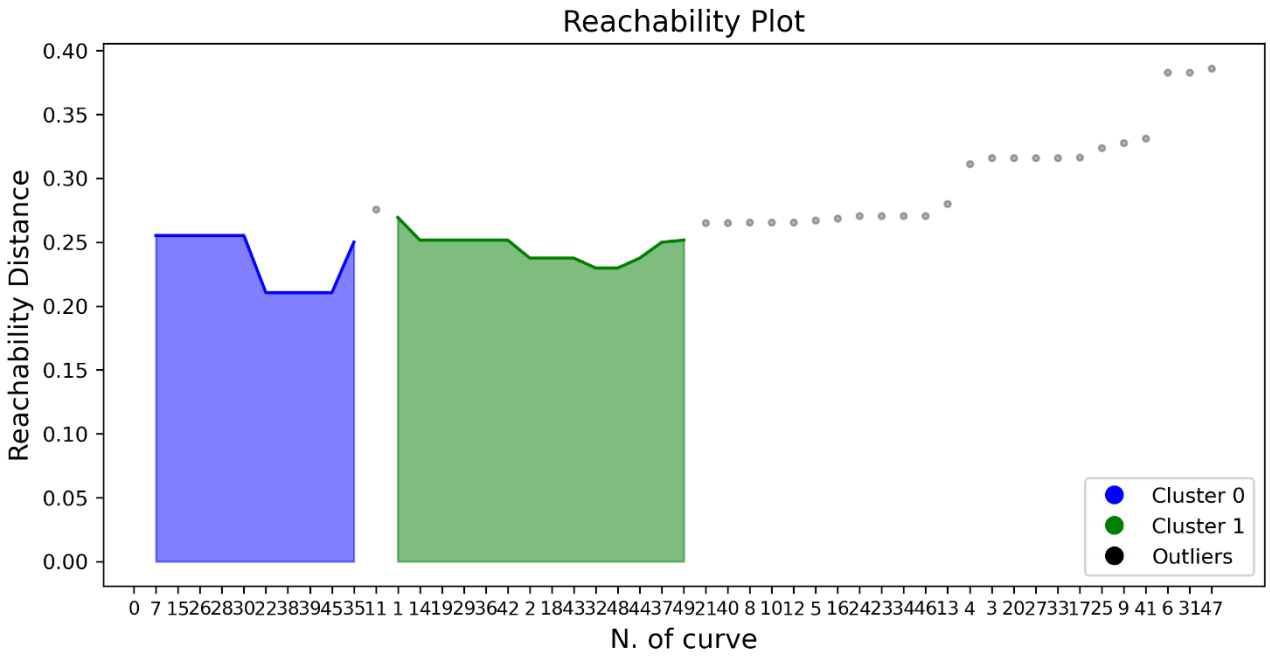


Figure 32 - Reachability plot, result of OPTICS clustering applied on the dataset of Fig.20. Two valleys are detected, so two clusters are found: 0 (blue) and 1 (green), all the other profiles are marked as outliers (black).

After that OPTICS computes the reachability plot, it automatically returns the cluster labels for each apparent resistivity curve to the command *optics.labels_*. At this point, it is possible to obtain the plot of the curves and the depth/pseudo-depth rescaling functions divided into clusters (Fig.33). The results shown by the Reachability plot are confirmed by the OPTICS results graphs, confirming the unreliability of the algorithm on this type of dataset.

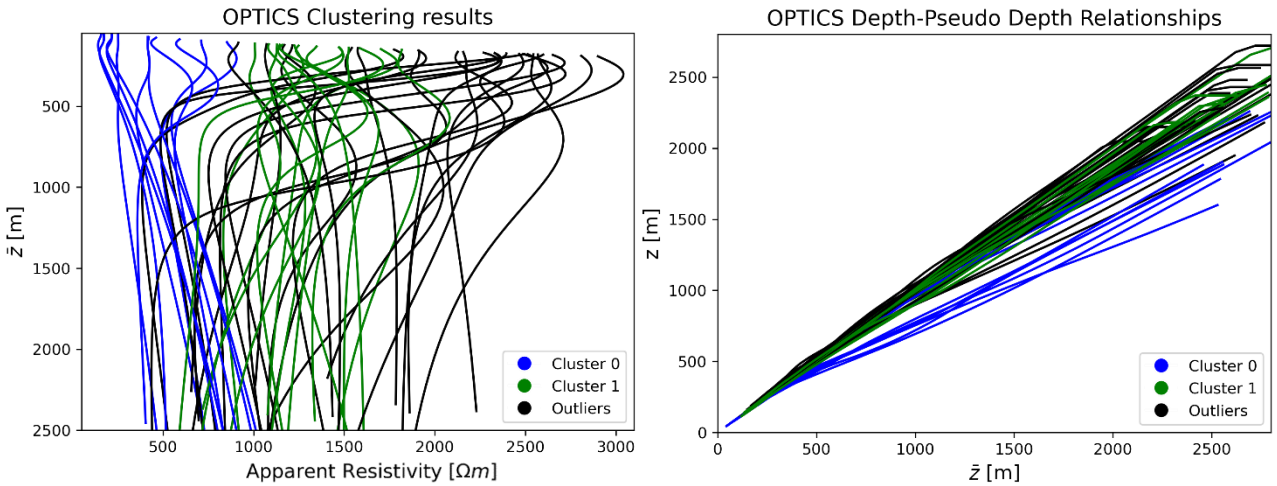


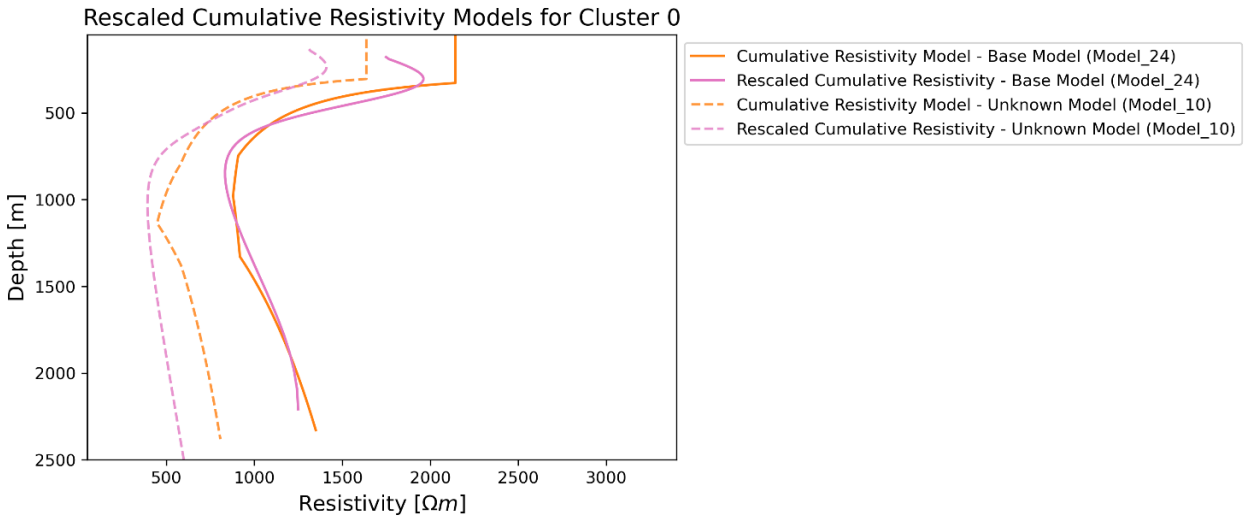
Figure 33 - OPTICS results for dataset of Fig.20. Apparent resistivity curves on the left, are divided in 2 clusters, 0 (blue), 1 (green) and all the other profiles are considered as outliers (black) On the right, are shown the corresponding depth/pseudo-depth rescaling functions for each curve, divided in clusters.

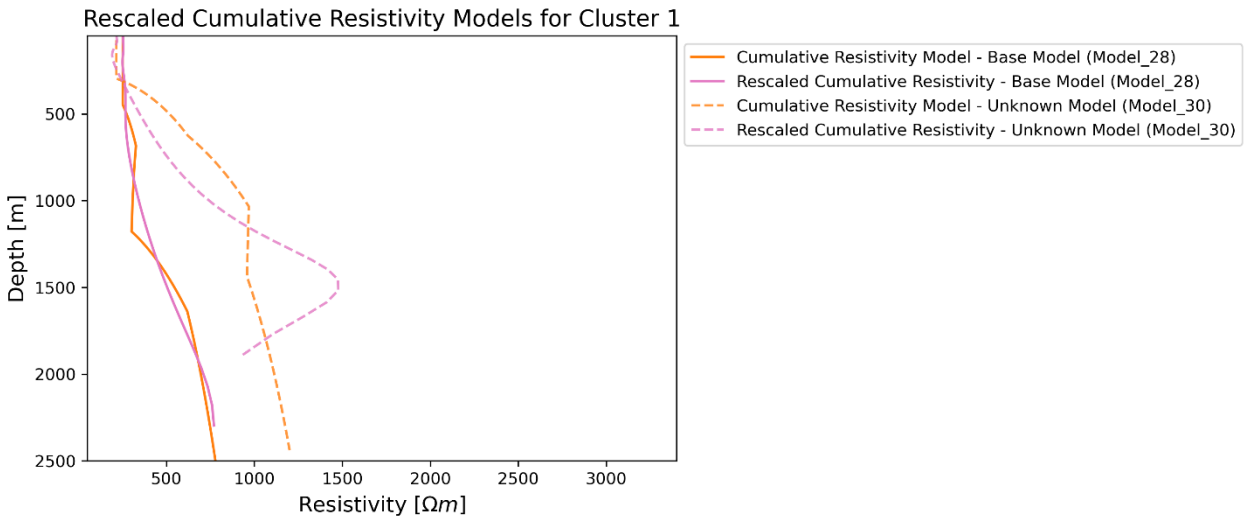
Given the not meaningful results obtained from OPTICS, the box plots related to the cross-rescaling of data into models will not be shown, as they would not consider more than 50% of the data, classifying them as outliers.

At this point, it is possible to determine the best algorithm among the three by analysing the test results. The choice is between the first two, k-means and CURE. Although the second one provides similar results to the first, for this type of dataset and for datasets composed of a higher number of curves, based on how the curves are distributed among clusters, k-means represents the best choice.

4.3 Final results

Once the ultimate combination of parameters is obtained and the best algorithm is determined, the final results are achieved. These consist of the whole set of rescaled resistivity models (of unknown models), obtained using a reference depth/pseudo-depth rescaling function for each cluster.
In the following graphs (Fig.34) one example of each cluster is shown. In the plot the unknown rescaled cumulative results (dashed purple line) obtained from the application of the reference rescaling functions, produced by the base known models of each cluster (solid orange line), are compared to their respective true cumulative resistivity models generated (dashed orange line).





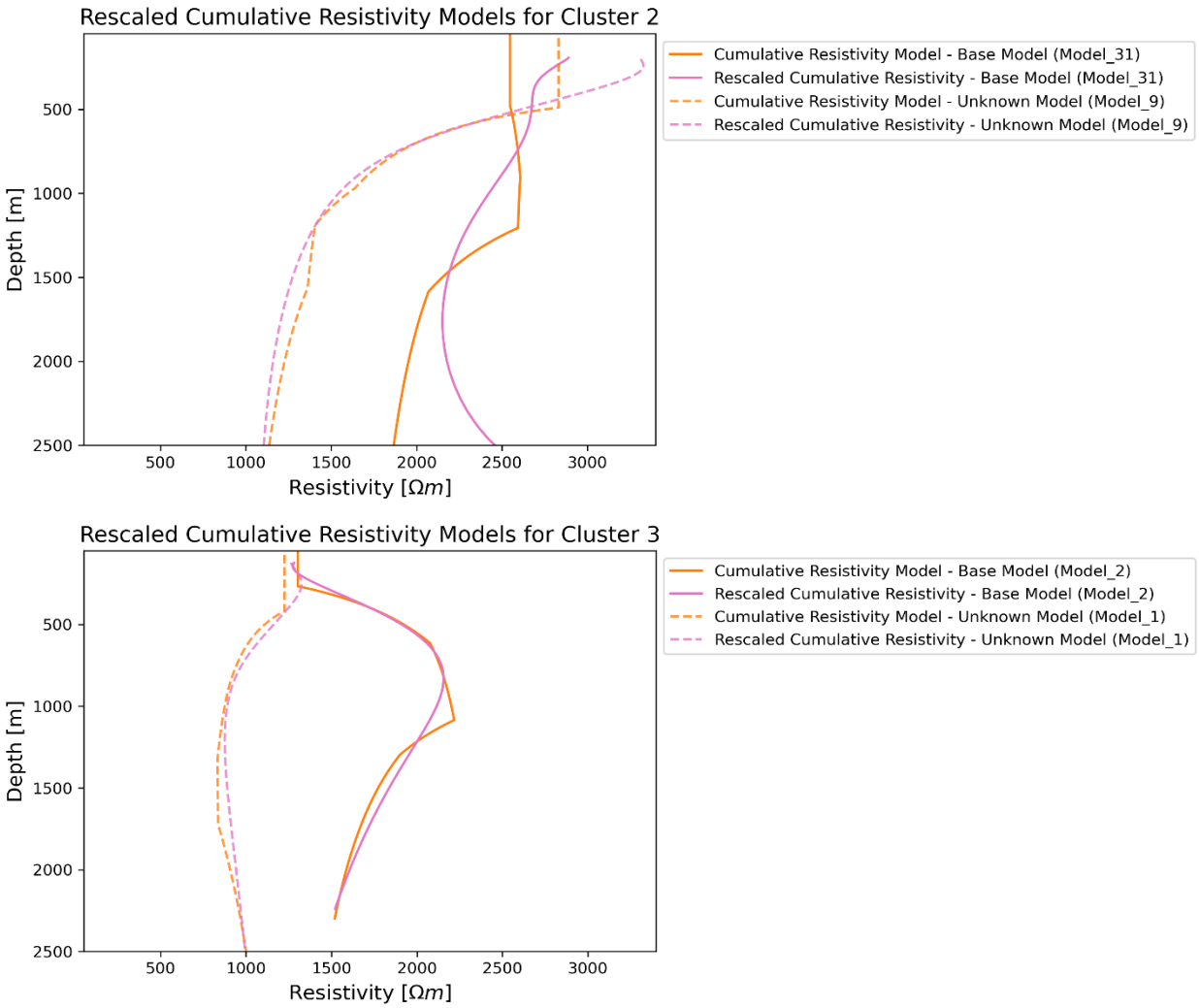
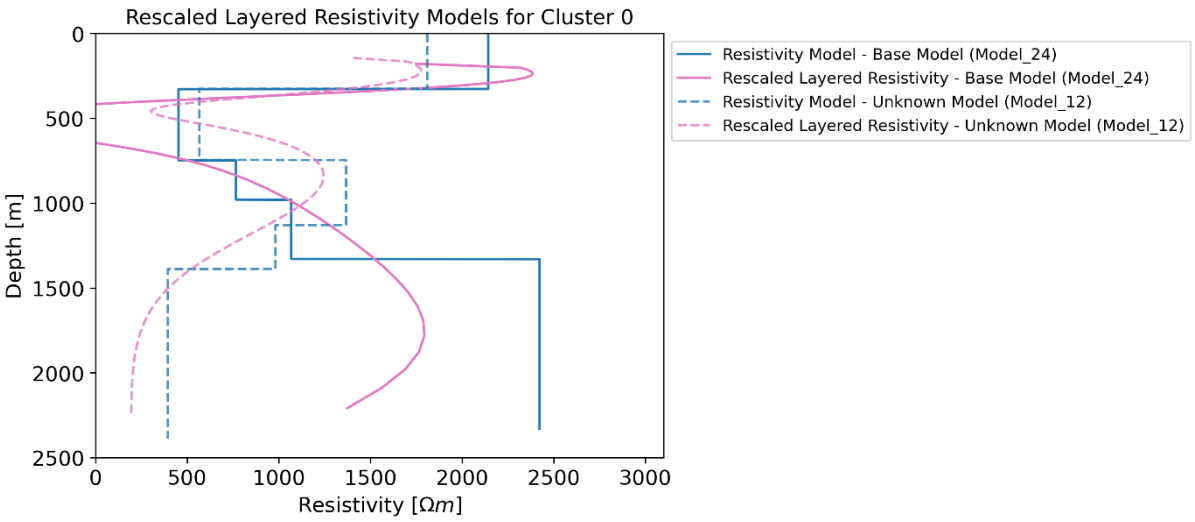


Figure 34 - 4 examples of the cumulative results obtained, 1 for each cluster. In each plot are represented, with a solid line, the cumulative base model (in orange) from which the depth/pseudo-depth rescaling function for the cluster in analysis is obtained and the rescaled one (in purple). On the other hand, with a dashed line, are shown the unknow cumulative model (in orange) and the rescaled one (in purple).

Same comparison, always one for each cluster, is done for the unknown rescaled layered models vs the generated layered resistivity ones (Fig. 35).



