Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Simon Markus Hoff Olsen

Automated Interpretation of Depth to Bedrock from Airborne Electromagnetic Data Using Machine Learning Techniques

Master's thesis in Informatics
Supervisor: Jingyue Li
May 2019

NTNU
Norwegian University of
Science and Technology

Simon Markus Hoff Olsen

Automated Interpretation of Depth to Bedrock from Airborne Electromagnetic Data Using Machine Learning Techniques

Master's thesis in Informatics
Supervisor: Jingyue Li
May 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

Traditional methods of bedrock modeling from airborne electromagnetic data has previously relied on time-consuming manual labor. Novel techniques has introduced the aspect of automation, but it has also presented a new challenge in uncertainty estimation of the automated results. This thesis aims to further enhance the state of the art, by not only proposing a new technique for automation, but also by evaluating the suitability of three different construction techniques for prediction intervals as an uncertainty measure. A case study shows how aspects from the academic field of computer vision can be used in conjunction with more conventional machine learning techniques to improve the current standard. The results from the case study shows how the approach brings reductions in Mean Absolute Error and Root Mean Squared Error of bedrock predictions up to $\sim 50\%$ and $\sim 52\%$ respectively, compared to a conventional neural network method. A separate case study shows how different construction methods for prediction intervals fit different situations depending on factors in the dataset such as the uncertainty distribution. The thesis presents new knowledge in the still insufficiently explored intersection between machine learning and electromagnetic geotechnical data, and new knowledge in the feasibility of prediction intervals as an uncertainty measure for bedrock depth interpretation.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) in Trondheim, and concludes my Master's degree in Informatics. The work in this thesis was conducted from August 2018 to May 2019 at the Department of Computer Science (IDI).

First of all, I want to thank Associate professor Jingyue Li from the Department of Computer Science at NTNU for the guidance he provided as my supervisor. I also want to thank Dr. Andreas Aspmo Pfaffhuber, Dr. Malte Vöge, Dr. Zhongqiang Liu and Craig William Christensen from Norwegian Geotechnical Institute for the guidance they provided as co-supervisors.

I want to thank my parents, who has been an absolute source of unwavering support throughout my years of studies.

Finally, I want to thank my girlfriend, Kristin, for all her love and support.

Table of Contents

List of Tables

List of Figures

Abbreviations

0.1 General

NTNU	=	Norwegian University of Science and Technology

0.2 Geological & Geotechnical

NGI	=	Norwegian Geotechnical Institute
GIS	=	Geographic Information System
EM	=	Electromagnetism
AEM	=	Airborne Electromagnetics
DTB	=	Depth to Bedrock
ERT	=	Electrical Resistivity Tomography
LSI	=	Localized Smart Interpretation
MSI	=	Manual Spline Interpolation
LCI	=	Laterally Constrained Inversion
SCI	=	Spatially Constrained Inversion
TS	=	Total Sounding
RPS	=	Rotary Pressure Sounding

0.3 Technical & Statistical

ML	=	Machine Learning
MLP	=	Multilayer Perceptron
ANN	=	Artificial Neural Network
ReLU	=	Rectified Linear Unit
CNN	=	Convolutional Neural Network
PI	=	Prediction Interval
PICP	=	Prediction Interval Coverage Probability
MPIW	=	Mean Prediction Interval Width
CWC	=	Coverage Width Criterion
CI	=	Confidence Interval
IDW	=	Inverse Distance Weight
SV	=	Semi Variance
MVE	=	Mean Variance Estimation
AE	=	Absolute Error
MAE	=	Mean Absolute Error
MSE	=	Mean Squared Error
RMSE	=	Root Mean Squared Error

Chapter 1

Introduction

This chapter will introduce and present the main topics of interest for the thesis. Section 1.1 will provide the underlying motivation for working with bedrock modeling, Airborne Electromagnetics (AEM), and automation of the related data interpretation. Section 1.2 provides an overview of the general problem statement. It offers a short summary of the background, research questions, research methodology, research results, and the research's main contributions.

The chapter concludes with Section 1.3, which gives an overview of the thesis' structure.

1.1 Motivation

The Norwegian Geotechnical Institute has in recent years been experimenting with novel techniques for various kinds of interpretation of data obtained from use of airborne electromagnetic surveys. One such type of interpretation is mapping of Depth To Bedrock (DTB), over large areas. Bedrock topography is deemed important for establishing knowledge of subsurface stability and mass balance, which are important factors to be considered in planning of large scale infrastructural construction projects [10].

The use of airborne electromagnetism for geotechnical purposes is not novel, and was first used in the early 1950's [29]. New use cases for AEM data has, however, emerged over the years [4][35][10], and precise interpretation of AEM data used for DTB tracking has remained a challenging task.

In the summer of 2018, NGI posted a proposal for a master thesis on the website of the department of Computer Science at NTNU, (the Norwegian University of Science and Technology), in Norway. The complete proposal can be found in Appendix A. The general problem that was presented proposed that potential improvements and benefits could be obtained with innovative use of data from AEM surveys for the problem of automated DTB interpretation.

Manual techniques such as collecting depth related data from boreholes can, in comparison with AEM surveys, provide highly accurate data on DTB levels for single and distinct locations. However, such boreholes are costly and as the survey area expands, boring a large number of boreholes becomes necessary to obtain enough information to construct and model a spatially continuous and complete picture of the area's bedrock topography. Hence, use of boreholes for bedrock modelling is certainly useful, but relying solely on boreholes for mapping bedrock topography over large areas is infeasible with respect to cost. Boreholes do arguably also impose a larger disturbance on the environment than AEM surveys due to both physical impact and noise. The need for a surveying technique that can better scale with the size of the area may thus seem clear.

Airborne electromagnetic surveys can collect data for larger areas in a far less costly manner than numerous boreholes. However, its measurements are far less precise than those of ground based nature and confident automation of data interpretation has yet to be firmly established in academia. The initial output from a single AEM sounding in itself does not hold much computationally informative value. The complete process of interpreting DTB values from data obtained from AEM can largely be thought of as a two phase process, consisting of an initial *inversion* and a subsequent *interpretation*. The inversion phase consists of converting the raw electromagnetic data, as obtained from the AEM soundings, to discrete resistivity profiles which are more conveniently used for further computational analysis. The geophysical process of conducting such inversion is complex, and also out-of-scope for this thesis. However, the interpretation as described in the later sections would not make much sense without the processed data to interpret. A brief introduction and description of the inversion process will therefore be given in Section 2.1 on Geotechnical Definitions.

The interpretation phase utilizes the resistivity profiles to approximate a best estimate for the DTB at the location of a resistivity profile. This thesis will focus on automation and technical aspects of the interpretation phase. In other words, the focus lies on the final interpretation of the output data from the inversion.

The benefits of obtaining resistivity profiles in a continuous space are significant for many use cases. Current and modern interpretation techniques allow geotechnical experts to gain insight about DTB values across a spatial continuum by analyzing spatially continuous sets of resistivity profiles. While resistivity profiles must be regarded as approximations, they can still yield highly informative and continuous models of bedrock topography over large areas without the need for numerous and costly boreholes. The benefits of such knowledge at early stages in expensive construction projects are significant, allowing for extensive potential savings in the requirements for further detailed subsurface mapping.

The aforementioned benefits are, however, diminished by time-consuming and tedious manual labor during the interpretation process [9]. The manual process relies on a single geotechnical expert to interpret visualizations of obtained resistivity profiles along a flight line. Upon completion of the interpretation, the goal is for the analyst to estimate a set of DTB values along the flight line, which can ultimately be used as reliable point predictions in the construction of a bedrock model. This manual interpretation does pose various issues, which lays grounds for the first motivation of this thesis. The manual labor of assigning DTB values is tedious, and for flight ranges spanning several hundreds of kilometers, the time consumed is substantial. Moreover, a direct consequence of the resulting DTB values being plain output of the geotechnical expert's interpretation is that the resulting DTB values may also be drastically biased by the human analyst. Reproduction of the results from the interpretation also becomes a difficult problem, potentially relying on a single human resource. Thus, the underlying motivation for an automated and unbiased approach for the interpretation phase is a combination of the challenges presented above.

The second motivator for the thesis is the industry's desire for knowledge about the uncertainty that comes with predictions of DTB levels from automated systems. In the summer of 2015, NGI delivered a set of bedrock models to the Norwegian Railroad Association for early phases of a railroad construction project [10]. Such construction projects require precision and certainty in knowledge of the subsurface topography, and corresponding uncertainties relating to each DTB distance within the bedrock models were delivered accordingly. NGI utilized a semi-automated approach for prediction of the required DTB levels, but the uncertainties corresponding to each prediction had to be manually registered and user-assigned [5]. This manual approach is not only time consuming, but it also presents human bias and difficulties in reproduction of the results. Thus, the market has shown a need for knowledge of the uncertainty that is related to the depth predictions in DTB models. Based on the needs from the industry, a claim can be made that any potential benefits of automation of DTB prediction will be reduced if uncertainties related to the output must be manually registered.

1.2 Problem Specification

Airborne Electromagnetism (AEM) has been used frequently in the past years to investigate ground properties for planning and optimization of large road or railroad projects in Norway. For each measurement along the AEM flight line, a resistivity profile is acquired from aforementioned inversion techniques, which provides information about the geological structure of the subsurface. In this thesis, the goal is to identify the boundary between overburden and bedrock. Bedrock usually has a rather high resistivity, while most (but not all) layers in the overburden have lower resistivities. The relation between resistivities and geological material is complex and non-unique. Therefore, converting these profiles into useful geotechnical information requires thorough interpretation. The general problem can be divided into two distinct problem areas. The research questions are explicitly stated in Chapter 4, and a brief introduction follows here. The first problem area can be described by the challenge of identifying an interpretation rule that can accurately interpret the bedrock depth from resistivity profiles. The automation of this task presented in this thesis is based on training data provided by resistivity profiles acquired from AEM measurements and by borehole data. A sufficiently precise automated approach bears the potential to significantly reduce the overall time that is required for the interpretation phase. Researchers at NGI has earlier proposed a novel solution utilizing an artificial neural network for automated interpretation [27]. However, the solution cannot account for contextual information, as it solely analyses single resistivity profiles at distinct locations. This thesis aims to propose a novel technique for interpretation inspired by state of the art Machine Learning (ML) technology from the academic field of computer vision, and the relating research question revolves around how such ML technology can best be used in combination with data from AEM surveys. A case study is carried out with data from historic surveys. The results from the case study shows significant potential for improved DTB predictions with use of ML techniques from the field of computer vision, albeit along a drastic increase in the required computational resources.

The second problem area is the identification and representation of the uncertainty that comes with each prediction. As the measurements vary with geological complexity and the consisting materials in the overburden, the interpretation will have a varying degree of confidence. Thus, an uncertainty measurement is desired for the boundary, and is currently not accounted for in established academia, nor is it part of the current state of the art industry approach. The aspect of uncertainty is therefore also considered. This thesis evaluates the industry's current methodologies of uncertainty prediction, and proposes new techniques where the final uncertainty values correlates directly to the ML model's confidence, as contrasted by the uncertainty of the geotechnical expert's biased view on the matter. The second research question therefore asks how prediction intervals can best be computed for representing the confidence of DTB predictions from automated techniques. A set of methods for construction of prediction intervals are compared and evaluated in a case study, also using real-world data from historic surveys. The results from the evaluation shows how each of the compared methods excel under different circumstances.

1.2.1 Scope

The entire process from the initial survey of an area to representation of a spatially continuous bedrock model is long, and it consists of multiple geophysical and mathematical processes. The scope of this thesis only encompasses a fraction of the entire procedure. Anything up until the resistivity profiles are acquired are considered out of scope. Furthermore, anything after the production of DTB values and respective uncertainties will not be accounted for.

1.2.1.1 Primary Contribution

The primary contribution of this thesis is the addition of new knowledge in the field of automated AEM data interpretation. More specifically, the thesis provides new knowledge in the area of contextually and spatially aware automated AEM data interpretation. The intersection of AEM data and machine learning is novel and insufficiently explored and established by current academia. The thesis will not propose a complete and automated solution of bedrock modeling from raw AEM data, but rather focus on the automation of contextual interpretation of DTB for a survey area. It is of importance to note that the goal of the proposals presented in this thesis should not be seen as an attempt to remove the need for human interpretation of the results, but it rather aims to empower the geotechnical analyst by allowing more time spent on overall interpretation, as opposed to manual analysis and interpretation of each and every measurement. The results from the research show significant indications that the proposed approach bears the potential to improve current state-of-the-art techniques, with reductions in Mean Absolute Error and Root Mean Squared Error up to more than 49% and 52% respectively.

In addition to providing point predictions for bedrock depths, the characteristics of a desired system include the ability to deliver uncertainties for each measurement. This thesis provides a comprehensive study and evaluation of three separate methods for uncertainty inference from regression type models. The research results presents findings that shows that no single method proved inherently best, but rather that the distinct methods each excel under different circumstances dependent on factors such as the models performance and the uncertainty distribution. The research illustrates how data analysis thus becomes an important early factor for selecting the optimal technique for uncertainty representation.

1.3 Thesis Outline

Chapter 1 introduced you to the underlying motivation for the thesis. The problem specification, scope, and primary contribution of the thesis was established.

Chapter 2 will provide a deeper explanation on the background for the thesis, including concepts and challenges that are considered relevant. The usefulness and context for the current state of the technology will be stated, and a more thorough introduction to the applicable technologies will be given. The relevant data will be discussed and accounted for, and the chapter will conclude with a technical introduction to uncertainty estimation.

The state of the art and related work is covered in Chapter 3. This encompasses common current techniques for DTB interpretation, an introduction to the relevant technical concepts from the academic field of machine learning, and established academic methods for uncertainty inference.

Chapter 4 covers the research design and research implementation that was used for conducting the research. The research motivation is stated alongside a set of research questions, which accurately denotes what information this thesis aims to investigate. The chapter describes how the research was conducted, and includes details on the technical implementation. This includes not only the approach for production of predictions and uncertainty estimations, but also the methods that were used to evaluate and compare the results.

The immediate output and results from the research are somewhat complex. Chapter 5 presents and elaborates on the final data, and offers a numerical analysis of the results and answers to the research questions.

Chapter 6 will present a discussion and an analysis of the complete research process. The discussion explores the outcome of the research, compares the results to related work and the current state of the art, and reviews the limitations of the research.

The thesis concludes with Chapter 7. This chapter aims to explore possibilities for future work that has the potential to establish knowledge which could further improve the current standard.

Chapter 2

Background

This chapter introduces established theories, technical concepts, and challenges that are relevant for the thesis.

The chapter begins with Section 2.1 which establishes and defines relevant geotechnical concepts that the reader will encounter in the thesis. Section 2.2 aims to describe the potential benefits of increased automation in DTB interpretation. Section 2.3 will delve deeper into the challenges that automated solutions presents. Section 2.4 presents a set of related software and tools, and offers a brief explanation of each.

The related data for the case study is explored in Section 2.5, while Section 2.6 provides an introduction to the problem of relating borehole data to resistivity profiles. This step is required for obtaining training data for the ML approach.

The chapter concludes with Section 2.7 which addresses and considers the need for uncertainty estimations for DTB predictions. It also addresses the challenges and potential solutions in the pursuit of ranged DTB representations.

2.1 Geotechnical Definitions

Bedrock can in layman's terms be defined as the rock foundation that lies beneath various layers of softer geological material. The bedrock surface is defined as the uppermost layer of the bedrock. DTB is the distance from the ground surface to the bedrock surface. These depths range from zero to hundreds of meters. Such cases, where bedrock is directly exposed on the surface, is typically referred to as *outcrop*, or *outcropping bedrock*. As earlier described, obtaining information about these depth values allows for improved planning and execution of various large scale projects, often related to infrastructural construction or maintenance.

A map of DTB levels over a spatially continuous area is commonly referred to as a *Bedrock Model*. Bedrock models are highly useful in multitudes of settings. Infrastructural expansion is one example of an industry which commonly relies on solid understanding of the subsurface topology for proper planning and execution of projects. These types of projects typically include construction of railways and highways, where large bedrock depths can cause significant instability and difficulties during both construction and maintenance. Obtaining descriptive models at early phases of projects may thus allow for improved risk management and drastic reductions in the overall cost of further investigation phases. Greater knowledge of the subsurface topography can reduce the required amount of necessary boreholes, and allow for smarter positioning.

In Norway it is not uncommon for such infrastructural projects to encounter complex subsurface geological settings, due to demanding topography and geology [10].

Boreholes, as referenced in this thesis, are vertical shafts bored in the ground for geotechnical investigations. Such boreholes can provide exact measurements for DTB values at specific locations. However, the cost of the process of drilling holes and obtaining such measurements are high. Boreholes play a significant role in the training phase of the technical machine learning model as presented by NGI [27]. The approach uses supervised training, where data from boreholes are used as labeled training data in the training process. The process of extracting labeled training data from boreholes and AEM data are, however, not straightforward. NGI's proposal utilized a method known as Kriging to address this issue [9]. Briefly described, the Kriging process allows for obtaining an approximation of a single and initially unknown resistivity profile at some location in a two-dimensional plane by weighted interpolation from surrounding and known resistivity profiles. The Kriging process carries many similarities to Inverse Distance Weighted (IDW) interpolation, but differs in the calculation of the weights. A more concrete description of this process is later described in Section 2.6.1.

AEM cannot currently supersede the accuracy of direct borehole sampling. However, Christensen *et al.* concluded that it can drastically reduce the amount of required boreholes for detailed surveying, by being able to yield models covering much larger areas than ground based methods at much lower cost [9]. NGI has, in recent years, utilized AEM for multitudes of projects [10][2][5].

AEM surveys are carried out by an airborne vessel, typically flying approximately 35 meters above the ground they survey. During an AEM survey, the airborne vessel carries a transmitter loop which transmits a magnetic dipole field in pulses into the ground. Since electrical conductivity is dependent on the matters composition and water content, the response from the pulses can be processed to detect different types of subsurface matter.

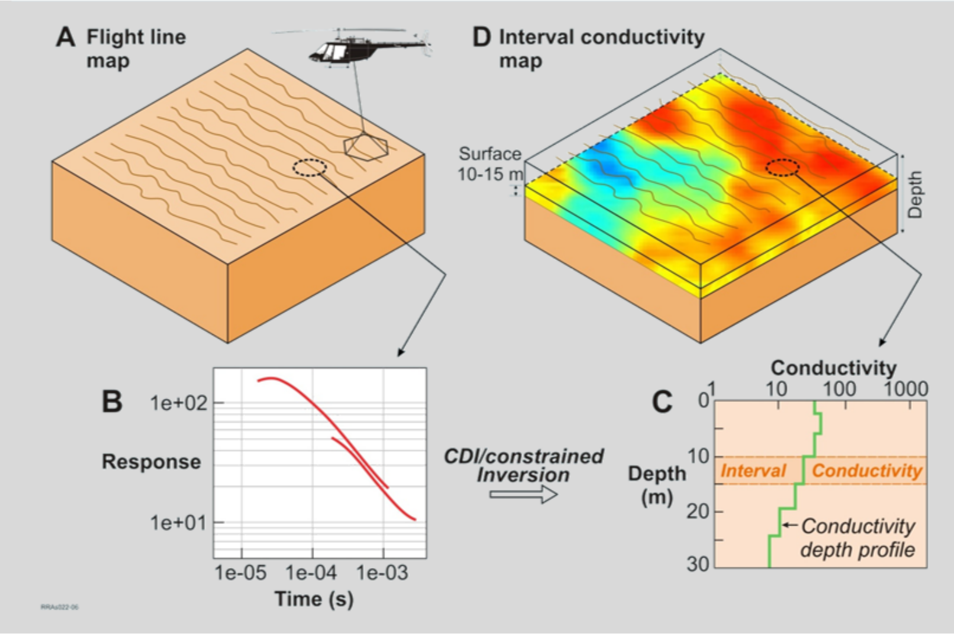


Figure 2.1: Visualization of the AEM workflow. The figure is adapted from Ley-Cooper *et al.* (2015) [26]

Figure 2.1 presents a visualization of the AEM workflow. Part A shows how an airborne vessel surveys the area, and part B shows a visualization of the raw electromagnetic data that are collected. Part C shows a vertical visualization of the resistivity data which are obtained from inversion of the raw electromagnetic data from part B. Part D shows how the resistivity data from C can be combined to form informative 3D resistivity by depth maps.

Inversion of raw AEM data provide layered 3D data of resistivities in the subsurface by transforming the collected raw data into the more useful form of resistivity profiles. The output of an inversion of a sounding from a survey typically contains data of the sounding's geographic location, topography, and a number of layered resistivity values by depth. Inversion is a complex topic within the academic field of geophysics, which is out of scope for this thesis and thus, only a brief introduction will be provided here. Recall that our desired output data from inversion of a single sounding is a layered resistivity profile. Such a profile consists of a mapping of depths to respective resistivities. The data obtained from a single AEM sounding can, in its most simplistic form, be regarded as a polynomial curve. If d represents a sounding and m represents a resistivity profile, an operator F can be defined such that $d = F(m)$. The operator F represents the calculation of the magnetic field d given a known resistivity model m, for which (in the 1D case) quasi-analytical solution exists. Thus, $d = F(m)$ can be calculated very efficiently. However, as in our case the soundings are known and the resistivity models are unknown, the inverse of F has to be found: $m = F^{-1}(d)$. The approach for finding an approximation of F^{-1} relies on making educated guesses on values in m and comparing the resulting sounding to the actual known sounding. A gradient descent type of inversion (e.g. Gauss-Newton) method is used to improve the guesses until a sufficiently matching sounding can be found. Local minimums in the gradient descent can cause challenges, and various techniques such as varying stepping sizes, spatial regularization, or randomizing values can be used as mitigation.

Different techniques can be used for making the educated guesses. Simple techniques relies on prior geophysical knowledge about the area of sounding, where assumptions can be made for the number of layers in m corresponding to beliefs of their contents, *e.g. sediments, clay, etc.* and their depths. However, this technique renders the post interpretation somewhat useless, as it already assumes some knowledge of where the bedrock would be found. Instead, more complex techniques rather assume a fixed number of layers with increasing sizes corresponding to increasing uncertainty of the sounding at deeper levels. However, another challenge exists, where resistivities of nearby soil may interfere. Thus, regularization is used to provide more soft transitions. Generally speaking, contextual awareness may provide increasingly more accurate inversions. LCI *(Laterally Constrained Inversion)* therefore also considers correlations between both vertically and horizontally neighboring resistivity values. SCI *(Spatially Constrained Inversion)* also does this, but it takes the whole process one step further by also correlating with further nearby values that does not need to be exact neighbors.

Increasingly complex inversion techniques requires increasingly more time to be completed, at the higher end ranging close to weeks of computation time. Simpler techniques are therefore normally used initially to obtain a general understanding before more complex techniques are used.

Electromagnetic noise also poses a challenge in interpretation of AEM data. Christensen *et al.* noted in their paper on the Norwegian highway construction project, that anthropogenic noise invalidated parts of the data obtained during their AEM survey [9]. Causes for such interference may stem from electrical transmission lines, as also exemplified in the paper from NGI [9].

While the topic of this thesis is prediction of DTB values based on resistivity profiles, AEM has also been proven useful in various other settings as well. In 2015, Anscütz *et al.* proposed a system for mapping of hazardous landslide risks with use of AEM surveying [4]. The case study presented in the paper allowed the researches to obtain information about a sliding plane in a matter of weeks, where traditional methods would have required many times both the time and cost. Pfaffhuber *et al.* presented in 2017 a demonstration of how AEM could be used to detect quick clay, which can lead to landslides and in turn result in massive damages [36]. A third article by Pfaffhuber *et al.* presented a use case where AEM was used to detect alum shale [35]. Norwegian alum shale contains large amounts of sulphides and Uranium, making it a potentially toxic mineral due to radon gas. Thus, it can cause significant damage to building projects, as it generally has to be excavated for construction projects to proceed.

2.2 The Usefulness of Automated DTB Interpretation

Linear infrastructure has in recent years become a significant focus area for the Norwegian geotechnical industry [10, p 1]. Norwegian infrastructural organizations such as *Bane Nor* and *Vegvesenet* are continuously carrying out multitudes of large scale construction projects, with one of the recent ones being a combination project for railway and highway called *Ringeriksbanen and E16* [6]. The project's plan description, released in 2018, estimated a total cost of 32 billion Norwegian kroner, and emphasized a determination to work towards cost reductions and optimization over the span of a year [6, p 95].

Researchers from NGI underlined the potential cost benefits of early site investigations with use of AEM in a paper written as part of a project planning phase for the Norwegian Rail Administration, where more than 230 km of railway was planned for construction around Norway's capital. The investigation's primary delivery was a descriptive model of DTB, where the predictions were based on historically successful use of AEM for site investigations [9][1].

To understand why automation within the field of AEM data interpretation may present potential cost benefits, it is of relevance to first understand and appreciate the benefits from using airborne surveying techniques over its traditional ground based counterpart. A brief introduction to AEM and its functionality was provided in Section 2.1. However, its primary benefits can briefly be described to be that the airborne vessel, e.g. helicopter or plane, allows for mapping over large survey areas at low cost. Moreover does its non-intrusiveness allow for large scale surveys to be conducted with little to no interference with nature and wildlife [41]. Nevertheless, its disadvantages are also considerable. The accuracy of the collected data is heavily dependent on the geological complexity of the surveyed area. The interpretation of the collected data is also to be considered a challenging task, relying on complex geophysical and mathematical processes and thorough analysis.

Sparsely distributed boreholes can provide accurate DTB measurements. However by definition, a single borehole cannot present more than a single DTB level for a single geographical point on the surface. Thus, they provide no contextual information about the DTB levels between them. Simplistic methods such as drawing straight lines between DTB points provided by boreholes in a three dimensional space can construct spatially continuous models. However, such models can not be claimed to be accurate when there are considerable distances between the boreholes. In such cases these models can hardly be considered useful. AEM surveys, on the other hand, can cover large areas with intervals between each measurement as small as only a few meters [27, p 1]. Thus, processed AEM data carries the potential of providing great benefits, because it has the potential to construct much more spatially accurate continuous models.

An article from 2016 on the use of AEM in railway corridor mapping for a Norwegian railway construction project found AEM to be a strong candidate for early phase investigations due to its high efficiency, advantageous economic performance, and survey robustness [10]. The same article also notes that resistivity profiles obtained from airborne surveys are also comparable to those acquired from surface ERT (Electrical Resistivity Tomography) techniques, though AEM profiles has a somewhat lower precision in the few topmost meters. ERT provides similar measurements to AEM, but cannot cover equally large areas, as it is ground based. Furthermore, the article emphasizes the significance of accurate processing and inversion of the AEM data. In its closing statements the paper notes the remaining challenge in quantification of depth uncertainties. This challenge of performing automated interpretation in order to obtain bedrock models with uncertainty in meters is, in NGI's paper [10], stated to be missing from both state of practice and established academia.

Traditional methods of obtaining DTB values from resistivity profiles have previously relied on manual labor [5]. A field expert reviews the collected data while accounting for the measurements context, surface observations, and geotechnical expertise [38] [32]. As airborne surveys can cover several hundred kilometers, containing vast amounts of measurements, such manual interpretation quickly becomes both a costly and time consuming task. Moreover, the resulting extracted information may also be biased by the field expert assessing the data.

Anschütz *et al.* noted in 2017 that automated approaches for AEM data interpretation are beneficial for providing first indications, but that they are unfit for engineering purposes that require high precision and accuracy [5, p 13]. The article proposed a novel interpretation technique called *Localized Smart Interpretation*, henceforth referred to as LSI. In short LSI can be described as an interpretation technique that relies on the approximation of a linear operator that can map any resistivity profile to a depth prediction automatically. The paper emphasizes the benefits of the LSI technique to be the freeing of time from manual interpretation, which allows the human interpreter to spend more time on the overall interpretation. Nonetheless, this technique also requires some manual interpretation in order to provide sufficiently precise results.

The industry now desires an investigation of novel automation techniques with the hopes of further reducing the required amount of manual labor for DTB interpretation.

Based on the evidence provided in this section, a hypothesis can be declared that sufficiently accurate automated techniques carry the potential to increase both the time and cost efficiency, but also facilitate, to a greater extent, unbiased reproducibility of results. Elimination of tedious manual work can be noted as a driving factor of the research, where the time of geotechnical experts are considered more valuable if spent on verification, or overall interpretation. Improved efficiency and accuracy of automation could in turn allow for surveying of much larger areas with less requirements for time, effectively reducing the geotechnical expert's work from production and construction, to verification and fine-tuning of results. While the LSI technique showed promising results, it cannot be considered a fully automated approach. Furthermore, it can not provide any measurement of uncertainty of its models, and it cannot account for contextual information for each prediction, making it prone to noise in the data.

Based on the problems and drawbacks presented above, this thesis will focus on improving the process of automated bedrock modelling from processed AEM data. A proposal for an automated solution for contextual interpretation of DTB from sets of spatially continuous resistivity profiles using modern techniques from the academic field of computer vision will be presented and evaluated with a comparison to the current state of the art. The solution is based on NGI's initial work on an MLP regressor for the same purpose [27]. The introduction of an ANN for addressing the problem by NGI was novel, and yielded the benefit of achieving a non-linear interpretation rule, where previous techniques assumed that there existed a linear relationship between resistivity profiles and their corresponding DTB levels [27]. Generally speaking, the introduction of neural networks allowed for the automatic interpretation to account for more complex geological settings, and further reduce the requirements for manual work in the interpretation process.

2.3 Challenges Related to Automated DTB Interpretation from Resistivity Profiles

Resistivity profiles are noisy, and not an exact form of measurement. The data from inversion can not be said to accurately depict the subsurface, it merely provide a set of approximated resistivity values for increasingly deeper levels, with increasing uncertainty [27, pp 1]. Various sediments, materials, and different combinations affect the electromagnetic pulses from the survey differently. Furthermore, overlying sediments will greatly affect the resistivity measure for underlying materials. Thus, there is no exact one-to-one correlation between resistivity values and a given material, which poses challenges for automation.

In general, bedrock provides higher resistivity than sand, soil, and clay. Thus, a simplistic method for attempting to extract DTB from resistivity profiles could simply be looking for data-points with resistivity between some set threshold for which we would denote that we have found bedrock. This method was evaluated by researchers at NGI in collaboration with the Norwegian Railroad Administration in 2017, when they were tasked with establishing DTB information for an area with little prior geotechnical knowledge [5]. The investigation was necessary to facilitate the planning of a major expansion of the railroads surrounding the Norwegian capital, Oslo, in order to more efficiently be able to organize commuting from the surrounding municipals.

The threshold technique is described as a simplistic method where an upper and a lower threshold value are set for expected resistivity of bedrock in the area under investigation. The depth interval corresponding to the resistivity values which fall within the threshold bounds are thus predicted to be bedrock.

The paper from NGI on automated bedrock mapping mentions an automated algorithm from the software system Aarhus Workbench, which was used to obtain the values for the upper and lower resistivity threshold [5]. The technique benefits from being relatively quick and simple to apply, and also being easily reproducible. However, the choice of threshold is subject to individual bias by the geotechnical expert performing the analysis, and the technique cannot cover complex geological settings where the geological topography may vary to a larger extent.

