Norwegian University
of Life Sciences

Master's Thesis 2023   30 ECTS
Faculty of Science and Technology

Spatial Estimation of Depth to Bedrock using Borehole Data: A Gaussian Process Framework

Marisha Gnanaseelan
MSc Data Science

Preface

This thesis marks the end of my five-year master's degree in data science at the Norwegian University of Life Sciences (NMBU). The topic was introduced by Nils-Otto Kitterød, associate professor at the Faculty of Environmental Sciences and Natural Resource Management at NMBU. I began working on this thesis in January 2023, and the journey has been challenging yet rewarding, allowing me to acquire valuable knowledge and skills.

I would like to express my deepest gratitude to everyone who has helped guide me through the various stages of writing this thesis. First and foremost, I would like to thank my supervisors: associate professor Nils-Otto Kitterød, Erlend Briseid Storrøsten from the Norwegian Geotechnical Institute (NGI), Dagrun Aarsten from 3D Analyse, and assistant professor Ivar Maalen-Johansen, for their guidance and support throughout the course of this thesis. Their expertise and advice have been invaluable in shaping my research, and their unwavering encouragement has been a constant source of motivation. I am also grateful to Asgeir Olaf Kydland Lysdahl from NGI, for his valuable feedback and insights which have greatly contributed to the quality of my work. Additionally, I would like to thank Amir Arfan for proofreading and providing feedback on my work.

Finally, I want to thank my fellow classmates at NMBU as well as my close friends, and family for their unwavering support and encouragement. Their love, understanding, and patience have been the bedrock of my accomplishments, and I couldn't have reached this point without them.

Marisha Gnanaseelan
Ås, May 15, 2023

Abstract

Depth to Bedrock (DTB) is a critical parameter in several fields of study, including geology, hydrology, soil sciences, and civil engineering. However, obtaining this parameter through near-surface geophysical methods can be challenging and expensive, particularly in difficult terrain. Fortunately, high-quality borehole data from previous geotechnical investigations can be used to estimate the DTB in areas where no boreholes have yet been created.

This thesis presents a machine learning framework for estimating the DTB value in areas of interest using Gaussian Process models. The performance of different kernel functions, including Radial Basis Function (RBF), Matérn 3/2 kernels, and combined linear and RBF kernels, is evaluated, along with the impact of implementing anisotropy in the models.

The results show that the Matérn 3/2 kernel with anisotropic implementation performs the best in estimating DTB. However, challenges in hyperparameter optimization, non-Gaussian target variables, and model selection are highlighted, and further investigation into these areas is recommended. The framework presented here provides practical implications for geotechnical engineering. Further, it provides a basis for future research in this area, where the incorporation of additional geological and remotely sensed data could potentially improve the quality of DTB estimation.

Contents

List of Abbreviations and Acronyms

$\mathbf{R}^2$ Coefficient of Determination

1D One dimensional

2D Two dimensional

3D Three dimensional

AdaGrad Adaptive Gradient Algorithm

Adam Adaptive Moment Estimation

AI Artificial Intelligence

CDF Cumulative Distribution Function

CPT Cone Penetration Test

CPU Central Processing Unit

CRPS Continuous Rank Probability Score

CV Cross-validation

DEM Digital Elevation Model

DSM Digital Surface Model

DTB Depth to Bedrock

DTM Digital Terrain Model

GIS Geographic Information Systems

GNSS Global Navigation Satellite System

GP Gaussian Process

GPR Gaussian Process for Regression

GPS Global Positioning Systems

IMU Inertial Measurement Unit

LiDAR Light Detection and Ranging

MCMC Markov Chain Monte Carlo

NGI Norwegian Geotechnical Institute

NGU Geological Survey of Norway

NLML Negative Log Marginal Likelihood

RAM Random-Access Memory

RBF Radial Basis Function

RMSE Root Mean Squared Error

RMSProp Root Mean Square Propagation

TS Total Sounding

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Bedrock is the solid rock that appears as outcrops on the earth's surface or concealed beneath numerous layers of soil [9]. Fields of study such as hydrology, geology, soil sciences, and civil engineering, use Depth to Bedrock (DTB) as a crucial parameter for, among other things, modelling ground-water flow and movement in chemicals and soils [45], landslide risk management [84], and construction of infrastructure [64]. Knowledge about DTB can be acquired from geotechnical investigations such as Total Sounding (TS), rotary pressure sounding, and other near-surface geophysical methods [69]. Despite providing data of high accuracy, tests of this type are economically costly, time-consuming, and challenging to execute in difficult terrains, thus resulting in limited information about DTB at various locations.

The relationship between the underlying bedrock topography and the overlying surface topography is often correlated with each other, especially in shallow areas such as glacial uplands and lowlands [65]. With the availability of high-quality remote sensing data and data acquired from geotechnical investigations, geostatistical methods can be applied to estimate DTB values in areas of interest.

Kriging [43] is a well-established spatial interpolation method that for the past decades has been widely used within the field of geostatistics. Through kriging, values at unsampled locations can be predicted by considering the spatial correlation between neighbouring sample points [21]. This technique also produces estimations of the uncertainty associated with the predictions, which can be important for decision-making in geotechnical engineering applications [95].

The fundamental form of kriging is computationally demanding when working with large datasets. Through the implementation of Gaussian Process (GP) models in machine learning, the kriging method can be scaled to handle larger datasets [77]. In machine learning, GPs can be adapted to solve classification, prediction, and regression problems [83], where Gaussian Process for Regression (GPR) [68] is a technique that is closely related to the kriging method.

GPR is a non-linear approach that uses a covariance function (or kernel), as its key component. The purpose of the kernel is to capture the underlying structure and relationship within the data, with different kernels being able to capture different patterns of correlation, such as periodicity, smoothness, and non-stationarity [18]. The correlation structure of data can exhibit anisotropy, indicating that the underlying pattern varies depending on the direction of measurement [38]. To model anisotropy in GPs, an anisotropic kernel can be used to define the correlation function in multiple directions. By defining the covariance between pairs of data points, the kernel enables the GPR to infer the underlying pattern of the data and make predictions for new input values [60].

The purpose of this thesis is to study how GPR and different kernel functions can be used to estimate the DTB values at areas of interest. By applying methods from geostatistics and machine learning, we aim to construct a framework that can be used to help engineers achieve better knowledge about the DTB parameter and be less dependent on manual testing.

1.2 Related Works

There are currently several studies available on using geostatistical methods to estimate parameters in geotechnical data. Li et al. [47] use kriging to identify the soil profile at unsampled locations. They use data from Cone Penetration Test (CPT) and calculate the measurement uncertainties related to the soil classification from the tests. The method they developed works well in a 2D plane, and the performance of the method is evaluated based on how well the results are consistent with the CPTs. Li, Hicks, and Vardon [48] also extended their work so the method could be applied to 3D data. However, this method did not take changes in spatial dependence in different directions, i.e. anisotropy, into account.

A study conducted by Shen and Gelfand [79] explored the potential benefits of using anisotropic kernel functions within a Bayesian framework for modelling spatial correlation. The study focused on the Matérn kernel and exponential kernel functions and incorporated anisotropy into the models using linear transformations of coordinates. The objective was to compare the predictive performance of anisotropic models to isotropic models in geostatistical settings. The results showed that anisotropic models outperformed isotropic models when the data significantly departed from isotropy, the spatial variance was much greater than the pure error variance, and the sample size was fairly large. Both anisotropic-generated data (100 and 500 samples) and real data (546 samples) were used to test the models. Given the increasing amount of data generated today, more computational resources and efficient methods for modelling and analyzing larger datasets are required. In this context, incorporating frameworks that use machine learning can be a valuable approach.

Machine learning can handle large datasets and provide faster computation times compared to traditional statistical methods. There are several studies that have applied machine learning methods for spatial data analysis, including estimating DTB. For instance, Shangguan et al. [78] applied two tree-based ensemble methods, namely Random Forests and Gradient Boosting Tree, to predict the global DTB at a spatial resolution of 250 meters. They used a combination of soil profile and borehole data, as well as additional features derived from remote sensing, such as lithology maps, hydrological and morphological derivatives based on Digital Elevation Model (DEM), and satellite-based sensor data, among others. The resulting models achieved moderate performance, explaining 59% of the variance for absolute DTB and 34% for censored DTB.

A later study by Yan et al. [94] presents a method to produce high-resolution (100 meters) DTB maps of China. The maps were created using two sets of data, one based on ensemble predictions from Random Forests and Gradient Boosting Tree models and the other based on Quantile Regression Forests. Although the results showed higher accuracy compared to the maps in Shangguan et al. [78], the accuracy of the models was just above 50%, indicating that they are performing only slightly better than random guessing.

In general, while machine learning models can provide faster computation times and handle large datasets, it can be challenging to understand why the model makes the predictions it does. This is particularly true when it comes to uncertainty quantification, where standard machine learning models may not provide clear indications of how confident the model is in each of its predictions [85]. Therefore, more advanced techniques, such as Gaussian Processes (GPs) combined with machine learning frameworks, may be required to better understand and quantify the uncertainty associated with predictions.

1.3 Approach

The primary goal of this thesis is to develop a framework that applies GP models and explore the use of different kernel functions to estimate the DTB at unsampled locations. To achieve this objective, we will first study the theoretical background of geotechnics, geostatistics, and machine learning.

We will then apply GP models using Python [89] and TensorFlow [52], to two datasets from different locations: Oslo East and a highway (E18) in Bærum. The datasets exhibit varying degrees of anisotropy in their structure. To investigate the ability of the models to detect anisotropy in the data, we will test the models using both modified anisotropic kernels and standard (isotropic) kernels.

The models will be trained on a subset of the datasets and then tested on the remaining subset. To assess their performance, we will compare the predicted values with the actual values measured in the interpolated field. Finally, we will apply the models to a larger field that includes the measured observations from both datasets to determine their generalization beyond the specific locations used for training and testing.

The results of this project will provide insights into the use of GPs for predicting DTB values in different locations with varying degrees of anisotropy. It will also demonstrate the importance of selecting appropriate kernel functions for modelling anisotropic data. Ultimately, this framework can be utilized for estimating other geotechnical parameters at unsampled locations using other geotechnical datasets, which can be valuable for geotechnical engineering and construction projects.

1.4 Objectives

The main objectives of this thesis are:

(i) Develop a framework that combines geostatistical methods and machine learning to estimate DTB values in areas of interest, using data from geotechnical investigations.

(ii) Investigate the use of GPs in combination with different kernel functions to estimate DTB.

(iii) Evaluate the performance of the proposed framework in terms of accuracy, efficiency, and computational scalability for large datasets.

(iv) Assess the potential benefits of incorporating anisotropic kernel functions within the framework for modelling spatial correlation in the models.

(v) Investigate the impact of different kernel functions on the performance of the models and select the most suitable kernel function(s) for estimating DTB values.

1.5 Thesis Structure

The thesis is divided into seven chapters, starting with Chapter 1 which provides an introduction to the thesis' background, related works, and objectives. Chapter 2 presents the theoretical concepts and principles of the methodology employed in this study. In Chapter 3, the description, exploration and analysis of the data used in this study are described. Chapter 4 explains the research methods and techniques used to address the objectives of this thesis. The findings and outcomes of the study are presented in Chapter 5. Chapter 6 discusses the results in the context of the research objectives and presents possible implications for future research. Finally, Chapter 7 summarizes the main findings, provides recommendations for future work, and concludes the thesis. Appendix A includes the source code and software used to construct the framework, Appendix B contains all interpolation results for the Oslo East dataset, and Appendix C contains all interpolation results for the Bærum E18 dataset.

Chapter 2

Theory

This chapter covers several important topics that form the basis of the methods used in this study. The chapter begins with an overview of geotechnics and the significance of the Depth to Bedrock parameter in geotechnical investigations. Then, remote sensing and its various applications in geology and geotechnics are discussed. The chapter further delves into the theory behind the implemented machine learning models, particularly Gaussian Processes. Lastly, we discuss the use of Geographic Information Systems in integrating geotechnical data.

2.1 Geotechnics

Geotechnics is a field within civil engineering that studies the behaviour of natural materials found on or near the earth's surface. Natural materials within geotechnical engineering usually include soil and rock. The primary distinction between soil and rock is that soil is defined as loose matter consisting of a mixture of organic, mineral and other materials. In contrast, rock is mainly composed of mineral matter which leads to rock having strong internal cohesive and molecular bonds that bind the mineral grains more effectively together than those found in soils [31]. Soils are usually a result of geologic processes like weathering and erosion. The weathering process breaks down or dissolves the rock, while erosion is the process of transporting the decomposed material. Physical, biological, and chemical processes that are affected by factors such as water, wind, and climate act either by themselves or together to form the aforementioned processes [19].

Due to geologic soil-forming processes, soils have different characteristics that can be classified into layers or *horizons* [81]. The master soil horizons are illustrated in Figure. 2.1.1. It is important to notice that the soil horizons shown in the figure may not always be present in every type of soil, and even when present, they may not always occur in the same order [74].

The result of vertically stacking various soil horizons together after conducting geotechnical investigations is called a *soil profile*. It is essential to recognize the significance of soil profiles from an engineering standpoint, given that all infrastructure rest on the soil. Knowing the strength and properties of the soil at a site and relevant areas for an intended project ensures both safe and stable work practice [31].

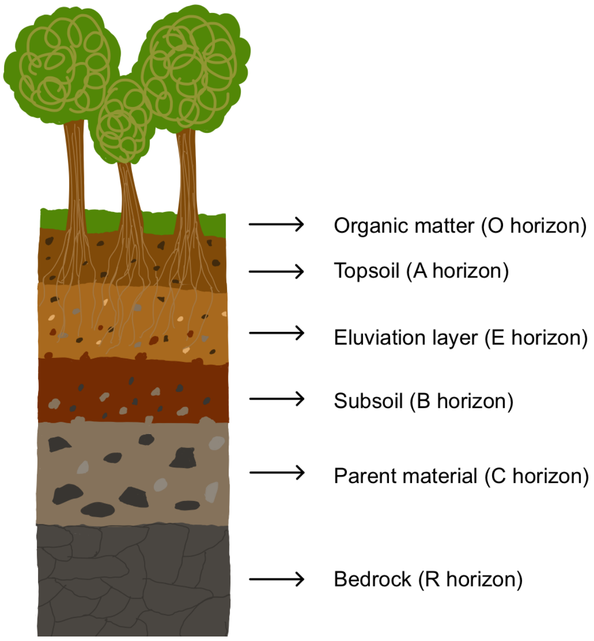

Organic matter (O horizon)
Topsoil (A horizon)
Eluviation layer (E horizon)
Subsoil (B horizon)
Parent material (C horizon)
Bedrock (R horizon)

Figure. 2.1.1: *A visual representation of common master soil horizons located beneath the earth's surface. The horizons in the illustrated soil profile are arranged in six distinct layers. The topmost layer, the O horizon, consists mainly of organic materials. The A horizon or topsoil is rich in dark, organic material called humus and is often highly weathered. The light-coloured layer, the E horizon, contains mineral particles removed by water or other processes. The subsoil, the B horizon, contains minerals from the horizons above and below. The parent material, or the C horizon, consists of poorly or unweathered geologic material. Finally, the hard unweathered parent rock is identified as the bedrock or R horizon [74].*

2.1.1 Geotechnical Definitions

Bedrock is defined as consolidated, i.e. solid, rock that exists beneath layers of unconsolidated geologic materials such as soil [9]. The uppermost layer of the bedrock is defined as the *bedrock surface*. The DTB is therefore measured as the distance from the terrain surface to the bedrock surface. Obtaining precise information about DTB is mainly done with geotechnical investigation methods, and the range of the depth can be anything from zero up to a hundred meters. Locations where DTB is measured as 0, are called *outcrops* and are defined as the part of the bedrock which is exposed at the earth's surface [3].

To determine geotechnical soil properties, various techniques are available, including the widely used Cone Penetration Test (CPT). However, CPT is not effective for soils containing gravel or larger particles, and it cannot be used for sampling bedrock [31]. In contrast, Total Sounding (TS) is an in-situ technique that combines the principles of rotary pressure sounding and rock control drilling. The process of TS involves penetrating drilling rods into the ground at a constant penetration and rotation rate until rocks or boulders are reached. When this happens, the drilling switches to rock control mode, increasing the rotation rate, and then switch back to rotary pressure mode when the penetration is through the soil layer with rocks. In the case that the drilling rod hits the bedrock surface, the drilling rod is bored approximately 3 meters into the bedrock to confirm that it is not just rocks or boulders. This results in data from TS to be highly accurate [29, 69].

2.1.2 Geomorphology

Soil-forming processes that occur at the surface or deep in the earth's crust modify the shape of the earth's surface. The scientific term for studying the origin, evolution, shape, and present landform of the earth's surface is known as geomorphology [93]. Topography, which is another term used in connection with describing the earth's surface is measured by the change in elevation across the earth's surface. There is currently a vast selection of measurement tools available for land surface investigations, which include Global Positioning Systems (GPS), Digital Elevation Model (DEM), and Light Detection and Ranging (LiDAR) [26].

2.2 Remote sensing

The use of remote sensing makes it possible to acquire information about objects or phenomena on the earth's surface from a distance. Remotely sensed data are collected from sensors that are aboard aircraft and satellites [76]. The sensors detect reflected and emitted radiation from Earth to collect data. These sensors can be divided into two categories: passive remote sensing sensors and active remote sensing sensors. The former measures radiation which is reflected or emitted from an external source of energy, such as sunlight, whereas the latter relies on an artificial source of energy, such as a laser beam [32]. Derived data from remote-sensing systems may differ based on the intended application of the data, where spatial resolution, spectral resolution, and temporal resolution, are parameters that specify the quality of the generated data [44]. The application of remote-sensing technology spans numerous fields such as meteorology, agriculture, climate change detection, government, environmental monitoring, mapping, and others [96].

2.2.1 LiDAR

Light Detection and Ranging (LiDAR) is a remote sensing technique that measures ranges (distances). Ranges are calculated by using the time it takes for an emitted laser signal to travel between the sensor and a target. LiDAR is an active remote-sensing sensor typically mounted on spaceborne, airborne, and ground-based platforms [41]. Since the positional- and terrain data relevant to this study is generally collected through airborne laser scanning, we will take a closer look at this technology.

Airborne LiDAR is a common technique for producing high-quality 3D digital representations like point clouds and has proven to be highly beneficial in terms of mapping larger areas. Significant components of airborne LiDAR systems consist of (i) an aircraft which is used to fly over the area of interest; (ii) a LiDAR sensor which continuously sends laser pulses between the terrain and aircraft; (iii) a Global Navigation Satellite System (GNSS) receiver on the aircraft that works with a ground-based GNSS station to record the precise location of the sensor; (iv) an Inertial Measurement Unit (IMU) sensor that measures the acceleration and rotation of the sensor; (v) and an onboard computer that collects the data from the LiDAR, GNSS, and IMU sensors [49].

2.2.2 Data Processing

An airborne laser scanning survey provides the following datasets: data from the GNSS ground station, navigation data from GNSS and IMU sensors on aircraft, and range measurements. Once the datasets are processed, point clouds with x, y, and z coordinates are interpolated to create the Digital Surface Model (DSM) and the Digital Terrain Model (DTM). The primary distinction between a DSM and a DTM is that the former includes details regarding features located above the terrain surface such as vegetation and human-made objects, whereas the latter simply provides information about the elevation of the underlying terrain. The generation of a DTM from a DSM requires classifying the points within the point cloud as either ground points or non-ground points, followed by the application of interpolation methods to fill in areas with missing data [76].

2.2.3 Høydedata

The Norwegian Mapping Authority (Kartverket) administer høydedata.no, which is an insight and download solution. The data is collected through different airborne laser scanning and aerial photography surveys. The available data can be downloaded as point clouds, DSM, and DTMs with various resolutions. In 2016 the National Detailed Elevation Model Project was started where the goal was to map the accurate height of every square meter of Norway. This project was finished in the summer of 2022. Areas of application for elevation data can be calculating flood zones and avalanche danger, planning infrastructure, analysing placement of wind turbines, mapping overgrowth, and uncovering hidden cultural heritage, such as grave mounds [39].

2.3 Geostatistics

According to Waldo Tobler, "Everything is related to everything else, but near things are more related than distant things" [87]. This phrase is considered Tobler's First Law of Geography and is the principle of spatial dependence (or autocorrelation). The effects of spatial autocorrelation are widespread in spatial data and can be exemplified by various sources. Examples include soil properties [30], air pollution concentration [72], and water quality parameters such as temperature, pH, and dissolved oxygen [80]. To better understand and predict parameters in such data, we need statistical methods that explicitly account for spatial autocorrelation.

Geostatistics is a branch of statistics that focuses on analyzing spatially distributed data, by providing a set of methods that can be used for modelling spatial autocorrelation, making spatial predictions, and quantifying uncertainty [22]. Professor Georges Matheron [53] is recognized for defining the principles and basic methodology in geostatistics, earning him the title of the founder of Geostatistics [2]. Geostatistics is often associated with the Kriging method, which was originally developed by Daniel G. Krige in the 1950s for mining applications, hence the name of the method [43]. Despite its origins in mining, kriging has been widely adopted in various disciplines for interpolating spatial data and generating predictions.

The subsequent sections in this Chapter (2.3) will provide an introduction to essential concepts and terminology in geostatistics.

2.3.1 Regionalized Variables and Random Functions

The concept of regionalized variables was introduced by Matheron in 1965 [53]. Standard statistical literature does not take spatial variability into consideration, which is why a variable that varies depending on location in space is said to be *regionalized.* At each point x in the domain D (i.e. an area of interest), we have a regionalized variable, denoted by $z(x)$, which represents the realization (or outcome) of a random function (or random field), $Z(x)$. The random function is a set of random variables which can be defined as $\{Z(x) : x \in D\}$. Note that we use capital Z to denote random variables and lowercase z to denote their realizations [91].

The following equation is defining a cumulative distribution function (CDF) for a random function $Z(x)$:

$$F_{x_1,\ldots,x_n}(z_1,\ldots,z_n) = P(Z(x_1) \leq z_1,\ldots,Z(x_n) \leq z_n) \tag{2.1}$$

Where,

$z_1,\ldots,z_n$ is a set of regionalized variable values

$x_1,\ldots,x_n \in D$ is a set of points in a given domain

$F_{x_1,\ldots,x_n}$ is a CDF

P is a probability function that measures the probability of the outcome of $Z(x_n)$

In simpler terms, Equation 2.1 allows us to calculate the probability of observing certain regionalized variable values at different locations in a dataset, based on the probability distribution of the underlying random variables, which helps us quantify the uncertainty in our estimations.

The statistical properties of a random function, such as its mean, variance, and correlation structure, can be characterized using techniques like variogram modelling and covariance functions, which will be explained further in the following sections.

2.3.2 Statistical Moments

Statistical moments refer to a set of mathematical measures that describe the characteristics of the probability distribution of a random function. If we have a random function $Z(x)$ at a point x, the moments are defined as the expectation E of $Z(x)$ [38].

The *first-order-moment* is the mean, denoted by $\mu(x)$, which is mathematically calculated as:

$$\mu(x) = E\{Z(x)\} \tag{2.2}$$

The first *second-order-moment* is the variance, dentoted by $\sigma^2(x)$, and is calculated as:

$$\sigma^2(x) = Var\{Z(x)\} = E\{(Z(x) - \mu(x))^2\} \tag{2.3}$$

The *second-order-moment* between $Z(x_1)$ and $Z(x_2)$ is the covariance, denoted by Cov, and is calculated as:

$$Cov(x_1, x_2) = E\{(Z(x_1) - \mu(x_1))(Z(x_2) - \mu(x_2))\} \tag{2.4}$$

The *second-order-moment* between the difference $Z(x_1) - Z(x_2)$ is the variogram, denoted by γ, and is calculated as:

$$2\gamma(h) = Var\{Z(x_1) - Z(x_2)\} \tag{2.5}$$

Where h is the Euclidian distance between x_1 and x_2. The semivariogram is half of the variogram and is defined as:

$$\gamma(h) = \frac{1}{2}Var\{Z(x_1) - Z(x_2)\} \tag{2.6}$$

2.3.3 Stationarity

Statistical characteristics of a random function (e.g. mean, variance, and other statistical moments) are said to be stationary if they remain constant over time and space for all points in a domain.

According to Chilès and Delfiner [11], there are two types of stationarity in geostatistics that are based on the first two statistical moments (i.e., mean and variance):

(i) The assumptions of *second-order stationarity* are that the mean and variance are constant in space. These assumptions can be formulated as:

$$\mu_{Z(x)} = \mu_{Z(x+h)} = \mu_Z \tag{2.7}$$

and

$$Var\{Z(x)\} = Var\{Z(x+h)\} = \sigma_Z^2 \tag{2.8}$$

By inserting Equation 2.7 and Equation 2.8 we can derive the following expression for the covariance:

$$C(h) = Cov\{Z(x), Z(x+h)\} = E\{Z(x)Z(x+h)\} - \mu_Z^2 \tag{2.9}$$

Where h is the *lag distance*, which is a vector that describes the distance separating a pair of points in the given space, and $C(h)$ is the *covariance function*. Note that if we define $h = 0$ we get the *nugget* $C(0)$ which equals to the variance σ^2 at lag distance 0:

$$C(0) = E\{(Z(x) - \mu_Z)^2\} = Var\{Z(x)\} = \sigma_Z^2, \quad \forall \;\; x, x+h \in D \tag{2.10}$$

(ii) The assumption of *intrinsic stationarity* is slightly weaker than the assumption of second-order stationarity. Intrinsic stationarity assumes that the statistical properties of a random variable do not depend on the location in space, but only on the distance between locations. In other words, if two points in space have the exact same distance between them, then their statistical properties should be the same:

$$Z(x+h) - Z(x) \tag{2.11}$$

Assuming the variance of the increment, Equation 2.11, corresponding to two different locations depends only on the vector h, we can derive the semivariogram, denoted by the function $\gamma(h)$:

$$\gamma(h) = \frac{1}{2} Var\{Z(x+h) - Z(x)\} = \frac{1}{2} E\{(Z(x+h) - Z(x))^2\}, \;\; \forall \;\; x, x+h \in D \tag{2.12}$$

2.3.4 Covariance Functions and Semivariogram Models

The relationship between the semivariogram and the covariance can be expressed mathematically using the formula [6]:

$$\gamma(h) = C(0) - C(h) \tag{2.13}$$

With this relation, we can estimate the semivariogram to study the spatial correlation within a dataset. This can then be transformed into a covariance function, which is necessary for the estimation of values across a random field (i.e. spatial interpolation).

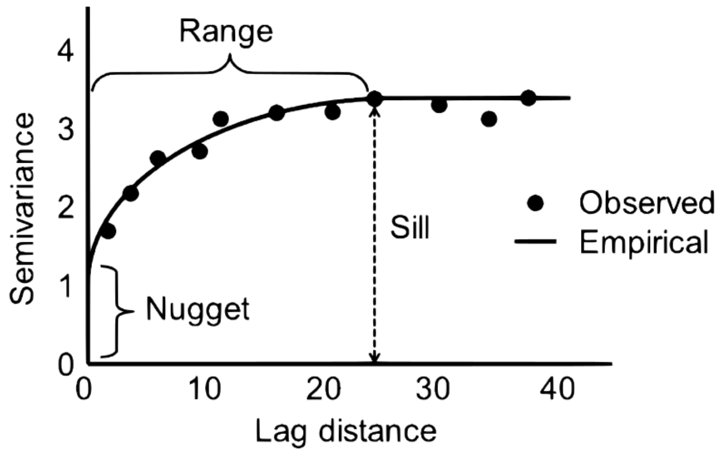


Figure. 2.3.1: *A typical example of a semivariogram showing different components. Illustration from [8] and under CC-BY-3.0 licence.*

The semivariogram can be visualized as a model, which can be derived from the covariance function. To obtain the semivariance, the differences between pairs of sample points in a real-world dataset are calculated and we obtain an empirical estimate. To ensure the validity of an interpolation method, it is necessary to calculate an empirical semivariogram model from a chosen theoretical semivariogram model and the empirical estimate. The shape of a semivariogram model is characterized by specific parameters, including range, sill, and nugget, and is illustrated in Figure. 2.3.1. The *range* is the distance where the data points are no longer correlated to each other, the *nugget* represents the discontinuity in the semivariogram at lag distance 0, and the *sill* reflects the maximum dissimilarity between the data points at a distance [4].

It is essential that a semivariogram model is *positive definite* [37], meaning that the covariance between any two points cannot be negative, which ensures that the variance of any linear combination is positive [5]. There are different types of theoretical models that already exist. The most common type of model is *isotropic*, where the shape of the semivariogram is symmetric in all directions. *Anisotropic* models, on the other hand, allow for variations in the directionality of the semivariogram. We will now take a closer look at commonly used theoretical models, and the concept of isotropy and anisotropy.