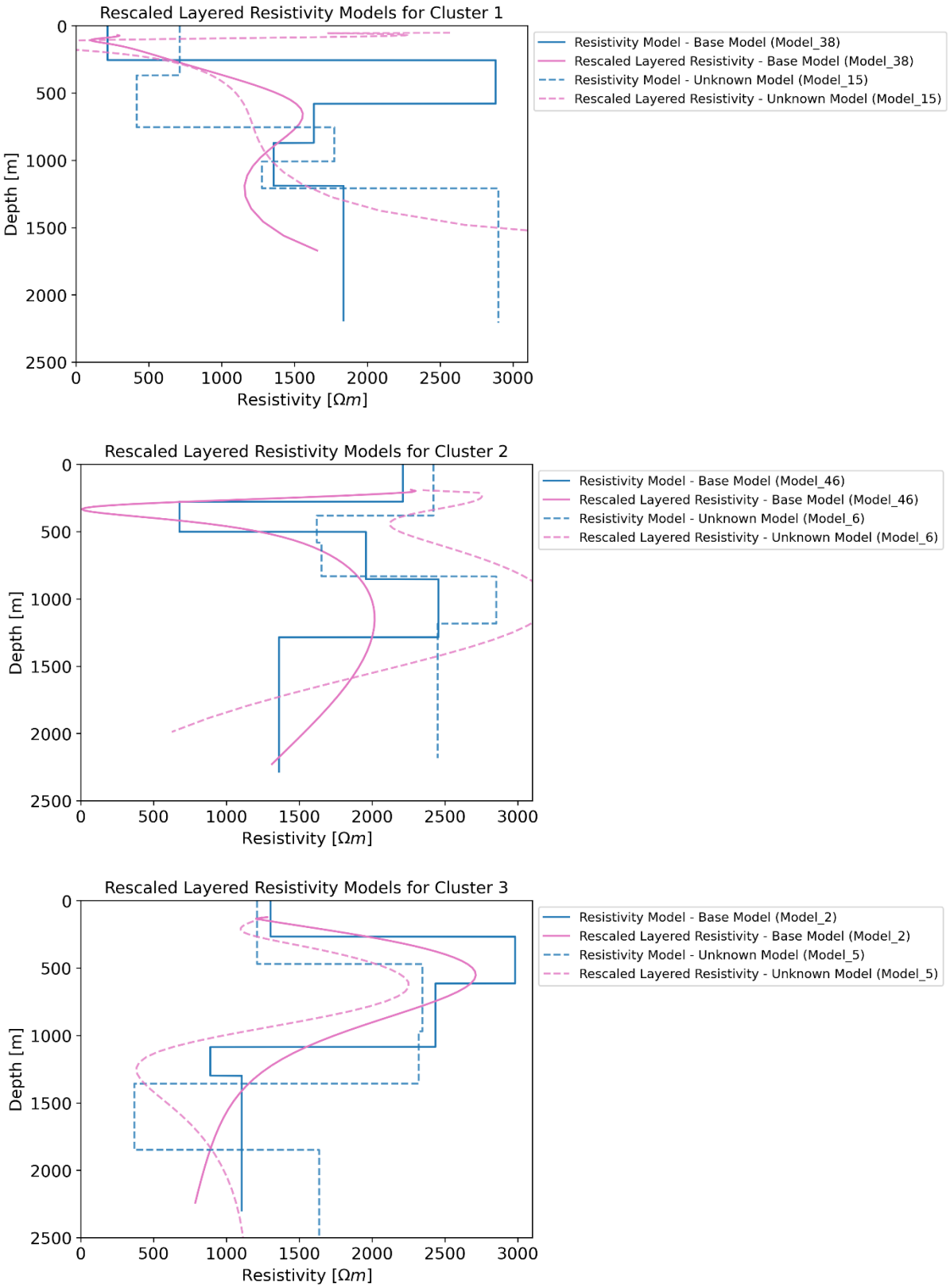


Figure 35 - 4 examples of the layered final results obtained, 1 for each cluster. In each plot are represented, with a solid line, the layered base model (in blue) from which the depth/pseudo-depth rescaling function for the cluster in analysis is obtained and the rescaled one (in purple). On the other hand, with a dashed line, are shown the unknow layered model (in blue) and the rescaled one (in purple), the whole set of rescaled unknow models correspond to the final results of the methodology.

In Fig. 35, the final results (rescaled layered unknown models, dashed purple line) obtained from the application of the reference depth/pseudo-depth rescaling functions, produced by the base known models of each cluster (solid blue line), are compared to the resistivity models generated at the beginning of the test (dashed blue line).

From the graphs, it can be observed that while the results for Clusters 0, 2, and 3 are highly positive (rescaled layered resistivity of the unknown model close to its resistivity model), this is not the case for Cluster 1. This is due to the fact that the results of this cluster are influenced by the presence of outliers, which increase the error in rescaling other models using the rescaling functions produced by these outliers (as shown in the second plot of Fig. 29).

Following the description of the method used in this Thesis, the next chapter will specifically show the results obtained with larger datasets compared to the sample used for explaining the method, consisting of 200 and 1000 curves respectively. Lastly, will be shown a real-case test.

Chapter 5

5. Results

In this chapter, the main results provided from the application of the method explained in the previous one, tested on larger datasets, are collected. The designated algorithm for the tests is k-means, which has proven to be the best algorithm for achieving the lowest errors percentage. The analysis of the results, will begin with the selection of the best combination of parameters, as preannounced in section 4.2.2, but here it will be demonstrated by comparing the results (always in terms of error due to the rescaling of clustered models) given by the combination of different parameters. This will be followed by the results obtained on a synthetic dataset composed of 200 curves, one composed of 1000 and, in the end, final test on a real dataset.

5.1 Best combination of clustering criteria

To find the best parameter combination that provides the lowest error between the models rescaled with a depth/pseudo-depth rescaling function per cluster, and the real one (used to generate the synthetic data), 31 tests were conducted on a dataset consisting of 200 apparent resistivity curves divided in 10 clusters, trying different parameter combinations, the results of which are reported in Tab. 2. In each test, the input matrix of k-means was modified by adding or removing parameters (and therefore dimensions). In the table are shown, for each test, the weighted average errors, computed calculating the mean value across all clusters (taking into account the number of curves in each cluster), and the maximum errors values (whisker) without considering outliers. These findings, led to the selection of the best parameter combination (grade 1). The 12 clustering criteria, for simplicity and clarity of the table, were assigned to a list of letters as follows:

a) Average Resistivity
b) Average Depth
c) Initial Resistivity
d) Final Resistivity
e) Highest Local Maximum
f) Lowest Local Minimum
g) Pseudo-Depth of the Local Maximum
h) Pseudo-Depth of the Local Minimum
i) Number of Gradients
j) Highest Gradient Change

k) Pseudo-Depth of the Highest Gradient Change
l) Ratio between Total Area and Length of the Curve

N. of test	N. of dimensions	Combination of parameters	Avg. Error [%]	Whisker [%]	GRADE
INDIVIDUAL CATEGORY PARAMETERS					
1	2	a + b	5.2%	96%	**2**
2	2	c + d	5.2%	185%	**4**
3	4	e + f + g + h	12.6%	383%	**27**
4	3	i + j +k	25%	3205%	**29**
5	1	l	36.8%	2680%	**31**
MIXED SET OF PARAMETERS					
6	4	a + b + c + d	4.8%	93%	**1**
7	6	a + b + e + f + g + h	6.9%	122%	**13**
8	6	c + d + e + f + g + h	7%	183%	**15**
9	4	i + j + k + l	25.3%	3205%	**30**
10	7	e + f + g + h + i + j + k	9.7%	236%	**25**
11	5	e + f + g + h + l	12.6%	383%	**28**
12	8	e + f + g + h + i + j + k + l	10.3%	397%	**26**
13	5	c + d + i + j + k	6.4%	96%	**9**
14	3	c + d + l	5.4	185%	**7**
15	6	c + d + i + j + k + l	7%	185%	**16**
16	5	a + b + i + j + k	6%	90%	**8**
17	3	a + b + l	5.4%	117%	**5**
18	6	a + b + i + j + k + l	6.2%	170%	**10**
19	9	a + b + e + f + g + h + i + j + k	7.9%	222%	**23**
20	7	a + b + e + f + g + h + l	7.3%	122%	**18**
21	10	a + b + e + f + g + h + j + i + k + l	8.1%	236%	**24**
22	9	c + d + e + f + g + h + i + j + k	7.5%	236%	**21**
23	7	c + d + e + f + g + h + l	7.1%	190%	**17**
24	10	c + d + e + f + g + h + i + j + k + l	7.7%	236%	**22**
25	8	a + b + c + d + e + f + g + h	6.5%	146%	**12**
26	7	a + b + c + d + i + j + k	5.5%	93%	**6**
27	5	a + b + c + d + l	5.1%	99%	**3**
28	8	a + b + c + d + i + j + k + l	6.2%	185%	**11**
29	11	a + b + c + d + e + f + g + h + i + j + k	7.3%	190%	**19**
30	9	a + b + c + d + e + f + g + h + l	6.9%	146%	**14**
31	12	a + b + c + d + e + f + g + h + i + j + k + l	7.5%	190%	**20**

Table 2 - Selection of the best combination of parameters. Among the 31 different combinations, the test number 6 is selected as the best result, as it shows the least value of avg. error and max. error (whisker) percentages.

As shown in the table above, the results that ensure the lowest possible error, emerge from the combination of resistivity parameters (1st category, section 4.2.1). Specifically, most combinations containing initial & final resistivity, avg. resistivity & average depth, yielded average errors around

5% on the tested dataset, with peaks around 90%. For this reason, combining these 4 parameters resulted in the best outcome (Test 6, highlighted in the table). Different outputs has occurred for the parameters of gradients and ratio area/length, which, although characterizing the curves, did not prove useful to the cause with average errors above 15-20%.

After confirming the final set of parameters, the results of k-means with those criteria, on the same dataset of 200 curves used for error calculation in 5.1, will be presented.

5.2 Results with 200 synthetic data

As explained in section 4.1, the first step of the method is the random generation of 1D resistivity models, according to the limits shown in Tab.1. In this case, 200 models were generated, which will not be shown here due to the low comprehensibility of the plot. Subsequently, through the implementation of the Python routine "empymod," 200 resistivity curves were obtained. The dataset is reported in Fig. 36 as a function of pseudo-depth.

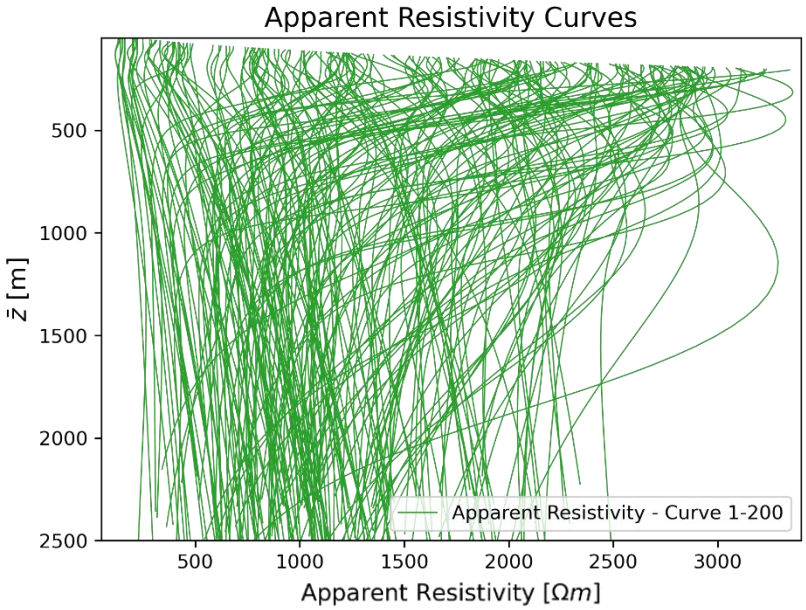


Figure 36 - 200 apparent resistivity curves dataset, obtained by "empymod" routine and defined as a function of pseudo-depth.

At this point, the Elbow and Silhouette methods are tested to establish the optimal number of clusters for this dataset, which is set equal to 10.

Thus, the initial and final resistivity values, average resistivity, and average depth are calculated to compose the input matrix for k-means. In Fig. 37, the results of k-means on this dataset are illustrated. As visible from the legend, 10 colours have been randomly generated to define the 10 clusters.

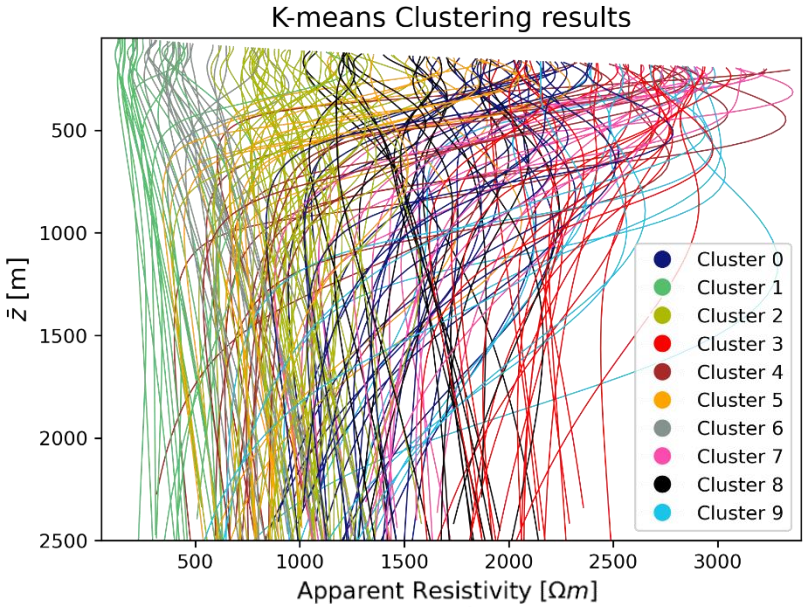
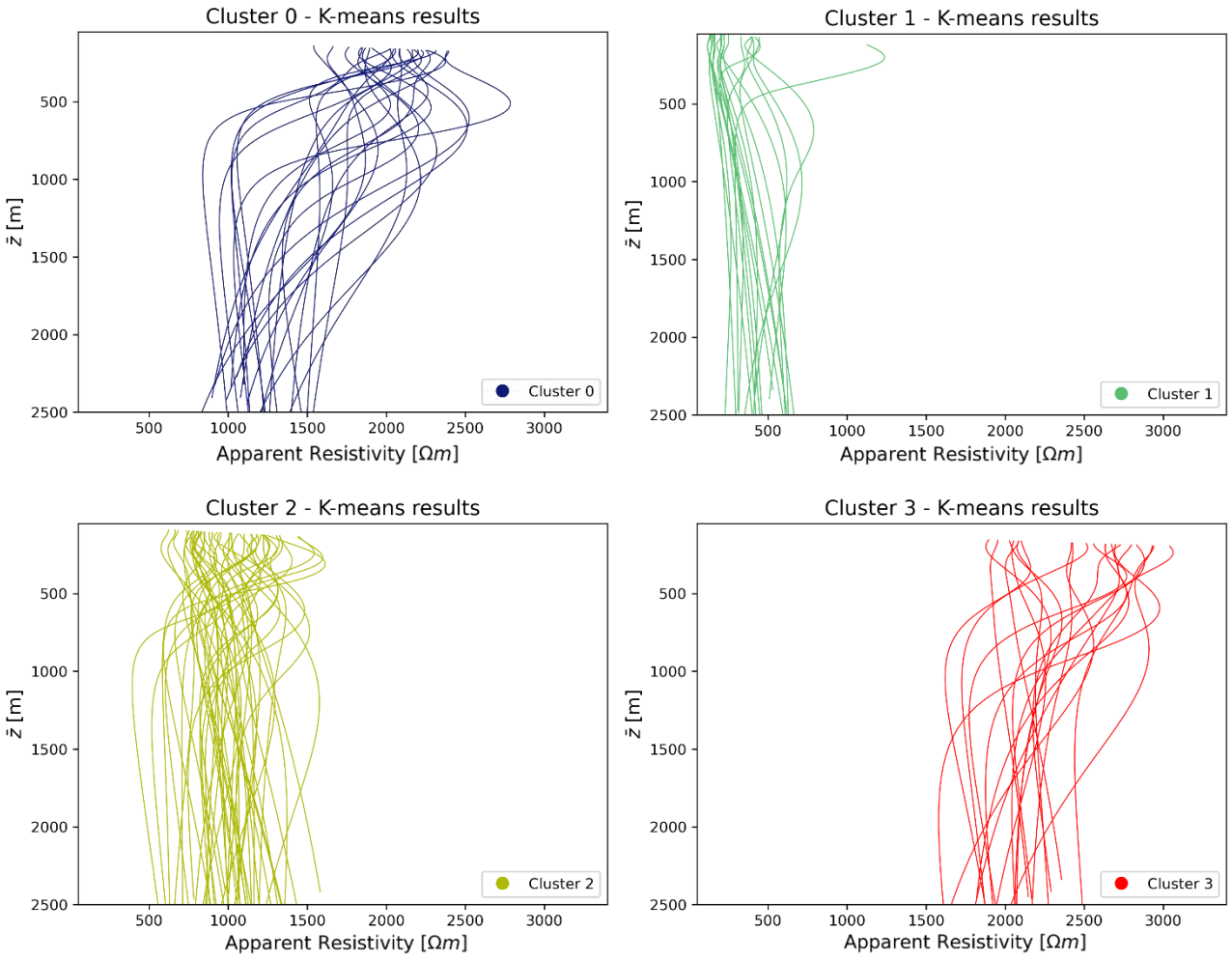


Figure 37 - K-means results for the dataset of Fig.36. Apparent resistivity curves are divided in 10 clusters, shown in the legend.

Fig. 38 shows the curves of apparent resistivity for the 10 clusters, with each individual subplot that illustrate the curves that belong to the cluster.



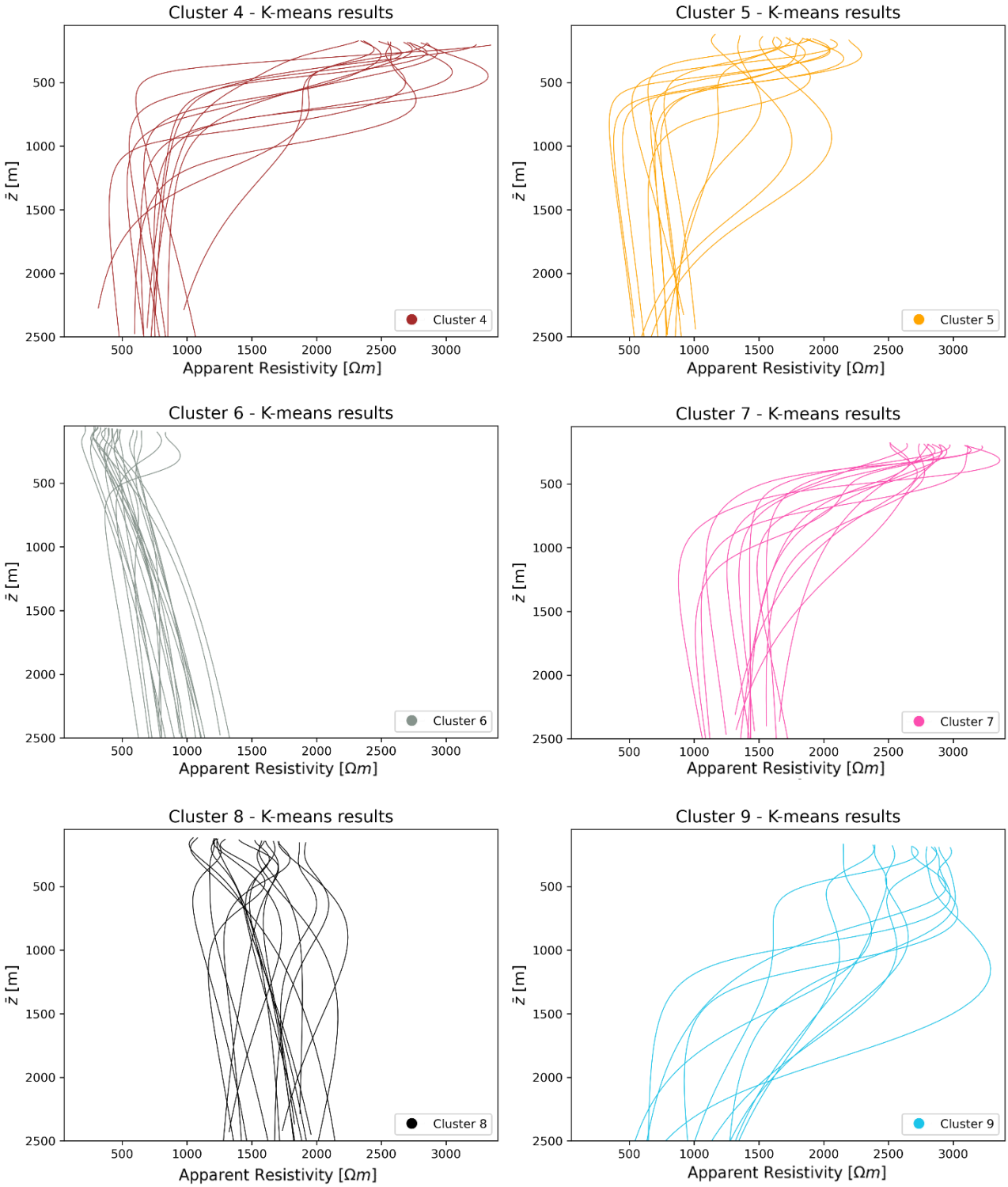


Figure 38 - Same results of Fig.34 but each plot represents the set of curves that belong to a single cluster.