While the method has commonly been used, it does present a set of challenges. Firstly, the thresholds for which in between one could actually find bedrock varies extensively from one geological area to another. Anschütz *et al.* also noted that threshold resistivity can vary considerably over even small areas [2]. Thus, manual exploration of the geological setting is still necessary, and manual intuition must be used to find valid thresholds for the geological area and setting. Furthermore, the method breaks down under anything else than what can be considered simplistic geological settings. The technique has, however, been used successfully under somewhat complex geological settings, albeit alongside manual picking guided by data-points from boreholes, as documented for a Norwegian highway project by Anscütz *et al.* [3] and Anschütz *et al.* [2].

To understand the circumstances for under which the simplistic threshold method will not be sufficient, a closer look at the AEM data visualization presented in Figure 2.2 follows. The following paragraphs will provide a manual and human interpretation of the figure, and denote where, and how, the threshold technique will fail to accurately convert the measurements to DTB values. The goal of providing this interpretation is to present the reader with an understanding of the intuition that geotechnical experts use when interpreting such resistivity profiles. It is exactly this intuition which is difficult to capture in an algorithm or computer program, and makes the task difficult to automate.

The figure depicts resistivity profiles along the horizontal axis, where each resistivity profile is layered down to a depth of approximately 400 meters below the surface. The vertical axis represents the distance from the surface to the layered resistivity measurement. The measurements are done for a straight line on a map, as covered by a flight-line. The vertical bar on the far right provides a mapping from measured resistivity to color codes, which allows for the visualization of the measurements.

An observant reader might note that there exists white gaps within the chart, also noted

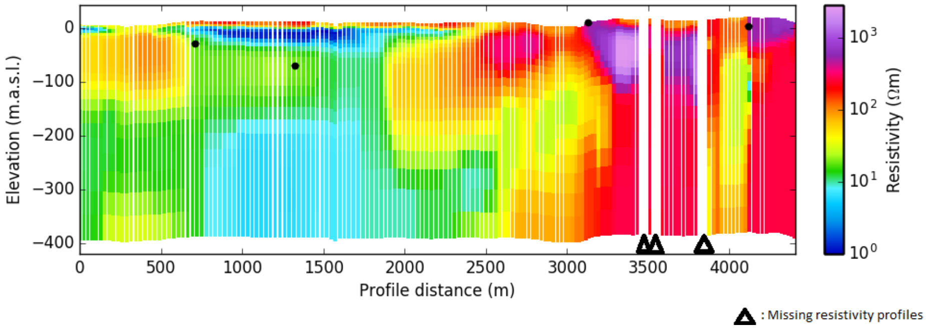


Figure 2.2: Visualization of AEM collected data.

in the chart by black triangles near the horizontal axis. These white gaps denote missing resistivity profiles. The reason for why these gaps exist can be varying. For example could it be that there was not taken any measurement at that exact location. However, measurements can also be stripped away manually by the interpreter, if they do not seem to provide accurate results. Inaccurate measurements can be caused by different scenarios, but any electrical interference commonly presents inaccuracy. In the example of Figure 2.2, the gap within the range from 3450 to 3500 meters could for instance be caused by the airborne vessel crossing over a high voltage power line. Missing resistivity profiles are not uncommon, and thus the interpreter must be able to understand the context in order to *fill in the blanks*.

Four black dots can be found within the chart, located at approximate distance values of 600, 1400, 3100 and 4000 meters. The black dots represents true bedrock depths, with accurate data obtained from boreholes close to the measurements' geographical coordinates.

A geotechnical expert can use both expert knowledge and human intuition to draw relatively accurate conclusions on DTB provided AEM data and precise borehole information. Where an automated algorithm might evaluate a single resistivity profile, a geotechnical expert may consider contextual information that would influence the prediction. The DTB on two sequential measurements within close proximity would for example probably not vary more than a couple of meters, as steep drops in bedrock topology are extremely rare.

For the manual interpretation of this figure, let's start from the rightmost edge and move towards the start. Remembering that solid bedrock typically yields high resistivity values, one can quickly interpret the high resistivity at the surface in the range from 2500 to 4500 meters to represent bedrock near the surface and outcropping bedrock. The low resistivity values between 100 meters and 300 meters depth at approximately 4000 distance meters are most likely faulty and inaccurate values. A threshold technique would most likely handle this exact range quite well, by simply denoting the uppermost part of each resistivity profile where the resistivity was above $10^2\Omega$. However, the next range from 2500 to 500 meters shows how the technique breaks. A thin high resistivity layer is found throughout the surface of this range, located above a 50 meter thick layer of low resistivity sediments. These resistivity profiles are also increasingly difficult to interpret, as there is no layer with clearly high resistivity values indicating bedrock, there are merely gradually increasing resistivities topping at the green level around 100 meters depth. Such decreasing resistivity values with depth are common whenever the overlying sediments, such as e.g. clay or sand, provide low resistivity values. For such cases, exemplified by the range from 2500 to 500 distance meters, simple thresholds will not be sufficient to provide an accurate prediction of the DTB, as the thresholds would be defeated by the top layer, which is not true bedrock.

A human interpreter could quickly observe that the top layer would in fact not correspond to bedrock. Such thin, high resistivity measurements at the topmost layers can be caused by rocky surfaces or even bodies of water. While a single measurement within this complex range might be challenging to interpret, its surrounding measurements and the general context can provide intuitive clues allowing for smarter predictions. A human interpreter could intuitively note the more clearly defined bedrock at the measurements located at 550 and 2000 distance meters, and visualize a connection between these. Additionally, a human interpreter could make use of the a priori knowledge of the boreholes, providing accurate measurements at 550 and 1450 meters to guide their connection between these points in a curved manner.

The final range from 600 to 0 meters provides a more clear representation of bedrock which can be easily plotted. Thus, a final interpretation for the DTB along the measured line could be visualized such as depicted in Figure 2.3.

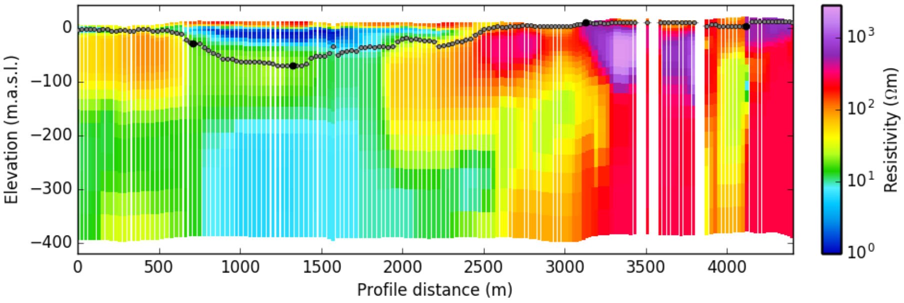


Figure 2.3: Visualization of AEM collected data with DTB interpretation plotted.

In summary, the problems with automation discussed in this section are significantly related to lacking use of expert knowledge, contextual awareness, offset measurements, and human intuition. While there exist various other techniques for extracting DTB models from AEM data, cognitive methods, manual labor, and handpicking has previously been shown to provide accurate models [21][38].

Alternative and improved methods to the threshold technique exist and are currently being used. Each does, however, have their benefits and downsides. These are further discussed in Chapter 3 on State of the Art.

Deliveries of bedrock models typically require a certain degree of accuracy and certainty. There are many factors that can contribute to the uncertainty level of DTB predictions for any given position. These factors may include the analyst's previous experience within the field, prior geotechnical knowledge about the contextual area, and distances to boreholes with known DTB values within close proximity.

As new techniques are continuously emerging for increasingly automated conversion of AEM data to DTB values [36][27], the industry is faced with new challenges in assigning uncertainty values for each DTB value. Manual or semi-automated approaches allow for manual assignment of uncertainty values for DTB predictions. The values, similarly to manual DTB predictions, may be biased by the human behind the interpretations. Thus, reproducibility of the results can become troublesome and can even be seen to rely on a single person. Moreover, assignment of the uncertainty values are also a time consuming task.

Thus, the rules that guide a geotechnical expert's interpretation of a spatially continuous set of resistivity profiles are not straightforward. Firstly, when setting a new plot for a resistivity profile, the expert does not solely consider the profile for which he is setting the plot. Neither does he consider only the direct neighbors of that profile, but rather a wide array of neighboring profiles. Thus, for an ANN to attempt to mimic this behavior of contextual interpretation, it would need to be able to account for some number of neighboring profiles.

The problem of obtaining such DTB predictions can be approached in different manners. Different types of ANNs are these days quite common tools for solving regression problems [13] [14].

2.4 Software & Tools

There exist an abundance of different software and tools that provide frameworks and approaches for analyzing and working with geotechnical data. Thus, only a selection of the most relevant tools have been selected for investigation in this thesis.

2.4.1 Aarhus Workbench

Aarhus Workbench is a software package developed by the danish company Aarhus Geosoftware. The most basic implementation consists of a framework that can house modules for more specific operations. However, its core functionalities can be noted to be inversion of EM data and visualization of geotechnical data. Visualization and geological interpretation of inversion results are done via a GIS (Geographic Information System) interface. For the sake of this thesis, it is only of importance to bear in mind that the original AEM data is processed by this tool for construction of the resistivity profiles that are further used in the experimental approaches.

2.4.2 ArcGIS & ArcPy

ArcGIS is a location based analysis and visualization platform. Its main goal is to facilitate extensive geographical analysis of data. ArcGIS is used within many different industries. Basically, it seeks to provide a framework for working with any location based data. GIS can allow data to easily be plotted onto maps to provide an intuitive interface for location based analysis.

ArcPy is a python package that allows third parties to make use of functionality from ArcGIS in their own software. ArcPy provides much functionality straight out of the box, allowing the users to quickly make use of geographical algorithms and functionality without having to start from a blank sheet. It contains modules, classes, and functions, only a few of which is currently used for state of the art inversion data analysis. ArcPy is closely related to ArcGIS, and their output formats are interchangeable.

2.4.3 NGI's DTB Plugin

NGI has made use of ArcPy to implement their own module for depth to bedrock tracking from the inversion results from Aarhus Workbench's inversion module. NGI's module covers several additional aspects, but only the functionality related to DTB tracking is further covered in this thesis. The three core approaches for DTB tracking currently provided by NGI's modules are Manual Spline Interpolation, described in Section 3.1.1, Localized Smart Interpretation, described in Section 3.1.2, and an ANN regressor, described in Section 3.2.2.

2.5 Data Description

This section aims to provide the reader with an understanding of the raw data that is used in the research. Several datasets are used in the case studies in this thesis. Each of these were provided by NGI. One of the datasets originated from a Norwegian highway construction project from 2013 [9]. The data in each of the provided datasets largely follows a similar structure, and this section uses the highway construction project's data to provide the reader with a basic understanding of how the data is organized and what is represents.

The data can largely be divided into two sets. The first dataset contains data from boreholes that were conducted during the project. The second dataset contained a set of resistivity profiles spatially mapped by corresponding coordinates.

2.5.1 Borehole Data

NGI stores data from known boreholes in a large database. The provided dataset contains data from all known boreholes that were located within a 300 meter radius of any AEM sounding from the resistivity profile dataset.

There are several different approaches for obtaining DTB values from borehole data, and the degree of accuracy varies with the methods. Two of these methods are called Total Soundings and Rotary Pressure Soundings, where Total Soundings are considered to be highly accurate.

The data was provided in a Microsoft Excel format. Each data point contained much meta data, but the relevant columns for each data point are listed and described below.

- **FID**: Identifier denoting the line number of the data point in the original dataset.
- **Method**: Identifier denoting the type of method used in the conduction of the borehole. The identifier corresponds to the methods described in following sections.
- **Stopcode**: Identifier denoting what caused the boring to stop. This provides an indication of the accuracy of the DTB estimate. The reasoning is further described in the following sections on the different boring methods.
- **Depth**: Drilled depth in meters.
- **X**: Distance in the eastern direction in meters from a reference point.
- **Y**: Distance in the northern direction in meters from a reference point.

2.5.1.1 Rotary Pressure Soundings

The Rotary Pressure Sounding (RPS) method was developed by NGI and the Norwegian Public Road Administration in 1967 [7]. The technique is modified to fit Norwegian soil and subsurface conditions. The bit of a multipurpose drilling rig is forced through the soil at a constant penetration rate of 3 meters per minute and a constant rotation speed of 25 rotations per minute. Depending on the soil and the various sediments that the bit encounter, the thrust needed to maintain the constant penetration rate may differ. The thrust versus depth profile is plotted for the entire depth of the borehole. The RPS method cannot penetrate bedrock, and thus it assumes bedrock when it cannot further maintain the penetration rate. The method is fairly accurate, however, subsurface obstacles such as large boulders may also result in inability to maintain the penetration rate, and thus also result in inaccurate measurements.

2.5.1.2 Total Soundings

Total Sounding is a more novel method, and it combines the aforementioned RPS method with rock control drilling [16]. Where the RPS method is unable to penetrate bedrock, Total Sounding excels in having that ability. Boreholes where total sounding is used can thus verify bedrock depth by drilling an additional 3 meters into the bedrock, in order to ensure that the increased resistance does not stem from a subsurface obstacle. It may occur, nonetheless, that this verification is not conducted. This may be the result of time restrictions or other interruptions. A stop code is recorded for every drilling, denoting whether or not this verification has been executed. If the stop code represents that verification was conducted, the DTB value can be recorded with a high degree of certainty.

2.5.1.3 The Borehole Dataset

The provided dataset contained the complete set of nearby boreholes with related data. The set had been pre-processed by NGI to filter away bad data points and average duplicates which might have occurred in the dataset.

The dataset contained a total of 1240 boreholes with relating DTB values. The table below shows the distribution of the different methods and their related verification.

Figure 2.4 presents a 3D projection of the complete borehole dataset. Each data point represents a DTB value at a given point in the survey area. The X and Y axis of the plot denotes meters in the eastern and northern directions where the origo represents the westernmost and southernmost location where an AEM sounding point was taken. This

Borehole Type	**Assumed DTB Values**	**Verified DTB Values**	**Total**
Total Sounding	16	120	136
Rotary Pressure Sounding	1097	0	1097
Other	7	0	7
Total	**1120**	**120**	**1240**

Table 2.1: Borehole Distribution

location will henceforth be referred to as the reference point. The Z axis represents meters below surface.

The chart provides some general important information about the survey area and the borehole data in general. A first note to make is the size of the survey area. The area as seen in the plot below spans more than 20 kilometers in the eastern direction and more than 15 kilometers in the northern direction.

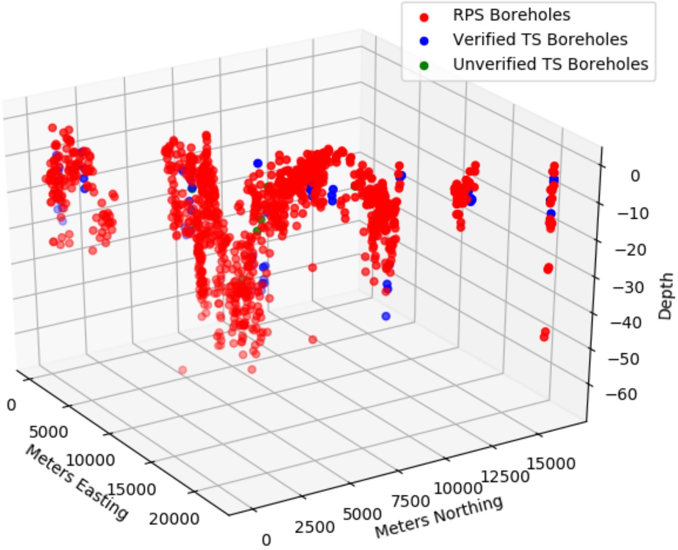


Figure 2.4: Boreholes plotted by method

While the distances in the above figure are needed to present a plot containing all boreholes, they do not correlate to the actual size of the survey area. Figure 2.5 shows a top down view of the boreholes on a map. As seen on this image, only a fraction of the aforementioned area can be regarded as the actual survey area.

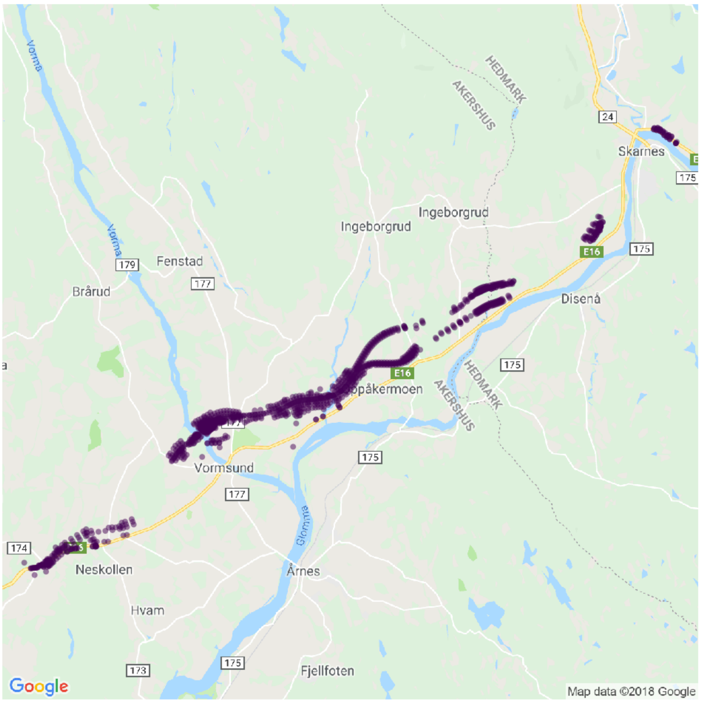


Figure 2.5: Boreholes plotted on map

2.5.2 Resistivity Profiles

Inversion data from the AEM soundings conducted in relation to the highway project were also provided. The AEM data is the same as that which was used during the project, however, the inversion data used in the project differs from the inversion data used in this thesis. NGI has developed more accurate and precise inversion techniques since 2013. New techniques were thus used to derive the inversions that were used for the sake of this thesis.

The inversion of the soundings were done in Aarhus Workbench, and the data was provided in a proprietary data format as output from their software. Each data point contained much data and meta data, only some of which were used during this study. The relevant fields for each data point are listed and described below.

- **Line**: Each sounding is related to a single flight line. Each flight line is generally a relatively straight aerial line covering an arbitrary length and an arbitrary number of soundings. Each sounding in a flight line is typically performed with $\tilde{3}$0 meter intervals.
- **X**: Distance in meters in the eastern direction from the reference point.
- **Y**: Distance in meters in the northern direction from the reference point.
- **FID**: Identifier denoting the line number representing the data point in the original dataset.
- **Topo**: Surface elevation above sea level in meters.
- **Alt**: Altitude of the airborne vessel above the surface during the sounding.
- **RHO_I_**$\{i\}$: Resistivity measure for the *i*'th depth layer of the sounding, where *i* is in the range 1-25.
- **DEP_TOP_**$\{i\}$ Upper bound in meters for the *i*'th depth layer, where *i* is in the range 1-25.
- **DEP_TOP_**$\{i\}$ Lower bound in meters for the *i*'th depth layer, where *i* is in the range 1-24.

Figure 2.6 presents a visualization of a subset of the data points from the original dataset. Each vertical bar in the figure denotes a single inversion data point from a selected flight line in the survey area. Each vertical bar consists of 25 smaller vertical color coded bars. The 25 smaller bars are vertically distributed, each with increasing ranges. The colors on these bars denote the degree of resistivity for that depth layer of the sounding. The uppermost depth layers are smaller, starting with a range of 1 meter, while the ranges increase with their respective depths. The gradually increasing ranges are set due to increasing uncertainty at deeper subsurface levels. The color coded visualization is set at a logarithmic scale due to the large resistivity difference of subsurface sediments and rocks.

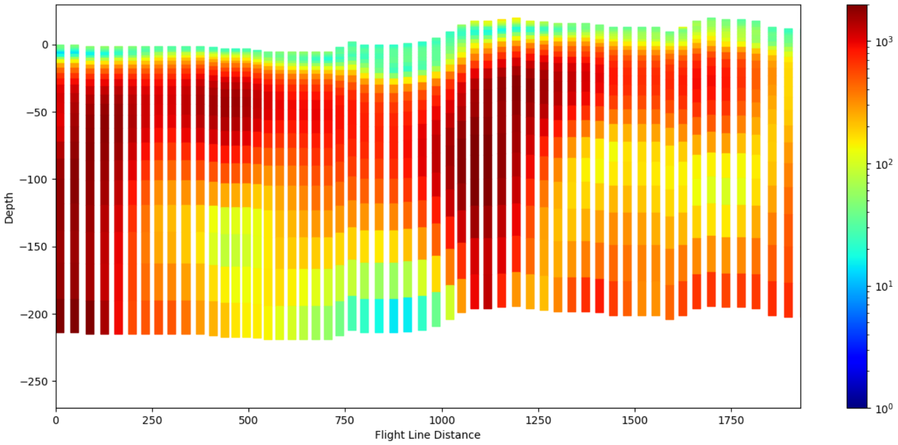


Figure 2.6: Resistivity Profiles for a single flightline

2.6 Interpolation & Kriging

Any ML technique that uses a supervised learning approach requires a set of labeled data points. In essence, this means that a data set must exist with normal input data, but the value that we want to predict must also be known. For the sake of DTB interpretation, this correlates to knowing the DTB level as mapped to resistivity profiles.

Thus, the aspect of obtaining labeled, or true data-points must therefore be considered. Consider the example depicted in Figure 2.2. In this example, four black dots represent known bedrock depths, as provided by boreholes. Assume now that an ANN model exists and accepts as input the single resistivity profile of evaluation. To perform supervised training of this model, a dataset containing single resistivity profiles as input data with respective true depth values is required. Using the resistivity profiles and borehole points from Figure 2.2, one could quickly find the profiles where borehole points exist, label the input with the depth of the borehole and use the obtained dataset as training data. Unfortunately, such cases where boreholes are found directly under AEM measurements rarely, if ever, happen in real world scenarios. In other words, there will not exist any labeled resistivity profiles, because no AEM sounding will have the exact same X and Y coordinates as any borehole. Thus, there is a need for an approach which can map the known depth labels to resistivity profiles that can be used in the derived model. Consider the map presented in Figure 2.7 below. This figure depicts a top-down view of a map, where the AEM soundings are plotted with black squares, and borehole locations are plotted with red circles.

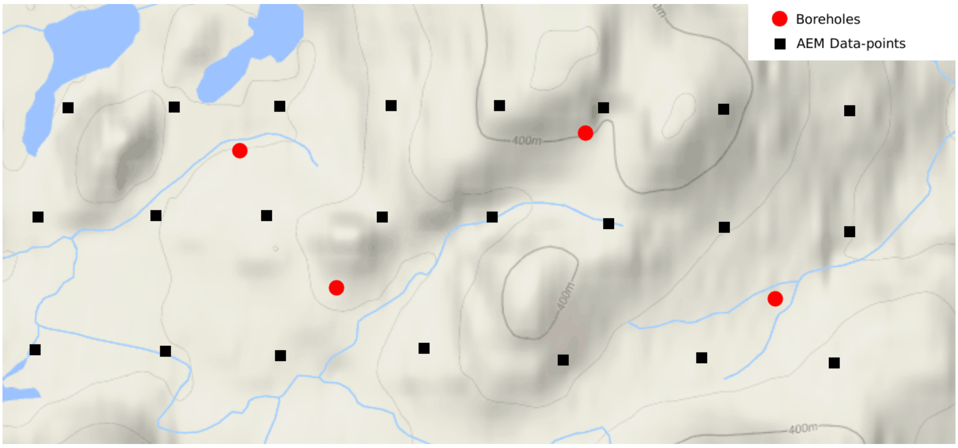


Figure 2.7: Example map of a surveyed area.

Recall that the goal is mapping the known labels which are plotted in red color, (borehole depths), to resistivity profiles that could be used with the ANN model. A simplistic approach could simply relate borehole depths with the nearest known resistivity profile, such as depicted in Figure 2.8. This approach would provide a simple method for relating a single resistivity profile for each borehole. However, much uncertainty is introduced as the distance is not properly accounted for. A threshold technique where any relations with distances larger than the threshold between them would be discarded may mitigate this problem. Though, this would lead to much discarded data, and there is no contextual awareness present.

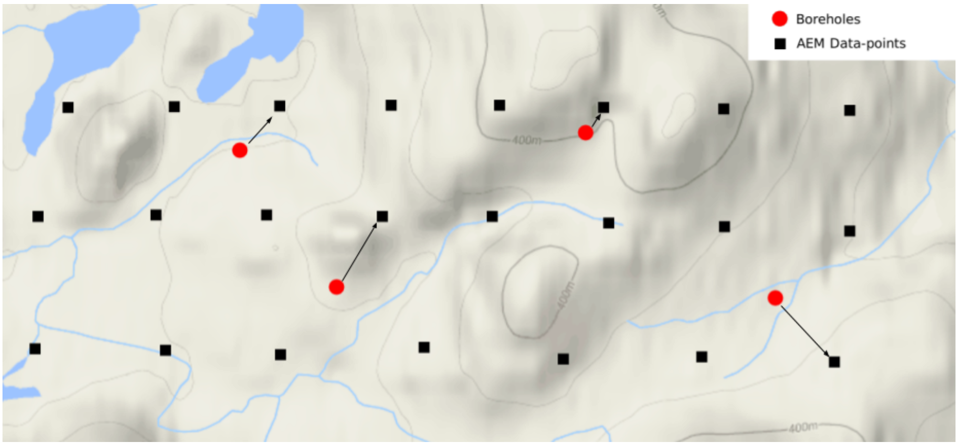


Figure 2.8: AEM data points related to boreholes by projection.

A better solution is the interpolation of multiple nearby resistivity profiles, resulting in a single resistivity profile for each borehole by IDW, such as shown in Figure 2.9. The number of nearby data points to use may be limited both by a distance threshold and a maximum number threshold. Recall the layered structure of a resistivity profile, each consisting of the same number of layers. To interpolate a single resistivity profile from the nearby ones, a weighted average can be computed for each layer, weighted by their distances. This approach can thus account for the distances between the data points. Moreover, it can also consider contextual differences to some extent. However, exceptions may occur, where also this approach may be biased and of poor accuracy. An example may be a case where all nearby AEM soundings of a borehole may be located on a single side of the borehole. For such a case, an interpolation of the nearby soundings will, to some extent, consider contextual awareness as the resulting profile is an interpolation of its surroundings. However, it can not be claimed to accurately represent the resistivity profile of the point of interpolation if resistivity profiles on the other side present vastly different values.

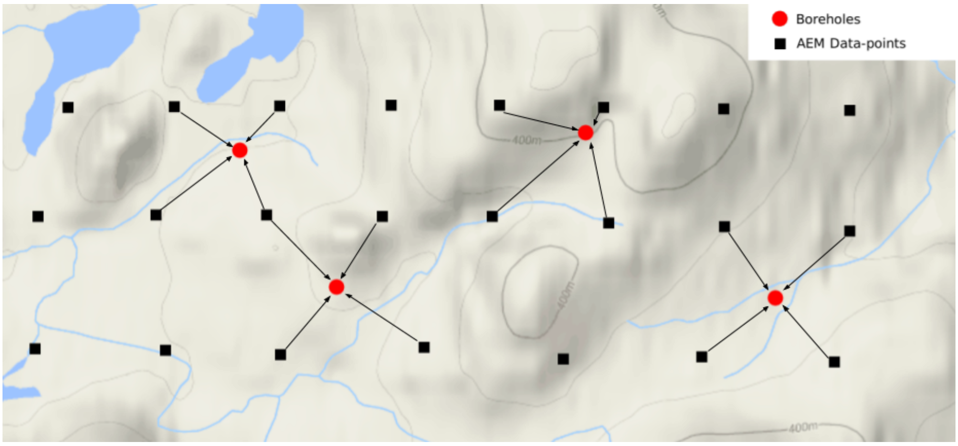


Figure 2.9: AEM data points related to boreholes by interpolation.

Current state of the art takes this one step further with the use of Kriging for this purpose [9]. This technique is further discussed and elaborated on in the next section.

The approaches presented in this section mitigates the problem of obtaining training data for a model that takes as input a single resistivity profile. Consider now a contextually aware model that provides DTB predictions from sets of neighboring resistivity profiles. The problem of obtaining training data for such a model further complicates the issue, and is central to the technique of automation presented in this thesis.

2.6.1 Spatial Data & Interpolation with Kriging

The First Law of Geography, according to Waldo Tobler, is *"Everything is related to everything else, but near things are more related than distant things."*

Recall from the previous section that the task of relating known DTB values from boreholes to a single resistivity profile presents its own set of challenges. Furthermore is also AEM data prone to noise [9], which effectively means that resistivity profiles obtained from automated inversions of the raw data may contain inaccuracies.

NGI conducted a project in 2015 with the task of supplementing geotechnical investigations for a highway construction project in Norway. A research was conducted in relation to the construction project, where the aim was to develop an automated algorithm for extracting DTB using data from both AEM surveys and boreholes [9]. The research proposed and presented the use of the geostatistical spatial interpolation technique, Kriging, for the problem of relating borehole depths to corresponding resistivity profiles.

The problem which Kriging attempts to solve in the aforementioned project is, in simpler terms, to estimate an unmeasured resistivity profile based on spatially surrounding measured samples. Kriging carries some similarities to the somewhat simpler technique of IDW, as described in the previous section. Recall that in IDW a weighted average is computed of the surrounding data samples, and the weighting correlates to the distance from the data point of estimation.

Equation 2.1 describes the overarching problem which both IDW and Kriging solves, where $\hat{z}(x_i, y_i)$ is the predicted value at coordinates (x_i, y_i), N is the number of true values as obtained by measurement, and λ_j is a weight that denotes the degree of which the true value at j, $z(x_j, y_j)$ should affect the prediction. Similar for both techniques is that the weights sum to 1.

$$\hat{z}(x_i, y_i) = \sum_{j=0}^{N} \lambda_j (z(x_j, y_j)) \tag{2.1}$$

IDW and Kriging differs, however, in the way they computes the weight parameters. IDW sets the weight parameters corresponding to the spatial distance between the point of prediction, (x_i, y_i), and the true data points, (x_j, y_j). Hence, IDW can be described as a deterministic interpolation method, since the resulting interpolation is directly related to the surrounding data points. Kriging, on the other hand, also adjusts the weights corresponding to spatial variance. A function is initially fitted to represent a generalization of the spatial variance, which is later used for approximating $\hat{z}(x_i, y_i)$. The fitting of this model is done by analysis of a semivariogram. A semivariogram presents the semivariance at incremental distances, or lags. The semivariance for each lag is half of the variance for each pair of data points, where the distance between the two data points is less than the number of lags multiplied by the lag distance. This is represented by Equation 2.2, where $SV(D, l_d, l)$ denotes the semivariance for the data set D at the l'th lag where each lag is of distance l_d. P is the set of all pairs of data points in D, such that the spatial distance between each pair, $P^i \in P$, is less than the distance obtained by $l \cdot l_d$, but larger than the distance obtain by $(l-1) \cdot l_d$. N denotes the number of pairs in P, while $z(P_0^i)$ and $z(P_1^i)$ represent the values of the two data points in in P^i.

$$SV(D, l_d, l) = \frac{1}{2}(\frac{\sum_{i=1}^{N}(z(P_0^i) - z(P_1^i))^2}{N}) \tag{2.2}$$

The X-axis on the semivariogram represents the distance, while the Y-axis represents the semivariance. The lags effectively provide borders for the data samples to be included in the calculation of the semi variance. The borders separate the data point pairs such that the semivariance for the first lag is computed by only the pairs whose distance is less than $l_d \cdot 1$. The semivariance at the second lag is computed solely by the pairs whose distance is larger than $l_d \cdot 1$, but less than $l_d \cdot 2$, and so forth.