Isotropy

A random function is said to be isotropic if it only depends on the length of the directional vector h [6]. In simpler terms, it means that the semivariogram is only dependent on the distance between two points and not on their orientation. Common isotropic variogram models include:

(i) Spherical:

$$\gamma(h) = \begin{cases} 0, & |h| = 0 \\ C(0) + C[\frac{3}{2}\frac{|h|}{a} - \frac{1}{2}(\frac{|h|}{a})^3], & 0 < |h| \leq a \\ C(0) + C, & |h| > 0 \end{cases} \tag{2.14}$$

(ii) Exponential:

$$\gamma(h) = \begin{cases} 0, & |h| = 0 \\ C(0) + C[1 - exp(-\frac{|h|}{a})], & |h| > 0 \end{cases} \tag{2.15}$$

(iii) Gaussian:

$$\gamma(h) = \begin{cases} 0, & |h| = 0 \\ C(0) + C[1 - exp(-\frac{|h|^2}{a^2})], & |h| > 0 \end{cases} \tag{2.16}$$

Where $C(0)$ is the nugget constant, C is the sill, $|h|$ is the length of the distance, and a is the range of the model. Figure 2.3.2 displays the abovementioned models to visualize a comparison.

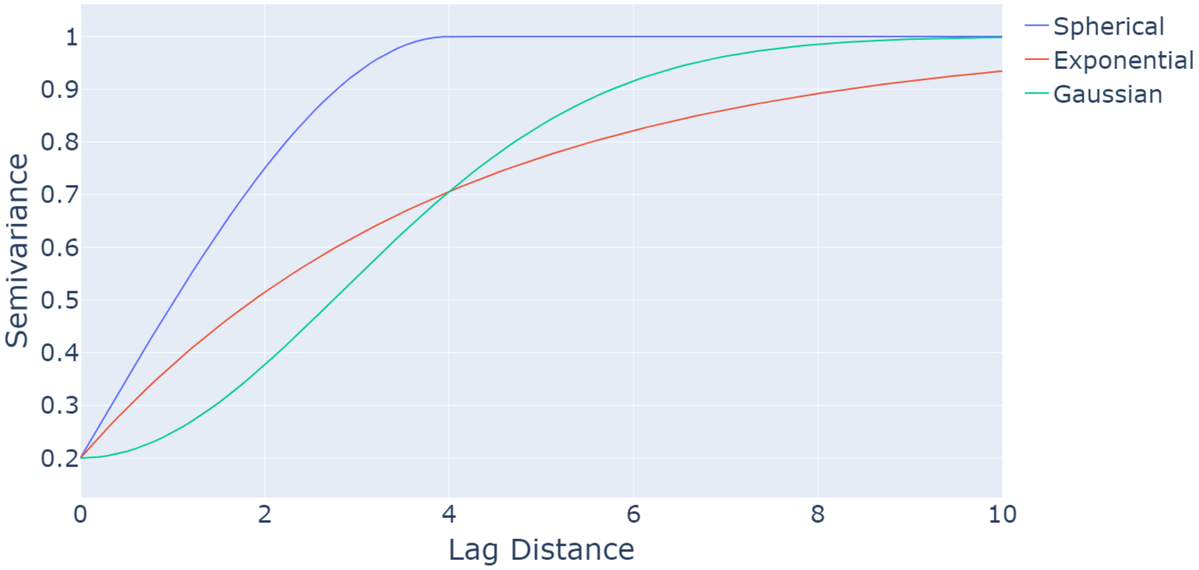


Figure. 2.3.2: *Theoretical variograms of the Spherical, Exponential, and Gaussian models are compared in this plot. The nugget, sill, and range parameters are set to 0.2, 0.8, and 4, respectively. The x-axis denotes the lag distance, while the y-axis represents the semivariance. The Spherical model reaches the sill at the shortest lag distance, while the Exponential model reaches it at the longest lag distance. The lag distance of the Gaussian model lies between these two models.*

Anisotropy

Anisotropy occurs when the spatial correlation is not the same in all directions [38]. This means that the semivariogram will depend on both the distance and direction between two points. Anisotropy is often observed in natural systems, such as in geological formations, where the spatial structure of minerals can be stretched in certain directions [24]. There are two types of anisotropy: one is known as *geometric anisotropy* and the other is referred to as *zonal anisotropy* [97]. Zonal anisotropy will not be discussed in this thesis. However, a comprehensive explanation of this concept can be found in Journel and Huijbregts [38] work.

Geometric anisotropy occurs when the range changes with the direction of the semivariogram, but the sill remains constant. By applying a linear transformation to the coordinates in a corresponding isotropic model the geometric anisotropy can be corrected. This type of anisotropy is determined by two parameters, a scaling parameter and a rotation parameter [14]. By applying the concepts describing geometric anisotropy in *"Mining Geostatistics"* by Journel and Huijbregts [38], we can explain the linear transformation of a point in a 2D space with three steps:

(i) Let the point in the 2D space be represented by a vector $x = (x_1, x_2)^\intercal$ in a coordinate system with origin in $o = (0, 0)^\intercal$. We start the transformation by rotating the coordinate system by an angle θ to obtain new coordinates (y_1, y_2). This can be done by using the rotation matrix R:

$$R = \begin{bmatrix} cos\theta & sin\theta \\ -sin\theta & cos\theta \end{bmatrix} \tag{2.17}$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = R \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.18}$$

(ii) The second step is to stretch or shrink the rotated coordinate system along each of its axes by the scaling factors λ_1 and λ_2, respectively. This results in another set of new coordinates (y_1', y_2') that are obtained using a diagonal matrix, denoted by D:

$$D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \tag{2.19}$$

$$\begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = D \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{2.20}$$

(iii) The last step rotates the coordinates (y_1', y_2') through the inverse rotation matrix

R^{-1} in order to derive the final transformed coordinates (x'_1, x'_2):

$$R^{-1} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \tag{2.21}$$

$$\begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = R^{-1} \begin{bmatrix} y'_1 \\ y'_2 \end{bmatrix} \tag{2.22}$$

The transformation process can now be simplified by calculating the product of the three matrices $R^{-1} \cdot D \cdot R$. This results in the transformation matrix T, which can be used directly on the initial coordinates (x_1, x_2) to obtain the transformed coordinates (x'_1, x'_2):

$$\begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = R^{-1}DR \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2.23}$$

The transformation method for correcting geometric anisotropy can be applied to an isotropic model to transform it into an anisotropic model. Figure 2.3.3 uses the exponential semivariogram model from Equation 2.15 to visualize how the parameters of anisotropy affect the spatial correlation of data points in space. From the figure, we see the isotropic kernel function has circular symmetry and the same value in all directions, while the anisotropic kernel function is elongated in one direction and has rapidly decreasing values along the minor axis.

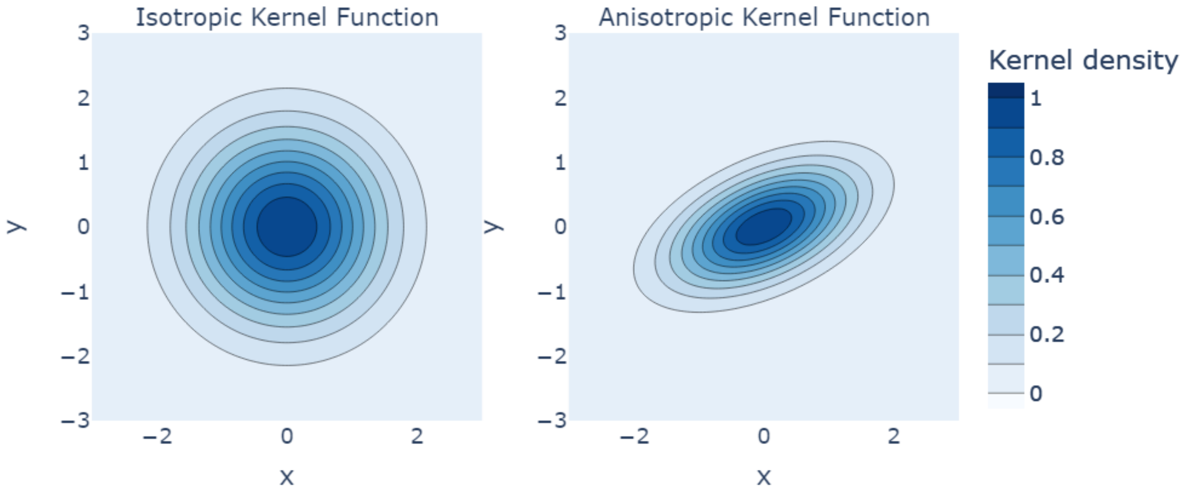


Figure. 2.3.3: *This plot illustrates the differences between isotropic and anisotropic covariance (or kernel) functions. The figure shows two contour plots of 2D exponential kernel functions, with the x and y values representing the points in the space. The left plot illustrates an isotropic kernel function, while the right plot illustrates an anisotropic kernel function. The length scale parameter of the anisotropic kernel function is set to 4, and the rotation angle is set to 2 radians. The colours represent the density of the kernel function, with darker colours indicating higher density.*

2.3.5 Kriging

Kriging is a spatial interpolation technique that estimates unknown values of interest over a domain on the basis of a limited set of sampled observation points [4]. The estimation of a point at a location with no observation is obtained by a weighted sum of the observations, which can mathematically be expressed as:

$$\hat{Z}(x_0) = \sum_{i=1}^{n} \lambda_i Z(x_i), \quad i = 1, \dots, n \tag{2.24}$$

Where,

$\hat{Z}(x_0)$ is the estimated value at the unobserved location x_0

$Z(x_i)$ is the observed value at the sampled location x_i

λ_i is the unique weight assigned to the observation at the ith location

The weights λ_i are calculated by minimizing the estimation variance σ_E^2. We can find the set of weights that results in the most accurate estimate of the unknown value, while also quantifying the uncertainty of the estimate. The estimation error is considered to be the difference between the estimated $\hat{Z}$ and the true value Z at a point x_0:

$$\hat{Z}(x_0) - Z(x_0) \tag{2.25}$$

If we assume that the expected difference between estimates and observations is equal to 0, the estimator E is said to be unbiased:

$$E\{\hat{Z}(x_0) - Z(x_0)\} = 0 \tag{2.26}$$

The estimation variance σ_E^2 is then expressed as:

$$\sigma_E^2 = Var\{\hat{Z}(x_0) - Z(x_0)\} = E\{(\hat{Z}(x_0) - Z(x_0))^2\} \tag{2.27}$$

There are several forms of kriging. The choice of kriging form depends on the assumptions made about the underlying *trend*. The trend can be defined as the systematic patterns of variations observed in the dataset [11]. The following presents a summarized overview of the key differences between the most commonly used kriging forms [38]:

(i) *Simple kriging*: assumes that the mean of the unknown variable is known and constant across the entire area of interest. This means that the assumption is second-order stationarity, refer to Equation 2.7. The estimate at an unsampled location is then calculated using the following formula:

$$\hat{Z}(x_0) = \mu + \sum_{i=1}^{n} \lambda_i (Z(x_i) - \mu), \quad i = 1, \ldots, n \tag{2.28}$$

Where μ is the known mean.

(ii) *Ordinary kriging*: assumes that the mean is unknown but constant over the entire area of interest. The estimate at an unsampled location of interest can therefore be calculated using Equation 2.24.

(iii) *Universal kriging*: also known as *kriging with a trend*, assumes that the mean differs according to a spatial trend, while the variance is constant over the area of interest. To take the trend component into account, we can define a stochastic function $Y(x)$:

$$Y(x) = m(x) + R(x) \tag{2.29}$$

Where $m(x)$ is a deterministic trend function and $R(x)$ is the stochastic residual. The trend function modelled as a polynomial of order K can be expressed as:

$$m(x) = \sum_{k=0}^{K} \beta_k f_k(x) \tag{2.30}$$

$$Z(x) = \sum_{k=0}^{K} \beta_k f_k(x) + R(x) \tag{2.31}$$

Where f_k is a deterministic function of spatial coordinates and β_k are coefficients estimated from the observations.

Figure. 2.3.4 gives a visual comparison of the aforementioned sub-types of kriging.

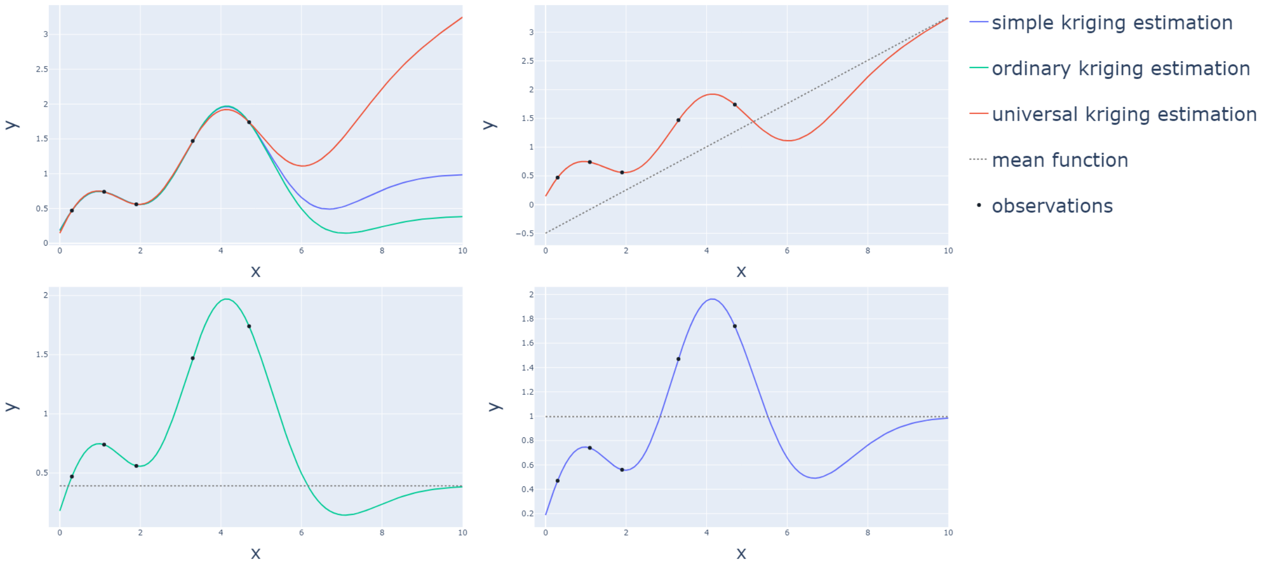


Figure. 2.3.4: *This figure shows the results of three kriging methods (simple, ordinary, and universal) applied to five observations (black dots). The x-axis represents random points on a field, and the y-axis represents the estimated values at those positions. Blue, green, and red lines represent the estimated field for each method, with dashed grey lines representing their mean functions.*

2.4 Machine Learning

The field of Artificial Intelligence (AI) has become increasingly popular in recent years due to the availability of large amounts of data and advances in computing power. Machine learning is a subset of AI. In the 1950s, Arthur Samuel, an AI pioneer, described machine learning as the "field of study that gives computers the ability to learn without being explicitly programmed" [73]. This definition holds true as machine learning involves the use of algorithms and statistical models to enable machines to discover patterns in data and make predictions about future events [67].

Nowadays, machine learning technology is widely used in many industries, including geotechnics. The application of machine learning in geotechnical engineering can be used to predict geotechnical parameters like Depth to Bedrock (DTB), which differs from the traditional empirical and statistical methods that require prior information about the relationships among the data [66].

There are three different types of machine learning: *supervised learning*, *unsupervised learning*, and *reinforcement learning.* Supervised learning involves training a model on a labelled dataset, where the algorithm learns to map the inputs to the correct outputs (labels). There are two major types of supervised learning: *classification* and *regression.* In classification tasks, the algorithm learns to assign labels to inputs, which results in discrete output values. On the other hand, in regression tasks, the algorithm is trained to predict a continuous output value [58]. Unsupervised learning trains a model on an unlabelled dataset, where the output labels are unknown. The goal is for the model to find patterns or structures in the data without guidance or supervision, allowing it to make sense of complex or unstructured data [67]. Reinforcement learning involves training a model to take actions in an environment and receive rewards or penalties based on those actions. The model must learn by itself the best strategy (policy) that maximizes the rewards it receives over time [23]. Overall, the type of machine learning to use depends on the nature of the task and the available data. Note that unsupervised and reinforcement learning is outside the scope of this study. The focus of this thesis is on regression-based supervised learning. Figure 2.4.1 shows a typical workflow for supervised machine learning.

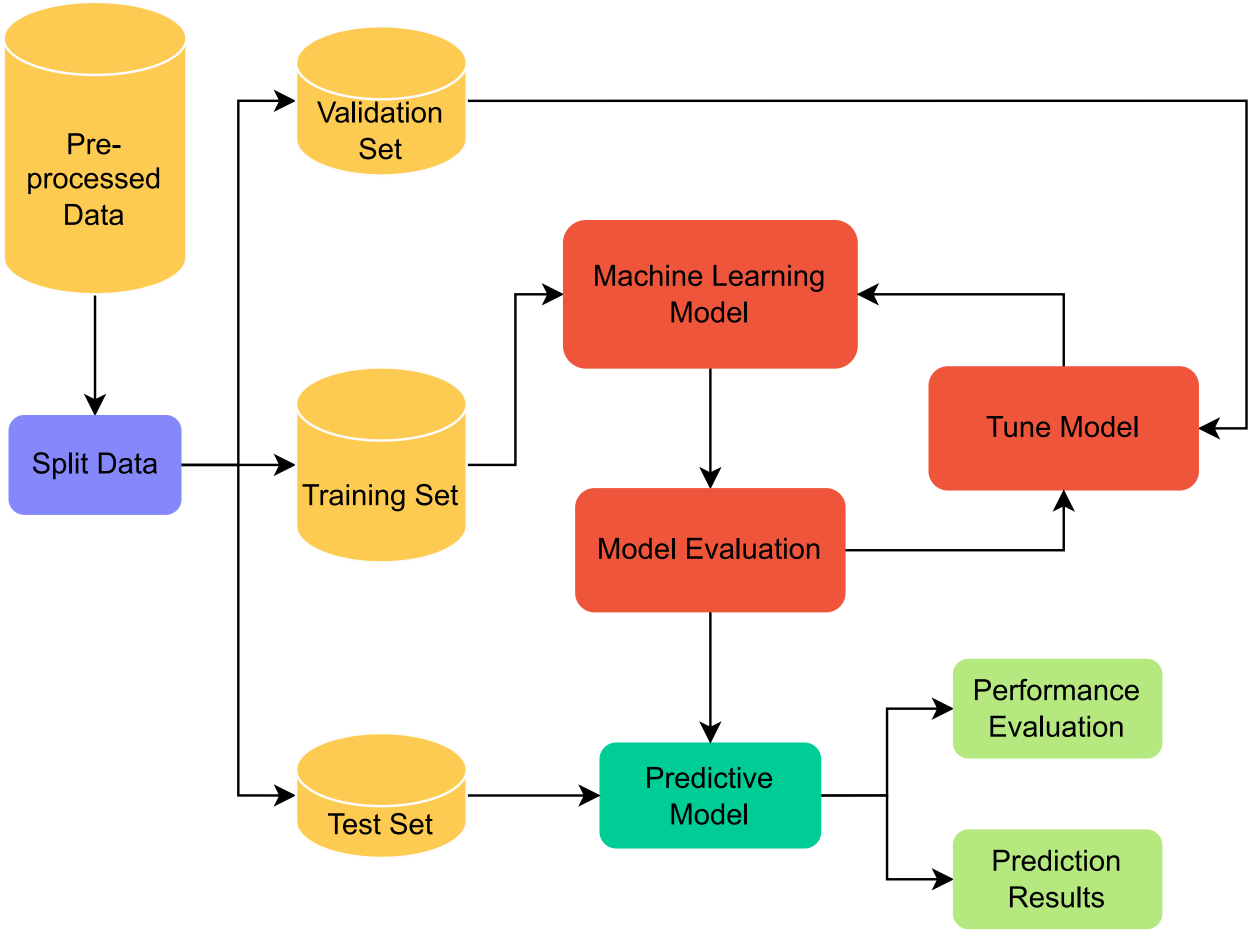


Figure. 2.4.1: *This flowchart illustrates the process of a supervised machine learning model.*

Like any other technology, machine learning also has limitations and challenges that should be considered. One of the most significant challenges is the problem of bias. Bias can arise from the data, the model, or the algorithm used to train the model, and when the model produces results that misrepresent the true relationship between the input features and the output variable [12]. Another challenge is overfitting, which occurs when the model is too complex and fits the training data too closely, leading to poor performance on new, unseen data [67].

2.5 Gaussian Processes

Gaussian Processes (GPs) are a generalized supervised learning method that can be used for both regression and classification tasks in machine learning [63]. They are based on the Gaussian probability distribution (also known as the normal distribution) and are commonly used for non-parametric modelling (i.e. not limited by a functional form). By providing a measure of confidence in the predictions, GPs can help identify situations where the model may not be reliable, allowing for a more reliable decision-making process [68]. This feature makes GPs a valuable tool in a wide range of applications, including geotechnical parameter estimation.

Unlike other machine learning methods that learn a finite set of parameters to model the data, GPs define a probability distribution over functions $f(x)$. For the function values, x, this distribution is defined by a mean function $m(x)$, which gives the expected value of the function at each input point, and a positive semidefinite [37] covariance function (kernel function) $k(x, x')$, which measures the similarity between the function values at any two points in the input domain [68].

In mathematical terms, we can represent a GP as follows [68]:

$$f(x) \sim GP(m(x), k(x, x')) \tag{2.32}$$

2.5.1 Gaussian Process Regression

Gaussian Process Regression (GPR) implements GPs for regression tasks. GPR works well on small datasets and since it is based on GPs it also provides the uncertainty related to the predictions [63].

The methodology behind GPR can be explained as follows according to Section 2.2 in Rasmussen, Williams, et al. [68]:

(i) We can start by assuming that the unknown function values follow a GP, as described by Equation (2.32). We also assume that the observed output (target) values are noisy versions of the true function values, where Gaussian noise with variance is added to each observation. So, given a set of observations y, we can write:

$$y = f(x) + \epsilon \tag{2.33}$$

Where ϵ is the Gaussian noise (i.e. irreducible error) which is defined as:

$$\epsilon \sim \mathcal{N}(0,\ \sigma_y^2 I) \tag{2.34}$$

Here, I denote the identity matrix and σ_y^2 is the variance. ϵ follows a distribution $\mathcal{N}$. The mean μ in the distribution is set to 0 since GPs are able to model the mean from an arbitrary value. In other words, this is a Gaussian distribution.

(ii) Given the observations n, we assume a training dataset in the form of a vector

$X = [x_1, \ldots, x_n]$, with corresponding target values y. The goal of GPR is to estimate the function values f_* at a new input (test) point x_*. Gaussian distribution with an expected mean of 0 can therefore be expressed as:

$$\begin{bmatrix} y \\ f_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{2.35}$$

Where,

K is a matrix constructed by using any covariance function $k(x,x')$.

$K(X,X)$ is the covariance matrix between training data points X

$K(X_*,X)$ is the covariance matrix between the test data points X_* and the training data points X

$K(X_*,X_*)$ is the covariance matrix between the test data points X_*.

(iii) The key predictive equations are then calculated as:

$$f_*|X,y,X_* \sim \mathcal{N}(\bar{f}_*, cov(f_*)), \quad \text{where} \tag{2.36}$$

$$\bar{f}_* \triangleq \mathbb{E}[f_*|X,y,X_*] = K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1} y, \tag{2.37}$$

$$cov(\bar{f}_*) = K(X_*,X_*) - K(X_*,X)[K(X,X) + \sigma_n^2 I]^{-1} K(X,X_*) \tag{2.38}$$

(iv) To simplify the equations from (iii) we can define $K = K(X,X)$ and $K_* = K(X,X_*)$, which constitutes a column vector of n kernel functions for one test point x_*, i.e. $x_* \mapsto k(x_i, x_*)$. Hence, we can write the predictive equations as a function:

$$\bar{f}(x_*) = \sum_{i=1}^{n} \alpha_i k(x_i, x_*), \quad i = 1, \ldots, n \tag{2.39}$$

Where $\bar{f}(x_*)$ represents the mean prediction from the predictive function $f(x_*)$. x_i refers to the training points, with n being the total number of training points. α_i is a weight assigned to each training point based on its distance from the new test point and its uncertainty.

The similarity between Equation 2.39 and the kriging Equation 2.24 in Section 2.3.5 is evident, where both equations estimate the value of an unobserved point by using a linear combination of observed data points. The weights in both equations are determined based on the correlation or covariance structure between the observed data points and the unobserved point. Figure 2.5.1 illustrates the key stages involved in making predictions with a GPR model.

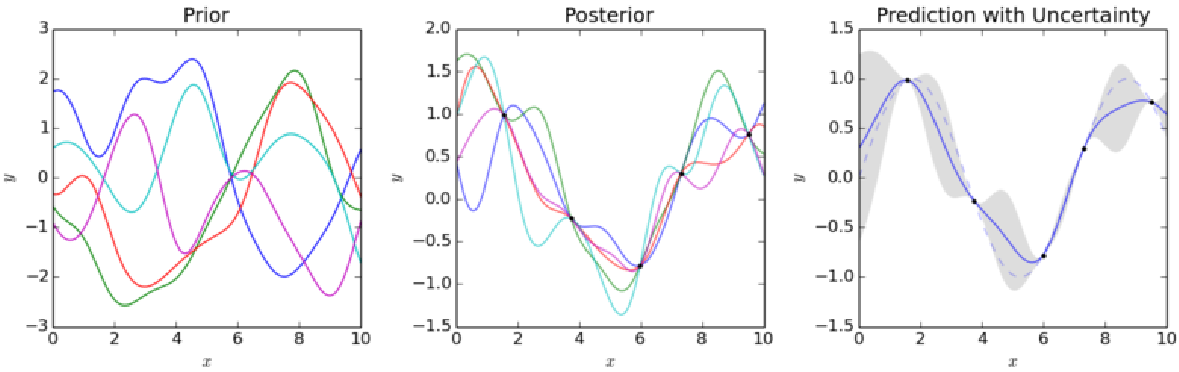


Figure. 2.5.1: *Gaussian Process Regression (prediction) with a squared exponential kernel. The left plot show draws from the prior distribution of functions. The middle plot show draws from the posterior predictive distribution. The right plot is the mean prediction (blue line) with 1 standard deviation (shaded grey) and one realization (blue dashed line) [10]. Illustration under CC-BY-SA-4.0 licence.*

2.6 Kernels

The kernel function, also referred to as the covariance function, is a crucial component in GPR. The kernel function determines the shape and smoothness of the prior distribution over functions, which in turn affects the behaviour of the posterior distribution over functions. We have already covered the main concepts regarding kernel functions in Section 2.3.4. This section will therefore mainly focus on the relevant kernels for this thesis, which are the Radial Basis Function (RBF) kernel, the Matérn 3/2 kernel, and a combined kernel with RBF and linear kernels.

The choice of kernel function can have a significant impact on the model's behaviour [18]. A common kernel function is the squared exponential kernel, also known as the RBF kernel or Gaussian kernel. This kernel is mathematically defined as [68]:

$$k(x, x') = \sigma^2 \exp\left(-\frac{1}{2\ell^2}\|x - x'\|^2\right) \tag{2.40}$$

Where σ^2 is the signal variance parameter (where σ is known as the *amplitude* parameter), ℓ is the length scale parameter which describes the correlation length, and $\|x - x'\|^2$ is the Euclidean distance between x and x'. The squared exponential kernel assumes that the correlation between two points decays exponentially with distance and is smooth and infinitely differentiable.

Another commonly used kernel is the Matérn kernel, which is a family of kernels that allows us to control the smoothness of the function being modelled. The Matérn kernel is defined as [27]:

$$k(x, x') = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}}{\ell} \|x - x'\| \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}}{\ell} \|x - x'\| \right) \tag{2.41}$$

where Γ is the gamma function, K_ν is the modified Bessel function, ℓ is the length scale, and ν is a parameter controlling the smoothness. When:

(i) $\nu \to \infty$, the kernel converges to the squared exponential kernel from Equation 2.40, resulting in very smooth functions.

(ii) $\nu = \frac{1}{2}$, the kernel yields the exponential kernel, also known as the *Ornstein-Uhlenbeck Process* [88], and it results in functions that are continuous.

(iii) $\nu = \frac{3}{2}$, the kernel is differentiable once, which means that the resulting functions are smooth and can be used for modelling data that has some level of noise and exhibits trends.

(iv) $\nu = \frac{5}{2}$, the kernel is differentiable twice, which results in even smoother functions that can be used for modelling data with less noise and more complex trends.

In simpler terms, lower values of ν yield rougher functions, and the Matérn kernel is a flexible kernel that can be used to model a wide range of functions with varying degrees of smoothness [68].

Both of the abovementioned kernel functions are isotropic. To take geometric anisotropy into account when determining the distribution over the functions, the kernel function can be modified to include different length scales and rotation angles for each input dimension as described in Section 2.3.4.