After obtaining the 10 clusters of apparent resistivity curves, the test on the depth/pseudo-depth rescaling functions is performed. They are divided into their respective clusters related to the clustered curves. This check is done to qualitatively assess the efficiency of clustering, so defining a good clustering if similar rescaling functions are grouped in the same cluster. Fig. 39 displays the 200 depth/pseudo-depth rescaling functions related to the dataset of Fig. 36.

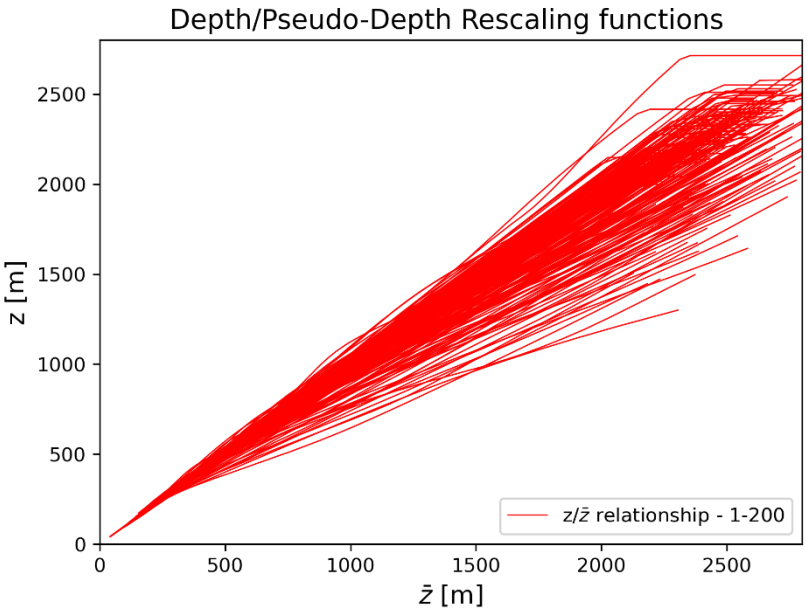


Figure 39 - Set of 200 depth/pseudo-depth rescaling functions, related to the apparent resistivity curves of Fig.36.

Instead, Fig. 40 represents the same rescaling functions but divided into the 10 clusters.

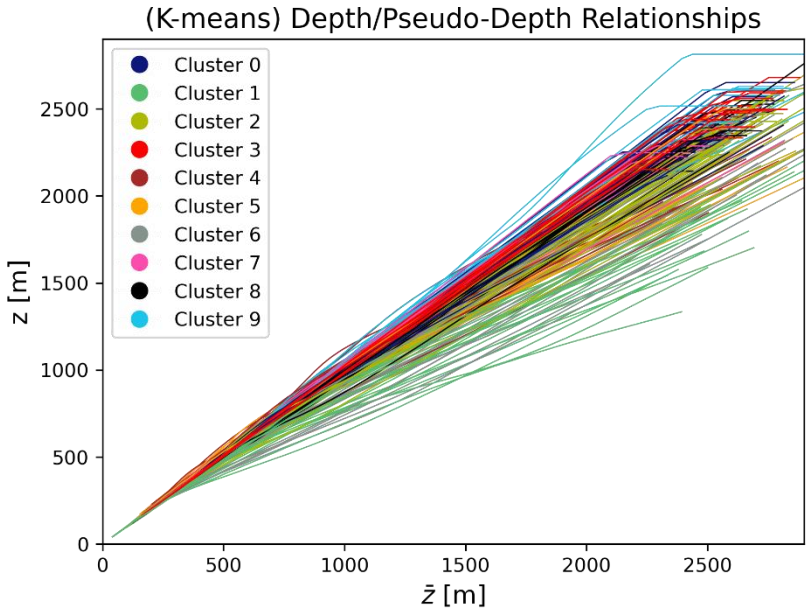
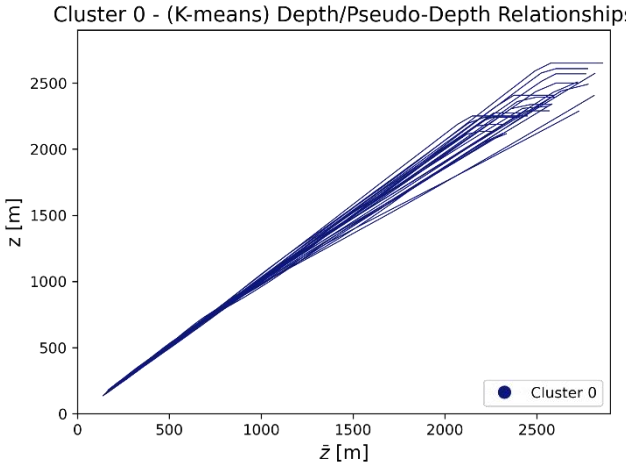
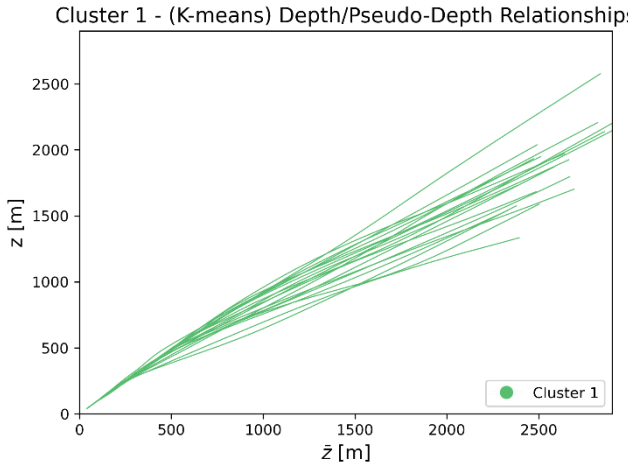
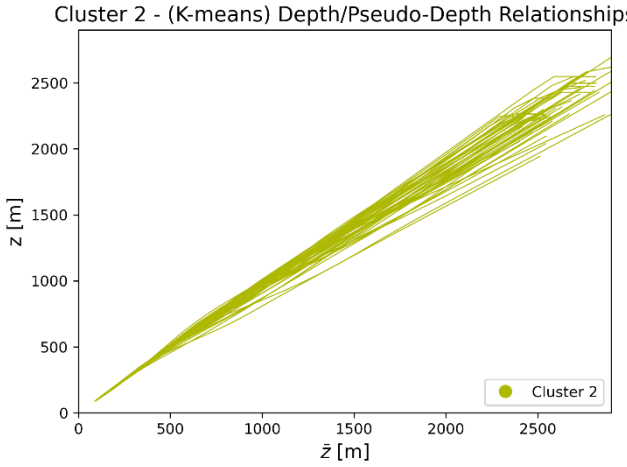


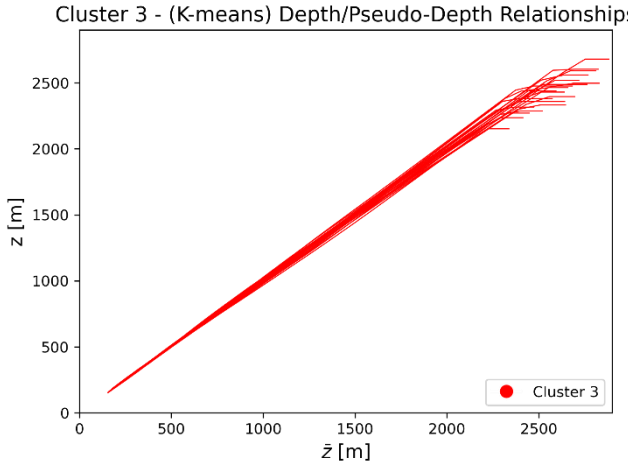
Figure 40 - K-means results of Fig. 37, applied on the set of depth/pseudo depth rescaling functions. These relationships are divided in 10 clusters, taking same colours for each cluster obtained with the related apparent resistivity data.

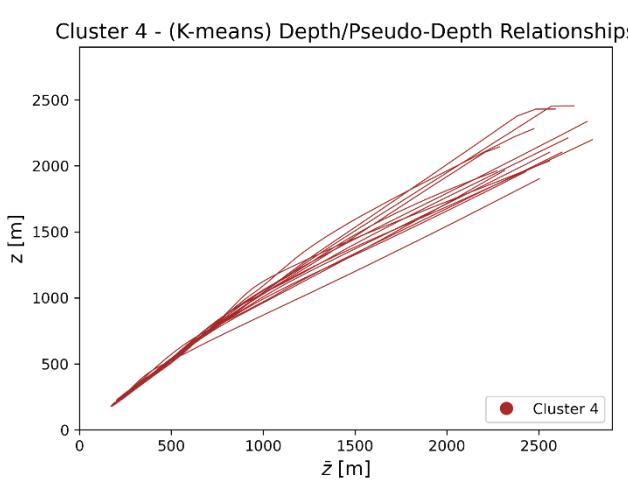
From Fig. 41, which represents the rescaling functions divided by clusters, the results given by the clustering can be perceived. Among them, there are clusters that are nearly perfect, such as cluster no. 3 or no. 7, where the rescaling functions are very close to each other, and so, low errors due to cross-rescaling are expected. At the same time, there are worse clusters such as no. 1 or no. 6, with more distant rescaling functions, and higher errors are expected from them.

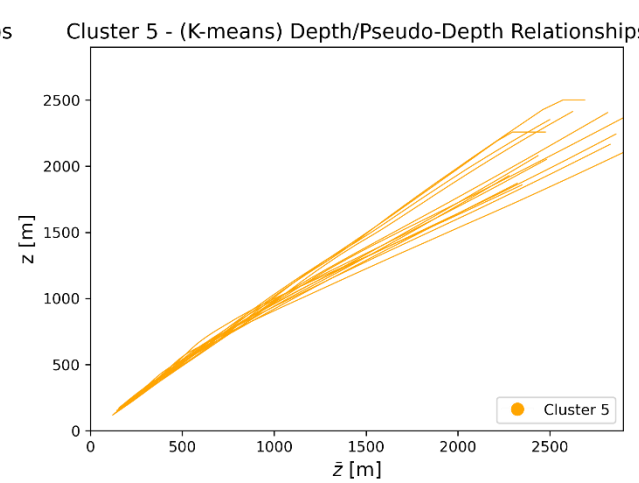

Cluster 0 - (K-means) Depth/Pseudo-Depth Relationships
z [m]
$\bar{z}$ [m]
Cluster 0

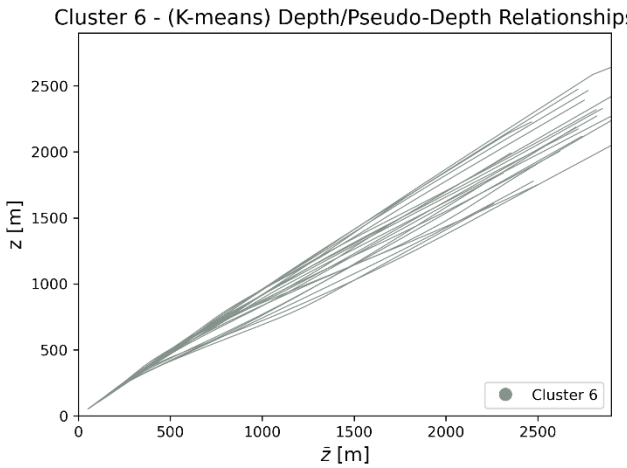

Cluster 1 - (K-means) Depth/Pseudo-Depth Relationships
z [m]
$\bar{z}$ [m]
Cluster 1

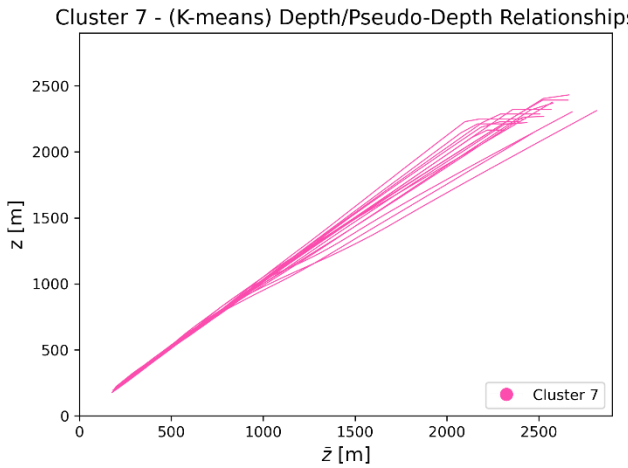

Cluster 2 - (K-means) Depth/Pseudo-Depth Relationships
z [m]
$\bar{z}$ [m]
Cluster 2


Cluster 3 - (K-means) Depth/Pseudo-Depth Relationships
z [m]
$\bar{z}$ [m]
Cluster 3


Cluster 4 - (K-means) Depth/Pseudo-Depth Relationships
z [m]
$\bar{z}$ [m]
Cluster 4


Cluster 5 - (K-means) Depth/Pseudo-Depth Relationships
z [m]
$\bar{z}$ [m]
Cluster 5


Cluster 6 - (K-means) Depth/Pseudo-Depth Relationships
z [m]
$\bar{z}$ [m]
Cluster 6


Cluster 7 - (K-means) Depth/Pseudo-Depth Relationships
z [m]
$\bar{z}$ [m]
Cluster 7

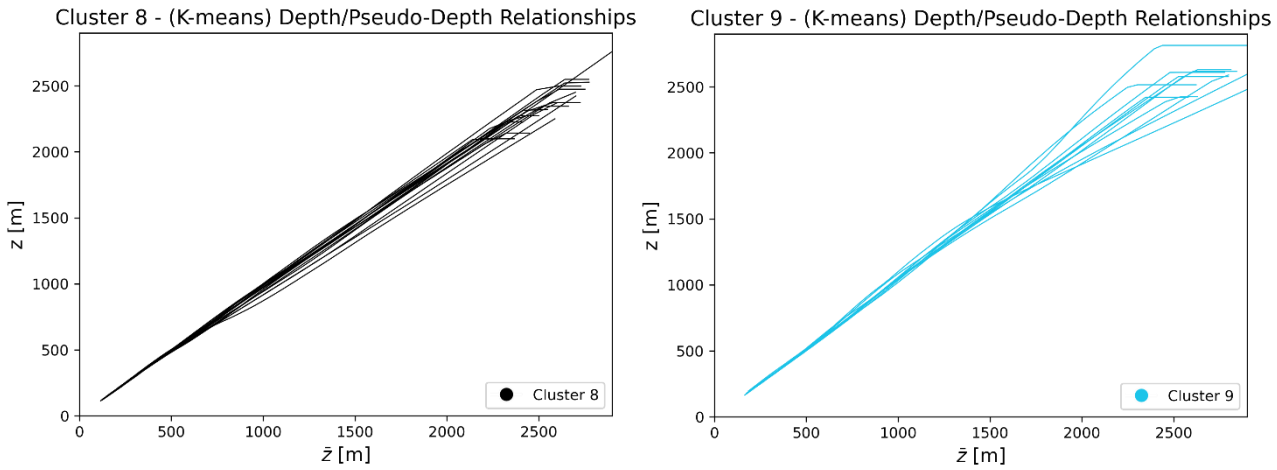
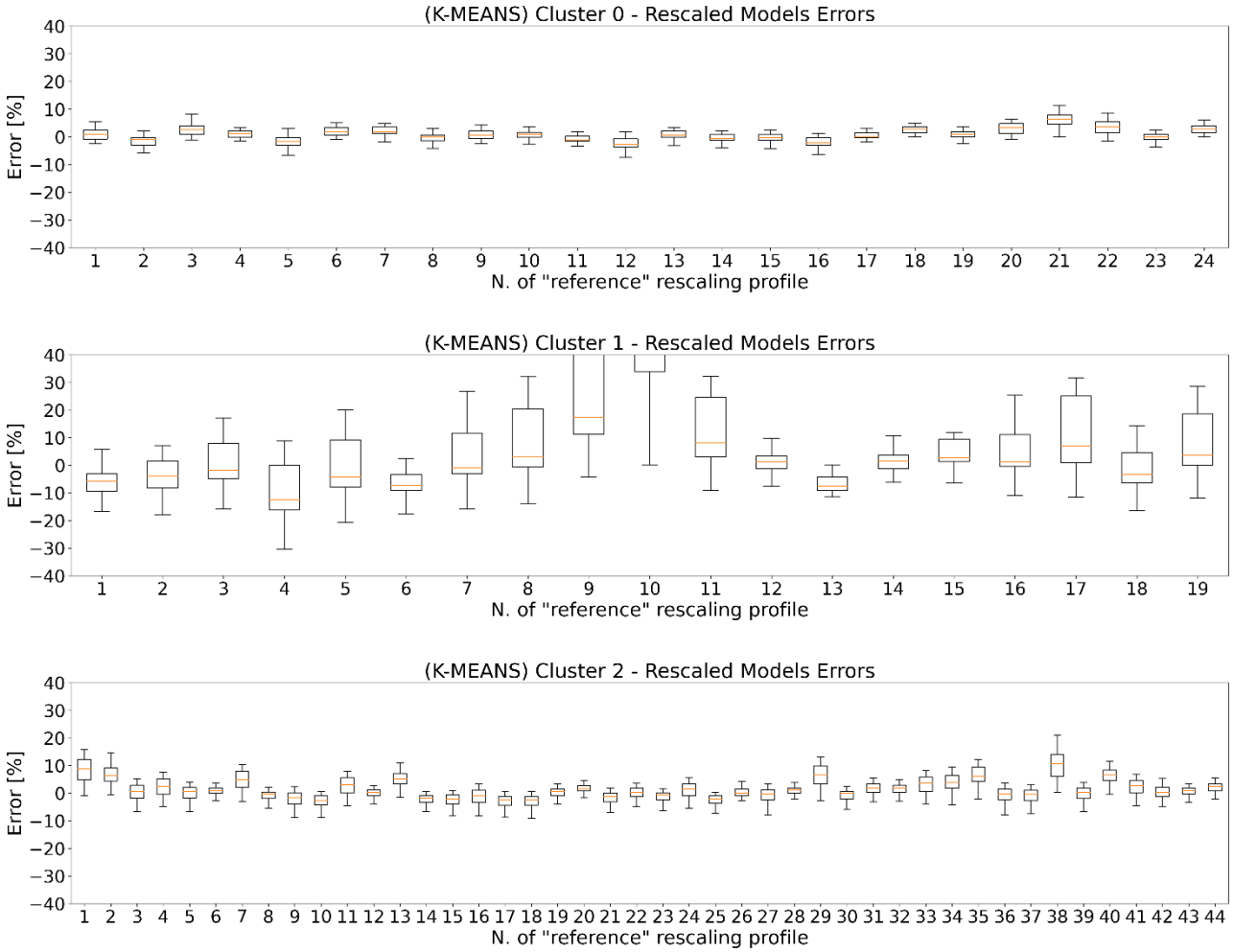
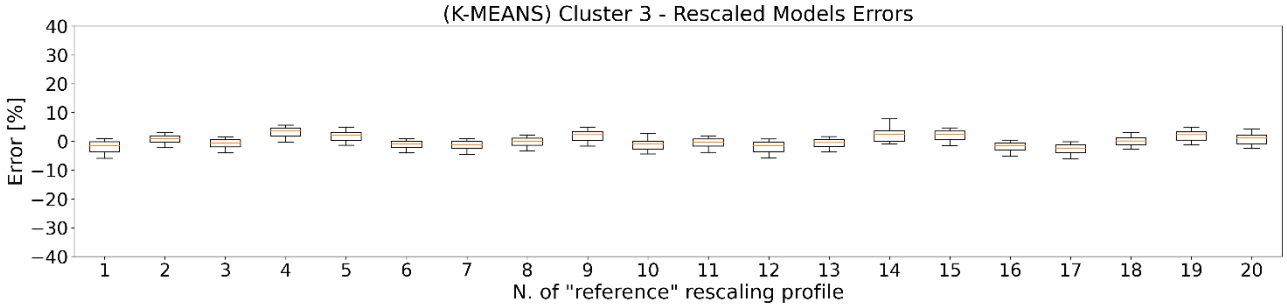
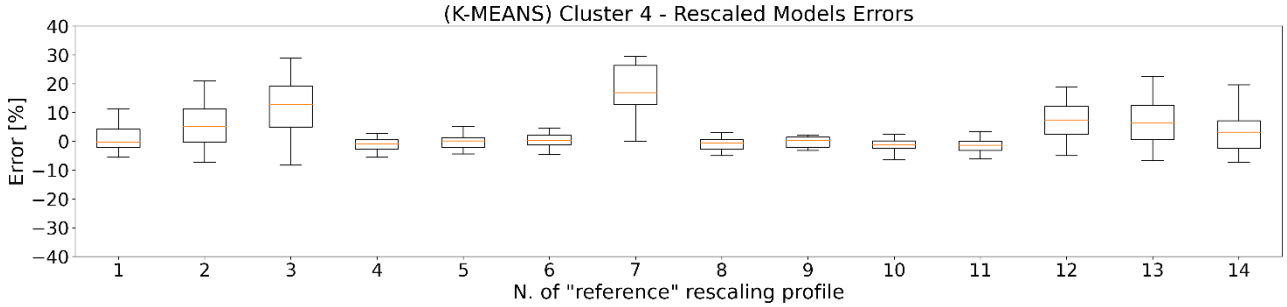