Figure 2.10 presents an example of such a semivariogram. The nearby points, which corresponds to left on the X-axis, typically carries lower semivariances than pairs of points with larger distances between them, (further right on the X-axis). This corresponds to Waldo Tobler's law, which emphasizes the fact that geographic data that are closely spatially located tend to be more similar than ones that are further spatially distributed. Thus, data points that are located at the high end of the X-axis should, by Tobler's principle, have higher Y-values.

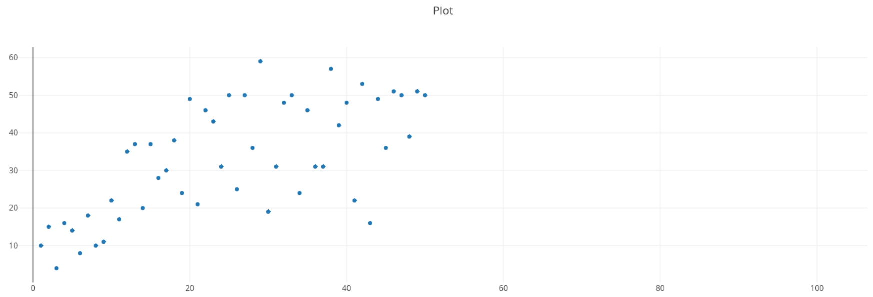


Figure 2.10: Semivariogram with manually generated data

Recall that Kriging interpolation relies on a model that can represent the data that is visualized in this diagram. The fitting of a such a generalized model consists of selecting a model type to use for the modelling, before approximating the parameters. Some of the model types that can be used are listed below.

- Gaussian
- Exponential
- Spherical
- Linear
- Power
- Hole-Effect

The type of model that is most fit may vary depending on the problem, and experimentation is commonly used to find the model that best represents the plotted data points in the semi-variogram.

The parameters used to define the semivariogram models are the distance, d, the partial sill, p, the range, r and the nugget, n.

The distance corresponds to the values on the X-axis in the semivariogram. The distance at which the model is flattening, or no longer increasing in the Y-axis, is known as the range. The value of the Y-axis when the X-axis is at the range is known as the sill. The nugget is the value of the semivariogram at $d = 0$. The partial sill is the sill minus n.

The gaussian model function is defined in Equation 2.3, and the exponential model function is defined in Equation 2.4. Both equations assume $d > 0$.

$$gaussian(d, p, r, n) = p \cdot (1 - e^{-\frac{d^2}{(\frac{4}{7}r)^2}}) + n \tag{2.3}$$

$$exponential(p, d, r, n) = p \cdot (1 - e^{-\frac{d}{r/3}}) + n \tag{2.4}$$

By trial and error the parameters can manually be fine-tuned such that the chosen model accurately represents a generalization of the data points in the semivariogram. However, manual trial and error may quickly become time consuming, and thus different forms of gradient descent techniques minimizing some cost function for some loss, e.g. the Mean Squared Error loss, can be used to automate the parameter approximation.

Interpolation predictions can then be made, after the semivariogram model has been obtained by selecting a type and approximating the parameters. The process of producing an interpolation for any geographic point is similar to that of IDW, but the weights for the interpolation can now be obtained by the approximated model. Thus, consideration of spatial variability is included in the predictions.

2.7 Representing the Uncertainty of Predictions

The added value that automated DTB predictions from AEM data presents are impeded by the lack of automated computation of respective uncertainty values. This is because the usefulness of any bedrock model is reliant on its accuracy and precision, as such models are commonly used in projects that require high precision and accuracy. A bedrock model with no knowledge of precision or certainty is thus less valuable.

2.7.1 Uncertainty in Regression Type ANNs

A theoretical limitation of standard regression type ANNs is that they do not provide any information about how confident their predictions are, as they only output a prediction based on a best fitting regression. The importance of quantification of such uncertainty is not specific to geotechnical areas such as bedrock modeling, and are evidently present in many other real world applications where ANNs or other ML techniques are used [12][24].

The uncertainty that is related to the depth predictions from artificial neural networks can largely be split into two main categories. Since the resistivity profiles can be considered mere noisy approximations, the first category of uncertainty can be described as aleatoric uncertainty, which represents the uncertainty that stems from the inherent stochasticity in the data (resistivity profiles and DTB). The second category of uncertainty is related to the misspesification of the model that is being used for estimating the depth predictions. This category is typically referred to as epistemic uncertainty, and can in theory be mitigated by expanding the size of the training data set. The epistemic uncertainty is not solely determined by the misspecifications of the parameters, (*i.e. weights and biases*), in the model, but also the overall topology of the network. Epistemic uncertainty originating from a network's topology may occur when the topology of a simplistic network does not allow the network to attain the required level of complexity to solve a given problem.

The combination of the two categories is exhaustive, and collectively they provide the total uncertainty.

The numerical output of a regressor in itself may seem inherently precise, carrying numerous decimals. However, a single regression model cannot provide any indication of the model's confidence for its prediction. Thus, an analyst solely taking into account the output of such a regression network will have no way of differentiating between a prediction which is confident, and a prediction which is not. This is simply because the model does not know itself how confident it is. It is nothing more than a regression of the input data, adapted to fit the training data as closely as possible. Since the model does not compute any confidence in itself, obtaining uncertainty values for such a network is a more challenging task.

Based on the discussion above, the industry's desire can be said to be the acquiring of depth interval predictions, rather than only single point predictions. It is of much larger usefulness to provide a depth estimation saying, with 90% confidence we claim the bedrock will lie between these boundaries, rather than a depth estimation saying; the DTB probably lies at this depth, but we have no idea of how accurate that is. Such intervals is known as prediction intervals (PI). The experimental approach of this thesis considers and evaluates a set of approaches for construction of such intervals for the case of DTB interpretation.

As discussed in this chapter, the introduction of uncertainty values related to each prediction can ultimately allow a geotechnical expert to more efficiently perform complex analysis and verification of automated bedrock models. Knowledge about the certainty of each plot in a bedrock model is of high usefulness in planning and further execution of projects that require precise awareness of the underlying bedrock of an area.

2.7.2 Prediction Intervals & Confidence Intervals

A regression network is fit to provide predictions in a continuous output space. However, it is a theoretical limitation that it lacks the benefit of being able to provide any form of confidence level for its output. Still, the predictions *will* be less reliable and accurate if the training data is limited or noisy, i.e. if the combination of epistemic and aleatoric uncertainty is high. Thus, a more complex approach is needed for an ANN regressor to be able to account for an uncertainty range. Recall that there is an industry desire for such an uncertainty range, where predictions in the form of PIs could yield beneficial outcomes. Previous research exist on PI construction for ANN regressors, since many industries commonly relies on information about the accuracy of point predictions for decision making and risk management. By definition, PIs provide an interval for in which an observation will fall by a certain probability given historic observations. It consists of upper and lower bounds that contain the value of a future prediction with a probability known as the confidence level, α, where $0 \leq \alpha \leq 1$, corresponding to some percentile. Knowledge of such intervals allows the interpreters to consider best and worst case scenarios. By definition, wide PIs represent high uncertainty, while narrow PIs represent low uncertainty for the point prediction at that location. Consequently, this knowledge can allow decision makers to avoid selection of risky actions when the uncertainty is high, and allow for more confident decisions when the uncertainty is low.

A few approaches for learning uncertainty in ANN's have been proposed and well established.

For any regression model, a data point at i can be modeled by Equation 2.5, where dp_i is the data point, y_i is the true regression value at i, and ϵ_i is the error, or distance from the actual regression value to the data point's value.

$$dp_i = y_i + \epsilon_i \tag{2.5}$$

Using some model, an estimate of the true regression point at i, $\hat{y}_i$, can be obtained. For a data point at i, we can model the discrepancy between the data point and the estimation of the true regression point, stemming from the model, by Equation 2.6. If $\hat{y}_i$ is 100%

accurate in predicting y_i, the left side of this equation would have the same value as ϵ_i, and thus correspond to the error at i.

$$dp_i - \hat{y}_i = (y_i - \hat{y}_i) + \epsilon_i \tag{2.6}$$

CIs, (Confidence Intervals), represent that for some probability p, the best fit regression, y, for the complete dataset dp lies within its boundaries. Consequently, CIs quantify the uncertainty between the prediction, $\hat{y}_i$ and the true regression point y_i.

PIs on the other hand represent that for some probability p, the true data point, dp_i will lie within its boundaries. Thus, PIs quantify the uncertainty between the prediction, $\hat{y}_i$, and the actual data point dp_i.

Whenever the errors are normally distributed around the true regression, prediction intervals rely on the total variance at i, σ_i^2, which can be divided into two distinct categories, such that $\sigma_i^2 = \sigma_{\hat{y}_i}^2 + \sigma_{\epsilon_i}^2$.

$\sigma_{\hat{y}_i}^2$ represents the uncertainty in the model, the epistemic uncertainty, which encompasses our ignorance in selecting a model that most accurately can explain the data, but also parameter estimation errors. $\sigma_{\epsilon_i}^2$ represents the uncertainty inherently in the data, the aleatoric uncertainty, which can be exemplified with similar data points with differing labels, or noisy data points.

Chapter 3

Related Work & State of The Art

This chapter aims to establish and present related work and the current state of the art in relation to interpretation of geotechnical resistivity profiles.

Section 3.1 covers current established techniques that have commonly been used for interpreting DTB levels from AEM data.

Subsequently, Section 3.2 covers relevant concepts and definitions related to the academic field of machine learning and artificial neural networks. It establishes important theory which is useful for understanding how ANNs are able to address the same problem as the techniques from the previous sections. NGI's ANN regressor is also presented and covered in this section, after the relevant theories have been presented. The section lays ground for important concepts that is used for a proposed new automation technique later in the thesis.

Section 3.3 wraps up this chapter by presenting a set of established techniques for construction of prediction intervals from regression type ANNs.

3.1 Interpretation Techniques

The following sections will provide an overview of two common techniques for DTB interpretation from resistivity profiles.

3.1.1 Manual Spline Interpolation

The manual spline interpolation approach is the most simplistic approach for DTB tracking currently housed in NGI's analysis module. A spline can be described as a polynomial function that is *piecewise defined*. In basic terms this means that the complete interval of the polynomial can be split into sub-intervals, where each interval can be defined by its own distinct polynomial function. The benefits of using a spline approach over a high order polynomial approach for interpolation problems are illustrated in the two plots in Figure 3.1. The problem with the higher order polynomial interpolation can be seen between the two final plots, where the high order polynomial oscillate, resulting in a deep curve. This behavior is commonly known as Runge's phenomenon, and is perhaps the main reason why spline is commonly preferred over high order polynomials for interpolation problems, where the goal is to find an initially unknown value between two points.

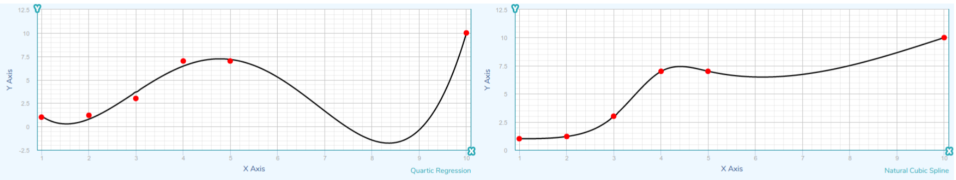


Figure 3.1: Example of regression using 4th Order Polynomial (left) and Cubic Spline (right).

Recall that boreholes are commonly sparsely distributed at early phases of geotechnical site investigations. Assume now that the X-axis in in Figure 3.1 represents an AEM flight line, the Y-axis represents depth, and the red dots depict known depths to bedrock provided by boreholes. In such a case, it might be simple to see that the interpolation provided by the Spline more accurately represents a potential one-dimensional bedrock model for the flight line.

However, an observant reader might again spot that there is a discrepancy between known data and the data used in this approach, as the described process does not account for the AEM soundings' resistivity profiles. Rather, it only uses the bedrock depths at certain locations. This is correct, and also considered a significant drawback of this method. However, the approach as implemented in NGI's plugin takes the process one step further by allowing for further manual interpretation and analysis. It does this by displaying the resistivity profiles behind the interpolation. This allows a geotechnical expert to identify correlations between the known DTB values from the boreholes and the values in the resistivity profiles.

The regression is typically initially constructed by data from boreholes. However, the expert may introduce new data points to more accurately fit the spline in order to account for the resistivity profiles as displayed in the background. This is an example of introduction of bias in the process, as various experts may interpret the resistivity models slightly differently, resulting in potentially varying interpolations. Reproducability is also significantly damaged by this last step.

An uncertainty measurement for each interpolation point can be computed by a similar spline interpolation from uncertainty values along the X-axis, where the uncertainty values correspond to some uncertainty range for that point, as predicted by experts.

3.1.2 Localized Smart Interpretation

Localized Smart Interpretation is an interpretation technique that takes the resistivity profiles into consideration when computing the DTB values. The technique was first proposed by Guldbransen *et al.* in 2015 [15]. The overarching aim of this technique is to construct a statistical model that can describe an assumed linear relation between some arbitrary quantifiable data and some arbitrary geological interpretation (for the sake of this thesis, DTB). This can be written as a function of d and M, $f(d, M)$ which results in a probability distribution that can describe the relation between the quantifiable data M and the interpretation d.

In the case of DTB tracking from AEM inversion results, M would correspond to the resistivity profiles, while d would correspond to DTB values. Thus, d can be seen as a vector of DTB values and M can be regarded as a matrix, where each row represents a vector describing a single resistivity profile. A larger set of boreholes with known DTB values can provide the statistical model with more data, which in turn can be used to improve the parameters of the probability distribution. Essentially this mean that more valid data leads to improved predictions. Current use of this technique may also rely on a geotechnical expert to enter some manually interpreted data points, *M to d mappings*, as the number of boreholes may not be sufficient.

The underlying assumption is that there exist a linear relation between d and M, such that finding an unknown value $d_i \in d$ from known values $m \in M$ corresponds to picking the most likely value from the probability distribution $d_i|m$, which is given by the statistical model $f(d, M)$.

In turn, this means that after the model has been inferred from the aforementioned data points, it can be fed a single instance of quantifiable data that was not known during the inference of the statistical model m, and provide a prediction of DTB d_i. This can be written as $d_i = mg$ where g describes the linear operator connecting them. A concrete approach for computing g can be found in Guldbrandsen *et al.* 2015 [15].

The assumption of a linear relation means that the same rule for interpretation is used regardless of the context and the geological setting of the area. While this provides a well suited approach for interpretation of surveys where the geological setting does not drastically change, it cannot properly account for flight lines where the geological setting change within the area of investigation [27, p 2].

3.2 Machine Learning & ANN Definitions

NGI has previously experimented with the use of simplistic artificial neural networks for automated DTB interpretation of resistivity profiles obtained from AEM data. This section aims to provide the reader with a brief introduction to feedforward regression type artificial neural networks. The introduction provided here is vastly limited. A thorough introduction to the concepts that are briefly described here can be found in the book *Deep Learning* by Goodfellow *et. al* [13].

Massive amounts of neurons are connected in complex networks inside the human brain. ANNs, or *Artificial Neural Networks*, are highly simplistic computational representations of such networks. These are commonly used for approaching both classification problems and regression type problems. While standard algorithmic approaches require the programmer to explicitly state the rules of which the algorithm is to follow, ANNs are able to independently learn approximations of these rules.

An artificial neural network consists of layers of connected artificial neurons. Each artificial neuron in the connected network can receive input signals from its input connections, do some processing on that data and then pass a new signal further to its output connections. The signals that each neuron sends are typically real numbers, and the processing done within each neuron is commonly some aggregation of the input numbers.

The connections between the artificial neurons are typically referred to as edges. Each edge in the network has its own weight, which amplifies or weakens the output of its initial artificial neuron. The weights are adjusted during training of the network, which is how the network is able to adapt to different computational problems.

The processing that is done in each artificial neuron is obtaining the weighted sum of its input, x, followed by performing some activation function on x. There exist a vast number of different activation functions. A quite commonly used activation function is the rectified linear unit (ReLU), where the output is 0 if x is less than 0, and x otherwise, as shown in Equation 3.1a. Another approach, albeit not as commonly used, is to use some activation function that normalizes the output value, such that the output will always represent a value between some interval, e.g. $[0, 1]$. This is exemplified by the Sigmoid activation function as presented in Equation 3.1b.

$$ReLU(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ x, & \text{else} \end{cases} \tag{3.1a}$$

$$Sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3.1b}$$

Furthermore comes the aspect of biases. Biases in ANNs are single values in each artificial node in the network. The bias affects the activation function similarly to each of the artificial node's inputs. However, the bias is not weighted, and it is not affected by anything earlier in the network. In other words, they provide biased values.

Thus, for any artificial neuron in a densely connected artificial neural network, its output can be written as presented in Equation 3.2, where α represents an arbitrary activation function, n represents the number of neurons in the previous layer, a_i represents the ith artificial neuron in the previous layer, and b represents the bias in the neuron. The fact that the network is *dense* means that any artificial neuron in any layer is connected to every artificial neuron in the previous layer.

$$out = \alpha(\sum_{i=1}^{i=n}(w_i \cdot a_i) + b) \tag{3.2}$$

An ANN can consist of infinitely many layers, but will always have *(at least)* one input layer and one output layer. Any layers located between the input and output layers are commonly referred to as *hidden layers*. The number of neurons in the output layer generally varies depending on the networks task. For classification problems the number of neurons in the output layer generally corresponds to the number of possible classifications. For regression problems the output layer typically consists of only a single neuron providing the resulting prediction from the regressor.

To put all of this in perspective, assume an artificial neural network takes as input a vector of 125 numbers, resulting in 125 artificial neurons in the input layer. Assume the network has two hidden layers of sizes 50 and 20, and an output layer consisting of a single neuron. The total number of parameters that needs to be optimized in this network is the total amount of weights and biases that exist in this network. The total number of weights are $(125 \cdot 50) + (50 \cdot 20) + (20 \cdot 1) = 7270$, and the total number of biases are $50 + 20 + 1 = 71$, resulting in a total of 7341 parameters for optimization.

Before the ANN is trained, all of its weights and biases are initialized with random values. An ANN with only random parameters are, unfortunately, not very useful. In order for the networks to learn to produce sensible predictions, a training phase must first be conducted. So how does an artificial neural network learn to make sensible predictions? In a lecture held by Jordan Peterson at the University of Toronto, he claimed that "Every time you learn something, you learn because something you did didn't work[...]" [34]. This quote also applies to how ANNs can iteratively become better by learning from trying.

Different types of learning exist, however, only supervised learning is of significant interest for this thesis, and thus only this is briefly introduced here. During supervised learning, a data set of labeled input data D_1 must be provided, such that D_1^x represent the input data and D_1^y represent the labels for each element in D_1^x. The goal of the process is to adjust each of the edges' weights and neurons' biases, such that the network as a whole is able to predict results that are as close as possible to the labeled data by approximating some function of the input data, $F(x)$, where $x \in D^x$. The ultimate goal of the training phase is for the model to both fit to the training data, but also to generalize, such as to perform well on data points that was never seen during the training phase. This can be tested after training, by providing the network with a second labeled dataset, D_2, and comparing the ANN's predictions with the labels in D_2^y.

While attempting to find the best values for the weights and biases may initially seem daunting due to the sheer number of parameters. It can, however, be simplified a whole deal by breaking down the entire process into smaller sub-processes. The supervised training phase for a regressor contains many cycles where the network first produces a prediction, $\hat{y}_i$ for each input data $x_i \in D^x$, then compares the predictions with the labeled value for that prediction, $y_i \in D^y$. After the comparison has been made, an error, ϵ_i, is computed, which represents the deviation from y_i to $\hat{y}_i$. ϵ_i can be computed in different ways, varying with the objective of the network. The function that is used to compute ϵ_i is commonly referred to as the loss function. The square loss, as presented in Equation 3.3, is an example of a loss function which is often used for regression problems.

$$\epsilon_i = (\hat{y}_i - y_i)^2 \tag{3.3}$$

The Cost function computes a metric of the loss after training over some number of data points. Goodfellow *et. al* notes in his book on Deep Learning [13], that the terms *objective function, criterion, cost function, loss function,* and even *error function* are used interchangeably in his book, which is a statement that is often true in the wild as well, though it is mentioned that some publications assign special meanings to some of the terms. Why no single phrasing for the computation has stuck, one can only wonder. A simple man may be inclined to believe that computer scientists have a tendency to overcomplicate simple terminology. Nonetheless, the metric provides a single value, denoting how good (or bad), the network is currently performing. The cost function can thus also be described as a function of all the parameters in the network, which yields a score for the network as a whole based on the performance of the training data. In mathematical terms, the cost function of the weights and biases in the network can be written as presented in Equation 3.4, where C is the cost function, w is the weights in the network, b is the biases,

N is the number of losses that has been computed, and ϵ_i is the loss at i, or, if MSE is used as the loss function, the squared residual, as from Equation 3.3.

$$C(w, b) = \frac{1}{N} \sum_{i=1}^{i=N} \epsilon_i \tag{3.4}$$

In short, if the output of C is low, the network is performing well. On the other hand, an increase in the output of C means that the network is performing worse. Thus, the goal of optimizing the model can be described as the equivalent of *minimizing the cost function*. In other words, the goal is to find the set of parameters, i.e. weights and biases, that produces the lowest value for C. Achieving this is done by an approach called *gradient descent*, where the parameters of the model are iteratively tuned, each step aiming to reduce the output of C. The most common learning algorithm, which incorporates the idea of gradient descent, is the backpropagation algorithm. In the backpropagation algorithm, each weight and bias in the network is evaluated as to whether it should increase or decrease, and also with which magnitude, as to improve the model's performance. The desired changes for each weight and bias are averaged after all the data points have been checked.

Modern approaches rarely iterate over the entire training data set for each computation of the gradient, as it would take rather long time with complex models and large data sets. Instead, the training data set is often shuffled, then split into smaller batches, and each batch is evaluated before the gradient is computed and the parameters shifted. While this approach does not find the optimal shift after each batch, it generally allows the models to converge much faster.

3.2.1 Deep Learning & Deep Neural Networks

Deep learning has, in the later years, dramatically improved the state of the art in many technological areas [25]. Deep learning is a broad subset of all machine learning techniques. With respects to neural networks, a network is by definition considered *deep* if it consists of more than one hidden layer, excluding the input and output layer. The best number of hidden layers to include in any model may vary with the network's objective and the complexity of the task. Too many hidden layers may lead to overfitting, while too few layers may result in the network not being able to learn the required ”rule-set”. In layman’s terms, one could say that deeper neural networks are able to use more rules to analyze its input than shallower ones.

3.2.1.1 Convolutional Neural Networks

Convolutional neural networks, (CNNs), are a subset of all deep neural networks, which has played an integral part of recent progress in computer vision problems [25]. CNNs are perhaps most commonly used in image classification problems, where layering of convolutions may allow a network to account for increasingly more complex patterns. In a layered CNN consisting of three convolutional layers, the complex problem of classifying a digit may be realized by each layer performing a different, and increasingly complex task. The first convolutional layer may only detect lines and edges. The second layer may identify combinations of the lines and edges detected in the first layer, allowing it to identify more complex shapes such as corners or curvatures. The last convolutional layer may then be able to identify combinations of the shapes from the second layer, corresponding to digits.

Assume an image is represented by a matrix, and each cell in the matrix is representing a color value. The mathematical operation of a single convolution would then consist of two input matrices, one being the original input image, and the second being the *kernel*. A single matrix is computed and output, typically referred to as a feature map of the original image by the kernel. The kernel is a matrix containing a small pattern. The kernel is then matched with patches of the original image to produce the product of the convolutional operation. This process is visualized in Figure 3.2, where the colored cells corresponds to ones, and white colors correspond to zeros. The product, as output in the output filter, or feature map, is the element-wise product and sum of the kernel and the patch of the input image.

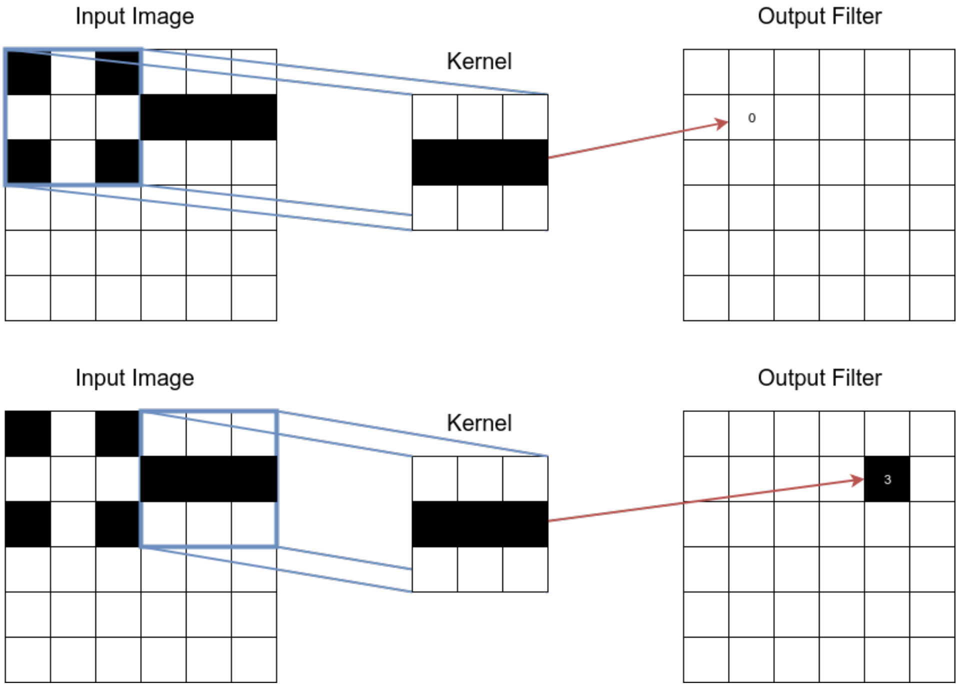


Figure 3.2: Visualization of convolution for a single kernel over two different patches of the input image (Not two subsequent patches)

The approach presented in Figure 3.2 shows how one kernel affects one cell in the feature map based on the input image. Let I denote the complete image, and $I_{i,j}$ represent a 3 by 3 patch of the image, centered around the ith cell from the top, and the jth cell from the left. In the first example from the figure, the convolution operation is performed on the patch $I_{1,1}$ by the kernel, k. The result of the convolutional operation, following element wise multiplication and sum, is 0. The value for the second example is 3. The same procedure is followed for every individual patch in the input image. The resulting feature map thus provides a map where each cell in the map reports the degree of match in the input image, at a single location, for a single feature.

There can exist many different kernels in a *feature set*. The process described above is repeated for each feature in the feature set, each resulting in a different feature map. The values of the feature matrices are learned during the training phase in the same way as the weights and biases in the model.

Straight forward convolution will, by definition, shrink the size of the original image. This is a direct consequence of the process requiring surrounding values for computation of the value of a cell in the feature map. This is fine for some use cases, but other cases may rely on retaining the original size of the input data, or desire to retain as much data as possible. For such cases, *padding* extra layers on each edge of the input data allow for convolution over the outermost cells. Thus, allowing for the retention of the original size.

Pooling is an additional layer that is often used alongside convolutional layers. Pooling effectively reduces the resolution of a feature map, by computing some aggregation over patches in a feature map. The most common type of pooling is perhaps max pooling, where only the cell with the highest value is retained in the output of the pooling layer.

3.2.2 ANN Regressor

A study published in 2018 by researchers from NGI presents and evaluates an approach for predicting DTB by use of a multi-layered artificial neural network regressor [27]. This approach produced improved results in settings where the geological setting is varying and complex, compared to the LSI method, described in Section 3.1.2, where a linear function approximator algorithm is used.

The data from inversions of AEM survey data can be described as a matrix of N rows and M columns, where N denotes the number of measurements *(soundings)*, and M represents the layered resistivities at increasing depths, *(resistivity profiles)*. Each row can also be enriched to contain coordinates , elevation or any other relevant information.

The aim of the proposed ANN approach is to mitigate the problem that arise in use of the LSI approach, as it would use the same rule for every sounding, regardless of location. As subsurface topography is heavily dependent on the geological setting at the sounding's location, it can drastically affect actual DTB values.

3.2.2.1 Description of Method

Non-linearity is achieved in the ANN approach by the inclusion of a hidden layer in the network. The ANN is built on the Scikit-Learn framework for the programming language Python.

The training data for the model can be obtained from known DTB values from boreholes or expert manual interpretation of the resistivity profiles. Thus, training points do not necessarily always exactly match the actual DTB, but rather provide meaningful representations. The study presented in NGI's paper found that the required number and density of the training points needed to obtain good predictions were largely dependent on the complexity and the size of the surveyed area [27]. As AEM data resolution decreases with depth, larger uncertainty in predictions are expected at deeper layers.

The method also allows for a type of representation of uncertainty, where a second network is labeled with manually assigned uncertainty values measured in meters. This allows the first network to predict a depth, and the second network to predict an uncertainty range.

3.2.2.2 Result Comparison with Traditional Method

NGI compared the ANN approach with the somewhat simpler LSI method for DTB prediction in two different projects and geographic areas. The two locations differed in complexity and size, with one being smaller and of a simpler, and more consistent nature than the other. Both methods proved viable, increasing their accuracy with more training points and converging at approximately 3.5 meters of average mismatch. Most predictions did actually have smaller mismatches, but some data-points where bedrock could be as deep as 70 meters had large mismatches resulting in a higher average.

The more complex area required more training data for achieving an average mismatch of about 8 meters. The ANN approach provided improved results by approximately 30% compared to the non-linear algorithmic approach.

NGI's article concludes that further work should evaluate different regularization schemes and solvers as well as different AEM inversion schemes for improving the method. They also mention that smarter training sets and inclusion of geologic information in the ANN could improve the predictions.

3.3 Uncertainty in ANN's and Prediction Interval Construction

Recall from Section 2.7 that there is a desire from the industry to retain some sort of knowledge about the uncertainty for predictions in automated approaches. While NGI's ANN proposal can produce both DTB predictions and a form of uncertainty range, the uncertainty is still user-assigned [5]. Moreover, it is not entirely clear what the uncertainty range covers, whether it denotes a confidence interval or a prediction interval, and what level of confidence it holds. The uncertainty values are effectively prone to human bias, since they are user-assigned. In turn, this also makes the ANN predicting biased values, as it approximates a function that fits the training data. Thus, the current industry standard does not provide automated uncertainty values for the model's predictions. Rather it produces predictions of biased human assigned estimations. The fact that the problem is one of regression makes the problem more difficult. Section 2.7.2 provided a brief introduction to the aspect of prediction intervals. This section presents a set of techniques that have commonly been used for construction of such intervals. These techniques are important for the methods which will be compared and evaluated in the case study presented later in the thesis.

3.3.1 Mean Variance Estimation

Nix & Weigend proposed an approach which allows for the estimation of data noise variance, or aleatoric uncertainty, as a function of the input for the regression [30]. The approach relies on a single ANN with two output nodes, where the first node outputs the prediction for the regression, which is also the true mean of the target distribution, and the second output node predicts the variance of the same distribution. The method can largely be thought of as a combination of a standard regression network and a maximum likelihood estimator for the aleatoric variance of its prediction.