Combining kernels can be of great use when wanting to model more complex functions and detect different structures in the data that cannot be detected by a single kernel alone [56]. Multiplying kernels results in a kernel which has a high value only if both of the two base kernels have a high covariance value while adding kernels results in a kernel that has a high value if either of the two base kernels has a high covariance value [18]. A possible kernel combination is a linear kernel and a RBF kernel, which can be interesting to test if there is both linearity and anisotropy present in a dataset. If we add these kernels together, we can mathematically express it as:

$$k(x, y), (x', y') = k_x(x, x') + k_y(y, y') \tag{2.42}$$

Where $k_x(x, x')$ is the kernel function for the RBF kernel from Equation 2.40 and $k_y(y, y')$ is the kernel function for the linear kernel which can be expressed as:

$$k(y, y') = \sigma_b^2 + \sigma_s^2(y - c)(y' - c) \tag{2.43}$$

Where σ_b is the bias amplitude that is equal to adding an uncertain offset to the model, σ_s is the slope amplitude, and c is the shift which sets the x-coordinate for the point of intersection for all lines in the posterior function [18].

2.7 Hyperparameter Selection

Hyperparameters in machine learning are parameters that are not learned from data but are set before the training process begins and remain constant during training. They control the learning process and can significantly affect the model's performance [23].

In GPR, hyperparameters define the covariance function's characteristics, which, in turn, determines the shape of the prior distribution over the functions. These hyperparameters can be tuned so the optimal values for the hyperparameters are selected to improve the model's performance. There are several methods that can be used for hyperparameter selection, such as Grid search, Random search, and Bayesian optimization [75]. In GPR, maximizing the log marginal likelihood of the training data is a commonly used approach for fitting the hyperparameters [68].

Before defining the log marginal likelihood, it is necessary to introduce the marginal likelihood. The marginal likelihood is obtained by marginalizing (or integrating), the product of the likelihood function and a prior distribution over the model parameters [57]. The marginal likelihood is given by:

$$p(y|X) = \int p(y|f, X)\ p(f|X)\ df \tag{2.44}$$

Where $p(y|f, X)$ is the likelihood function of the observed data given the function values, and $p(f|X)$ is the prior distribution over the function values.

In the case of a GPR, the marginal likelihood can be denoted as $p(y|X, \theta)$, which represents the likelihood of observing the target values y, given the input data X and the hyperparameters θ. It can be expressed as

$$p(y|X, \theta) = \mathcal{N}(y|0, K_\theta) \tag{2.45}$$

Here, $\mathcal{N}(y|0, K_\theta)$ denotes a multivariate Gaussian distribution with mean 0 and covariance matrix K_θ, which is parameterized by the hyperparameters θ.

As previously stated in this section, the optimal hyperparameter values are obtained by maximizing the log marginal likelihood. By taking the logarithm of Equation 2.45, we can obtain the log marginal likelihood [68]:

$$\log\ p(y|X,\theta) = -\frac{1}{2}y^{\intercal}K_{\theta}^{-1}y - \frac{1}{2}\log(|K_{\theta}|) - \frac{n}{2}\log(2\pi) \tag{2.46}$$

Where n is the number of points in the input data, and $|K_{\theta}|$ is the determinant of K_{θ}.

Finally, the maximization of the log marginal likelihood function can be written as:

$$\hat{\theta} = \arg\max_{\theta}\ (\log p(y|X,\theta)) \tag{2.47}$$

Where $\hat{\theta}$ are the trained hyperparameters.

2.8 Accuracy Assessment

To evaluate a model's performance, it is essential to measure its accuracy by training and comparing multiple models. To compare the models, we must select appropriate metrics that evaluate their performance. When performing supervised regression tasks, it is necessary to calculate *residuals*, which are the difference between the true and predicted values. The closer the residual value is to 0, the better the model performs. The following metrics are used to compare models in this thesis.

2.8.1 R^2 Score

The R^2 score, also known as the coefficient of determination measures how well a model makes its predictions based on a scale between 0 and 1. A score of 1 indicates that the model fits the data perfectly, while a score of 0 suggests that the model is predicting the mean value of the observed data [58]. The R^2 score can be calculated using the following formula:

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2}, \quad i = 1, \ldots, n \tag{2.48}$$

Where y_i is the true value, $\hat{y}_i$ is the predicted value, and $\bar{y}$ is the mean value of the observed data. RSS stands for the *sum of squares of residuals* and TSS stands for the *total sum of squares.*

2.8.2 Root Mean Squared Error

A commonly used performance metric for regression problems is the Root Mean Squared Error (RMSE). It is the square root of the average squared distance between true and predicted values, i.e. the *standard deviation* [23]. The smaller the RMSE, the better the model's performance. The mathematical formula to compute the RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \quad i = 1, \ldots, n \tag{2.49}$$

Where n is the number of samples in the dataset you are measuring the RMSE on.

2.8.3 Continuous Rank Probability Score

The Continuous Rank Probability Score (CRPS) [25] is a metric that compares an actual true value to its predicted distribution. The smaller the CRPS value, the more accurate the predicted distribution is. We can define the formula for calculating CRPS as:

$$CRPS(F, y) = \int_{\mathbb{R}} \left[F(x) - H(x \geq y)\right]^2 \; dx \tag{2.50}$$

Where,

y is the actual true value

x is the parameter of interest

F is the predicted CDF

$H(x)$ is the Heaviside step function [1]:

$$H(\hat{y}_i) = \begin{cases} 1, & x = 0 \\ 0, & x \leq 0 \end{cases} \tag{2.51}$$

2.9 Loss Function and Optimization

In machine learning, a loss function measures how well a model's predictions match its intended target value. It quantifies the error between predicted and actual values, and training a model involves minimizing this loss function. An optimizer adjust the weights and biases of a model in a direction such that the loss function is minimized. The choice of what loss function to use depends on the problem [12].

2.9.1 Negative Log Marginal Likelihood

The Negative Log Marginal Likelihood (NLML) is a loss function which is generally used to find optimal hyperparameters $\hat{\theta}$ for GP models [68]. It is defined as the negative of the log marginal likelihood function, refer to Equation 2.46. The negative sign is added to convert the maximization problem into a minimization problem. The formula for NLML can be written as:

$$\hat{\theta} = \arg\min_{\theta} \ (-\log \ p(y|X, \theta)) \tag{2.52}$$

2.9.2 Adaptive Moment Estimation

Adaptive Moment Estimation optimizer, also known as the Adam optimizer, is a gradient descent [46] based optimization algorithm that was presented by Diederik Kingma and Jimmy Ba [42]. It combines the advantages of two other optimization methods, namely Adaptive Gradient Algorithm (AdaGrad) [17] and Root Mean Square Propagation (RMSProp) [86]. The Adam optimizer adjusts the learning rate of each weight in the model during training by computing a running average of the first and second moments of the gradients, i.e. the mean and the variance, respectively. The algorithm then uses these estimates to update the hyperparameters, resulting in faster convergence and better performance [71].

2.10 Geographic Information System

Geographic Information Systems (GIS) is a system designed to capture, store, manipulate, analyze, and present spatial or geographic data. It allows users to overlay different layers of information onto a map, creating a more complete picture of a geographic area [61].

Geotechnical data can be integrated into a GIS system to create a comprehensive understanding of the geological and geotechnical features of a particular site or region. For example, geological maps, borehole logs, and geophysical surveys can be digitized and integrated into a GIS, allowing for easy visualization and analysis of the subsurface conditions. This information can then be used to make informed decisions about site selection, design, and construction, as well as to identify potential geohazards such as landslides or sinkholes [92].

Chapter 3

Data Exploration

To gain a better understanding of the data used in this thesis, an initial data exploration was conducted. In this chapter, we will start by discussing the datasets in more detail, including the type of data collected, the location of the study area, and the time frame of the data collection. Next, we will provide an overview of the geological setting of the study sites, which is important for understanding the geotechnical properties of the soil and rock formations. Finally, we will discuss semivariogram models, which are used to characterize spatial dependence and variability in geotechnical data.

3.1 Datasets

This section provides a description of the datasets that were used to train and test the machine learning approach. Two datasets were used for both training and testing the models and include boreholes obtained from TS in geotechnical investigations conducted by NGI. NGI stands for the Norwegian Geotechnical Institute and is an independent research centre that combines geotechnical knowledge and technology to provide services within a wide range of sectors, including energy, infrastructure, construction, and the environment [59].

Two borehole datasets were used in this study. They were retrieved from NGI's geodatabase and converted to comma-delimited Excel files. The first borehole dataset is from geotechnical investigations near a highway called E18 in Bærum municipality in Norway. We will therefore refer to this dataset as *Bærum E18*. The geographical location of the dataset is presented in Figure. 3.1.1.

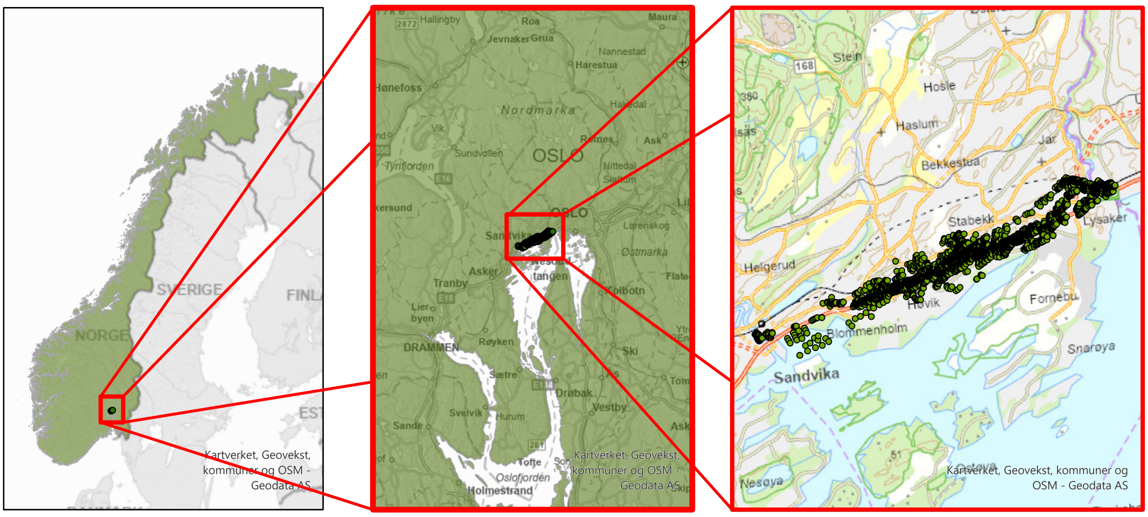


Figure. 3.1.1: *Study area: E18, Bærum, a highway in Bærum municipality in Norway.*

As shown in Figure. 3.1.1, the data has high point density and the boreholes are linearly distributed along the highway. The dataset consists of a total of 8376 boreholes derived from TS.

The second borehole dataset is from geotechnical investigation projects that cover a part of the eastern part of Oslo municipality in Norway. Hence, is why we will refer to this dataset as *Oslo East*. The geographical location of this dataset is presented in Figure. 3.1.2. This dataset contains 2335 boreholes that are also derived from TS. The area is more rectangular and the boreholes are spread more evenly than in the Bærum E18 dataset.

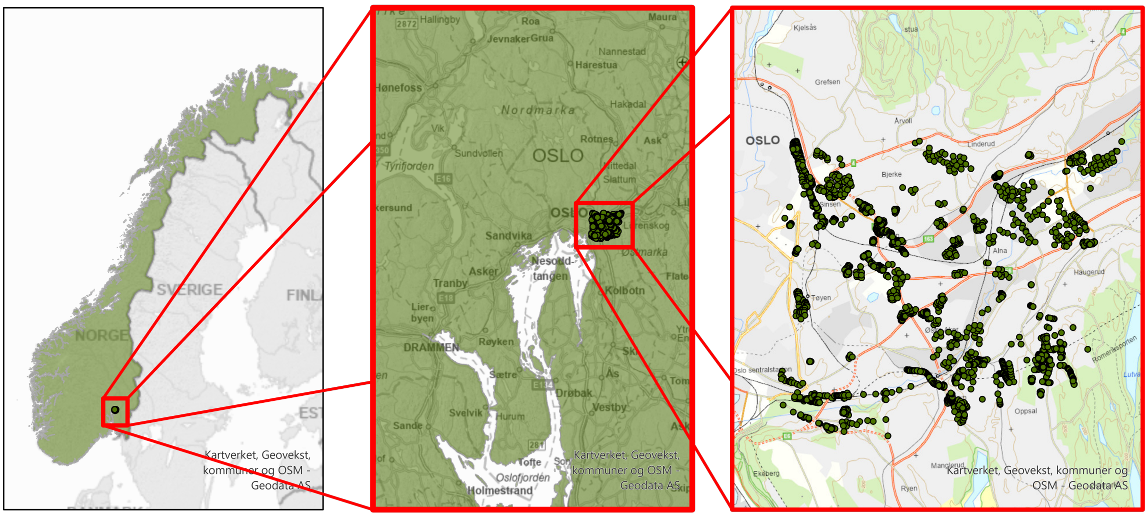
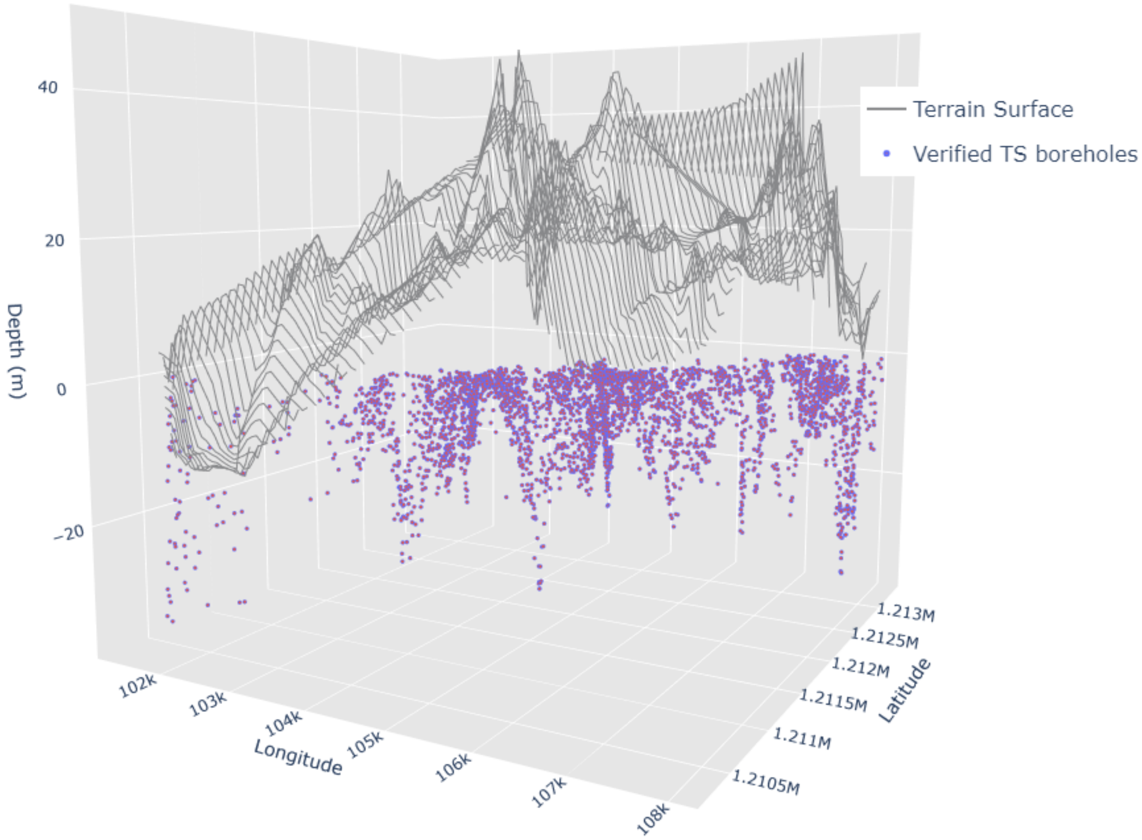
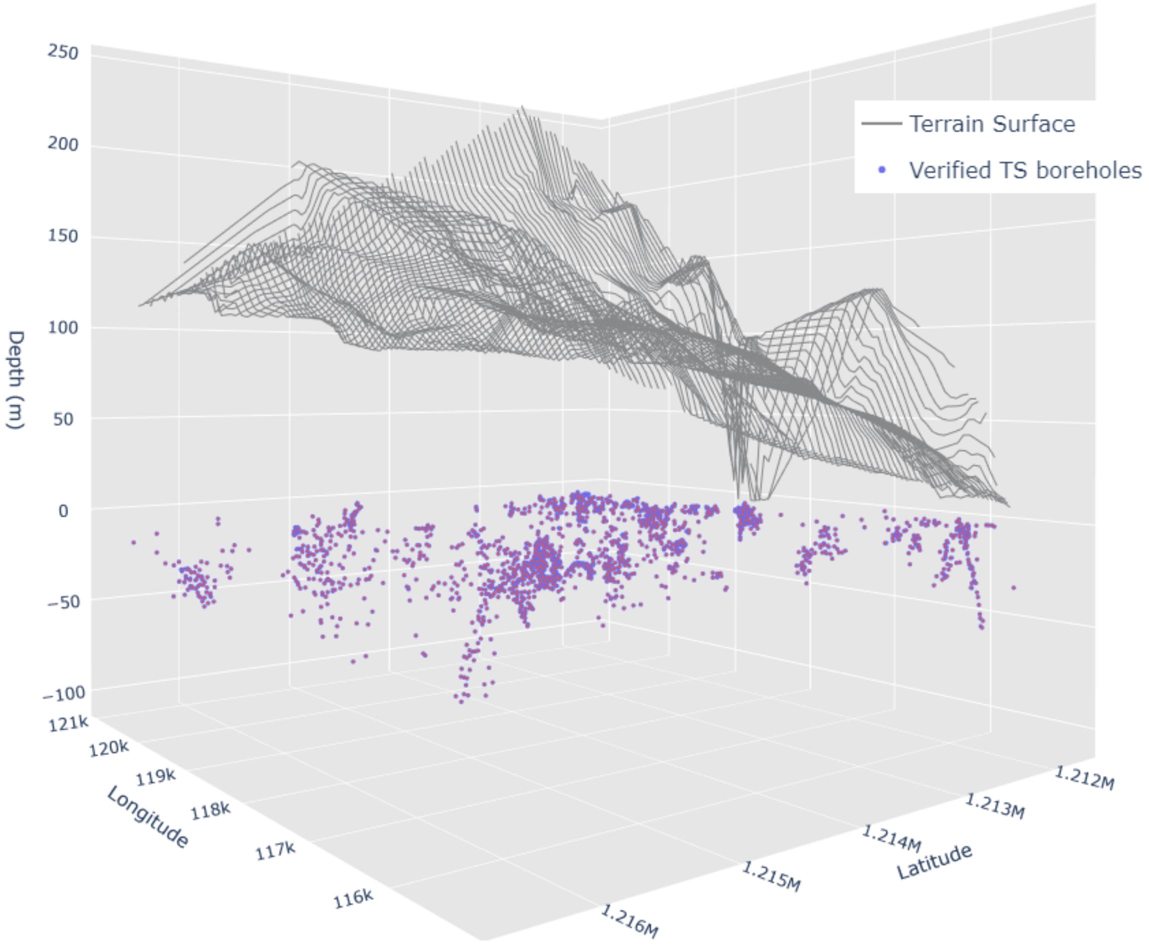


Figure. 3.1.2: *Study area: Eastern part of Oslo municipality in Norway.*

The datasets used in the study contain information about the DTB for all boreholes. To visualize the boreholes and terrain surface in the study area, 3D projections of the datasets were generated. This is presented in Figure. 3.1.3. The longitude and latitude coordinates of the study area are given in the EUREF89 NTM Zone 10 (EPSG:5110) map projection and are represented by the x- and y-axis, respectively. The z-axis represents the depth of the boreholes and the terrain in meters. The DTM-derived gridded grey surface above the borehole points is also shown in the same map projection. The figure indicates that all borehole values are below 0. This is due to the fact that outcrops, i.e. bedrock visible on the earth's surface, are defined as 0 in the datasets; thus, even if an outcrop is at a terrain height of, for instance, 4 meters, it is defined as 0 meters in the datasets.



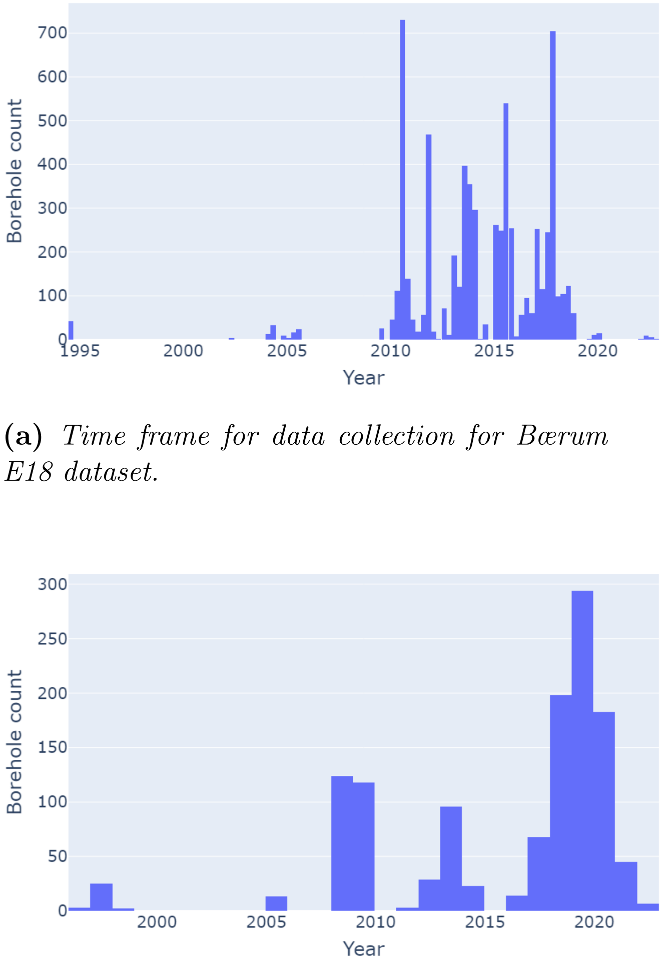
(a) *Terrain surface and boreholes in Bærum E18 dataset.*



(b) *Terrain surface and boreholes in Oslo East dataset.*

Figure. 3.1.3: *The subfigures 3.1.3a and 3.1.3b are 3D projections of the terrain surface and borehole depth in the datasets provided by NGI. The x- and y-axis represent the latitude and longitude coordinates respectively, while the z-axis represents the depth of the boreholes and the terrain surface in meters.*

The data collection time frame is based on the individual projects that the boreholes originate from. In Figure. 3.1.4 we show the month and year of investigation for each borehole in the datasets. A majority of the boreholes are from investigations conducted between 2010 and 2020, with some older data points dating back to 2005 and 1994. One might wonder if the quality of the data depends on the age of the data. As this is geotechnical data collected thorough TS the information about the borehole depth itself is quite accurate. However, according to Lysdahl (personal communication, Apr. 20, 2023) at NGI, there can be some inaccuracy in the actual GPS positioning on the surface. Data collected before 2010 could have a 1-3 meters inaccuracy in GPS positioning compared to data collected after 2010, particularly in areas with forests or near buildings. Data collected after 2010 has a requirement of at least a minimum accuracy of 10 centimters for coordinates and terrain height.



(a) *Time frame for data collection for Bærum E18 dataset.*

(b) *Time frame for data collection for Oslo East dataset.*

Figure. 3.1.4: *The subfigures 3.1.4a and 3.1.4b show the time frame for when the borehole data was collected in the datasets provided by NGI. The x-axis represents the month and year for when data was collected, whilst the y-axis refers to the number of boreholes from the time frames.*

3.2 Geological Setting

In order to gain a more comprehensive understanding of the geology in the study areas, we will explore the geological setting present in the borehole datasets. Since the study sites are closely located, many of the geological properties are similar. A quaternary geological map of the study areas, derived from the Geological Survey of Norway (NGU) map database, is shown in Figure. 3.2.1. About 55% of Norway has exposed bedrock or bedrock with a thin layer of quaternary sediments, as reported by Olsen et al. [62]. It is evident from Figure. 3.2.1 that some areas in both datasets coincide with exposed bedrock regions, while the remaining areas are characterized by a continuous sediment cover, which is according to Olsen et al. [62] estimated to have an average thickness of approximately 6 meters.

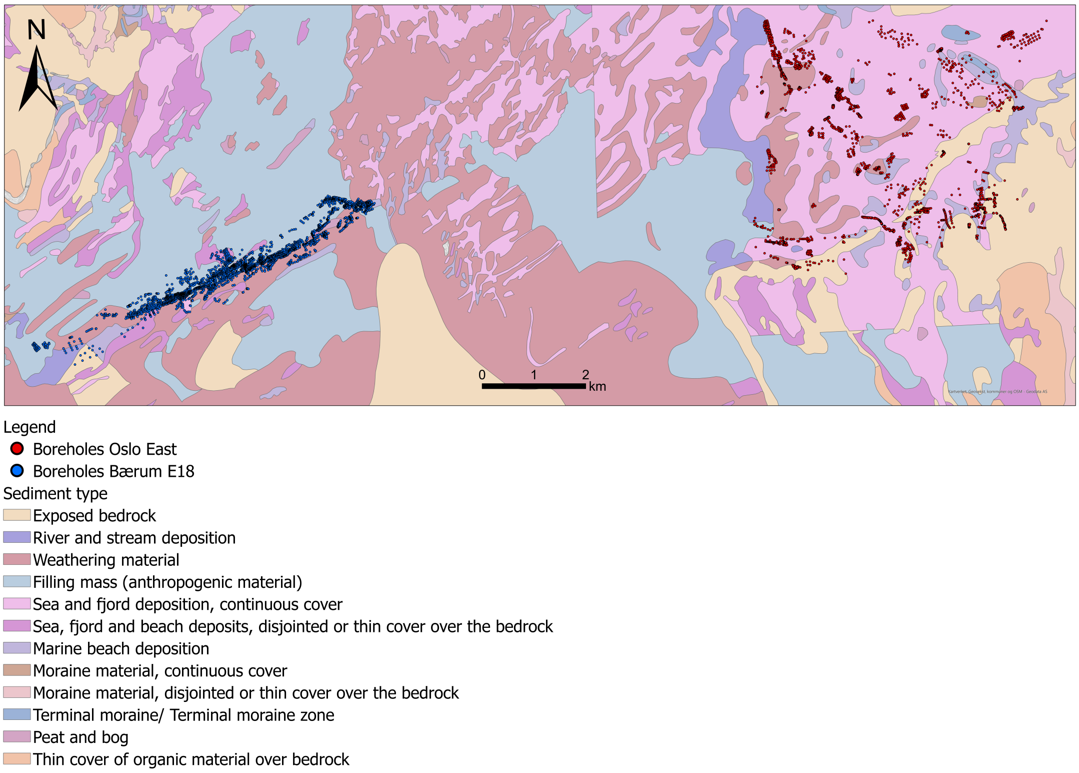
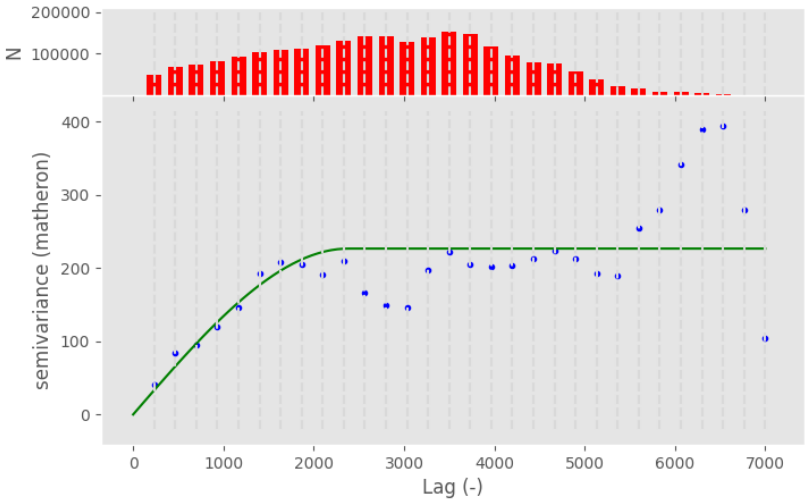

N
0
1
2
km
Legend
● Boreholes Oslo East
● Boreholes Bærum E18
Sediment type
Exposed bedrock
River and stream deposition
Weathering material
Filling mass (anthropogenic material)
Sea and fjord deposition, continuous cover
Sea, fjord and beach deposits, disjointed or thin cover over the bedrock
Marine beach deposition
Moraine material, continuous cover
Moraine material, disjointed or thin cover over the bedrock
Terminal moraine/ Terminal moraine zone
Peat and bog
Thin cover of organic material over bedrock