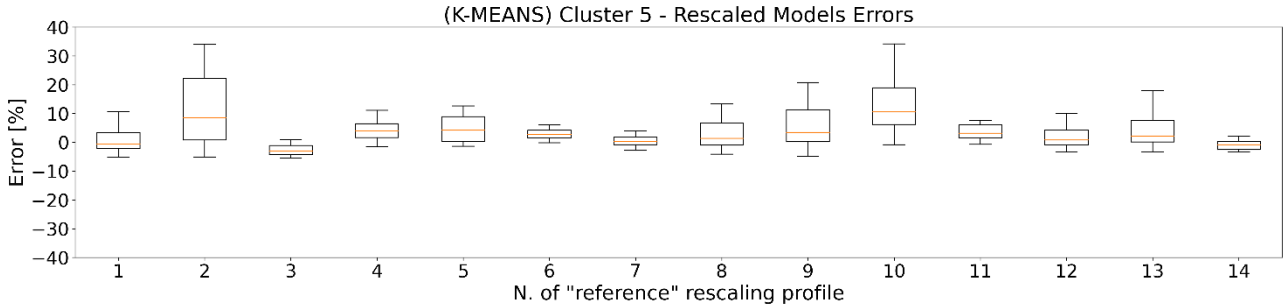
Figure 41 - Same results of Fig.40 but each plot represents the set of relationships that belong to a single cluster.

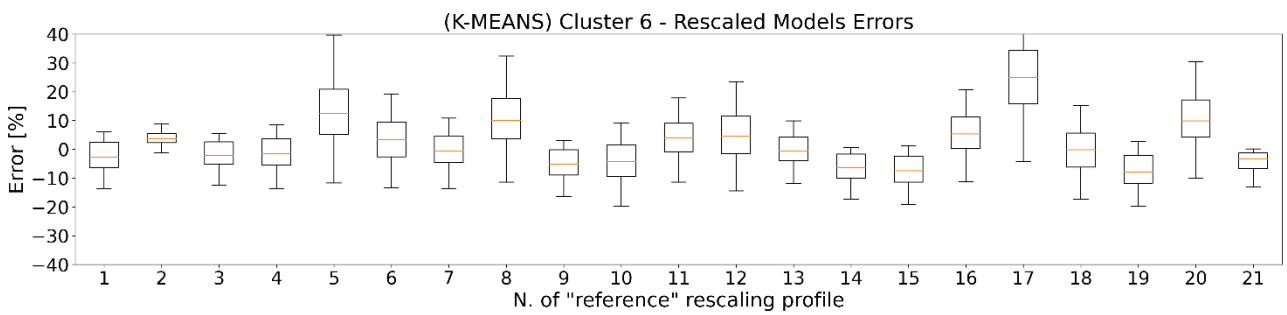
The last step, before obtaining the final results from the cross-rescaling of the different clusters, is to calculate the error due to the cross-rescaling of each cluster computed by in turn using a reference depth-pseudo depth to rescale all the cluster data at each iteration. In Fig. 42, these results are represented in the form of box plots. In numerical terms, they are the same as those of Test n.6 in Table 2, with an average error of 4.8% and a maximum error of 93% due to outlier n.10 in cluster 1.

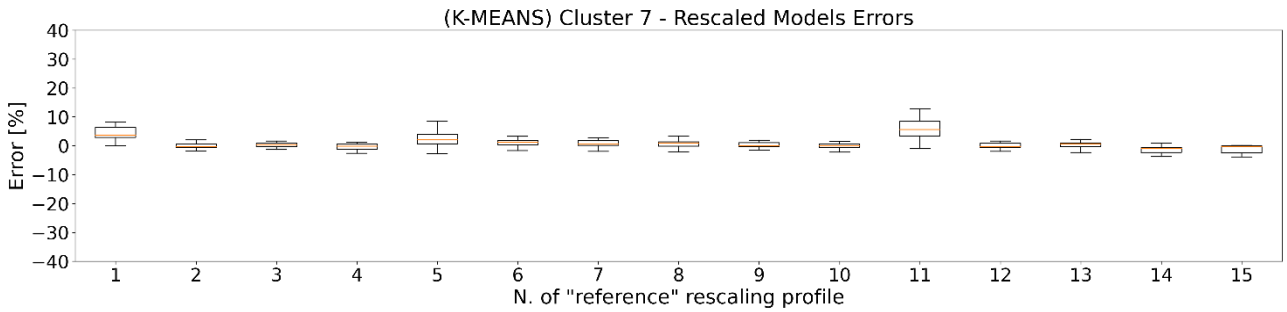


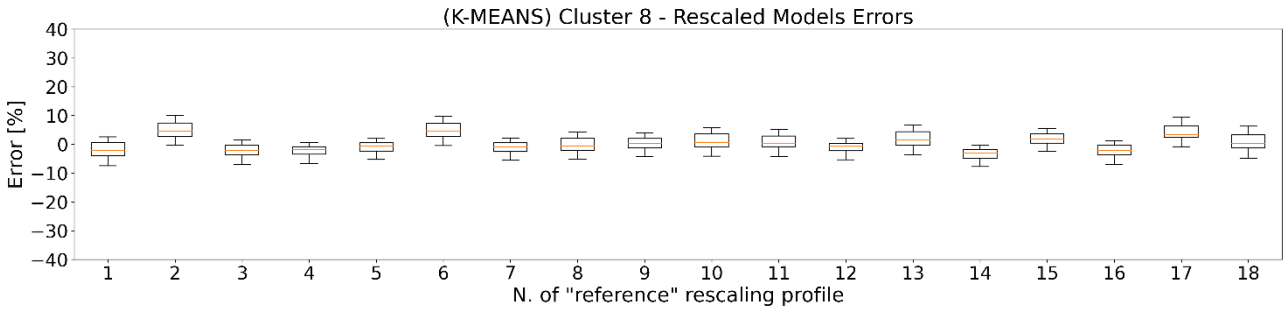

(K-MEANS) Cluster 3 - Rescaled Models Errors
Error [%]
N. of "reference" rescaling profile


(K-MEANS) Cluster 4 - Rescaled Models Errors
Error [%]
N. of "reference" rescaling profile


(K-MEANS) Cluster 5 - Rescaled Models Errors
Error [%]
N. of "reference" rescaling profile


(K-MEANS) Cluster 6 - Rescaled Models Errors
Error [%]
N. of "reference" rescaling profile


(K-MEANS) Cluster 7 - Rescaled Models Errors
Error [%]
N. of "reference" rescaling profile


(K-MEANS) Cluster 8 - Rescaled Models Errors
Error [%]
N. of "reference" rescaling profile

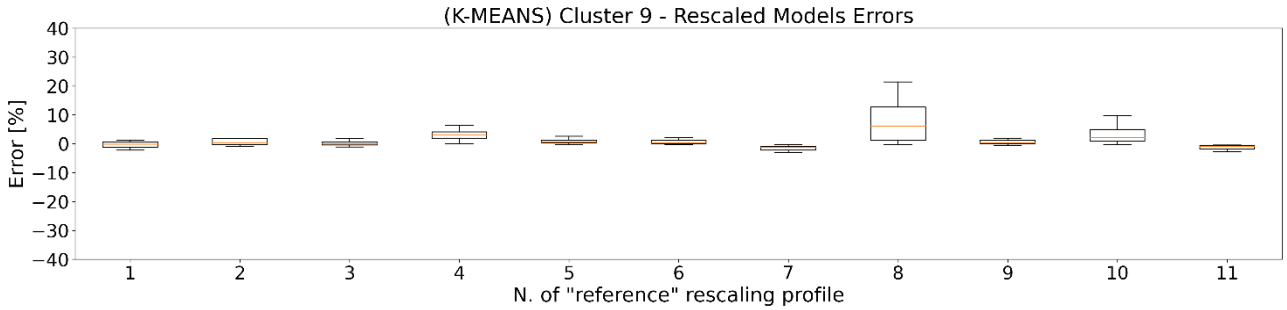


Figure 42 - Error box-plots of the 10 detected clusters by k-means algorithm. Global avg. errors found is 4.8% confirming the values on Tab.2. Instead maximum error is around 90% (due to the presence of the outlier 10 of the cluster 1).

Apart from the overall results, the outcomes obtained for each cluster confirm the predictions given by the clustering of the depth/pseudo-depth rescaling functions, with clusters 1 and 6 characterized by higher average errors (14% and 8.4%, respectively), while clusters 3 and 7 show the best findings (average error 1.8% for both).

An example for each of these 4 clusters is depicted in Fig. 43, where the unknown rescaled cumulative results (dashed purple line) obtained from the application of the reference depth/pseudo-depth rescaling functions, produced by the base known models of each cluster (solid orange line), are compared to their respective true cumulative resistivity models generated (dashed orange line).

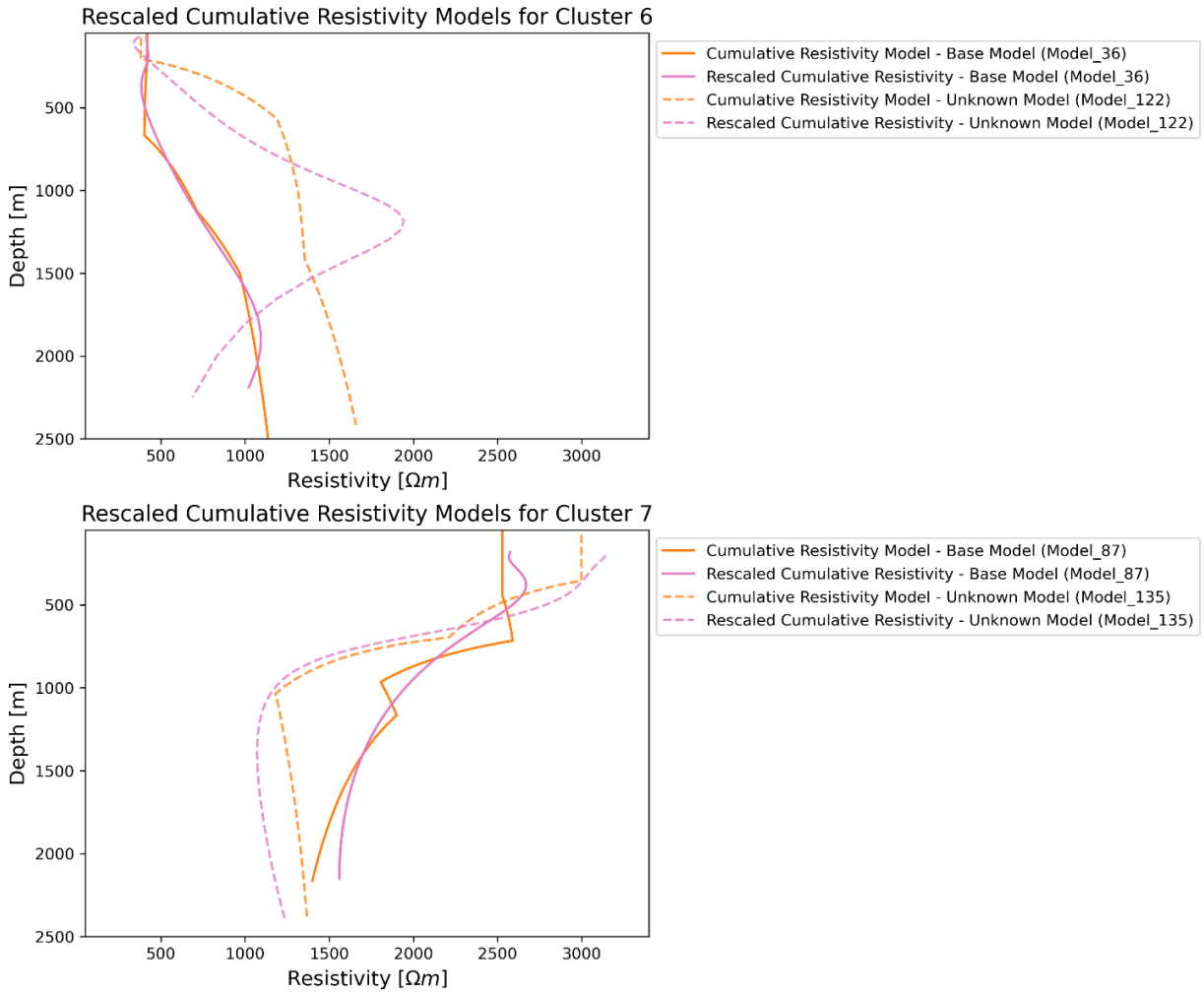
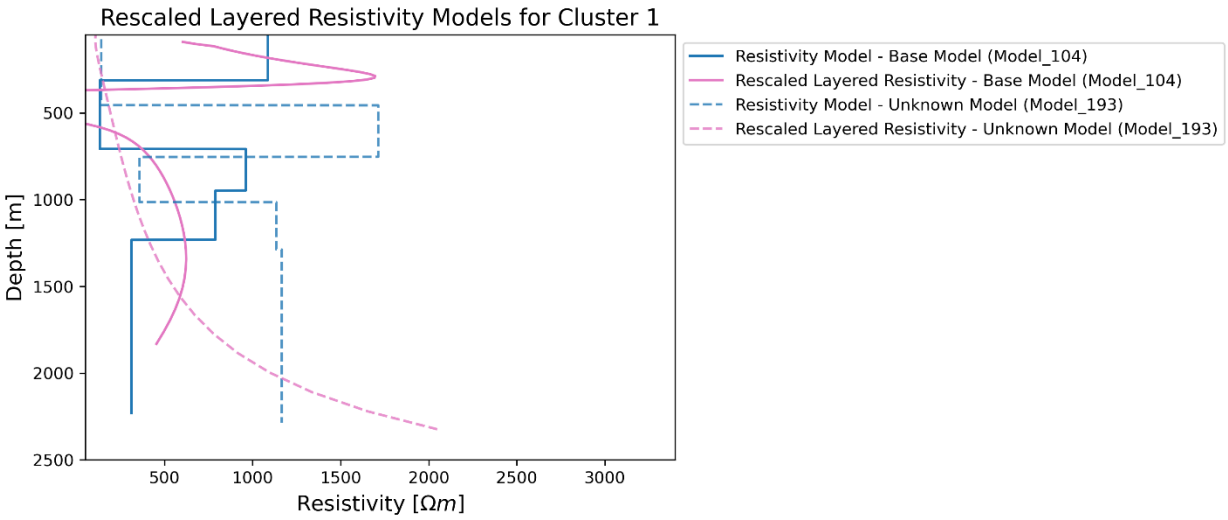


Figure 43 - 4 examples of the cumulative results obtained, 1 for each cluster (number 1,3,6 and 7). In each plot are represented, with a solid line, the cumulative base model (in orange) from which the depth/pseudo-depth rescaling function for the cluster in analysis is obtained and the rescaled one (in purple). On the other hand, with a dashed line, are shown the unknow cumulative model (in orange) and the rescaled one (in purple), the whole set of rescaled unknow models correspond to the final results of the methodology.