3.3.2 The Bootstrap Method

The bootstrap method was originally proposed by Heskes in 1996 [17]. The method presents an approach for obtaining model uncertainty, or epistemic uncertainty, by training an ensemble of many networks on a set of random sampling from the original training data. Each trained model will thus be slightly different as a result of random initialization of weights and biases, but also as a result of somewhat differing perspectives on the data. An estimation of the epistemic variance for any point in the regression can thus be found by the distribution of predictions for that regression point.

Heskes also proposes in his paper from 1996 [17] a method that incorporates some aspects of Nix & Weigend's MVE technique, allowing for a method that can combine estimations of both epistemic *and* aleatoric uncertainty. Effectively, this technique allows for estimations of the total uncertainty of a regressor's predictions.

Chapter 4

Research Design & Implementation

This chapter will provide the reader with a description of the approach of the conducted research for the thesis.

Section 4.1 will present the two fundamental motivations that are core to this thesis. The section concretizises the motivation in the form of a set of research questions, which accurately denote the exact information that the research aims to uncover.

Section 4.2 delves deeper into the data sets, and establishes a more thorough understanding of the data that was provided for the case study.

Section 4.3 proposes a new technique for automated DTB interpretation, inspired by novel technology from the academic field of computer vision. The section begins with a thorough explanation and description of the method. This includes a proposed solution to a new problem that arises with the automated technique, encompassing the obtaining of valid training data. The section concludes with a section on the approach used for the evaluation of the method.

Section 4.4 describes three different techniques for the construction of prediction intervals for regression type ANNs. The descriptions thoroughly inspect both technical and non-technical aspects of each of the techniques' functionalities and requirements. After the methods have been established, the section concludes with a description of the approach used for evaluating and comparing the techniques.

4.1 Research Motivation

The underlying motivation for the research is twofold.

Firstly, the research aims to understand how AEM data can be used in combination with machine learning techniques to produce accurate predictions of distance from soil to bedrock. The research initially aims to firmly establish both the practicality and added value of inclusion of ML techniques in DTB interpretation. Cost reduction and elimination of tedious manual labor in the interpretation process is a driving factor, where time spent by geotechnical experts is considered more valuable when used on other problem areas such as verification of results and overall interpretation.

The second motivator addresses the issue of confidence related to automated DTB predictions. The added value of simplistic point predictions, where no notion of confidence in the predictions are present, are considered little. Thus, to adhere to the industry's desire, the research aims to uncover a practical method for quantification of confidence for automated predictions.

4.1.1 Research Questions

This sections concretizises the motivation by establishing a set of research questions that clearly states the aims of the research.

- **RQ1** *What machine learning technique could increase performance for the problem of DTB predictions from AEM data, compared to NGI's current MLP regressor approach?*
- **RQ2** *What method of constructing prediction intervals for representing uncertainty in automated regression point predictions for DTB levels is most fit for the problem?*

Research Question 1 strongly correlate to the thesis' primary contribution as described in Section 1.2.1.1, and can be seen as the thesis' focal research question which provides the overarching direction of the research. It covers the considerable challenges in the unexplored territory of the intersection between AEM data and machine learning. Research Question 1 presents a problem where a number of ML techniques and models can be used for solving the problem. An evaluation two methods is covered in the experimental approach of the thesis.

Research Question 2 addresses the next step of quantifying and intuitively representing the confidence of the predictions as provided by some regression type predictor. While a resulting ML regressor from Research Question 1 could provide such predictions, the PI construction methods do not require the predictions to stem from a ML technique, and the two questions are therefore distinct. In the case study presented in this thesis, a proposed ML technique is, however, used. A selection of approaches are evaluated in the thesis' approach. The question concretizises the problem of representing prediction results in a manner that provides domain experts with knowledge that goes beyond the standard point predictions from regression models by including the metric of uncertainty.

4.2 Exploratory Data Analysis

An initial exploratory data analysis was done to gain an improved understanding of the data related to the case study where the data was provided by NGI. The exploratory analysis presented here relates to the same dataset as discussed in Section 2.5 and from NGIs paper [9].

A large set of 3D scatter plots such as presented in Figure 2.4 were used to gain a visual understanding of how DTB values tend to vary with distances. Figure 4.1 shows such a variogram with 50 lags. The variogram attempts to describe the degree of spatial dependence of the DTB values. In general, geological data tend to vary more at increasing distances, resulting in a logarithmic curve. However, analysis of the produced variograms reveal that this assumption only holds true for the surveyed area for a short initial distance. From the 3D visualization in Figure 2.4 this can be understood by noting the 'dip' at approximately 10000 meters easting and 5000 meters northing. The vast majority of the boreholes are located around this dip, resulting in large variances at close distances for these boreholes. Additionally, a note should be made of the sparsity in the coverage of the area, as there exist considerable gaps.

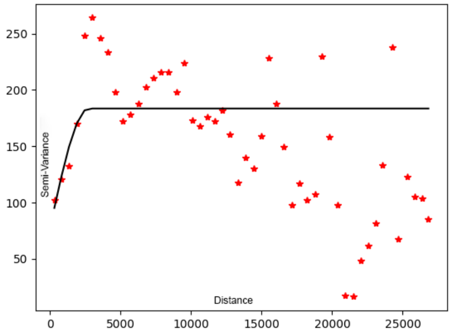
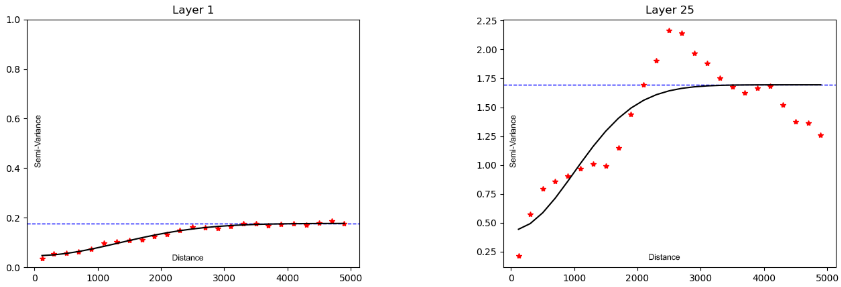


Figure 4.1: Variogram for borehole depths of the complete survey area

The variance of the resistivity profiles seem to increase at deeper levels. Another interesting observation is the hole-effect drop that appears at approximately 4000 distance meters [37]. This is caused by the dataset containing much data from two river valleys, which are quite similar, and spaced approximately 4000 meters apart, and cause the variance to decrease. Figure 4.2 shows the variograms for the uppermost and deepest inversion layers respectively.



(a) Variogram for the uppermost inversion layer **(b)** Variogram for the deepest inversion layer

Figure 4.2: Variograms for two layers of the sounding inversions.

4.3 Convolutional ANN Proposal

This section presents the proposal for a new approach for DTB prediction, and relates to the previously defined Research Question 1. Section 4.3.1 presents a description of the intuition behind the approach, before a detailed explanation of the neural network model is presented. Section 4.3.3 presents the method that was used for comparison and evaluation of the new approach and the already established MLP regressor approach from NGI [27].

4.3.1 Description of Approach

Section 2.3 described the intuition that a geotechnical expert makes use of during manual interpretation of resistivity profiles in a continuous space. From the description of the process, it can quickly be established that contextual patterns play an important role in the interpretation process, as such patterns may influence the interpreters predictions. Since the MLP proposal from NGI, which is described in Section 3.2.2, only accounts for a single resistivity profile during processing, no patterns crossing the horizontal spectrum can be claimed to be accounted for in its predictions.

To account for the contextual information that exists in the spatial continuum of the resistivity profiles, a Convolutional Neural Network, (CNN), is proposed. The idea behind the proposal is that allowing the neural network to view the input as a two dimensional image could allow it to detect patterns such as slants, peaks and valleys that may influence it's predictions, somewhat similarly as it would influence a human interpreter's predictions. More specifically, the pattern detection is done in a convolutional layer by the use of *filters*.

Let rp denote a resistivity profile consisting of some number of layers. A logarithmic transformer, L, is used to transform the resistivity values in each layer of rp into values that would correspond to the colors, similarly as done in Figure 2.2. The transformation by L is performed by taking the base 10 logarithm of the resistivity value, as the original values grow exponentially with denser materials.

Single noisy resistivity profiles would also pose far less disturbance to the predictions, as data from its neighboring resistivity profiles could alleviate the discrepancy in the input data.

Let M denote the convolutional neural network and X represent a complete dataset consisting of elements where each element is an array of 5 logarithmic transformed neighboring resistivity profiles, which represents an image. Let Y denote the depth labels for each image in X, such that Y_i is the depth label corresponding to X_i. A visualization of one such element with the depth label displayed as a horizontal black line is presented in Figure 4.3. Note that the goal is to identify the depth at the horizontal center of the image.

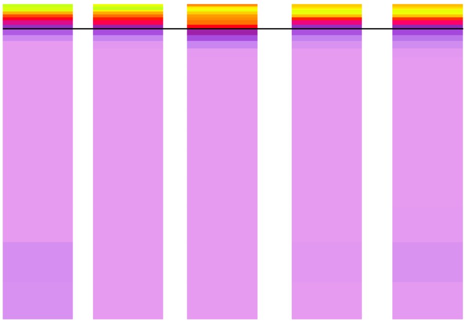


Figure 4.3: Visual representation of one element from X

The proposed topology of M consists of an input layer, a single convolutional layer, one flattening layer, and a densely connected output layer resulting in a single neuron.

Figure 4.4 presents a schematic visualization of the CNN's proposed topology. Each of the boxes after the input box represent a single layer in the network. The boxes on the left side denote identifiers and names for the layers. The boxes on the far right denote the shape of the numerical data in dimensions as it flows through the network. The first dimension represents the number of data points that will flow through the layer. The fact that the these dimensions are set to *None* represents that there is no limitation on the number of data points that is used during training. In the input of the convolution layer, the second dimension, set to 5, corresponds to the number of resistivity profile arrays that a single data point consists of. The third dimension represents the number of resistivity values in each of the 5 resistivity profiles. The last dimension, set to 1, simply states that each resistivity value is a single number. The output of the convolution is a set of 20 feature maps, as the result of convolution using 20 different kernels. The input and the output of the flattening layer show how it reduces the dimensions to a single *flattened* dimension. The input and the output of the densely connected layer show how the resulting 2500 neurons are reduced to a single final output neuron.

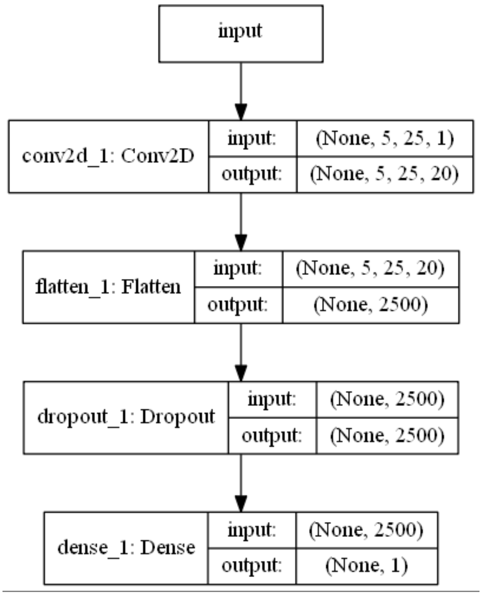


Figure 4.4: Schematic representation of the CNN model's topology

Figure 4.5 presents a simplified graphic overview of the networks topology, and the operations used within.

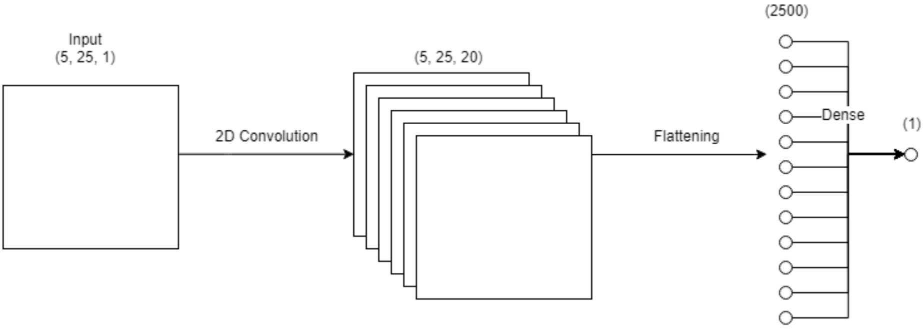


Figure 4.5: Graphic representation of shape changing operations in the CNN

The input layer takes a set of 5 neighboring resistivity profiles as input, which can be thought of as an image, where each layer in each resistivity profile corresponds to a single pixel that has some resistivity value. Another way to think about this input dimension is simply as a matrix consisting of 25 rows and 5 columns. The first layer performs convolution over some number of filters, f. Each filter is a small matrix of width f_x, and height f_y.

The filter size is also commonly referred to as the *kernel size*. Thus, a filter with $f_x = 3$ and $f_y = 3$ would have a total of $f_x \cdot f_y = 9$ values. Initially, these values are randomly set, but during training these converge at common patterns.

In the convolutional layer, for each filter f_i, the filter is matched over the entire input image in small sections, by sliding over the image, so that every block of pixels of the same shape as the filter has been matched. The process of sliding the filter over the images is commonly referred to as *convolving*. For each convolved block, the dot product of the filter and the block from the input is retained as output. Thus, for each filter, $f_i \in f$, a new matrix is produced, consisting of a new representation of the input image as convolved by the filter. The process is repeated for every filter. Padding is used to maintain the original size of the matrix, such as discussed in Section 3.2.1.1. Hence, the final output of the convolutional layer is a set of n new matrices, where n is the number of filters in f.

The flattening layer flattens the n matrices to a one dimensional array of neurons. A dropout layer is placed between the densely connected layer and the single output neuron. During training, the dropout layer randomly drops 10% of the neurons. This is done in an attempt to avoid overfitting the network, due to the large amount of neurons in the last layer. The value of 10% was set as it produced the best results after observing several tests with different values. This technique was proposed by Srivasta *et. al* in 2014 [39], and has proved to be successful in various use cases since [18][19].

Both the convolutional layer and the densely connected final layer both use the ReLU activation function, carrying the benefits of being simplistic and non-linear, while at the same time enforcing non-negative output.

Training of parameters in the network follows the same back-propagation procedure as used in NGI's proposed MLP regressor.

4.3.2 Spatially Distributed Kriging

The method of obtaining training data for the CNN model is based on the similar method used for obtaining training data for the standard MLP approach. The two methods differ, however, in that the standard approach only requires the approximation and interpolation of a single resistivity profile for each borehole, whereas the CNN approach requires an array of neighboring resistivity profiles. This additional demand for neighboring resistivity profiles poses further requirements for the interpolation process. The same method of interpolation is used, such as described in Section 2.6.1, with some additional steps.

Similar for both is that the interpolation approach uses the base 10 logarithmic values of the original resistivity values in the layers of each resistivity profile. Essentially, this means that the resulting interpolations will be the base 10 logarithm of the original values, and it is the reason for why the ANN models use base 10 logarithmic values as their input data.

The method of layered 2D Kriging was selected for interpolation. Thus, interpolating a single resistivity profile for a single point requires 25 individual Kriging models, one for each layer in the resistivity profile. 25 distinct semi-variograms were initially produced from all resistivity profiles in the dataset, each using the resistivity values located at the same depth. A visualization of the semi-variograms can be found in Appendix D, Figure 7.17. The gaussian function was selected as the model type, as it provided the best generalization of the data points after a manual inspection. The model parameters, the sill, range, and nugget were automatically fitted using a soft L1 minimization scheme as implemented in the open source Python library *PyKrige* [28]. This essentially fits the selected model such that the MAE loss is attempted minimized for the known points.

Figure 4.6 displays the kriged interpolations of two resistivity profiles, *colored in blue*, located evenly spaced between two known resistivity profiles, *colored in red*. The Y-axis represents the depth below the surface. The X-axis represents the base 10 logarithm of the resistivity value at the depth provided by the Y-axis.

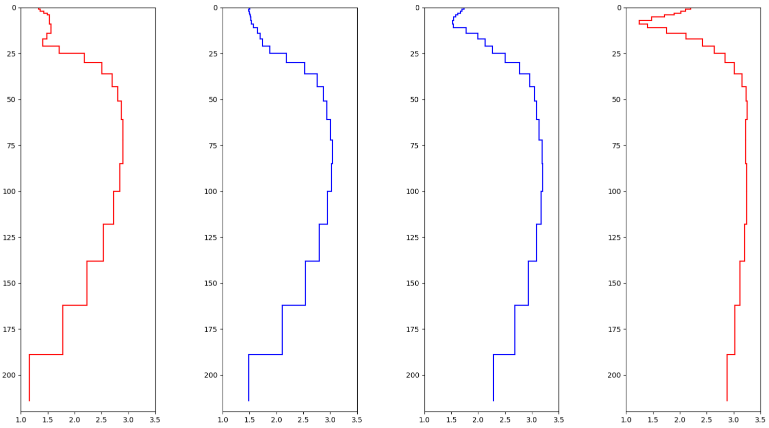


Figure 4.6: Visualization of interpolations between two known points

The layered Kriging approach allows for approximation of a single resistivity profile $\hat{A}_{x,y}$ by individually computing $\hat{a}^l_{x,y} \in \hat{A}_{x,y}$ for each layer in $\hat{A}_{x,y}$, where x and y represent easting and northing on a map. Let $b_{x,y}$ represent a borehole at a given location, and $K(i, j, m_l)$ represent the method of Kriging a single layer for a resistivity profile at a specific location, where i represents the easting, j represents the northing, and m_l is the Kriging model that has been produced for the layer l.

Obtaining the interpolation of a full resistivity profile $\hat{A}_{i,j}$, assuming the 25 Kriging models already exist, corresponds to combining the results of $K(i, j, m_l)$ for each $m_l \in M$ where M is the collection of Kriging models, organized by layers.

Thus, the resulting $\hat{A}_{i,j}$ for $i = x$ and $j = y$ provides the approximation of a single resistivity profile on the exact same location as the borehole. The standard MLP approach requires only a single resistivity profile for each borehole value. The depth value from the borehole can then be related to the interpolated resistivity profile and used as training data.

However, as the CNN model is reliant on an array of n consecutive resistivity profiles, an additional set of resistivity profiles related to each borehole need to be approximated to produce valid input training data for the borehole depths. The following exemplification assumes that $n = 5$, such as represented in the model in Section 4.3.

Figure 4.7 shows how an additional 16 resistivity profiles, *(red and blue dots)*, could be located around a borehole at the green dot. Four straight lines can then be drawn to cover n resistivity profiles such as to produce a valid array which can be used as training data for the CNN model. An important note is that the order of the profiles must be maintained within each array for the input to be valid. Essentially, this means that for any data point that can be extracted, the resistivity profiles in the resulting array must be indexed corresponding to their order of appearance on the drawn line. Consequently, each of the black lines can produce two distinct arrays of input data, one for each direction. This means that a total of $4 \cdot 2 = 8$ input data points can be produced for each borehole. Table 4.1 presents an exemplification of the two valid data points that may be extracted from the horizontal line in Figure 4.7, maintaining the correct order.

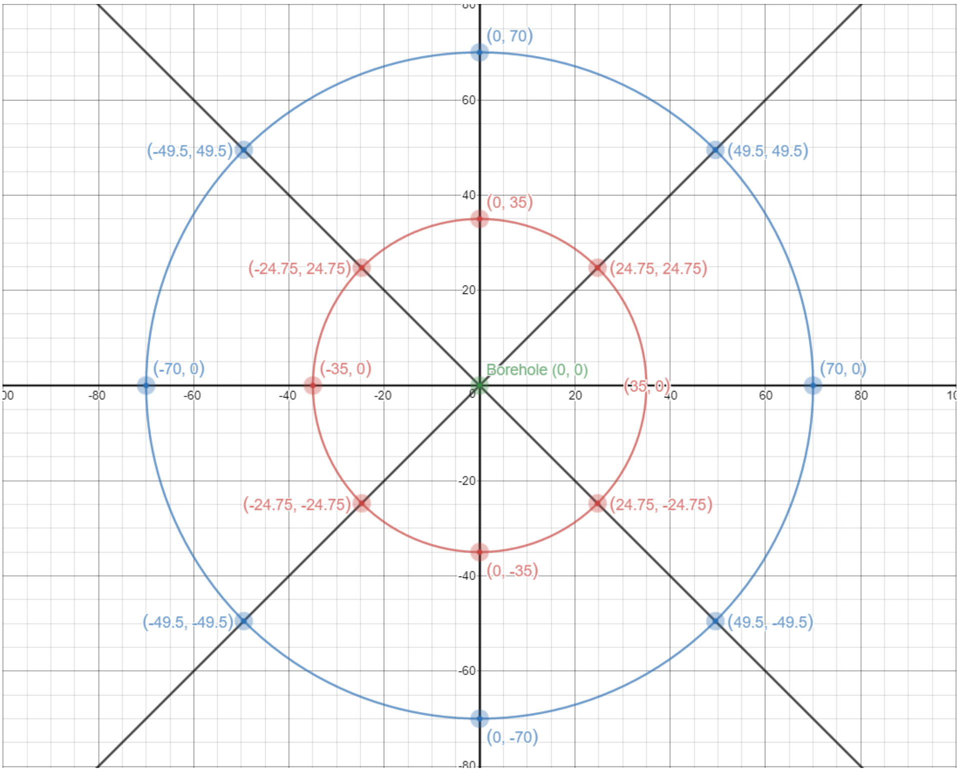


Figure 4.7: Visualization of resistivity profile approximations by location

	Index 0	Index 1	Index 2	Index 3	Index 4
Array Left to Right	(-70, 0)	(-35, 0)	(0, 0)	(35, 0)	(70, 0)
Array Right to Left	(70, 0)	(35, 0)	(0, 0)	(-35, 0)	(-70, 0)

Table 4.1: Example of valid indexing of resistivity profiles which maintains the order of appearance for the profiles in the horizontal array from Figure 4.7

It may seem like a drastic extrapolation for a single known value to produce 8 input data points, but the resulting arrays of resistivity profiles may be vastly different. If one of the arrays represents a slant in one direction, including a reversed copy of it allows the CNN model to see the slant from both directions which again allows it to gain more insight. This can be explained in that the depth would not differ, no matter what direction the airborne vessel conducting the soundings came from.

Table 4.2 presents a similar, top-down view of the grid, where the borehole is located in the center cell. Each cell contains a tuple representing the distance offset from the borehole in meters, in the form of easting and northing respectively. The distance of 35 meters between each point was selected as it is the average distance between consecutive sounding measurements as conducted by the airborne vessel for the provided dataset. The black lines from Figure 4.7 are represented by colored table cells. The horizontal and vertical points can be obtained by simple shifting of the boreholes easting and northing respectively by 35 meters. The points for the diagonal lines are obtained by shifting of both the easting and the northing by the distance as obtained by the Pythagorean Theorem such as shown in Equation 4.1, where d represents the desired distance between the points and $\Delta_{x,y}$ represents the offset in the easting and northing. $d = 35$ then yields a diagonal offset of $\approx \pm 24.75$ meters.

$$d = \pm\sqrt{2 \cdot \Delta_{x,y}^2} \tag{4.1}$$

Knowledge of the spatial difference allows for Kriging of the required surrounding locations, as the actual spatial position can be obtained by adding the easting and northing differences to the respective values for the borehole.

(-49.5, 49.5)		(0, 70)		(49.5, 49.5)
	(-24.75, 24.75)	(0, 35)	(24.75, 24.75)	
(-70, 0)	(-35, 0)	(0, 0)	(35, 0)	(70, 0)
	(-24.75, -24.75)	(0, -35)	(24.75, -24.75)	
(-49.5, -49.5)		(0, -70)		(49.5, -49.5)

Table 4.2: Offset in meters, (easting, northing), for obtaining surrounding locations of a borehole

4.3.3 Evaluation & Comparison

The following section covers the implementation and design of the approach used for testing, evaluating, and comparing NGI's MLP regressor and the proposed CNN approach. The approach for the evaluation is separated into two distinct conceptual techniques of analysis.

The first concept is a purely numerical measurement of the model's performance as measured by a set of performance metrics. This technique is explained and concretized for this case in Section 4.3.3.1.

The second evaluation concept is a visual inspection and evaluation where both interpretation methods performs predictions across a spatially continuous flight-line, as obtained from the real case study. The visual evaluation concept is further explained and concretized for this case study in Section 4.3.3.2.

4.3.3.1 Numerical Metrics Analysis

The numerical metrics analysis is based on a set of well established metrics for performance measurement of regression models. The time consumed for training was also tracked and stored, in addition to the metrics defined and described in the following paragraphs.

In supervised learning ANN regression models are trained to fit labeled data as accurately as possible while attempting to generalize to unknown data and also to avoid overfitting. The models are continuously evaluated, and their parameters are modified in attempts to improve the ongoing evaluations during training. Similar evaluations can also be performed after the training has completed, to evaluate how well the model is performing. Below follows an explanation of the error metrics that were considered in the final evaluation that took place after the training phase was completed.

4.3.3.1.1 Mean Absolute Error

The AE (Absolute Error) is defined as the absolute value of the predicted value subtracted by the labeled value. The MAE (Mean Absolute Error) is obtained by taking the average over the AE for all predictions as presented in Equation 4.2, where n represents the number of predictions, y_i represents the true value at i, and $\hat{y}_i$ represents the predicted value at i. During training n corresponds to the number of data points that were considered in a single training batch, while during evaluation of a model, n typically corresponds to the number of data points in the testing set.

$$MAE = (\frac{1}{n})\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

The mean absolute error is thus the mean of the sum of the absolute differences between predictions and known values. Hence, it provides the interpreter with an idea of how wrong the predictions are on average. Higher MAE values correspond to increasingly erroneous predictions, while a score of 0 means totally accurate predictions. The scores are absolute values, and thus this metric cannot yield the interpreter information about the direction of the error, e.g. in the case of DTB levels, if the predictions were too shallow or too deep. However, since the output is presented in meters below the surface, the MAE metric provides a sensible and intuitive measure in terms of meters as to the average error of the predictions.

4.3.3.1.2 Mean Squared Error & Root Mean Squared Error

The model's robustness can also be measured by the MSE (Mean Squared Error) for all its predictions. The SE (Squared Error) for a single measurement is obtained by subtracting the actual value from the predicted value and then squaring it. Thus, a squared error of 0 corresponds to perfect predictions, and higher values correspond to less ideal accuracy.

To obtain the Mean SE over multiple predictions the average over each prediction's SE is computed. The formula for obtaining the MSE is presented in Equation 4.3, where n represents the number of predictions, y_i represents the true value at i, and $\hat{y}_i$ represents the predicted value at i. Similarly as to MAE does n correspond to the number of data points considered in the respective training batch, or the number of data points in the testing set when used for a final evaluation of performance.

$$MSE = (\frac{1}{n})\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.3}$$

The Root Mean Squared Error (RMSE) provides a more sensible error measurement than MSE in the sense that it returns the error to the actual unit size, (meters in the case of DTB interpretation), by returning the square root of the MSE. The fact that the residuals are squared makes large residuals affect the RMSE to a vastly greater extent than smaller residuals.

$$RMSE = \sqrt{MSE} \tag{4.4}$$

4.3.3.1.3 Consumed Time During Training

The network topology in ANNs greatly affects the required time for a model to converge at an optimal approximation. A reasonable argument can be made that any ANN model is of little use if its resource demand exceeds that which is available to the user. The time consumed by each model for reaching convergence is therefore also evaluated and considered in this approach.

4.3.3.1.4 Approach

Two separate datasets were considered during the comparison, divided into 5 distinct cases. The first four cases uses data derived from the highway construction project dataset described in Section 2.5. The fifth case uses data from a more recent highway and railroad project (Ringeriksbanen & E16)[1]. This second dataset presented a significantly more complex subsurface topography.

[1] See project description at: https://www.banenor.no/Prosjekter/prosjekter/ringeriksbanenoge16/

The first case uses data points where depth levels have been manually assigned to a subset of the resistivity profiles from a small survey area. In this survey area the subsurface topography can be considered rather simplistic. The survey area for the first case spans a flight range covering approximately 1900 meters. The second dataset also uses manually selected depth labels, where the data points are obtained from a larger survey area with a vastly more complex subsurface topography. The distance of the flight range for the second case spans more than 16 kilometers, with a gap covering approximately 6 kilometers where no data exist. The third case uses data points where the depth labels are directly retrieved from the TS boreholes. The resistivity profiles for this case are obtained using the Kriging method as described in Section 4.3.2. The fourth case is similar to the third, but also includes the RPS boreholes. The fifth case applies the same test to a different dataset containing survey data from another geographical region in Norway. A total of 431 TS boreholes was included in the second dataset.

Where Case 3, Case 4 and Case 5 uses data points obtained from the Kriging technique, an additional step was included to filter away any boreholes that had no AEM soundings within a 30 meter distance. This was done as interpolating data points with no reasonably close soundings was deemed to not yield the training process any significant benefit.

Figure 4.8 presents the approach used to evaluate the metrics in the comparison.

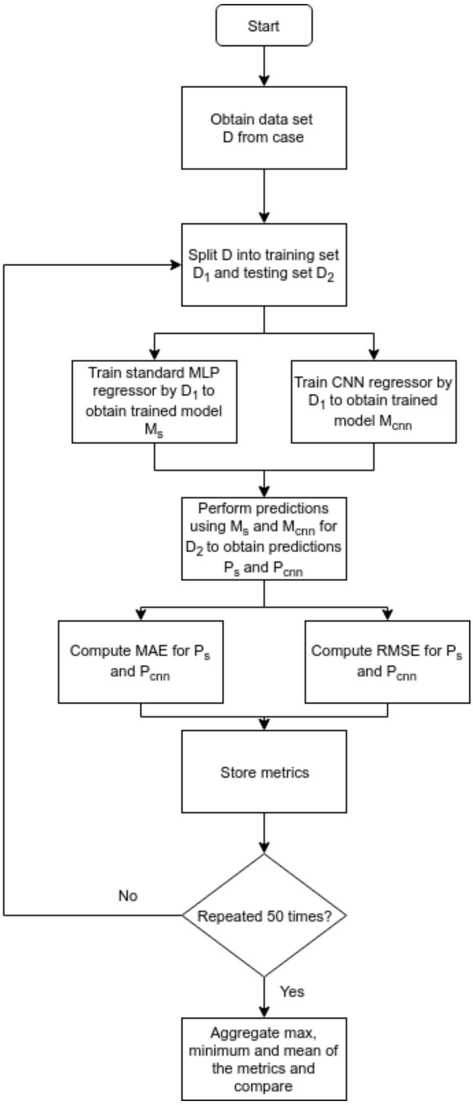

Start
Obtain data set D from case
Split D into training set D_1 and testing set D_2
Train standard MLP regressor by D_1 to obtain trained model M_s
Train CNN regressor by D_1 to obtain trained model M_{cnn}
Perform predictions using M_s and M_{cnn} for D_2 to obtain predictions P_s and P_{cnn}
Compute MAE for P_s and P_{cnn}
Compute RMSE for P_s and P_{cnn}
Store metrics
Repeated 50 times?
No
Yes
Aggregate max, minimum and mean of the metrics and compare

Figure 4.8: Visualization of the approach for metric comparison

The metrics were computed using cross-validation to most accurately assess how the results would generalize to an independent dataset. Cross-validation is a very common technique used when evaluating and validating how well a model would generalize to an independent dataset.