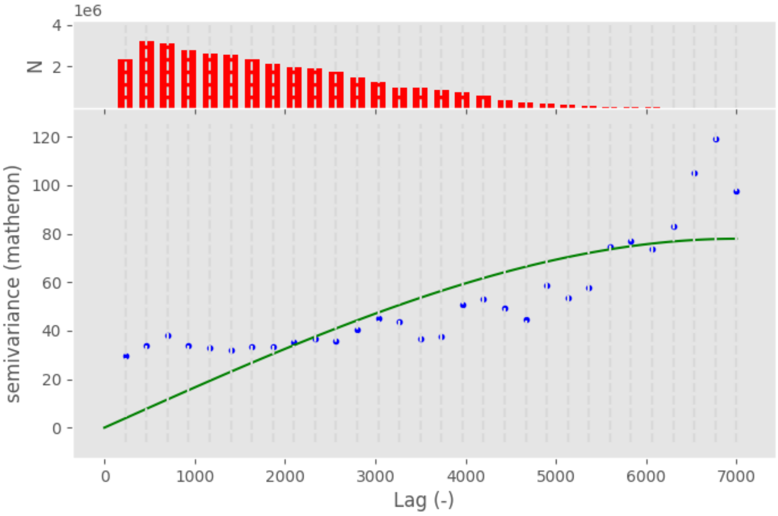
Figure. 3.2.1: *Quaternary geological map of the study areas, based on data from the NGU map database. The map shows the distribution of various sediments and the boreholes from the NGI datasets. The red points in the map refer to the boreholes in the Oslo East dataset, whilst the blue points refers to the boreholes in the Bærum E18 dataset. The different sediment types are represented as coloured polygons in the map, where the map legend specifies what sediment type each colour represents. More information about the map can be found at ngu.no.*

3.3 Semivariogram Models

Semivariograms were produced and studied to gain insights into the spatial correlation of the data. The semivariograms were created using the Matheron estimator [54] and the isotropic spherical model given by Equation 2.14. From the results presented in Figure 3.3.1, we can observe that both datasets have a moderate to strong spatial correlation. The effective ranges of the models are around 7000 meters, indicating that the correlation between data points diminishes beyond this range. The sills of the models are relatively high, meaning that there is a strong degree of correlation between the data points. The



(a) *Semivariogram for Oslo East dataset.*



(b) *Semivariogram for Bærum E18 dataset.*

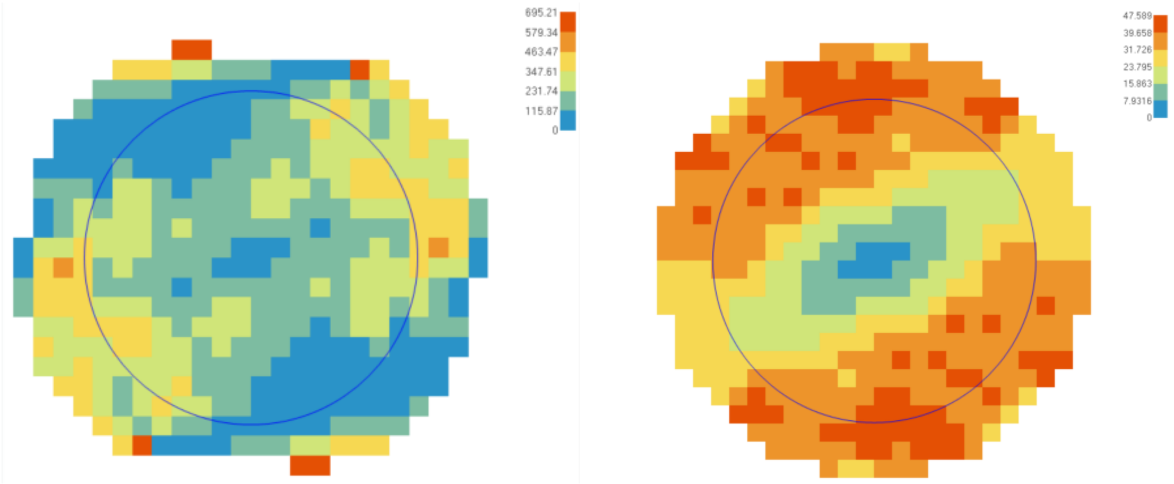
Figure. 3.3.1: *Subfigures 3.3.1a and 3.3.1b portray the semivariograms of the datasets provided by NGI. These semivariograms were computed using SciKit-GStat [51] with the Python backend. The semivariograms were estimated using the Spherical theoretical model (green line) and fitted to an experimental variogram (blue dots) with 30 lag classes up to a maximum of 7000 units (x-axis). The histogram at the top of the plots indicates the number of point pairs for each lag class and shares the x-axis with the semivariogram.*

semivariogram for the Bærum E18 dataset has a nugget effect of 14.25 meters, indicating the presence of measurement error or noise that cannot be explained by the model. The exact values of the parameters calculated from the semivariograms are listed in Table 3.3.1.

In addition to the semivariograms in Figure 3.3.1, semivariogram maps were created by using ArcGIS Pro [34], a GIS software. Semivariograms are usually presented as 1D curves showing the relationship between the semivariogram value and the lag distance along a

Table 3.3.1: *Semivariogram parameters for two locations.*

Dataset	Model	Estimator	Effective Range	Sill	Nugget
Bærum E18	Spherical	Matheron	6999.96 m	59.73 m	14.25 m
Oslo East	Spherical	Matheron	7110.54 m	304.88 m	0.00 m



(a) *Semivariogram map for Oslo East dataset.* **(b)** *Semivariogram map for Bærum E18 dataset.*

Figure. 3.3.2: *The subfigures 3.3.2a and 3.3.2b show the semivariogram maps for the datasets provided by NGI. Both semivariogram maps are created with the GIS software, ArcGIS Pro.*

particular direction. However, to gain a global understanding of the spatial correlation in all directions, we can create a 2D representation in the form of a semivariogram map. The map is created by calculating the semivariogram at a set of regularly spaced points throughout the study area and interpolating between these points to create a continuous surface [15]. Figure 3.3.2 illustrates the semivariogram maps for both borehole datasets. The maps show the semivariogram values as a colour scale, where darker colours (red) indicate large semivariance values. The maps also provide insights into the directionality of the spatial correlation, as evident from the elliptical shape of the contours. Especially in Figure 3.3.2b, we see a clear anisotropy in the spatial correlation, where the correlation is stronger in the northeast-southwest direction compared to the northwest-southeast direction. This information is valuable for selecting appropriate kernel functions in the GPR model, as anisotropic models may be required to accurately capture the spatial correlation of the data.

Chapter 4

Methods

This chapter presents the methodology employed in this thesis, with a focus on the framework developed for estimating DTB. The chapter discusses the software and tools used, data pre-processing, and the implementation of the models. The source code used for this study can be accessed via the GitHub links provided in Table A.0.1 in Appendix A.

Figure 4.0.1 shows a workflow that summarizes the methodology in this thesis.

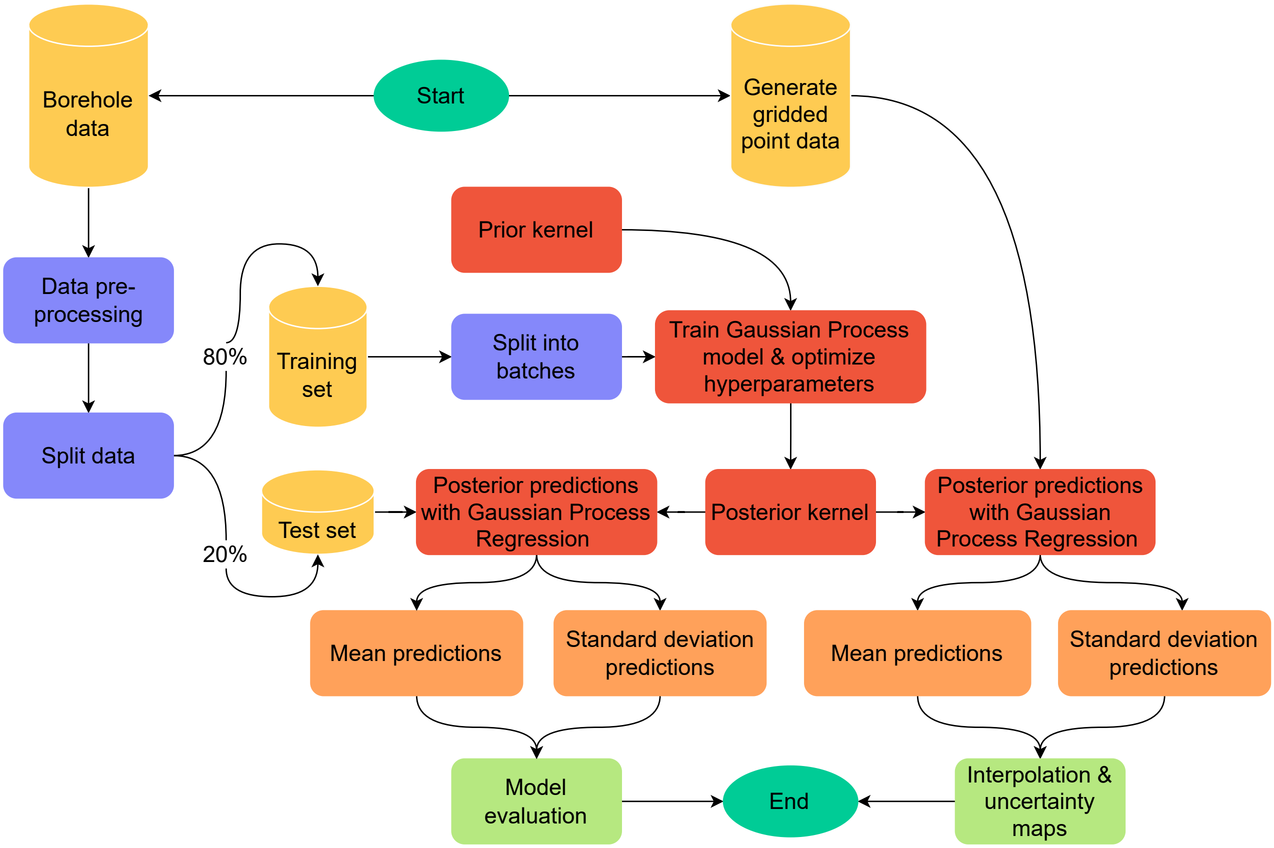

Borehole data
Start
Generate gridded point data
Prior kernel
Data pre-processing
80%
Training set
Split into batches
Train Gaussian Process model & optimize hyperparameters
Split data
Test set
20%
Posterior predictions with Gaussian Process Regression
Posterior kernel
Posterior predictions with Gaussian Process Regression
Mean predictions
Standard deviation predictions
Mean predictions
Standard deviation predictions
Model evaluation
End
Interpolation & uncertainty maps

Figure. 4.0.1: *Workflow of the methodology in this study for estimating Depth to Bedrock using Gaussian Process models.*

4.1 Software and Hardware

Python [89] version 3.10.10 was used in this study. Data manipulation and analysis were performed using the Pandas module [55], while basic scientific computations were carried out using the NumPy module [28]. TensorFlow version 2 [52] was utilized to implement the models, and the results were visualized and evaluated using the Plotly library [35] with Python. A complete list of the software and Python modules used to create the source code can be found in Table A.0.2 in Appendix A.

For this study, all computations were performed on a standard laptop with a CPU of 2.80 GHz and 16 GB of RAM. The machine ran on a 64-bit Windows operating system. Despite not having high-end hardware, the laptop was able to run all the necessary tasks in a timely manner, and the results were obtained successfully. Although more powerful hardware could have potentially sped up some of the processes, the laptop used for this study was capable of completing all the required tasks.

4.2 Data Pre-processing

In addition to the data exploration from Chapter 3, it was necessary to do some data pre-processing before using the datasets in the models. Both borehole datasets provided by NGI were relatively prepared. Subsets from the datasets used in the study are presented and described in Table 4.2.1. We are using 3 columns from each dataset for the models, where 2 of the columns are the input features and the last column is the target values. The input features represent the longitude and latitude coordinates that are the position of the boreholes, and the target values represent the DTB values in meters. The datasets were stored as Pandas [55] DataFrames.

To avoid biased models and incorrect results we checked for missing values, which resulted in the Bærum E18 dataset having none, and the Oslo East dataset having 177 missing DTB values. Many machine learning models. including GPs, are unable to handle missing values [36]. As the number of missing values was relatively small, we decided to remove all rows with missing DTB values from the dataset. Consequently, the final number of boreholes in the Oslo East dataset used for the models was 2158.

The final step in data pre-processing involved examining the distribution of the datasets. The original data distribution for the datasets used in this study is illustrated in Figure 4.2.1. An important observation is that the input data, represented by the longitude and latitude coordinates, has a significantly larger scale compared to the output data or target

(a) *Subset from Bærum E18 dataset*

X	Y	DTB
101247.056	1210636.918	18.7
101261.11	1210620.965	29
101265.003	1210652.979	20.8
101274.151	1210609.009	25.1
101276.233	1210669.003	11.6
101277.056	1210637.019	20.9
101279.962	1210665.029	14.4
101283.285	1210569.039	24.5
101289.11	1210621.059	21.3
...	...	...

(b) *Subset of Oslo East dataset*

X	Y	DTB
118552.99	1212222.01	<Null>
118855.86	1212450.93	5.96
118866.6	1212451.11	5.6
115726.65	1212913.03	1.1
115727.1	1212912.48	1.3
115719.38	1212914.61	1
115720.35	1212914.7	0.8
115719.65	1212915.64	0.5
115712.62	1212918.44	2.3
...	...	...

Table 4.2.1: *Subsets from borehole datasets provided by NGI. Tables 4.2.1a and 4.2.1b provide the structure of the datasets, where the first two columns (X and Y) correspond to the longitude and latitude coordinates, respectively. These coordinates are expressed in the EUREF89 NTM Zone 10 (EPSG:5110) map projection. The last column (DTB) contains the depth to bedrock values, which are the target values in this study.*

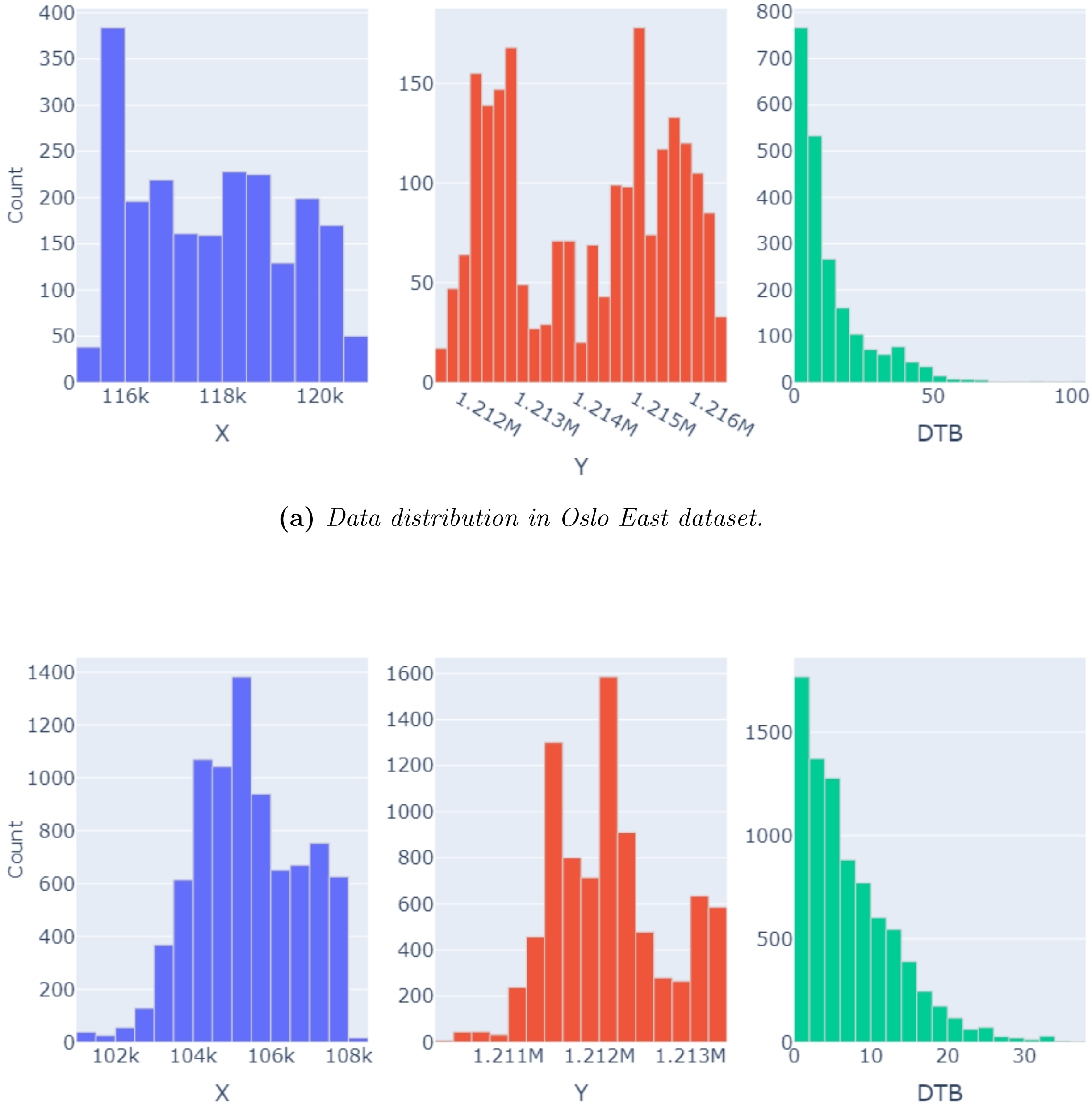
values, which is evident by comparing the x-axis values in the figure.

GPR is a non-parametric approach that does not make any assumptions about the underlying distribution of the data. Instead, we select a prior distribution that best represents the data being modelled. As a result, GPR does not require any scaling of the input data. However, the kernel function can be affected by the scale of the input features. If the input features are on different scales, the kernel function may be biased towards the features with larger scales, leading to inaccurate estimates [68].

To ensure that all data used in the model are on a similar scale, both the input data and the output data (target values) were standardized. Standardization is the process of transforming data so that it has a mean μ of zero and a standard deviation σ of one:

$$x_{scaled} = \frac{x - \mu}{\sigma} \tag{4.1}$$

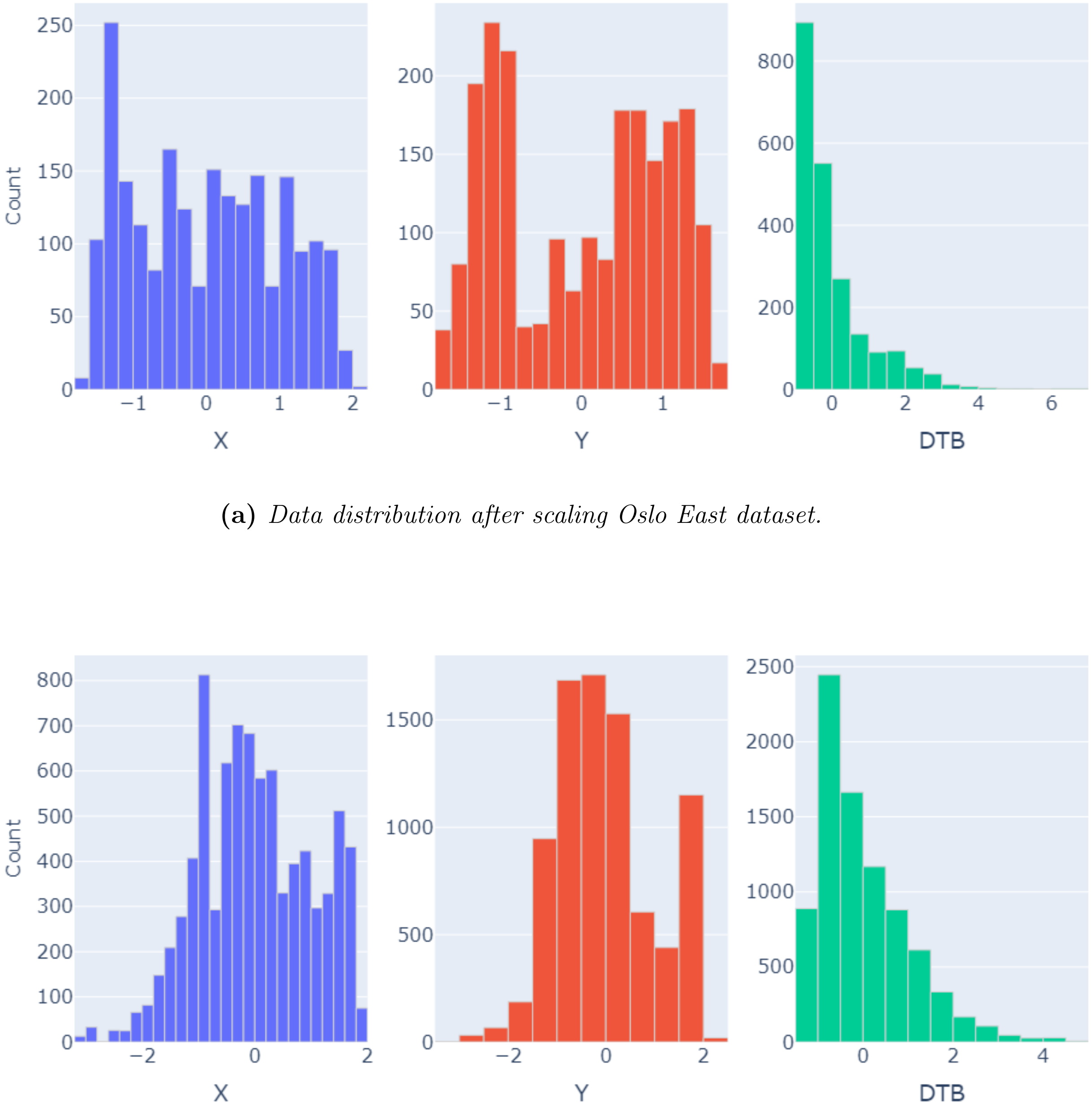
Standardizing the target values ensures that they are on the same scale as the input features, thereby preventing the model from overemphasizing the impact of the coordinates and underemphasizing the impact of the DTB feature and vice-versa. Figure 4.2.2 shows the data distribution after standardization.



(a) *Data distribution in Oslo East dataset.*

(b) *Data distribution in Bærum E18 dataset.*

Figure. 4.2.1: *The subfigures 4.2.1a and 4.2.1b show the data distribution for each column in the borehole datasets provided by NGI. X and Y correspond to the longitude and latitude coordinate values, respectively, and DTB is the depth to bedrock values. The values in the columns are represented on the x-axis, while the count of those values is shown on the y-axis.*



(a) *Data distribution after scaling Oslo East dataset.*

(b) *Data distribution after scaling Bærum E18 dataset.*

Figure. 4.2.2: *The subfigures 4.2.2a and 4.2.2b show the data distribution for each column in the borehole datasets provided by NGI after standardization. X and Y correspond to the longitude and latitude coordinate values, respectively, and DTB is the depth to bedrock values. The scaled values in the columns are represented on the x-axis, while the count of those values is shown on the y-axis.*

4.3 Model Implementation

In this section, we describe the implementation of the GP models used in this study. The implementation is strongly based on the source code by Roelants [70] and is created using the TensorFlow [52] and TensorFlow Probability [16] frameworks. TensorFlow provides a powerful set of tools for building and training machine learning models, while TensorFlow Probability allows us to construct probabilistic models that can capture the uncertainty inherent in the given problems.

The specific GP models implemented in this study are the `GaussianRegressionModel` and `GaussianProcess` models from the TensorFlow Probability module. These models allow us to construct and train GPs using different types of kernel functions and likelihood functions. In addition, these models also enable us to make predictions on new unseen data. In the following sections, we will provide a detailed explanation of the implementation process, including kernel functions, hyperparameter optimization, Cross-validation (CV), and model evaluation.

4.3.1 Choice of Kernel

When using the GP models from TensorFlow, any kernels from the positive-semidefinite kernels package in TensorFlow Probability can be used. For this study, the following three kernel functions were used:

- RBF
- Matérn kernel with $\nu = \frac{3}{2}$ (Matérn 3/2)
- Linear

All three kernel functions are described in more detail in Section 2.6.

The reason behind choosing the RBF kernel for this study is that it relies on the principle that similar inputs have similar outputs [33]. This principle is essential for our study, as we work with spatial data. However, this kernel has a tendency to quickly decrease exponentially towards 0 the further the extrapolations are away from the training data. According to Stein [82], the RBF kernel is too smooth and is unrealistic when it comes to modelling many physical systems, and therefore recommends using the Matérn kernel. Therefore, the Matérn 3/2 kernel was chosen, as it strikes a good balance between smoothness and flexibility. The linear kernel is only used in combination with the RBF kernel to try to capture any underlying linear trends in the data, which can be useful in cases where the data has a clear or some degree of linearity.

Anisotropy was included by combining the kernels with an input-transformed kernel using the `FeatureTransformed` function from the positive-semidefinite kernels package in TensorFlow. This function creates a transformation function that transforms the input data and then passes them to the base kernel, which in this case can be either RBF, Matérn 3/2, or the combined kernel with the RBF and linear kernels. The transformation function was implemented using the formulas in Section 2.3.4.

4.3.2 Model Training

Hyperparameter Optimization

To optimize the hyperparameters, we use the Adam optimizer from TensorFlow. The Negative Log Marginal Likelihood (NLML) of the model is calculated using the training data and the negative of the `log_prob` function from the TensorFlow Probability module. The hyperparameters are updated using gradient descent until convergence, which is determined by monitoring the change in the NLML between iterations. The optimization process is repeated several times with different initial values of the hyperparameters to ensure that the optimization process does not get stuck in a local minimum. The final set of hyperparameters is chosen based on the model with the lowest NLML value.

Each kernel has its own set of trainable hyperparameters. The trainable hyperparameters for each kernel used in this thesis are listed in Table 4.3.1. The mean function for all models was set to 0. All hyperparameter prior values were set with a Gaussian (normal) distribution in mind, and the hyperparameters were constrained to be positive to avoid negative hyperparameter values after optimization. For the RBF and Matérn 3/2 kernels, the trainable hyperparameters are σ and ℓ, which represent the amplitude and length scale of the kernel, respectively. The amplitude controls how much the functions vary in vertical scale, while the length scale represents the "wiggliness" of the function which determines how far away from the data the model can extrapolate [18]. The length scale parameter is also trainable for the geometric anisotropy transformation. But the difference here is that there are two length scale parameters $\ell_1,$ and ℓ_2, one for each input dimension to detect the dominating direction of the input data. In addition to ℓ_1 and ℓ_2, the transformation also has a trainable angle parameter θ which controls the orientation to rotate the input data [38]. The bias amplitude σ_b, slope amplitude σ_s, and shift c parameters in the linear kernel control the offset, slope, and intercept just like in a standard linear regression function [18]. Lastly, the observation noise variance parameter σ_y controls the amount of noise added to the observed data. This hyperparameter is used for all models regardless of which kernel is chosen, and its value is learned during the optimization process along with the other hyperparameters [68].

Table 4.3.1: *Trainable hyperparameters.*

Kernel/ Transformation	Formula	Trainable Parameters	Parameter Description
RBF	2.40	σ and ℓ	Amplitude and length scale
Matérn 3/2	2.41	σ and ℓ	Amplitude and length scale
Linear	2.43	σ_b , σ_s , and c	Bias amplitude, slope amplitude, and shift
Geometric Anisotropy	2.23	ℓ_1 , ℓ_2 , and θ	Length scale for input dimension 1, length scale for input dimension 2, and rotation angle in radians
Observation noise variance	2.33	σ_y	The variance of the Gaussian noise

Cross-Validation

Cross-validation (CV) is a method used to evaluate the performance of a machine learning model on a given dataset. This process involves partitioning the dataset into training and test sets, training the model on the training set, and evaluating the model on the test set. Hence, the datasets were split randomly into training and test sets, where 80% of the data was used for model training and 20% for testing and evaluating model performance.

To further ensure the model is not overfitting to a specific subset of the training data, we applied shuffling to the training set and split it into batches of size 128. This allows us to perform hyperparameter optimization, as described in Section 4.3.2, multiple times with different subsets of the training data. In our framework, we ran 500 iterations as we observed that the loss functions for the models stabilized after approximately 250 iterations. The final model with the optimal hyperparameter values is then evaluated on the test set to estimate the model's performance on unseen data.

Model Evaluation

When evaluating the performance of the GPR models, we use the metrics that were defined in Section 2.8. These metrics include the R^2 score, RMSE, and CRPS. The R^2 metric provides an indication of how well the model fits the data on a scale between 0 and 1, while the RMSE metric measures the difference between the predicted values and the actual values. The CRPS metric, on the other hand, evaluates the probabilistic predictions of the model by comparing them to the actual values.