Same comparison, for same clusters, is done for the unknown rescaled layered models vs the generated layered resistivity ones (Fig. 44).



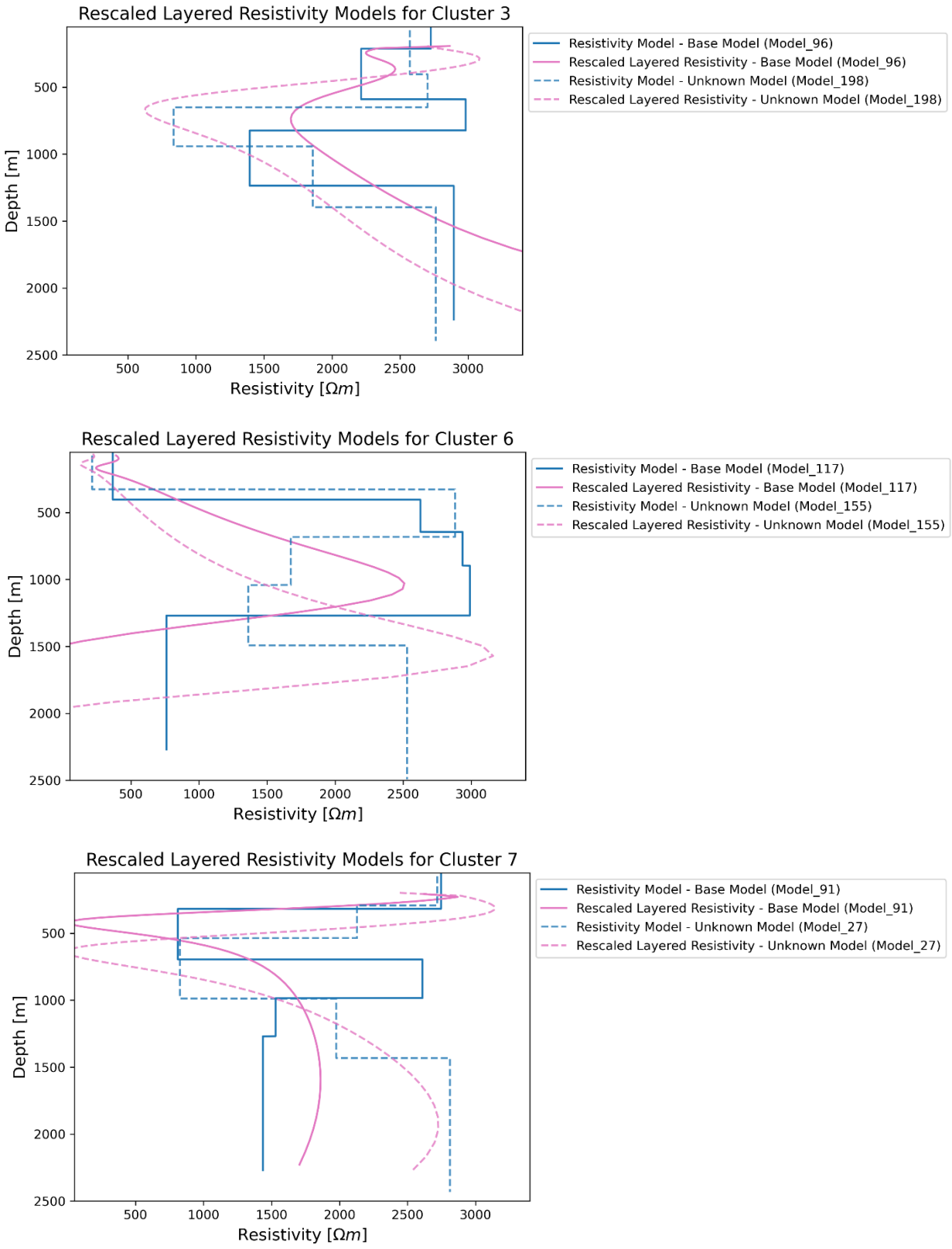


Figure 44 - 4 examples of the final layered results obtained, 1 for each cluster (number 1,3,6 and 7). In each plot are represented, with a solid line, the layered base model (in blue) from which the depth/pseudo-depth rescaling function for the cluster in analysis is obtained and the rescaled one (in purple). On the other hand, with a dashed line, are shown the unknow layered model (in blue) and the rescaled one (in purple), the whole set of rescaled unknow models correspond to the final results of the methodology.

In this case the final results (rescaled layered unknown models, dashed purple line) obtained from the application of the reference depth/pseudo-depth rescaling functions, produced by the base known models of each cluster (solid blue line), are compared to the resistivity models generated at the beginning of the test (dashed blue line).

As demonstrated by the 4 graphs of Fig. 43 and Fig. 44, both in cumulative and layered domain, the trends of the rescaled unknown models of clusters 3 and 7 appear to be very similar to their own generated resistivity models, while the other two clusters example show larger discrepancies. These worst cases are exceptions, confirming the validity of the model given by the very low average error <5%, obtained knowing only 10 out of the 200 rescaled models (5% of the models).

To confirm the validity of the proposed methodology, a larger dataset consisting of 1000 apparent resistivity data was examined. The results are reported in the next sub-chapter.

5.3 Results with 1000 synthetic data

With the same approach used for the dataset analysed previously, 1000 models were randomly generated, and subsequently, through "empymod," the 1000 apparent resistivity curves, represented in Fig. 45, were obtained.

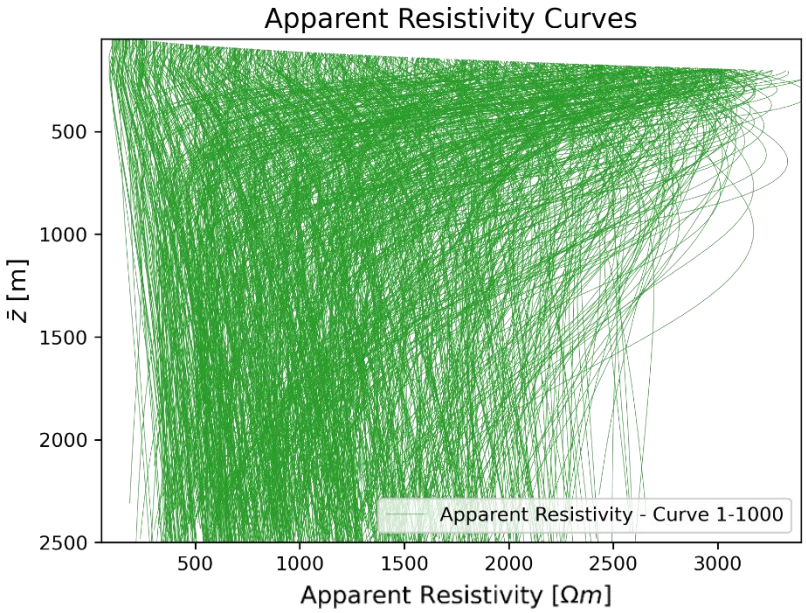


Figure 45 - 1000 apparent resistivity curves dataset, obtained by "empymod" routine and defined as a function of pseudo-depth.

Then, k-means is run on this dataset keeping the number of clusters equal to 10, to observe how the algorithm performs with larger datasets while keeping the number of clusters constant, which in this case corresponds to 1% of the dataset (10 out of 1000). In Fig. 46, the results of k-means on this dataset are illustrated with the same colours as the previous dataset.

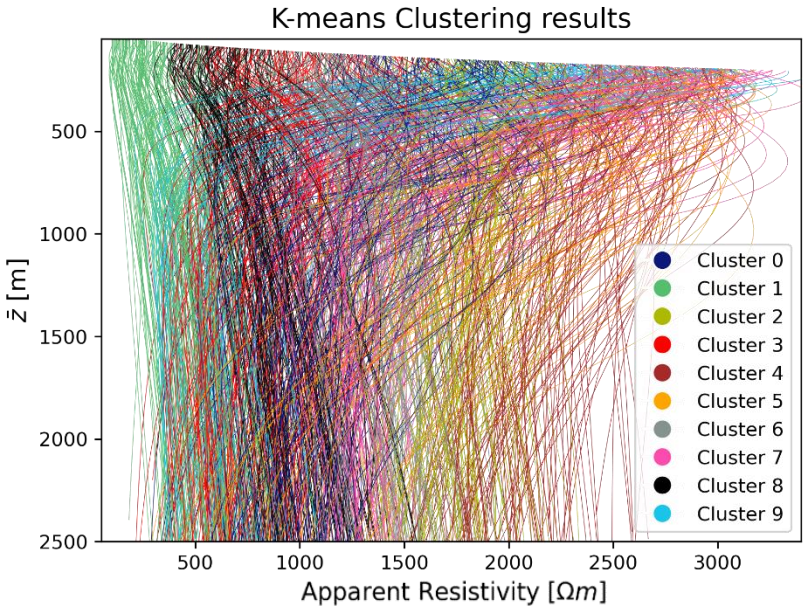
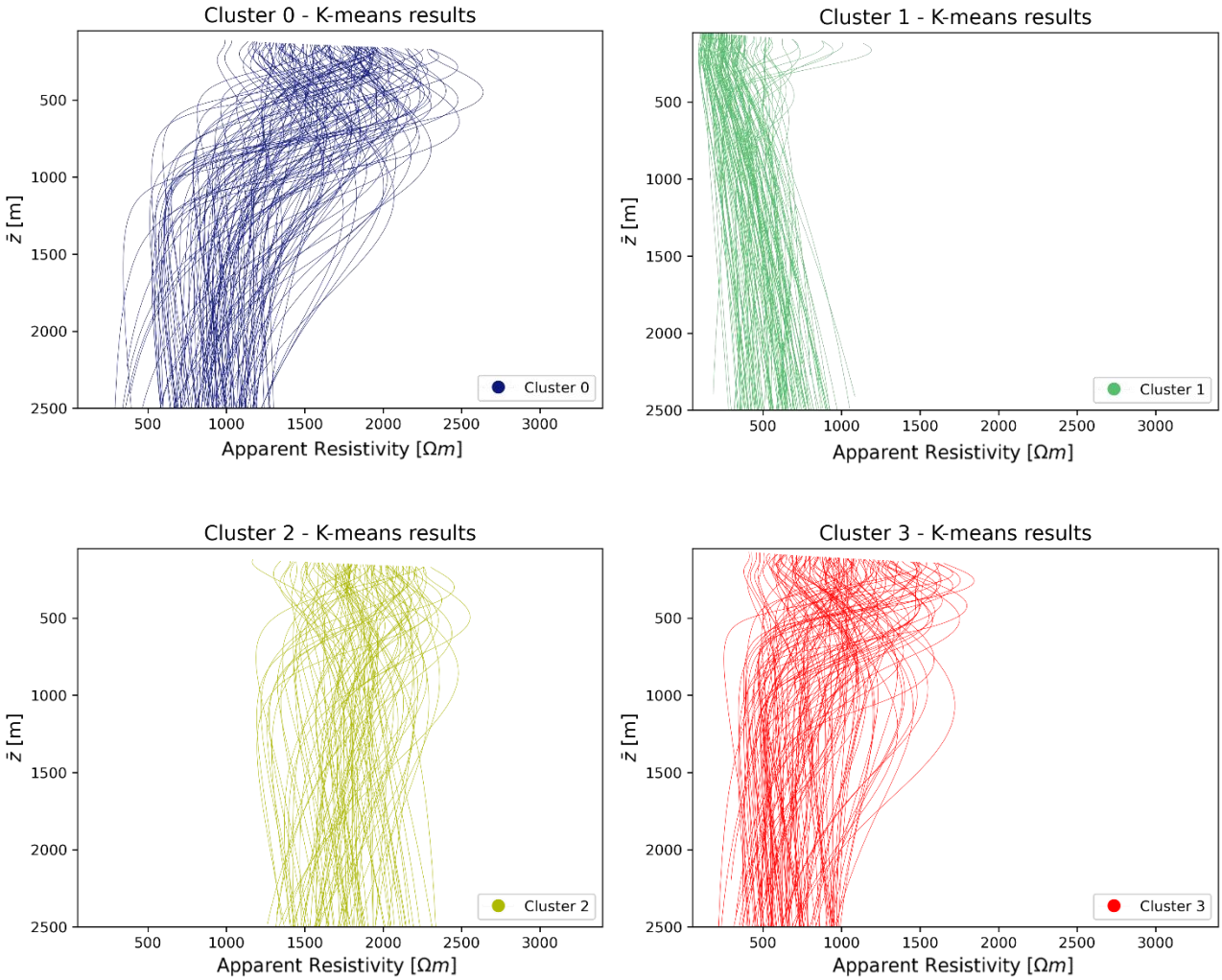


Figure 46 - K-means results for dataset of Fig.45. Apparent resistivity curves are divided in 10 clusters, shown in the legend.

Fig. 47 shows the curves of apparent resistivity for the 10 clusters, with each individual subplot that illustrate the curves that belong to the cluster.



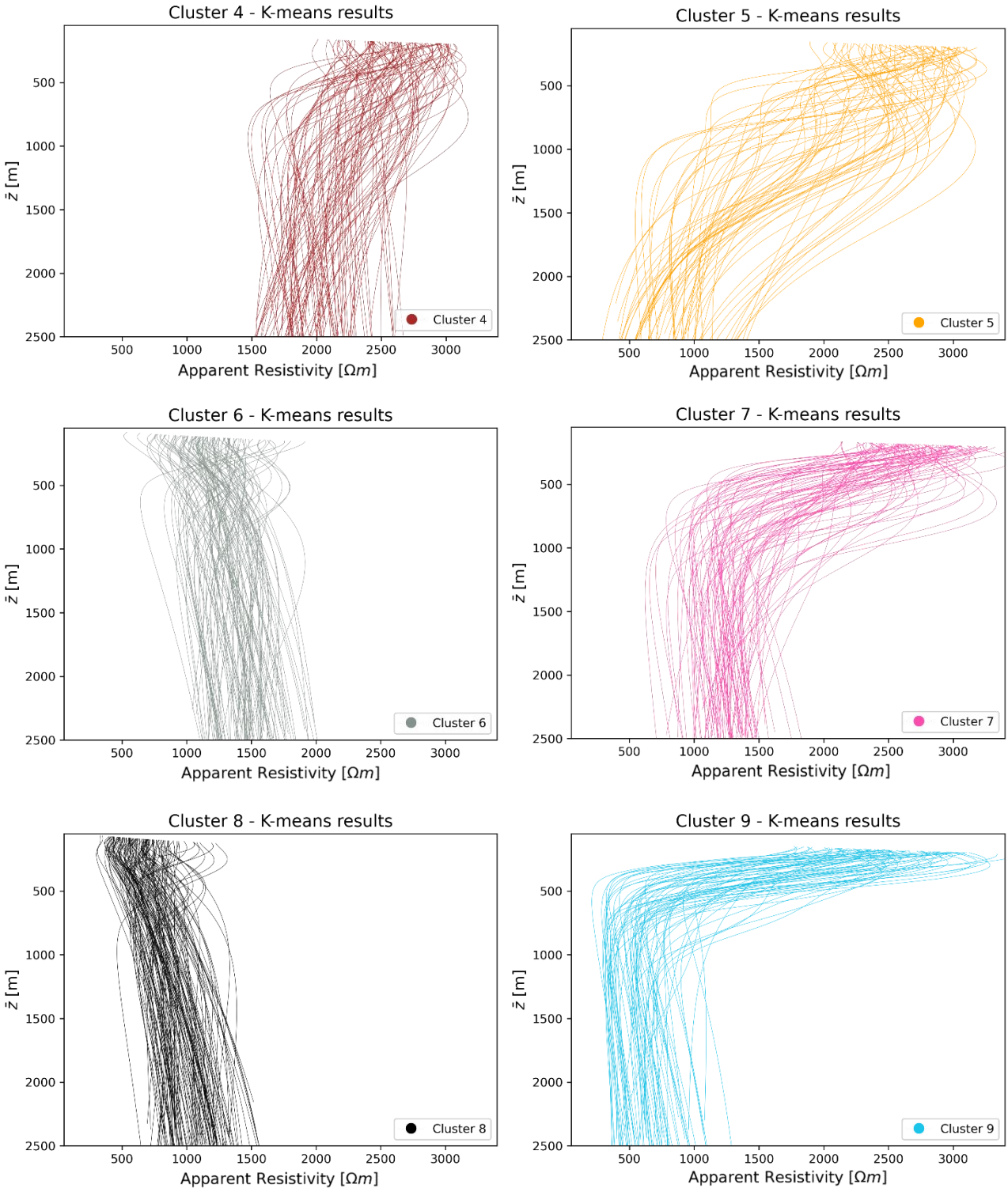


Figure 47 - Same results of Fig.46 but each plot represents the set of curves that belong to a single cluster.

Fig. 48 displays the 1000 depth/pseudo-depth rescaling functions related to the dataset of Fig. 45.

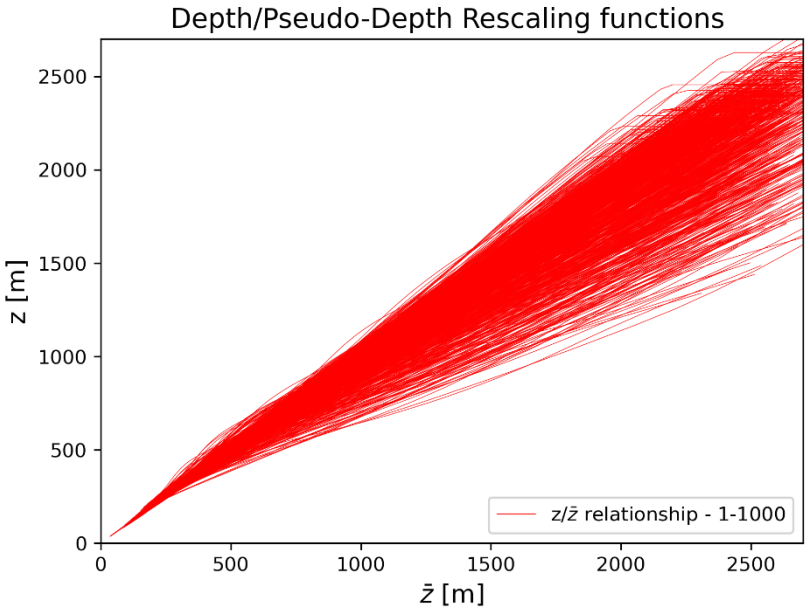


Figure 48 - Set of 1000 depth/pseudo-depth rescaling functions, related to the apparent resistivity curves of Fig.45.

Fig.49 shows the depth/pseudo-depth rescaling functions divided into their respective clusters related to the clustered curves, to qualitatively assess the efficiency of clustering.