K-fold Cross validation consists of performing some number, r, of distinct validations on r trained instantiations of the same model, M, where each instantiation of M have a somewhat differing perspective on the data. Each round (or fold) of cross validation uses the complete labeled dataset, D, from the respective case of validation. In each round the complete dataset, D is randomly split into two smaller datasets, D_{train} and D_{test}. D_{train} corresponds to the portion of the data that is used to train each model. D_{test} corresponds to the validation set. The datasets are randomly split such that 70% of the complete dataset will reside in D_{train} and the remaining 30% will reside in D_{test}. The models will thus not have seen any of the the data points in the validation set during training. In each round, after both models are trained, the MAE and RMSE metrics is computed from the predictions of the data points in D_{test}. After each round two new datasets are derived from D and the process is started anew. In this case study, a total of $r = 50$ rounds of cross-validation are performed for each case.

The RMSE and MAE metrics are stored for each round, and used to derive three metrics each after the completion of all 50 rounds. The three metrics that are finally presented are the mean, max, and minimum of the metrics that are produced during the 50 rounds.

The consumed computation time is also stored for each round of training for each model. The averaged training time can then be presented as a separate metric in the comparison. The computation time is recorded in seconds as performed on a single Intel Core i7-2600 CPU running at 3.40GHz.

Both models use the MSE metric as cost function for optimization during training. However, the training phase for each of the methods differ in the selection of the optimization algorithm. The CNN method used the stochastic Adam optimizer [22], while the standard MLP approach used L-BFGS [31, Sec. 7.2]. The L-BFGS optimizer was selected for the standard MLP approach to most accurately provide a similar model to NGI's proposal [27], where the same optimizer was used. The standard non-linear MLP regressor is also implemented using the SciKit learn framework [33], whereas the CNN method is implemented using Keras [8]. In short, the standard MLP consists of an input layer that takes a single transformed resistivity profile as input followed by a single hidden layer consisting of 20 neurons. The hidden layer uses the non-linear ReLU activation function. The output layer consists of a single neuron that outputs the depth prediction.

4.3.3.2 Visual Analysis

Alongside the metrical comparison described in the previous section, a visual analysis was conducted in collaboration with geotechnical experts for each of the cases to provide a qualitative performance measure.

The visual analysis was deemed necessary to provide an additional layer of understanding of how well the models were able to adapt and generalize to unknown data points. The manual approach of depth interpretation has previously relied on geotechnical experts to visually analyze a set of spatially continuous resistivity profiles as obtained from a flight line. Thus, geotechnical experts may provide an indication on how well the depth predictions would correspond to their own judgment and domain expertise.

4.3.3.2.1 Approach

The visual analysis uses the same cases as described in the approach for the numerical metrics in Section 4.3.3.1.4. For Case 1 and Case 2 the produced visualizations correspond to the same flight lines from which the training points were collected. This allows the interpreter to also see which data points were used during training of the model.

For Case 3, 4 and 5, the visualization process is somewhat different, as the data points in D_{train} are selected from the entire dataset containing all flight lines for the survey area. Thus, random fight lines were selected for the model to perform depth predictions.

4.4 Prediction Intervals Construction & Evaluation

This section aims to discuss and account for a set of procedures for computing and evaluating the degree of successfulness of prediction intervals in the use of DTB interpretation, and thus relates to Research Question 2. The evaluation of the PIs is a necessary step, as it will inform us on how well any individual approach is able to solve the problem.

The initial sections provide an overview of the approaches that are evaluated in the experimental approach. Each of the approaches described in the following sections assumes that there exist some model, M, that can produce predictions for some input data. Each of the methods also assumes that there exist a labeled dataset, D, which can be used to train

M. For the following sections, let X represent the input data in D, and Y represent the labels for X such that any label $y_i \in Y$ provides the corresponding label for $x_i \in X$.

4.4.1 Ensembled PI Construction Assuming Fixed Aleatoric Uncertainty

The first method of PI construction can be regarded as the most simplistic method evaluated in this thesis. Two core assumptions lay grounds for the approach. Firstly, the residuals, ϵ, are assumed to be normally distributed around the true values in Y, and are independently and identically distributed. This means that for any prediction, $\hat{y}_i$, with a related total uncertainty variance, σ_i^2, a boundary for a prediction interval covering some percentile of the normal distribution can be obtained from $\pm k \cdot \sqrt{\sigma_i^2}$, where k is a scalar value based on the desired coverage. Thus, there is assumed to be an equal probability of the deviation occurring in both directions. This can be visually interpreted by Figure 4.10. Thus, if an approximation exist for the uncertainty variance at a single point, σ_i^2, along with a point prediction for the same point, $\hat{y}_i$, a prediction interval can be inferred for any percentile by inferring two new points on each side of $\hat{y}_i$ respective to σ_i^2. Recalling that $\sigma_i^2 = \sigma_{\hat{y}_i}^2 + \sigma_{\epsilon_i}^2$, the problem of finding an approximation for σ_i^2 can be simplified into finding best estimates for $\sigma_{\hat{y}_i}^2$ and $\sigma_{\epsilon_i}^2$.

According to the second core assumption, the aleatoric uncertainty is assumed to be fixed and constant.

Initially, before M is trained, D is randomly split into two new datasets, D^{train} and D^{test}, N times. In the research implementation for this case study the value of $N = 20$ was used. Generally speaking, higher values for N would yield more reliable results. The selected value was, however selected due to it providing a good compromise between the required computation time to train N separate models and the data point coverage. This process results in N sets for D^{train} and N sets for D^{test}, such that the union of the two relating sets is the complete set D, such as shown in Equation 4.5, and the intersection of the two relating sets is an empty set, as shown in Equation 4.6.

$$D_i^{train} \cup D_i^{test} = D \tag{4.5}$$

$$D_i^{train} \cap D_i^{test} = \emptyset \tag{4.6}$$

The ensembled approach of repeating this process N times is done as a measure to account for the aleatoric uncertainty, $\sigma^2_{\hat{\epsilon}_i}$, to a greater extent, as this will be fixed for any future predictions where the label is unknown. The N random splits also allow each of the models to have a slightly different perspective on the data.

Furthermore, it allows for the obtaining of an approximation for the epistemic uncertainty, $\sigma^2_{\hat{y}_i}$, as proposed by Tom Heskes [17, p. 178]. By proxy this also means that the aleatoric uncertainty can be approximated. Assume that an approximation for $\sigma^2_{\hat{y}_i}$ exists, as well as a residual from a prediction made on a data point that was not seen before the obtaining of the approximation of $\sigma^2_{\hat{y}_i}$. It then follows from the aleatoric and epistemic distinctness that the remainder of the squared residual when $\sigma^2_{\hat{y}_i}$ is subtracted can only be explainable by $\sigma^2_{\hat{\epsilon}_i}$.

The optimal size of the split is not straight forwardly decided. A larger training set allows the models to more accurately fit the known data. A larger testing set, on the other hand, allows for a more accurate prediction of the aleatoric variance, $\sigma^2_{\epsilon_i}$, which will remained fixed after the testing phase. The best split generally depends on the original size of the labeled dataset. The case study data presented by NGI contained a relatively small amount of labeled data. In such cases, where the size of the known data can be considered small, a larger portion of the data should be assigned to training of the model, as the accuracy of the PI is of little usefulness if the model's predictions are poor. For the research implementation of this case study the split was set such that D_{train} contains 60% of the available data points, and D_{test} contains 40% of the available data points.

A two phased process is then carried out. In the first phase, N models are trained on each of the randomly sampled training sets. After each model has gone through the first phase of training, the second phase is carried out.

The second phase consists of each model making a prediction for each data point $dp_i \in D$, if the model in question did not see dp_i during its training. For each dp_i an estimation of the epistemic variance at i is found by taking the variance of all predictions made for dp_i. The point prediction is found by taking the mean of all predictions. The remainder of the squared residual, r_i^2, when the epistemic uncertainty is subtracted, is then not explainable by the epistemic uncertainty. Since the true label t_i is known, this remainder can found by Equation 4.7. A second note to make regarding the equation is that negative remainders are set to 0, as negative values should not further affect the estimation for $\sigma^2_{\epsilon_i}$ to less than 0, as it cannot be negative. A value of 0 is, however, valid, as a perfect dataset with no noise holds no aleatoric uncertainty.

$$r_i^2 = max[(t_i - m(dp_i))^2 - \sigma^2_{\hat{y}_i}, 0] \tag{4.7}$$

Obtaining r_i^2 is, however, not a straight-forward task for any dp_i where the true label, t_i, is unknown, which is what happens whenever the network makes an actual prediction after the training phase is completed. However, since the approach carries the assumption of a fixed aleatoric variance, σ^2_ϵ, an approximation of this fixed uncertainty can be computed and stored directly after the training phase. After a prediction has been made for each $dp_i \in D$, the mean of all the remaining squared residuals are stored as an approximation of $\sigma^2_{\epsilon_i}$ for any new input where the true value is unknown. Notably, this is the crux of the method, and what the next method attempts to improve.

To summarize, an estimation of the epistemic uncertainty is found by the variance of N predictions by the ensembled approach. Since the total uncertainty can be written as the sum of epistemic and aleatoric uncertainty, an approximation of the aleatoric uncertainty can be found by the discrepancy of the squared total residual and the epistemic variance.

Figure 4.9 presents a visualization of the overarching approach for setting up the method for computing the point prediction and the width of PIs.

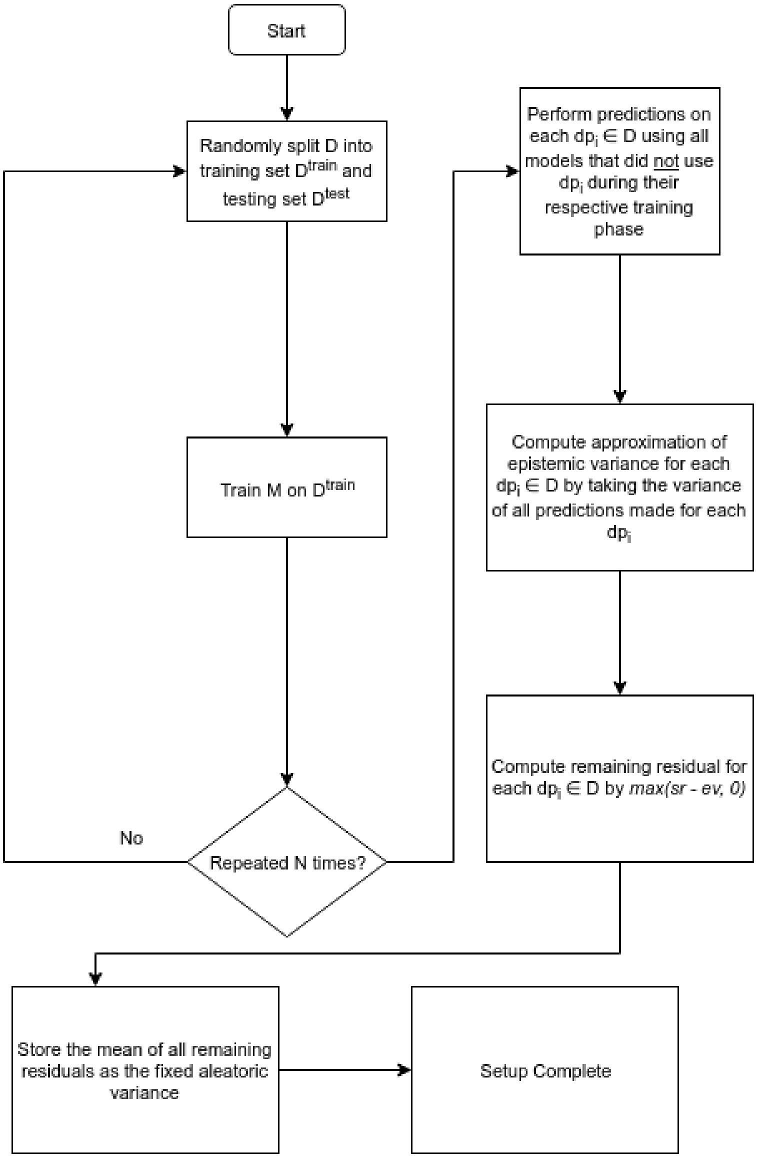

Start
Randomly split D into training set D^train and testing set D^test
Train M on D^train
Repeated N times?
No
Perform predictions on each dp_i ∈ D using all models that did not use dp_i during their respective training phase
Compute approximation of epistemic variance for each dp_i ∈ D by taking the variance of all predictions made for each dp_i
Compute remaining residual for each dp_i ∈ D by max(sr - ev, 0)
Store the mean of all remaining residuals as the fixed aleatoric variance
Setup Complete

Figure 4.9: Overarching approach for setting up the method where *sr* denotes the squared residual and *ev* denotes the epistemic variance

After obtaining the estimation of the aleatoric variance, the ensembled method is ready to make predictions for unknown data points. Recall that the total variance, σ^2 is required for the construction of prediction intervals when the target distribution is assumed gaussian, and that the total variance can be written as the sum of $\sigma^2_{\hat{y}_i}$ and $\sigma^2_{\hat{\epsilon}_i}$.

For any new input where the goal is to construct a prediction interval, a total of N predictions can be made, one for each of the trained models, such that each prediction from a single model, $\hat{y}_i \in \hat{Y}_i$, and $|\hat{Y}_i| = N$. The mean value of the N predictions is considered to be the single point prediction, and the variance of the N predictions $\hat{Y}_i$ corresponds to $\sigma^2_{\hat{y}_i}$.

Thus, the total variance, σ^2, can be obtained by adding the fixed aleatoric variance $\sigma^2_{\hat{\epsilon}}$. Using σ^2 to compute the distance from the single point prediction to the the upper and lower PI boundary is a relatively straightforward task, and it corresponds to solving the inverse of the cumulative distribution function. The Python library SciPy provides a simplistic function for this exact problem, the percent point function. In simpler terms this can be described as obtaining the point on the x-axis of the normal distribution that covers *some* percentile of the same distribution from the mean[2] [20].

Figure 4.10 presents a normal distribution curve that illustrates the correlation between the coverage of the distribution and σ.

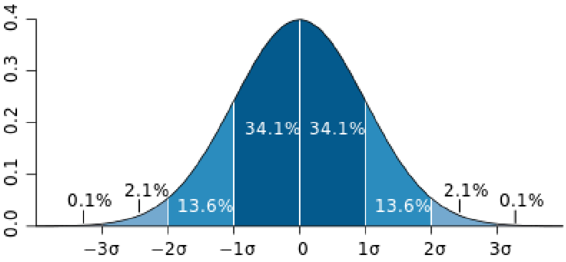


Figure 4.10: Standard deviation diagram[40]

Let α denote some probability, represented as a number such that $0 \leq \alpha \leq 1$. The percent point function for $\alpha = 0.5$ would then be 0, as it would correspond to the mean of the normal distribution. Thus, Equation 4.8 and Equation 4.9 show the functions for obtaining the upper and lower scalar values, s^α, where $pp()$ corresponds to the percent point function, such that the covered area is centered around the mean.

$$s^{\alpha}_{upper} = pp(.5 + (\frac{\alpha}{2})) \tag{4.8}$$

[2] See Percent point function; https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.norm.html

$$s^{\alpha}_{lower} = pp(.5 - (\frac{\alpha}{2})) \tag{4.9}$$

The scalar values must then be multiplied with the standard deviation of the distribution for obtaining the actual distances from the distribution's mean to the upper and lower boundaries for the prediction interval such that α of the distribution is covered. In essence, it is approximating the distance from the distribution's mean to the boundaries for the approximated uncertainty, bearing in mind that the mean of the distribution corresponds to the point prediction.

Since the point prediction corresponds to the mean of the normal distribution, it suffices to compute only one of the scalar values, as the distances on each side of the mean are equal, albeit of the opposite direction. Let s^{α} denote the absolute value of the scalar value. The values for the boundaries can then be computed by Equation 4.10 and Equation 4.11, where $\hat{y}_i^{\alpha_{upper}}$ and $\hat{y}_i^{\alpha_{lower}}$ correspond to the upper and lower boundaries at i respectively. When dealing with the case study of DTB interpretation, this wording might seem a bit curious, with the *upper* boundary being of less numerical value than the *lower* boundary. This is simply due to the nature of the case study. Recall that the boundaries represent distances below the surface level, and thus smaller values denote boundaries closer to the surface, thus *upper* boundaries, while higher distance values represent increasingly deeper distances, and thus *lower* boundaries.

$$\hat{y}_i^{\alpha_{upper}} = \hat{y} - (s^{\alpha} \cdot \sqrt{\sigma^2}) \tag{4.10}$$

$$\hat{y}_i^{\alpha_{lower}} = \hat{y} + (s^{\alpha} \cdot \sqrt{\sigma^2}) \tag{4.11}$$

Figure 4.11 presents a visualization of the complete approach for computing the predictions and the width of the PI for any data point, dp_i, where the true value is unknown.

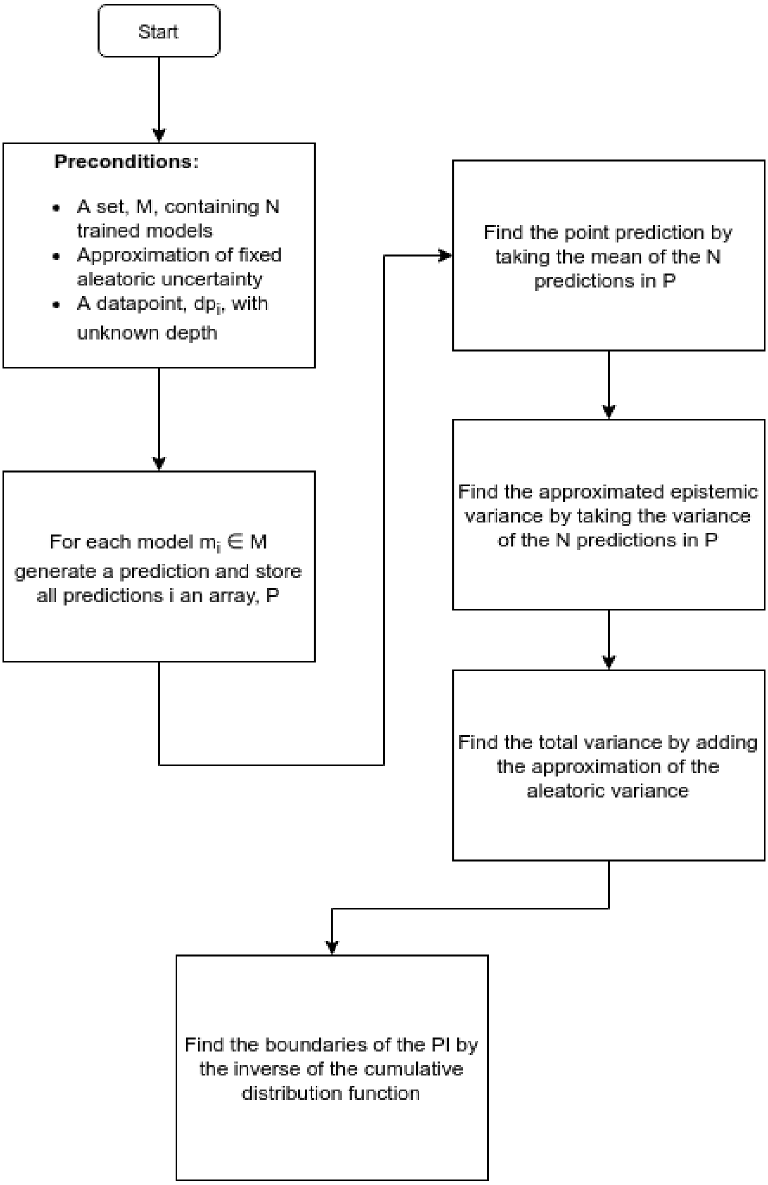

Start
Preconditions:
A set, M, containing N trained models
Approximation of fixed aleatoric uncertainty
A datapoint, dp_i, with unknown depth
For each model $m_i \in M$ generate a prediction and store all predictions i an array, P
Find the point prediction by taking the mean of the N predictions in P
Find the approximated epistemic variance by taking the variance of the N predictions in P
Find the total variance by adding the approximation of the aleatoric variance
Find the boundaries of the PI by the inverse of the cumulative distribution function

Figure 4.11: Overarching approach for computing point predictions and PI

The assumption of normally distributed errors lays ground for the simplicity of the method, however, it does also present significant drawbacks. In practice this means that the upper and lower boundaries of the constructed PI will be of equal distance from the actual prediction. Thus, if during testing the model tends to predict deeper DTB levels than the labeled depth, the PI will not only grow in the deeper direction, but also in the opposite direction.

The assumption of the aleatoric uncertainty being constant, independent of the input data, allows for a simpler implementation. However, it is perhaps also the most significant drawback. Consider a model being able to accurately predict DTB values for a large portion of the resistivity profiles in its testing set. If the same model produce large residuals on only a small subset of the testing data σ_{ϵ}^2 grows, and results in a drastic increase of the width for all future intervals. This loss of precision is due to the PIs boundaries being affected by the large residuals of the noisy profiles, even when simple and clean input data is being processed, as the approach assumes a fixed aleatoric variance.

Much valuable information is also unarguably lost if the majority of the uncertainty stems from the aleatoric portion.

4.4.2 Ensembled PI Construction with Variable Aleatoric Uncertainty

The second approach for PI construction is similar to the first, but differs in the way that it does not assume that the aleatoric variance, $\sigma_{\epsilon_i}^2$, is independent of the input data, x_i. Rather, it assumes that there exist a relation such that some function, $\sigma_{\epsilon_i}^2(x_i)$, yields the aleatoric variance from the input data. The assumption of the errors being independent and identically normally distributed, however, still holds true for this approach.

The method of obtaining the epistemic uncertainty is exactly the same as in the previous approach, randomly sampling N training sets and N testing sets from the original dataset, D. Thus, the first phase of training is similar, where N distinct models are trained. For the research implementation in this case study, the values for N and the size of the dataset split was set similarly as to the previous approach, such that D_{train} contains 60% of the available data points, and D_{test} contains 40% of the available data points, and $N = 20$.

Tom Heskes proposed a method in his paper on *Practical confidence and prediction intervals* [17], where the assumed function $\sigma_{\epsilon_i}^2(x_i)$ for the aleatoric variance could be estimated by the inclusion of an additional ANN. Heskes does not impose any rules or limitations for the topology of the ANN, other than requiring an activation function that imposes a positive value for the final regression neuron. In the approach presented here, the topology is set to be exactly the same as the topology for the networks that are used for obtaining the point predictions, with an exponential final activation function.

For any data point, dp_i, a total of N predictions can be made, allowing for the obtaining of a single point prediction, the mean, $m(dp_i) = \frac{1}{N}\sum_{n=1}^{N}\hat{y}_i^n$, where n is the n'th model in the set of trained models. The estimate of the epistemic variance, $\sigma^2_{\hat{y}_i}$ is still obtained from the variance of the N predictions. If the data point is labeled, the part of the total squared residual that cannot be explained by the estimated epistemic variance can then be found by Equation 4.7, where t_i is the true, labeled value for the data point dp_i. Similarly as to the previous approach, r_i^2 yields the remaining squared residual after the estimation of the epistemic variance has been subtracted.

From the assumption of normally distributed errors, the aleatoric gaussian probability distribution for the true value t_i, given the input data x_i, is represented by Equation 4.12, which can also be written as in Equation 4.13.

$$p(t_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2_{\epsilon_i}}}^{(-\frac{1}{2}\frac{((\hat{y}_i - t_i)^2 - \sigma^2_{\hat{y}_i})}{\sigma^2_{\epsilon_i}})} \tag{4.12}$$

$$p(t_i|x_i) = \frac{1}{\sqrt{2\pi\sigma^2_{\epsilon_i}}}^{(-\frac{1}{2}\frac{r_i^2}{\sigma^2_{\epsilon_i}})} \tag{4.13}$$

Finding the value for $\sigma^2_{\epsilon_i}$ which maximizes Equation 4.13 corresponds to finding the value for $\sigma^2_{\epsilon_i}$ which makes t_i the most likely value for the depth prediction, or in other words, the mean of the normal distribution. To find the best approximation for $\sigma^2_{\epsilon_i}$ a maximum likelihood scheme can thus be used. In order for the new ANN to adapt according to this objective during training, the network attempts to maximize the log-likelihood function shown in Equation 4.14, which can be simplified to Equation 4.15.

$$\ln p(t_i|x_i) = \ln\left[\frac{1}{\sqrt{2\pi\sigma^2_{\epsilon_i}}}^{-\frac{r_i^2}{2\sigma^2_{\epsilon_i}}}\right] \tag{4.14}$$

$$\ln p(t_i|x_i) = -\frac{1}{2}\ln(2\pi) - \frac{1}{2}\ln(\sigma^2_{\epsilon_i}) - \frac{r_i^2}{2\sigma^2_{\epsilon_i}} \tag{4.15}$$

Maximizing the log-likelihood function can also be described as an attempt to find the value for $\sigma^2_{\epsilon_i}$ which leaves the minimum value for the combination of the negative terms from Equation 4.15. Ignoring the constant term occurring first on the right hand side of Equation 4.15 yields the cost function shown in Equation 4.16, which is used as the cost function to train the new ANN, where b is the number of data points in each batch.

$$C = \sum_{i=1}^{b} \frac{\ln(\sigma^2_{\epsilon_i})}{2} + \frac{r^2}{2\sigma^2_{\epsilon_i}} \tag{4.16}$$

Thus, during training the network requires the input data X along with the remaining residuals from 4.7. It may seem curious to the reader that there is a mismatch between the ANN's input labels and the variable it attempts to predict. This is, however, simply because the cost function which is used during training does not report loss respective to the deviation from the label, but rather uses the label to perform gradient descent in order to maximize the likelihood, which is ultimately the goal.

Technical implementation of the maximum likelihood estimator ANN is not all that trivial. Few, if any high-level neural network APIs such as e.g. the open source python library Keras[3], comes with a log-likelihood cost function out-of-the-box. Custom cost functions are, however, possible to write with high level programming languages, even if the high level ANN API runs its networks on a computational graph on a C-based back-end, which is most often the case due the drastic efficiency requirements. Generally speaking, any such components that are meant to be run as part of the network must be vectorized in order to be able to run on the highly efficient back-end interfaces.

Another technical oddity of the implementation is the proposal of a summed cost function rather than the far more common mean version. The minor benefit of using a mean is perhaps most evident when dealing with relatively large data sets, as in the case study for this thesis. The reasoning behind this is that using a summed cost function would only perform optimally if the batch size remained constant during training. If the batch size were to change within an epoch, (a full run over the training dataset), it would cause skewed step sizes for the gradient descent. Such changes in batch sizes may occur when the remaining data points in an epoch is less than the normal batch size. Using the mean is then more

[3] Documentation for the library can be found at https://keras.io/

applicable and provides similar derivative values for the optimization algorithm, which is used to compute the step size. Moreover, the summed cost is of considerably larger numerical value than the mean, which again results in an even more significant numerical difference in the computed gradient. If the optimization algorithm does not properly account for this through the use of learning rates and decay, it could potentially result in too large step sizes, making convergence more difficult to achieve. To prove why these can be used interchangeably with respect to the computation of the gradient, consider the cost of a single batch containing a total of b data points. The total loss for a summed and mean cost would then be connected by the relation that the summed cost is equal to the averaged cost multiplied by b. The derivative, or the gradient, which is used in the optimization algorithms, is defined by Equation 4.17.

$$\frac{dL}{dx} = \lim_{\Delta \to 0} \frac{L(x+\Delta) - L(x)}{\Delta} \tag{4.17}$$

Multiplying the loss in Equation 4.17 with some constant, c, yields Equation 4.18. Simplifying this equations gives Equation 4.19, which shows that the same scaled relation holds for the losses respective derivatives as well. Effectively, this means that both computed gradients are in the same direction for any x.

$$\frac{d(c \cdot L)}{dx} = \lim_{\Delta \to 0} \frac{c \cdot L(x+\Delta) - c \cdot L(x)}{\Delta} \tag{4.18}$$

$$c \cdot \frac{dL}{dx} = c \cdot \lim_{\Delta \to 0} \frac{L(x+\Delta) - L(x)}{\Delta} \tag{4.19}$$

Nonetheless, the training of the new ANN then relies on a set of labeled data points from D, containing the labeled depth, t_i, the input data, x_i, and the remaining squared residuals, r_i^2, which is obtained from Equation 4.7. However, to generalize to unseen data points, any model from the N originally trained models that used dp_i during training is not used in the obtaining of r_i^2. However, for each data point $dp_i \in D$, we can obtain a reasonable estimate of the point prediction, $\hat{y}_i$, which we previously defined to be the mean of the predictions of all models, by taking the average of the predictions for all models that did not use dp_i during training. Effectively, this allows for the new maximum likelihood estimator ANN to train on all data points in D, provided there exist at least one model for each dp_i that did not use dp_i during its training. On the off chance that some data point was seen by all models during their training, the data point should be skipped.

Figure 4.12 presents a visualization of the overarching approach for setting up the method for computing the point prediction and width of PIs.

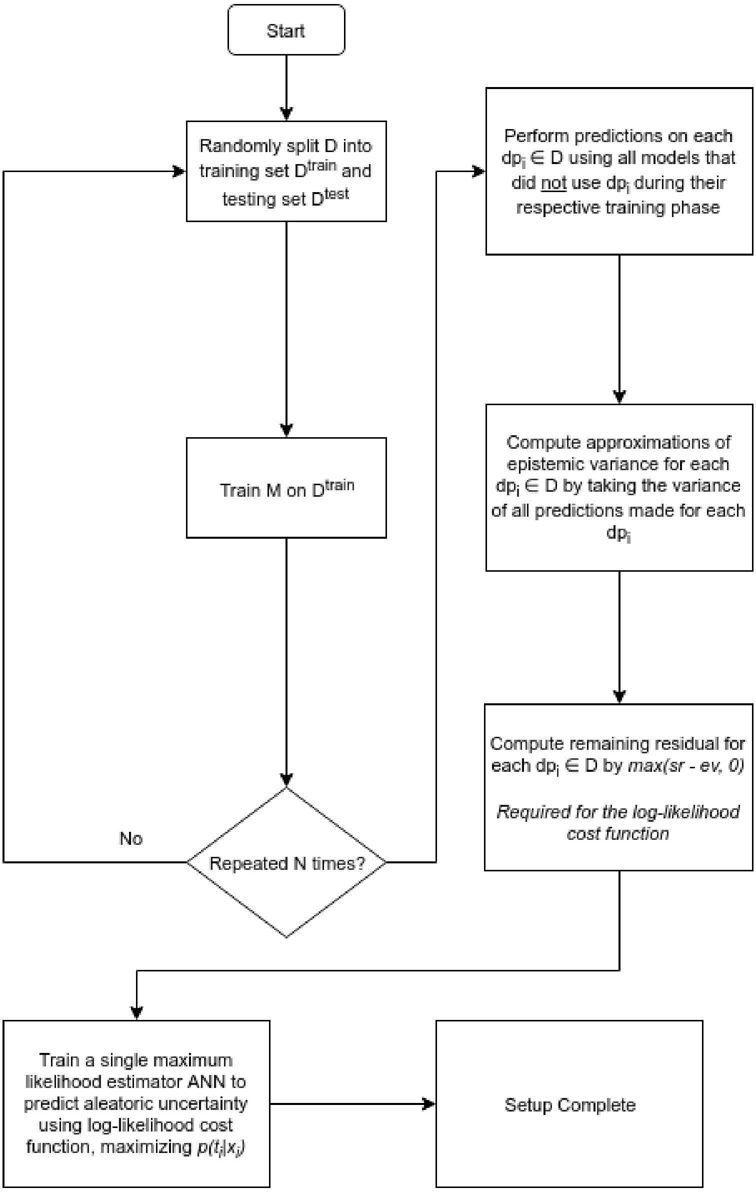

Start
Randomly split D into training set D^{train} and testing set D^{test}
Train M on D^{train}
Repeated N times?
No
Perform predictions on each $dp_i \in D$ using all models that did not use dp_i during their respective training phase
Compute approximations of epistemic variance for each $dp_i \in D$ by taking the variance of all predictions made for each dp_i
Compute remaining residual for each $dp_i \in D$ by *max(sr - ev, 0)*
Required for the log-likelihood cost function
Train a single maximum likelihood estimator ANN to predict aleatoric uncertainty using log-likelihood cost function, maximizing $p(t_i|x_i)$
Setup Complete

Figure 4.12: Overarching approach for setting up the method where *sr* denotes the squared residual and *ev* denotes the approximated epistemic variance

After the maximum likelihood estimator for the aleatoric variance has been trained, an estimated total variance can be computed for any new data point by $\sigma_i^2 = \sigma_{\hat{y}_i}^2 + \sigma_{\epsilon_i^2}^2$. After obtaining the estimated total variance, prediction intervals can be computed similarly as to the first approach, treating $\hat{y}_i$ as the single point prediction, and obtaining the PI width using the inverse of the cumulative distribution function.