We used these metrics to assess the performance of the models after training and testing. This allows us to determine if the models are overfitting to the training data or if they generalize well to new, unseen data.

4.3.3 Interpolation

The aim of the model is to interpolate DTB values at locations without any observations. To accomplish this, additional data had to be created. We generated an evenly spaced 2D grid of points using the Pandas [55] module. The x- and y axes of the grid were set to match the range of the borehole datasets, ensuring we had known DTB values available in the interpolation field to check if the predictions are consistent with the actual measured values in that area. We used 100 equally spaced samples along each axis, resulting in a final grid of 10,000 points that were used for the interpolation.

The DTB values from the interpolation were obtained as the mean prediction from the trained model. The standard deviation of the predictions was used as a measure of uncertainty, providing insight into how confident the model was in its predictions. For example, if the standard deviation is high at a certain point, it indicates that the model is less certain about the prediction at that point. On the other hand, if the standard deviation is low, the model is more confident about the prediction at that point. The mean predictions and the standard deviation values were calculated based on the posterior predictive distribution obtained from the trained model.

The mapping of interpolation results involved two visualizations: one for the mean predictions and one for the estimated uncertainties. The maps were created by transforming the points in the predicted field into rasters with a 100-meter resolution. The borehole dataset observations were overlaid as points on the raster that displayed the mean predictions and were coloured based on their actual DTB values. To highlight the difference between the maps, another colorscale (yellow to red) was used to visualize the raster that displayed the uncertainties associated with the mean predictions. The maps were created using ArcGIS Pro [34].

Chapter 5

Results

This chapter presents and compares the results obtained from the tested models. The chapter is organized into two sections, each corresponding to the borehole datasets used to train and test the models. For each dataset, six different types of kernels were used to train the models. Therefore, for each dataset, we obtained six distinct models. A detailed description of the datasets can be found in Chapter 3, while the methodology for the model implementation is described in Chapter 4.

In each section, we provide the results obtained from training the models, including optimized hyperparameters, metric results, results from the loss function, and visualizations of the interpolation and uncertainty results.

5.1 Oslo East Dataset

5.1.1 Training Results

The training of GPR model on the Oslo East dataset only took approximately 4-6 seconds to complete for each of the six models. During the training process, the hyperparameters were optimized using CV as described in Section 4.3.2. The optimized hyperparameters for each of the six kernels used to train the models are shown in Table 5.1.1. The table shows both the prior values set for each hyperparameter as well as the posterior values after training.

The loss function for each model during the training process is shown in Figure 5.1.1. As expected, the loss function decreases with each iteration and eventually converges to a steady-state value. The convergence rate and final loss value vary depending on the kernel type used. The models using the Matérn 3/2 kernel achieve the lowest final loss value.

Table 5.1.1: *Hyperparameter optimization results for the Oslo East dataset. The posterior is presented in scaled values.*

(a) *RBF kernel*

Hyperparameter	Prior	Posterior
Isotropy		
Amplitude σ	1.0	4.955
Length scale ℓ	1.0	0.220
Observation noise variance σ_y	0.001	0.002
Geometric anisotropy		
Amplitude σ	1.0	3.831
Length scale 1 ℓ_1	1.0	3.279
Length scale 2 ℓ_2	2.0	7.104
Angle θ	1.57	1.034
Observation noise variance σ_y	0.001	0.002

5.1.2 Metric Results

The metric results for each model trained on the Oslo East dataset are shown in Table 5.1.2. The R^2 score, which measures the amount of variation explained by the model, ranges from 0.600 to 0.829 across the six models. The RMSE ranges from 0.364 to 0.559, and the CRPS ranges from 0.153 to 0.263. The best model performance for both isotropic and anisotropic models is achieved by the Matérn 3/2 kernel. Where the best-performing model is the anisotropic Matérn 3/2 model with an R^2 score of 0.829, RMSE of 0.364, and a CRPS of 0.152.

The model performance was visualized using scatter plots that display the mean prediction values against the true values (or ground truth), as shown in Figure 5.1.2. The plots indicate that the model predictions are generally close to the true values, as most points are near the diagonal line. However, some outliers are present, indicating potential areas of improvement for the model. Overall, the scatter plots demonstrate that the models perform well in predicting DTB values for the Oslo East dataset.

5.1.3 Best Model Results

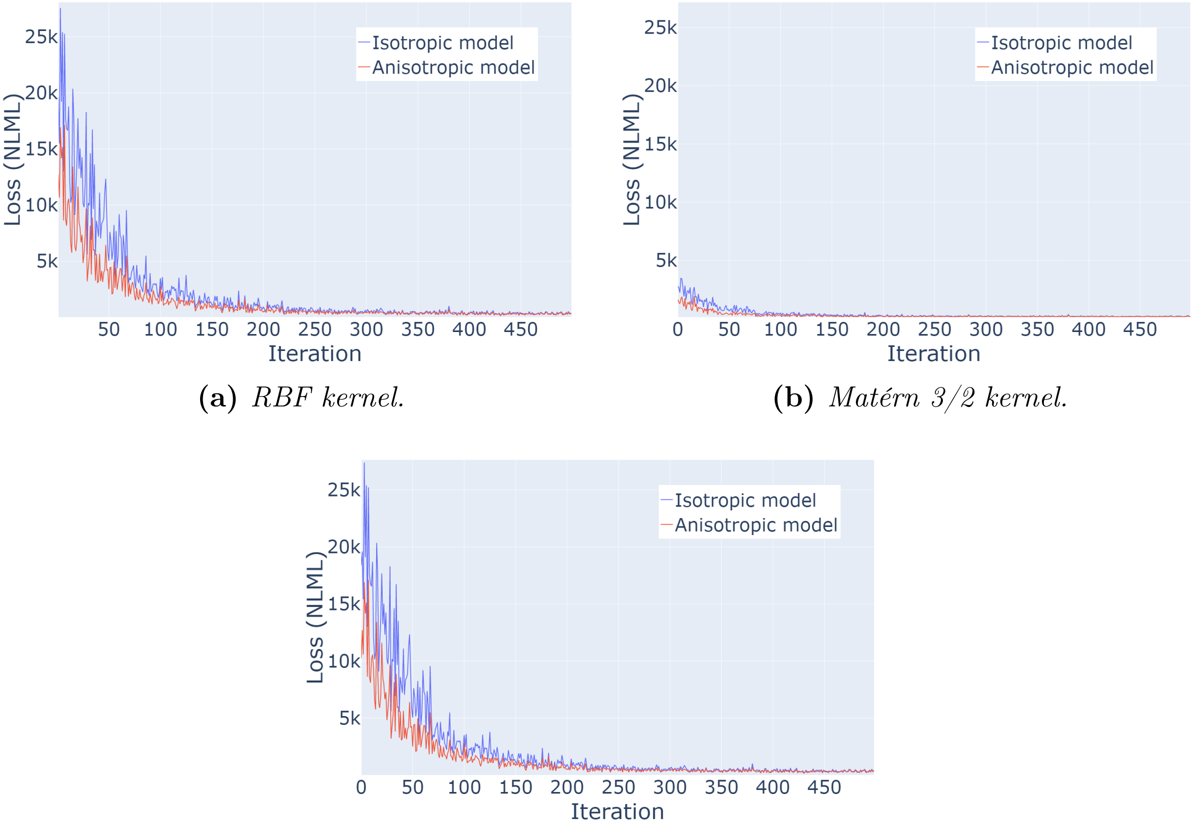
In this section, we will display the interpolated field and associated uncertainties for the best-performing model trained on the Oslo East dataset. To create the figures, we followed the procedure described in Section 4.3.3. Based on the metrics presented in Table

(b) *Matérn 3/2 kernel*

Hyperparameter	**Prior**	**Posterior**
Isotropy		
Amplitude σ	1.0	2.834
Length scale ℓ	1.0	0.312
Observation noise variance σ_y	0.001	0.002
Geometric anisotropy		
Amplitude σ	1.0	2.226
Length scale 1 ℓ_1	1.0	2.807
Length scale 2 ℓ_2	2.0	5.805
Angle θ	1.57	0.945
Observation noise variance σ_y	0.001	0.002

(c) *Linear + RBF kernel*

Hyperparameter	**Prior**	**Posterior**
Isotropy		
Bias amplitude σ_b	0	0
Slope amplitude σ_s	1.0	0.566
Shift c	0	7.469
Amplitude σ	1.0	4.964
Length scale ℓ	1.0	0.220
Observation noise variance σ_y	0.001	0.002
Geometric anisotropy		
Bias amplitude σ_b	0	0
Slope amplitude σ_s	1.0	0.292
Shift c	0	8.321
Amplitude σ	1.0	3.843
Length scale 1 ℓ_1	1.0	3.288
Length scale 2 ℓ_1	2.0	7.095
Angle θ	1.57	1.034
Observation noise variance σ_y	0.001	0.002



(a) *RBF kernel.*

(b) *Matérn 3/2 kernel.*

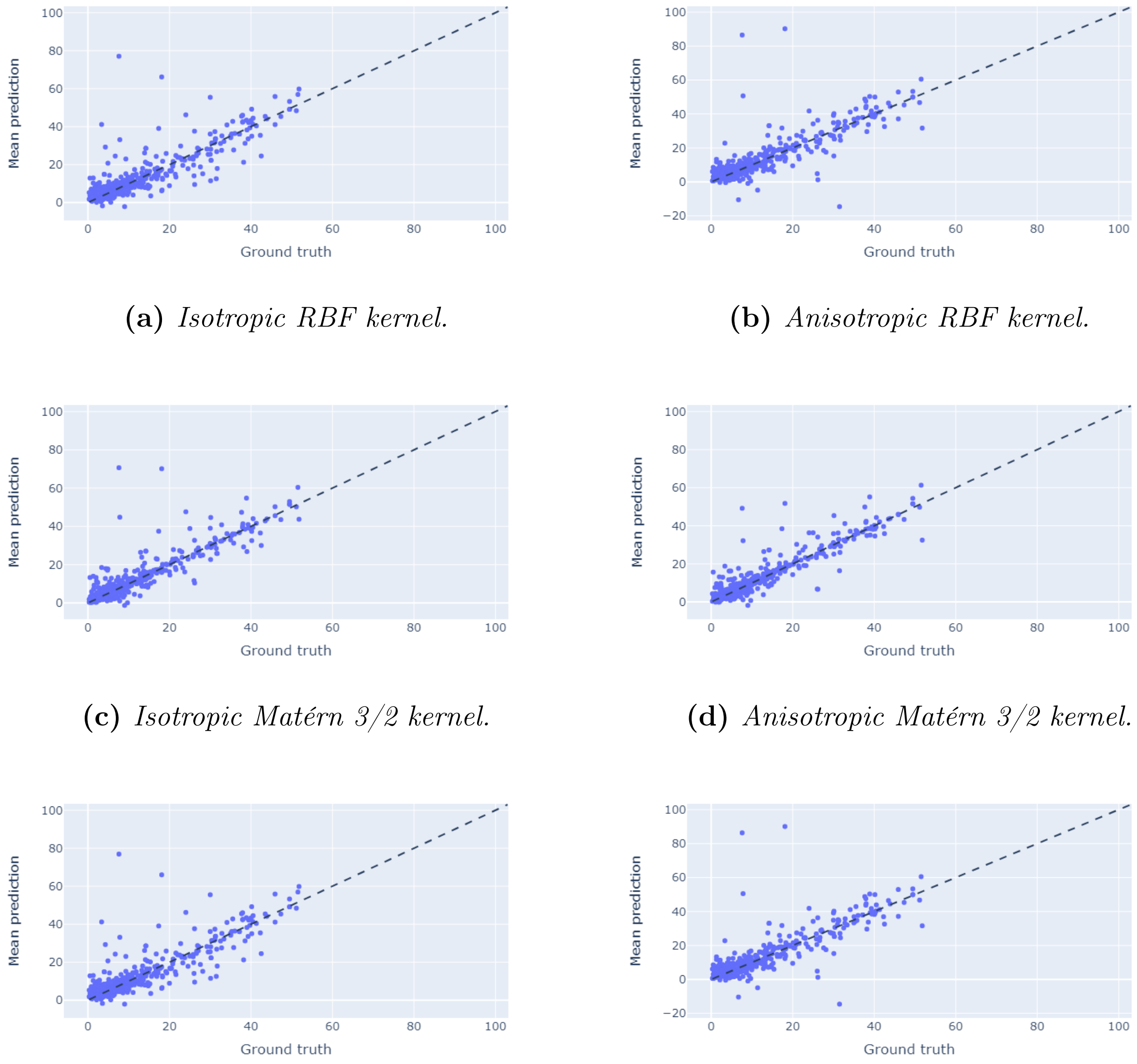
(c) *Linear + RBF kernel.*

Figure. 5.1.1: *The NLML for six models trained on the Oslo East dataset. The blue and red lines represent the loss during training for the isotropic and anisotropic models, respectively. The x-axis displays the NLML value, while the y-axis represents the number of iterations.*

5.1.2, the anisotropic model with the Matérn 3/2 kernel performed the best for the Oslo East dataset. The results for this model are shown in Figure 5.1.3. The interpolation and uncertainty results for all models trained on the Oslo East dataset can be found in Appendix B.

Table 5.1.2: *Metric results for Oslo East dataset. R^2 is the coefficient of determination, Root Mean Squared Error (RMSE), and Continuous Rank Probability Score (CRPS).*

Kernel	Metric	Isotropy	Geometric anisotropy
RBF	R^2	0.676	0.600
	RMSE	0.502	0.558
	CRPS	0.260	0.262
Matérn 3/2	R^2	0.755	0.829
	RMSE	0.437	0.364
	CRPS	0.172	0.153
Linear + RBF	R^2	0.677	0.601
	RMSE	0.502	0.558
	CRPS	0.260	0.261



(a) *Isotropic RBF kernel.*

(b) *Anisotropic RBF kernel.*

(c) *Isotropic Matérn 3/2 kernel.*

(d) *Anisotropic Matérn 3/2 kernel.*

(e) *Isotropic Linear and RBF kernel.*

(f) *Anisotropic Linear and RBF kernel.*

Figure. 5.1.2: *Scatter plots of the mean prediction values (y-axis) versus ground truth values (x-axis) obtained from the six Gaussian Process for Regression (GPR) models trained with six different kernels on the Oslo East dataset. All values are in meters.*

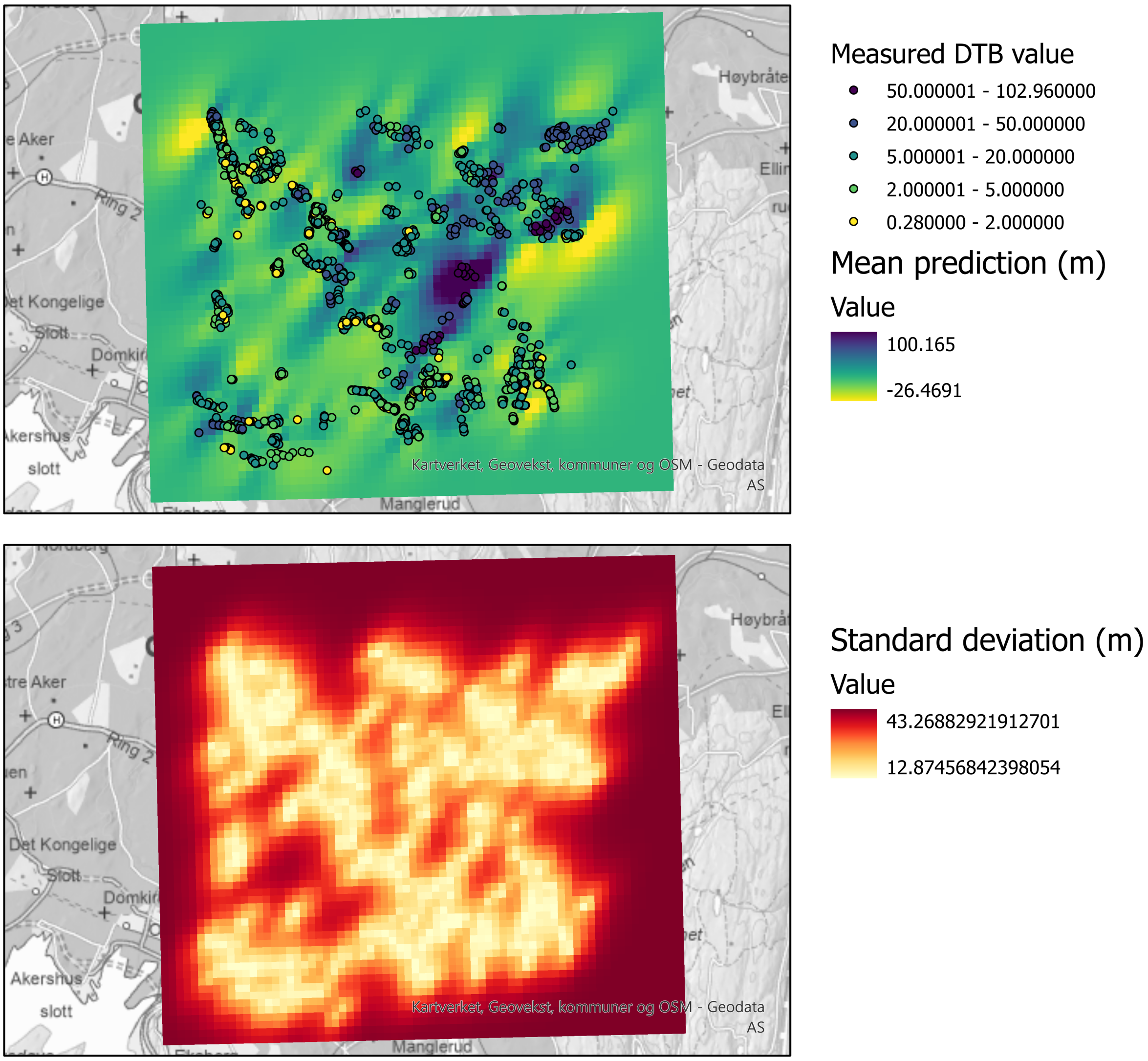


Figure. 5.1.3: *Interpolation and uncertainty for the anisotropic model with Matérn 3/2 kernel trained on Oslo East dataset.*

5.2 Bærum E18 Dataset

This section presents the same workflow presented in Section 5.1 but for the Bærum E18 dataset. The dataset contains 8376 boreholes, making it the larger of the two datasets.

5.2.1 Training Results

Similar to the Oslo East dataset, the GPR models were trained on the Bærum E18 dataset using six different kernels. The training process took approximately 32-40 seconds to complete for each of the six models. The results from the hyperparameters optimized during training for each of the six kernels are shown in Table 5.2.1. The table displays both the prior values set for each hyperparameter and the posterior value after training each model.

Table 5.2.1: *Hyperparameter optimization results for Bærum E18 dataset. The posterior is presented in scaled values.*

(a) *RBF kernel*

Hyperparameter	**Prior**	**Posterior**
Isotropy		
Amplitude σ	1.0	7.199
Length scale ℓ	1.0	0.122
Observation noise variance σ_y	0.001	0.003
Geometric anisotropy		
Amplitude σ	1.0	5.709
Length scale 1 ℓ_1	1.0	6.421
Length scale 2 ℓ_2	2.0	11.531
Angle θ	1.57	1.695
Observation noise variance σ_y	0.001	0.002

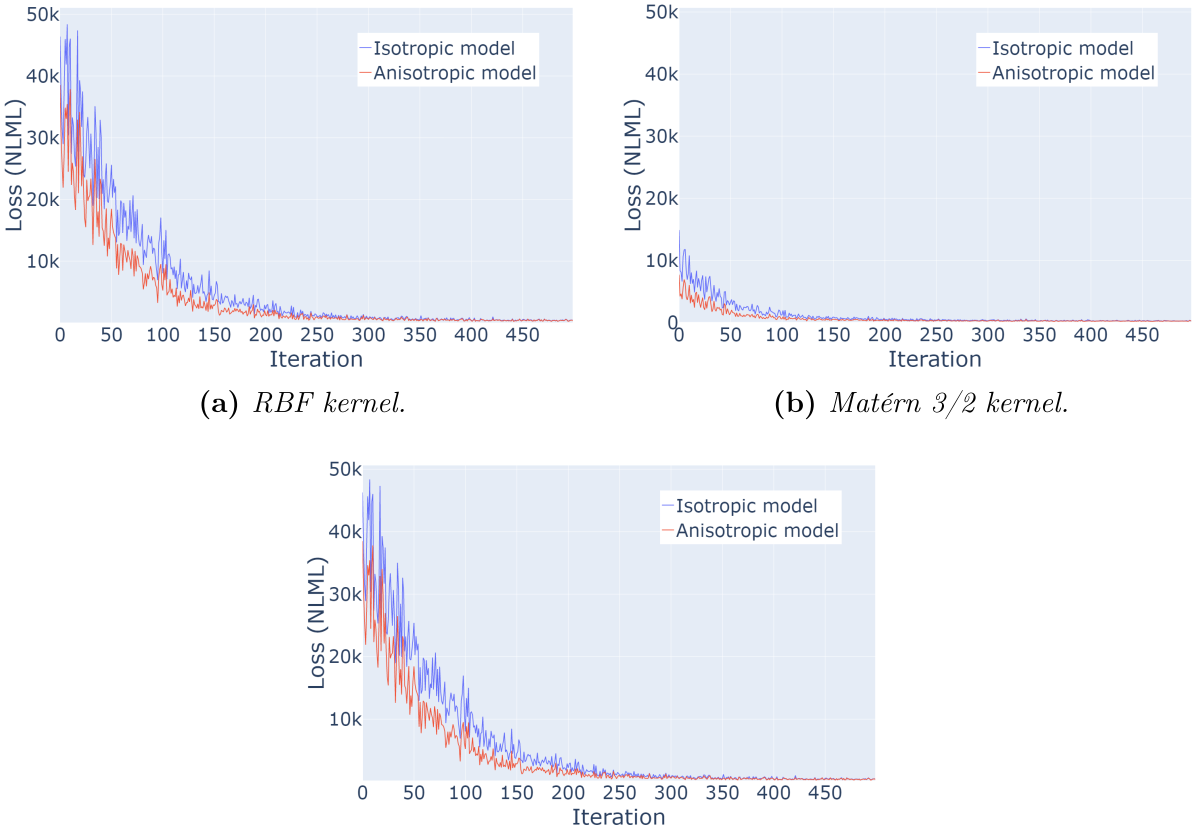
Figure 5.2.1 shows the loss functions for the six models trained on the Bærum E18 dataset. The loss functions for all six models decreased during training, indicating that the models were learning and improving their predictions. The anisotropic Matérn 3/2 model yielded the lowest loss values.

(**b**) *Matérn 3/2 kernel*

Hyperparameter	**Prior**	**Posterior**
Isotropy		
Amplitude σ	1.0	3.592
Length scale ℓ	1.0	0.271
Observation noise variance σ_y	0.001	0.002
Geometric anisotropy		
Amplitude σ	1.0	3.082
Length scale 1 ℓ_1	1.0	3.181
Length scale 2 ℓ_2	2.0	6.466
Angle θ	1.57	1.263
Observation noise variance σ_y	0.001	0.002

(**c**) *Linear + RBF kernel*

Hyperparameter	**Prior**	**Posterior**
Isotropy		
Bias amplitude σ_b	0	0
Slope amplitude σ_s	1.0	1.611
Shift c	0	5.970
Amplitude σ	1.0	7.234
Length scale ℓ	1.0	0.122
Observation noise variance σ_y	0.001	0.003
Geometric anisotropy		
Bias amplitude σ_b	0	0
Slope amplitude σ_s	1.0	1.216
Shift c	0	6.974
Amplitude σ	1.0	5.723
Length scale 1 ℓ_1	1.0	6.419
Length scale 2 ℓ_1	2.0	11.527
Angle θ	1.57	1.695
Observation noise variance σ_y	0.001	0.002



(a) *RBF kernel.*

(b) *Matérn 3/2 kernel.*

(c) *Linear + RBF kernel.*

Figure. 5.2.1: *The NLML for six models trained on the Bærum E18 dataset. The blue and red lines represent the loss during training for the isotropic and anisotropic models, respectively. The x-axis displays the NLML value, while the y-axis represents the number of iterations.*

5.2.2 Metric Results

Table 5.2.2 presents the results of the metrics used to evaluate the performance of the six models trained on the Bærum E18 dataset. The anisotropic model with the Matérn 3/2 kernel achieved the best performance, with an R^2 score of 0.954, RMSE of 0.211, and a CRPS of 0.078. These results suggest that this model is able to accurately predict the DTB values for the given dataset.

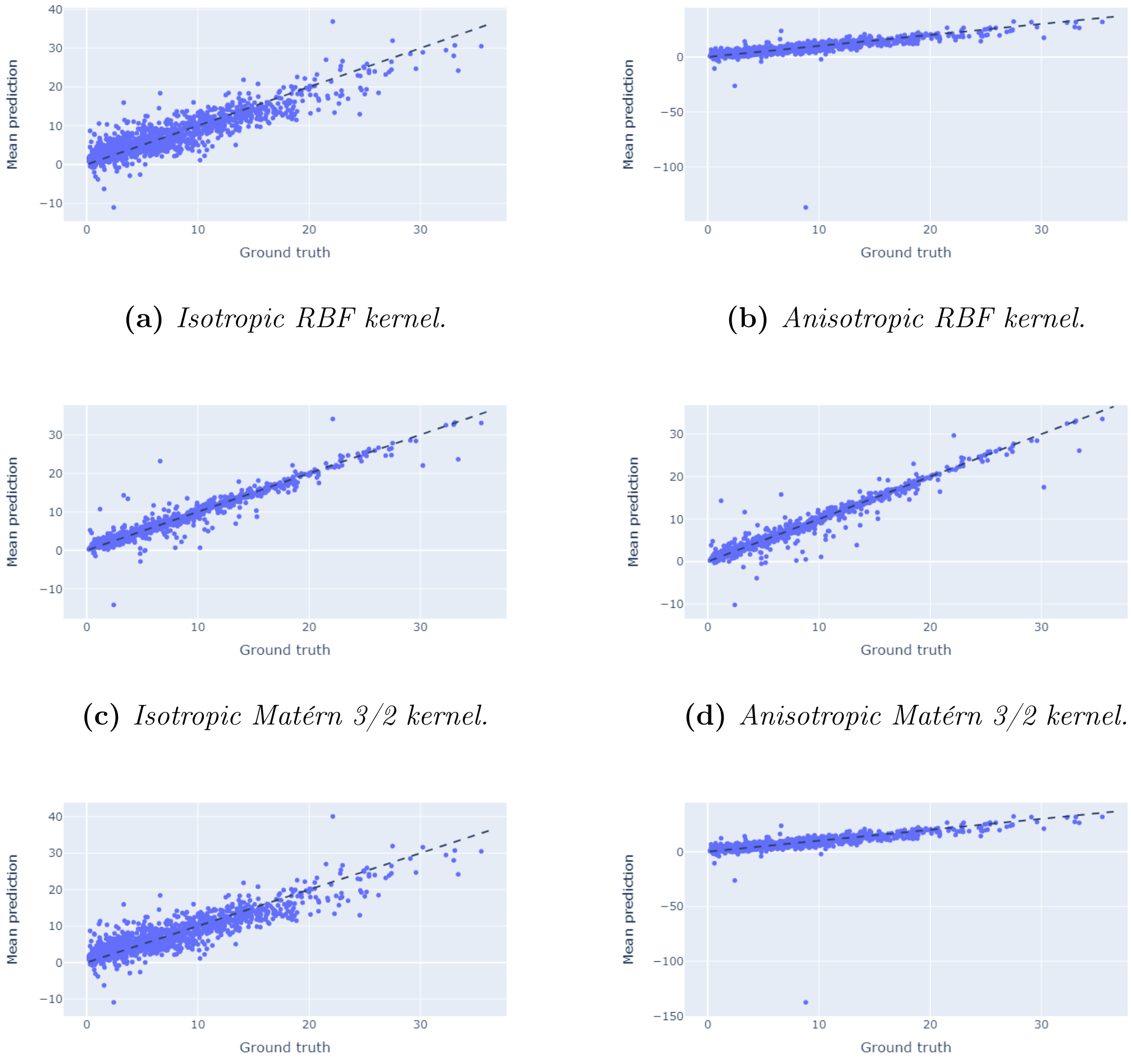
5.2.3 Best Model Results

This section presents the interpolated field and associated uncertainties for the best-performing model trained on the Bærum E18 dataset. Based on the metrics results presented in Table 5.2.2, the best model performance is obtained by the anisotropic model with the Matérn 3/2 kernel. The results are visualized in Figure 5.2.3.