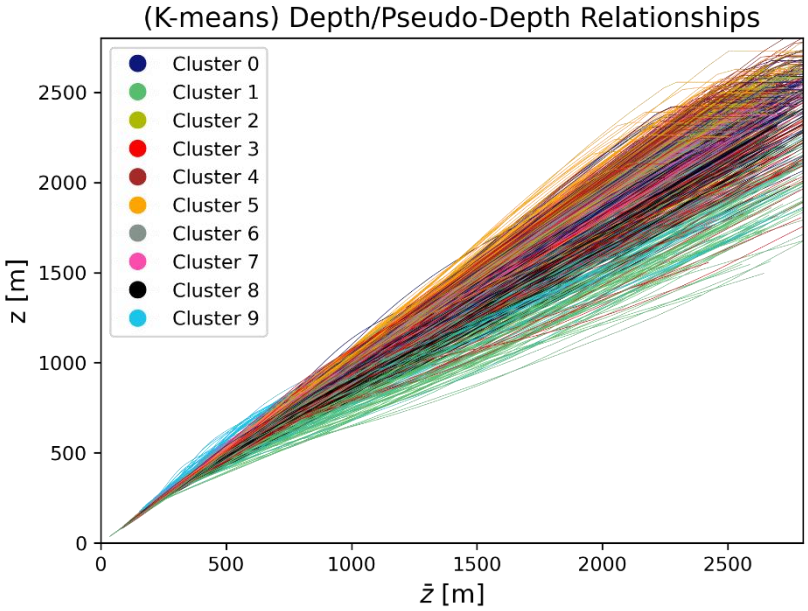
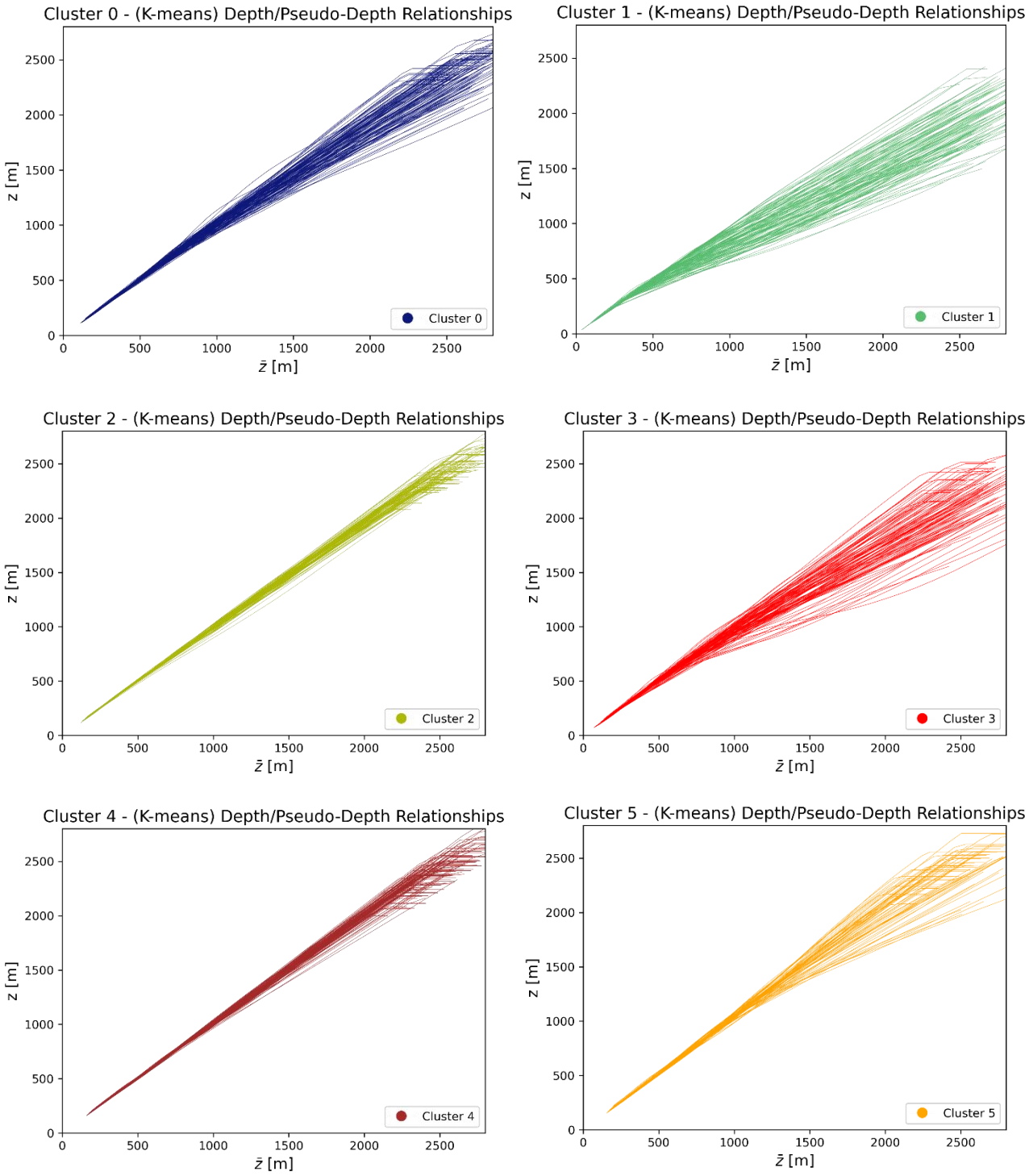


Figure 49 - K-means results of Fig. 46, applied on the set of depth/pseudo depth rescaling functions. These relationships are divided in 10 clusters, taking same colours for each cluster obtained with the related apparent resistivity data.

From Fig. 50, which represents the rescaling functions divided by clusters, the results given by the clustering can be estimated. Among them, there are clusters that are nearly perfect, such as cluster no. 2 or no. 4, where the rescaling functions are very close to each other, and low errors due to

cross-rescaling are expected. At the same time, there are worse clusters such as no. 1 or no. 9, with more distant rescaling functions, and higher errors are expected from them.



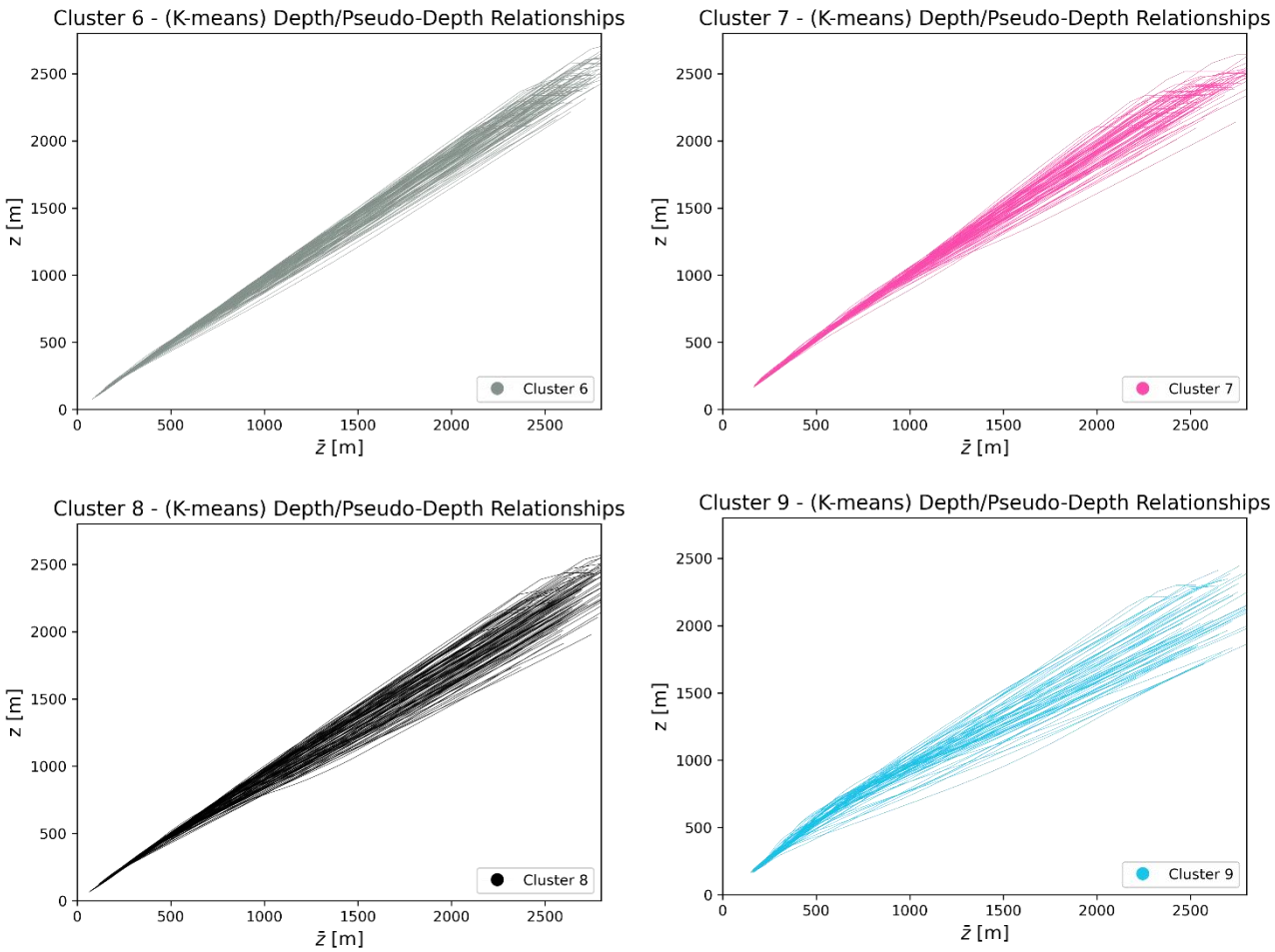


Figure 50 - Same results of Fig.45 but each plot represents the set of relationships that belong to a single cluster.

The last step, before obtaining the final results from the cross-rescaling of the different clusters, is to calculate the error due to cross-rescaling of each cluster computed by in turn using a reference depth-pseudo depth to rescale all the cluster data at each iteration.

In this case, the errors are not represented by the usual box plots, as they were too confusing due to the high number of curves in each cluster, but are summarized in Tab. 3. As visible from the table, the application of this methodology yields excellent results even with a dataset composed of 1000 curves divided into 10 clusters. In fact, a global average error of 5.7% and a maximum error of around 170% due to the whisker of cluster 1 are obtained.

	Cluster 0	**Cluster 1**	**Cluster 2**	**Cluster 3**	**Cluster 4**	**Cluster 5**	**Cluster 6**	**Cluster 7**	**Cluster 8**	**Cluster 9**
number of curves	124	120	90	114	107	65	103	92	118	67
avg error [%]	3.9%	13.5%	1.8%	7.9%	1.5%	6%	3.8%	2.6%	5.3%	11.6%
whisker [%]	44.7%	174%	13%	124%	8.5%	164%	32.7%	30.1%	41.4%	89.4%

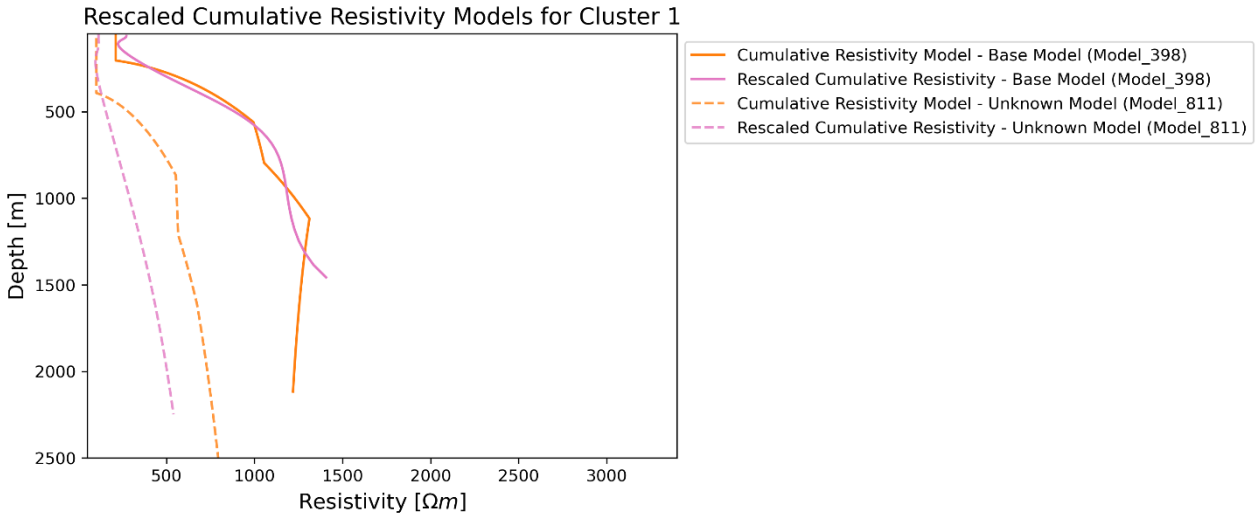
GLOBAL CLUSTERING RESULTS	
Avg. error of clustering [%]	**5.7%**
Whisker of clustering [%]	**174%**

Table 3 - Error box-plots of the 10 detected clusters by k-means algorithm. In the first table are reported avg. errors and whiskers of each single cluster. Below are shown global results; avg. errors found is 5.7%. Instead maximum error is around 170% (due to the whisker of the cluster 1).

Apart from the overall results, the outcomes obtained for each cluster confirm the predictions given by the clustering of the depth/pseudo-depth rescaling functions, with clusters 1 and 9 characterized by higher average errors (13.5% and 11.6%, respectively), while clusters 2 and 4 show the best findings (average error 1.8% and 1.5%, respectively).

An example for each of these 4 clusters is depicted in Fig. 51, where the unknown rescaled cumulative results (dashed purple line) obtained from the application of the reference depth/pseudo-depth rescaling functions, produced by the base known models of each cluster (solid orange line), are compared to their respective true cumulative resistivity models generated (dashed orange line).

As demonstrated by the graphs below, the trends of the rescaled unknown models of clusters 2 and 4 appear to be very similar to their own generated resistivity models, while the other two clusters example show larger discrepancies. Also for this dataset, these worst cases are exceptions, confirming the validity of the model given by the very low average error <6%, obtained with only 10 out of the 1000 rescaled models (1% of the models), consequently improving the result obtained with 200 curves (5% of avg. error, knowing the 5% of the models).



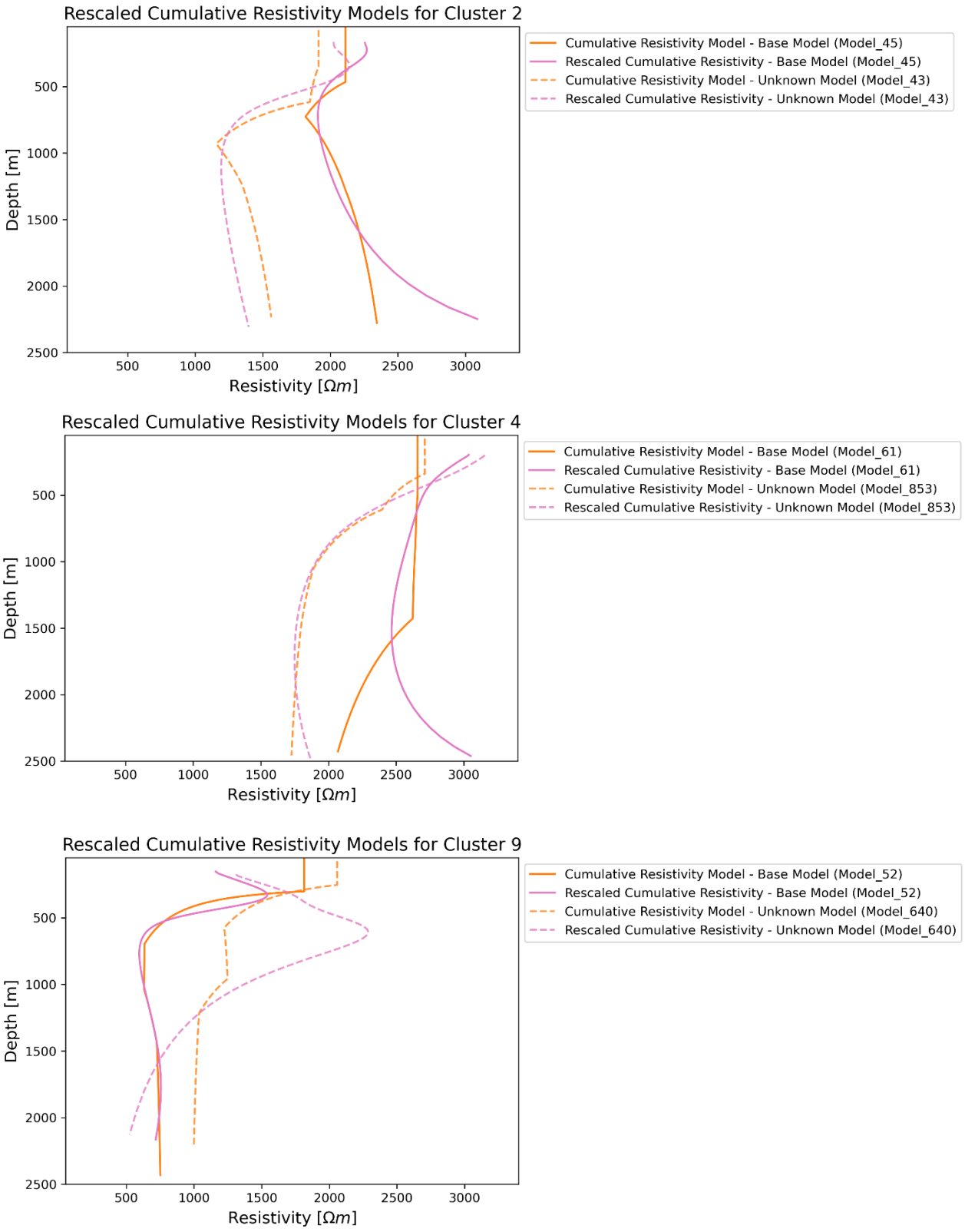
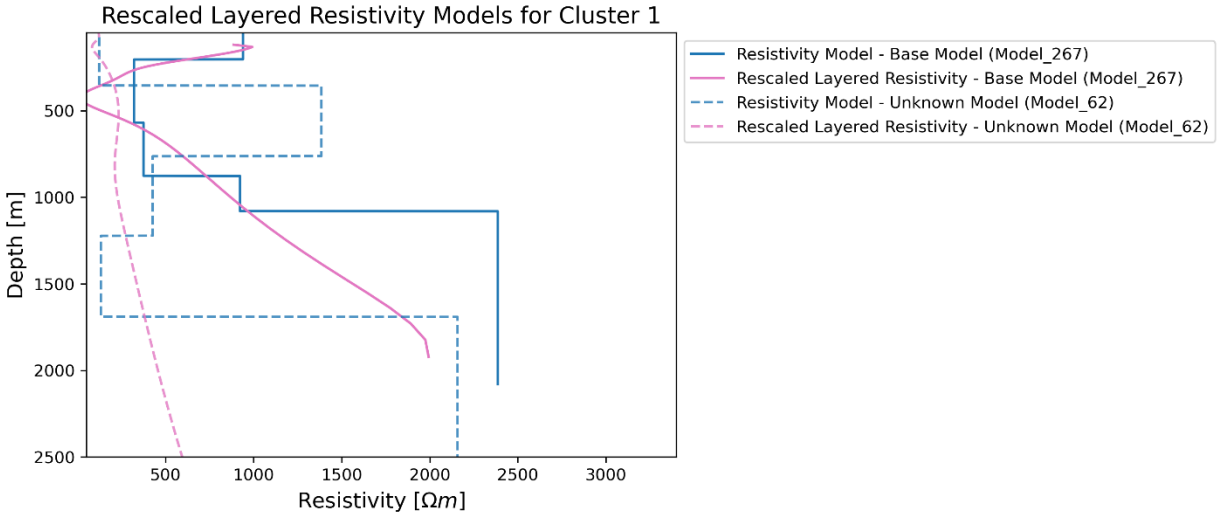
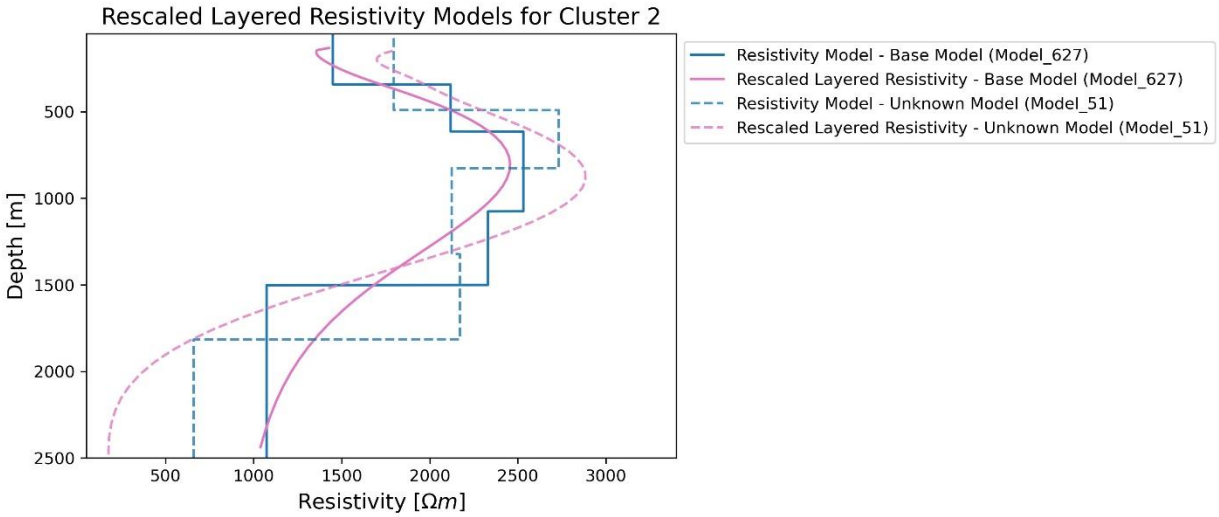
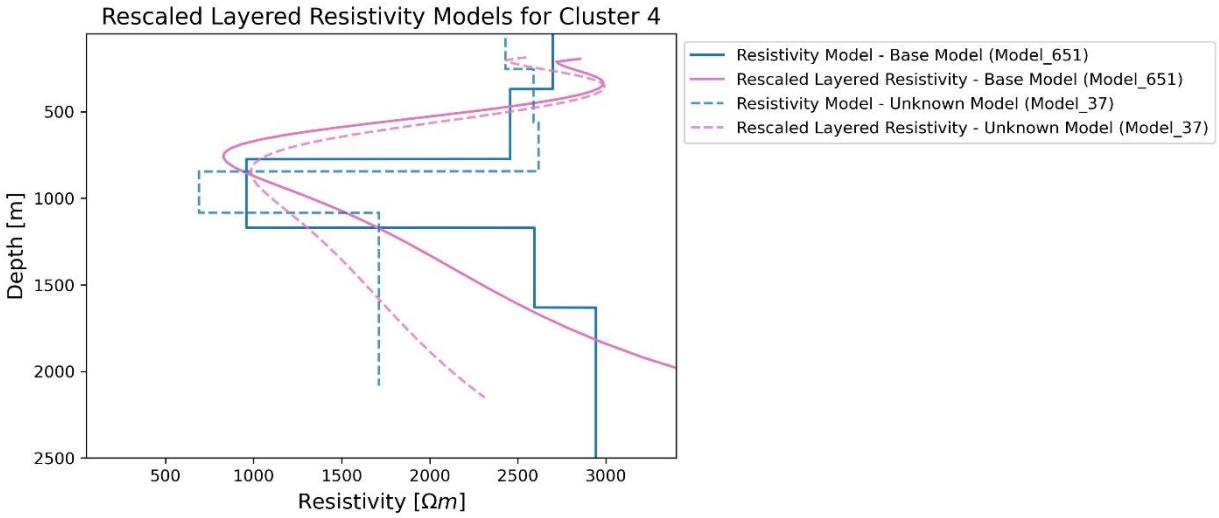


Figure 51 - 4 examples of the cumulative results obtained, 1 for each cluster (number 1,2,4 and 9). In each plot are represented, with a solid line, the cumulative base model (in orange) from which the depth/pseudo-depth rescaling function for the cluster in analysis is obtained and the rescaled one (in purple). On the other hand, with a dashed line, are shown the unknow cumulative model (in orange) and the rescaled one (in purple), the whole set of rescaled unknow models correspond to the final results of the methodology.

Same comparison, for same clusters, is done for the unknown rescaled layered models vs the generated layered resistivity ones (Fig. 52). In this case the final results (rescaled layered unknown models, dashed purple line) obtained from the application of the reference depth/pseudo-depth rescaling functions, produced by the base known models of each cluster (solid blue line), are compared to the resistivity models generated at the beginning of the test (dashed blue line).







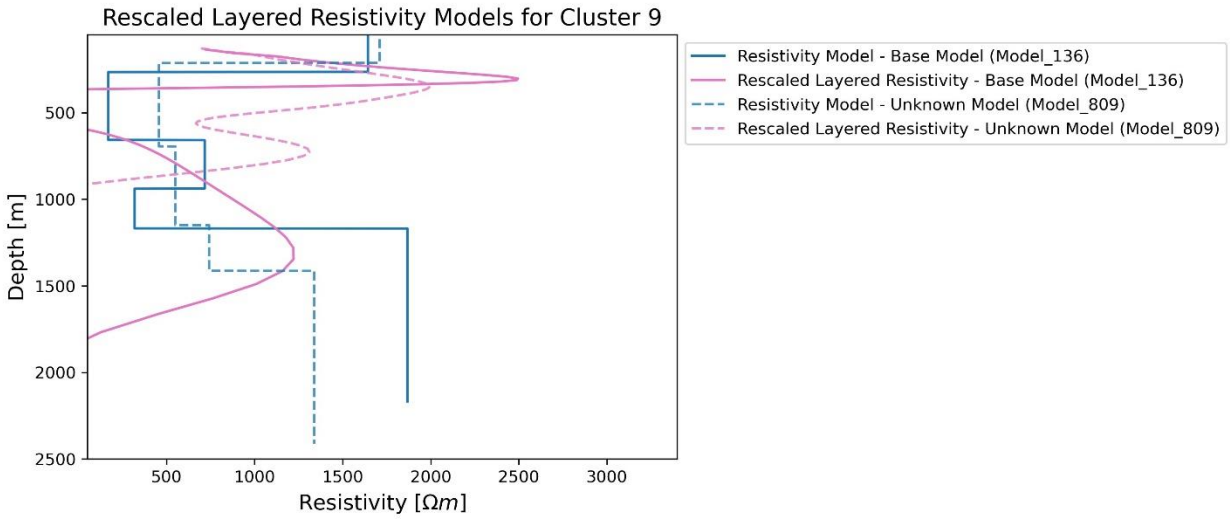


Figure 52 - 4 examples of the final results obtained, 1 for each cluster (number 1,2,4 and 9). In each plot are represented, with a solid line, the base model (in blue) from which the depth/pseudo-depth rescaling function for the cluster in analysis is obtained and the rescaled one (in purple). On the other hand, with a dashed line, are shown the unknow model (in blue) and the rescaled one (in purple), the whole set of rescaled unknow models correspond to the final results of the methodology.

After achieving good results with two synthetic datasets, the method was submitted to a final test on a real-case scenario.

5.4 Real Dataset

The real dataset concerned is "COPROD2," consisting of a set of 35 apparent resistivity data acquired from an MT Geological survey in Canada. More precisely, "*the data are from stations along a 400 km east-west profile in southern Saskatchewan and Manitoba, Canada, crossing the thick Palaeozoic sediments of the Williston Basin*". [14]
These data, before being clustered, were trimmed within the available frequency range [1.1^{-3}, 0.5] Hz, and a depth limit of 30 km was imposed. The COPROD2 dataset is represented in Fig. 53.

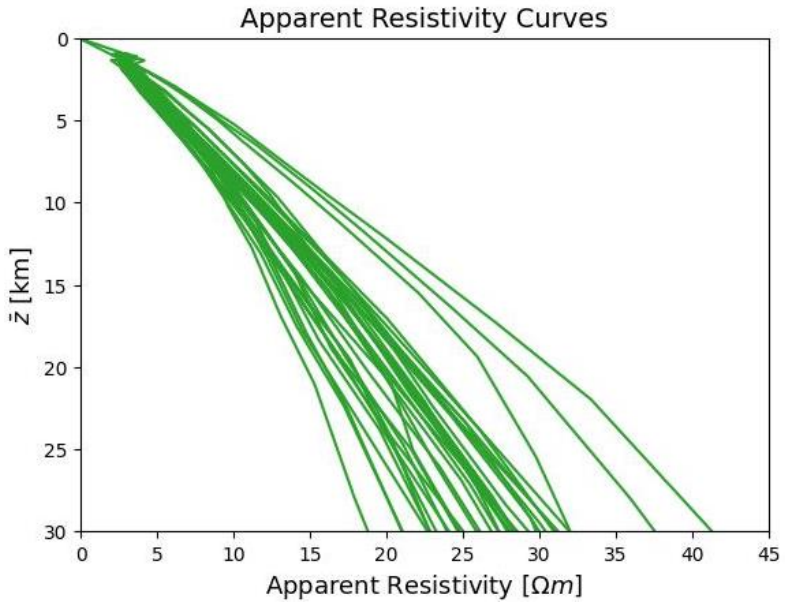


Figure 53 - COPROD2 dataset composed by 35 apparent resistivity curves, represented as function of pseudo-depth.

The second figure (Fig.54), was obtained using the "colormesh" function in Python, and the data were sorted by increasing East coordinate.

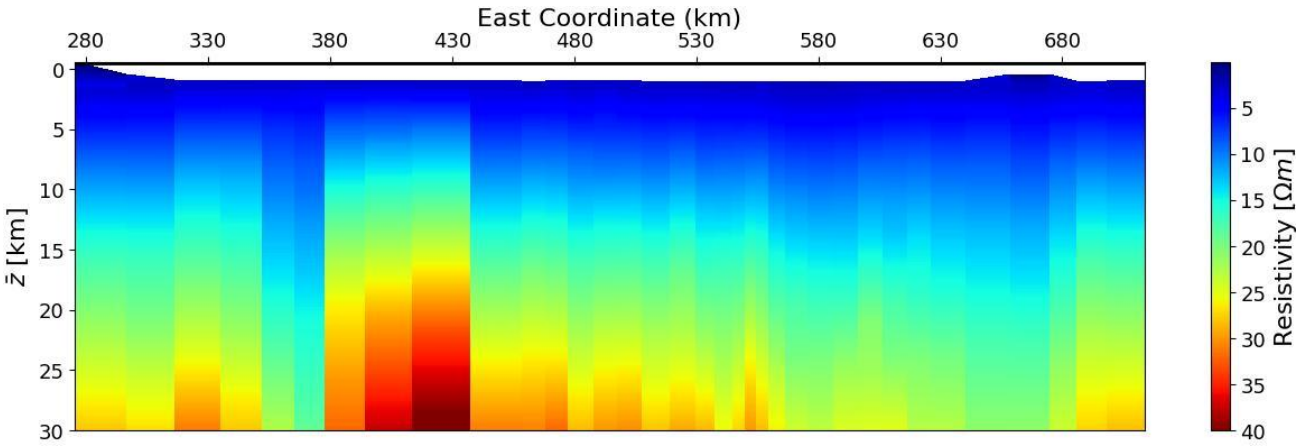


Figure 54 - "Colormesh" plot representing the data distribution of the COPROD2 dataset as function of pseudo-depth. Data are sorted on the East-West line.

The next step, is to compose the input matrix of the k-means algorithm using the combination of parameters established in section 5.1. Before running the algorithm, the optimal number of clusters was determined using Elbow and Silhouette methods, which converge to k=3. The results of clustering on the COPROD2 dataset are represented in Fig. 55. In particular, Cluster 2 (black) groups the three models that can be considered outliers (higher resistivity values) compared to the data belonging to the other two clusters (0 in blue, and 1 in red).

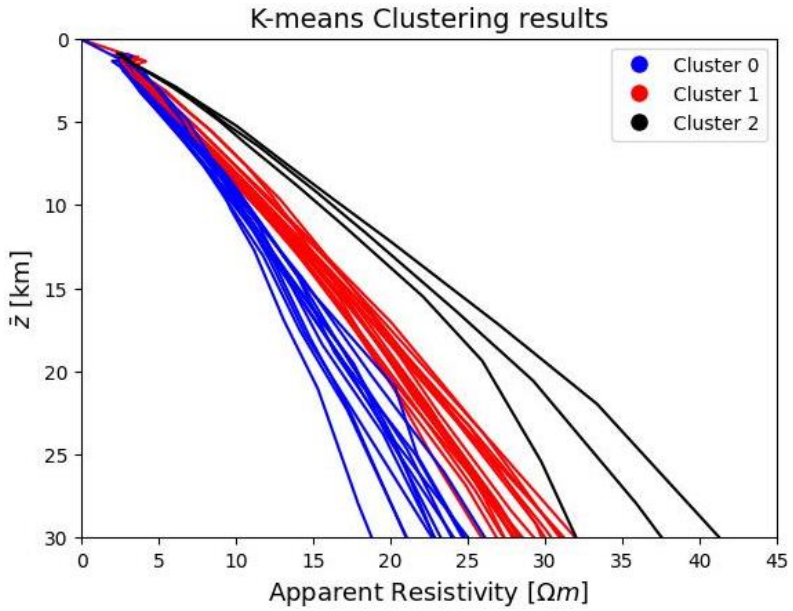


Figure 55 - K-means results for the COPROD2 dataset. Apparent resistivity curves are divided in 3 clusters, 0 (blue), 1 (red), 2 (black)

In Fig. 56, the "colormesh" plot of Fig. 54 is represented with the addition of coloured circles depicting the three clusters. As visible from the graph, Cluster 0 (blue circles) is characterized by

lower resistivity values, Cluster 1 (red circles) by higher amplitudes, and finally, Cluster 2 (black circle), which has much larger values than the other two. Furthermore, in the top of the figure, there are also represented 3 reversed coloured triangles corresponding to the colour of each cluster, which denote the location of the three data selected as 'reference' or base of each cluster, which are inverted to obtain one resistivity model for each cluster.

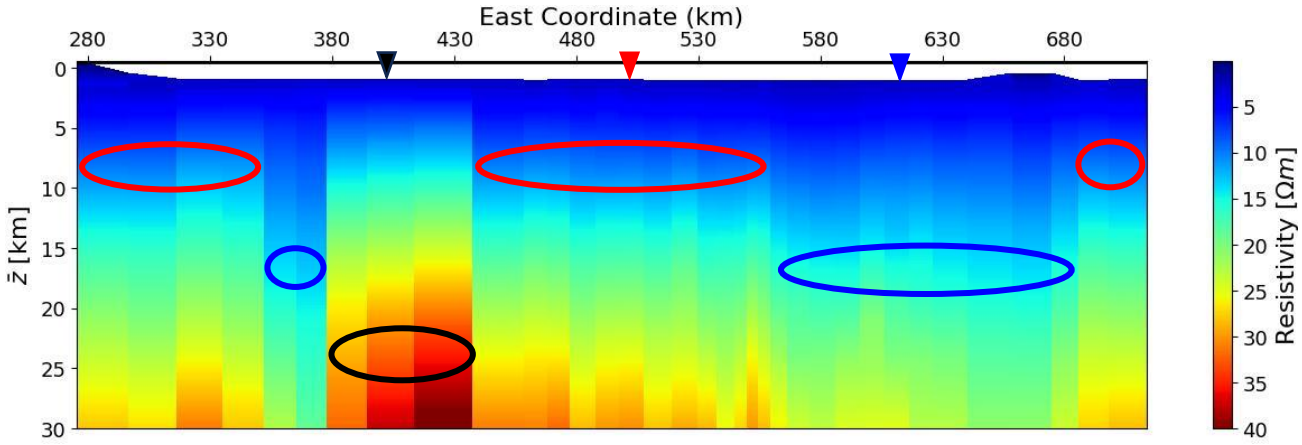


Figure 56 - Clustered version of the "colormesh" plot of Fig. 54. Apparent resistivity curves are divided in 3 clusters, 0 (blue), 1 (red), 2 (black). In addition, are shown (with coloured reversed triangles) the three selected curves as representative for each cluster. These data are then inverted to obtain three models, used for the computation of the three rescaling functions, one for each cluster.

The inversion process, performed on the three apparent resistivity data, generates three 1D layered resistivity models defined for fixed depths. These models were then used to determine (in the cumulative resistance domain) the three reference depth/pseudo-depth relationships, one for each cluster, used to rescale the entire dataset into resistivity models.

The set of rescaled models, and so the final results of the methodology on this dataset, is represented in the 'colormesh' plot of Fig. 57, where they are sorted in the East direction, interpolated between each other and horizontally smoothed.