Figure 4.13 presents a visualization of the approach for computing the prediction and the width of a PI for any data point, dp, where the true value is unknown.

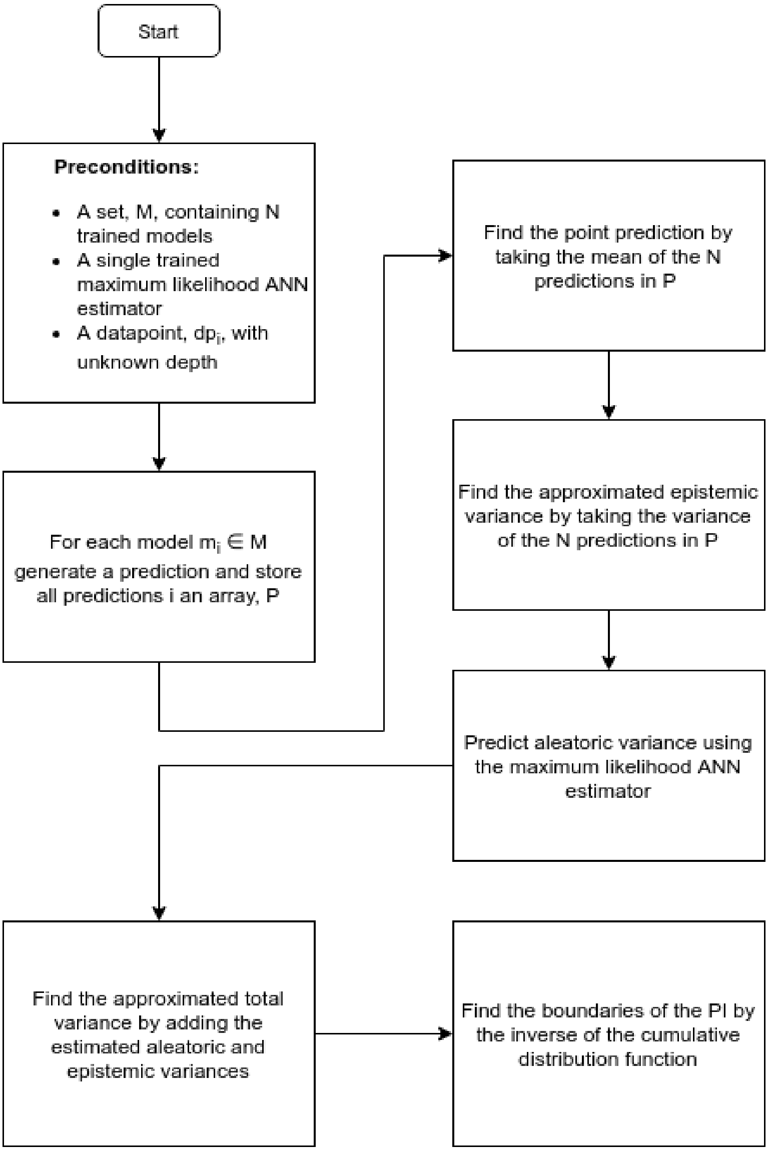

Start
Preconditions:
A set, M, containing N trained models
A single trained maximum likelihood ANN estimator
A datapoint, dp$_i$, with unknown depth
For each model m$_i$ ∈ M generate a prediction and store all predictions i an array, P
Find the point prediction by taking the mean of the N predictions in P
Find the approximated epistemic variance by taking the variance of the N predictions in P
Predict aleatoric variance using the maximum likelihood ANN estimator
Find the approximated total variance by adding the estimated aleatoric and epistemic variances
Find the boundaries of the PI by the inverse of the cumulative distribution function

Figure 4.13: Overarching approach for computing point predictions and PI

4.4.3 Quantile Regression

The Mean Absolute Error is a very common cost function used to train regression type ANNs. The approach as presented here does not account for epistemic uncertainty, and thus only the aleatoric uncertainty is accounted for. This is a significant drawback of this method.

Recall that the MAE cost function is a direct measure of the average of the absolute values of each residual in a training batch, and that it is used to optimize the parameters of a network with the goal of minimization.

A quantile function for some variable v, and some probability p computes the value $q(v)$ such that the probability of $v \leq q(v) = p$.

With quantile loss functions the aim is not to predict the most likely value, (the value that yields the minimal absolute error), but rather the value of a quantile for some percentile α. α denotes a number such that $0 \leq \alpha \leq 1$ and represents the percentile for which the single quantile covers. An α value of 0.5 corresponds to minimizing the MAE cost function, which also corresponds to learning the mean of the target probability distribution. An α value of 0.1 corresponds to the value that covers the lowest 10% of the target probability distribution. Thus, intuitively, a prediction interval can be said to be the product of two quantiles centered around a regression for the mean of the target probability distribution, (such as with MAE used as cost function).

However, the function for obtaining some quantile for DTB predictions provided some input X, is initially unknown. Essentially this means that the function $Q(y_i|x_i \in X, \alpha)$ for obtaining the desired quantile value for the α percentile is unknown. The goal then becomes to fit a regression to an arbitrary quantile, provided the known and labeled dataset. This is what is referred to as *quantile regression*.

If we assume that M is a fully trained model optimized with respect to MAE as the cost function, then any predictions made by M would be predictions corresponding to the mean of the target probability distribution, which corresponds to the quantile for $\alpha = 0.5$. Koenker *et al.* explained in 2001 how arbitrary quantiles could be constructed via the solving of a weighted and parametric optimization problem [23, pp. 145-146].

In the approach for training a quantile regressor, a customized cost function is used to train the network. Recall that α represents a number such that, theoretically, $0 \leq \alpha \leq 1$ and that the value of $\alpha \cdot 100$ denotes the percentile of coverage for the quantile. The loss function for a single prediction for a quantile that represents some coverage of the target probability distribution is defined by Equation 4.20, where ϵ_i is the residual of the prediction at i that occurred during training and $\mathbb{1}$ is an indicator function such that $\mathbb{1}_{(\epsilon>0)} = 1$ if $\epsilon > 0$ and 0 otherwise. Thus, Equation 4.21 is a simplified version of the same loss function, somewhat more easily readable for computer scientists.

$$L(\epsilon_i|\alpha) = \epsilon(\alpha - \mathbb{1}_{(\epsilon<0)}) \tag{4.20}$$

$$L(\epsilon_i|\alpha) = \begin{cases} \alpha\epsilon_i & \text{if}\epsilon_i \geq 0 \\ (\alpha - 1)\epsilon_i & \text{otherwise} \end{cases} \tag{4.21}$$

Recalling the earlier statement that α theoretically can have values such that $0 \leq \alpha \leq 1$ may now seem somewhat odd to the observant reader. This is because if $\alpha = 0$ and $\epsilon \geq 0$ at the same time, the loss would be 0 no matter the size of ϵ_i. The same phenomena would also occur for any situation where $\alpha = 1$ and $\epsilon_i \leq 0$. While it does not defeat the quantile properties, it cannot *reasonably* be claimed to be the desired effect for a prediction interval, as it would allow the quantile regressions to grow infinitely far beyond the uppermost and lowermost points in the dataset for any input. Thus, in practice whenever quantile regression is used for the specific goal of constructing prediction intervals, the valid value interval for α should be $< 0, 1 >$.

From Section 4.4.2 the reader may recall that technical implementations of custom loss functions are typically required to be of a vectorized format. This makes Equation 4.21 somewhat tough and troublesome to work with. However, considering the two conditional functions in the same equation, one can deduce that for any α such that $0 < \alpha < 1$ and a residual, ϵ_i which may be of either a positive or negative value, the maximum of the two conditional statements will always correspond to the true condition. Any positive ϵ_i would by the conditional function $\alpha\epsilon_i$ yield a positive loss, whereas the other conditional function, $(\alpha - 1)\epsilon_i$, would yield a negative loss. Any negative value for ϵ_i would yield the same effect in the opposite manner. For the perfect case where $\epsilon_i = 0$ the conditional function would not yield any difference, as both conditions would yield the same loss of 0.

Thus and by proxy, for a technical and vectorized implementation of the loss function, a simple maximum function of the two conditional functions would be sufficient. Consider the cost function calculation for a single training batch. Let *y_true* represent the vector containing the true labels for the data points in the batch and *y_pred* represent the vector containing the predicted values, such that *residuals* is the vectorized result of the vector subtraction operation performed on *y_true* and *y_pred*. Let *q* represent the percentile for the desired quantile of the regressor. The python code in Listing 4.1 then presents a python implementation of such a vectorized loss function, where *y_true* and *y_pred* are the provided parameters.

Listing 4.1: Python code for the quantile cost loss function

```
def quantile_loss(y_true, y_pred):
    residuals = y_true - y_pred
    loss = K.mean(K.maximum(self.q * residuals, (self.q -
        ↪ 1) * residuals))
    return loss
```

The important note to make for the loss function is that the computed loss for any individual prediction is dependent on the signum of the predictions residual. To exemplify why this matches the desired behavior, consider an α value of 0.1 and a training set consisting of only a single data point a, with a true label of $y_a = 0$. Let M represent a model under training. Assuming now that during training M overshoots and predicts $\hat{y}_a = 1$, which yields a residual value of $\epsilon_a = 1$. Now, the perfect quantile for α with respect to y_a and $\hat{y}_a$ corresponds to the value which lies between the two points, placed $(\alpha \cdot 100)\%$ of the distance between the points from y_a, which would be the value of 0.1. Let $Q(y_a|\hat{y}_a)$ denote this optimal quantile value. The distance from $\hat{y}_a$ to $Q(y_a|\hat{y}_a)$ is what is denoted as the loss of an individual prediction, and is also the result of the loss calculation rule $\alpha \cdot \epsilon_a$ when the residual, exemplified by ϵ_a here, is positive. Consider now the case where $\hat{y}_a = -1$, which yields the residual value of $\epsilon_a = -1$, with an emphasis on the negative signum. The optimal quantile value for α of y_a and $\hat{y}_a$, $Q(y_a|\hat{y}_a)$, would then be -0.1, being located $(\alpha \cdot 100)\%$ of the interval between the two points from y_a moving towards $\hat{y}_a$. Thus, whenever the residual, exemplified by ϵ_a in this case, is negative the distance between $Q(y_a|\hat{y}_a)$ and $\hat{y}_a$ is found by $(\alpha - 1) \cdot \epsilon_a$.

Averaging Equation 4.20 over some batch, where b denotes the batch size, gives the cost function presented in Equation 4.22, which is used to train the ANN in this approach.

$$C = \frac{\sum_{i=1}^{b} L(\epsilon_i|\alpha)}{b} \tag{4.22}$$

Thus, to obtain the upper and lower boundaries, two copies of M, M_1 and M_2 must be separately trained with alpha values α_1 and α_2. Let M_1 now represent a model that is fully trained with $\alpha_1 = 0.05$, and M_2 represent a model that is fully trained with $\alpha_2 = 0.95$. Let $\hat{y}_i^1$ represent a prediction from M_1 and $\hat{y}_i^2$ represent a prediction from M_2 for some input x_i. The final output would then be lower bound by $\hat{y}_i^1$ and upper bound by $\hat{y}_i^2$ to construct a prediction interval with a confidence of 90%.

4.4.4 Evaluation

Cross validation is, again, a tried and tested approach that is suitable for testing the performance of models with respect to both accuracy and precision. The separation of the dataset into training and testing sets is therefore necessary. However, a success criterion must still be established.

Establishing such a success criterion is relying on a thorough understanding of what the objective for the models actually is. The objective of the proposed approaches is to obtain a prediction interval of depth that should provide a realistic estimate of the actual DTB at the location of the measurement. Thus, the accuracy is of importance. An intuitive measurement of success for a typical regression model is the deviation from the predicted depth to the actual, labeled depth. Since the final output of the proposed approaches is not a single point, but rather an interval, such a measurement would, sadly, fall short. For PIs, one measurement of accuracy can be described as to what extent the model is able to provide true estimations in the sense that its estimation boundaries cover the actual, labeled DTB. This is more formally known and referenced as PI Coverage Probability, or PICP.

This metric alone has frequently been used to assess the quality of prediction intervals [11, p 6].

The PICP can be computed by Equation 4.23 where c_i is set to 1 if the PI covers the testing point and set to 0 if the testing point lies outside the PI's boundaries [11, p 6].

$$PICP = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} c_i \tag{4.23}$$

Another important aspect is the precision of the estimations. Precision for prediction intervals can intuitively be defined as the width of the predicted interval. A model that can guarantee 100% accuracy can, nonetheless, hardly be considered useful if its boundaries cover such a distance that it exceeds the required degree of precision for its use case. Thus, smaller prediction ranges are considered beneficial, however, only to that extent that they can still provide reasonable degrees of accuracy. None of the approaches proposed in this thesis provide a consistently constant PI width. Rather, they produce varying PI widths based on the confidence of not only the epistemic, but also aleatoric uncertainty. A mean of the ranges can be used to evaluate and compare the precision of the approaches. The mean of the ranges, or Mean PI Width (MPIW), can be computed as shown in Equation 4.24, where w_i denotes the range of the i'th PI, and n_{test} denotes the number of data points used during testing.

$$MPIW = \frac{1}{n_{test}} \sum_{i=1}^{n_{test}} w_i \tag{4.24}$$

Consideration of both metrics alone are definitively useful. However, since they are averaged across a distribution of data points, they cannot give any indication as to how often the true label falls outside the boundaries while considering the width of the interval at the same time.

A combined index for evaluating the PIs using both PICP and MPIW is therefore necessary. The CWC metric offers such a combined metric for evaluation. The metric, Coverage Width-based Criterion, weighs PICP heavier than MPIW, and attempts to compute a compromise metric of both metrics.

CWC can be computed by Equation 4.25, where μ is the PI's coverage probability and η is a control parameter which can be fine-tuned to provide the desired growth in the metric when the desired PI coverage is not reached. $\gamma(PICP)$ is set to 0 if $PICP \geq \mu$, or 1 if $PICP \leq \mu$.

$$CWC = MPIW(1 + \gamma(PICP)e^{-\eta(PICP - \mu)}) \tag{4.25}$$

4.4.4.1 Approach

The experimental approach for the evaluation and the comparison of the three PI construction methods follows the same systematic procedure for each of the evaluated methods. Two labeled datasets are used for evaluation of each of the methods. Since no true related observed resistivity profile exists for any borehole location, Kriging is used to construct estimations of the input data for the locations where bedrock depths are known from boreholes. The approach uses only the verified TS boreholes, as these provide the most accurate DTB values, and most closely resembles real world situations. A five stage process is followed to evaluate and compare the different methods. The process is repeated twice to compute metrics for both a 90th percentile PI and a 50th percentile PI.

In the first stage, the dataset is split into a training set (70%) and a testing set (30%). Where some of the models require splits in their training data during their internal training, the split is done on the 70% training set provided for the entire training. Each method is trained and tested on the same subset splits.

The second stage consists of training the models for each method. The training phase differs for the different approaches, however, regardless of required time for training, each method is allowed to train until convergence. The consumed time is stored for efficiency performance evaluation, as performed on a single Intel Core i7-2600 CPU running at 3.40GHz.

After each method has completed their respective training phases, they produce predictions for the data points in the testing set in the third stage. In the fourth stage the $PICP$, $MPIW$ and CWC metric scores are calculated for each of the methods individual predictions. Each metric is then stored so that aggregates can later be computed.

The process then restarts from the first stage, until the process has been completed 10 times. Upon completion of the 10th iteration, the mean, minimum and maximum value for each methods' metrics are aggregated and reported.

Chapter 5

Research Results

This chapter presents the results from the research, and offers answers to the research questions.

5.1 CNN Proposal

The following section covers the complete and comprehensive set of results from the evaluation of the proposed ML technique for DTB interpretation. Thus, the section relates to Research Question 1. The obtained performance measurements for the CNN model are evaluated and compared to the performance measurements of the standard MLP regressor previously proposed by NGI [27]. The performance is measured by both numerical metrics and a visual analysis.

5.1.1 Numerical Metrics Results

This section presents the comparable results of the proposed CNN model regressor and the standard non linear MLP regressor as described in Section 3.2.2 with respect to the numerical metric evaluation.

5.1.1.1 Metrics & Comparison

Table 5.1a presents the metrics as obtained in Case 1, where a total of 15 data points were selected from a single flight line. Table 5.1b presents the metrics obtained in Case 2, where a total of 130 data points were picked from a single flight line. Table 5.1c presents the metrics in the third case, where the data points were obtained from the Kriging method of the TS boreholes. Table 5.1d presents the metrics from the fourth case, where the data points were obtained from the Kriging method of all boreholes. Table 5.1e presents the metrics from the fifth case, where the data points were obtained from the Kriging method of TS boreholes using data from the more recent railroad & highway project.

The visualizations of the results for each dataset can be found in Appendix B, Section 7.7.

The metrics show that the CNN model's predictions are superior in terms of error magnitude. The only metric that benefit the MLP regressor was the minimum MAE achieved for a single round in Case 1. Interestingly, this is not true for the minimum RMSE achieved for any single round in the same case. This indicates that while the standard MLP regressor was able to achieve a lower MAE for the round, it still had sporadic erroneous spikes with magnitudes that exceeded those of the CNN model. The reasoning behind the deduction is that RMSE penalizes larger errors to a greater extent.

An inspection of the metrics alone shows clear evidence that the CNN model outperforms the standard MLP regression technique for the survey area investigated in the case study. The difference in performance is most clear for small data sets with manually selected depth labels, such as provided in Case 1. In this case, the CNN model provides a reduction in the mean RMSE of 52.6%, and a reduction in the mean MAE of 49.8%. When the size of the training dataset increased, the performance differences was reduced. However, for the largest dataset with manually assigned labels tested in this evaluation (Case 2), the CNN approach produced a reduction in the mean RMSE of 25.1%, and a reduction in the mean MAE of 27.3%, which can still be considered substantial.

Using the larger data sets, where the depth labels were obtained from boreholes, such as with Case 3, Case 4 and Case 5, the CNN model still yields consistently improved performance. Case 3 and 5 are perhaps the most important cases, as they most closely

Metric	MLP	CNN
Training time (Seconds)	0.1	3.2
MAE (Mean)	8.20	4.12
MAE(Max)	19.13	7.67
MAE (Min)	1.43	1.69
RMSE (Mean)	11.71	5.55
RMSE (Max)	27.50	10.19
RMSE (Min)	1.58	1.28

(a) Metrics for Case 1

Metric	MLP	CNN
Training time (Seconds)	0.16	14.46
MAE (Mean)	5.52	4.01
MAE(Max)	8.04	6.17
MAE (Min)	4.24	3.19
RMSE (Mean)	8.12	6.08
RMSE (Max)	11.27	8.02
RMSE (Min)	5.65	4.09

(b) Metrics for Case 2

Metric	MLP	CNN
Training time (Seconds)	0.1	42.1
MAE (Mean)	2.31	2.12
MAE(Max)	2.51	2.30
MAE (Min)	2.15	1.94
RMSE (Mean)	3.03	2.69
RMSE (Max)	3.42	2.93
RMSE (Min)	2.73	2.42

(c) Metrics for Case 3

Metric	MLP	CNN
Training time (Seconds)	.8	71.6
MAE (Mean)	4.76	3.95
MAE(Max)	5.04	4.42
MAE (Min)	4.43	3.94
RMSE (Mean)	6.33	5.49
RMSE (Max)	6.67	5.84
RMSE (Min)	5.81	5.37

(d) Metrics for Case 4

Metric	MLP	CNN
Training time (Seconds)	.5	65.4
MAE (Mean)	5.78	4.19
MAE(Max)	6.51	4.39
MAE (Min)	4.59	4.01
RMSE (Mean)	7.53	5.30
RMSE (Max)	8.84	5.56
RMSE (Min)	5.91	5.12

(e) Metrics for Case 5

Table 5.1: Metrics obtained from Cross-Validation evaluation of each case

resembles real world situations. This is because TS boreholes are more common than RPS boreholes in surveys conducted for mapping of bedrock topography.

Nonetheless, the CNN model requires significantly longer computation time for training the network, as visible on the metrics for the cases with the larger data sets.

5.1.2 Visual Results & Comparison

The complete set of visual representations that were produced during the experiment can be found in Section 7.7 of Appendix B. In the visualizations for Case 1 and Case 2, the depth labels are marked by black dots on the chart. For simplicity, the elevation of the ground surface has been flattened, such that the top of each vertical resistivity profile is found at 0 depth meters. The input data for the standard MLP regressor for each depth label is the single resistivity profile in which each depth label is found. This is also true for the CNN model. However, the CNN model also requires the two nearest neighboring profiles on each side of the profile containing the label. Thus, no prediction can be made for the two outermost resistivity profiles on either side of the flight-line, as no valid input data exist for the CNN model.

In each visualization, a white line with black borders represents the depth predictions from the CNN model. A black line with white borders represents the depth predictions from the standard MLP regressor.

Three visualizations of Case 1 is represented in Figure 7.1, 7.2, and 7.3. Each of the visualizations contains a different, and increasing number of training data. The need for a visual analysis may seem evident upon inspection of Figure 7.1 and Figure 7.2, where the standard MLP regressor produces good results near the available training data points, but the general predictions for the complete flight-line is far less than satisfactory. The same phenomena is perhaps more visually evident in Figure 7.4 and Figure 7.5, where the input data is varying to a much larger extent.

Figure 7.6 presents the second case with 130 data points used in the training phase. The dots have been removed from the visualization to make it more clear.

Figures 7.7, 7.8, 7.9, 7.10, and 7.11 present visualizations of the models as performed for Case 3.

Figures 7.12 and 7.13 present visualizations of the performances for Case 4.

Figure 7.14 presents a visualization of the performance for Case 5.

5.1.3 Analytic Conclusions

Research Question 1 ask the questions of what ML technique that may improve the results for automated DTB interpretation. The numerical metric analysis showed significant potential for convolutional type neural networks for the task of DTB interpretation from AEM data. The findings undoubtedly make it a possible candidate for automated DTB interpretation. The case study shows how techniques from the academic field of computer vision can be used in combination with more conventional features from machine learning to approach the geotechnical problem of DTB interpretation. Allowing the model to consider larger portions of the context yielded beneficial results.

5.2 PI Methods

This section covers the results as related to the experimental approach of evaluating the PI construction methods for the problem of automated DTB interpretation, and it also provides answers to Research Question 2.

Due to the long names of the proposed PI construction methods, a set of placeholder names will be used in this section. Method 1 represents the ensembled method under the assumption of a single fixed aleatoric uncertainty, as discussed in Section 4.4.1. Method 2 represents the slightly more advanced method where the aleatoric uncertainty is approximated from a separate maximum likelihood estimator ANN, as discussed in Section 4.4.2. Method 3 corresponds to the Quantile Regression technique, as presented in Section 4.4.3.

5.2.1 Metrics & Comparison

The following section aims to present and evaluate the resulting metrics from the experimental approach on the construction and evaluation of prediction intervals.

The numerical results from the approach is presented in Table 5.2 and Table 5.3 for the 50% PI and the 90% PI respectively for the first dataset from the highway construction project.

The metric referenced as *Setup Time* presents the average time in minutes that was required for training and setting up the respective method.

PI Size	Metric	Aggregation	Method 1	Method 2	Method 3
50% PI	**PICP**	**Max**	0.677	0.571	0.688
		Min	0.442	0.429	0.338
		Mean	0.544	0.492	0.526
	MPIW	**Max**	5.192	4.770	6.844
		Min	4.654	4.122	4.020
		Mean	4.938	4.438	5.173
	CWC	**Max**	13.665	12.970	24.400
		Min	4.655	4.269	4.605
		Mean	6.409	8.243	8.267
	Setup Time (Minutes)	**Mean**	7.22	9.2	1.4

Table 5.2: Aggregation of the results for the 50% PI for the first dataset

PI Size	Metric	Aggregation	Method 1	Method 2	Method 3
90% PI	**PICP**	**Max**	0.935	0.949	1.0
		Min	0.845	0.792	0.812
		Mean	0.901	0.895	0.912
	MPIW	**Max**	12.530	14.132	14.550
		Min	11.579	8.546	10.880
		Mean	12.134	12.367	12.764
	CWC	**Max**	32.0883	35.137	36.093
		Min	12.154	11.930	12.046
		Mean	19.471	22.293	20.590
	Setup Time (Minutes)	**Mean**	7.40	9.83	1.32

Table 5.3: Aggregation of the results for the 90% PI for the first dataset

The comparison of the mean PICP for each method for the first dataset shows that all methods are close to the desired coverage of 50% for the first case. Method 2 seems to have a tendency to slightly under-predict the PI's width, such that its mean coverage is less than the desired coverage. This is seen by Method 2 having a mean PICP which is 0.8% and 0.5% less than the desired coverage of 50% and 90% for the two cases respectively. For the 50% PI, Method 2 does, however, seem to be the most stable of the three, with a discrepancy from the max and min PICP of $PICP_{max}(0.571) - PICP_{min}(0.429) = 0.142$, as opposed to Method 1 with 0.235 and Method 3 with 0.350. The minimum PICP values of 0.442 and 0.338 for Method 1 and Method 3 show that these methods also have the potential to significantly under-predict the required width of the PIs, but judging from the mean values this seems to be drastic exceptions. For the 90% PI, Method 1 provides the most stable PICP results, and Method 3 still holds the largest discrepancy from the maximum to the minimum.

An analysis of the MPIW reveals that there are larger deviations on average for the methods when constructing large PIs. This may also seem intuitive, as such a large PIs would require capturing drastic outliers, which often are irregularly distributed. For the 50% PI, Method 1 and 2 yields the lowest mean widths of their PIs, with 4.938 and 4.438 meters respectively. Method 3 yields somewhat larger widths with a mean of 5.173 meters. The worst case (maximum value) scenarios for the MPIW for the 50% PI shows a significantly worse result for Method 3 than Method 1 and Method 2. However, for the 90% PI the deviance in the maximum MPIW from Method 1 and Method 2 to Method 3 is drastically reduced.

Method 1 has, on average, more coverage than Method 2 while still being similar in width. This is also reflected in the CWC metrics, where Method 1 has the lowest mean for both the 50% and the 90% PIs. For the 50% PI, Method 3 yields a very high maximum CWC. This large maximum seems reasonable, due to its very low minimum PICP coverage, and the two are most likely related. For the 90% PI, the CWC scores are much more similar across the three methods. While neither Method 2 or Method 3 come close to the performance of Method 1, it is interesting to note how Method 3 surpasses Method 2 in performance for the 90% PI, but not for the 50% PI. This seems to be a direct result of Method 2 having a lower PICP on average, and thus being drastically more penalized by the CWC, as it considers PICP to a larger extent than the MPIW.

The difference in the computational demand is, nonetheless, significant. Method 1 and Method 2 requires the training of $N = 20$ and $N = 20 + 1$ separate models respectively, bearing in mind that Method 2 requires an additional model for the prediction of the aleatoric variance. On average, Method 1 used more time than Method 3 for setting up the method by a factor of more than 5. Method 2, which is even more computationally demanding requires more than 7 times the computational time of Method 3 for setting up the method.

Table 5.4 and Table 5.5 presents the same results for the more recent railroad & highway project, which presented data with a vastly more complex geology.

PI Size	Metric	Aggregation	Method 1	Method 2	Method 3
50% PI	**PICP**	**Max**	0.617	0.444	0.570
		Min	0.565	0.388	0.523
		Mean	0.591	0.416	0.547
	MPIW	**Max**	10.703	7.684	12.986
		Min	8.645	6.976	12.546
		Mean	9.674	7.330	12.766
	CWC	**Max**	10.703	28.387	12.986
		Min	8.645	21.146	12.546
		Mean	9.674	24.766	12.766
	Setup Time (Minutes)	**Mean**	10.12	13.60	2.0

Table 5.4: Aggregation of the results for the 50% PI for the second dataset

PI Size	Metric	Aggregation	Method 1	Method 2	Method 3
90% PI	PICP	Max	0.958	0.916	0.921
		Min	0.939	0.907	0.883
		Mean	0.951	0.911	0.901
	MPIW	Max	34.404	30.234	41.740
		Min	32.264	28.920	37.315
		Mean	34.335	29.575	39.527
	CWC	Max	36.403	30.233	81.466
		Min	32.264	28.914	41.739
		Mean	34.334	29.575	61.603
	Setup Time (Minutes)	Mean	14.51	17.13	2.63

Table 5.5: Aggregation of the results for the 90% PI for the second dataset

The more challenging geological topography in this dataset resulted in larger residuals on the performance evaluation of the CNN (The previous dataset corresponds to Case 3, while this dataset corresponds to Case 5 as presented in Table 5.1). Intuitively this should result in wider prediction intervals for covering the same probability. While this is shown to hold true for this test case, it is interesting to note the magnitude of the width increase, and how the more complex geology affects the PI construction methods' performance. The average mean MPIW for all the methods increased from 4.85 meters to 9.92 meters for the 50% coverage interval. The increase is much more significant for the 90% coverage interval, where the average mean MPIW increased from 12.42 meters to 34.78 meters.

Method 2 again showed a tendency to under-predict the width of the PI, resulting in a significantly lower PICP than the desired 50%. Method 1 and Method 3 both had higher than expected PICP metrics, even for the lowest record of the 10 runs.

Method 2 constructed the intervals with the narrowest widths. However, by having a too narrow interval, it is unable to cover the required percentage of testing points and yields a drastic increase in the combined CWC metric. Method 1 was able to find the best combination for width and coverage of the three methods, width a mean PI width of 9.674

meters.

For the 90% PI of the second dataset the results shifts somewhat, where Method 2 yields the lowest mean CWC, even with the lowest mean PI width.