Table 5.2.2: *Metric results for Bærum E18 dataset. R^2 is the coefficient of determination, Root Mean Squared Error (RMSE), and Continuous Rank Probability Score (CRPS).*

Kernel	Metric	Isotropy	Geometric anisotropy
RBF	R^2	0.812	0.471
	RMSE	0.428	0.720
	CRPS	0.303	0.298
Matérn 3/2	R^2	0.943	0.954
	RMSE	0.235	0.211
	CRPS	0.093	0.078
Linear + RBF	R^2	0.809	0.468
	RMSE	0.432	0.721
	CRPS	0.303	0.298

Visualizations of the interpolation and uncertainty results for all models are provided in Appendix C. The maps are generated the same way as described in Section 5.1.3.



(a) *Isotropic RBF kernel.*

(b) *Anisotropic RBF kernel.*

(c) *Isotropic Matérn 3/2 kernel.*

(d) *Anisotropic Matérn 3/2 kernel.*

(e) *Isotropic Linear and RBF kernel.*

(f) *Anisotropic Linear and RBF kernel.*

Figure. 5.2.2: *Scatter plots of the mean prediction values (y-axis) versus ground truth values (x-axis) obtained from the six Gaussian Process for Regression (GPR) models trained with six different kernels on the Bærum E18 dataset. All values are in meters.*

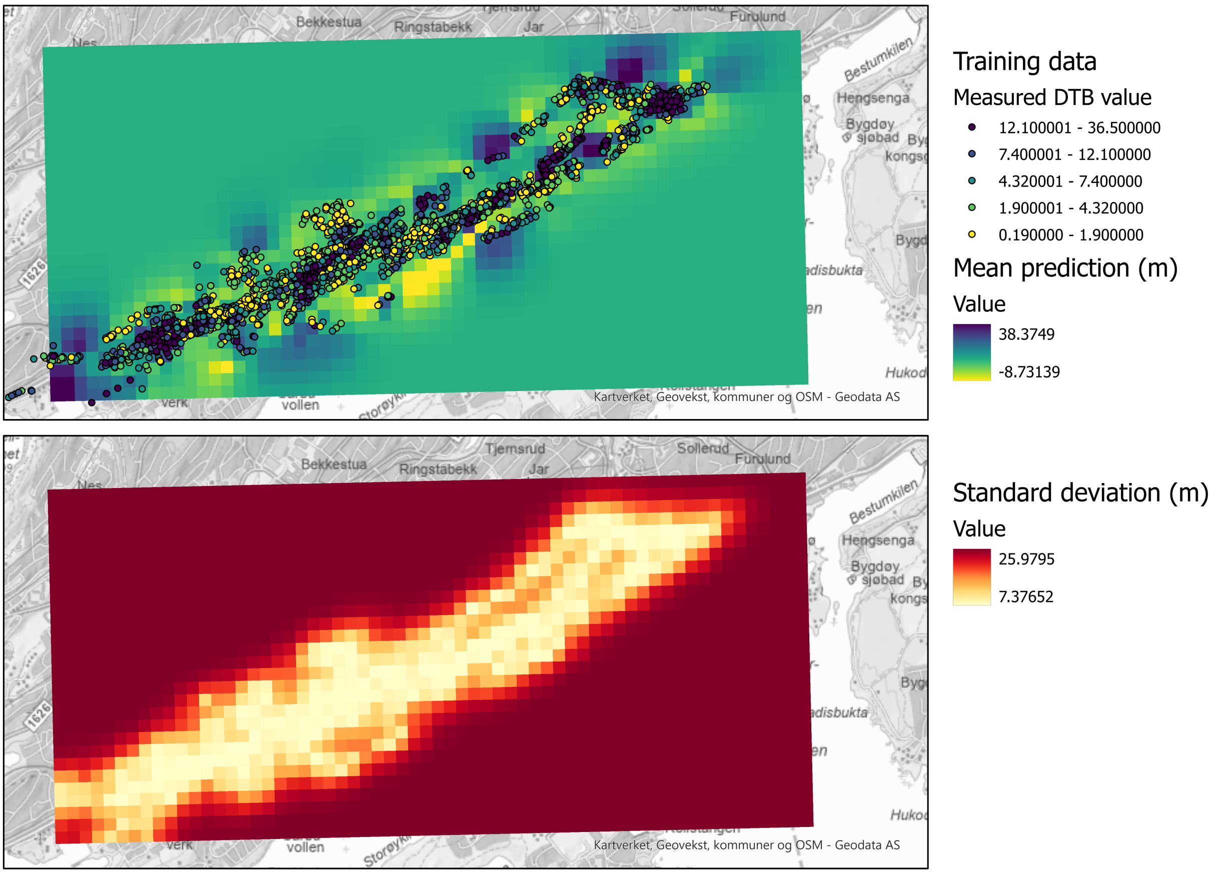


Figure. 5.2.3: *Interpolation and uncertainty for anisotropic model with Matérn 3/2 kernel trained on Bærum E18 dataset.*

Chapter 6

Discussion

The preceding chapters have presented the related theory, methodology, and results of this study, which aimed to investigate the effectiveness of GP models in predicting DTB values for geotechnical investigations, as well as the possible benefits of using anisotropic models for random effects in geostatistical settings. In this chapter, the implications of these findings and their potential contributions to the field of geostatistics, machine learning, and geotechnics will be discussed.

First, we will examine the quality of the data used in this study. Then, we will discuss the choice of kernels and hyperparameters and how they influenced the model performance. Next, we will go into the details of the model performance and analyze the strengths and limitations of the models in predicting DTB values. After discussing the model performance, we will compare our results to related works presented in Chapter 1. Finally, we will address the remaining challenges and future work.

6.1 Data Quality

The datasets used in this study have generally high quality. Both datasets contain information about boreholes from geotechnical investigations conducted by NGI. The geotechnical data is considered to be of high quality due to the strict quality control procedures implemented by NGI during data acquisition. It is worth noting, however, that the accuracy of the borehole data acquired before the year 2010 may be compromised as the GPS position of these coordinates could be offset from the actual borehole locations.

When pre-processing the data, we found missing values in the Oslo East dataset. These values were removed from the dataset, as we had enough observations available. However, missing values are common in many real-life datasets, and imputation techniques can be used to handle them. The framework used in this thesis is not equipped to handle missing values, but future work could explore incorporating imputation techniques into the GP models.

It is important to keep in mind that while the geotechnical investigation data used in this study is of high quality, it is still subject to some degree of uncertainty and variability, which can affect the accuracy of our interpolated depth estimates. The interpolation results in this study revealed irregular bedrock morphology in the Oslo area, indicating significant variation in sediment thickness.

6.2 Choice of Kernel and Hyperparameters

The choice of kernel and hyperparameters is an important aspect of GPs that can significantly impact the performance of the model. The kernel choice in a GP model forms the prior assumption about the underlying function which is the basis for how the function will further learn from the data. In our study, we experimented with several kernels, including the RBF, the Matérn 3/2, and a combined kernel with linear and RBF components.

The choice of the kernel in our study was based on literature that described the behaviours of the different kernels [18, 68, 82]. The RBF kernel assumes smooth variations in the data, while the Matérn 3/2 kernel captures both smooth and abrupt variations. The combined kernel, on the other hand, was used to capture both linear and nonlinear relationships between the variables.

The depth of the boreholes used in our study varied significantly, resulting in the Bærum E18 dataset having a higher degree of anisotropy which was clear in the northeast-southwest direction. The Oslo East dataset had more evenly distributed borehole depths in all directions resulting in a smaller degree of anisotropy. To assess the effectiveness of different kernels in capturing these varying data structures, we experimented with a combination of linear and RBF kernels. This was done primarily to evaluate whether the kernel could accurately capture any linear trends present in the datasets.

In addition to the kernel combination, we also explored modifications that allowed us to include anisotropy. By incorporating this modification, we aimed to understand better how the model was affected by variations in the orientation and structure of the data points.

The results of the hyperparameter optimizations in Tables 5.1.1 and 5.2.1 indicate that models with anisotropic kernels are capable of identifying the major axis of the input data. This can be observed by examining length scale parameters in the different models after training.

For instance, the optimization process revealed that the length scale ℓ parameter for all isotropic models trained on both datasets was very low, below 1. While in the anisotropic models, the posterior values for the length scale parameters for two directions ℓ_1 and ℓ_2, yielded higher values even though the prior value for the ℓ_1 parameter was set to the same value as in the prior value of the length scale ℓ parameter in the isotropic models. By inverse transforming the values for ℓ_1 and ℓ_2 with the standardization method used for the observations we can obtain the optimized length scale parameters in meters, which can give us a better indication of at what length in the different directions the points in the datasets are spatially correlated with each other.

The average posterior value for ℓ_1 and ℓ_2 from the anisotropic models trained on the Oslo East dataset is approximately 3 and 6 respectively. These values in meters after inverse transformation yield that the average minor axis has a length of around 55 meters and the major axis has a length of around 105 meters. For the Bærum E18 dataset, we have that the average ℓ_1 and ℓ_2 posterior values are 5 and 10 respectively. The minor axis is approximately 86 meters and the major axis is around 148 meters. From these results, we can interpret that there is slightly more anisotropy detected in the Bærum E18 dataset compared to the Oslo East dataset. However, from the semivariogram maps in Figure 3.3.2 we see that the degree of anisotropy in the Bærum E18 dataset should be much larger than what the optimized hyperparameter results show.

It is worth noting that the choice of kernel and hyperparameters can be highly dependent on the specific problem being addressed. In some cases, more complex kernels may be needed to capture the underlying relationships, while in other cases, simpler kernels may be sufficient. This is also why the prior distribution for the hyperparameters actually can play a part in how the posterior predictive function turns out. The kernel functions used for this project assume that the target variable can be modelled as a Gaussian distribution, which is not true according to the data distributions presented in Figure 4.2.2, we will go into more detail about this in Section 6.3. By choosing a prior distribution based on what we priorly know about the input data and with the help from experts like geologists we can ensure that the chosen kernel and hyperparameters are more appropriate for the specific problem we want to model.

6.3 Model Performance

The models in this study were trained using a batch training approach, where the datasets were partitioned into batches and trained in 500 iterations. As shown in Figure 5.1.1 and Figure 5.2.1, the loss functions demonstrate that the models were able to learn relatively quickly from the data. Specifically, the models using the RBF kernel and the combined kernel with linear and RBF kernels reached a steady-state loss value after approximately 250 iterations, while the models using the Matérn 3/2 kernel converged much earlier at around 150 iterations. These observations were consistent for both datasets. Moreover, the Negative Log Marginal Likelihood (NLML) values for the models using the Matérn 3/2 kernel were significantly lower compared to the other models, suggesting that the Matérn 3/2 kernel was more efficient in capturing the underlying pattern of the data.

The models in this study had a relatively fast training time, despite being trained locally on a standard laptop. For instance, the models trained on the Oslo East dataset with around 2000 data points had an average training time of 4 seconds, while the average training time for the Bærum E18 dataset with approximately 8000 data points was around 35 seconds. However, when predicting the posterior mean values over a high-resolution field, a challenge was discovered. The interpolation results shown in Figure 5.1.3 and Figure 5.2.3 were obtained using a grid of size 100 times 100, which equals 10,000 data points. This resolution may not be considered high, particularly when dealing with complex or detailed data. However, when attempting to predict over a grid with a larger resolution, such as 100 000 data points, the model failed due to insufficient RAM. The reason for this failure is that when estimating over a high-resolution field using a GP model, the size of the covariance matrix increases rapidly. For an $n \times n$ matrix, the computational complexity of computing the inverse is $O(n^3)$, which becomes prohibitively expensive for large values of n, where n is the number of data points. This is known as the "curse of dimensionality" problem [7].

Table 5.1.2 and Table 5.2.2 present the metric results obtained after running the posterior prediction function over the test subsets for each model in the two datasets. The results for the Oslo East dataset, shown in Table 5.1.2, indicate that the Matérn 3/2 kernel had the best performance for both isotropic and anisotropic models. The R^2 scores for the Matérn 3/2 kernel were 0.755 and 0.829 for the isotropic and anisotropic models respectively. The results for the remaining two kernels were nearly identical to each other, suggesting that combining the RBF kernel with the linear kernel did not yield significant improvements compared to using the RBF kernel alone. By doing some further research it was found that the linear kernel is never more accurate than a tuned RBF kernel, hence it is not necessary to combine these two together as the RBF kernel will outperform the linear kernel as shown in the results of this study [40].

Table 5.2.2 presents the metric results for the Bærum E18 dataset. Once again, the Matérn 3/2 kernel performed the best among all the models, with R^2 of 0.943 for the isotropic model and 0.954 for the anisotropic model. Moreover, the Continuous Rank Probability Score (CRPS) for these models was also quite low, indicating a good match between the predictive functions and the observed outcomes. However, it's important to note that the dataset has a higher point density than the Oslo East dataset, and the points are more clustered and very closely located. This can result in high goodness of fit values that may be a sign of overfitting, where the model is fitting the noise in the data rather than the underlying pattern.

The metrics results for the RBF kernel and the combined kernel models (Table 5.2.2) were almost identical to each other, similar to the results for the Oslo East dataset. As previously mentioned, the Bærum E18 dataset has a higher degree of anisotropy compared to the Oslo East dataset, so it was expected that the anisotropic models would perform better. However, the metrics results for the anisotropic models using RBF and the combined kernel were below 0.471, while the isotropic models performed much better with an R^2 of around 0.800. This suggests that the anisotropic models did not perform as well as expected on the Bærum E18 dataset despite the higher degree of anisotropy and that the isotropic models generally are more stable than the anisotropic models. Such low R^2 values for the anisotropic models indicate that they are performing worse than predicting the mean value. This makes them unreliable and unfit for modelling the DTB values, as the uncertainty related to the predictions is too high.

To visualize the metrics results, scatter plots that show the mean prediction values and the actual values were created. These are shown in Figure 5.1.2 and Figure 5.2.2. The subfigures in Figure 5.1.2 for the Oslo East dataset show that there are very few outliers and that most of the predictions are consistent with the observed output values. However, in subfigure 5.2.2b and subfigure 5.2.2f in Figure 5.2.2 for the Bærum E18 dataset, there is a very significant outlier that has a large negative value. This means that there are negative values present in the Bærum E18 dataset which was used in the training processes. This error was not discovered until later in the study, thus might be a large reason for one of the biggest challenges in this study where the models predict negative values. However, when checking the Oslo East dataset for negative values in the target values used for model training, we find none.

The best-performing model for both datasets is the anisotropic model using the Matérn 3/2 kernel. Figure 5.1.3 and Figure 5.2.3 show the interpolation results and the uncertainty associated with the predictions for this model for the Oslo East dataset and the Bærum E18 dataset. These figures demonstrate a common trait in all models where the predictions become more uncertain the farther away they are from the observed data points. At these locations, the model typically predicts the mean value of the expected outcome, which is a typical behaviour of a GPR. This is because the locations closer to the observed data points are expected to have similar properties and outcomes to the observed data points.

Another trait is that all six models predict negative values, even though all observations are above 0. The isotropic and anisotropic models using the Matérn 3/2 kernel predict much lower negative DTB values compared to the remaining four models. This is a challenge that needs to be addressed. As mentioned above, there was a negative value present in the Bærum E18 dataset, but none in the Oslo East dataset, which indicates that the reason for the models predicting negative values can be influenced by something else.

In Chapter 4, we mentioned that we assume all prior hyperparameters are based on Gaussian distribution and that the hyperparameters were constrained to be positive. However, the distribution of the input data as illustrated in Figure 4.2.2 shows that the target variable has a right-skewed distribution. In such cases, the target values can be transformed to remove the skewness in the data and approximate a Gaussian distribution, for example, by using a log-transformation. However, these types of transformations should be applied cautiously as the results of statistical methods conducted on data that has been log-transformed are often not applicable to the initial, untransformed data [20]. It is also possible to rather set the prior hyperparameter values so it fits the distribution of the input data better, but this needs to be tested.

6.4 Comparison to Related Works

Shen and Gelfand [79] conducted a similar study that also explored the use of anisotropic kernel functions within the Bayesian framework. They used the Matérn kernel and an exponential kernel where one of the metrics used to validate the model performances was CRPS. Comparing our study's best-performing model, the anisotropic model with Matérn 3/2 kernel, to Shen and Gelfand [79]'s models trained on real data from $PM_{2.5}$ monitoring stations, we observed that our model yielded lower CRPS values. However, it is essential to consider various factors when comparing the results of the two studies, including the differences in the datasets, the experimental setup, and the choice of hyperparameters. Additionally, it is important to consider the specific goals and objectives of each study. While both studies focus on using Bayesian methods with anisotropic kernel functions, they have different purposes. Shen and Gelfand [79] aimed to predict $PM_{2.5}$ concentration, while our study's primary objective was to predict DTB.

In Chapter 1, we mentioned a couple of other studies based on standard machine learning models [78, 94]. These studies mainly utilized tree-based ensemble models to estimate the DTB for a larger field than what we interpolated in our study. In contrast, we used a regression-based approach, which makes it difficult to directly compare our results to those of Shangguan et al. [78] and Yan et al. [94]. Although our best-performing models have higher R^2 scores, it is important to consider many factors when comparing results from different studies. The differences in application scenarios, datasets, and experimental setups can all impact the model performances, making direct comparisons between studies challenging.

6.5 Remaining Challenges and Future Work

The work presented in this study focuses on the choice of kernels in Gaussian Processes (GPs) and the impact these kernels have on the model performance. The study explores several kernel choices, including the Radial Basis Function (RBF), Matérn 3/2, and a kernel combination of linear and RBF components. The anisotropy of the data is also taken into account by incorporating modifications that allow for anisotropy. The results of the hyperparameter optimizations show that models with anisotropic kernels are capable of identifying the major axis of the input data, and the length scale parameters provide insight into the spatial correlation of the data points. However, the degree of anisotropy detected by the hyperparameter optimizations may not reflect the actual degree of anisotropy in the data. Additionally, the models using anisotropic kernel functions were less stable when compared to the isotropic models.

One of the remaining challenges in the models of this study is the optimization of hyperparameters, which can be a computationally demanding task. To tackle this issue, Markov Chain Monte Carlo (MCMC) and sampling methods can be utilized to optimize hyperparameters and reduce computational costs [68]. Another challenge is selecting appropriate kernels and hyperparameters for specific problems. For instance, in this study, we attempted to use a combined kernel, but the combination of the RBF kernel and the linear kernel did not yield the desired results. Nonetheless, there are several other kernels that can be combined and tested to uncover patterns and structures in spatial data that cannot be detected by a single kernel.

The selection of kernels and hyperparameters can be highly dependent on the problem at hand. The choice of the prior distribution for the hyperparameters influences the posterior predictive function. Expert knowledge and selecting a prior distribution that fits the data being modelled can help overcome this challenge.

As mentioned in Section 2.5, another essential component of GPs beside the kernel function is the mean function. In this study, the mean function was set to 0 as GPs are able to model the mean from an arbitrary value [68]. However, a mean function that is able to detect the mean value of the observations locally based on the location in the input domain could possibly reduce the uncertainties related to the predictions in areas farther away from observations.

Another recommendation for future work is to explore the underlying causes for the negative predictions generated by the models. One possible reason could be the mismatch between the distribution of the prior function and the data distribution. To address this, one could explore options such as log transforming the target values or adjusting the prior values of the hyperparameters to better align with the data distribution.

The data used in this study only included longitude (x), latitude (y), and the DTB values, without taking surface topography into consideration. Chapter 2 mentions the easy accessibility of remote sensing data nowadays, which can provide more accurate and comprehensive data about surface topography. The models used in this study directly incorporated DTB values, which may not be the most realistic representation of the terrain height in areas with significant construction work or other man-made changes to the ground surface.

One potential solution to address this issue is to subtract the terrain height obtained from a DTM from the DTB values to obtain the bedrock topography. This approach may lead to a more realistic interpolation of the DTB by taking the actual terrain into consideration. Alternatively, the anisotropic implementation can be scaled to a 3D problem, where the terrain height can be included as an additional input feature, along with the x and y coordinates. Incorporating quaternary geological maps into the model can also provide valuable geological information to improve the quality of the DTB estimates.

To conclude, further research should investigate combining different kernel functions, explore the use of mean functions, identify the underlying causes of negative predictions, and incorporate remote sensing data and geological maps into the framework.

Chapter 7

Conclusion

The main objective of this thesis was to develop a machine learning framework for estimating Depth to Bedrock (DTB) in areas of interest using Gaussian Processes (GPs). To achieve this goal, several kernel functions, including Radial Basis Function (RBF), Matérn 3/2, and combined kernels with linear and RBF functions, were evaluated in conjunction with GPs. Additionally, the impact of implementing anisotropy in the kernels was evaluated. While the implementation of anisotropy resulted in a slight improvement in the best-performing model, it was observed that the anisotropic models were more unstable compared to the isotropic models in the remaining experiments. Matérn 3/2 kernel with anisotropic implementation performed the best, achieving the lowest CRPS and highest R^2 score in estimating DTB.

Future research can focus on optimizing hyperparameters, selecting appropriate kernels, exploring the use of mean functions, and incorporating additional geological and remotely sensed data to improve DTB estimation and to create a more robust framework. Further investigation should also address the issue of models estimating large negative values and the effects of non-Gaussian variables.

In conclusion, this thesis presents a flexible framework for estimating DTB using GPs, which not only demonstrates the importance of selecting an appropriate kernel function and optimizing its hyperparameters but also provides a basis for future research in this area.

References

[1] Abramowitz, M. and Stegun, I.A. *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables.* Applied mathematics series. Dover Publications, 1965. ISBN: 9780486612720.

[2] Agterberg, Frederik P. "GEORGES MATHERON: FOUNDER OF SPATIAL STATISTICS". In: *Earth Sciences History* 23.2 (2004), pp. 325–334. ISSN: 0736623X, 19446187. URL: http://www.jstor.org/stable/24137099 (visited on 04/06/2023).

[3] Allaby, Michael. *A Dictionary of Geology and Earth Sciences.* 4th ed. Oxford University Press, 2013, p. 416. ISBN: 978-0-19-965306-5. DOI: 10.1093/acref/9780199653065.001.0001.

[4] Armstrong, M. *Basic Linear Geostatistics.* Springer Berlin Heidelberg, 1998. ISBN: 9783540618454.

[5] Armstrong, Margaret and Diamond, Phil. "Testing Variograms for Positive-Definiteness". In: *Mathematical Geology* 16 (4 1984), pp. 407–421.

[6] Bárdossy, András. "Introduction to geostatistics". In: *Institute of Hydraulic Engineering, University of Stuttgart* (1997). URL: https://osf.io/6jzpn/download.

[7] Belyaev, Mikhail, Burnaev, Evgeny, and Kapushev, Yermek. "Exact Inference for Gaussian Process Regression in case of Big Data with the Cartesian Product Structure". In: (2014). arXiv: 1403.6573.

[8] Biswas, Asim and Si, Bing Cheng. *A typical example of semivariogram showing different components.* This work is licensed under the Creative Commons Attribution 3.0 Unported license. To view a copy of this license, visit https://creativecommons.org/licenses/by/3.0/deed.en. Feb. 2012. URL: https://www.intechopen.com/chapters/39857.

[9] Canarache, A., Vintila, I.I., and Munteanu, I. *Elsevier's Dictionary of Soil Science: Definitions in English with French, German, and Spanish word translations.* Elsevier Science, 2006, p. 93. ISBN: 978-0-08-056131-8.

[10] Cdipaolo96. *Gaussian Process Regression (prediction) with a squared exponential kernel. Left are draws from the prior distribution of functions. Middle are draws from the posterior predictive distribution. Right is the mean prediction with 1 standard deviation shaded.* This work is licensed under the Creative Commons Attribution-Share Alike 4.0 International license. To view a copy of this license, visit `https://creativecommons.org/licenses/by-sa/4.0/deed.en`. Mar. 2016. URL: `https://www.intechopen.com/chapters/39857`.

[11] Chilès, J.P. and Delfiner, P. *Geostatistics: Modeling Spatial Uncertainty.* 2nd ed. Wiley Series in Probability and Statistics. Wiley, 2012. ISBN: 9780470183151.

[12] Chollet, F. *Deep Learning with Python, Second Edition.* 2nd ed. Manning, 2021. ISBN: 978-1-61729-686-4.

[13] Costa-Luis, Casper da et al. *tqdm: A fast, Extensible Progress Bar for Python and CLI.* Version v4.65.0. Mar. 2023. DOI: `10.5281/zenodo.7697295`.

[14] Cressie, Noel AC. *Statistics for spatial data.* Wiley Series in Probability and Statistics. John Wiley & Sons, New York, 1993, p. 64. ISBN: 9780471002550.

[15] Deutsch, C V and Journel, A G. *GSLIB: Geostatistical Software Library and User's Guide Second Edition.* Oxford University Press, 1997, p. 55. ISBN: 9780195100150.

[16] Dillon, Joshua V. et al. *TensorFlow Distributions.* 2017. arXiv: `1711.10604 [cs.LG]`.

[17] Duchi, John, Hazan, Elad, and Singer, Yoram. "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of machine learning research* 12.7 (2011), pp. 2121–2159.

[18] Duvenaud, David. "Automatic model construction with Gaussian processes". PhD thesis. University of Cambridge, 2014.

[19] Encyclopaedia Britannica, The Editors of. "weathering". In: *Encyclopedia Britannica* (2023). URL: `https://www.britannica.com/science/weathering-geology` (visited on 03/13/2023).

[20] Feng, Changyong et al. "Log-transformation and its implications for data analysis". In: *Shanghai Archives of Psychiatry* 26 (2 Apr. 2014), p. 105. ISSN: 10020829. DOI: `10.3969/J.ISSN.1002-0829.2014.02.009`.

[21] Firouzianbandpey, S. et al. "Effect of Spatial Correlation Length on the Interpretation of Normalized CPT Data Using a Kriging Approach". In: *Journal of Geotechnical and Geoenvironmental Engineering* 141.12 (2015), p. 04015052. DOI: `10.1061/(ASCE)GT.1943-5606.0001358`.

[22] Fouedjio, Francky and Klump, Jens. "Exploring prediction uncertainty of spatial data in geostatistical and machine learning approaches". In: *Environmental Earth Sciences* 78 (1 2019), pp. 1–24. ISSN: 18666299. DOI: `10.1007/S12665-018-8032-Z/FIGURES/19`.

[23] Géron, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2017. ISBN: 9781491962299.

[24] Giusti, Domenico et al. "Recursive anisotropy: a spatial taphonomic study of the Early Pleistocene vertebrate assemblage of Tsiotra Vryssi, Mygdonia Basin, Greece". In: *Boreas* 48 (3 July 2019), pp. 713–730. ISSN: 15023885. DOI: 10.1111/BOR.12368.

[25] Gneiting, Tilmann and Raftery, Adrian E. "Strictly Proper Scoring Rules, Prediction, and Estimation". In: *Journal of the American Statistical Association* 102.477 (2007), pp. 359–378. DOI: 10.1198/016214506000001437.

[26] Gregory, Kenneth J. *The earth's land surface: Landforms and processes in geomorphology*. SAGE Publications, 2010, pp. 24–29. DOI: 10.4135/9781446251621.

[27] Handcock, Mark S. and Wallis, James R. "An Approach to Statistical Spatial-Temporal Modeling of Meteorological Fields". In: *Journal of the American Statistical Association* 89.426 (1994), pp. 368–378. DOI: 10.1080/01621459.1994.10476754.

[28] Harris, Charles R. et al. "Array programming with NumPy". In: *Nature* 585 (2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2.

[29] Haugen, E et al. "A preliminary attempt towards soil classification chart from total sounding". In: *17th Nordic Geotechnical Meeting* (2016).

[30] Heuvelink, Gerard B.M. and Webster, Richard. "Spatial statistics and soil mapping: A blossoming partnership under pressure". In: *Spatial Statistics* 50 (Aug. 2022), p. 100639. ISSN: 2211-6753. DOI: 10.1016/J.SPASTA.2022.100639.

[31] Holtz, Robert D., Kovacs, William D., and Sheahan, Thomas C. *An Introduction to Geotechnical Engineering*. 3rd ed. Pearson, 2022, pp. 1–3. ISBN: 978-0-13-760438-8.

[32] Horning, N. et al. *Remote Sensing for Ecology and Conservation: A Handbook of Techniques*. Oxford biology. OUP Oxford, 2010, p. 39. ISBN: 9780199219957.

[33] Hu, Zhiting, Yen, Chao-Ming, and Huang, Biwei. *24 : Gaussian Process and Deep Kernel Learning*. https://www.cs.cmu.edu/~epxing/Class/10708-17/notes-17/10708-scribe-lecture24.pdf. Accessed: 2023–05-08. 2017.

[34] Inc., Esri. *ArcGIS Pro 3.0.2*. 2022. URL: https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview.

[35] Inc., Plotly Technologies. *Collaborative data science*. 2015. URL: https://plot.ly.

[36] Jafrasteh, Bahram et al. "Gaussian processes for missing value imputation". In: *Knowledge-Based Systems* (2022). URL: https://doi.org/10.48550/arXiv.2204.04648.