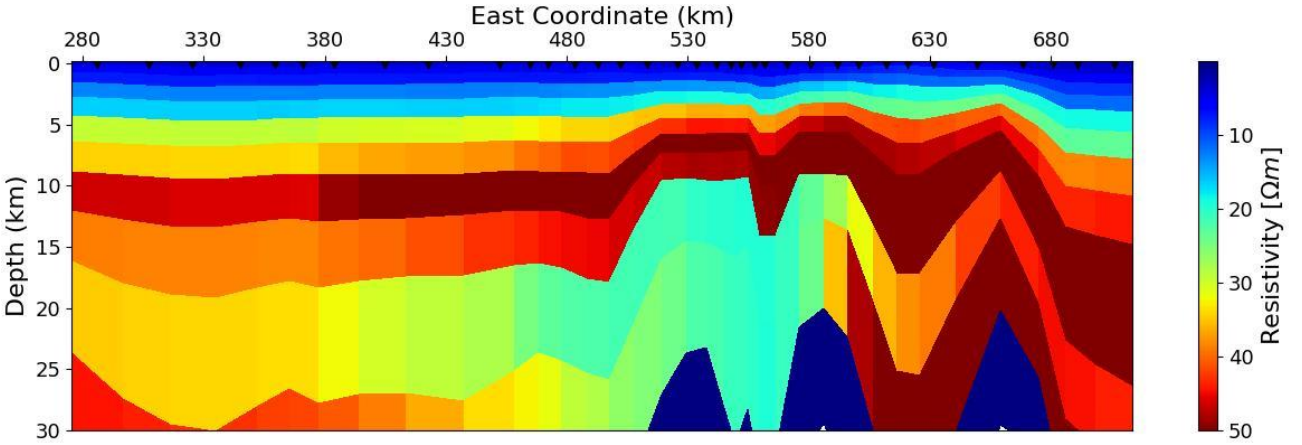


Figure 57 - "Colormesh" plot of the rescaled models, sorted onto the East-West line, obtained from the rescaling of the COPROD2 dataset.

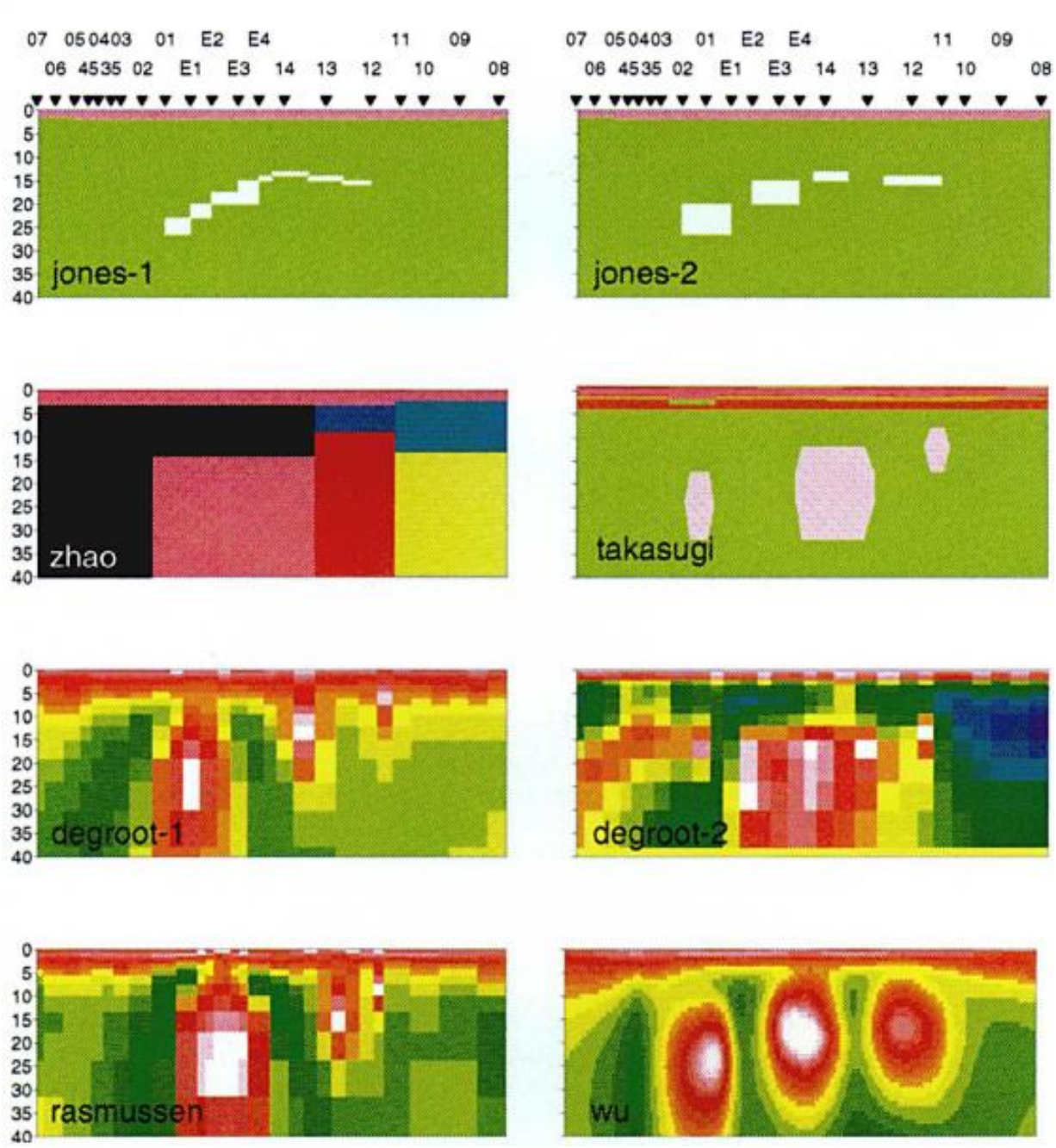


Figure 58 - Inverted resistivity models of COPROD2 MT data done by many geophysicist through 80's and 90's [14]

Since COPROD2 is a dataset of crustal depth, it has been the focus of various inversion tests conducted by several geophysicists between the 1980s and 1990s. From the resulting models of these processes, as visible in Fig. 58, all the limitations of inversion and the problems in terms of solution that can derive from it emerge, even if they result in solutions to the same problem. The only solutions comparable to our results are the last two tests produced by Rasmussen and Wu, in which conductive bodies (white areas) are retrieved, as in our case, among the remaining resistive layers. [14]

Since it's not possible to consider these results as benchmarks, an inversion process was conducted in this Thesis starting from the 35 apparent resistivity data (Fig. 53) and inverting them one by one. The results are shown in Fig. 59. As we can see from the comparison between these benchmark results and the rescaled ones, the overall behaviour is being retrieved by the rescaled models, and it is possible to locate the effect of some conductive body or bodies in the stations at 530-580 km East that can also be seen in the inverted profile. Additionally, in the easternmost section of Fig. 57, there is also a conductive effect between 2 layers (10-25 km depth), and even if it doesn't have the same values as the inverted profile (Fig. 59), the conductive effect is retrieved.

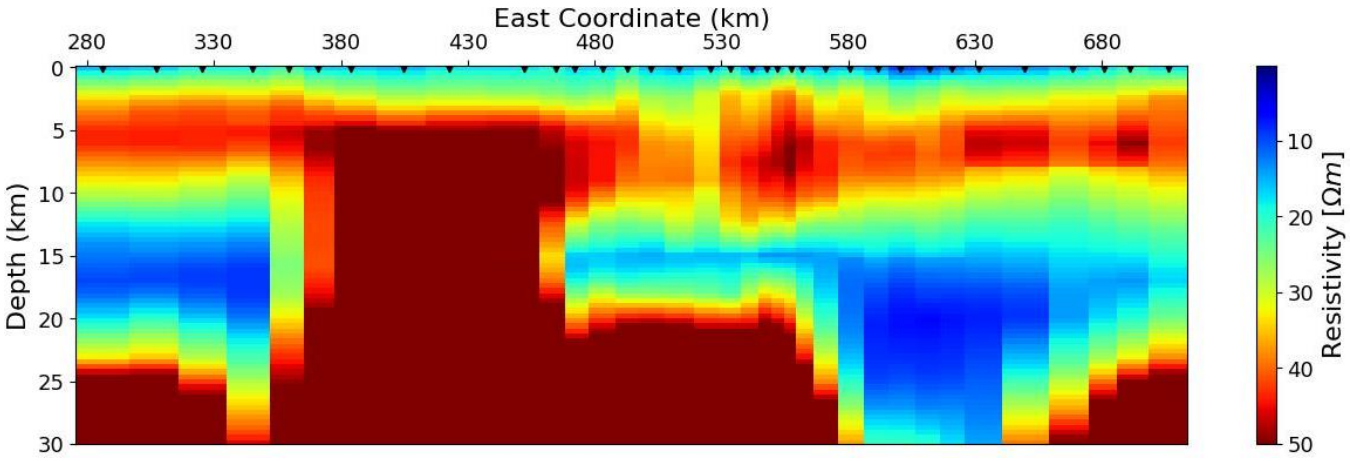


Figure 59 - Inverted resistivity models of COPROD2 MT data. These results are used as a benchmark to compare rescaled data of Fig. 52 obtained using our methodology.

As shown in the graphs, there is a bulk shift in values between the rescaled data and the inverted benchmark ones. This is because the benchmark results were obtained through inversion. As explained, these processes are affected by the non-uniqueness and the non-linearity problems, whereby completely different results may yield the same solution (as we can see in the 80's and 90's tests of Fig.58).

Apart from this bulk shift, the results have demonstrated the validity of the method, which with only 3 known models, is able to transform the entire dataset into models and obtain a similar trend to the results obtained from a traditional inversion process where all the data are inverted one by one (much higher computational cost).

This enhances the value of the findings, making the rescaled models (almost at this stage of development of this methodology) useful as reference models for subsequent 2D inversion processes as a way to avoid biases.

Chapter 6

6. Conclusions

The Thesis work outlined, has demonstrated a methodology to cluster apparent resistivity curves ,acquired through the MT method, in order to reduce the number of known models required to rescale an entire set of apparent data into 1D layered resistivity models. This method constitutes an alternative to the conventional inversion processes used to determine resistivity models from MT data.

The Rescaling process, following the research of *Calderon Hernandez O., 2023 [15]*, is performed by applying to apparent data the so called depth/pseudo-depth rescaling function, computed from the relationship between the depth for which cumulative models are defined, and the measured pseudo-depth of the apparent data.

The main problem, addressed by the present study, was the inability to rescale an entire dataset of heterogeneous curves with a single depth/pseudo-depth rescaling function relative to only one resistivity model, as huge errors were obtained, even for small datasets.

The Clustering process represents the solution to this problem. Indeed, by grouping the apparent resistivity data, according to criteria related to the mathematical parameters of the curves, it is possible to use one rescaling function for each cluster to rescale an entire dataset.

About 30 tests were conducted on synthetic datasets, to determine the best algorithm among those tested (k-means, CURE, and OPTICS), and the best combination of mathematical parameters among those explored (local maxima and minima, gradients, average values, etc.) which compose the multidimensional input matrix for the clustering algorithms. These results were chosen because they guaranteed the lowest error between: the cross-rescaled models using a depth/pseudo-depth rescaling function per cluster, and the true models generated randomly in the first place.

The findings were:

→ *Best algorithm*: K-means
→ *Best combinations of parameters* : Initial and Final Resistivity,
Average Resistivity and Average Depth

Then, this methodology was tested on two datasets composed of 200 and 1000 apparent resistivity data, obtained through a Python routine ("empymod") from synthetic resistivity models generated. After creating the datasets, in both cases, these curves are clustered in 10 clusters using the criteria

and algorithm stated previously, and finally cross-rescaled by in turn using a depth/pseudo-depth for each cluster.
The errors due to each cross-rescaling constitute the main results to assess the validity of that methodology, in fact, low average errors were found in both tests:

→ *200 data test*: average error <5%, knowing 5% of the models (10 out of 200)
→ *1000 data test*: average error <6%, knowing 1% of the models (10 out of 1000)

After achieving good results with two synthetic datasets, the method was submitted to a final test on a real-case scenario. In this test, the COPROD2 dataset consisting of 35 apparent resistivity data acquired with the MT method in a geological survey in Canada, was divided into 3 clusters. Subsequently, one 'reference' data per cluster was inverted, thus obtaining a layered resistivity model for each cluster, and computing one rescaling function per cluster used to rescale the entire dataset. The rescaled data were then compared with the benchmark models obtained from an inversion process, and they confirmed the presence of conductive bodies in the investigated area with a bulk shift between the two sets of results due to inversion issues.

In the end, rescaled models exhibited the same overall trend found in the inversion results (with a significantly reduced computational cost), so almost at this stage of the clustering+rescaling methodology, they can be used in real scenarios as reference starting models for 2D inversion processes, as a way to avoid biases.

References

[1] Simpson, J., *"Magnetotelluric data for exploration – Geological surveying of Queensland"* available at: https://smi.uq.edu.au/files/43560/Dec18KTW_MT_workshop.pdf

[2] Pellerin, L., *"Applications of electrical and electromagnetic methods for environmental and geotechnical investigations",* 2002.

[3] Florio, G., *"Mapping the depth to basement by iterative rescaling of gravity or magnetic data",* 2018.

[4] Socco, L. V., Comina, C., and Khosro Anjom, F., *"Time-average velocity estimation through surface-wave analysis: Part 1—s-wave velocity",* 2017.

[5] Basokur, A.T., *"Definitions of apparent resistivity for the presentation of magnetotelluric sounding data"*, 1994

[6] *"Magnetotellurics"* available at: https://em.geosci.xyz/content/geophysical_surveys/mt/index.html#mt-index

[7] Nabighian, M.N., *"Chapter 8 - The Magnetotelluric Method, Electromagnetic Methods In Applied Geophysics – Part A&B",* 1991

[8] *"General Solution of Maxwell equations for a Planewave",* 2018, available at: https://em.geosci.xyz/content/maxwell1_fundamentals/harmonic_planewaves_homogeneous/derivation.html#harmonic-planewaves-homogeneous-derivation

[9] Griffiths, D., J., "*Introduction to electrodynamics*", 1999.

[10] Di Giuseppe, M., G., *"SEPARAZIONE DI CONTRIBUTI DI ONDA PIANA E DI CAMPO VICINO PER L'INVERSIONE DI DATI MAGNETOTELLURICI",* available at: https://amsdottorato.unibo.it/154/1/TesiDottorato.pdf

[11] *"Apparent Resistivity"* available at: https://em.geosci.xyz/content/maxwell1_fundamentals/harmonic_planewaves_homogeneous/apparentresistivity.html

[12] NIST, "Fundamental physics constant – vacuum magnetic permeability", 2006, available at: https://physics.nist.gov/cgi-bin/cuu/Value?mu0

[13] Jones, A., G., *"On the Equivalence of the "Niblett" and "Bostick" Transformations in the Magnetotelluric Method",* 1983

[14] Jones, A., G., "*The COPROD2 Dataset: Tectonic Setting, Recorded MT Data, and Comparison of Models",* 1993

[15] Calderon Hernandez, O., *"Direct 1D Resistivity Estimation from Data Rescaling Using Cumulative Resistance Models",* 2023

[16] Werthmüller, D., *"empymod Phyton routine – Magnetotelluric", 2017,* available at: https://empymod.emsig.xyz/en/stable/gallery/fdomain/magnetotelluric.html

[17] Arvai, K., *"K-Means Clustering in Python: A Practical Guide"*, 2020, available at: https://realpython.com/k-means-clustering-python/#what-is-clustering

[18] Beigy, H., "Machine learning theory - Theory of clustering", 2022, available at: https://sharif.edu/~beigy/courses/14002/40718/Lect-28.pdf

[19] Jeffares, A., *"K-means: A Complete Introduction",* 2019, available at: https://towardsdatascience.com/k-means-a-complete-introduction-1702af9cd8c

[20] Di Giuseppe , M., Troiano, A., Troise, C., De Natale, G., "*k-Means clustering as tool for multivariate geophysical data analysis",* 2014

[21] Tan, P.N., Steinbach, M., Kumar, V., *" Introduction to Data Mining",* 2014, available at: https://www.ceom.ou.edu/media/docs/upload/Pang-Ning_Tan_Michael_Steinbach_Vipin_Kumar_-_Introduction_to_Data_Mining-Pe_NRDK4fi.pdf

[22] *"Visualizing K-means clustering*", available at: https://www.learnbymarketing.com/methods/k-means-clustering/

[23] Banerji, A., "*K-Means: Getting the Optimal Number of Clusters",* 2023, available at: https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/#Methods_to_Find_the_Best_Value_of_K

[24] Joshi, S., *"What is Clustering in Machine Learning: Types and Methods"*, 2022, available at: https://www.analytixlabs.co.in/blog/types-of-clustering-algorithms/

[25] Sharma, H., "Hierarchical Clustering", 2021, available at: https://harshsharma1091996.medium.com/hierarchical-clustering-996745fe656b

[26] Xu, A., "*Clustering Algorithms BIRCH and CURE",* 2019

[27] Leskovec, J., Rajaraman, A., Ullman, J., *"The CURE Algorithm (Advanced)*", 2016, available at: https://www.youtube.com/watch?v=JrOJspZ1CUw

[28] *"Basic understanding of CURE algorithm", 2021,* available at: https://www.geeksforgeeks.org/basic-understanding-of-cure-algorithm/

[29] Kassambara, A., *"Advanced Clustering - DBSCAN: Density-Based Clustering Essentials"*: available at: https://www.datanovia.com/en/lessons/dbscan-density-based-clustering-essentials/

[30] Plakalovic, A., "*Clustering Algorithms: DBSCAN vs. OPTICS*", 2023, available at: https://www.atlantbh.com/clustering-algorithms-dbscan-vs-optics/

[31] Pegoraro, E., *"Statistica per Data Science - Capitolo 9 Introduzione all'algoritmo DBSCAN"*, 2019, available at: http://www.r-project.it/_book/introduzione-allalgoritmo-dbscan.html

[32] *"DBSCAN"* available at: https://it.wikipedia.org/wiki/Dbscan/

[33] Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J., *"OPTICS: Ordering Points To Identify the Clustering Structure"*, 1999, available at: https://www.dbs.ifi.lmu.de/Publikationen/Papers/OPTICS.pdf

[34] *"Illustration of the "reachability distance" concept in the DBSCAN, OPTICS and LOF algorithms"*, 2010, available at: https://en.m.wikipedia.org/wiki/File:Reachability-distance.svg

[35] Eric, J., *"OPTICS Clustering: From Novice to Expert in Simple Steps"*, available at: https://datarundown.com/optics-clustering/

[36] TIBCO, *"TIBCO Statistica User's Guide Conceptual Overviews - 2D Box (and Means with Error) Plots",* 2020, available at: https://docs.tibco.com/pub/stat/14.0.0/doc/html/UsersGuide/GUID-CD68E5DD-DCC2-49D0-9C5A-84D8526F9DB7.html