Table 5.6 presents an overview of the aggregated approximated variances for the predictions by Method 1 and Method 2 for the first dataset. The epistemic variance is shown to account for a small fraction of the total approximated variance. Method 2 has a large deviance from its minimum to its maximum approximation for the aleatoric uncertainty, whereas Method 1 has very similar values. These are intuitive results, as Method 2 attempts to approximate an aleatoric variance for each data point, whereas Method 1 computes a fixed approximation, which is used regardless of the input during prediction.

Method	**Aggregation**	$\sigma^2_{\hat{y}_i}$	$\sigma^2_{\epsilon_i}$
Method 1	**Max**	6.826	13.756
	Min	0.030	10.828
	Mean	0.636	11.939
Method 2	**Max**	5.519	27.480
	Min	0.067	2.809
	Mean	0.709	12.136

Table 5.6: Aggregation of approximated variances for Method 1 and Method 2 for the first dataset

Table 5.7 presents the same overview for the second dataset. The metrics shows how the approximated uncertainty variances grows significantly with minor decreases in the accuracy of the core regressor. The ratio between epistemic and aleatoric mean variances has also shifted to become more even. Nonetheless, the inaccuracy of the models for predictions made on this dataset is arguably too large for the intervals to be of significant benefit.

Method	Aggregation	$\sigma^2_{\hat{y}_i}$	$\sigma^2_{\epsilon_i}$
Method 1	**Max**	61.609	33.517
	Min	0.527	25.903
	Mean	26.867	29.710
Method 2	**Max**	62.501	146.689
	Min	0.296	0.107
	Mean	23.830	28.456

Table 5.7: Aggregation of approximated variances for Method 1 and Method 2 for the second dataset

5.2.2 Analytical Conclusions

Research Question 2 asks how PIs can best be automatically computed for providing a sense of uncertainty for regression point predictions for DTB levels. Based on the results provided in the previous section, Method 1 is deemed the generally most successful of the three evaluated methods. While all methods were able to produce intervals with close to desired coverage, Method 1 produced the best combination between coverage and width of the PIs for both small and large intervals for the first dataset. On the second dataset, Method 2 produces the best intervals for 90% coverage.

Method 1 generally produced the best metrics. On the other hand, it yielded a large increase in time consumption for the setup phase, compared to Method 3. Method 3 provided significantly less requirements for setup time, and thus it should also be evaluated as a potential candidate when the computational resources are limited or whenever the data sets grow to significant sizes. However, Method 3 seemed to produce much too wide intervals in cases where the uncertainty was high, such as for the second dataset. Nonetheless, this case study showed that the computationally demanding task of ensembled approaches, such as with either Method 1 or Method 2, did provide increased precision.

Chapter 6

Discussion

6.1 CNN Method Proposal Discussion & Evaluation

While NGI's earlier proposal took the first step in uncovering the benefits of automated predictions from ANNs in the field of DTB interpretation [27], the new CNN proposal shows significant advances by approaching the problem as a computer vision problem, and including a convolutional layer.

The CNN's predictions proved superior to the standard MLP regression approach with sole regards to the error metrics. The visual analysis revealed that when few data points are available for training, the standard MLP approach is far less able to adapt to unseen data points and provide desired and sensible predictions. This was seen, as the standard MLP's predictions contained large outliers. As the size of the training data set increases, the standard MLP approach is increasingly able to generalize to the data, and provide progressively better predictions, albeit never surpassing the CNN in terms of accuracy.

Case 5 uses data from an area with a vastly more complex geological setting than the other cases. While the results from this case carried too much uncertainty for practical use, the performance increase also holds true for this case, which yields evidence that the approach is adaptable to different geological topographies.

The benefits from using the CNN model were more evident when the depth labels were manually assigned to resistivity profiles, rather than obtained from boreholes and interpolation of resistivity profiles. This indicates that the method is increasingly more beneficial when the training data contains less noise. The reasoning behind this is that the methods where the resistivity profiles in the training data is interpolated are subject to more noise than those where it is obtained directly from the inversion data. The additional noise stems from uncertainty in the interpolation step.

The largest outliers in the standard MLP's predictions occurred when the size of the training sets were considerably small. This may be caused by the model overfitting to the known data points. Figure 7.5 from Appendix B clearly strengthens this theory. The figure shows that the standard MLP's predictions are clearly good for the unknown data points that are both similar, and in close proximity to the known data points. However, the predictions are not remotely rational for a large portion of the data points that differ from those seen during training. While the standard MLP regressor as implemented with SciKit learns framework includes early stopping in an attempt to avoid overfitting, the Dropout technique as included in the CNN model may make the CNN less prone to the same type of overfitting. Moreover, the CNN has vastly more neurons in the last densely connected layer, which allows it to learn a vastly larger ”rule-set” for its predictions. The visualizations from Case 3 also revealed that with a training set of significant size, the two methods did not differ drastically in their predictions.

Different topologies of the standard MLP such as increasing its depth or horizontal size in the hidden layer(s) may be able to increase its accuracy and accountability. This was not experimented with, explored, or accounted for during the evaluation.

The visualizations, nonetheless, also revealed sporadic spiking for the standard MLP's predictions, even when trained on the largest training set. Examples of these spikes can be found in Appendix B in Figure 7.7 at approximately 1600 distance meters and in Figure 7.10 at approximately 1450, 1500 and 1700 distance meters. Similar spikes were far less common for the CNN approach, even with smaller training sets. This discrepancy closely matches the motivation behind the proposal, where single noisy resistivity profiles are thought to cause larger irregularities when contextual interpretation is emitted from the interpretation process.

The perhaps most evident drawback of the CNN approach can be claimed to be the difficulty in obtaining training data, as it relies on numerous interpolations of resistivity profiles. An alternative method of obtaining training data could solely rely on expert picking, where a geotechnical expert selects depth labels for sets of 5 neighboring resistivity profiles. The amount of work required for obtaining a sufficiently sized training data set using such expert picking may vary with the size of the original dataset. However, as one of the core motivations behind this research is to eliminate manual labor, this cannot be regarded as anything less than a significant drawback. Furthermore, the loss of highly accurate training data impacts the overall accuracy of the model's predictions, which again reduces the actual usefulness of the automated contextual interpretation technique.

The drawbacks presented above spiked the motivation for the parts of the thesis related to valid training data for the CNN approach. A proposal for expansion on the conventional Kriging interpolation system was presented to provide a method for automatic obtaining of valid training data points. This spatially distributed Kriging technique, covered in Section 4.3.2, proved a viable solution. However, it also introduces new uncertainty sources in the interpolation process, which may further weaken the final predictions.

The lack of predictions for the $n = 2$ outermost resistivity profiles is also worthy of mentioning, where n denotes the amount of neighboring resistivity profiles included on each side of the resistivity profile which is under interpretation. This limitation effectively strips approximately 70 meters off each side of the distance of a flight line for in which depth predictions can be made.

The case study utilized different frameworks and optimizers for the two methods. This allowed for the testing of a model that was highly similar to NGI's proposal [27]. However, it also caused the training techniques of the two methods to differ slightly. The CNN method used dropout for avoiding overfitting, whereas the standard MLP approach relied on early stopping. Moreover, they also used different optimization schemes, which may have influenced the convergence of the training.

The results stemming from the DTB predictions with use of ANNs are easily reproducible. The parameters of any network can be stored and retained for later use. Thus, when reproduction of the results are desirable, a new model can be instantiated with the stored parameters, and the exact same results can be provided by the new model when the same input data is fed. This is a significant benefit that the automated approach presents when compared to conventional techniques, where manual labor is a significant factor in making the predictions.

A large portion of this thesis focuses on uncertainty estimation of predictions, and a model is thus required for the experimental approach of uncertainty estimations. Based on the discussion provided in this section, the CNN method was chosen for further evaluation with respect to uncertainty. This decision was made with two core concepts as reasoning. Firstly, the CNN model was able to produce significantly more accurate predictions. Secondly, even though the consumed computation time is significantly higher for the CNN model, it does not pose any drastic risk for completion of the uncertainty evaluation, and can thus be considered a valid trade-off.

6.2 PI Construction Methods Discussion & Evaluation

Each of the three methods proved viable solutions for construction of prediction intervals. Below follows two visualizations of the prediction intervals that each of the three methods produced for a sample flight line. The flight line is a real world example from the case study's survey area, and Figure 6.1 and Figure 6.2 shows the constructed PIs which represent 90% and 50% coverage respectively.

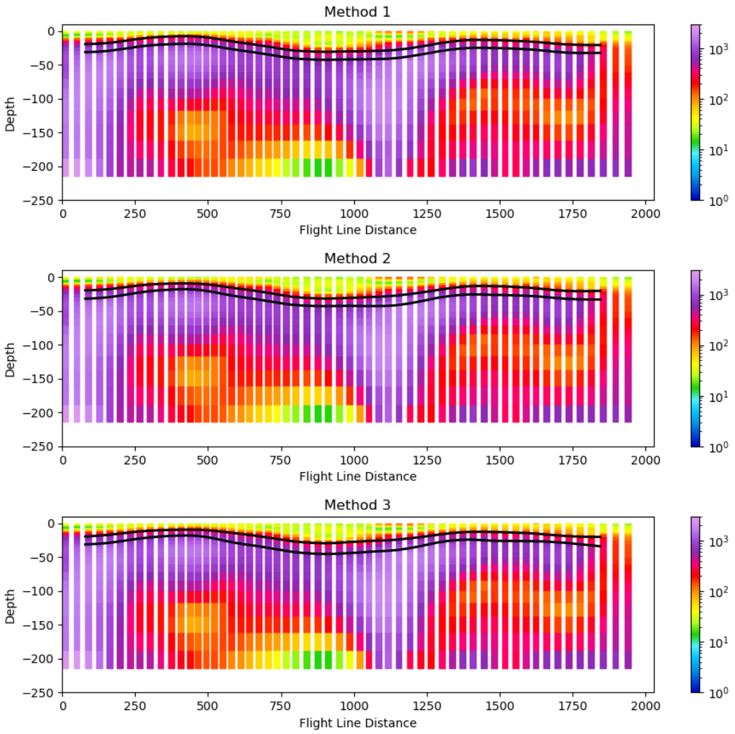


Figure 6.1: Visualizations of Prediction Intervals for 90% Coverage for a sample flight line

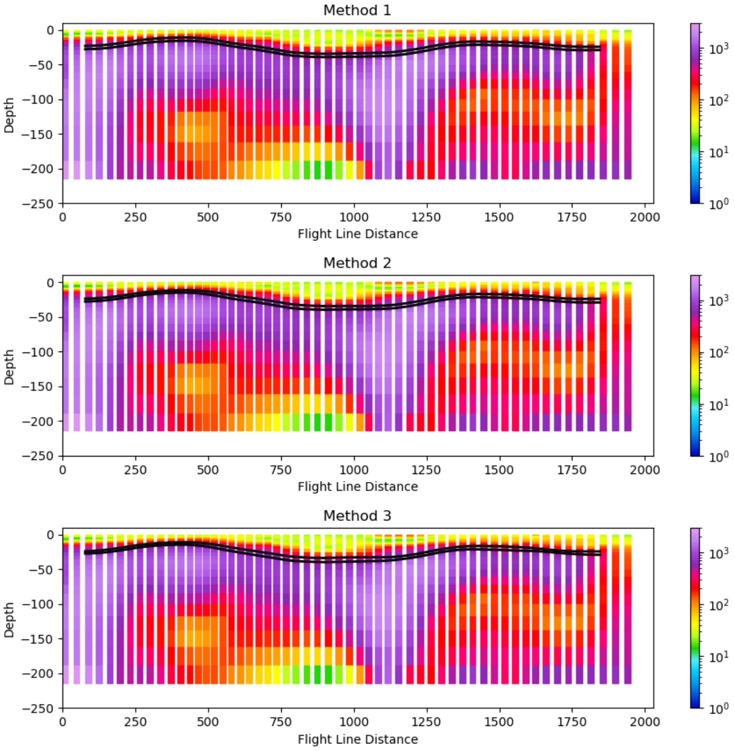


Figure 6.2: Visualizations of Prediction Intervals for 50% Coverage for a sample flight line

The visualizations show how the three methods are able to produce informative visualizations for arbitrary PI percentiles. The simplicity of the visualizations can be deemed an important factor of their usability, as geotechnical experts may use these for quickly evaluating the confidence of any automated DTB interpretation model.

The ensembled approach, originally proposed by Heskes in 1996 [17], yielded a slight benefit over the Mean Variance Estimation approach, as proposed by Nix & Weigend in 1994 [30]. The Mean Variance approach assumes the input dataset not only to be sufficiently large, but also to have diversity of its elements such that the epistemic uncertainty can be assumed reduced to 0. This assumption can not be said to hold true for the sake of DTB interpretation, where the availability of boreholes, and in turn, the availability of labeled data points is often small, especially in the early phases of investigations. The benefits of the ensembled approach were also evident in the research results in this thesis, where the quantile regression technique did not prove superior with respect to mean CWC for any of the test cases.

Moreover, the two ensembled approaches allow for even further analysis of the uncertainty, by producing both aleatoric and epistemic uncertainty approximations at any point. Analysts may use this information to increase their understanding on what should be regarded as bottlenecks for the accuracy of their predictions. High epistemic uncertainties correlate to insufficient training data, and high aleatoric uncertainty values are indicators of inherently noisy data. Thus, analysts may interpret the results to better understand where to focus their efforts in attempts to improve a models accuracy.

Reproduction of the intervals are also easily reproducible, following the same logic as described in the previous section. Storing the parameters of several models requires very little disk space, and allows the interpreters to quickly load the previously trained networks. Since each model will produce the exact same predictions for the same input data, the same intervals can be reconstructed. This presents a great benefit, since the traditional methods required a human interpreter to manually predict uncertainty values. The predictions of a manual interpreter can also be stored. However, it is considerably more challenging to store the reasoning, intuition, and expertise of a human interpreter.

The evaluated methods also have a clear definition as to exactly what they predict, namely a prediction interval for some percentile. The prediction interval is also a measure stemming from the previous preciseness of the predictor, whereas with manually assigned uncertainty values there is no strict relation between the two. Nonetheless, this also yields a slight drawback if the interpreter is not acutely aware of how the intervals are defined. The intervals does not claim that the bedrock surface will definitively reside within its boundaries for the percentage of confidence. Rather, it merely claims that the bedrock surface will reside within its boundaries for the given confidence level *based on its historic observations*. Essentially, this emphasizes the importance of noise reduction and filtering of bad data points for the training phase.

In the case study provided in this thesis using the first dataset, the aleatoric uncertainty accounted for the largest portion of the total uncertainty. Thus, for the model to improve its accuracy on DTB level predictions for these datasets, improving the preciseness of the input data may seem an intuitive place to start.

This uncertainty distribution is perhaps to be expected, as the complete interpretation process relies on several approximation and interpolation steps for the input data before they are ultimately fed to the interpretation model, each including their own approximation uncertainties. The different sources of uncertainty may stem from the inversion process, or even from the interpolation of unknown resistivity profiles surrounding boreholes, which are interpolated from the already approximated known resistivity profiles from the inversion process.

The test on the second dataset, which contained much more complex input data, showed a much more even distribution of the uncertainty variances. This may be partly explained by the increased complexity in the relation between the input data and the labels. The output space also range wider for this second dataset, which further increases the complexity. This increase in data complexity may also be part of the reason for the superior performance of Method 2 for the 90% PI. Method 2 may have have been more able than Method 1 to narrow the PI width where the aleatoric variance was approximated low, whereas Method 1 assumes a fixed aleatoric variance for any data point. To obtain 90% coverage Method 1 has to assume a high aleatoric variance, which damages the PI's performance on the data points where the aleatoric variance in reality is small.

The evaluation of the three different methods for construction of the intervals uncovered both benefits and flaws for each of the methods. Depending on the use case, these benefits and flaws can be evaluated such that a construction method can be chosen to yield the best results based on the problem statement and the availability of computational resources.

Method 1, using an ensembled method to account for the epistemic uncertainty, and a single simplistic computation for the fixed aleatoric uncertainty generally yielded the best results for this case study. A probable explanation for why this method performed better than Method 2 on the first dataset is that the relation between the input data and aleatoric uncertainty may have been too vague at each point for the maximum likelihood estimator to yield any significant difference. Thus, this estimator may have posed more noise than explanation of the total uncertainty. Method 1 can thus be claimed to be a better option when the point wise correlation between input data and aleatoric uncertainty is vague. It also requires less computational resources than its counterpart of Method 2.

Method 2 is similar to the aforementioned Method 1 with its ensembled approach, but it assumes a certain correlation between the input data and the aleatoric uncertainty at any point. Knowledge of the degree of presence for this correlation is not straight forwardly obtained, and relies on the fitting of some model to the data followed by a manual analysis of the models performance. Whenever there exists a strong correlation between the input data and the aleatoric uncertainty, Method 2 should theoretically be able to yield similar coverage with smaller mean width of its interval. The reasoning for this is that it should be able to reduce the PI width where the aleatoric uncertainty is small, and widen it where the aleatoric uncertainty is large. The additional computational resource requirements from the maximum likelihood estimator can be considered to be small, requiring the training of only a single extra model. Regarded in relation to the requirement of N separate models as result of the ensembled approach, an increase from N to $N+1$ is often insignificant. This intuition holds true for this case study, where the number of training data points can be regarded as relatively few, and of relatively small sizes, consisting of a mere $5 \cdot 25 = 125$ separate input neurons. Moreover, the topology of the ANNs is both small and simplistic. However, if the training dataset grows much larger, and the topology of the networks grows more complex, the addition of a single extra network may cause a considerable increase in the computational resource requirements.

Method 3, using the quantile regression technique, is by far the least computationally demanding of the compared approaches. Still, the approach did yield decent results, averaging at above the desired coverage for both the 90% and 50% coverage tests, albeit with higher mean widths of its intervals than the two ensembled methods. The quantile regression technique proved rather unstable, with high deviations from its maximum to minimum PICP. This arguably stems from the methods inability to account for epistemic uncertainty, and thus the degree of epistemic uncertainty that is accounted for in the resulting interval may be largely random. The quantile regression technique can thus be claimed to be a good choice for uncertainty estimation, when the known dataset is of sufficient size and variability, such that the epistemic uncertainty is insignificant, and when the availability of computational resources are low.

Limited availability to computational resources enforced a somewhat strict restriction on the number of separate models that were feasible to train for each of the ensembled methods during the experimental approach. The number of 20 separate models allows for 20 separate predictions, which cannot be claimed to be sizeable by any means. The approximation of epistemic and aleatoric uncertainty could therefore potentially be somewhat biased towards some subset of the input data if the random sampling had a sizeable portion of bad splits (many similar data points in the testing portion). The entire metrics computation process was repeated in 10 iterations and the results were averaged in an attempt to mitigate this limitation.

Table 6.1 shows a compressed presentation of the distinct methods with relating conditions.

	Allows for further analysis relating to uncertainty distribution (aleatoric & epistemic)	**Automatic result reproduction**	**Requires presence of relation between input data & aleatoric variance**	**Minimal computational resource requirements**
Method 1	Yes	Yes	No	No
Method 2	Yes	Yes	Yes	No
Method 3	No	Yes	No	Yes

Table 6.1: Conditional representation of the separate methods

Chapter 7

Conclusions & Future Work

This chapter concludes the thesis, and will discuss future work that bears potential of further improving the DTB interpretation process for bedrock modelling.

7.1 Taking the Research Further

This thesis showed potential for convolutional neural networks in a domain outside of what can strictly be defined as computer vision. However, it must not be considered an exhaustive experimentation of the possibilities that CNNs introduce. A natural next step may therefore be to conduct a more thorough investigation into different CNN approaches for DTB predictions. A more exhaustive investigation could attempt to understand what the optimal number of convolutional layers in the network is, what the optimal number of filters in each convolutional layer is, and also if pooling layers could introduce new improvements. Such an investigation would benefit from exploring the domain in an attempt to understand why some ANN methods supersede the performance of others.

7.2 More and Differing Data Sets

The case studies from this thesis presented an evaluation of the different methods using data from two geographically separate survey areas in Norway. New investigations exploring the performance of the different methods for different survey areas altogether, (perhaps across country borders), may also present new and important information, as geotechnical data varies to a large extent with the geological complexity of the area. Knowledge of why, how, and to what extent the automated methods are affected by the geological setting are important factors for uncovering new understandings on how the DTB interpretation process can be improved.

7.3 Interpolation of Prediction Data Points

The research results from the case study showed that the vast majority of the uncertainty stemmed from aleatoric origin. Recalling that aleatoric uncertainty arise from stochasticity and noise inherent in the data gives a strong indication on where the approach could be further optimized. Further research could examine where the majority of the aleatoric uncertainty stems from, and explore methods for reducing it. Future research could investigate different root sources for noise introduction. Such sources could be the original soundings form the AEM vessel, anthropogenic noise, inversion, or from interpolation of resistivity profiles for boreholes locations. Knowledge on where uncertainty is introduced in the process can allow for more targeted investigations with the aim of noise and uncertainty reduction.

7.4 3D Kriging

The method used for obtaining resistivity profiles at arbitrary locations in this thesis relied on layered 2D Kriging. While the resulting interpolated data proved viable, a more thorough examination could be done for the alternative 3D layering technique. The layered 2D technique can only interpolate values on a 2D plane, and thus a number of models are required for interpolating a full resistivity profile, which requires values for several depths. The 3D technique uses a single model that can account for semivariance in three axes, and thus interpolate values in a 3 dimensional space. A comparative study evaluating the performance of both techniques could allow for further improvements in the overall process of bedrock modelling from AEM data.

7.5 Intuitive Visual Representations

The visualizations of PIs in this thesis do not present the probability distribution within any prediction interval. Rather, it merely presents an interval, with no notion of the distribution within that interval. This statement is not entirely true, however, for the two ensembled approaches, where the error distribution is assumed gaussian, such that the center of the interval is always the distribution's mean. Nonetheless, Being able to manually inspect and analyze such intervals may also be highly beneficial for further analysis. Different types of color coding could let manual interpreters quickly understand how the probability is distributed, and allow for a more thorough analysis. A challenge occurs where displaying overlaying information may interfere with-, or hide underlying information which may also be of importance for the analysis.

Moreover, a feasibility study of probability distribution visualization of the concept for quantile regression could provide a set of its own challenges. Since the distribution is never assumed any type, e.g. gaussian, it is much more difficult to visualize its inner probability distribution.

A thorough investigation into visual representations of probability distributions in DTB predictions could further empower the human analyst, and improve the process of creating bedrock model deliverables.

Bibliography

[1] Anschütz, H., Bazin, S., Pfaffhuber, A. A., Sep 2015. Towards Using AEM for Sensitive Clay Mapping - A Case Study from Norway. First European Airborne Electromagnetics Conference.

[2] Anschütz, H., Christensen, C., Pfaffhuber, A. A., Sep 2014. Quantitative Depth to Bedrock Extraction from AEM Data. Near Surface Geoscience 2014 - 20th European Meeting of Environmental and Engineering Geophysics.

[3] Anschütz, H., Pfaffhuber, A. A., Auken, E., Pedersen, J., Schamper, C., Sagbakken, A., Effers, F., Oct 2013. Increasing geotechnical design efficiency by virtue of HTEM sediment mapping. 6th International AEM Conference & Exhibition.

[4] Anschütz, H., Pfaffhuber, A. A., Domaas, U., Rosenvold, B. S., Dec 2015. Combined airborne and ground geophysics as a first phase towards a landslide warning system - A Norwegian case study. Österreichische Ingenieur- und Architekten-Zeitschrift.

[5] Anschütz, H., Vöge, M., Lysdahl, A. K., Bazin, S., Sauvin, G., Pfaffhuber, A. A., Berggren, A. L., 2017. From Manual to Automatic AEM Bedrock Mapping. Journal of Environmental & Engineering Geophysics 22 (1), 35–49.

[6] Bane NOR SF, Apr 2018. Plan description with impact assessment - Regulation plan with impact assessment (IU) The joint project Ringeriksbanen and E16 Høgkastet - Hønefoss. https://www.banenor.no/globalassets/

`documents/prosjekter/ringeriksbanen-og-e16/reguleringsplan/horingsdokumenter-reguleringsplan/planbeskrivelse-med-konsekvensutredning.pdf`, (Accessed Sep 4, 2018.

[7] By, T. L., 1989. Crosshole seismics including geotomography for investigation of foundations. Norwegian Geotechnical Institute.

[8] Chollet, F., et al., 2015. Keras. `https://keras.io`, (Accessed Oct 3, 2018).

[9] Christensen, C. W., Pfaffhuber, A. A., Anschütz, H., Smaavik, T. F., 2015. Combining airborne electromagnetic and geotechnical data for automated depth to bedrock tracking. Journal of Applied Geophysics 119, 178–191.

[10] Christensen, C. W., Pfaffhuber, A. A., Anschütz, H., Smaavik, T. F., 2015. Regional geotechnical railway corridor mapping using airborne electromagnetics. Geotechnical and Geophysical Site Characterisation 5, 178–191.

[11] De Veaux, R., Schumi, J., Schweinsberg, J., Shellington, D., Ungar, L., Nov 1998. Prediction Intervals for Neural Networks Via Nonlinear Regression. Technometrics 40.

[12] Gal, Y., 2016. Uncertainty in deep learning. Ph.D. thesis, University of Cambridge.

[13] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Available online at `http://www.deeplearningbook.org` (Accessed Nov 12, 2018).

[14] Gooijer, J. G. D., Hyndman, R. J., 2006. 25 years of time series forecasting. International Journal of Forecasting 22 (3), 443–473.

[15] Gulbrandsen, M. L., Bach, T., Cordua, K. S., Hansen, T. M., Feb 2015. Localized Smart Interpretation - a data driven semi-automatic geological modelling method. ASEG Extended Abstracts 2015 (1).

[16] Haugen, E., Degago, S. A., Kirkevollen, O. V., Nigussie, D., Yu, X., May 2016. A preliminary attempt towards soil classification chart from total sounding. Challenges in Nordic Geotechnic.

[17] Heskes, T., 1996. Practical confidence and prediction intervals. In: Proceedings of the 9th International Conference on Neural Information Processing Systems. NIPS'96. MIT Press, Cambridge, MA, USA, pp. 176–182.
URL http://dl.acm.org/citation.cfm?id=2998981.2999006

[18] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. CoRR abs/1207.0580.
URL http://arxiv.org/abs/1207.0580

[19] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q., 2017. Densely connected convolutional networks. In: CVPR. Vol. 1. p. 3.

[20] Jones, E., Oliphant, T., Peterson, P., et al., 2001. SciPy: Open source scientific tools for Python. http://www.scipy.org/, (Accessed Dec 9, 2018).

[21] Jørgensen, F., Møller, R. R., Nebel, L., Jensen, N.-P., Christiansen, A. V., Sandersen, P. B. E., Dec 2013. A method for cognitive 3d geological voxel modelling of aem data. Bulletin of Engineering Geology and the Environment 72 (3-4), 421–432.

[22] Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. CoRR abs/1412.6980.
URL http://arxiv.org/abs/1412.6980

[23] Koenker, R., Hallock, K. F., 2001. Quantile regression. Journal of Economic Perspectives 15 (4), 143–156.

[24] Krzywinski, M., Altman, N., 2013. Points of significance: Importance of being uncertain. Nature Methods 10 (9), 809–810.

[25] Lecun, Y., 2015. Deep learning & convolutional networks. 2015 IEEE Hot Chips 27 Symposium (HCS).

[26] Ley-Cooper, A., Gilfedder Ibrahimi Annetts, M., , C., 01 2015. Inversion of legacy airborne electromagnetic datasets to inform the hydrogeological understanding of the northern eyre peninsula, south australia.

[27] Lysdahl, A., Andresen, L., Vöge, M., 2018. Construction of bedrock topography from Airborne-EM data by Artificial Neural Network. Numerical Methods in Geotechnical Engineering IX: Proceedings of the 9th European Conference on Numerical Methods in Geotechnical Engineering (NUMGE).

[28] Murphy, B., 2017. Pykrige. (Accessed Dec 1 2018).
URL `https://pykrige.readthedocs.io/`

[29] NGI.no, Pfaffhuber, A. A., (Accessed Oct 17, 2018). AEM. `https://www.ngi.no/eng/Services/Technical-expertise-A-Z/Geophysics-remote-sensing-and-GIS/AEM`.

[30] Nix, D., Weigend, A., 1994. Estimating the mean and variance of the target probability distribution. Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN94).

[31] Nocedal, J., Wright, S. J., 2006. Numerical Optimization, 2nd Edition. Springer.

[32] Oldenborger, G., Logan, C., Hinton, M., Pugin, A.-M., Sapia, V., Sharpe, D., Russell, H., May 2016. Bedrock mapping of buried valley networks using seismic reflection and airborne electromagnetic data. Journal of Applied Geophysics 128, 191–201.

[33] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830.

[34] Peterson, J., May 2013. Reality and the sacred. `https://youtu.be/2c3m0tt5KcE`, (Accessed Oct 12 2018).

[35] Pfaffhuber, A. A., Lysdahl, A. O., Sørmo, E., Bazin, S., Skurdal, G. H., Thomassen, T., Anschütz, H., Scheibz, J., Aug 2017. Delineating hazardous material without touching. First Break.

[36] Pfaffhuber, A. A., Persson, L., Lysdahl, A. O., Kåsin, K., Anschütz, H., Bastani, M.,

Bazin, S., Löfroth, H., Aug 2017. Integrated scanning for quick clay with aem and ground-based investigations. First Break.

[37] Pyrcz, M., Deutsch, C., Jan 2003. The Whole Story on the Hole Effect. Geostatistical Association of Australasia, Newsletter 18.

[38] Sapia, V., Oldenborger, G. A., Jørgensen, F., Pugin, A. J.-M., Marchetti, M., Viezzoli, A., 2015. 3d modeling of buried valley geology using airborne electromagnetic data. Interpretation 3 (4).

[39] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15, 1929–1958, (Accessed Jan 5, 2019).
URL `http://jmlr.org/papers/v15/srivastava14a.html`

[40] Toews, M. W., Apr 2007. Standard Deviation Diagram. `https://commons.wikimedia.org/wiki/File:Standard_deviation_diagram.svg`, (Accessed Nov 26, 2018) Published under the Creative Commons Attribution 2.5 Generic license. No changes were made to the image.

[41] United States Geological Survey, Nov 2016. Airborne Electromagnetics & Seismic Imaging FAQ. `https://volcanoes.usgs.gov/vsc/file_mngr/file-160/Nov16YVOFAQ.pdf`, (Accessed Dec 6, 2018).

Appendix A

7.6 Master Thesis Proposal

This section presents the master thesis proposal as presented to the students of NTNU during the summer of 2018.

Automated interpretation of depth to bedrock from airborne electromagnetic data using machine learning techniques

Airborne Electromagnetics (AEM) has been used frequently in the past years to investigate ground properties for planning and optimization of large road or railroad projects in Norway. For each measurement along the AEM flight line, a resistivity vs. depth profile is acquired, which provides information about the geological structure of the underground. In this project, the goal is to identify the boundary between overburden and bedrock. Bedrock usually has a rather high resistivity, while most (but not all) layers in the overburden have lower resistivity. The relation between resistivity and geological material is complex and non-unique. Therefore, converting these profiles into useful geotechnical information requires a thorough interpretation and especially for large surveys with several 100km of flight lines, this is a time consuming task, which potentially is biased by the person doing the interpretation. The automation of this task is based on training data provided by a geotechnical expert and/or by borehole data, where available. Depending on the necessary size of the training data set, the time-complexity of the interpretation can be reduced significantly and incorporation of borehole data can improve the reliability of the results. However, simple statistical methods are not able to provide an accurate interpretation in complex geological settings. In this project, potential deep learning pattern recognition techniques should be identified and the most promising techniques will be implemented in the existing python framework. Performance of the implemented techniques will be assessed on real data sets covering a range of geological structures. The project will be suitable for one to two students. This proposal relates to the NFR project ”Revolutionizing geotechnical site investigations for large infrastructure projects”, where NGI, Skytem, BGC, JDSI, Statens vegvesen and BANE NOR are partners.

Co-supervisor: Dr. Andreas Aspmo Pfaffhuber, NGI

Appendix B

7.7 Case Visualizations for Method Comparison

The following sections present the visualizations that were used for the visual analysis of the methods as described in Section 5.1.1. The standard MLP regressor is displayed by the black line, while the proposed CNN approach is displayed by the white line.

Case 1, 2, 3 and 4 relates to the highway construction project dataset described in Section 2.5, while Case 5 uses the more recent highway and railroad project (Ringeriksbanen & E16).

7.7.1 Case 1

The following visualizations are a single flight-line, where the training data consists of manually assigned depth labels for resistivity profiles within the flight-line.

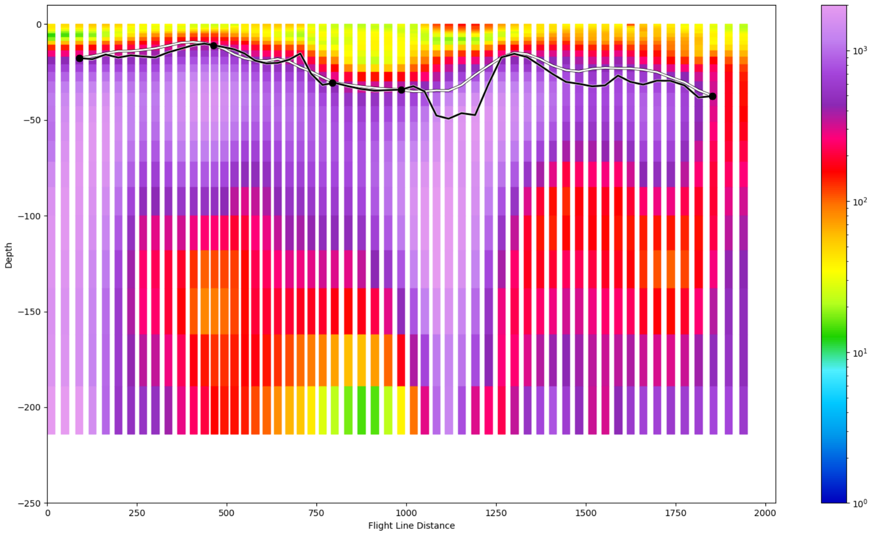


Figure 7.1: Case 1 with 5 labeled training data points

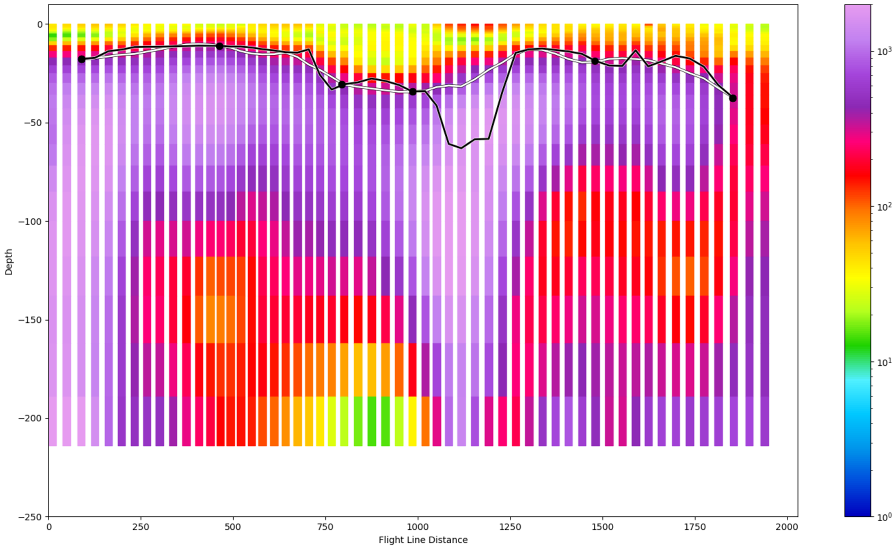


Figure 7.2: Case 1 with 6 labeled training data points

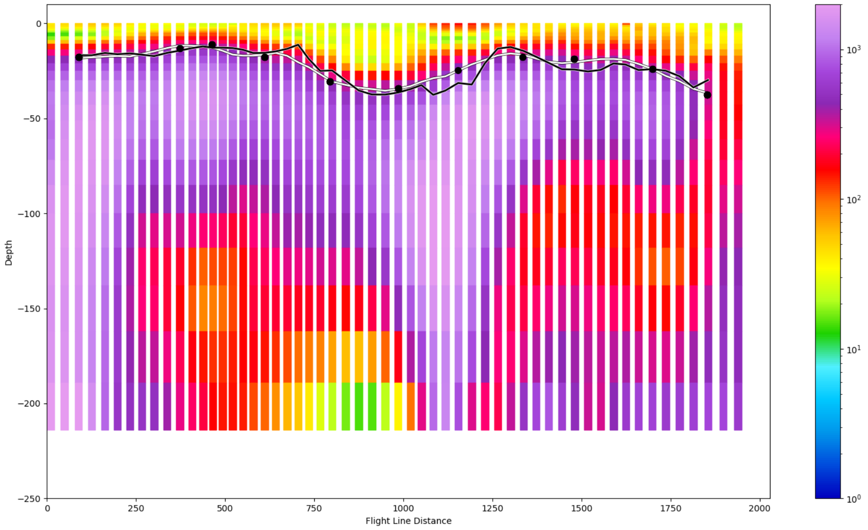


Figure 7.3: Case 1 with 11 labeled training data points

7.7.2 Case 2

The following visualizations are randomly selected flight-lines, where the training data consists of manually assigned depth labels for resistivity profiles within the flight-line. This case is vastly more complex than Case 1.

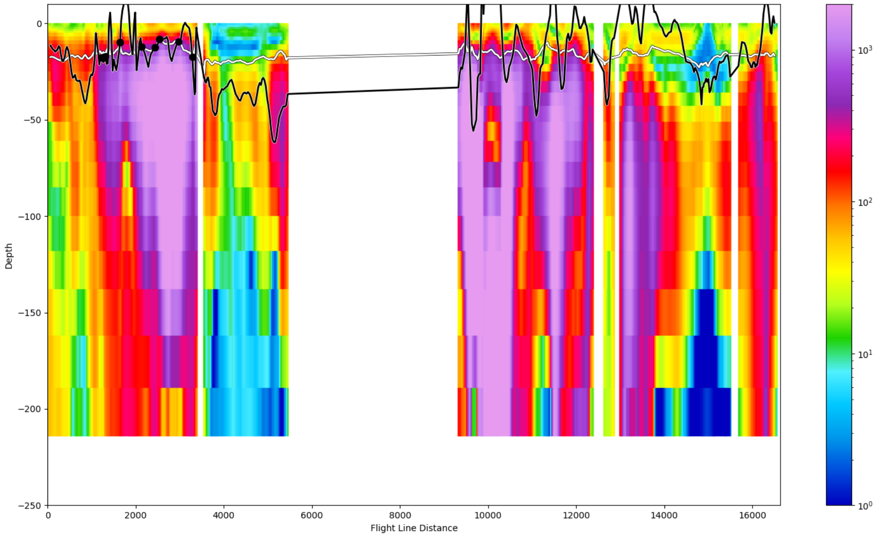


Figure 7.4: Case 2 with 8 labeled training data points

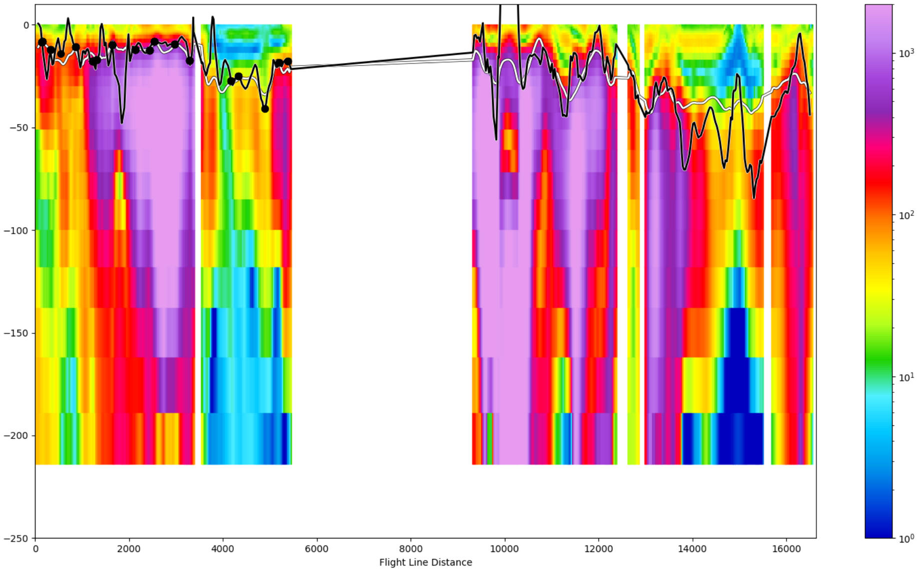


Figure 7.5: Case 2 with 20 labeled training data points

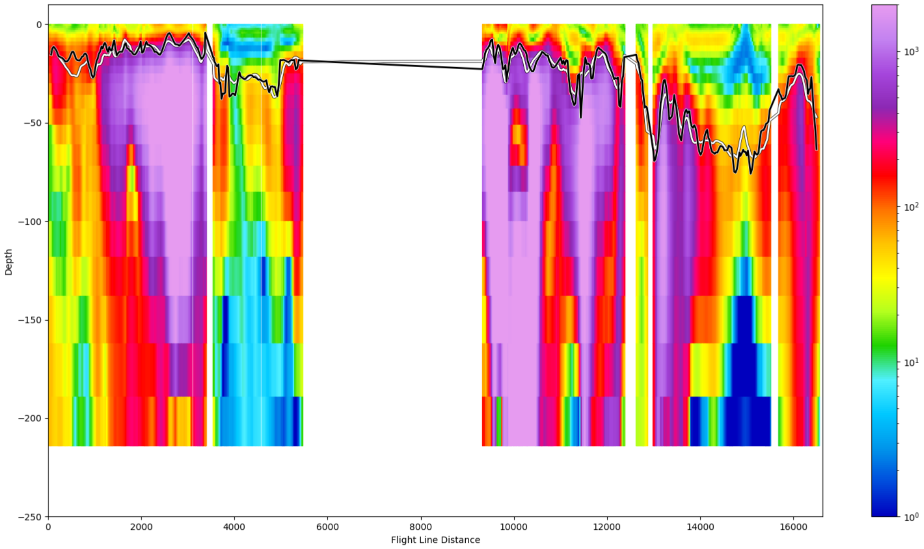


Figure 7.6: Case 2 with 130 labeled training data points, sparsely distributed over the flight-line (Training dots removed for clarity)

7.7.3 Case 3

The following visualizations are randomly selected flight-lines, where the training data consists of interpolated resistivity profiles and depth labels from TS boreholes.

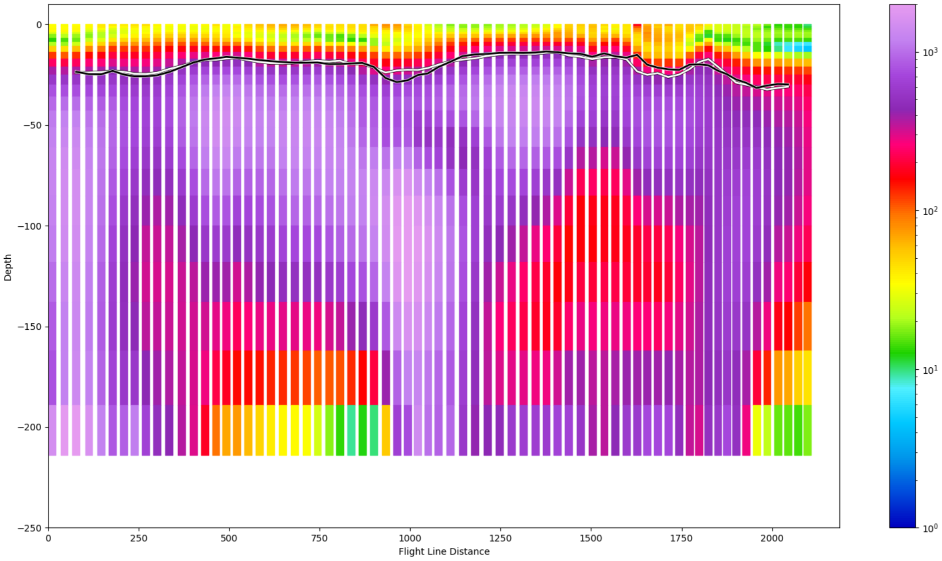


Figure 7.7: Flight-line visualization with training data from interpolated resistivity profiles and TS boreholes

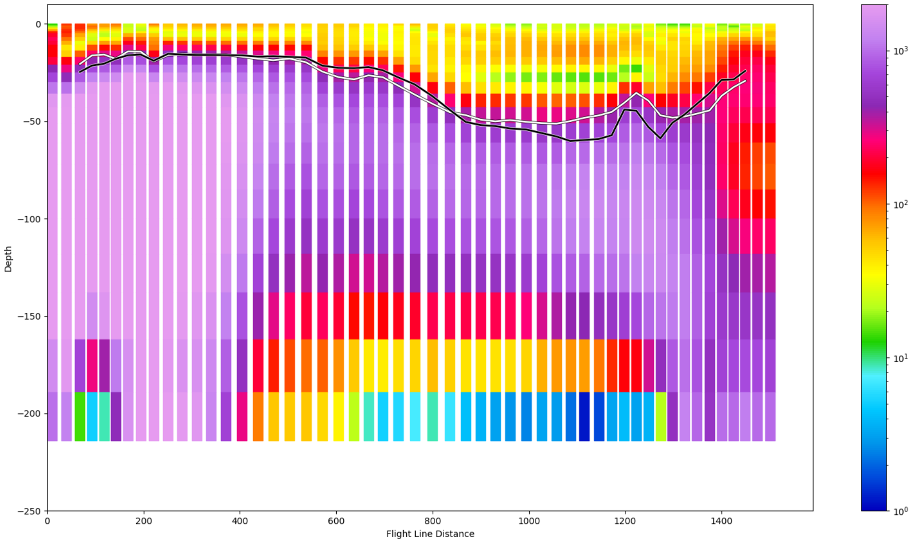


Figure 7.8: Flight-line visualization with training data from interpolated resistivity profiles and TS boreholes

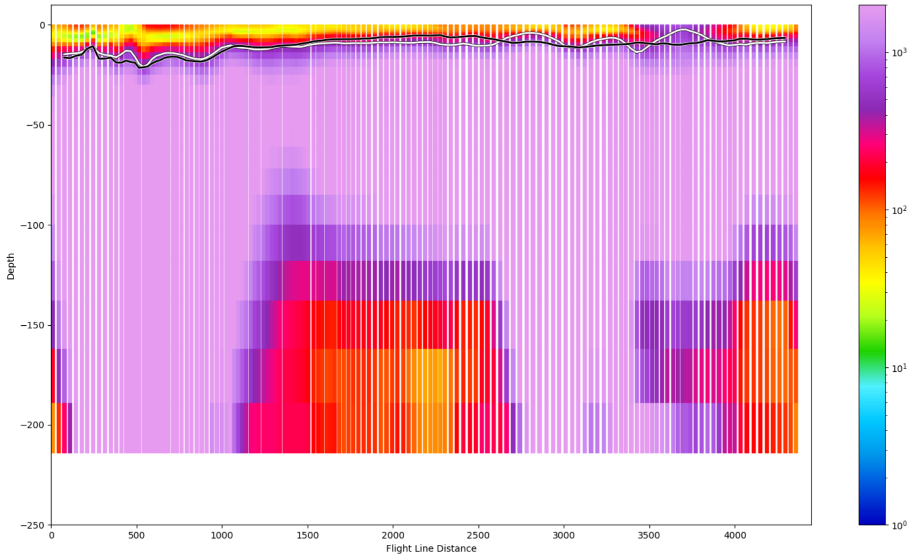


Figure 7.9: Flight-line visualization with training data from interpolated resistivity profiles and TS boreholes

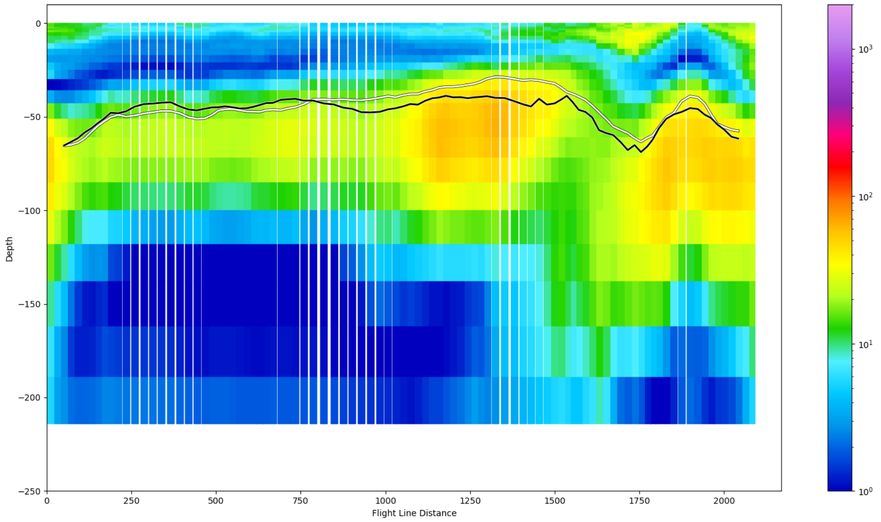


Figure 7.10: Flight-line visualization with training data from interpolated resistivity profiles and TS boreholes

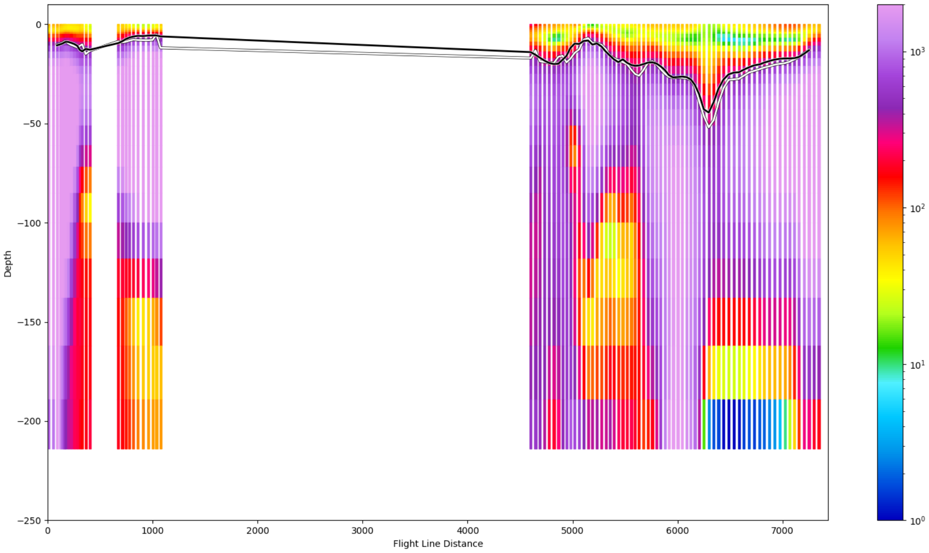


Figure 7.11: Flight-line visualization with training data from interpolated resistivity profiles and TS boreholes

7.7.4 Case 4

The following visualizations are randomly selected flight-lines, where the training data consists of interpolated resistivity profiles and depth labels from all boreholes.

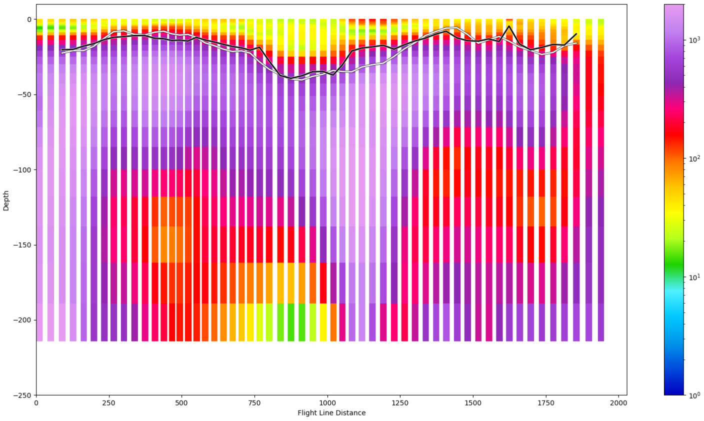


Figure 7.12: Flight-line visualization with training data from interpolated resistivity profiles and all boreholes

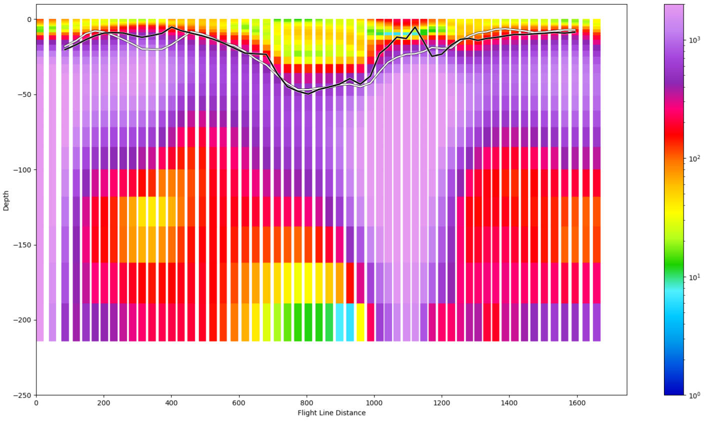


Figure 7.13: Flight-line visualization with training data from interpolated resistivity profiles and all boreholes

7.7.5 Case 5

The following visualization presents a randomly selected flight-line from the more recent highway and railroad project (Ringeriksbanen & E16). The training data consists of interpolated resistivity profiles and depth labels from TS boreholes within the survey area.

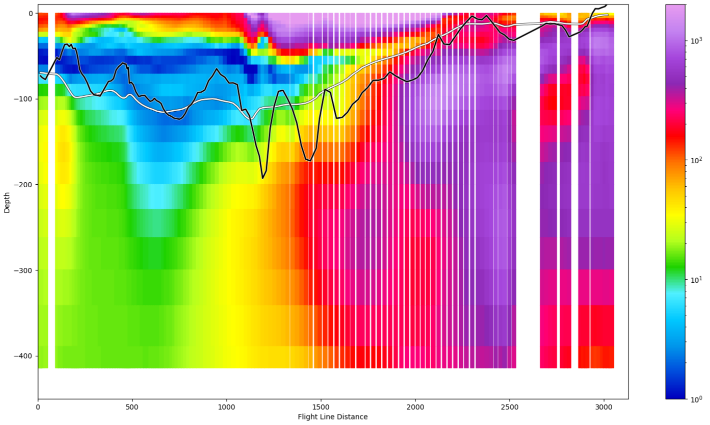


Figure 7.14: Flight-line visualization with training data from interpolated resistivity profiles and TS boreholes

Appendix C

7.8 Kriging and Boreholes

The training data for the CNN approach was derived using kriging for locations surrounding boreholes. As mentioned in Section 2.5.1, different types of boreholes present varying degrees of certainty in their depth labels. Moreover are the known resistivity profiles of varying certainty degree, with uncertainty stemming from the soundings and the inversion process.

Figure 7.15 shows a sample of kriged resistivity profiles with labeled depth marked as dots where the depths are provided by TS boreholes. Figure 7.16 presents a similar visualization, although, in this chart, the depths are provided by the less certain RPS boreholes.

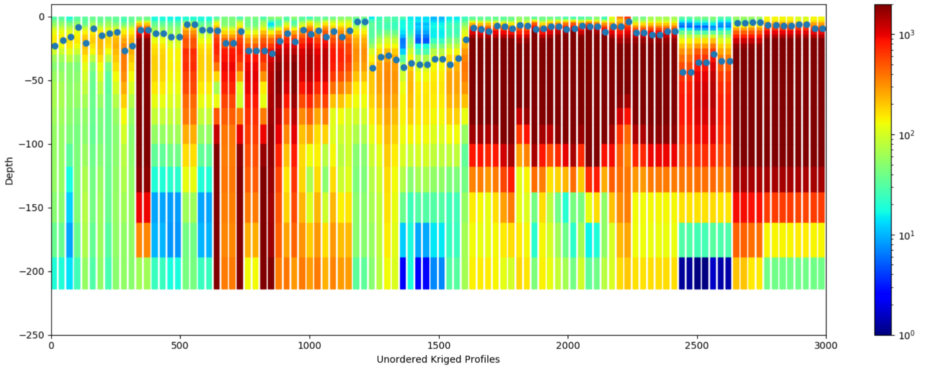


Figure 7.15: Sample of training data constructed from kriging of depth locations provided by TS boreholes

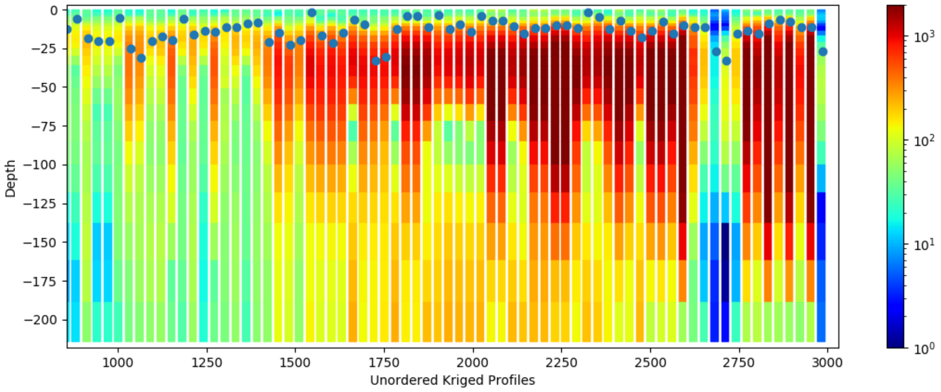


Figure 7.16: Sample of training data constructed from kriging of depth locations provided by RPS boreholes

Appendix D

7.9 Kriging & Interpolation

This appendix contains relevant resources for the Kriging and interpolation.

Figure 7.17 presents each of the semi-variograms that was used for the Kriging of each layer in the interpolation of the resistivity profiles. The X-axis represents the distance while the Y-axis represents the semi-variance. The semi-variograms are cut off at 5000 distance meters, as any information beyond this point was deemed irrelevant for the resulting models. This is not to say that there will never exist any reasonable correlation between any two points located with 5000 meters between them, which may often be the case for geotechnical properties. However, for the exact case of DTB levels in the geological areas related to the datasets discussed in this thesis, inclusion of further distanced pairwise points gave little valuable additional data for the model fitting.

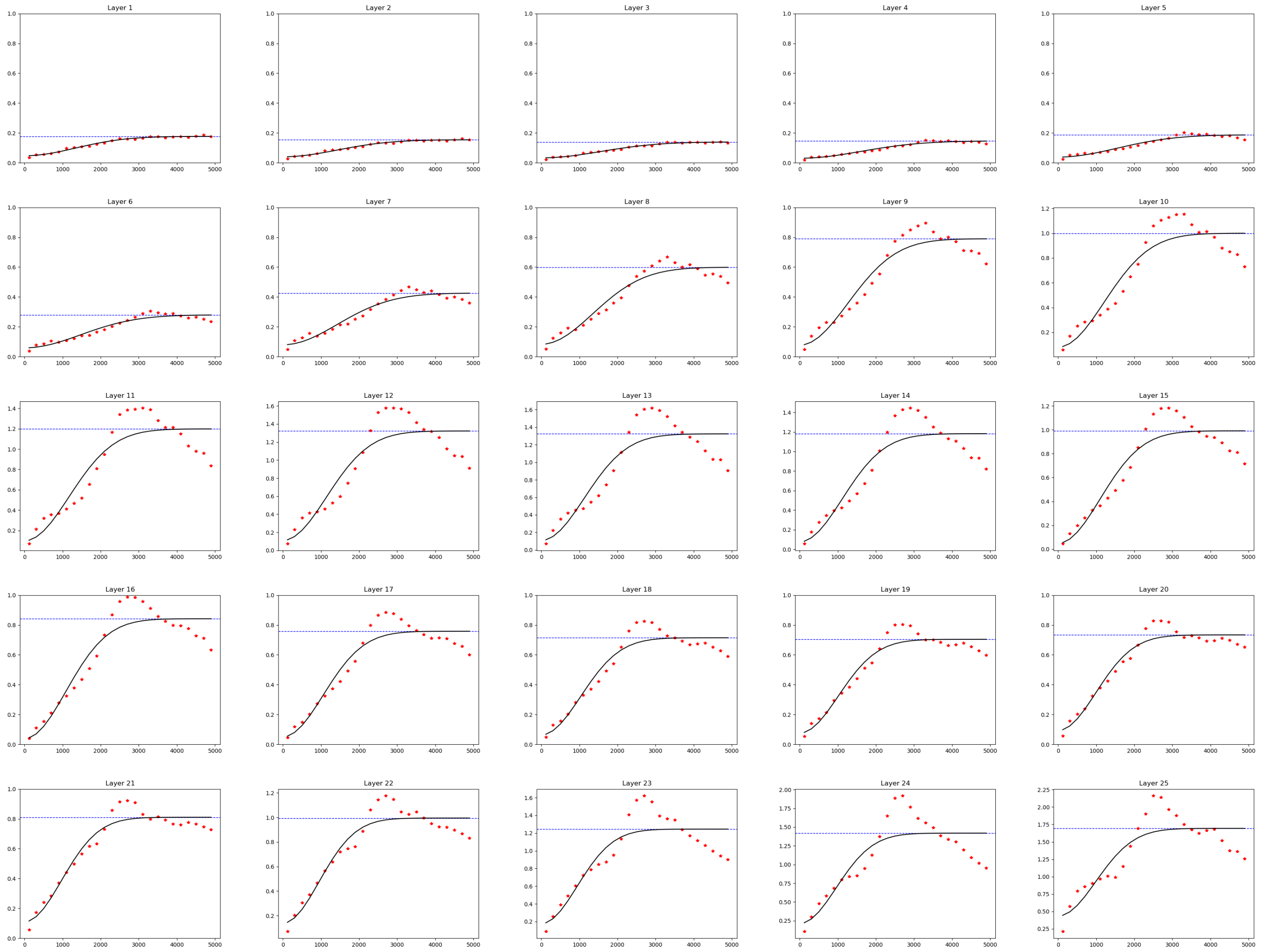


Figure 7.17: Semi-Variograms for each Layer

Simon Markus Hoff Olsen

Automated DTB Interpretation from AEM data

NTNU
Norwegian University of
Science and Technology