[37] Johnson, C. R. "Positive Definite Matrices". In: *The American Mathematical Monthly* 77.3 (1970), pp. 259–264. ISSN: 00029890, 19300972. DOI: 10.2307/2317709.

[38] Journel, A G and Huijbregts, C J. *Mining Geostatistics.* Academic Press, 1978. ISBN: 0-12-391050-1.

[39] Kartverket. *Høydedata og dybdedata.* Aug. 2022. URL: `https://kartverket.no/api-og-data/terrengdata` (visited on 03/02/2023).

[40] Keerthi, S. Sathiya and Lin, Chih-Jen. "Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel". In: *Neural Computation* 15.7 (2003), pp. 1667–1689. DOI: `10.1162/089976603321891855`.

[41] Killinger, D.K. *Lidar (light detection and ranging).* Elsevier, 2014, pp. 292–293. DOI: `10.1533/9780857098733.2.292`.

[42] Kingma, Diederik P. and Ba, Jimmy. "Adam: A Method for Stochastic Optimization". In: (2014). DOI: `https://doi.org/10.48550/arXiv.1412.6980`.

[43] Krige, D.G. "A statistical approach to some basic mine valuation problems on the Witwatersrand". In: *Journal of the Southern African Institute of Mining and Metallurgy* 52.6 (1951), pp. 119–139. DOI: `10.10520/AJA0038223X\_4792`. URL: `https://journals.co.za/doi/pdf/10.10520/AJA0038223X_4792`.

[44] Kumar, Sandeep et al. "Remote sensing for agriculture and resource management". In: *Natural Resources Conservation and Advances for Sustainability* (2022), pp. 99–100. DOI: `10.1016/B978-0-12-822976-7.00012-0`.

[45] Lane, John W. et al. "Estimation of bedrock depth using the horizontal-to-vertical (H/V) ambient-noise seismic method". In: *Symposium on the Application of Geophysics to Engineering and Environmental Problems* 2 (2008), pp. 1247–1259. DOI: `10.4133/1.2963289`.

[46] Lemaréchal, Claude. "Cauchy and the gradient method". In: *Doc Math Extra* 251.254 (2012), p. 10. DOI: `https://doi.org/10.4171/dms/6/27`.

[47] Li, J. et al. "Probabilistic identification of soil stratification". In: 66 (Dec. 2015), pp. 16–26. ISSN: 0016-8505. DOI: `https://doi.org/10.1680/jgeot.14.P.242`.

[48] Li, Y. J., Hicks, M. A., and Vardon, P. J. "Uncertainty reduction and sampling efficiency in slope designs using 3D conditional random fields". In: *Computers and Geotechnics* 79 (Oct. 2016), pp. 159–172. ISSN: 0266-352X. DOI: `10.1016/J.COMPGEO.2016.05.027`.

[49] Lohani, Bharat and Ghosh, Suddhasheel. "Airborne LiDAR Technology: A Review of Data Collection and Processing Systems". In: *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences* 87 (Nov. 2017), p. 568. DOI: `10.1007/s40010-017-0435-9`.

[50] Lysdahl, Asgeir Olaf Kydland. Apr. 20, 2023.

[51] Mälicke, Mirko et al. *mmaelicke/scikit-gstat: A scipy flavoured geostatistical variogram analysis toolbox.* Version v0.6.0. May 2021. DOI: `10.5281/zenodo.4835779`.

[52] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[53] Matheron, Georges. *Les variables régionalisées et leur estimation: une application de la théorie de fonctions aléatoires aux sciences de la nature*. Thèses présentées à la Faculté des Sciences de l'Université de Paris. Masson et Cie, 1965. URL: https://books.google.no/books?id=4b0rAAAAYAAJ.

[54] Matheron, Georges. *Traité de géostatistique appliquée*. 14. Editions Technip, 1962, p. 333.

[55] McKinney, Wes et al. "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56.

[56] Mellbye, Håvard Skåra. "Gaussian Processes for long-term trajectory prediction using historical AIS data". MA thesis. Norwegian University of Science and Technology (NTNU), 2021, p. 87.

[57] Melo, José. "Gaussian processes for regression: a tutorial". In: *Technical Report* (2012).

[58] Müller, A.C. and Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 2016. ISBN: 9781449369415.

[59] NGI. *NGI - About NGI*. Feb. 2023. URL: https://www.ngi.no/en/about-ngi/about-us/about-ngi/ (visited on 04/10/2023).

[60] Noack, Marcus M. and Sethian, James A. "Advanced stationary and nonstationary kernel designs for domain-aware Gaussian processes". In: *Communications in Applied Mathematics and Computational Science* 17.1 (Oct. 2022), pp. 131–156. DOI: 10.2140/camcos.2022.17.131.

[61] O'Sullivan, David and Unwin, David J. *Geographic Information Analysis: Second Edition*. 2010. ISBN: 9780470288573. DOI: 10.1002/9780470549094.

[62] Olsen, Lars et al. "Quaternary glacial, interglacial and interstadial deposits of Norway and adjacent onshore and offshore areas". In: *Quaternary Geology of Norway, Geological Survey of Norway Special Publication* 13 (2013), pp. 79–144.

[63] Pedregosa, F. et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[64] Pfaffhuber, A. A. et al. "Regional geotechnical railway corridor mapping using airborne electromagnetics". In: vol. 2. 2016.

[65] Popovs, Konrāds et al. "Bedrock surface topography of Latvia". In: *Journal of Maps* 18 (2 Dec. 2022), pp. 370–381. ISSN: 1744-5647. DOI: 10.1080/17445647.2022.2067011.

[66] Puri, Nitish, Prasad, Harsh Deep, and Jain, Ashwani. "Prediction of Geotechnical Parameters Using Machine Learning Techniques". In: *Procedia Computer Science* 125 (2018), pp. 509–517. ISSN: 18770509. DOI: 10.1016/J.PROCS.2017.12.066.

[67] Raschka, Sebastian and Mirjalili, Vahid. *Python Machine Learning: Machine Learning & Deep Learning with Python, Scikit-Learn and TensorFlow 2, Third Edition.* Packt Publishing Ltd, 2019. ISBN: 978-1-78995-575-0.

[68] Rasmussen, Carl Edward, Williams, Christopher KI, et al. *Gaussian processes for machine learning.* Vol. 1. Springer, 2006. ISBN: 978-3-540-23122-6. DOI: https://doi.org/10.1007/b100712.

[69] Resources, The Norwegian Water and (NVE), Energy Directorate. *Detection of brittle materials. Summary report with recommendation Natural hazards-infrastructure for floods and slides. Sub project 6 Quick clays.* 2016, pp. 11–20. URL: https://nve.brage.unit.no/nve-xmlui/handle/11250/2498557?show=full.

[70] Roelants, Peter. *peterroelants.github.io.* https://github.com/peterroelants/peterroelants.github.io/blob/main/notebooks/gaussian_process/gaussian-process-kernel-fitting.ipynb. 2023.

[71] Ruder, Sebastian. "An overview of gradient descent optimization algorithms". In: (2016). DOI: https://doi.org/10.48550/arXiv.1609.04747.

[72] Saez, Marc and Barceló, Maria A. "Spatial prediction of air pollution levels using a hierarchical Bayesian spatiotemporal model in Catalonia, Spain". In: *Environmental Modelling & Software* 151 (May 2022), p. 105369. ISSN: 1364-8152. DOI: 10.1016/J.ENVSOFT.2022.105369.

[73] Samuel, A. L. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: 10.1147/rd.33.0210.

[74] Schaetzl, R.J. and Thompson, M.L. *Basic Concepts: Soil Horizonation...the Alphabet of Soils.* Soils: Genesis and Geomorphology. Cambridge University Press, 2015, p. 30. ISBN: 9781107016934.

[75] Schilling, Nicolas, Wistuba, Martin, and Schmidt-Thieme, Lars. "Scalable Hyperparameter Optimization with Products of Gaussian Process Experts". In: *Machine Learning and Knowledge Discovery in Databases.* Ed. by Frasconi, Paolo et al. Cham: Springer International Publishing, 2016, pp. 33–48. ISBN: 978-3-319-46127-4.

[76] Schowengerdt, R.A. *Remote Sensing: Models and Methods for Image Processing.* Elsevier Science, 2006, pp. 1–3. ISBN: 9780080480589.

[77] Seeger, Matthias. "GAUSSIAN PROCESSES FOR MACHINE LEARNING". In: *International Journal of Neural Systems* 14.02 (2004). PMID: 15112367, pp. 69–106. DOI: 10.1142/S0129065704001899.

[78] Shangguan, Wei et al. “Mapping the global depth to bedrock for land surface modeling”. In: *Journal of Advances in Modeling Earth Systems* 9 (1 Mar. 2017), pp. 65–88. ISSN: 1942-2466. DOI: 10.1002/2016MS000686.

[79] Shen, Y. and Gelfand, A. E. “Exploring geometric anisotropy for point-referenced spatial data”. In: *Spatial Statistics* 32 (Aug. 2019), p. 100370. ISSN: 2211-6753. DOI: 10.1016/J.SPASTA.2019.100370.

[80] Singh, Ajit Pratap, Dhadse, Kunal, and Ahalawat, Jayant. “Managing water quality of a river using an integrated geographically weighted regression technique with fuzzy decision-making model”. In: *Environmental Monitoring and Assessment* 191 (6 June 2019). ISSN: 15732959. DOI: 10.1007/S10661-019-7487-Z.

[81] Sposito, Garrison. “soil”. In: *Encyclopedia Britannica* (Feb. 2023). URL: https://www.britannica.com/science/soil (visited on 03/13/2023).

[82] Stein, M.L. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Series in Statistics. Springer New York, 1999. ISBN: 9780387986296. DOI: 10.1007/978-1-4612-1494-6.

[83] Su, Guoshao et al. “Gaussian Process Machine-Learning Method for Structural Reliability Analysis”. In: *Advances in Structural Engineering* 17.9 (2014), pp. 1257–1270. DOI: 10.1260/1369-4332.17.9.1257.

[84] Take, W. A. et al. “Evaluation of landslide triggering mechanisms in model fill slopes”. In: *Landslides* 1 (3 Sept. 2004), pp. 173–184. ISSN: 1612-510X. DOI: 10.1007/s10346-004-0025-1.

[85] Tavazza, Francesca, DeCost, Brian, and Choudhary, Kamal. “Uncertainty Prediction for Machine Learning Models of Material Properties”. In: *ACS Omega* 6.48 (2021), pp. 32431–32440. DOI: 10.1021/acsomega.1c03752.

[86] Tieleman, Tijmen and Hinton, Geoffrey. “Lecture 6.5 - RMSProp, COURSERA: Neural networks for machine learning”. In: *University of Toronto, Technical Report* (2012).

[87] Tobler, Waldo R. “A Computer Movie Simulating Urban Growth in the Detroit Region”. In: *Economic Geography* 46 (1970), pp. 234–240. ISSN: 00130095, 19448287. DOI: 10.2307/143141.

[88] Uhlenbeck, G. E. and Ornstein, L. S. “On the Theory of the Brownian Motion”. In: *Phys. Rev.* 36 (5 Sept. 1930), pp. 823–841. DOI: 10.1103/PhysRev.36.823.

[89] Van Rossum, Guido and Drake, Fred L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[90] Virtanen, Pauli et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[91] Wackernagel, Hans. *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media, 2003. ISBN: 978-3-642-07911-5. DOI: 10.1007/978-3-662-05294-5.

[92] Wan-Mohamad, W. N.S. and Abdul-Ghani, A. N. “The use of geographic information system (GIS) for geotechnical data processing and presentation”. In: vol. 20. 2011. DOI: 10.1016/j.proeng.2011.11.182.

[93] Woodroffe, C. “Geomorphology”. In: *Encyclopedia of Ocean Sciences* (Jan. 2001), pp. 1142–1148. DOI: https://doi.org/10.1006/rwos.2001.0093.

[94] Yan, Fapeng et al. “Depth-to-bedrock map of China at a spatial resolution of 100 meters”. In: *Scientific Data 2020 7:1* 7 (1 Jan. 2020), pp. 1–13. ISSN: 2052-4463. DOI: 10.1038/s41597-019-0345-6.

[95] Zhang, J., Zhang, Limin M., and Tang, Wilson H. “Kriging numerical models for geotechnical reliability analysis”. In: *Soils and Foundations* 51 (6 2011). ISSN: 00380806. DOI: 10.3208/sandf.51.1169.

[96] Zhang, Zhihua and Moore, John C. “Remote Sensing”. In: *Mathematical and Physical Fundamentals of Climate Change* (2015), pp. 111–114. DOI: 10.1016/B978-0-12-800066-3.00004-8.

[97] Zimmerman, Dale L. “Another look at anisotropy in geostatistics”. In: *Mathematical Geology* 25 (4 May 1993), pp. 453–470. ISSN: 08828121. DOI: 10.1007/BF00894779.

Appendix A

Source Code and Software

Table A.0.1: *Source Code Links for GitHub.*

File Name	GitHub Link	Commit Hash
gaussian_process_regression.py	The main program file	84fec8a
geometric_anisotropy.py	Input transformed kernel for anisotropy	209c584
kernel_setup.py	Function for kernel selection	209c584
GitHub Repository: `https://github.com/marisha-g/M-DV-V2023`		

Table A.0.2: *Software and Python modules used in this study.*

Module/Software	Purpose of use	Version	Reference
Python	High level programming language	3.10.10	[89]
TensorFlow	System for large-scale machine learning	2.10.0	[52]
TensorFlow Probability	Probabilistic reasoning and statistical analysis	0.14.0	[16]
Plotly	Library for graphing and data presentation	5.9.0	[35]
NumPy	Scientific computing with Python	1.23.5	[28]
Pandas	Data manipulation and analysis	1.5.3	[55]
Scikit-learn	Machine learning library	1.2.1	[63]
SciPy	Scientific- and technical computing	1.9.3	[90]
tqdm	Progress bar	4.65.0	[13]
ArcGIS Pro	GIS software	3.0.2	[34]

Appendix B

Interpolation Results for Oslo East Dataset

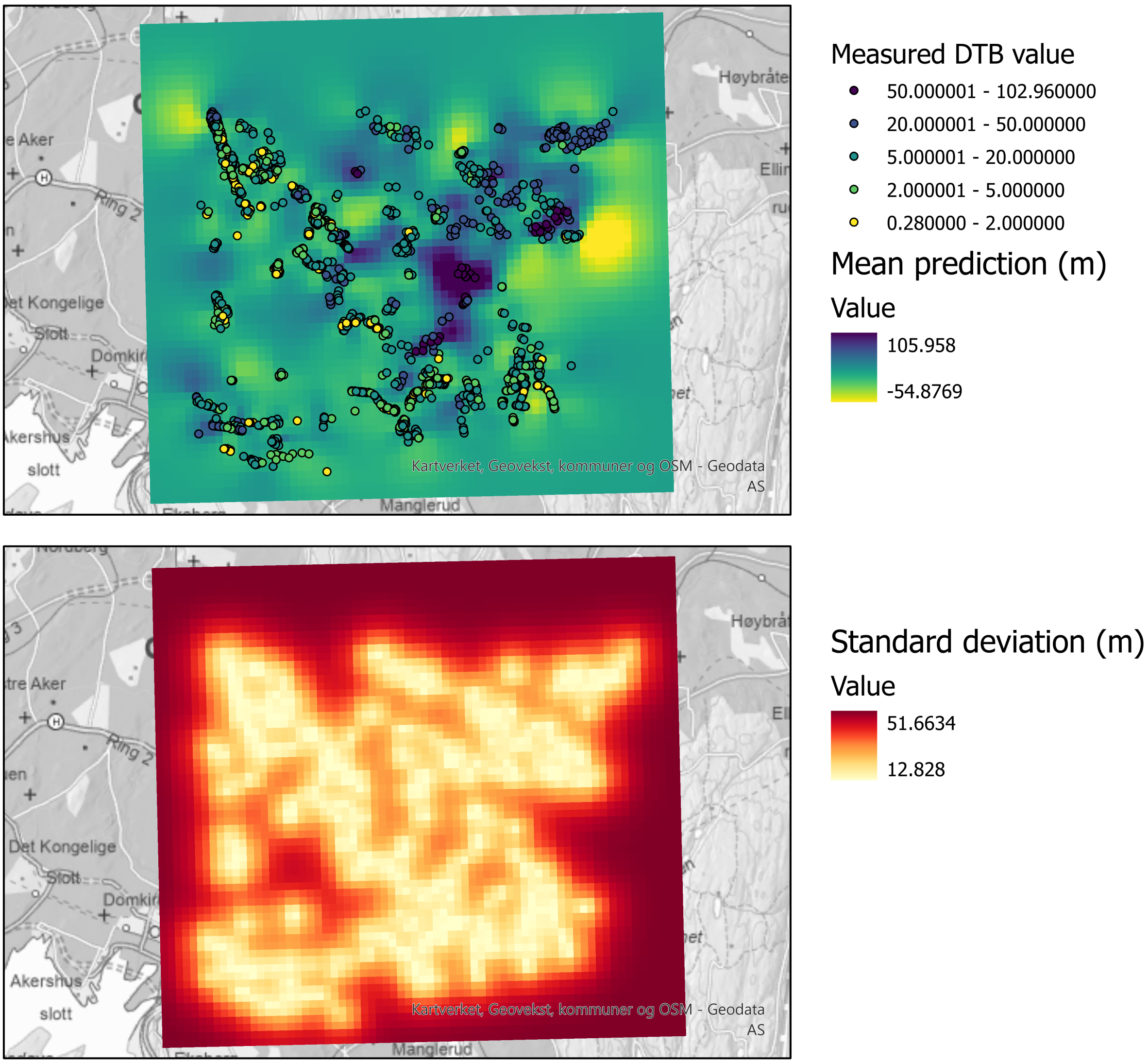


Figure. B.0.1: *Interpolation and uncertainty maps for the isotropic model with Matérn 3/2 kernel trained on the Oslo East dataset.*

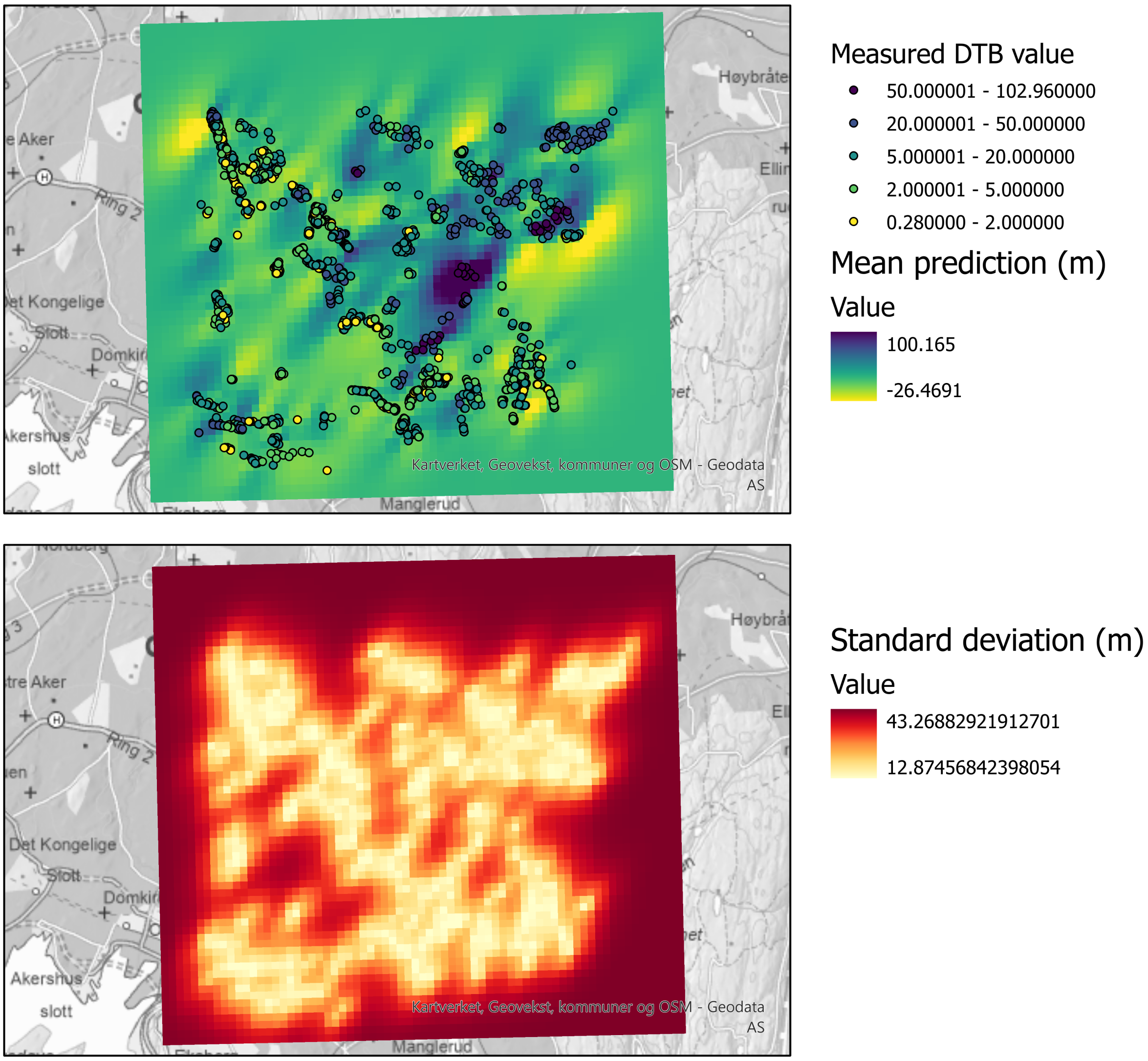


Figure. B.0.2: *Interpolation and uncertainty maps for the anisotropic model with Matérn 3/2 kernel trained on the Oslo East dataset.*

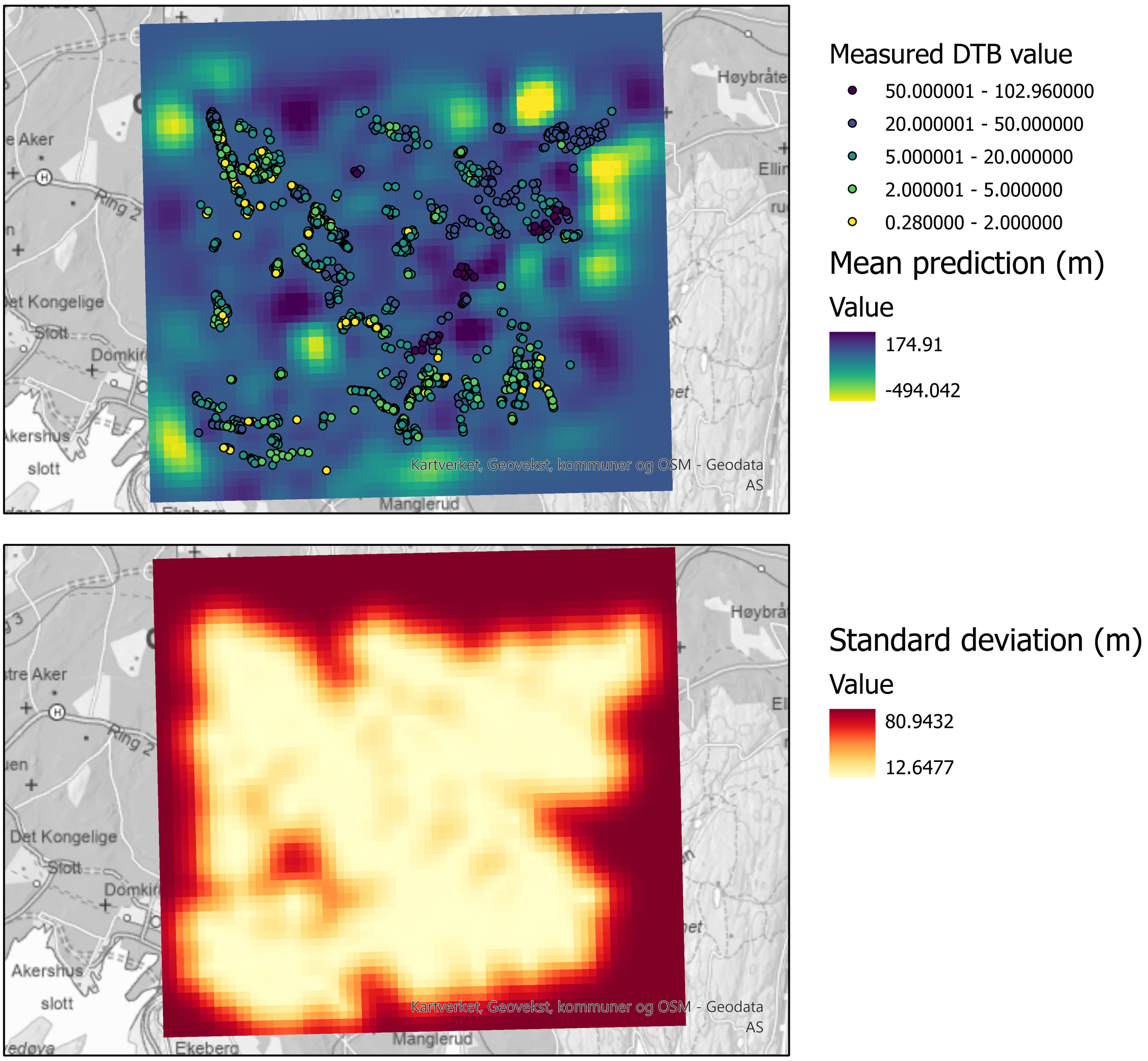


Figure. B.0.3: *Interpolation and uncertainty maps for the isotropic model with RBF kernel trained on the Oslo East dataset.*

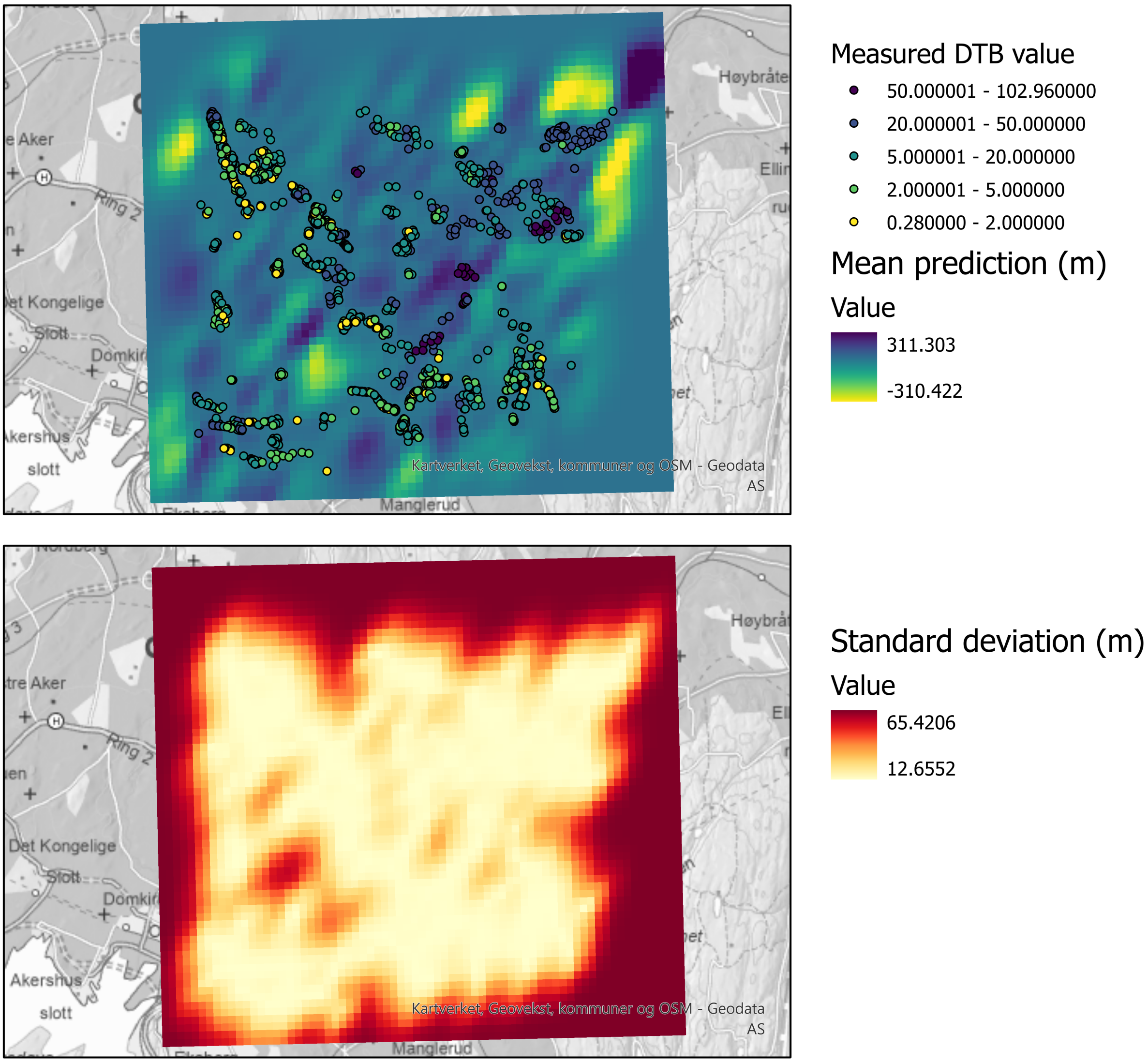


Figure. B.0.4: *Interpolation and uncertainty maps for the anisotropic model with RBF kernel trained on the Oslo East dataset.*

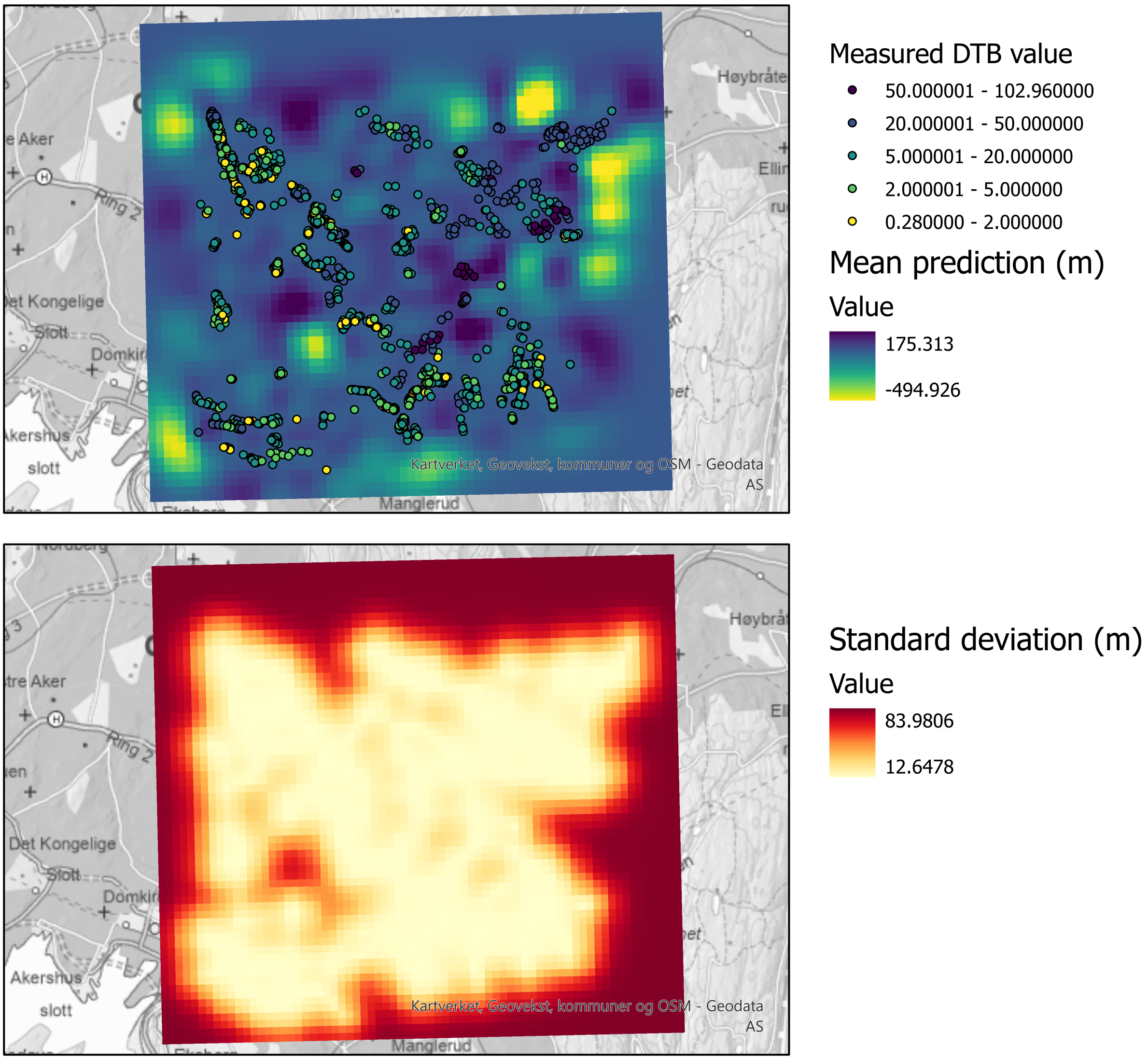


Figure. B.0.5: *Interpolation and uncertainty maps for the isotropic model with linear and RBF kernels trained on the Oslo East dataset.*

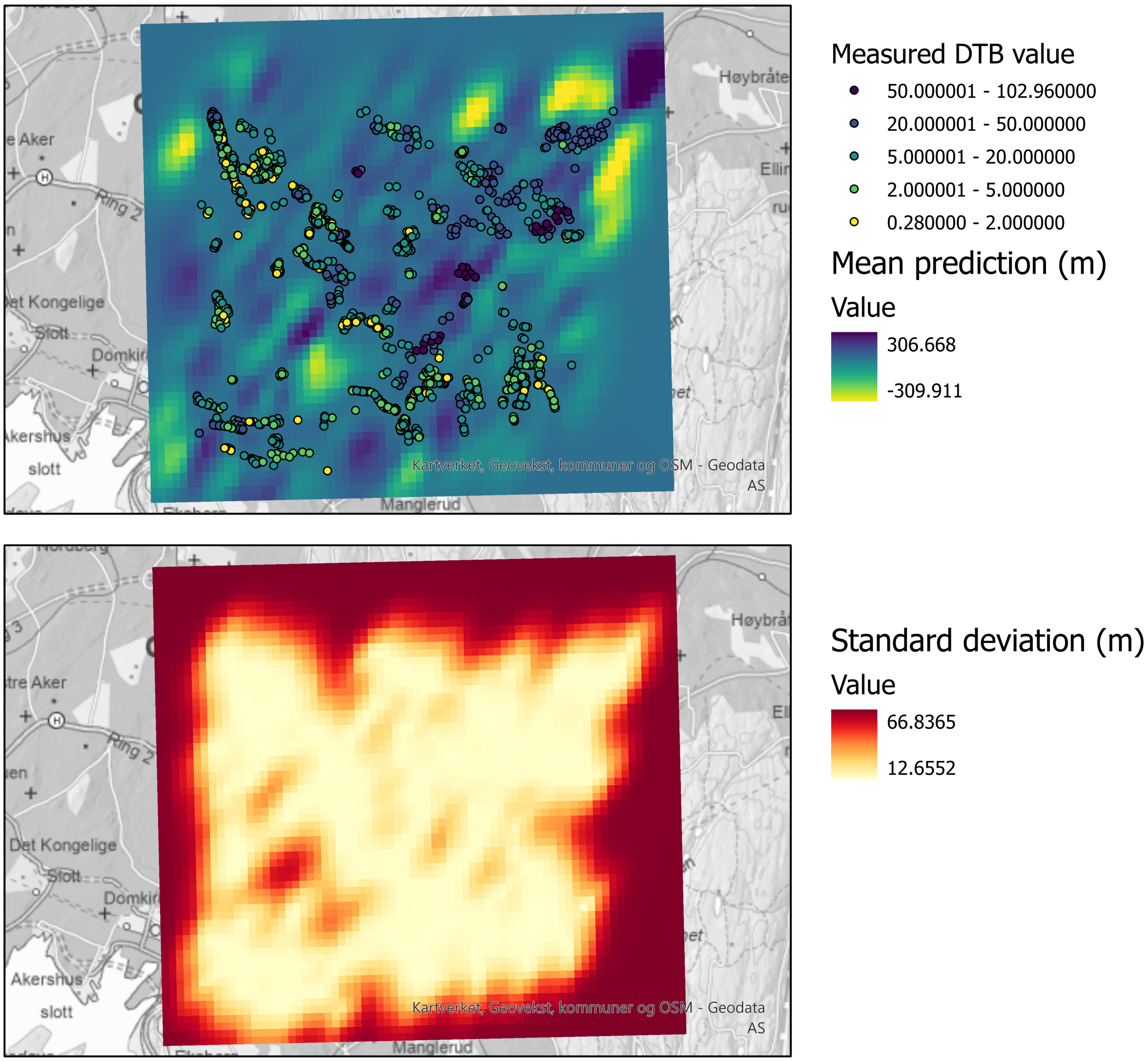


Figure. B.0.6: *Interpolation and uncertainty maps for the anisotropic model with linear and RBF kernels trained on the Oslo East dataset.*

Appendix **C**

Interpolation Results for Bærum E18 Dataset

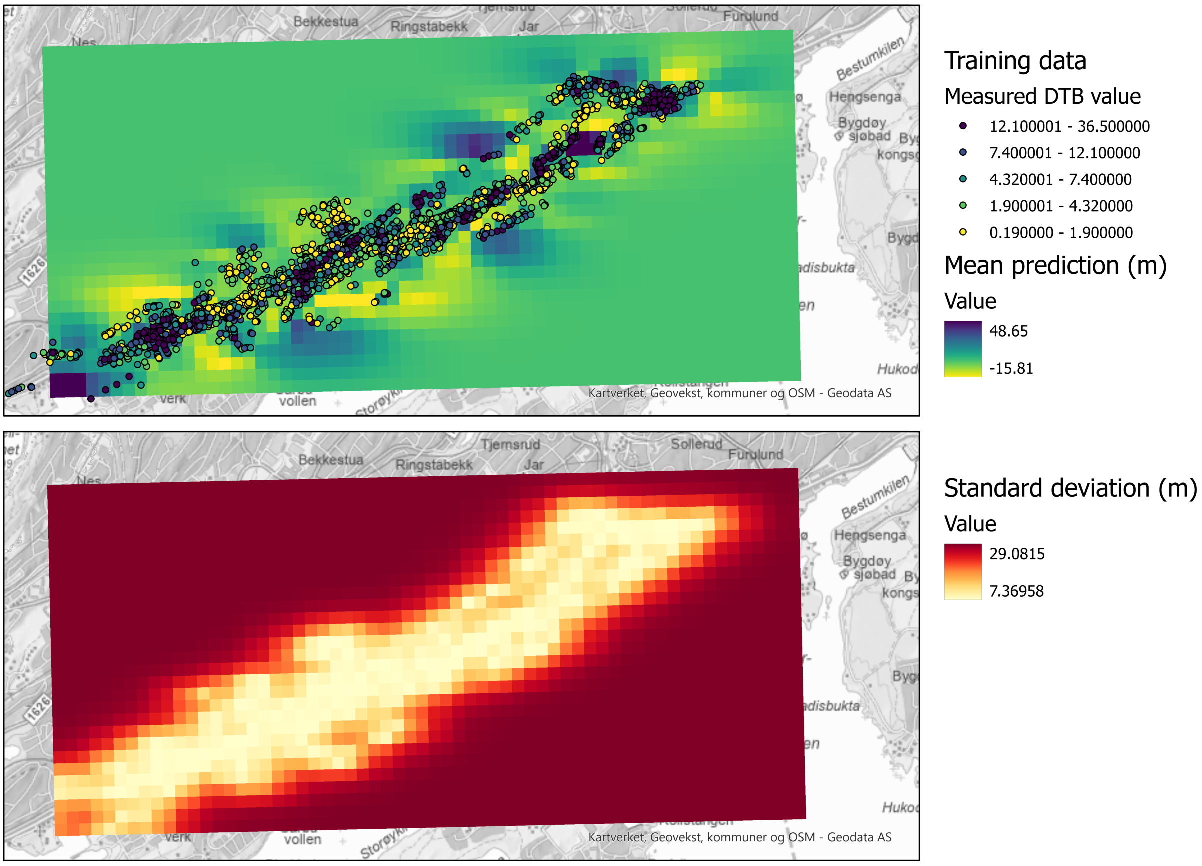


Figure. C.0.1: *Interpolation and uncertainty maps for the isotropic model with Matérn 3/2 kernel trained on the Bærum E18 dataset.*

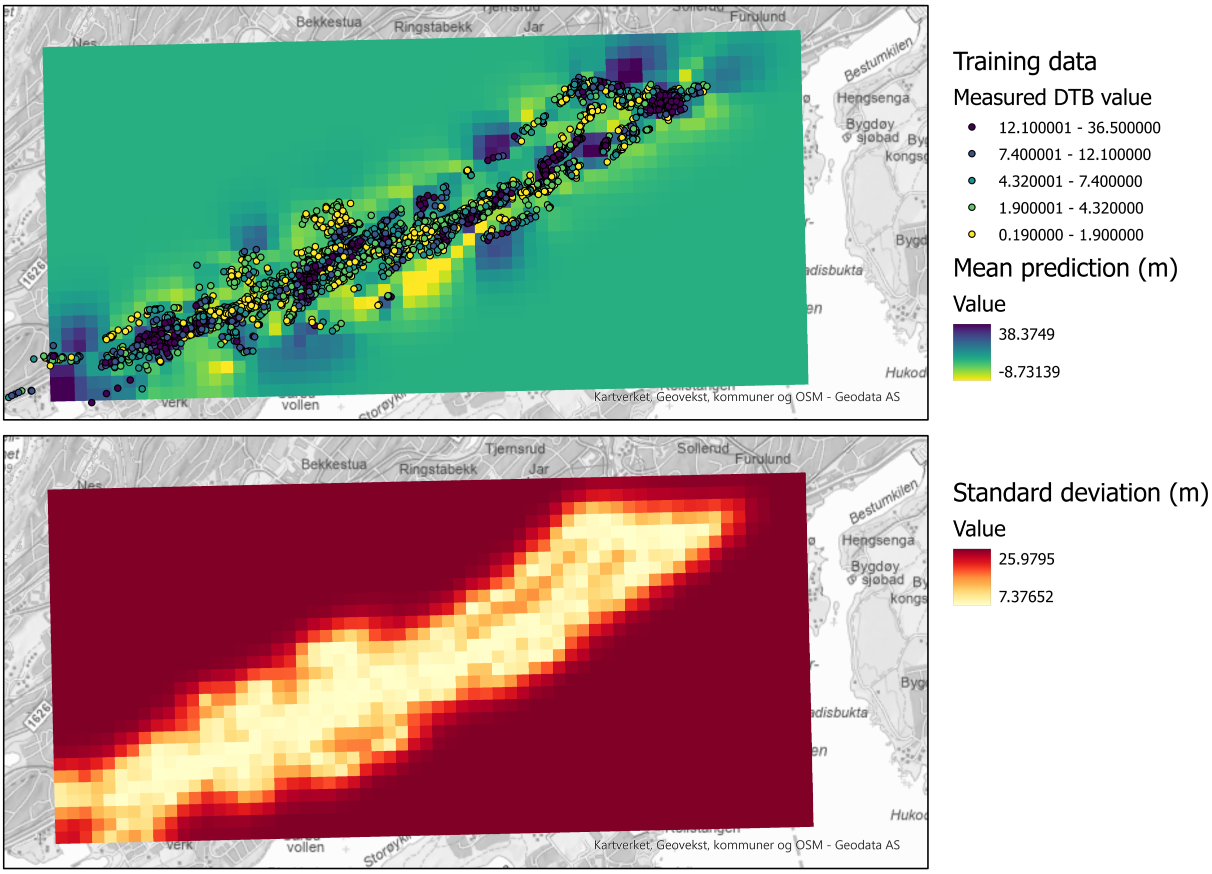


Figure. C.0.2: *Interpolation and uncertainty maps for the anisotropic model with Matérn 3/2 kernel trained on the Bærum E18 dataset.*

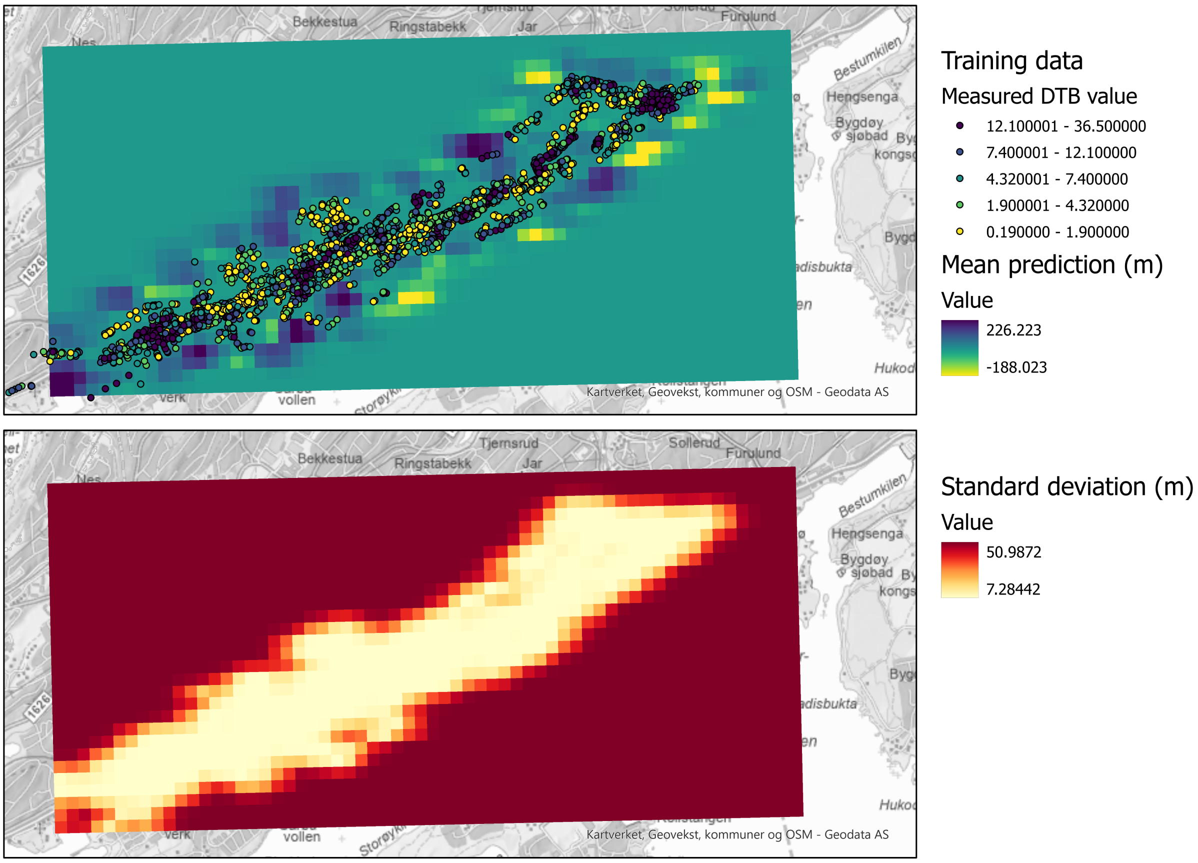


Figure. C.0.3: *Interpolation and uncertainty maps for the isotropic model with RBF kernel trained on the Bærum E18 dataset.*

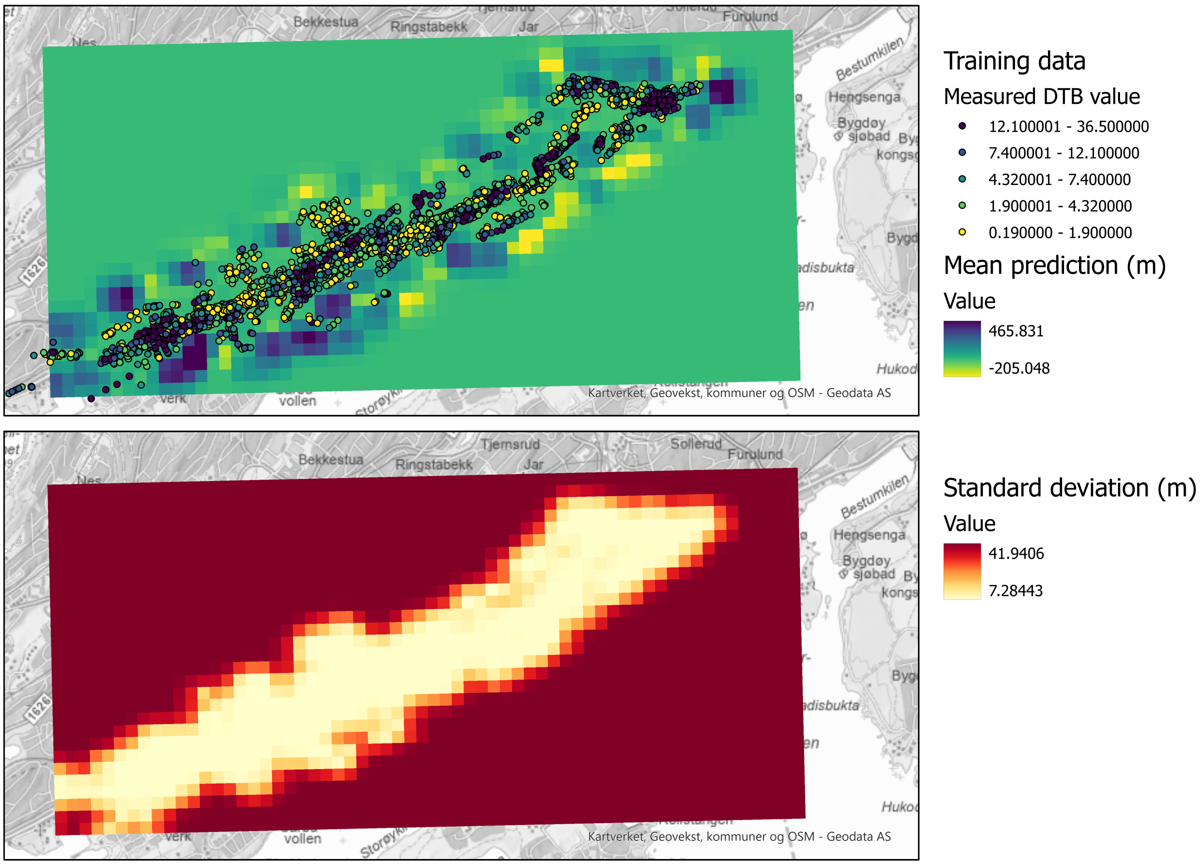


Figure. C.0.4: *Interpolation and uncertainty maps for the anisotropic model with RBF kernel trained on the Bærum E18 dataset.*

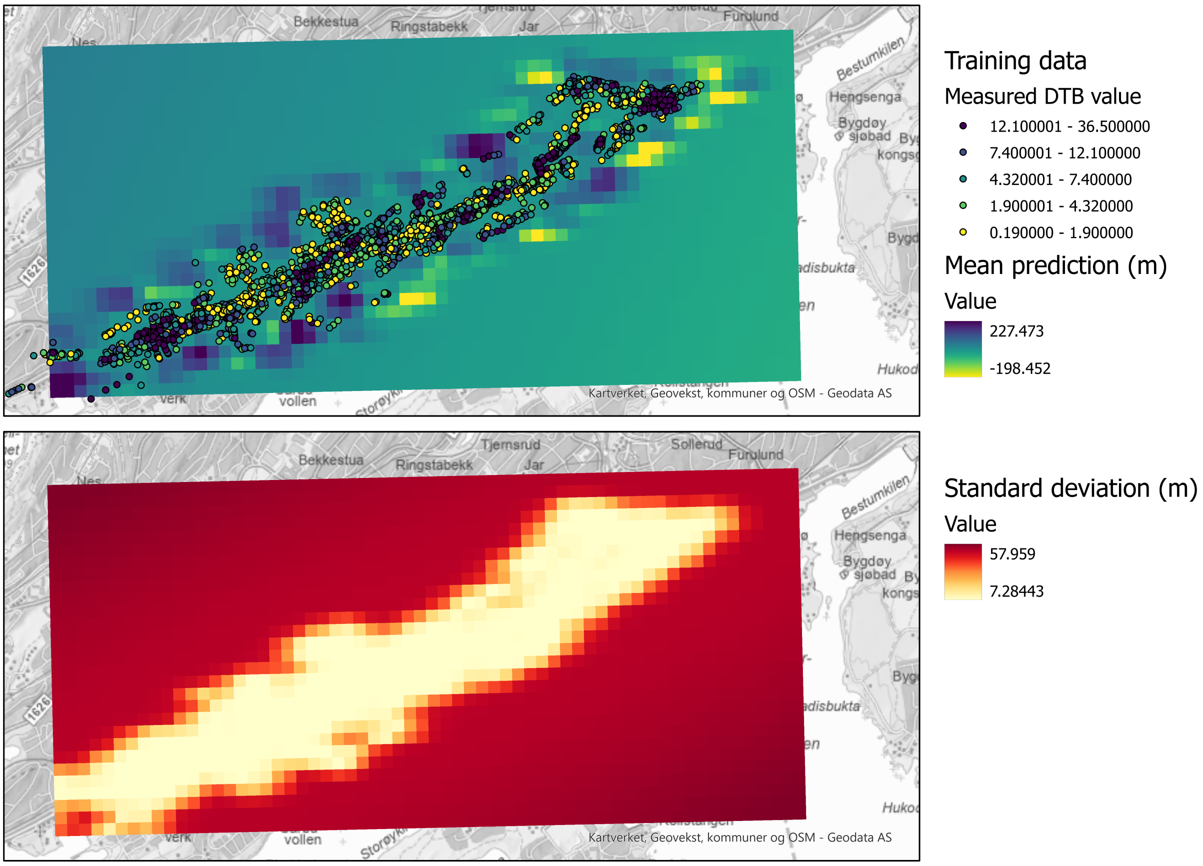


Figure. C.0.5: *Interpolation and uncertainty maps for the isotropic model with linear and RBF kernels trained on the Bærum E18 dataset.*

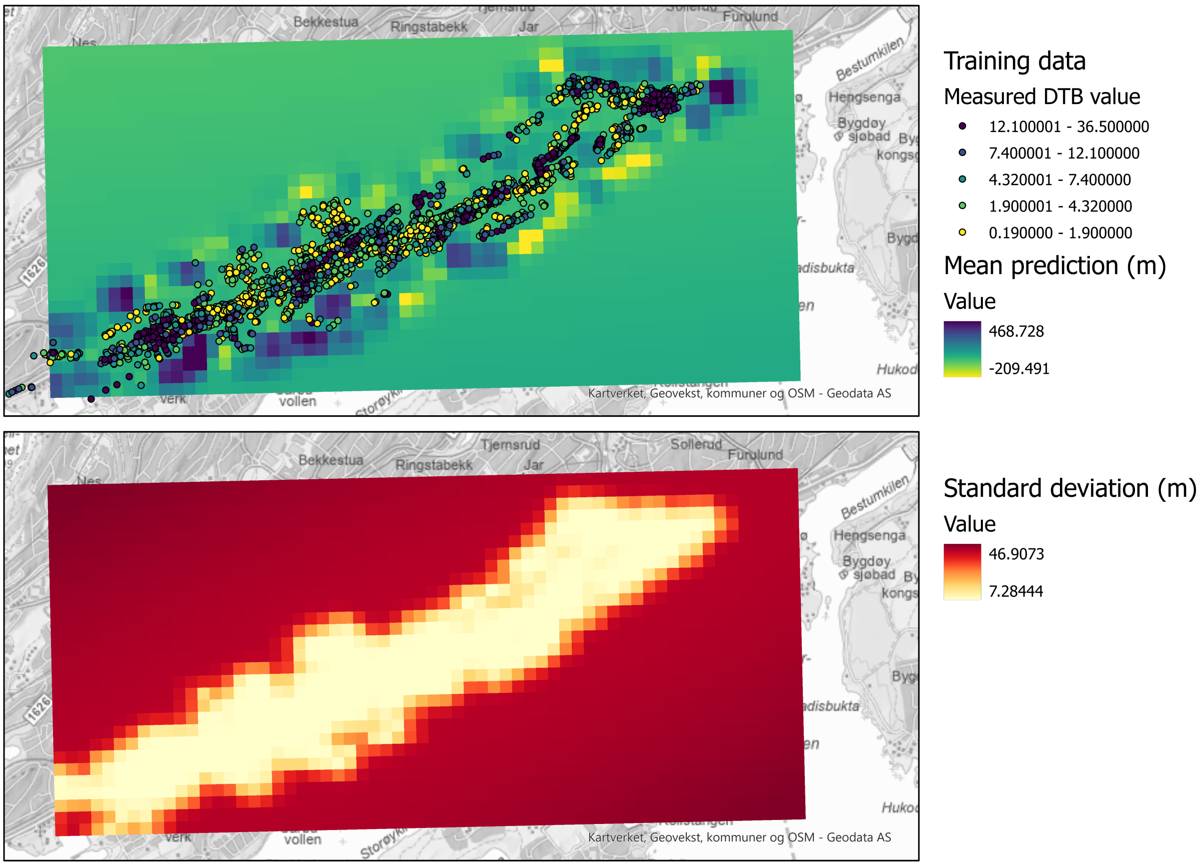


Figure. C.0.6: *Interpolation and uncertainty maps for the anisotropic model with linear and RBF kernels trained on the Bærum E18 dataset.*

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway