MSc thesis in Geomatics

Estimating building height from ICESat-2 data: the case of the Netherlands

Ziyan WU
2022

MSc thesis in Geomatics

Estimating building height from ICESat-2 data: the case of the Netherlands

Ziyan WU

June 2022

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

Supervisors:	Dr. Hugo Ledoux
	Ir. Maarten Pronk
Co-reader:	Dr. Azarakhsh Rafiee

Abstract

Building height information has been more used in a variety of industries in recent years. This information can be used in sustainable urban planning, urban climate research, population estimation, and three-dimensional (3D) building reconstruction, etc. Nowadays, building height information is obtained mostly by photogrammetry, high-resolution photos, and aerial Light Detection and Ranging (LiDAR) data. However, the existing technique is constrained by issues of scale, cost, and quality.

The Ice, Cloud and land Elevation Satellite-2 (ICESat-2) was launched in 2018, using photon-counting LiDAR technology to gather Earth's surface elevation data globally. It represents the highest level in space-borne laser altimeters and has been proved can estimate building height.

In this thesis, a method is proposed to estimate the height of all buildings in the Netherlands. To estimate the building height, elevation data from ICESat-2 and footprints from Basisregistratie Adressen en Gebouwen (BAG) are two used datasets. Spatial interpolation methods and percentile methods are used to get ground and roof elevation for each footprint, respectively. Random Forest Regression (RFR) method is used to deal with the low coverage of ICESat-2.

The result shows that less 3% of buildings could obtain their height from ICESat-2 data. Among these buildings, 90% of them are lower than 10 meters and half of them are between 5 - 10 meters. This caused a low performance of the prediction model with Mean Absolute Error (MAE) of 2.1m. The building between 5 - 10 meters has the smallest MAE of 1.1267m. Small amount of available of ICESat-2 data is the key reason, which leads to the training data of building over 10 meters is not enough.

The results reveal it is impossible to get the height of all buildings in the Netherlands with ICESat-2 data. But the proposed method is a feasible option for buildings between 5 and 10 meters in height.

Acknowledgements

Firstly, I would like to express my deepest appreciation to my two supervisors: Dr. Hugo Ledoux and Ir. Maarten Pronk, for their valuable feedback and guidance. Without their timely assistance and insights, this paper would have never been accomplished on time. Thanks so much for their patience and suggestion on my research.

Then I am also thankful to Dr. Azarakhsh Rafiee as my co-reader, for her providing useful suggestions and comments. Thanks should also go to Dr. John Heintz, for his suggestion on the presentation, making it more structured and understandable.

Lastly, I would be remiss in not mentioning my family, especially my parents. Their unconditional support has kept my spirits and motivation high during this process. I also would like to mention my friends and everyone I met, thanks to their company, which together made up the colorful two years in the Netherlands.

Contents

List of Figures

List of Tables

Acronyms

ICESat-2 Ice, Cloud and land Elevation Satellite-2 v
GEDI Global Ecosystem Dynamics Investigation xiii
LiDAR Light Detection and Ranging v
ALS airborne laser scanning 5
ATLAS Advanced topographic laser sltimeter system 5
NL Netherlands 2
BAG Basisregistratie Adressen en Gebouwen v
AHN Actueel Hoogtebestand Nederland 28
LOD Level of Detail xi
OCG Open Geospatial Consortium xi
CBD Central Business District 11
RF Random Forest 12
RFR Random Forest Regression v
SVR Support Vector Regression 2
GB Gradient Boosting 12
MLR Multiple Linear Regression 2
MAE Mean Absolute Error v
RMSE Root Mean Square Error 6
MAPE Mean Absolute Percentage Error 24
VIF Variance Inflation Factor 22
MIC Mutual information and maximal information coefficient 32
CV Cross validation 22
GridSearchCV Grid Search Cross validation 22
RandomizedSearchCV Randomized Search Cross validation 23
NN Nearest Neighbor 20
NNI Natural Neighbour 20
IDW Inverse Distance Weighted xii
TINL Linear Interpolation in TIN 20
IDW Inverse Distance Weighted xii

1. Introduction

Building height information is critical for understanding the effects of the built environment on the environment. Sustainable urban planning, urban climate research, population estimation, three-dimensional (3D) building reconstruction, etc. are all have a close relationship with the height of the building [Li et al., 2020]. Though elevation datasets (e.g. point clouds) are necessary, it is frequently unavailable in creating 3D city models [Biljecki et al., 2017].

Several kinds of remote sensing data, such as photogrammetry, high-resolution images, airborne LiDAR data, have been used to estimate the building height. Despite remote sensing and satellite image help a lot in related scientific studies, it has several limitations. First, it is costly and time-consuming to get [Wendel et al., 2016]. Second, even when elevation data is accessible, they may not always be useful for generating 3D models because they are obsolete or the resolution and quality are insufficient to construct 3D city models [Biljecki et al., 2017]. Furthermore, most of these studies are limited to local scales, with no data on the global scale [Li et al., 2020; Frantz et al., 2021].

Currently, two new published space-borne LiDAR data sets have been considered by researchers to extract building height information.

The ICESat-2 was launched in 2018, using photon-counting LiDAR technology to gather Earth's surface elevation data globally. Though ICESat-2 was designed to supervise changes in sea ice, the photon counting concept has been used in applications areas like forestry, biomass and building heights estimation [Dandabathula et al., 2021]. They proved that retrieving the building heights from the surface reflected ICESat-2 geolocated photons is feasible.

The Global Ecosystem Dynamics Investigation (GEDI) instrument was also launched in 2018, is designed to measure ecosystem structure and dynamics, also total carbon contained in all forests. The data provided by GEDI have been widely used in weather forecasting, forest management, snow and glacier monitoring, and digital elevation model generation, etc. [Dubayah et al., 2020]. Not like ICESat-2 employing photon-counting technology, it uses full waveform technology. Liu et al. [2021] evaluate evaluates the performance for terrain and canopy height retrievals with ICESat-2 and GEDI data, proved that they represent the top level in space borne laser altimeters in terrain and canopy field. Kokalj and Mast [2021] investigates the applications of GEDI data in archaeological feature recognition. For building height estimation area, there is no relevant research yet.

ICESat-2 and GEDI missions, which represent the highest level in space-borne laser altimeters in technical part [Liu et al., 2021]. Thus, as the latest datasets, it is worthwhile to investigate the applicability of ICESat-2 and GEDI data in estimating building height. This thesis will focus on estimating the height of all buildings in the Netherlands with ICESat-2 photon data and evaluate its accuracy with 3D BAG as the reference. The 3D BAG is open data containing 3D building models and building attributes of the Netherlands. It is combining two open data sets: the building data from the BAG and the height data from the AHN.

Because of the low coverage behaviour problems, which will be explained in Section 2.1, the machine learning method will be considered as a supplement when implementing height estimation at a large level.

There has been some effort in the literature to scale building height estimation with machine learning method. RFR, Multiple Linear Regression (MLR) and Support Vector Regression (SVR) have been used in this area [Biljecki et al., 2017; Roy, 2022; Lánský, 2020].

1.1. Research objectives

The goal of this thesis project is to estimate building heights from ICESat-2 and evaluate the accuracy of the data from 3D BAG. Because of not every building's footprint is intersected with ICESat-2 data, the first thing needs to do is to assess how many buildings in Netherlands (NL) are covered. For those that are uncovered, the Machine Learning method is used to solve this problem.

Based on this objective, the main research question is formulated as follows:

Can the height of all buildings in the Netherlands be estimated from ICESat-2 data and what accuracy can be achieved?

In order to achieve the main research objective, the following sub-questions are defined:

1. What's the percentage of building in NL are covered by ICESat-2 dataset? Is it enough to estimate all buildings with the ML method in NL?
2. How to identify the ground and roof points from ICESat-2 dataset? Which method can be used in getting ground and roof elevation for each footprint considering both efficiency and accuracy?
3. Which ML method should be used to predict the height of buildings which are not intersected with ICESat-2? And what attributes should be considered?
4. What's the accuracy of estimated building height and model performance? Where are those errors from?

1.2. Scope

This thesis will focus on estimating building height information from ICESat-2 data and extend to the whole Netherlands area with machine learning methods. The resulted building height will be evaluated with the height from 3D BAG dataset. Investigating which one ML method and which features are the best or compare their performance is not included in this project. The specific ML method and features to be used will be decided according to the previous research.

1.3. Outline

This thesis consists of five main chapters.

Chapter 2 presents the background of the research related to this thesis. Two types of Space-borne LiDAR are highlighted, including their characteristics and existing research. Different height references are also presented, as well as methods for obtaining building heights using machine learning. The focus is on exploring the possibility of estimating building heights from Space-borne LiDAR and scaling with ML methods.

Chapter 3 describes in detail the specific methods, including data collection, data cleaning, identify ground and roof points and building height calculation. Three parts of the machine learning method are also demonstrated: model generation, hyperparameter tuning, and accuracy of measurement.

Chapter 4 is about the specific implementation process of the ML method, including training data sets, feature selection and hyper parameter tuning.

Chapter 5 is about the results and the error analysis. The results and errors of the estimated building heights are shown, as well as the performance of the ML model, and the errors that exist and the potential causes are analyzed. Cases from the study area are also presented in this part to explain the causes of errors.

Finally, Chapter 6 is the concluding part of this thesis. The research questions raised in Chapter 1 are first answered, and then the contributions and limitations of this thesis are discussed. At the end, recommendations for future work based on the results of this thesis are presented.

2. Related work

2.1. Space-borne LiDAR

LiDAR is a laser-based technology for measuring distances. Space-borne LiDAR is a high-precision earth exploration technology developed in the 1960s, using satellites as a platform with high orbit and wide observation range, which can reach almost every corner of the world [Sampath and Shan, 2010].

Compared with airborne laser scanning (ALS), space-borne lidar can map globally. It can provide global coverage at a fraction of the cost (for example, $94 million for GEDI for all land between 51.6° N and S) [Hancock et al., 2021]. However, current space-borne LiDAR has very sparse sampling (GEDI will directly measure less than 4% of the Earth) [Hancock et al., 2021]. Preventing their use in a range of applications that require continuous coverage.

Figure 2.1 displays the orbits of two kinds of space-borne LiDARs.

Figure 2.1.: Orbit illustration of two space-borne LiDAR *Source:*Pronk et al. [2022]

2.1.1. ICESat-2 mission

The ICESat-2, was launched on September 15, 2018, coverage up to 88°N–88°S latitude (see Figure 2.1). It is distinguished by a photon counting technology supported by the Advanced topographic laser sltimeter system (ATLAS) instrument.

ATLAS sensor produces three pairs of beams, each contains a strong signal beam and a weak signal beam (energy ratio of 4:1). The distance between each beam pairs is 3.3 km in the cross-track direction. And the distance between the strong and weak beam in a pair is 90 m (depicted in Figure 2.2).

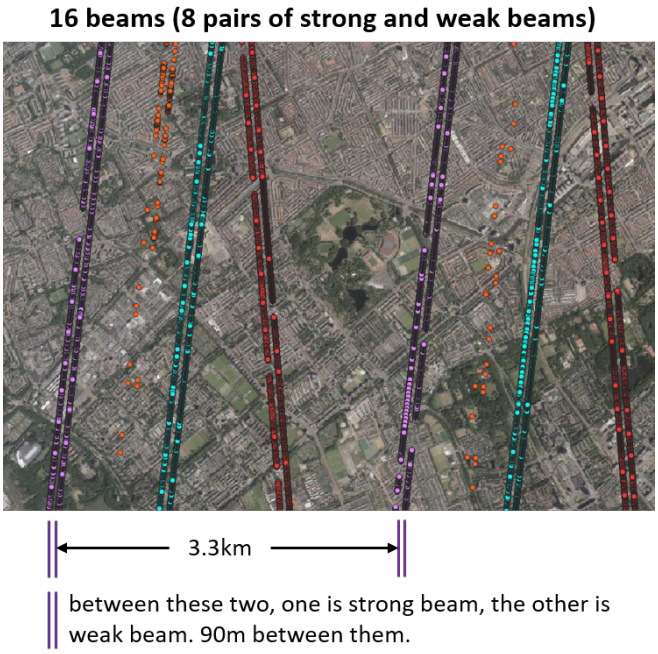


Figure 2.2.: Illustration of 16 beams (8 pairs of strong and weak beams)

Each point from these beams ideally has a footprint with about 17m diameter. And as time grows and energy decrease, this value could increase to about 20m in three years [Neuenschwander and Pitts, 2019]. In theory, the elevation obtained by ICESat-2 point could be any object inside this footprint.

Figure 2.3 shows the four levels of ICESat-2 dataset products, Level-1, Level-2, Level-3A and Level-3B. The Level-3 products focus on a surface-specific field, such as land ice, oceans, sea ice, water and vegetation.

Although ICESat-2's major scientific aim is to supervise changes in ice, its land product is also a vital supplement to current biomass and vegetation mapping activities [Narine et al., 2019].

For its application in the built environment area, especially in estimating building height, Dandabathula et al. [2021] demonstrated that it is feasible to obtain building heights from surface reflected ICESat-2 geolocated photons and then compared the results to field measurements to assess accuracy. The accuracy is up to 17 cm.

Lao et al. [2021] developed and validated a methodology to extract the building height based on the noise removal algorithm adopted by ATL03 product, random sample consensus (RANSAC) linear fitting, and mathematical statistics. They also compare the performance of day/night acquisition and strong/weak beam. Day acquisition/strong beam leads to errors lower than night acquisition/weak beam. The Root Mean Square Error (RMSE) between estimated and

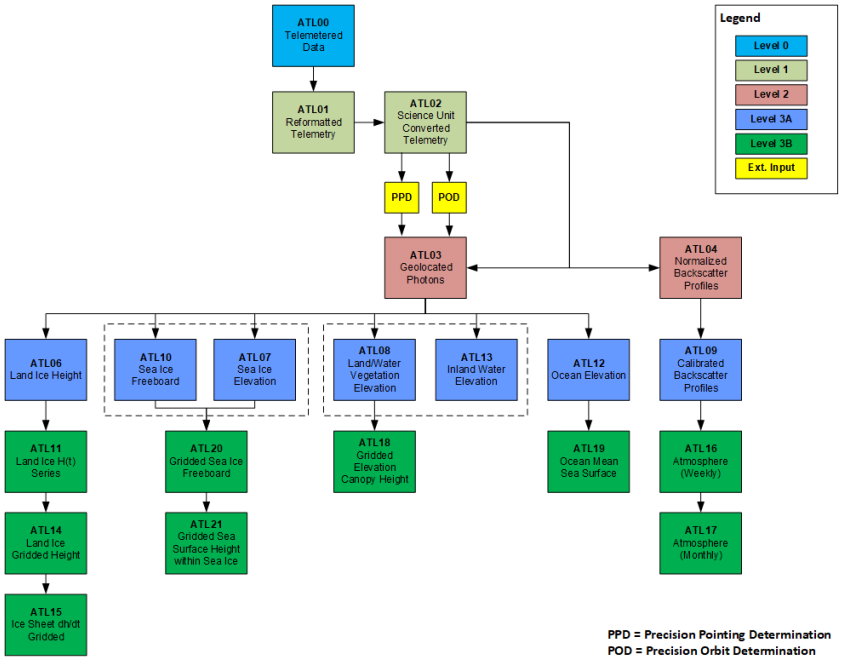


Figure 2.3.: ICESat-2 data processing workflow. *Source:* NASA/NSIDC offical website.

reference building height is between 0.35 m and 0.45 m, when do the comparison with different data acquisition conditions. Their result shows that, in urban areas, there is no huge difference between acquisition times and beam intensity of the ATL03 data when estimating building height. Thus, in this project, ATL03 data will be used as research data.

2.1.2. GEDI mission

The GEDI instrument was launched on 5 December 2018, from Cape Canaveral Air Force Station, Florida. The GEDI is a full-waveform LiDAR device that is installed on the International Space Station (ISS). For its core use – estimating global aboveground biomass – it gives exact measures of forest canopy height, canopy vertical structure, and surface elevation.

The GEDI measurements are made over the Earth's surface between 51.6° N and 51.6° S (see Figure 2.1). It includes three identical lasers. The GEDI instrument releases three laser beams at the same time at first. Then, one of them is split into two beams which is called "coverage" beam, and the other two lasers remain at full power which is called "full power" beam. Thus, there are four beams in total now. Next, each of them is dithered to create a total of eight ground tracks, separated by 60 m along track and by 600 m across track (depicted in Figure 2.4).

The Land Processes Distributed Active Archive Center[1] distributes GEDI land data processed to Level-1 to Level-4 (details show in Table 2.1). Each type has beens used in different applications.

[1]LP DAAC, is one of several discipline-specific data centers within the NASA Earth Observing System.

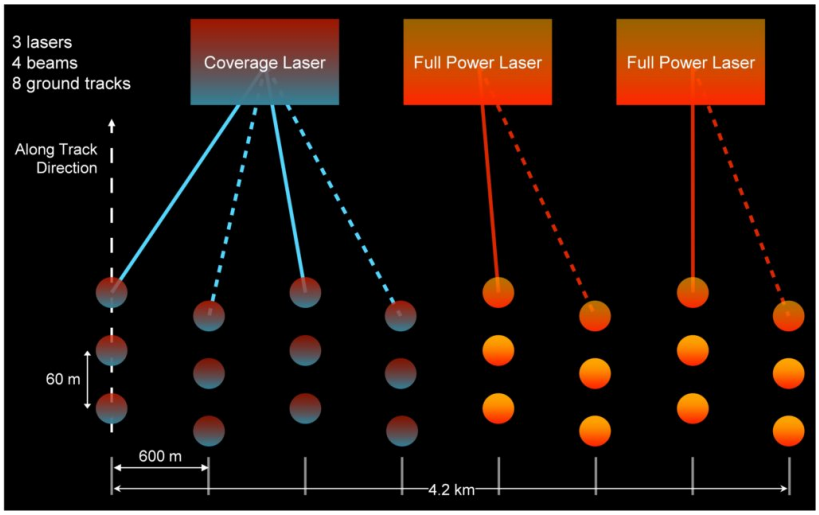


Figure 2.4.: GEDI's ground sampling pattern. *Source:* GEDI offical website

Liu et al. [2021] validate the accuracy of terrain and canopy height retrievals with the GEDI L2A product (version 2) and the ICESat-2 ATL08 product (version 4). Their result shows that these two products are reasonable to be used in terrain height estimation. And the performance of ICESat-2 is better than GEDI. Kokalj and Mast [2021] reanalyzing GEDI data(focused on the L2B dataset) for detection of ancient Maya buildings, demonstrates that the dataset is currently inappropriate for the intended application of archaeological feature recognition due to the lack of coverage and low density of GEDI points.

Table 2.1.: GEDI data product level

Level	Data Products	Resolution
Level-1A	Raw Waveforms (not publicly available)	25m
Level-1B	Geolocated Waveforms	25m
Level-2A	Ground Elevation, Canopy Top Height, Relative Height Metrics	25m
Level-2B	Canopy Cover Fraction (CCF), CCF profile, Leaf Area Index (LAI), LAI profile	25m
Level-3	Gridded Level 2 metrics	1km
Level-4A	Footprint level above ground biomass	25m
Level-4B	Gridded Above Ground Biomass Density (AGBD)	1km

The biggest limitation of GEDI data is the low coverage, less than 4% of the Earth Hancock et al. [2021]. And the Netherlands is right on the border of the area it covers, which will further affect the coverage rate. Therefore, the GEDI data are not used in this thesis.

2.2. Height reference

When talking about building height of 3d models, it is necessary to mention the concept of Level of Detail (LOD). This concept is often used to describe the complexity of 3D building models. The most widely accepted LOD concept definition contains five levels, which was

defined by Open Geospatial Consortium (OCG). These five levels are shown below (also see Figure 2.5:

1. LOD0: a representation of footprints
2. LOD1: a coarse prismatic model (extrusion of LOD0)
3. LOD2: has a simplified roof shape, the parts of the model can be recognized(e.g. roof, wall).
4. LOD3: an architecturally detailed model with windows and doors
5. LOD4: Indoor details are included based on LOD3

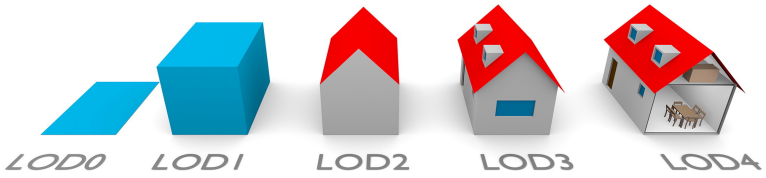


Figure 2.5.: The five LODs defined by the OCG CityGML 2.0 standard. *Source:*Biljecki et al. [2016]

To avoid some ambiguities OCG definition creates, this classification has been further refined to a series which contains 16 LODs by Biljecki et al.. The refined one has four LODs for each original LOD0 to LOD3 (See Figure 2.6).

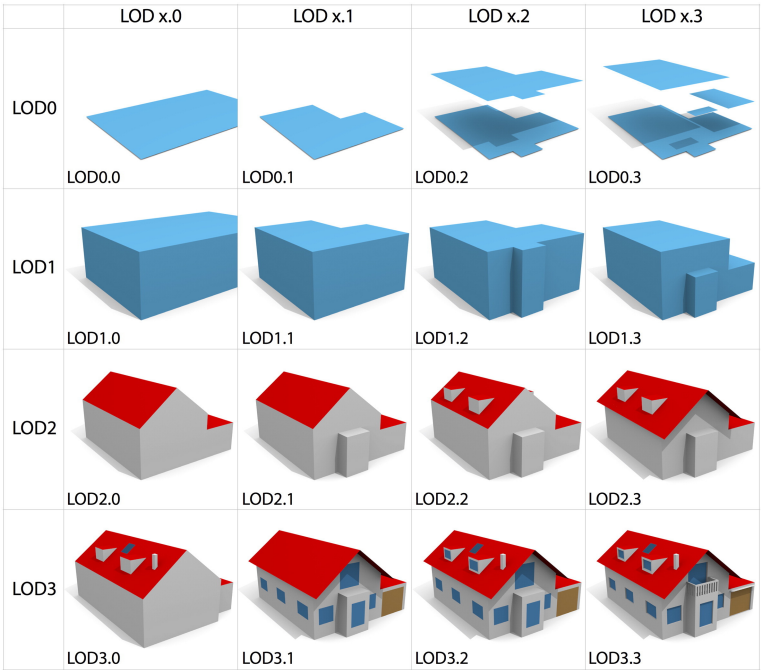


Figure 2.6.: Refined LOD models. *Source:*Biljecki et al. [2016]

Among them, LOD1.2, LOD1.3 and LOD2.2 have been used in 3D BAG in Netherlands. They can show the details of the 3d models in different levels and been used in different applications (see Figure 4.1). LOD1 models is the simplest volumetric 3D city model with a uniform height [Biljecki et al., 2016]. Therefore, LOD1 model can be used to generate 3d models more quickly and easily. But at the same time, the uniform height of LOD1 models also brings some unclear aspects for the building height. Different height references result in different top height of LOD1 models (Figure 2.7). Theses models are all valid, but the geometric representations is much different. Thus, the height reference used to define the top surface of LOD1 model can have a remarkable impact on final building height.

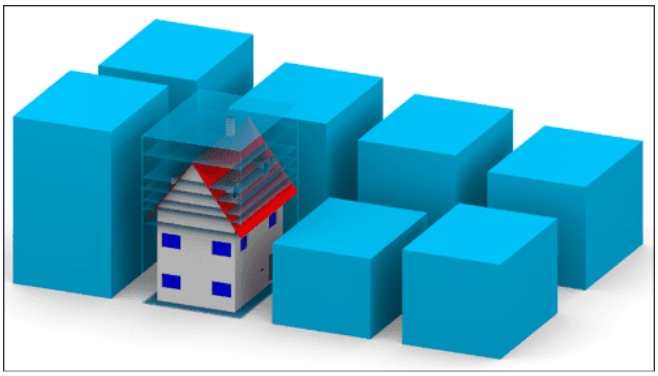


Figure 2.7.: Different geometric representations of the same building. *Source:*Biljecki et al. [2014]

Dukai et al. [2019] developed a method uses different percentiles of the points fall in each building polygon to generate six different reference heights for each building(Figure 2.8). This method can be used to define the height of 3d building model when estimating building height from point data.

2.3. ML for inference of building's height

Machine learning is a method that often is used to get building heights over large areas. Many machine learning methods and distinct features have been used in estimating building heights.

Biljecki et al. [2017] used supervised learning models (Random Forest) with unique features of buildings (predictors) to estimate their heights. The attributes (features) are derived from cadastral and statistical information, as well as the geometry of the building footprints. The features be used are divided into three categories: cadastral data, geometric properties and Statistical data. Cadastral data includes building use, year of construction, number of storeys above ground, and the net internal area (sum of floor area in all units in a building). Geometric properties contain three features, they are footprint area, shape complexity, and number of neighbouring objects. Statistical data (neighborhood characteristics) are obtained from census data from Statistics Netherlands (CBS), contains population density, average household size, and average income.

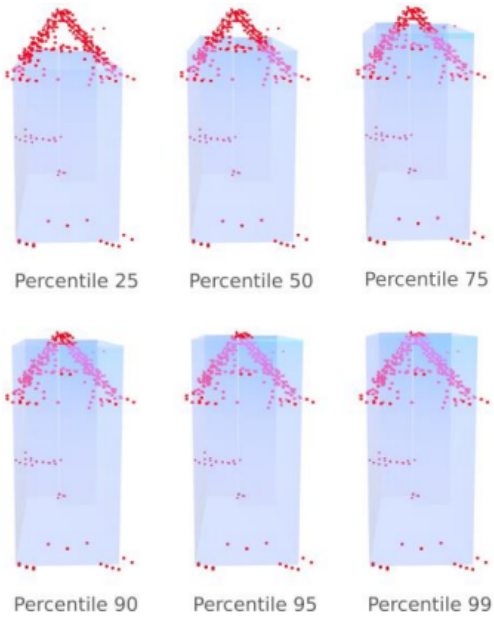


Figure 2.8.: Visualization of six difference reference height. *Source:*Dukai et al. [2019]

After trained predictive models based on different combinations (17 combinations in total) of predictors, Biljecki et al. [2017] found the number of storeys, building age, and net internal area are the most useful predictors. Through the evaluation of 200,000 buildings in the city of Rotterdam, the Netherlands, a MAE of 0.8 metres is achieved. The results demonstrate that, when cadastral data is available, geometric qualities and neighborhood characteristics are of little utility. And when the number of storeys was used in a predictive model, all other features were much less important. However, the storey data is not easily available, especially at the country-wide level.

Anh et al. [2018] applied a similar method and use the same building attributes as Biljecki et al. [2017] in the city of Hanoi, Vietnam. Cross-validation and grid-search approaches were employed to fine-tune and improve the model's accuracy. However, the performance of the predictor model is less accurate with a MAE of 7.12 metres.

Lánský [2020] implemented three machine learning methods, RFR, MLR and SVR to predict the height for all buildings in the USA within an acceptable time. According to the unique features of the CBDs (Central Business Districts) and Suburbs / Rural regions, two models were generated respectively and also a combined one. The features used are classified into two major categories: geometric features and non-geometric features. Geometric features are obtained from footprints, includes area, compactness, complexity, number of neighbours, number of adjacent buildings, length, width, slimness and number of vertices (see Table 2.2). While non-geometric features are from cadastral or statistical (census) data, includes year of construction, the building usage, the number of storeys above ground.

Lánský [2020] tested the performance of the prediction model with and without non-geometric features. The result shows that in Central Business District (CBD) model, all three ML methods are benefited from non-geometric features. While different Machine Learning methods

benefit differently from the addition of non-geometric features. MLR benefits the most with a reduction of 4 .46m in MAE. RFR is the next one with an improvement of 3.55m. SVR only improved 0.3m. For combined model and suburbs model, they have similar trends: non-geometric features improve the performance of the RFR method, while no significant improvements in MLR and SVR methods.

Milojevic-Dupont et al. [2020] focused on urban form when predicting building heights. Table 2.2 shows the attributes (features) they summarised and used in their study. Compared with what Biljecki et al. did in 2017, the street and street-block are given greater consideration, whereas statistical factors such as population density, average income, and typical home size are completely ignored. Also, the cadastral data, such as year of construction, are ignored. Both of them pay attention to the geometric data. Milojevic-Dupont et al. [2020] also proves that cross-country generalization is possible, and low buildings were predicted more accurately than high ones.

Roy [2022] implemented three ML methods to infer the number of floors for building footprints in the Netherlands. This is also related to the building height. These three ML methods are Random Forest (RF), Gradient Boosting (GB) and SVR. She used four feature sets: cadastral features, geometric features (2D and 3D) and census features. Cadastral features including construction year, building function, net internal area and number of units. Geometric features including 2D features (area, perimeter, number of vertices, number of neighbours and number of adjacent buildings) and 3D features (building volume, roof surface area, wall surface area, etc.). Census features including population per square kilometer, percent multi-household, etc.

Three methods have been used to do the feature selection, they are the filter-based method, the embedded method, and a method focused on multicollinearity reduction [Roy, 2022]. Her results show the 3D geometric features are useful for reducing prediction error (90.1% for buildings with less than 5 floors). However, even if with only cadastral features, a good level of accuracy (82.5% for buildings with less than 5 floors) can also be achieved. The worst performance (61.7% for buildings with less than 5 floors) is based on cadastral features and 2D geometric features. Her conclusion is that 3D geometric features and cadastral features are a good combination used in NL. But when considering the data availability globally, a combination of 3D geometric features and 2D geometric features is more applicable.

From the previous research, I decided to use RFR in this thesis, and in the consideration of data availability, non-geometric features will not be included in model training.

Table 2.2.: Summary of features used in building height estimation

Features	Biljecki et al. [2017]	Lánský [2020]	Roy [2022]
Geometric	footprint area	footprint area	footprint area
	complexity	complexity	-
	number of neighbours	number of neighbours	number of neighbours
	-	compactness	-
	-	number of adjacent buildings	number of adjacent buildings
	-	length	-
	-	width	-
	-	slimness	-
	-	number of vertices	number of vertices
	-	-	perimeter
Non-geometric	building use	building use	building type
	year of construction	year of construction	-
	number of storeys above the ground	number of storeys above the ground	-
	net internal area	-	-
	population density	-	population per km^2
	average household size	-	-
	average income	-	-
	-	-	percent multi-household
	-	-	average number of cafes in 1km

3. Methodology

In this section, the main methodology will be described. Section 3.1 summarizes the overall methodology. Section 3.2 describes the process of data pre-processing (cleaning). Section 3.3 and Section 3.4 focus on obtaining data from ICESat-2 and making an estimate of the building height. The last part, Section 3.5 is about building RFR model.

3.1. Overview

As shown in Figure 3.1, the method consists of three main parts: (1) Data preparation and noise removal, (2) Building height estimation, (3) Machine Learning method. More details about these three parts are shown in the following sections.

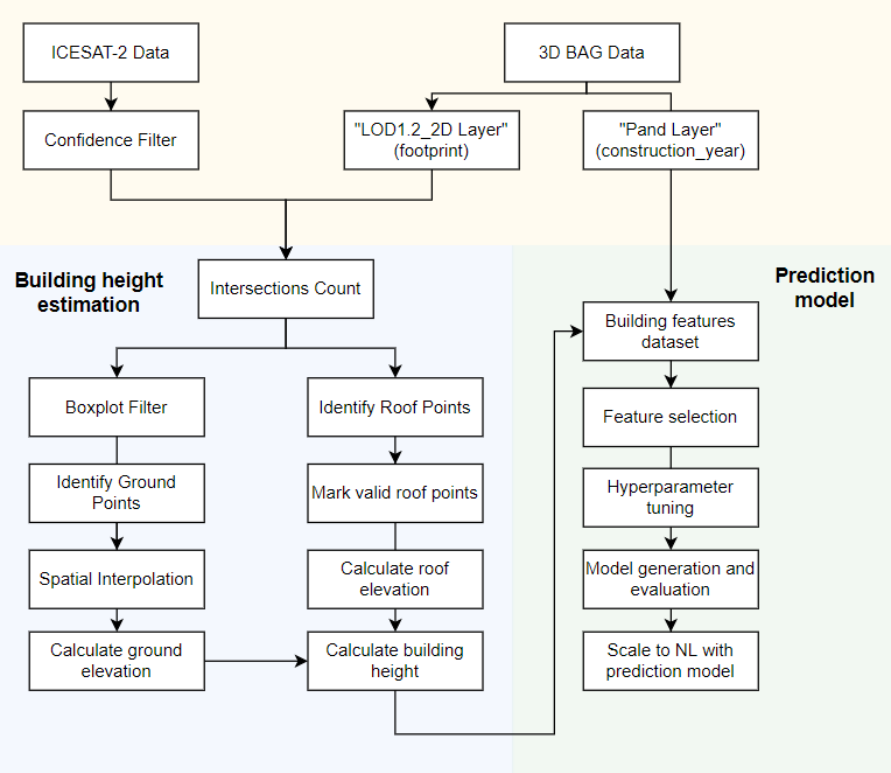


Figure 3.1.: Flowchart of methodology

3.2. Data cleaning

Data obtained directly from ICESat-2 needs to be cleaned, and some preliminary processing is needed to facilitate subsequent use. There are two major ways to clean the data: one is to use the `confidence` property of ICESat-2 data set itself, and the other is to use a boxplot to determine outliers and then remove them. The box plot filter method is only used for ground elevation calculation.

3.2.1. Confidence filter

For data cleaning, the ATL03 product has its own noise recognition algorithm, every photon has an attribute `confidence` before release (Table 3.1). By creating histograms of the number of photons versus the height and computing the signal-to-noise ratio of each histogram bin, the ATL03 product's noise reduction algorithm identifies each photon as either a likely signal photon or a background photon [Neumann et al., 2019].

As a result, it produces a confidence elevation ranging from -1 to 4 to show whether the photon is classed as noise, background, low, medium, or high confidence. Therefore, based on confidence level, the first step of data cleaning will be carried out.

Table 3.1.: Meaning of confidence number

Confidence	Description
-1, 0	noise
1	background
2	low
3	medium
4	high

3.2.2. Box plot filter

In this step, the data will be further cleaned. I assume that the distribution of ICESat-2 points follows the normal distribution, so further filtering is performed using box plot. There are five important numbers in box plot: `"minimum"`, `first quartile (Q1)`, `median`, `third quartile (Q3)`, and `"maximum"`. For the normal distribution, 0.7% of the data is an outlier. In box plot, the data outside the upper and lower edges is an outlier (green points in Figure 3.2).

3.3. Identify the ground and roof

To get the height of a building which has it's own footprint (from BAG), it is necessary to clearly understand the intersection situation of the footprint and ICESat-2 points. For a footprint with several ICESat-2 points, it is also important to think about how to define its ground points and the roof points. The difference between these two categories of points will be used to calculate the building height. In this section, ground and roof elevation values need to be obtained for each footprint.

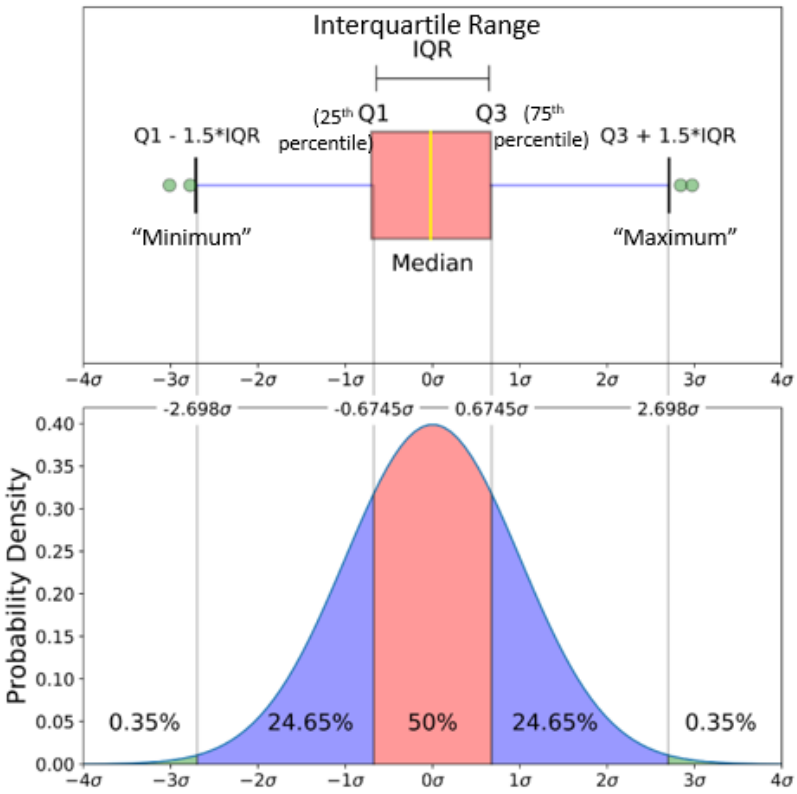


Figure 3.2.: Comparison of boxplot and normal distribution (Source: Galarnyk [2020])

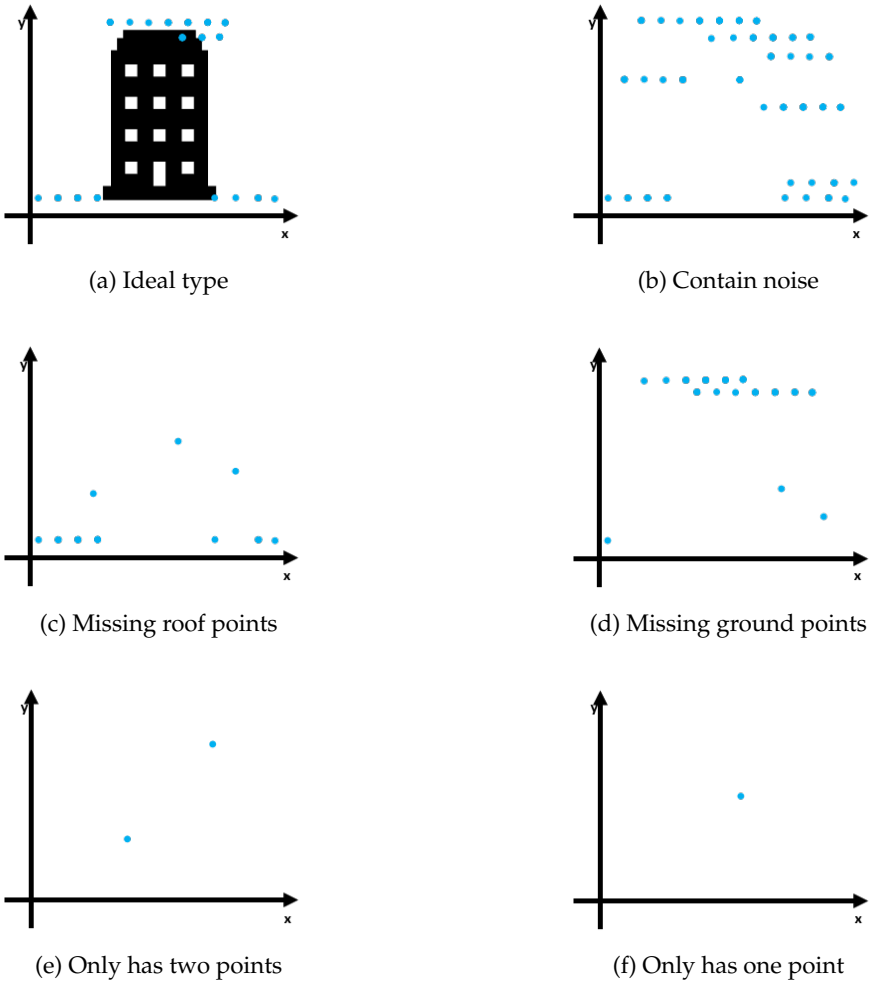
3.3.1. Intersection analysis of ICESat-2 points and building footprint

In order to calculate the height of each building, needs to know the coverage rate of all building footprints. Geopandas library is used to handle the intersections. For each building footprint, the number of ICESat-2 points that are inside it and their z-values need to be known. This is to calculate the elevation of the ground and the elevation of the roof of each building in the next step.

After pre-experiments, the following cases were found to exist:

- The first, most ideal type (Figure 3.3 a), perfectly contains ground data points and roof data points, which are distributed to form two distinct categories. In this case, the building height is the difference between the two categories.
- The second type, containing noise (Figure 3.3 b). It may be trees or other objects in the range. In this case, the access to ground elevation or roof elevation will be affected to some extent, and there are more challenges, such as how to distinguish between roof height and tree top height. It's hard to say which points are belong to roof.
- The third kind, missing data. Because of some unknown reason, the points dropped within the footprint cannot represent the height of the ground (Figure 3.3 c) or the roof (Figure 3.3 d), or in the worst case, neither. They just missing the elevation information.

Figure 3.3 e and Figure 3.3 f illustrates that sometimes there are not enough points in the footprint to determine the ground and roof elevations of this building. There is no way to get the building height from just one or two points.



(a) Ideal type

(b) Contain noise

(c) Missing roof points

(d) Missing ground points

(e) Only has two points

(f) Only has one point

Figure 3.3.: Distribution of points in the footprint

3.3.2. Identify ground

From the cases mentioned above, for some footprints (Figure 3.3a, Figure 3.3b and Figure 3.3c), it is possible to obtain ground points from the ICESat-2 points intersected with them. However, for other footprints (Figure 3.3d, Figure 3.3e and Figure 3.3f), it is impossible to obtain ground points from the ICESat-2 points intersected with them. Therefore, it is limited to consider only the points that fall within the footprint to identify the ground.

In reality, there is another situation that can be used to explain what happens in Figure 3.3d. Figure 3.4 shows a situation where there is no way to obtain the ground points of a house in a townhouse complex, such as building b, building c, building d.

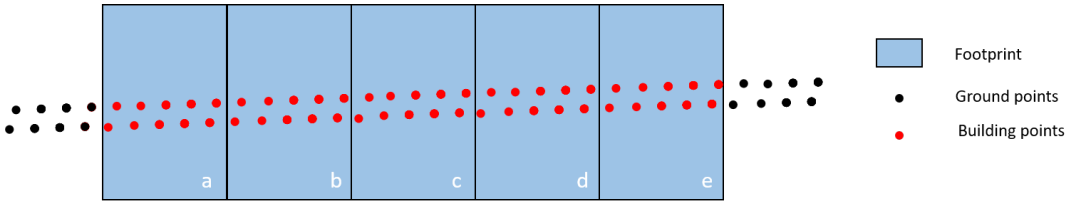


Figure 3.4.: Townhouse

Therefore, when determining the ground, a large range of ICESat-2 data should be considered, not just the ones falling within the footprint. The method of considering only the points within the footprint for determining the ground elevation has great limitations, especially in places with many townhouses like in the Netherlands. There is a need to define another ground determination method that also applies to these footprints without ground points. Since the distribution of ground points is spatially continuous, the first consideration is to use spatial interpolation.

To implement the spatial interpolation method, all ground points (black points in Figure 3.5) inside the dataset are used as known values. Then create a bounding box of the target area with QGIS and rasterize the ICESat-2 points in this area. Through the use of the rules established from these known points, the ground point value of each grid cell can be estimated for the entire extent of the boundary (red point in Figure 3.5). The ground elevation of each footprint is then determined by finding the nearest ground point to the center point (centroid, yellow point in Figure 3.5) of the footprint.

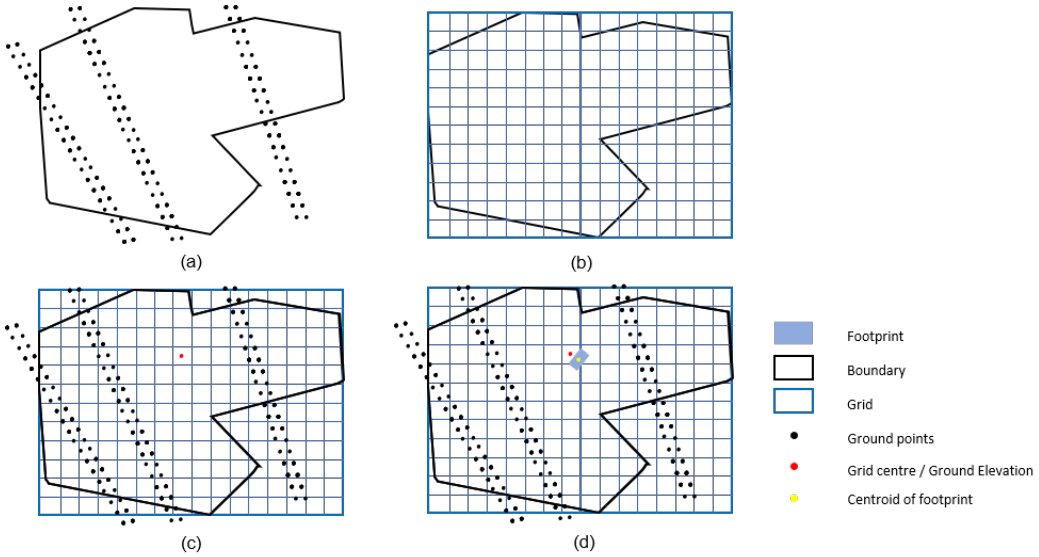


Figure 3.5.: Illustration of spatial interpolation

Three key points need to be considered here:

- How to define the ground points throughout the target area?
- Which spatial interpolation method to use?

- How to determine the size of the interpolation grid?

With the goal of obtaining as much valid ground data as possible, two different methods will be used to define the ground points, which is illustrated in Figure 3.6. The one on the left is the one that was finally chosen (see Section 5.2.1).

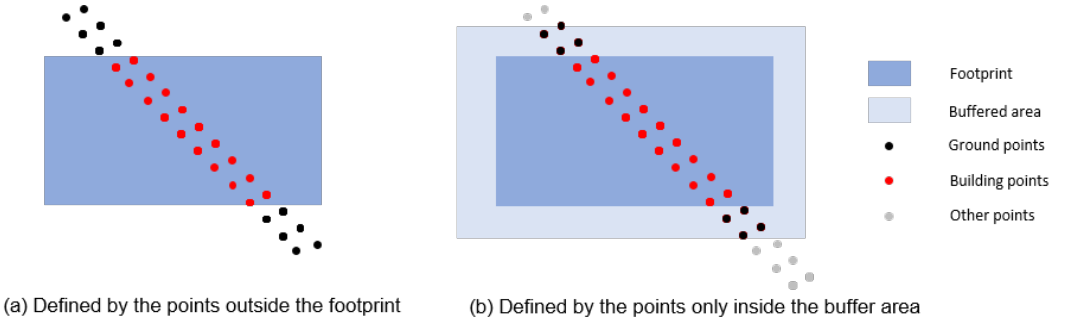


Figure 3.6.: Two definitions of ground points

- The left one in Figure 3.6 is to mark all the points that fall in the footprint as building points and the rest of the points as ground points.
- The right one in Figure 3.6 employs the buffer zone approach, which labels as ground points only the spots within 3 meters of each footprint. This approach can minimize the number of ground points, however for attached buildings, there will be points that fall within both the buffer and the footprint that will be classified as ground points. This is a disadvantage of this strategy.

When coming to interpolation implementation, five specific methods are used in this project, which are Nearest Neighbor (NN), Natural Neighbour (NNI), Laplace, Linear Interpolation in TIN (TINL), IDW. Four grid sizes (25m, 50m, 100m, 150m), four interpolation methods and two ground points definition are used creating 40 different combinations in total.

Using the different combination, ground elevation are obtained for each footprint. This value is compared with the reference value (`h_maaiveld` (height of ground level in Dutch) from BAG). The absolute error between the two will be used to measure the accuracy of this combination. And the one with the smallest error will be finally chosen.

3.3.3. Identify roof

Compared to the ground, the roof points are a little more straightforward to classify. Roof points always comprise points that fall in the footprint. But from Figure 3.3, it is known that not all the points falling in the footprint represent the roof. Among all these points, the ones with a larger z value are more likely to be roof points. Considering this distribution pattern, a percentile approach will be used to determine the elevation of the roof.

In order to make the calculated roof elevation more accurate, it is necessary to find out and exclude those footprints that don't have points higher than the calculated ground elevation. Because it is impossible to get roof elevation for them. Moreover, the minimum floor height should also under consideration, a building with 1m high is unmeaningful in real world.

Thus, the value used to filter out footprints which only have no meaningful roof points is the calculated ground elevation plus the minimum floor height.

From the bouwbesluit (The Dutch Building Decree), this value is determined to be 3 meters. For each footprint, if the elevation of all the points falling within it is less than its ground elevation plus 3 meters, this footprint is ignored in roof calculation. Because no useful points could be obtained from the ICESat-2 data to represent its roof height. The comparison of either ignore these footprints or not in roof calculation is shown in Section B.3.

This approach reduces the number of footprints that ultimately have an effective height, but is a necessary step. One of the possible consequences is that out of all footprints in a municipality area, only a tiny number of footprints can have a building height got from ICESat-2 satellite data, considering the intersection with ICESat-2 and a reasonable roof height. This may affect the amount of data available for the latter machine learning methods and thus lead to unacceptable model results.

The two attributes from BAG are used as standard roof elevation, `h_dak_50p` and `h_dak_70p`. They show the roof is calculated as the median and 70th percentile of all elevation points on the corresponding roof part, respectively. The calculated roof elevation will be compared with these two values to evaluate the accuracy.

3.4. Calculate building height

Finally, the building height will be determined by the difference between the roof elevation and the ground elevation Equation 3.1.

$$Building\_Height = Elev\_r - Elev\_g \tag{3.1}$$

3.5. Machine learning method

3.5.1. Model generation

Once the building heights are obtained from ICESat-2, the Random Forest Regression (RFR) will be used to estimate the footprint heights that are not intersected with the ICESat-2 data. The geometric features from Lánský [2020] are used, because it covers all geometric features. The description and computation of these features are shown in Figure 3.7. Also plus perimeter and construction_year (from BAG) two features. Thus, there are eleven features in total.

These features will be further selected based on the method mentioned in Section 4.4.

Next, a correlation matrix is used to measure the correlation among features [Pham-Gia and Choulakian, 2014]. The value of the correlation matrix is between -1 and 1. A high correlation between two features means these two have a strong positive correlation. The closer the value to zero, the weaker the correlation between the two. A negative number shows a negative correlation between the two.

	Feature	Description	Computation
1.	Area	The area of the building footprint	-
2.	Compactness	The Normalised Perimeter Index (NPI)	$\frac{2\sqrt{\pi A}}{P}$
3.	Number of neighbours	Buildings within a range of 100 metres of the footprint	Centroid distance
4.	Complexity	The irregularities in the footprint	$\frac{P}{\sqrt[4]{A}}$
5.	Number of adjacent buildings	Buildings within 1 metre of the footprint	Buffer intersection
6.	Length	Longest edge of MBR	-
7.	Width	Shortest edge of MBR	-
8.	Slimness	Ratio of the sides	$\frac{F_{length}}{F_{width}}$
9.	Number of vertices	Total number of vertices in the footprint	-

Figure 3.7.: Geometric features. *Source:*Lánský [2020]

After correlation analysis, the Variance Inflation Factor (VIF) is calculated. This is to measure the level of multicollinearity in the multiple linear regression model [Forthofer et al., 2007]. The value of VIF is greater than 1. The closer the value of VIF is to 1, the less severe the multicollinearity is, and vice versa. The values above 5 or 10 indicate high collinearity [Garvin, 2013].

3.5.2. Model tuning

There are several hyperparameters of RFR algorithm available in the `scikit-learn` library. By adjusting these hyperparameters, the accuracy of the model can be influenced. Improper selection of hyperparameters can lead to underfitting or overfitting problems. Therefore, in order to obtain a more accurate model, it is necessary to perform tests and select the most suitable hyperparameters. There are two ways to choose hyperparameters, one is to fine-tune them by experience, and the other is to run the model with different parameters, and pick the best performing ones. These two methods will be used in combination with hyperparameter tuning in RFR.

Before going to the details of tuning method, two functions from `scikit-learn` library need to be explained in first. Because they will be used in tuning.

The first one is Grid Search Cross validation (GridSearchCV)[1], a method of tuning parameters by using exhaustive enumeration [Ranjan et al., 2019]. The name GridSearchCV can actually be split into two parts, GridSearch and Cross validation (CV), i.e., grid search and cross-validation. The grid search is actually to search for the parameters. Cross-validation is to prevent overfitting caused by overly complex models. The concept of cross-validation is to divide the training data set into K sets (Figure 3.8). One of the sets is taken in turn as the validation set, and the rest is the training set to train the model and test the accuracy of the model on the validation set. The average accuracy of K experiments is the average accuracy of the model.

The principle of GridSearchCV is like finding the maximum value in an array. What GridSearchCV does is: in the specified parameter range, the parameters are adjusted sequentially in steps, and the model is trained using the adjusted parameters to find the parameter with the highest accuracy on the validation set from all the parameters. This is actually a training and comparison process. Since it is required to iterate through all possible combinations of parameters, it

[1] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

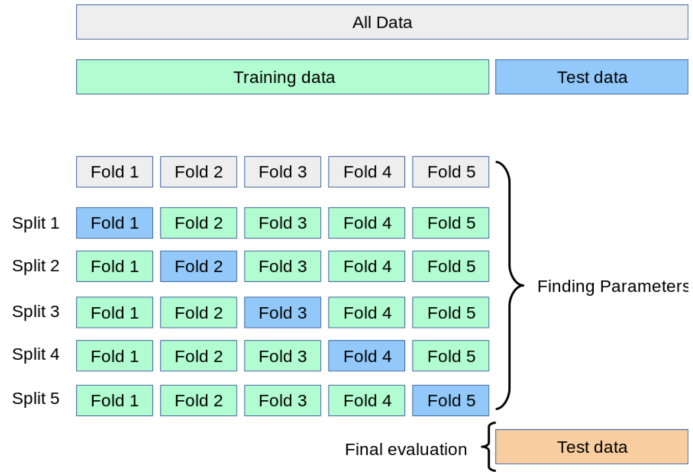


Figure 3.8.: Visualization of cross-validation *Source:* `scikit-learn` website

is guaranteed to find the parameter with the highest precision within the specified range of parameters. But it is very time-consuming in the face of large data sets and multiple parameters.

To alleviate GridSearchCV's time-consuming drawbacks, Randomized Search Cross validation (RandomizedSearchCV)[2] are proposed (Bergstra and Bengio [2012]). It is similar to the GridSearchCV, but instead of trying all possible combinations, it trains the model by a fixed number of parameter settings is sampled from the specified distributions (Figure 3.9).

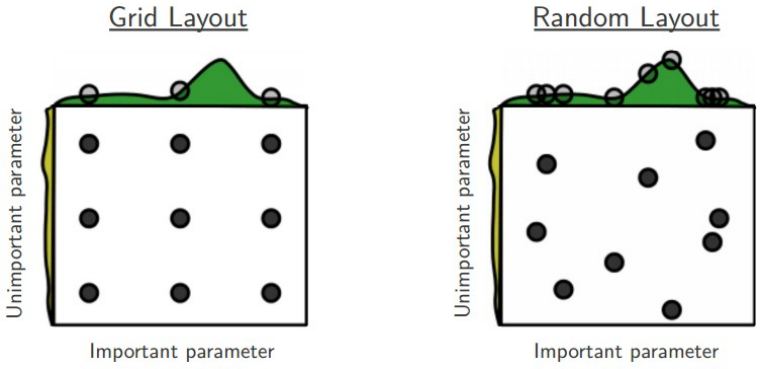


Figure 3.9.: Grid and random layout of parameters *Source:*Bergstra and Bengio [2012]

On the base of two methods mentioned above, the basic idea of RFR tuning used in this thesis is:

1. Determine the approximate selection range of each parameter to form a parameter dictionary. These approximate ranges can be obtained from previous studies

[2] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html`

2. Use RandomizedSearchCV to train the model with the parameters in the parameter dictionary. The best set of parameters in the random layout can be obtained.
3. Use the best parameter set obtained in the previous step as a criterion, take the surrounding values to form a new parameter selection range and a corresponding parameter dictionary.
4. Use sklearn's GridSearchCV, to train the model with the parameters in the new parameter dictionary to obtain the final best parameters

3.5.3. Model accuracy evaluation

Three error metrics are used to measure the accuracy of the prediction model. They are MAE, RMSE, R Squared (R^2 score) and Mean Absolute Percentage Error (MAPE).

MAE and RMSE are used to measure whether the correct value was predicted. R^2 score is used to measure whether sufficient information has been fitted to.

MAE is the average difference between the predicted value and the true value [Sammut and Webb, 2010]. It is used to assess closeness between the prediction results and the real data set (Equation 3.2). The smaller the value the better the result.

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \check{y}_i| \tag{3.2}$$

RMSE is the square root of the ratio of the square of the deviation of the observed value from the true value to the number of observations (Equation 3.3) [Chai and Draxler, 2014]. The consistency of the scale is ensured. RMSE is very sensitive to extreme errors in a set of data. The RMSE is used to measure the deviation between the observed value and the true value.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \check{y}_i|} \tag{3.3}$$

R^2 score determines how well the prediction model fits the real data (Equation 3.4). The numerator represents the sum of the squared differences between the real and predicted values. The denominator represents the sum of the squared differences between the real and mean values, similar to the variance. In normal situation, the result of R^2 score takes the range of (0,1) The closer to 1 means that the independent variable can explain the variance change of the dependent variable. The smaller the value means that the effect is worse. If R^2 score is 0, it means that the model fit is poor, and cannot predict the dependent variable. If the result is 1, it means the model is error free. If the result is a negative number, the model is performed poorly, even impossible to know how bad it is [Chicco et al., 2021].

$$1 - \frac{\sum_i(\check{y}_i - y_i)^2}{\sum_i(\bar{y}_i - y_i)^2} \tag{3.4}$$

MAPE is a measure used to forecast error (Equation 3.5), and works best if there are no extremes to the data (and no zeros) [Swamidass, 2000]. It's range is $[0, +\infty)$, a MAPE of 0% indicates a perfect model, and a MAPE greater than 100% indicates an inferior model.

$$\frac{100\%}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \tag{3.5}$$

4. Implementation

4.1. Datasets

The data sets required for this project come from two major sources, one is ICESat-2 ATL03 product (Version 5), the other is BAG dataset.

The ICESat-2 data provides point data, and the z-values of these points reflect the corresponding object elevation (like the ground and roof). ICESat-2 ATL03 product (Version 5) is the latest version, downloaded from EARTH DATA SEARCH and then convert into geopackage format from .h5 format with Julia and SpaceLiDAR.jl package [Pronk, 2022].

The BAG dataset will be used as ground truth to evaluate the calculated results, which is downloaded from 3D BAG. For BAG datasets, they have many layers and attributes. Such as "LOD12", "LOD13" which reflects the details of the different levels. The required properties from different layers will be combined to form a new geodataframe object (Figure 4.1). All relevant data distributed within the boundaries of the municipality will be collected. In the end, all these data are converted to EPSG: 7415 georeference with Geopandas, making sure they have a uniform georeference.

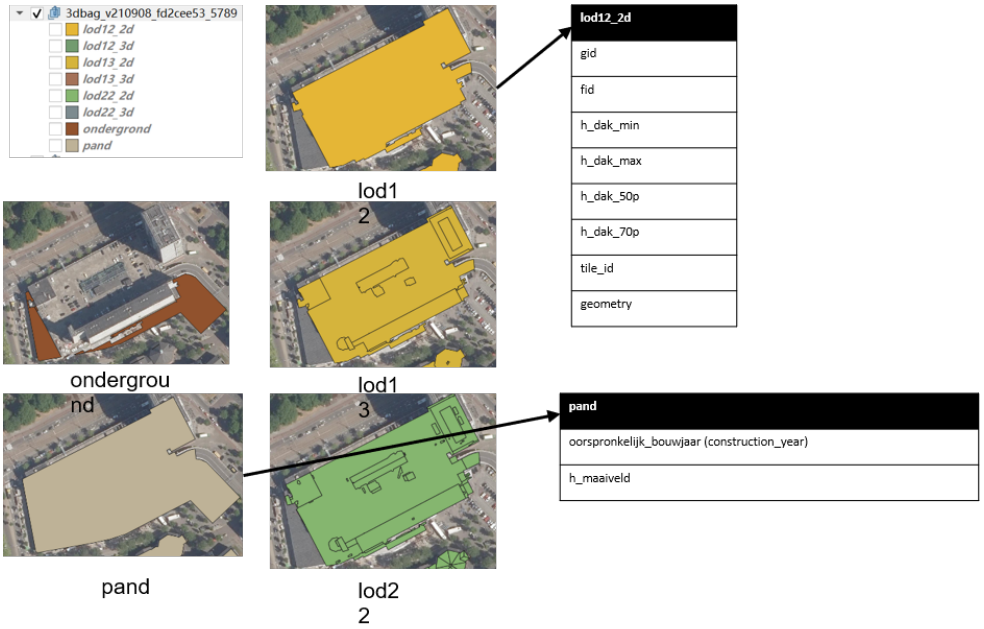


Figure 4.1.: Layers in one footprint

Considering the trajectory of the ICESat-2 satellite data, the topographic features of the Netherlands, and different municipal classes, Maastricht, Rijswijk and a northern village called Zuidbroek will be selected as experimental subjects. The location of these three cities in the Netherlands is shown in Figure 4.2. The topographic data is from Actueel Hoogtebestand Nederland (AHN) [1].

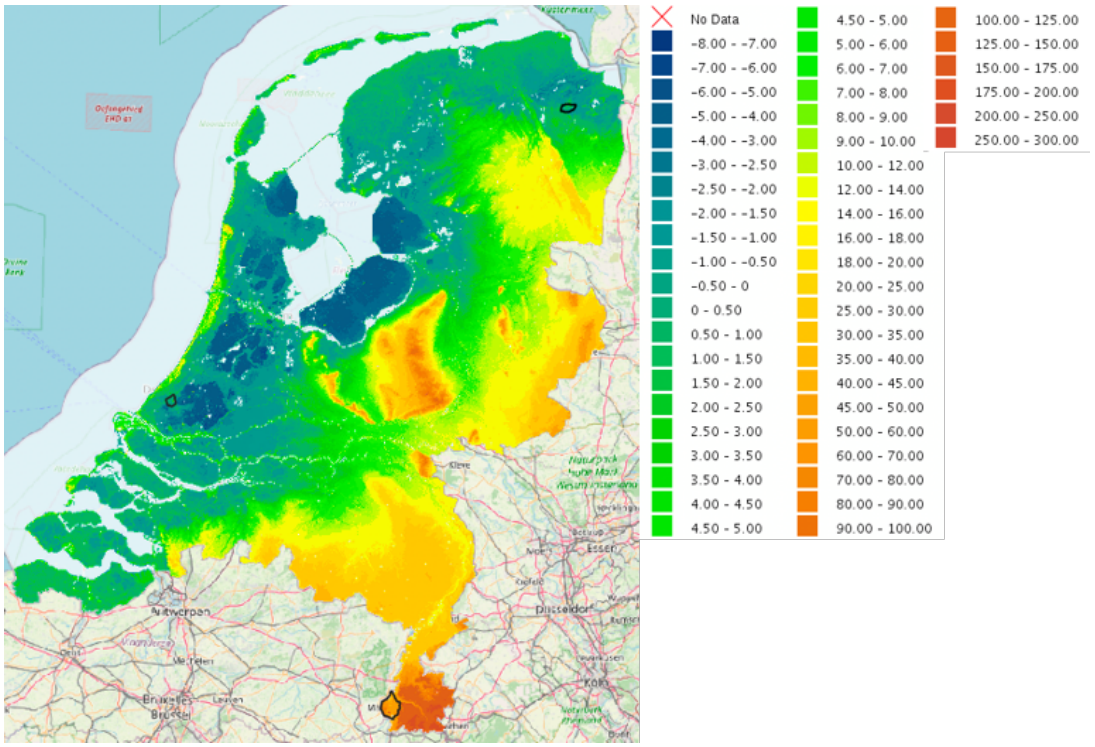


Figure 4.2.: Location of three datasets (in black edge)

Maastricht shares borders with Germany and Belgium. It's a university city on the southern tip of the Netherlands. Unlike most flat cities in the Netherlands, there is a hill to the south of the city that causes its topography to undulate. Rijswijk is in the Western Netherlands, between The Hague and Delft. It is on an flat coastal plain next to the North Sea. Zuidbroek is a small village in the province of Groningen. The three datasets correspond to cities, towns and villages and are in the south, west and north of the Netherlands, respectively. The purpose of this is to test whether the ICESat-2 data cover the whole of the Netherlands and to see whether the data performance is influenced by geographical or municipal influences.

Since ICESat-2 started publishing data in 13th October, 2018, all data within this time-frame are considered as of 1st April, 2022 data collection deadline.

However, within these three municipalities mentioned above, not all dates are covered, and the temporal distribution of the valid ICESat-2 data obtained is shown in Figure 4.3. It can be seen that in some time periods, such as 2021-02, Maastricht gets amounts of point data, while in other months, such as 2018-12 and 2020-06, there is almost no data. This type of situation also occurs within the other two datasets.

[1] https://www.ahn.nl/ahn-viewer

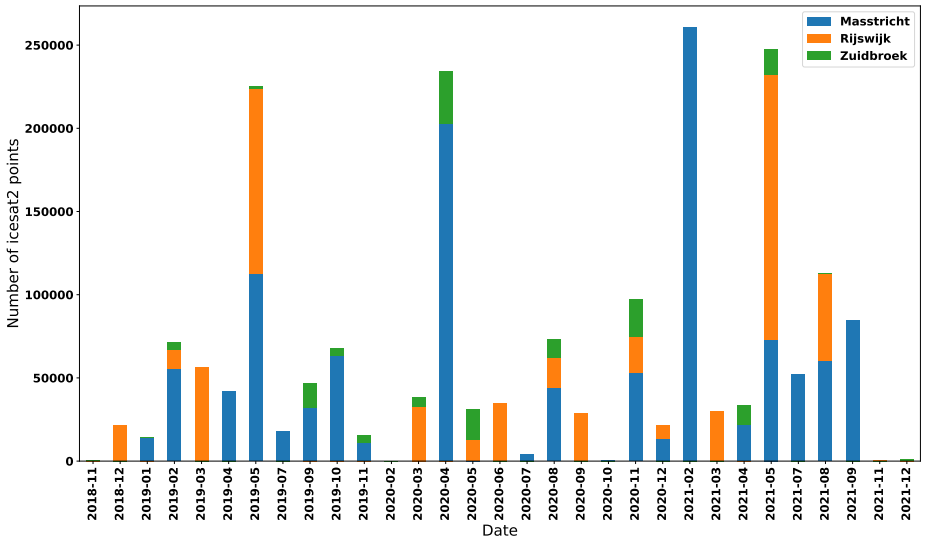


Figure 4.3.: Time distribution of ICESat-2 data in three datasets

Figure 4.4 provides an overview of spatial distribution of ICESat-2 points in three datasets. Clearly shows ICESat-2 points are not fully cover each dataset. Thus, the distribution of ICESat-2 is very random in time and space, at least within these three data sets.

The basic information of these three datasets, such as number of footprints, number of ICESat-2 data and area, is shown in the Table 4.1.

Table 4.1.: Basic information of Datasets

Municipality	No. ICESat-2 points	No. footprints	Area (km^2)
Maastricht	1,219,131	59,338	60.12
Rijswijk	597,636	17,684	14.49
Zuidbroek	146,693	2,825	17.28

4.2. Software

Python, Julia and QGIS are the three major tools used in this thesis project.

- Julia has the SpaceLiDAR library, which is easy and convenient to be used to download the ICESat-2 dataset and convert them into geopackage format.
- Python has several useful libraries that can be used in this thesis. Such as `geopandas`, `shapely`, `scikit-learn`, etc. These libraries are used to read and process the ICESat-2 data, such as filter the noise, classify the photons and calculate the building height for each footprint. Also used in generating prediction model.

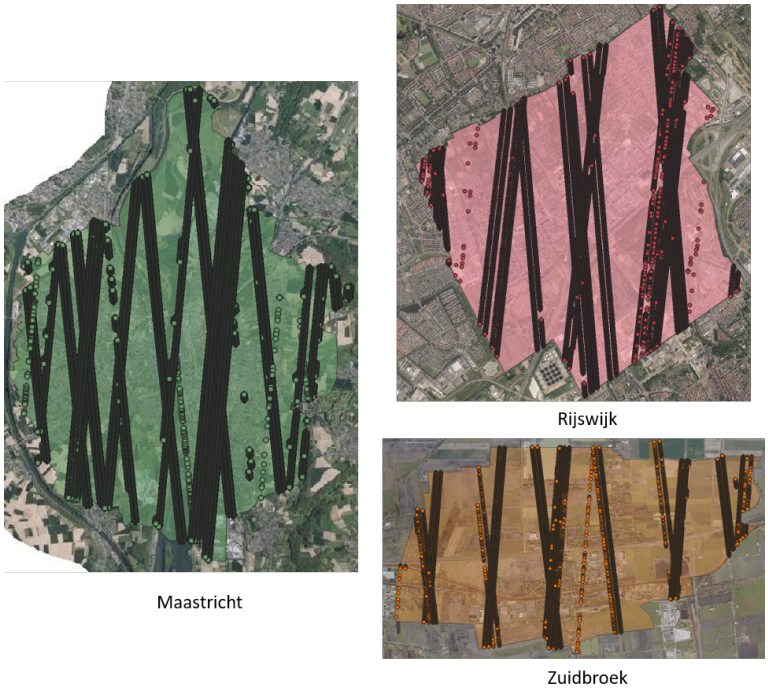


Figure 4.4.: Spatial distribution of ICESat-2 data in three datasets

- QGIS is used as a visual support tool during the entire process, visualize the ICESat-2 data and 3D BAG data (footprint of buildings).

4.3. Training dataset

The datasets used in training are from two parts. One is the building height information calculated by my method, another is the geometric features of footprints mentioned in Section 3.5.1, like area, length, etc.

There are only 2238 footprint samples from the training datasets containing all data from three datasets. The number of samples for each dataset in Maastricht, Rijswijk and Zuidbroek is 1640, 525 and 73, respectively. The comparison of the amount of valid footprint data and the amount of origin footprint data is shown in Table 4.2. It can be seen that less than three percent of the footprints in all three datasets have building heights obtained from the ICESat-2 satellite data compared to the original footprint data.

4.4. Feature selections

Feature selection is to select a suitable, relatively small subset of features from many features to be used as input to the ultimate model. Attributes that are useful for the current learning task are called ”relevant features” and attributes that are not useful are called ”irrelevant

Table 4.2.: The amount of final valid footprints

Municipality	No. footprints	No. intersected footprints	No. intersected footprints after filter	No. final valid footprints	Final valid footprint percentage
Masstricht	59,338	2,902	2,428	1,640	2.76%
Rijswijk	17,684	1,382	723	525	2.97%
Zuidbroek	2,725	140	107	73	2.68%

features" [Blum and Langley, 1997]. Selecting a subset of relevant features from a given set of features is called "feature selection". After feature selection, although the number of features is reduced, the model effect does not decrease significantly or even performs better [Blum and Langley, 1997].

Feature selection has three principal functions [Miche et al., 2007]. The first is to reduce the number of features, reduce dimensionality, make the model more generalizable, and reduce over-fitting. The second is to enhance people's understanding of the model, and fewer features are good for model interpretation. The third is that fewer features require fewer resources, which is beneficial for model training and inference, high modeling efficiency, and low maintenance cost.

The most common classification of feature selection methods are three major categories: Filter methods, Wrapper methods and Embedded methods [Jović et al., 2015].

In this section, three models based on each of the above three feature selection methods will be built. And these three models are evaluated and analyzed by correlation and multicollinearity among the keep features.

4.4.1. Filter method

The filtering method first requires selecting the scoring method. Then calculating the scores of all features, ranking the features, and finally filtering to get the selected features based on the threshold or the required number of features. This feature selection method does not involve subsequent model construction and is usually considered as an unbiased feature selection method Sánchez-Maroño et al. [2007].

The scoring methods are as follows:

Variance filtering

The variance metric is the simplest type of scoring method. The features are filtered by the variance of the features themselves Li et al. [2018]. Variance indicates the degree of dispersion of the data. If all the values of a feature are the same or close to each other, it means that its data distribution is more concentrated and not enough dispersion. In other words, this feature does not discriminate between the target variables. For example, a feature with a small variance means that the sample is essentially undifferentiated on that feature. One possibility is that most of the values in this feature are the same, or even that the entire feature takes the same values. Then, the feature is of little use for sample distinction. Therefore, the features with zero variance are to be filtered as a priority.

Correlation filtering

Correlation represents the relationship between features and features, and between features and target variables. A feature is correlated if a change in the feature value causes a change in the predicted value. The strength of the correlation represents the extent of this relationship. The goal of feature selection is to select features with strong correlation.

There are three commonly used methods to judge the correlation between features and labels in `scikit-learn`: chi-square test, F-test, and Mutual information and Mutual information and maximal information coefficient (MIC).

Chi-square test is a correlation filter for discrete labels (i.e., classification problems). It is not relevant to the regression model used in this thesis.

F-test, also known as ANOVA, is a filtering method used to capture the linear relationship between each feature and the label. It can do either regression or classification.

MIC is a filtering method used to capture arbitrary relationships (both linear and nonlinear) between each feature and label. Similar to the F-test, it can do both regression and classification. MIC between two random variables is a non-negative value, which measures the dependency between the variables. This estimator takes values between $[0, 1]$, where a value of 0 indicates that the two variables are independent and a value of 1 indicates that the two variables are perfectly correlated [Battiti, 1994]. In here, I'll use MIC to measure the correlation, because it can capture both linear and nonlinear relation.

Table 4.3.: The variance, MIC and VIF of selected features

Features	Variance	MIC	VIF
area	43618.3421	0.0530	22.8521
perimeter	3341.6861	0.0781	124.0321
construction_year	789.0000	0.2205	585.5995
length	348.2614	0.0741	31.6741
width	274.9515	0.0540	26.5574
complexity	194.0018	0.1028	358.4244
vertices	68.4588	0.0053	10.3204
neighbour	41.0000	0.0938	4.3174
slimness	0.7478	0.0581	15.6651
adjacent_buildings	0.5596	0.0000	-
compactness	0.0107	0.0601	286.8304

The result of the filter method is shown in Table 4.3. The variance values of all 11 features are not 0. There is only one feature's MIC value is 0, which is "adjacent_buildings". Indicating that this feature and the building height are two independent variables from each other. Thus, based on the filter method, 10 features except "adjacent_buildings" will be kept. VIF is used to measure the level of multicollinearity. Only one of ten is smaller than 5, indicating they are highly correlated with each other. The correlation matrix (see Figure 4.5) also shows a high correlation between "complexity", "length" and "width" these three features.

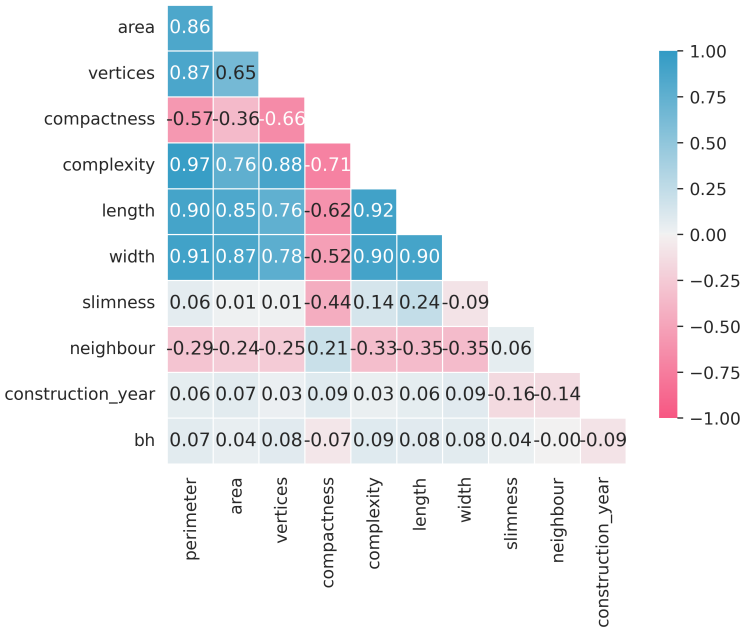


Figure 4.5.: Correlation matrix of filter method

4.4.2. Embedded method

The embedding method is a way to let the algorithm decide itself which features to use, which is means feature selection and algorithm training are performed simultaneously [Chandrashekar and Sahin, 2014]. When using the embedding method, certain machine learning algorithms and models are first trained to obtain the weight coefficients of each feature. These weight coefficients often represent some contribution or some importance of the features to the model.

When decided the machine learning algorithms as RFR the weight coefficients here are actually the `feature_importances_` attribute, which can list the contribution of each feature to the forest. The features are then selected based on the weight coefficients from the largest to the smallest. The threshold of feature importance can be determined with the help of the learning curve, and any feature whose importance is lower than this threshold will be removed.

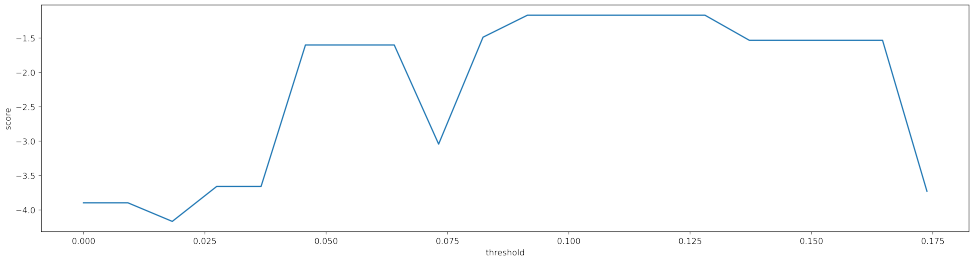


Figure 4.6.: Learning curve to find best threshold

From the learning curve (Figure 4.6), when the threshold is 0.09, the model has the best per-

formance. When set threshold as 0.09, there are 5 features retained (blue background in Table 4.4). Only one of five VIF values is smaller than 5, same as the result of the filter method. The correlation matrix (Figure 4.7) shows the "length" has the strongest correlation with "bh" (building height) among selected 5 features.

Table 4.4.: Feature_importances_ and VIF of selected features

Features	Feature_importances_	VIF
slimness	0.1992	7.3361
compactness	0.1893	116.2070
length	0.1437	9.3979
area	0.0995	4.8126
construction_year	0.0920	150.5292
complexity	0.0693	-
width	0.0654	-
neighbour	0.0605	-
perimeter	0.0511	-
vertices	0.0150	-
adjacent_buildings	0.0150	-

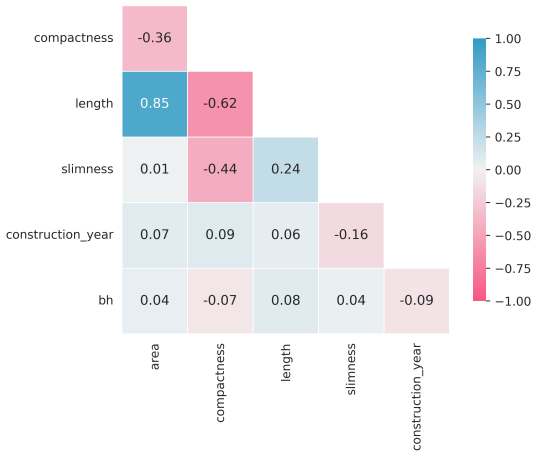


Figure 4.7.: Correlation matrix of embedded method

4.4.3. Wrapper method

The wrapper method is very similar to the embedded method and is also a method in which feature selection and algorithm training are performed simultaneously. It also relies on the algorithm's own properties, such as `coef_` properties or `feature_importances_` properties, to

perform feature selection. The difference, however, is that an objective function is used as a black box to help select features, rather than entering a threshold value.

The wrapper method trains the evaluator on the initial set of features and obtains the importance of each feature either through the `coef_` attribute or through the `feature_importances_` attribute. Then, the least important features are pruned from the current set of features. The process is repeated recursively on the pruned set until the desired number of features to be selected is finally reached. Unlike the filter and embedded methods, which solve all problems in one training, the wrapper method has to be trained several times using a subset of features, and therefore it requires the highest computational cost [Chandrashekar and Sahin, 2014].

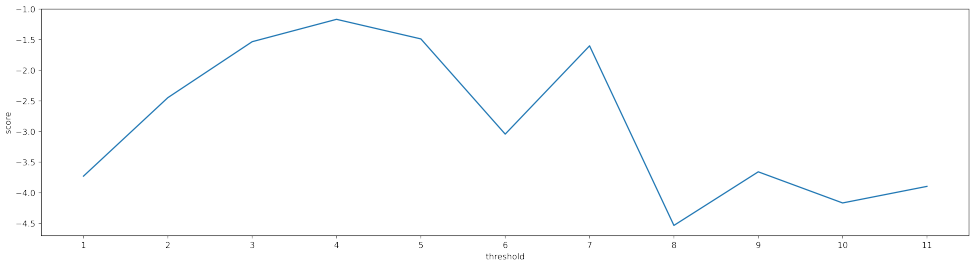


Figure 4.8.: Learning curve to find best number of retained features

From the learning curve plot (see Figure 4.8), it can be seen that the best amount of features to be retained is 4 (blue background in Table 4.5). The result of VIF is much better than the previous two methods. Two of them are smaller than 5, and the other two are closer to 5. The correlation matrix (Figure 4.9) shows the strongest correlation is between "length" and "area". And "length" also has the strongest correlation with "bh" (building height) among the selected 4 features.

Table 4.5.: Feature_importances_ of and VIF selected features

Features	Feature_importances_	VIF
slimness	0.1992	6.3519
compactness	0.1893	4.3945
length	0.1437	6.6754
area	0.0995	4.4167

4.5. Hyperparameter Tuning

In this section, the adjustment of hyperparameters was implemented. First, the general trend of the change of hyperparameter is displayed. Next, using the method mentioned in Section 3.5.2, hyperparameters are determined for each model.

In `scikit-learn`, the `RandomForestRegressor`[2] method has a lot of parameters. In this thesis, I mainly focused on six of them to do the hyperparameter tuning (See Table 4.6).

[2] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

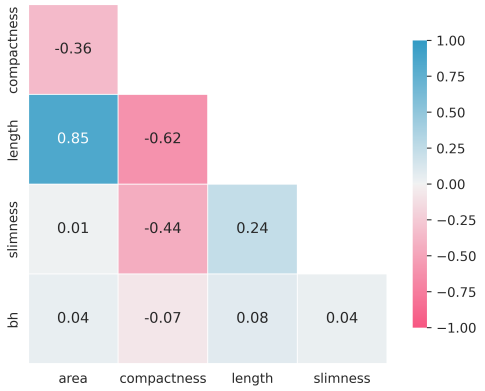


Figure 4.9.: Correlation matrix of wrapper method

Table 4.6.: Six hyperparameters for the RFR in the `scikit-learn` library

Hyperparameter	Type	Default	Description
`n_estimators`	*int*	100	The number of trees in the forest.
`max_depth`	*int*	None	The maximum depth of the tree.
`min_sample_split`	*int or float*	2	The minimum number of samples required to split an internal node.
`min_sample_leaf`	*int or float*	1	The minimum number of samples required to be at a leaf node.
`max_features`	*int or float*	auto	The number of features to consider when looking for the best split.
`bootstrap`	*bool*	True	Whether bootstrap samples are used when building trees.

4.5.1. General trend of change in hyperparameter

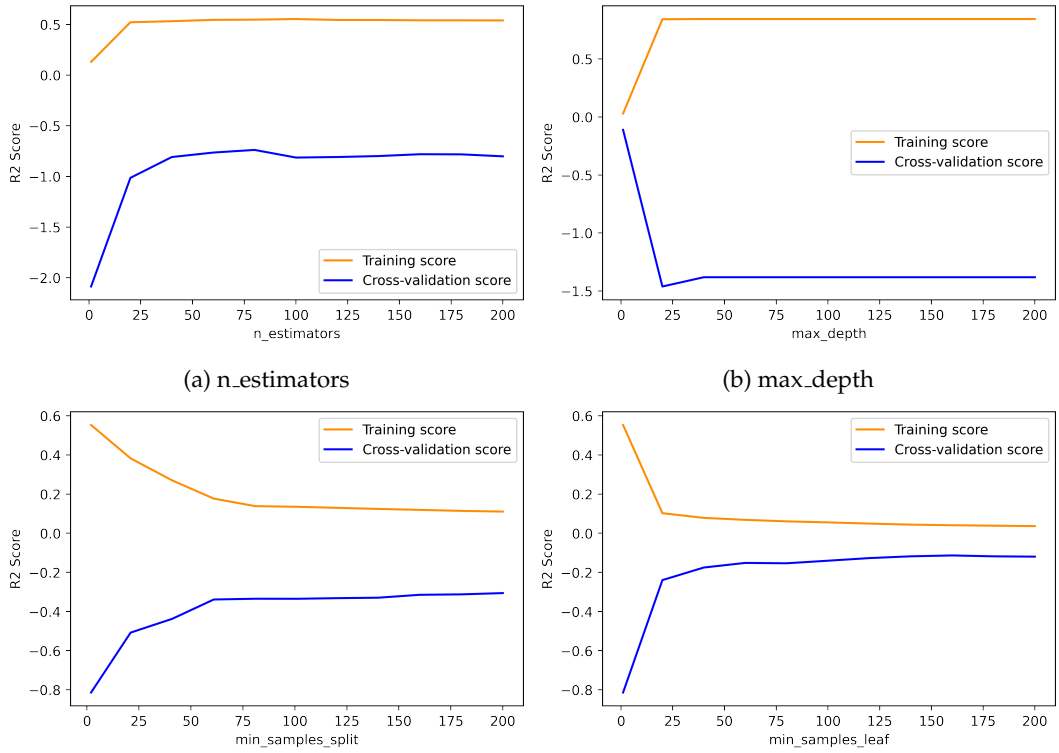
Figure 4.10 shows how the validation curve of the model (with all 11 features) changes as the parameters are varied. The validation curve is used to show how the model may go from underfit to fit to overfit as the hyperparameter settings are changed for the same model. The horizontal axis of the validation curve is some hyperparameter, such as `max_depth`, `min_sample_leaf`, etc. in some tree-integrated learning algorithms. The vertical axis represents the score, and the scoring method is chosen according to different model types (e.g.classification, regression). In here, I choose R^2 score to represent.

Figure 4.10a shows with the increase of `n_estimators`, the R^2 score for both training and cross-validation sets is first grows rapidly and then stays steady at around 40 without further change. Which means increase the number of estimators generally leads to better predictions. However, considering the complexity of the model also grows, this value should be set at the inflection point where the trend turns from upward to flat. This ensures that the model achieves a high level of performance while not being overly complex.

The increase of `max_depth` causes a decrease of R^2 score in cross-validation sets while a increase of that in training sets (Figure 4.10b). And the score of the training sets is always higher

than that of the cross-validation sets. This phenomenon indicates the model is overfitted as `max_depth` increases. Therefore, a smaller value of `max_depth` is more appropriate.

The change of R^2 score with the increase of `min_sample_leaf` and `min_sample_split` is similar (see Figure 4.10c and Figure 4.10d). The score of the training sets is first decrease and then stays steady. For the score of the cross-validation sets, it is increases first, then keeps steady. As the `min_sample_leaf` and `min_sample_split` increases, the score of training and cross-validation sets is getting closer.



(a) n_estimators

(b) max_depth

(c) min_samples_split

(d) min_samples_leaf

Figure 4.10.: Validation curves of four hyperparameters for RFR

4.5.2. Tested and selected hyperparameter

After understanding the general trend of the influence of the variation of hyperparameters on the model, the method mentioned in Section 3.5.2 is used to select hyperparameters for the three models obtained by the three feature selection methods and also the base model (maintain all features).

The tested details are shown in the two tables below. Table 4.7 shows the result of RandomizedSearchCV tuning method. Compared with the model selected by filter method, the two selected by embedded and wrapper method are more similar to each other. This is because the latter two keep very similar features. The features retained by the embedded method have only one more "construction_year" feature than the wrapper method.

The final selected hyperparameter is shown in Table 4.8. The test range of hyperparameter used in the test in GridSearchCV method is based on the result of RandomizedSearchCV.

Table 4.7.: Overview of tested and selected hyperparameter (RandomizedSearchCV)

Hyperparameter	Value tested			Value selected		
	Start	End	Num	Filter method	Embeded method	Wrapper method
n_estimators	1	200	11	100	200	200
max_depth	1	200	11	20	180	None
min_sample_split	2	200	11	81	2	2
min_sample_leaf	1	200	11	80	1	1
max_features(F)	1	10	11	7	-	-
max_features(E)	1	5	5	-	3	-
max_features(W)	1	4	4	-	-	4
bootstrap	True / False			True	True	True

Table 4.8.: Overview of tested and selected hyperparameter (GridSearchCV)

	Hyperparameter	Value tested			Value selected
		Start	End	Num	
Filter method	n_estimators	90	110	11	92
	max_depth	10	30	11	22
	min_sample_split	70	90	11	86
	min_sample_leaf	70	90	11	78
	max_features	6	7	2	7
	bootstrap	True / False			True
Embeded method	n_estimators	195	230	10	198
	max_depth	170	190	11	188
	min_sample_split	2	5	4	2
	min_sample_leaf	1	3	3	1
	max_features	3	5	3	4
	bootstrap	True / False			True
Wrapper method	n_estimators	195	230	10	202
	max_depth	None			None
	min_sample_split	2	4	3	2
	min_sample_leaf	1	3	3	1
	max_features	3	4	2	4
	bootstrap	True / False			True

5. Results

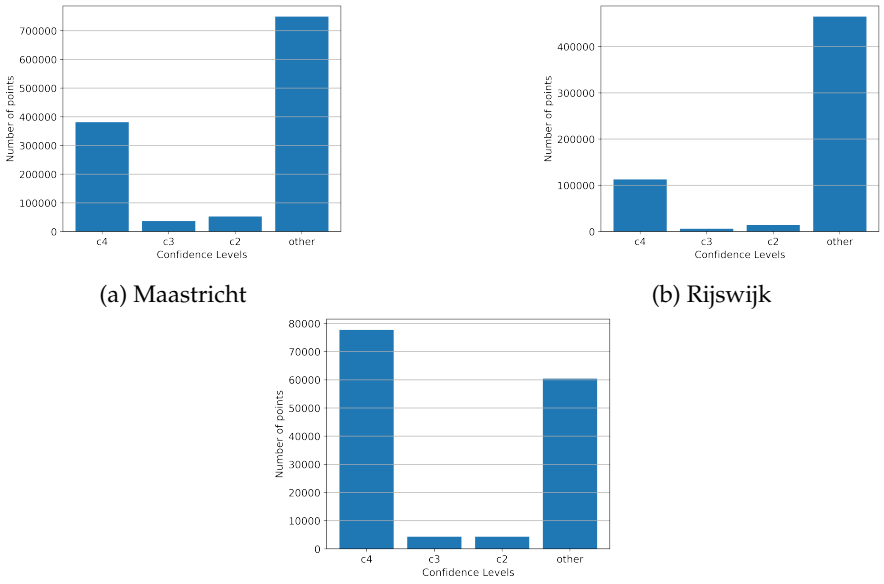
This chapter shows the results, experiments and analysis. Section 5.1 shows the result of data pre-processing. Section 5.2 shows the experiments in defining ground points and roof elevation percentile choosing. Section 5.3 contains the statistical analysis and cases study of error analysis. Section 5.4 is the model performance analysis.

5.1. Data pre-processing

5.1.1. Data cleaning

The ICESat-2 data was first cleaned using it's provided confidence property. The distribution of points with different confidence levels is shown in Section B.1.

Based on the attribute `confidence` provided in ICESat-2 data, these points are classified in to "c4", "c3", "c2" and "other". "c4", "c3" and "c2" represent the point with a confidence value of 4, 3 or 2. "other" means the point with a confidence value of 1 or 0 or -1.



(a) Maastricht

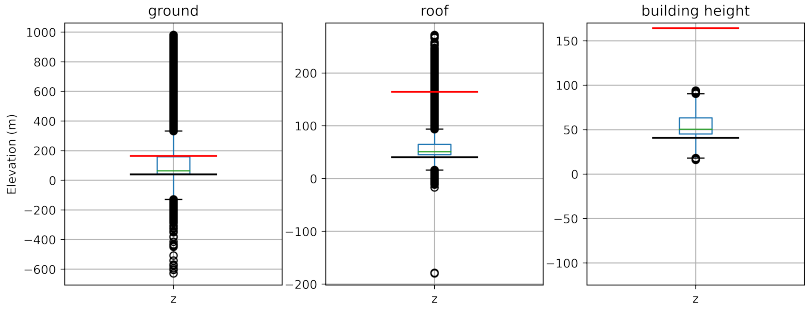
(b) Rijswijk

(c) Zuidbroek

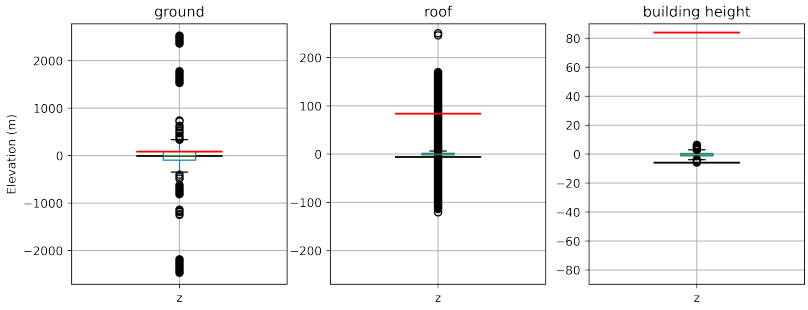
Figure 5.1.: The number of points with different confidence levels in three datasets

From Figure 5.1, it can be seen in Maastricht and Rijswijk, "other" category account for more than half of the total ICESat-2 data. And in Zuidbroek, this number is about 40%. While in Zuidbroek, half of the total number of ICESat-2 points with the confidence level of 4. Rijswijk data set has the least points with the confidence level of 4.

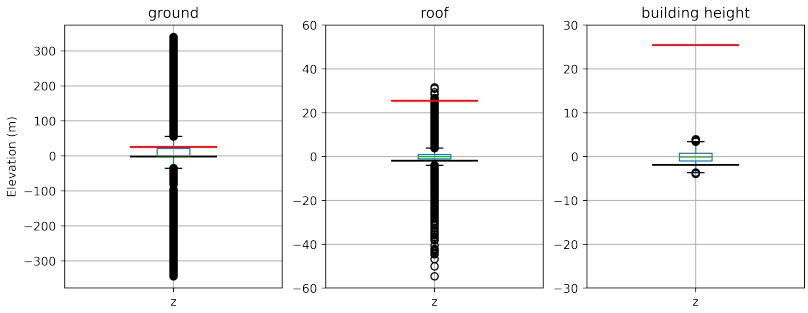
In this cleaning step, all points in "other" category are removed.



(a) Maastricht, reasonable range from BAG [40.67, 164.21]



(b) Rijswijk, reasonable range from BAG [-5.94, 83.93]



(c) Zuidbroek, reasonable range from BAG [-1.90, 25.43]

Figure 5.2.: Boxplot of data cleaning steps *(from left to right: original data, after confidence filter, after boxplot filter)*, black line is the minimum value of `h_maaiveld`, red line is the maximum value of `h_dak_max`

The next cleaning step is performed with the help of a box plot. After removing the outliers be identified by the box plot, the ultimate result of the data cleaning step is obtained. Figure 5.2 shows the box plot of original ICESat-2 data, and the data after each filtering step. The black

line is the minimum value of `h_maaiveld`, the red line is the maximum value of `h_dak_max`. The range between these two lines is the reasonable elevation range, also the aim of data cleaning.

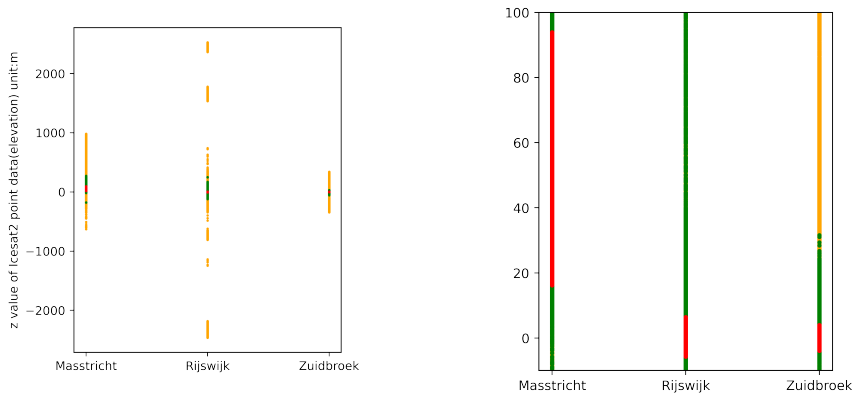
This box plot can be used to explain why the box plot filtering method is only used for ground elevation calculation, not for roof elevation calculation. From BAG data, it can be known the range of `h_maaiveld` and `h_dak_max` for three datasets. From the Figure 5.2, it can be seen after box plot filtering, that the maximum value of Maastricht is about 90m, while the maximum value from BAG is 164.2121m. This illustrates part of the roof point is missing after boxplot filtering. The minimum value is about 20m after boxplot filtering, smaller than the minimum value from BAG. This shows the ground points are not missing.

The situation is the same in Rijswijk and Zuidbroek. In Rijswijk, the maximum value after boxplot filtering is about 7m, much smaller than the maximum value from BAG which is 83.9725m. The minimum value is about -6m after boxplot filtering, smaller than the minimum value from BAG which is -5.4930m.

In Zuidbroek, the maximum value after boxplot filtering is about 4m, smaller than 25.4345m, which is the maximum value from BAG. The minimum value is about -4m after boxplot filtering, smaller than -1.8960m, which is the minimum value from BAG.

This comparison shows the data after box plot filtering is suitable for ground elevation, but not for the roof elevation. Because a lot of roof data is lost.

A scatter plot of z values of ICESat-2 data in three datasets is provided in Figure 5.3 to show the points kept and removed by the cleaning process. Figure 5.3b is to scale the y-axis to a range of -10m to 100m, demonstrating the range where all the red points (the final result of data cleaning) are located. The comparison with Figure 5.3a shows that more than half of the noise points were removed from the original ICESat-2 data. In short, 764,194 ICESat-2 points are removed from the Maastricht dataset, 486,545 and 65,507 for Rijswijk and Zuidbroek, respectively. Which is 62.68%, 81.41%, 44.66% of the total ICESat-2 points, respectively.



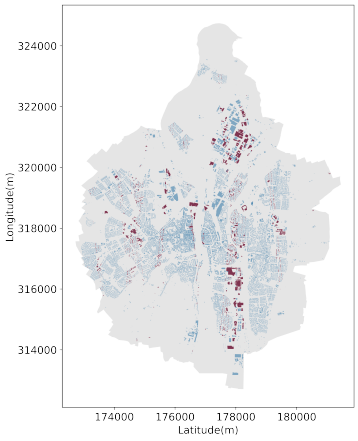
(a) Original size

(b) zoom in

Figure 5.3.: Scatter plot of ICESat-2 data kept and removed by the cleaning steps *(Yellow: raw data, Green: after confidence filter, Red: after boxplot filter)*

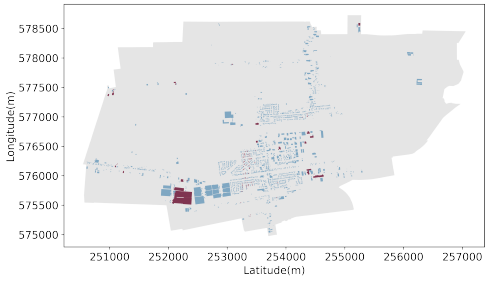
5.1.2. Intersection statistics

After filtering out the background noise, the intersection analysis of ICESat-2 point data and footprint data was performed. Only the ICESat-2 point locates inside the footprint is considered as the point belongs to that footprint. Figure 5.4 shows the intersection result, red footprint means it has at least one ICESat-2 point after cleaning, blue footprint means it is not intersected with ICESat-2 point. It can be observed that not all the footprints are intersected by the satellite data because of the sparsity of ICESat-2 data.



(a) Maastricht (4.09% intersected footprint)

(b) Rijswijk (4.09% intersected footprint)



(c) Zuidbroek (3.93% intersected footprint)

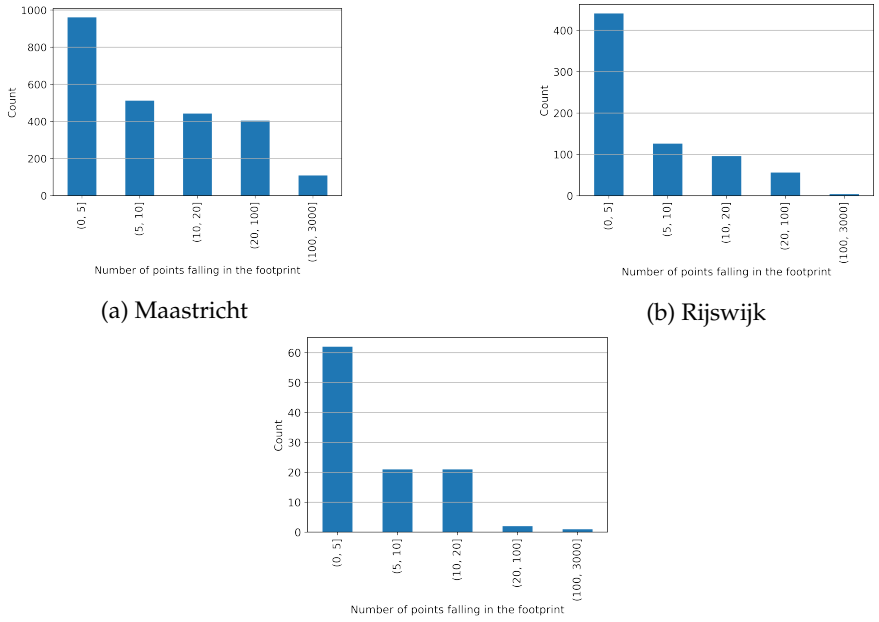
Figure 5.4.: The intersection of ICESat-2 and footprint *(Red means this footprint is intersected with ICESat-2 data)*

Table 5.1 summarizes the intersection of these three datasets before and after all cleaning steps. It can be seen that after filtering, about only four percent of the footprints of all three datasets are intersected with ICESat-2 point data.

Table 5.1.: Intersection before and after all filter steps

Municipality	No. Icesat-2 points		No. footprints	No. intersected footprints		Intersection percentage	
	Before	After		Before	After	Before	After
Masstricht	1,219,131	454,937	59,338	2,902	2,428	4.89%	4.09%
Rijswijk	597,636	111,019	17,684	1,382	723	7.81%	4.09%
Zuidbroek	146,693	81,186	2,725	140	107	5.14%	3.93%

Then, for those footprints that are intersected with the ICESat-2 data, the number of intersected ICESat-2 points of each footprint was counted. The number of points intersecting the footprint is grouped into five groups of boxes, (0,5], (5,10], (10,20], (20,100], (100,3000], and the number of each box is counted afterwards. The bar chart (Figure 5.5) shows in both Rijswijk and Zuidbroek, the most intersecting footprints have less than 5 ICESat-2 points. In Rijswijk, this number is 61% (441 out of 723), and 58% (62 put of 107) in Zuidbroek, 40% (961 out of 2428). In Maastricht, the situation is a little better compared with others, around 39% intersecting footprints have more than 10 ICESat-2 points.



(a) Maastricht

(b) Rijswijk

(c) Zuidbroek

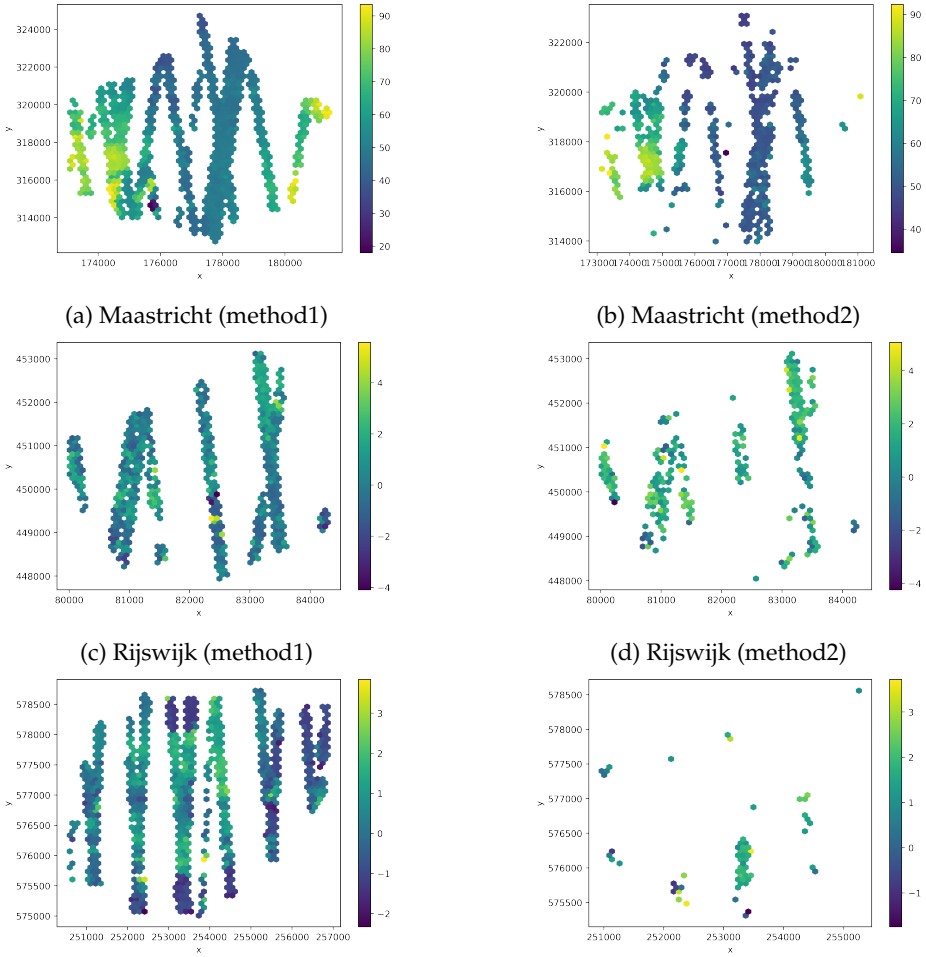
Figure 5.5.: The number of points falling in each intersected footprint

In summary, based on the analysis of the intersection, it can be concluded that about 4% of the footprints intersected with the ICESat-2 data, and nearly half of the intersecting footprints only have less than 5 ICESat-2 points.

5.2. Building Height

To calculate building's height, need to first obtain their roof elevation and also the ground elevation of their footprints. In this section, different methods are used to classify and calculate ground and roof elevation. And their results will be compared.

5.2.1. Ground Elevation



(a) Maastricht (method1)

(b) Maastricht (method2)

(c) Rijswijk (method1)

(d) Rijswijk (method2)

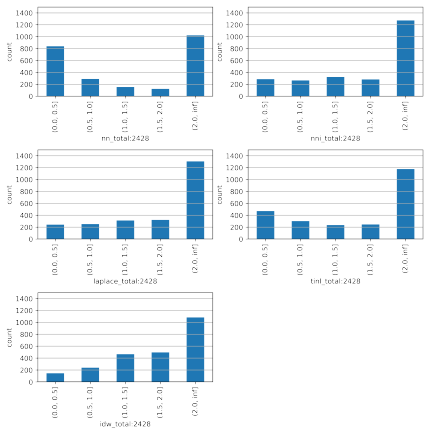
(e) Zuidbroek (method1)

(f) Zuidbroek (method2)

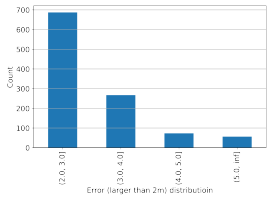
Figure 5.6.: Hexagonal bin plot of ground points (color bar represents elevation(m))

In Section 3.3.2, two methods of defining ground points are proposed. Figure 5.6 displays the ground points of these two methods in the three datasets. The horizontal axis indicates latitude, the vertical axis indicates longitude, and the different colors represent the elevation of the ground, which is the z-value of the ICESat-2 data. From Figure 5.6, it can be seen that when method-1 is used, more ground points are obtained and are more evenly distributed over the entire area range compared to method-2. This is significant in Zuidbroek data set.

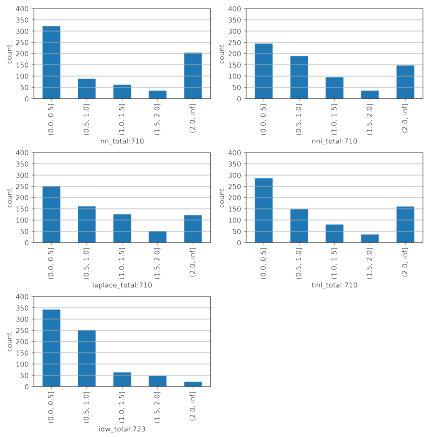
Considering that the entire region needs to be spatially interpolated in the next step, enough ground points should be obtained as much as possible. That means method-1 is more suitable for the need for spatial interpolation.



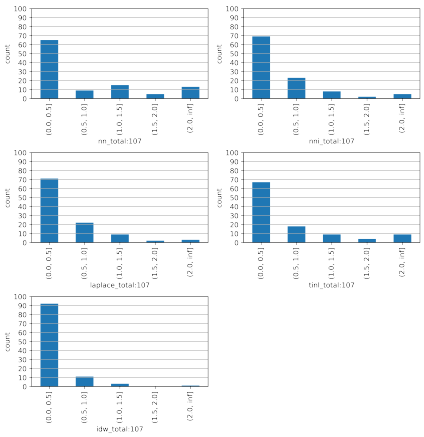
(a) Maastricht



(b) Maastricht with IDW (error larger than 2m)



(c) Rijswijk



(d) Zuidbroek

Figure 5.7.: The distribution of ground elevation errors with IDW (grid size 100m)

In the experiments, I used five interpolation methods (NN, NNI, Laplace, TINL, IDW) and four grid sizes (25, 50, 100, 150) to generate different combinations to calculate the ground elevation. Their results are compared with reference roof elevation from BAG. To characterize the distribution of errors from different combinations, all errors are grouped according to [0, 0.5), [0.5, 1.0), [1.5, 2.0), [1.5, 2.0) and larger than 2m, and bar graphs are drawn. So that

the distribution of errors can be observed and thus the most reasonable combination can be selected. These plots are shown in Section B.2. Consider the whole picture, IDW and 100m grid size are selected (Figure 5.7).

In Rijswijk and Zuidbroek, the optimal combination of grid size and interpolation method is 100m and IDW (Figure 5.7). It can be seen that among these five methods of spatial interpolation, IDW has the best results in both Rijswijk and Zuidbroek. In Rijswijk, there are close to 350 footprints with errors less than 0.5m, and the number of footprints with errors less than 1m is 600, together accounting for 83% of the total. In Zuidbroek, the number of footprints with error less than 0.5m is over 90, accounting for 85% of the total.

In the Maastricht area, the error distribution differs from Rijswijk and Zuidbroek. In all interpolation methods, most buildings have an error larger than 2m (Figure 5.7a). In terms of IDW method, there are 1083 out of 2428 buildings (about 43%) own errors larger than 2m (Figure 5.7b). Of these 1083 buildings, 687 of them have errors in (2.0, 3.0]m, 267 of them have errors between (3.0, 4.0]m, 73 of them have errors between (4.0, 5.0]m, and 56 of them larger than 5m. The largest error is 14.8163m. Figure 5.8 shows the location of those footprints has an error larger than 2m (in green color).

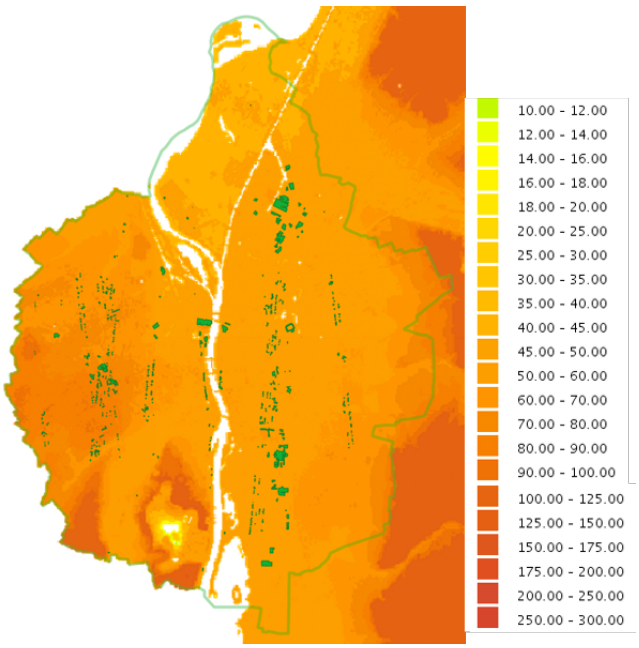


Figure 5.8.: The location of building with error higher than 2m in Maastricht (in green color)

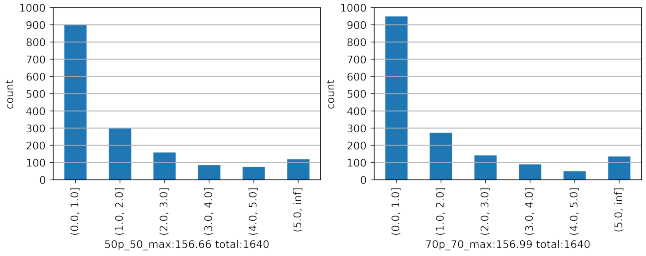
In summary, in the consideration of time cost and accuracy, IDW interpolation method and 100m grid size is the best combination to get ground elevation.

5.2.2. Roof Elevation

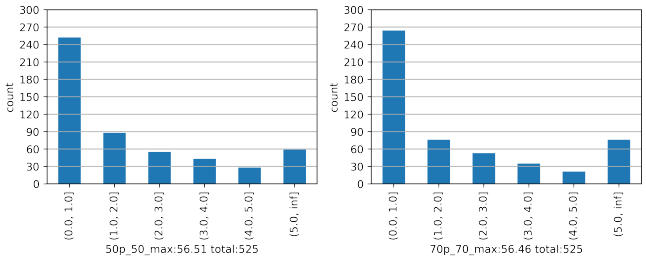
In the experiments, I used 50, 60, 70, 80, 85, 90, and 95 seven numbers as a percentile to calculate roof elevation. Their results are compared with reference roof elevation from BAG.

The plots are shown in Section B.3. Considering the entire picture, the 50 percentile is selected because of the relatively better performance. It has more data with smaller error.

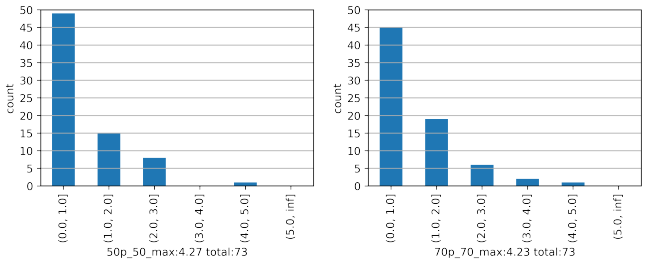
The error distribution of the roof calculation is presented in Figure 5.9. The errors are grouped into 6 categories, [0, 1.0), [1.0, 2.0), [2.0, 3.0), [3.0, 4.0), [4.0, 5.0) and larger than 5m. The bar shows the number of footprint in each category. In each subplot, the left one shows the error distribution with the `h_dak_50p` from BAG and the right one is the comparison with the `h_dak_70p`. The number of footprints and maximum error is shown in the x-axis label position.



(a) Maastricht



(b) Rijswijk



(c) Zuidbroek

Figure 5.9.: The distribution of roof elevation errors

In Zuidbroek data set (Figure 5.9c), it can be seen there are no footprints with errors greater than 5m and also no footprints with errors in (3.0, 4.0] meters. And the error distribution of the 50th percentile has more footprints in category (0.0 - 1.0] than the error distribution of the 70th percentile, 48 and 45 respectively. The maximum error is of the 50th percentile is 4.27m and 4.23m for the 70th percentile. Compared with the error distribution of the 70th percentile, the 50th percentile has more footprints with less error in general.

In Rijswijk (Figure 5.9b), the situation is similar. Though the result of the 70th percentile has

more footprints with errors in (0.0, 1.0] meters compared with the 50th percentile, the result of the 50th percentile perform better in general. The maximum error is of the 50th percentile is 56.51m and 56.46m for the 70th percentile.

In Maastricht (Figure 5.9a), the error distribution is the same as in Rijswijk. The maximum error is of the 50th percentile is 156.66 and 156.99m for the 70th percentile.

Table 5.2 shows the percentage of footprint with valid roof elevation in the whole intersected footprints. The value is about 70 percent for each data set, reducing the data in roof elevation calculation by 30%.

Table 5.2.: Percentage of footprint with valid roof points

	intersected footprint in total	footprint with valid roof	valided percentage	maximum error(50p)	maximum error(70p)
Maastricht	2428	1640	67.55%	156.66m	156.99m
Rijswijk	723	525	72.61%	56.51m	56.46m
Zuidbroek	107	73	68.22%	4.27m	4.23m

5.2.3. Building Height

From previous experiments, the ground and roof elevation is defined by the method with least error. That is IDW with 100m grid size and 50th percentile. Therefore, the building height is the difference between these two values.

Then all the buildings are divided into seven levels (Figure 5.10) according to the calculated height: buildings of 0-40 meters are divided into different levels every five meters, and above 40 meters are divided into one level.

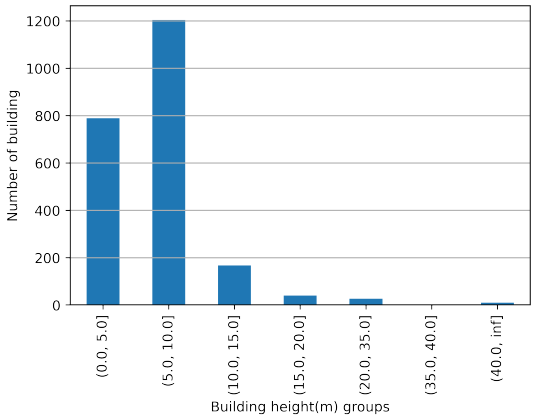
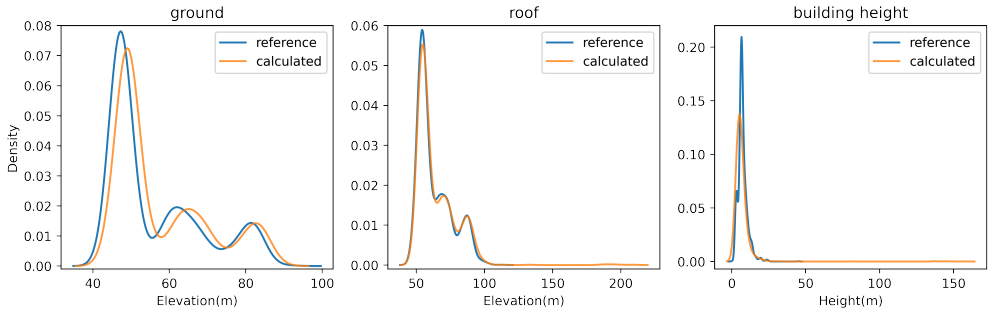


Figure 5.10.: Distribution of building height of three data sets

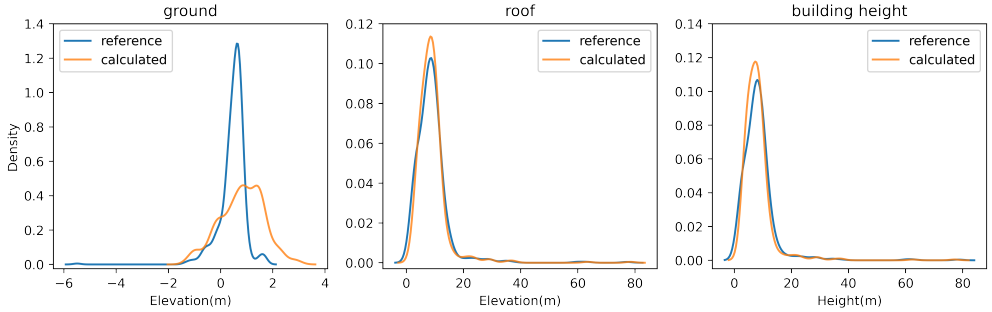
From Figure 5.10, there are 2238 buildings in total (results for all three data sets), 1200 of them have a height between 5 - 10m, more than 50%. And about 40% of these buildings have height in the range of 0 - 5m, which means buildings lower than ten meters tall accounted for 90% of the total building. Buildings over ten meters occupied only 10% of the total building.

5.3. Error Analysis

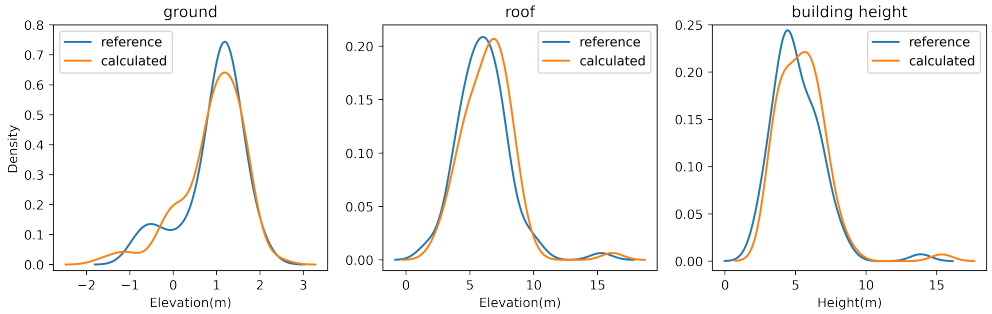
In this section, the error analysis will be performed by both data statistics and case studies. Section 5.3.1 shows the statistical information of calculated ground elevation, roof elevation, and building height in three data sets. Section 5.3.2 is the cases study. The top 10 maximum errors in each data set are chosen to do a more detailed analysis, which is 30 cases in total. Each case has a detailed analysis of the causes of error. Finally, I grouped the causes of errors into five categories which are listed in the Section 5.3.2.



(a) Maastricht



(b) Rijswijk



(c) Zuidbroek

Figure 5.11.: The density plot of BAG data and calculated data

5.3.1. Error statistics

In terms of region

Figure 5.11 shows the density plot of BAG data (ground truth) and the calculated data of ground elevation, roof elevation, and building height for these three data sets.

In terms of ground elevation calculation, the calculated values are overall larger than the reference value in all three data sets. In Maastricht (Figure 5.11a), the peak of reference value appears around 45m, while in the calculated result, the peak appears around 50m. This means in reference data, most of the ground elevation is around 45m, but this number is 50m in the calculated result. In Rijswijk (Figure 5.11b), most of the ground elevation is below 1m, while in the calculated result it's nearly 2m. And there is no ground elevation higher than 2m from reference data, however, some values are higher than 2m in the calculated result. In Zuidbroek (Figure 5.11c), three values from the calculated result are lower than -1m, while all values are higher than -1m in the reference data set. There are two peaks in the reference data set, the calculated data set has the same peaks at almost the same elevation values. But the peaks in the calculated data set are higher, which means it has more building in peaks.

In terms of roof elevation calculation, the performance of Rijswijk and Zuidbroek is better than Maastricht. In Maastricht, the calculated value is much larger than the reference value, especially since there are some abnormal values larger than 150m. In the reference data set, the highest value is only around 100m. This may be because the data used for the calculation contains outliers. In Rijswijk, the histograms of reference and calculated value overlap each other very well. It even has good overlap performance, even around larger elevation values (around 60m and 80m). In Zuidbroek, an offset can be observed, that is, the calculated data move a little to the right overall. This is more obvious at the left (2.5m) and right (10m and 15m) ends. Among the outcome of three data sets, the Rijswijk (see Figure 5.11b) has the best performance.

In terms of building height calculation, because the building height is determined by the difference between the roof elevation and the ground elevation, so the error from the ground and roof can affect the building height directly. This is well explained that the Maastricht data set has the largest error. This error could be inherited from the roof calculation, leading to the calculated result being much higher than the reference data. The outcome in Rijswijk is relatively good compared with the other two. There is also an offset in Zuidbroek, which can be observed in 5m, 9m, and 15m. The result of the calculated value in Zuidbroek is overall a little larger than the reference data.

In general, the performance of the Rijswijk data set is the best one among these data sets. The performance of Maastricht is the worst one because of extra large outliers.

Figure 5.12 shows the geographical distribution of the outliers in Maastricht. The outliers being set are those with `z` values greater than 165m. Because from BAG, it can be known the biggest `h_dak_max` is 164.2121m. They are concentrated in the northeastern region of Maastricht near the border. Most of the area covered by them is wooded, and a small part is residential. There are 2254 outliers in total. 2247 of them have the same timestamp `2019-04-29T05:37:59`, 7 of them have `2020-11-08T15:15:12` as their timestamp. However, not all points from these two timestamps are identified as outliers. The `confidence` attribute of these outliers is all 2. 1143 of them are from a strong beam, 1111 of them are from a weak beam, almost half and half.

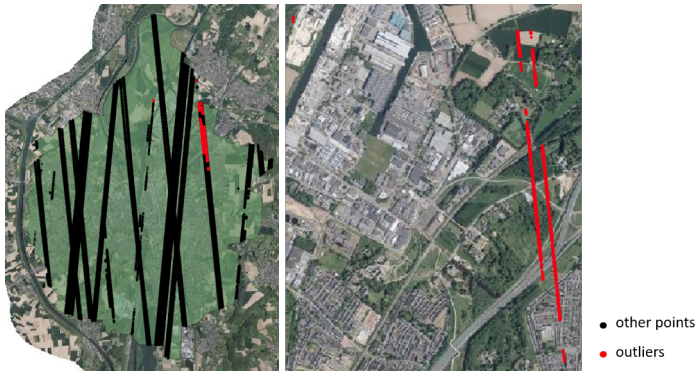


Figure 5.12.: Outliers in Maastricht (red points)

In terms of building height groups

Figure 5.13 shows the absolute error (mean, minimum, maximum) between the calculated roof, ground elevation, and the reference value for buildings within seven heights classes.

In terms of building height difference: It can be seen the group with the smallest mean absolute error is the group with a building height of 5-10 meters (1.8415m). The building height group of more than 40m owns the largest mean absolute error (92.3899m) also the largest maximum absolute error (155.4244m). The second-largest mean absolute error (21.3900m) is owned by the 35-40m group. And the 35-40m group also has the largest minimum error (8.1787m). Although the 0-5m height group has a quite small value in minimum absolute error (0.0041m), it has the second-largest error in maximum absolute error (55.7198m) at the same time compared to other heights groups.

Thus, a conclusion can be derived from this: the 5-10m building height group has the largest amount of buildings and the smallest mean absolute error (1.8415m). Then, as the building height increase, the number of buildings in each group decreases while the mean absolute error rises. When building height is lower than 5m (0-5m group), it has the second smallest mean absolute error (2.4752m) also the second-largest maximum absolute error (55.7198m).

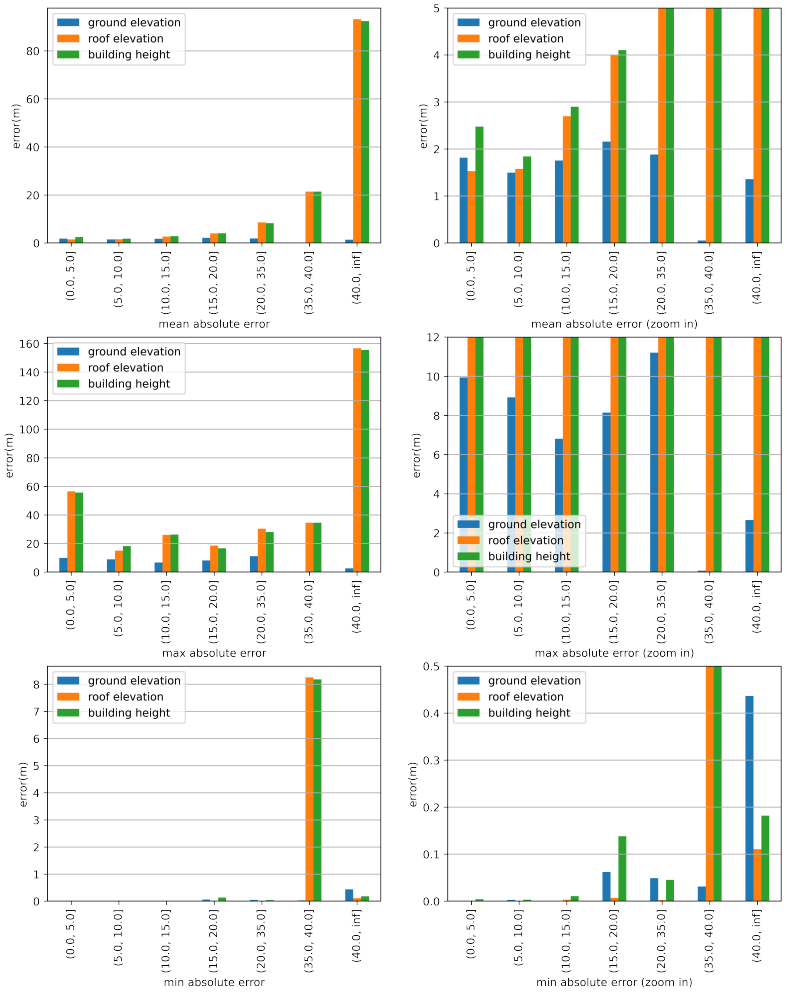


Figure 5.13.: Absolute difference of calculated and reference value among different building height levels

5.3.2. Case study

The difference between the roof elevation and the ground elevation determines the building height, thus the error analysis of the building height will be based on these two sections.

The ground elevation is estimated using the interpolation approach, which uses all points outside the footprints. The roof elevation is determined using the percentile approach from the points inside each footprint. Thus, compared with ground elevation calculation, the elevation of ICESat-2 points has a more remarkable and directly influence in roof elevation calculation.

In this case study part, I will focus on the first ten footprints with the largest error in roof calculation in each dataset. Scatter plot is used to show the distribution of elevation of points in each footprint. Each scatter plot not only shows the elevation information of ICESat-2 data

inside each footprint, but also the reference roof elevation (black lines) and calculated roof elevation (red lines) in each footprint.

The title of each scatter plot shows the `gid` of this footprint, and also the amount of valid roof data located inside this footprint. The reference elevation and calculated elevation of ground and roof are shown in the `x_label` position. The `g` means `ground`, `r` means `roof`. The `r` in the brackets means reference, `c` means calculated.

After getting the scatter plot, QGIS with Luchtfoto 2021 Ortho HR [1] and Google Earth is used to check why there is an large error.

After examining the top ten largest roof errors for each data set, I grouped the causes of the errors into the following categories: More details are given in the following specific cases.

1) Not enough valid data. Two cases exist in this category. One is that the overall number of valid points is tiny, maybe only one or two, which is not enough to calculate the roof elevation accurately. The other is that although the number of valid points is large, most of them do not capture the roof elevation information accurately, resulting in errors. Normally, making the calculated value lower.
2) Irregular roof shape. Usually, this is the case where a footprint actually comprises several parts of the roof and the elevations of these roofs are not the same. The non-uniform roof heights introduce errors into the calculations, making the calculated value either higher or lower.
3) Influence from surrounding objects. In most cases, those footprints which are easily be influenced by surrounding objects are not the building in the traditional sense. For example, a self-built carport in the backyard, a detached garage next to a residence, and underground parking lots. These buildings also have their own footprint in the BAG. These buildings are lower than other buildings around them and are often in areas with high building density. Therefore, the roof elevation is easily affected by the surrounding buildings or trees because of shading. Even though the ICESat-2 points are located inside its footprint, it may contain the elevation information of other objects. This makes the calculated value either high or lower.
4) Effect of significant outliers. This situation is only found in the Maastricht dataset. In some footprint, there are significant outliers (elevation is above 200m), causing the error.

Cases in Maastricht

Figure 5.15 shows the top 10 footprints with maximum errors in Maastricht.

Reason 4 is one cause of all these 10 cases. Except reason 4, reason 1 also cause the error in first, second, third, fourth, sixth, ninth and tenth. It can be inferred from the scatter plot (Figure 5.15) that even if there are no extra outliers, the seven footprints mentioned above still don't have enough valid roof points be used in calculation. If the outliers are excluded, there will be no valid roof points in the first, sixth, eighth, ninth and tenth one, only one valid roof point in the second, third, fourth, seventh one. Thus, the amount of valid points is obviously not enough for roof calculation.

But for the fifth one, it can be observed that it has enough valid roof points even if there are no outliers. This indicates that if the outliers are removed, the error of roof elevation

[1] https://data.overheid.nl/en/dataset/16186-luchtfoto-2021-hr-rgb-open-data

calculation in Maastricht will become smaller. Meanwhile, the amount of valid footprints is also decrease.

For the third one, reason 2 irregular roof shape is also a factor needs to be considered. From Figure 5.14 can be seen that there are actually three parts in this footprint (`gid` = 25975275). And each part has a different roof elevation.

Same situation is also exists in the fifth and the ninth one. From Figure 5.14, it can be known two parts with different elevation compose the fifth one with `gid` 25975346. And four parts compose the ninth one with `gid` 7013976.

Reason 3 is another cause of the eighth one. From Figure 5.16, it can be inferred this footprint belongs to a garage, and there is a tree that shades about a quarter of the footprint area. Thus, the only one point within this footprint which has the elevation of 8m, could probably actually reflects the tree height.

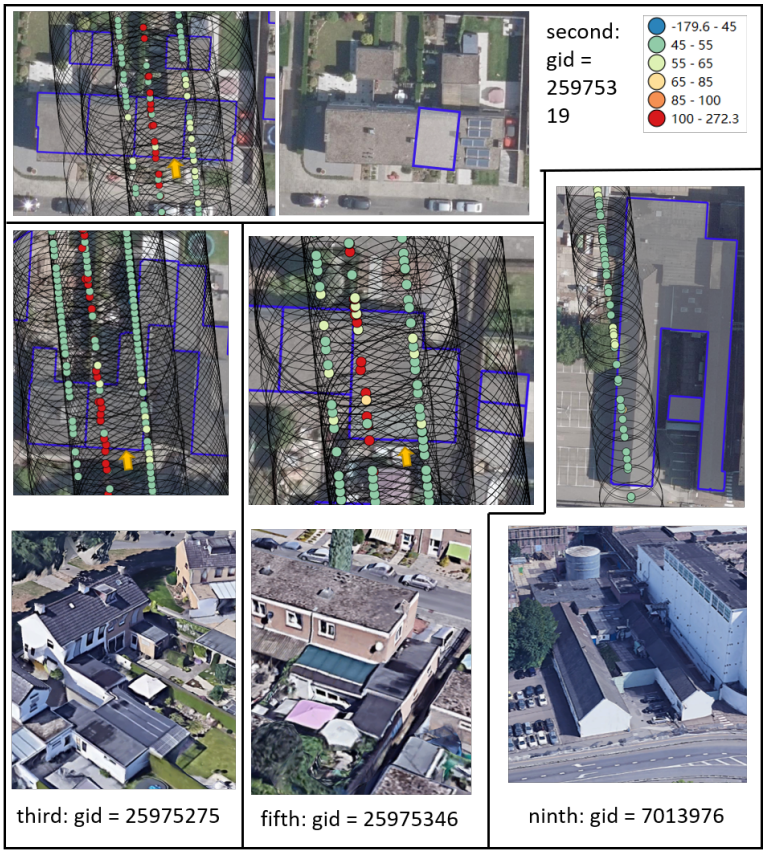


Figure 5.14.: Case study buildings (top ten footprint with maximum errors) in Maastricht (part 1)

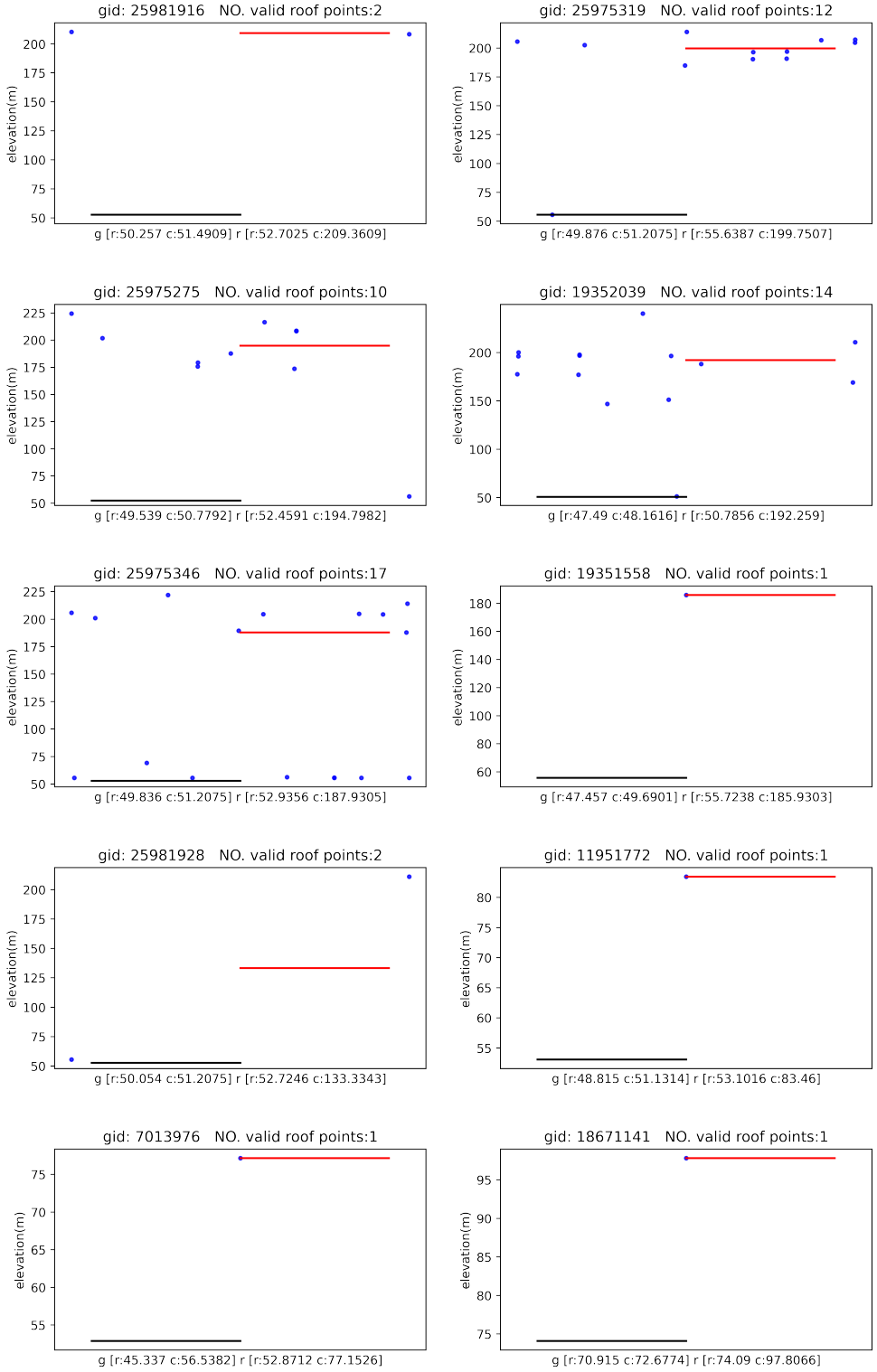


Figure 5.15.: Top ten footprint with maximum errors in Maastricht (Black line: reference value, red line: calculated value)

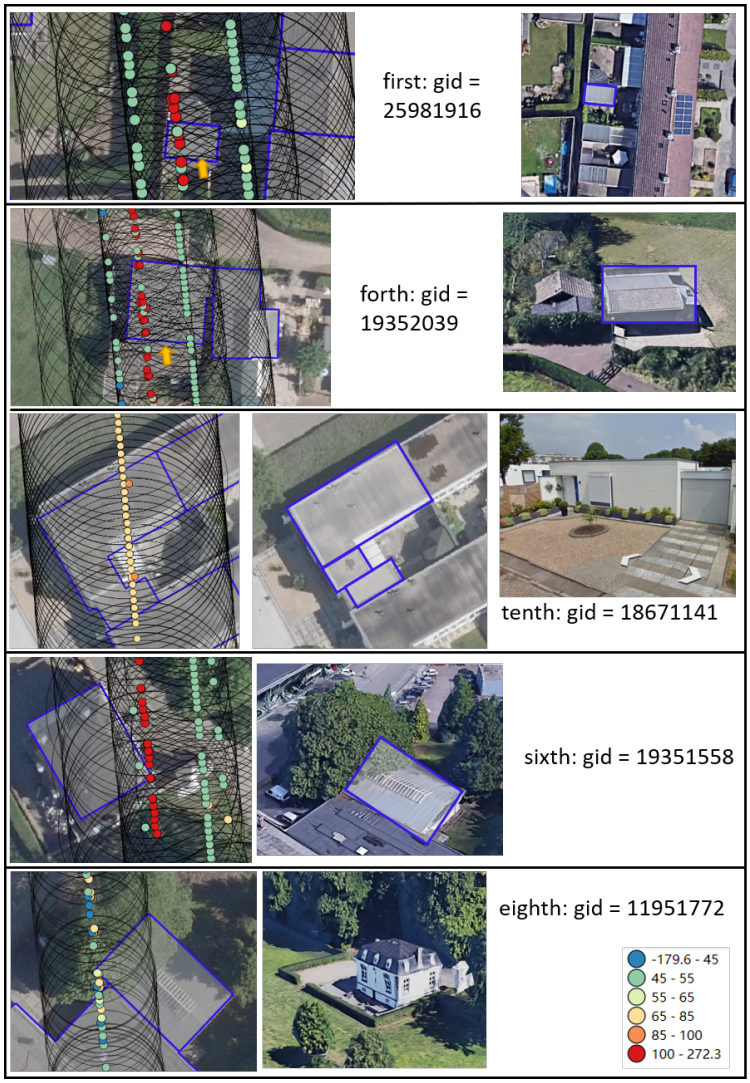


Figure 5.16.: Case study buildings (top ten footprint with maximum errors) in Maastricht (part 2)

Cases in Rijswijk

The scatter plot of Rijswijk is displayed in Figure 5.17. Next, the sources of the Rijswijk's top ten roof errors are analyzed.

The error of the first one in Figure 5.17 is caused by the reason 1. Though there are 536 ICESat-2 points inside it, most of them are obtained the elevation of ground. Thus, there are not enough valid roof points, leading to the calculated value is much lower than the reference value.

The fifth one is also caused by the reason 1. In this one footprint, the ground points make up the majority of the points, with only a few capturing the roof elevation information. This

results in a smaller roof elevation than the reference value using the 50th percentile calculation.

The second one (from left to right) in Figure 5.17 is caused by the reason 3. Actually, from check Google Map, this footprint is actually representing the top of an underground parking lot (The yellow polygon in Figure 5.18). The purple polygon represents the residential building around it. From Google Map, the garage door is demonstrated (the third figure in Figure 5.18). The reason for calculating value is higher than referenced is the points inside the footprint are "contaminated" by the surrounded tall building.

The third one is caused by reason 1. From Figure 5.18, it looks like there is an offset of the footprint (red line). But this is actually due to projection. The location of the footprint does match the actual. So the root cause is still not capturing enough effective roof points.

The fourth, sixth, eighth, ninth and tenth are all caused by the reason 1. Among them, fourth and eighth actually represent self-built buildings in the backyard, such as a carport. Sixth, ninth, and tenth represent garages. These buildings are usually lower than the surrounding buildings and are therefore vulnerable to overshadowing by the surrounding buildings. Thus it leads to the possibility that ICESat-2 falling within them may also capture elevations that are not belong to them.

The seventh is caused by reason 2. This building is Rijswijk Schouwburg, a performing arts theater. It can be seen that it has a complicated, designed roof shape, contains several parts and each part has a different height. The red line represents ICESat-2 data. The ICESat-2 data were mainly swept from the highest part, which caused the roof elevation obtained by the calculation to be much larger than the reference value.

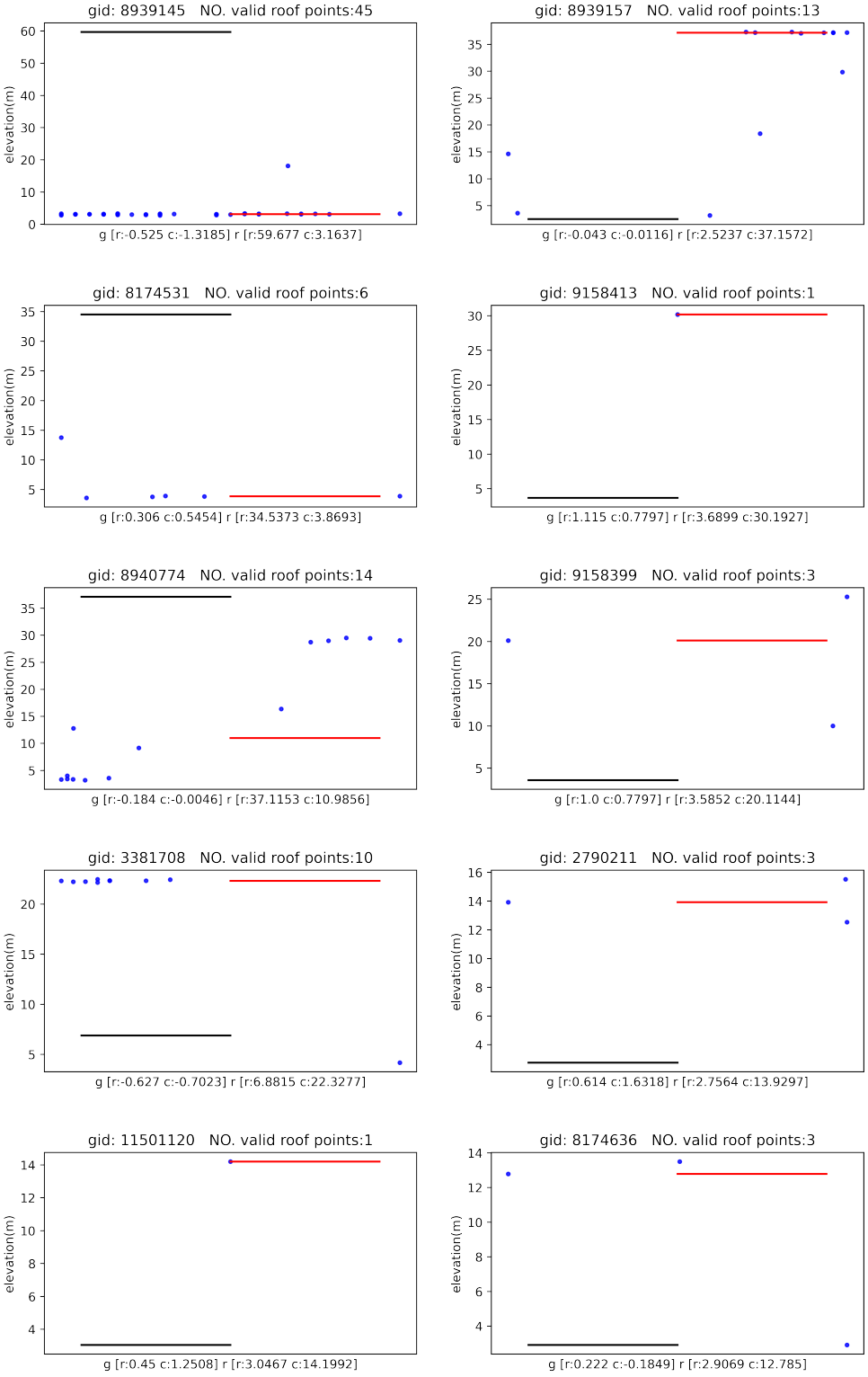


Figure 5.17.: Top ten footprint with maximum errors in Rijswijk (Black line: reference value, red line: calculated value)

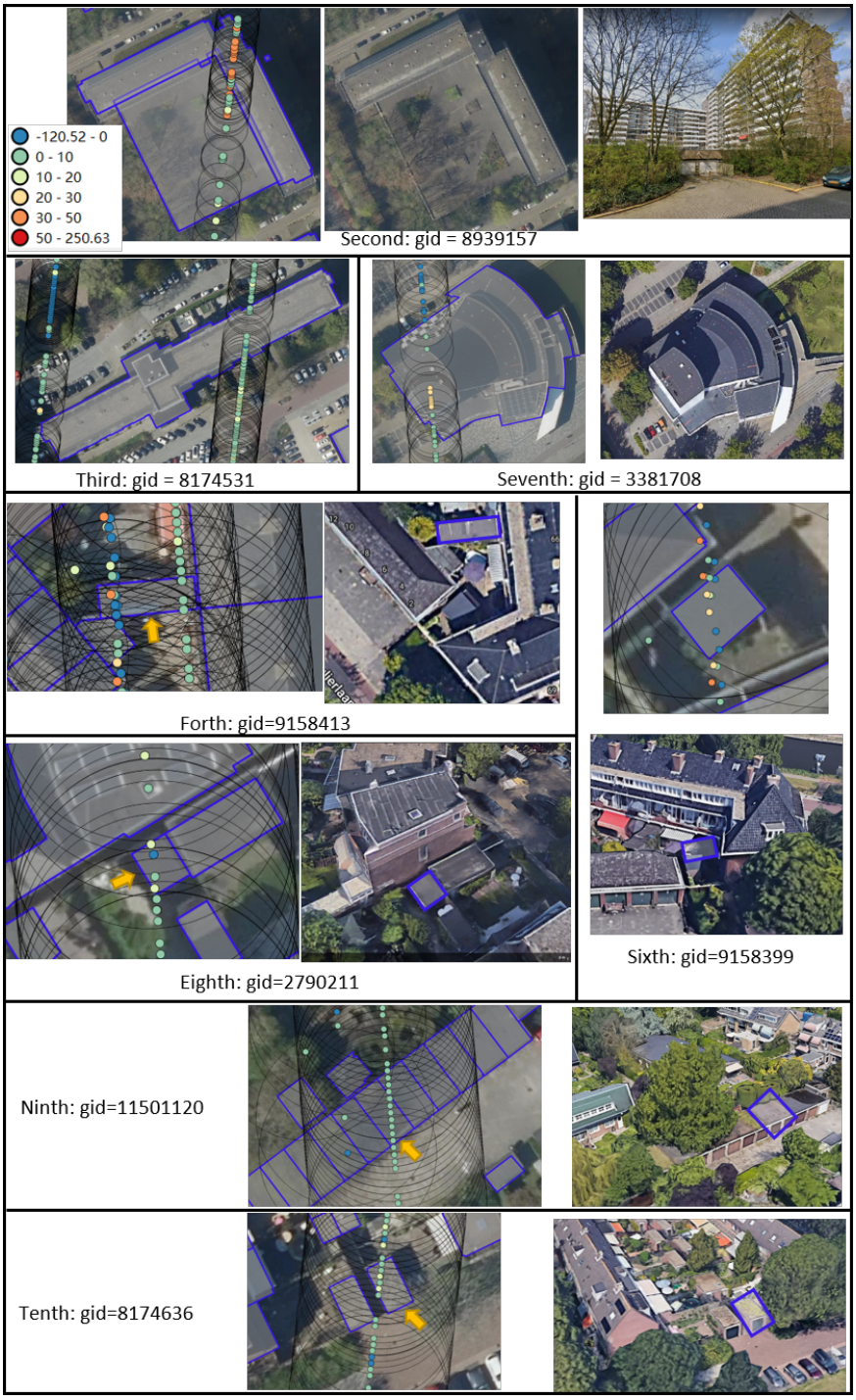


Figure 5.18.: Case study buildings (top ten footprint with maximum errors) in Rijswijk

Cases in Zuidbroek

Figure 5.19 shows the top ten footprints, with maximum errors in Zuidbroek. When comes to find the reason there is an enormous error, things in Zuidbroek is simplicity than Rijswijk. The biggest two reasons are reason 1 and reason 6.

The largest error of roof elevation in Zuidbroek is occurs in the footprint with a `gid` of 11986197. From the first scatter plot in Figure 5.19, it can be clearly seen that there are only two points inside this footprint. One's elevation is more than 8m, the other's is about 0m. This results in a roof height of 8.6248m calculated according to the 50th percentile method, but in reality the 50p reference elevation from the BAG is only 4.3521m. This directly produces an error of over four meters. Thus, this enormous error is caused by reason 1.

Reason 1 causes error in the third, fifth, eighth and tenth. In the third one, the reference value is 9.1893m, while the highest point got from ICESat-2 is only around 8m. Things are the same in the fifth. In the eighth one, the reference value is 10.2984m while the calculated value is 8.0882m. Though there is one point with around 10m elevation, it's not enough.

Reason 6 causes error in the second, fourth, sixth and ninth footprint. It can be seen from Figure 5.20 that these four buildings have irregular roof shape. This introduces some computational errors.

For the seventh, the cause should be reason 3. This building is a self-built garage. And a tall tree stands close to it, which may cause the calculated value is higher than the reference value.

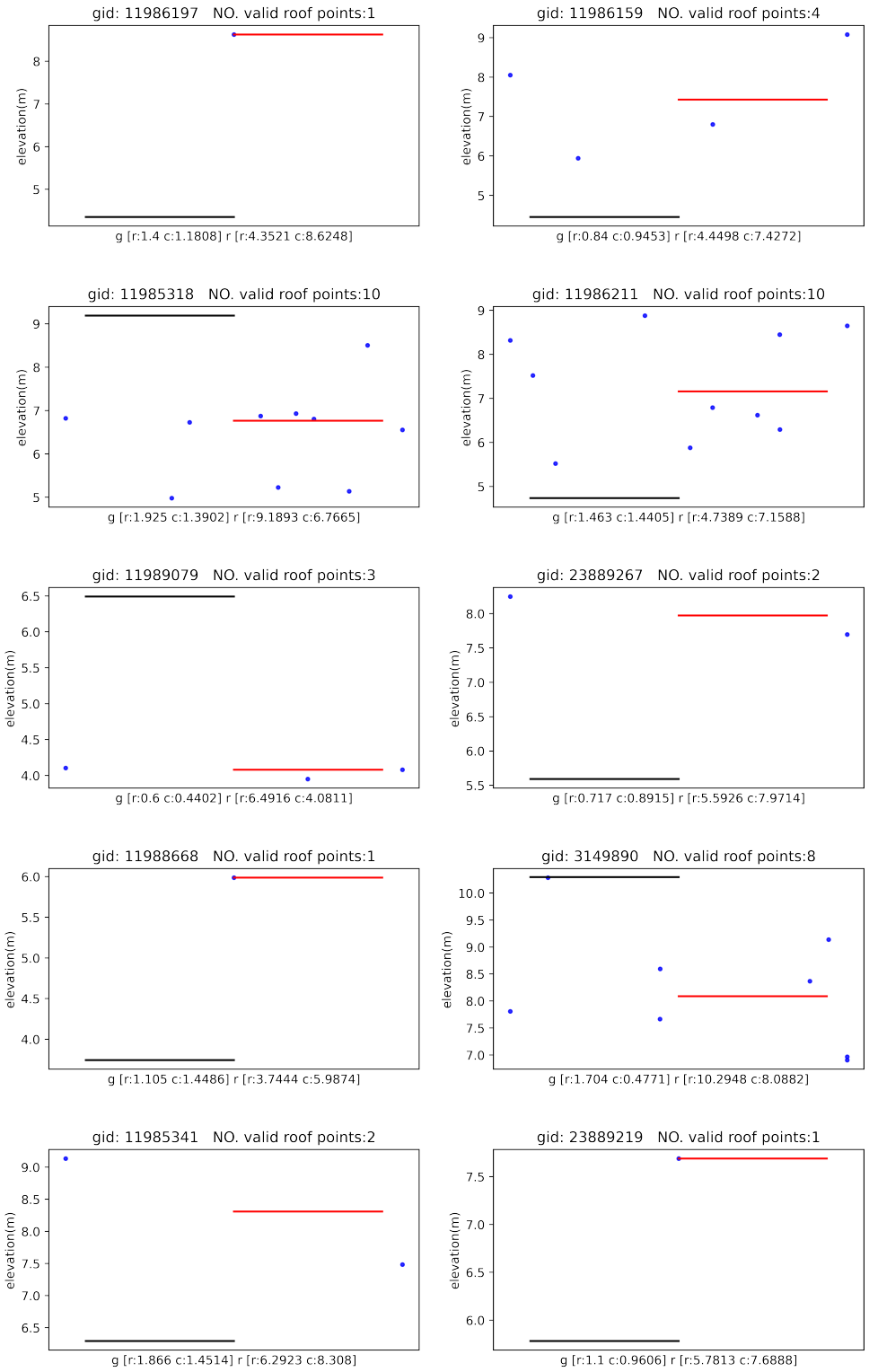


Figure 5.19.: Top ten footprint with maximum errors in Zuidbroek (Black line: reference value, red line: calculated value)

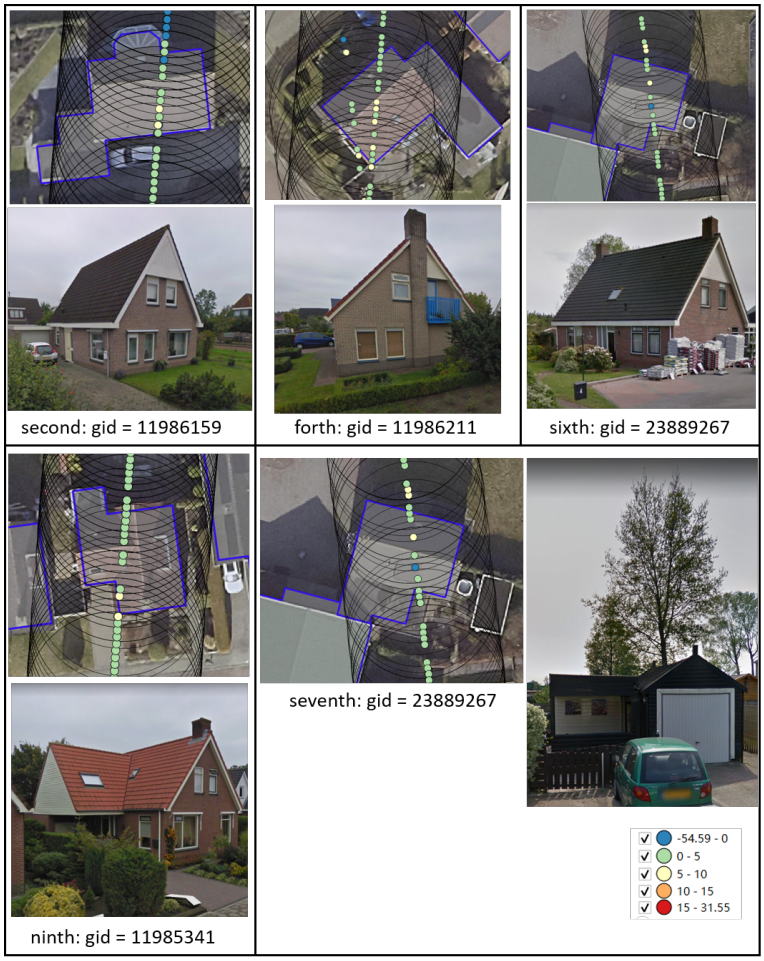


Figure 5.20.: Case study buildings (top ten footprint with maximum errors) in Zuidbroek

5.4. Model performance

The model is built using the features obtained in Section 4.4 and the adjusted hyperparameters in Section 4.5.2. And its performance is evaluated using the three evaluation methods mentioned in Section 3.5.3.

Table 5.3 shows the features in four models. Base model is with all 11 features, the other three models are generated with corresponding selection method.

5.4.1. Model accuracy

From Table 5.4, it can be seen the performance of the prediction model is poor. The maximum error is between 53 - 55m, and the R^2 score is small (0.03 - 0.05). The wrapper model has the

Table 5.3.: Feature of each model

Features \ Model	Base model	Filter model	Embedded model	Wrapper model
area	✓	✓	✓	✓
perimeter	✓	✓	-	-
construction_year	✓	✓	✓	-
length	✓	✓	✓	✓
width	✓	✓	-	-
complexity	✓	✓	-	-
vertices	✓	✓	-	-
neighbour	✓	✓	-	-
slimness	✓	✓	✓	✓
adjacent_buildings	✓	-	-	-
compactness	✓	✓	✓	✓

lowest R^2 score, 0.0297. The embedded model has the highest R^2 score, 0.0488. The difference between the results of other metrics is not much.

However, the poor performance could mainly be caused by the significant outliers from the Maastricht data set, which is explained in Section 5.3. Thus, it is interesting to see the performance of the three models without the outliers.

Table 5.4.: Model evaluation results

	MAE(m)	MAPE(%)	RMSE(m)	R^2	Max. error(m)
Base model	2.6200	45.6707	4.3610	0.0389	53.8345
Filter model	2.6041	45.1088	4.3538	0.0420	53.9006
Embedded model	2.6042	45.3805	4.3383	0.0488	54.3326
Wrapper model	2.6511	45.7930	4.3816	0.0297	54.8660

Remove outliers manually

After removing the outliers in the Maastricht data set manually, there are 2236 samples, reduced by 2 samples compared with the dataset before manual cleaning. Then, I repeated the previous step to generate new RFR models.

Table 5.5.: Model evaluation results after remove outliers

	MAE(m)	MAPE(%)	RMSE(m)	R^2	Max. error(m)
Base model	2.1305	36.6886	3.3989	0.1638	27.0906
Filter method	2.1561	36.9020	3.3967	0.1649	26.9635
Embedded method	2.2042	37.5127	3.4544	0.1363	26.9455
Wrapper method	2.2240	38.3844	3.4554	0.1358	26.9333

The accuracy of new models is shown in Table 5.5. The most obvious change is that the maximum error has been significantly reduced, from around 53 - 55m to around 27m for all

four models. MAPE, RMSE and MAE are all had different degrees of decline. R^2 score of all models are changed from 0.03 - 0.05 to 0.13 - 0.17. Based on R^2 score, the base model and filter model performed better than the embedded model and wrapper model.

Considering these metrics together, the results of the first two models (base model and filter model) and the last two models (embedded model and wrapper model) are close to each other. And the first two perform better than the last two. This explains to some extent that the model with more features (11 and 10) outperforms the model with fewer features (4 and 4) when the total number of features is 11.

Among the four methods, all models get the similar benefit from manual cleaning. The most beneficial one is the "max_error" metric, which is reduced by almost half for each model. The MAPE of each model also gets a significant change, decreasing by almost 10%. However, although the performance of the model improves after removing outliers, it still cannot be called a useful and reasonable model.

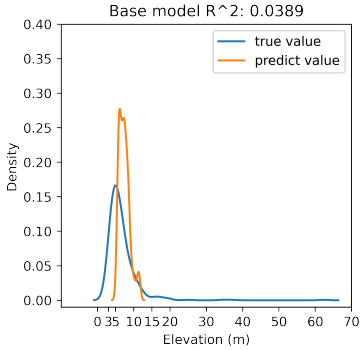
Figure 5.21 shows the density plot of the true and predicted values of three models. The predicted value is shown with the orange line and the true value in the blue line.

By comparing the plots before and after manual cleaning, a pattern can be found that all models have in common:

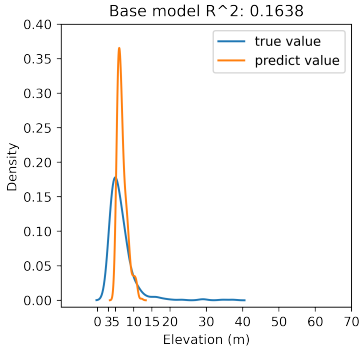
After cleaning, the maximum true value is reduced from around 70m to 40m. However, the range of predicted values does not change, which is still between around 5 - 15m. Thus, removing outliers in the Maastricht data set doesn't help improving the performance of predict model in predicting the height of tall buildings (more than 15m). Also, there is always a gap between the left end of the predicted value and the true value. This means the height of the building under five meters cannot be predicted well by all four models, whether cleaning outliers or not.

The reason for a dramatic decline between 10 - 15m can be found in Section 5.2.3. Figure 5.10 shows 90% data is between 0-10m. Therefore, there is not enough data to train the model. Lack of corresponding data in model generation results in poor performance in predicting building heights larger than 10m.

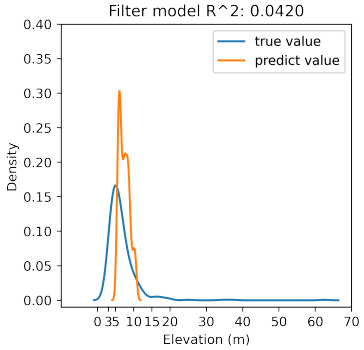
Moreover, from Figure 5.21, it seems the model cannot predict the building height lower than 5m. Although there is enough corresponding data (40%) in the data set.



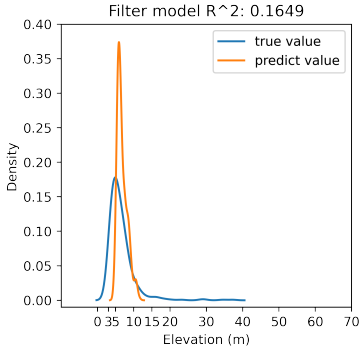
(a) Base model



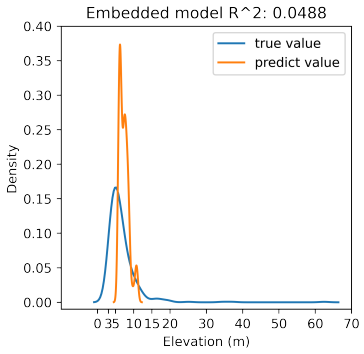
(b) Base model after manual cleaning



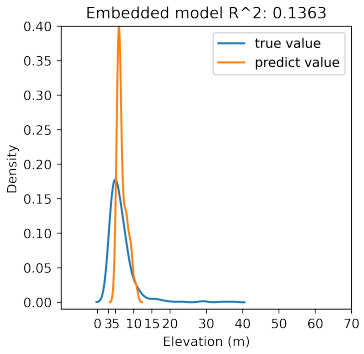
(c) Filter model



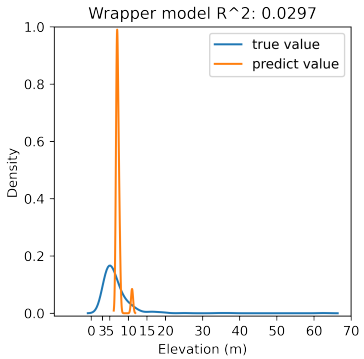
(d) Filter model after manual cleaning



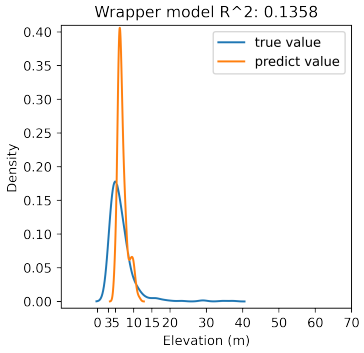
(e) Embedded model



(f) Embedded model after manual cleaning



(g) Wrapper model

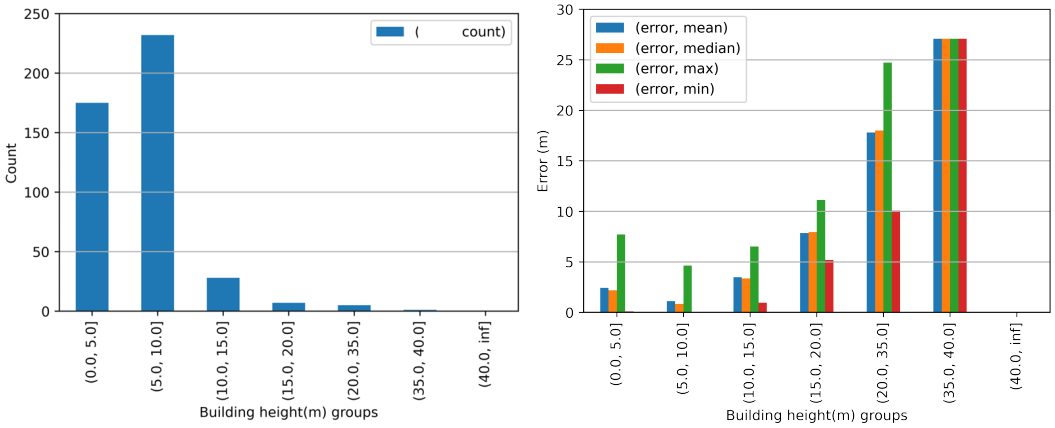


(h) Wrapper model after manual cleaning

Figure 5.21.: The density plot of reference and predict value

Figure 5.22 displays building height and the difference between the predicted value and the true value in the test data set. There are 448 data in this test data set, which is 20% of the entire data set used in model generation. The distribution of buildings in different height groups is the same as in the dataset used for model generation Figure 5.10. About 90% of the buildings are between 0-10 meters in height and only 10% of the buildings are above 10 meters Figure 5.22a. And in the test data set, there are no buildings with heights greater than forty meters.

In Figure 5.22b, the blue bar indicates the mean absolute error between true and calculated building heights of different groups. And the orange, green and red bar means median, maximum and minimum error of the difference between the true and predicted values, respectively. The error distribution pattern is the same as the pattern shown in Figure 5.13. It can be seen the building in 5-10m group has the smallest mean error (1.1267m). The 0-5m group is the next one with a mean error of 2.4243m. Then, the error gradually increases with the increase in building height. The same pattern of variation was observed for other metrics (median, maximum, and minimum error).



(a) Building height distribution

(b) Error distribution

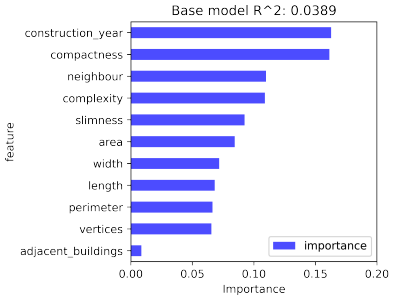
Figure 5.22.: Building height and error distribution in test data set

5.4.2. Feature contributions

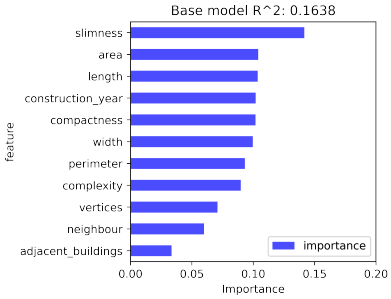
Figure 5.23 shows the importance of features of each model before and after manual cleaning.

By comparing the two bar charts, it can be seen that removing outliers or not influenced the ranking of feature importance. For the base model, the most important two features before removing outliers are "construction_year" and "compactness". While after removing, the most important one is "slimness", "area" and "length" are the next two. The "construction_year" dropped to the fourth place and the "compactness" dropped to the fifth important feature. "neighbour" is also dropped from third place to second place from the bottom of the rankings. In the filter model, the feature "neighbour" also dropped from third place to the third from the bottom of the rankings. Feature "compactness" dropped from the first place to the sixth place. Feature "perimeter" moved up from the last place to the fifth place.

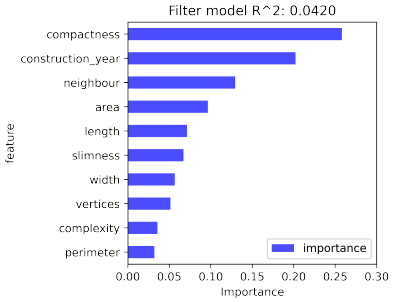
It can be observed that, before removing the outliers, the feature ”compactness” is the top important one in all models. ”construction_year” and ”neighbour” are the second important features of the base model and filter model. ”area” and ”length” are the second most important for embedded and wrapper models. After removing the outliers, the importance of features ”area” and ”length” increases. Especially the feature ”length”, become the top important feature of embedded and wrapper models.



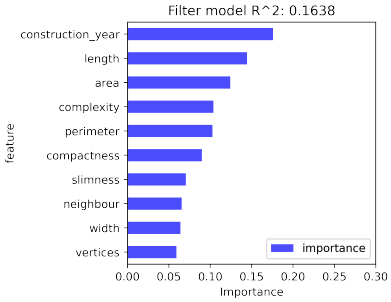
(a) Base model



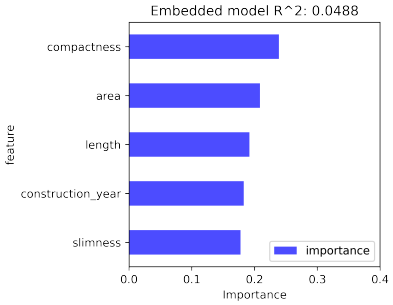
(b) Base model after manual cleaning



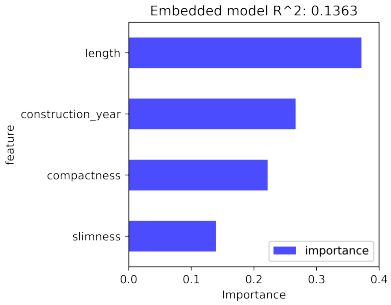
(c) Filter model



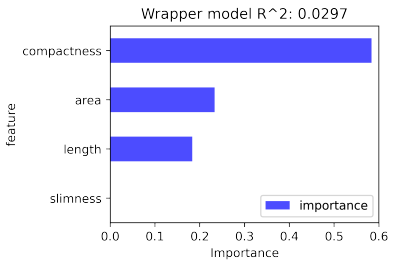
(d) Filter model after manual cleaning



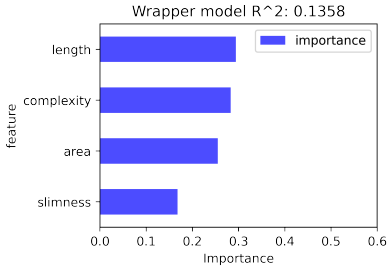
(e) Embedded model



(f) Embedded model after manual cleaning



(g) Wrapper model



(h) Wrapper model after manual cleaning

Figure 5.23.: The bar plot of feature importance

6. Discussion and conclusion

6.1. Research overview

In this section, the research question proposed in Section 1.1 are answered based on results in previous chapter.

Main research question: *Can the height of all buildings in the Netherlands be estimated from ICESat-2 data and what accuracy can be achieved?*

From the performance of RFR model that I got in Section 5.4, and the error analysis (Section 5.3), it is impossible to predict all building's height in the three selected datasets. Especially for building higher than 15m.

Thus, I would say it is impossible to get the height of all buildings in the Netherlands with ICESat-2 data. But maybe it is a feasible option for buildings between 5 and 10 meters in height.

I think the main reason is few ICESat-2 data is valid for estimating building height. A variety of factors can contribute to this:

- Sparsity and non-uniformity of the ICESat-2 data. Because of the trajectory characteristics of ICESat-2, there is always 3km between each beam. And the spatial distribution of these beams is uneven (see Figure 4.4). This results in not all footprints can be intersected by ICESat-2. Actually, only about 5% - 8% of footprints in each data set are intersected with ICESat-2 data.
- Requirements from building height estimation further reducing the amount of valid ICESat-2 data. From the three datasets used, it is known that in the original ICESat-2 data obtained at the beginning, nearly half of the ICESat-2 points with confidence less than or equal to 1 are classified as background (see Figure 5.1). These points are the first to be removed in the process of data cleaning. This results in even fewer points to be employed in building height estimation, reducing the number of intersected footprints to around 4% in each data set.

  As our goal is to estimate building height, it is necessary to require each footprint have identified roof points and ground points. However, in reality, even if one footprint is intersected with ICESat-2 data, the distribution pattern of ICESat-2 point is various (Section 3.3.1). And not all of these patterns meet the requirement (Section 3.3.3). This leads to a further decrease in the number of valid footprints. Eventually, this amount was reduced to less than 3%.
- The problem of data precision of ICESat-2 data at the footprint level. After the above data filtering steps, we got the ICESat-2 data, which is valid and can be used for building height estimation. However, the accuracy of them is still not satisfactory when used in footprint to estimate building height.

One thing is outliers. In Section 5.3, there are significant outliers appear in Maastricht for some unknown reason. This significantly affects the accuracy of the roof elevation calculation in this area, and also the performance of the final RFR model.

Another thing is the elevation of a ICESat-2 point falling in a footprint is influenced by other surrounding objects (reason 3 in Section 5.3.2). That is, even if a point falls in a footprint, is not a noise point, satisfies the estimated height requirement, and is not an outlier, it still could not provide the accurate elevation information of that footprint. This is because it could represent the elevation of other objects, not the object represented by the footprint it falls on. In fact, this is actually caused by the properties of the ICESat-2 photon itself. Each photon point from ICESat-2 has a footprint of approximately 17m in diameter. In theory, the elevation obtained by ICESat-2 point could be any object inside this diameter. Therefore, affecting the accuracy of building height estimation.

Once the available ICESat-2 data was obtained after screening, another challenge was faced in generating the prediction model:

- Lack of data for buildings over ten meters. Buildings under ten meters accounted for 90% of the total data used in model generation. This results in not enough training data for buildings over ten meters. It makes the final generated model perform poorly overall. However, it still obtained an acceptable performance in (5,10]m height group. The MAE is 1.1267m.

Next are the answers to the sub-questions:

1. *What's the percentage of building in NL are covered by ICESat-2 dataset? Is it enough to estimate all buildings with the ML method in NL?*

   Initially, around 5% - 8% of the footprints in all three datasets utilized in this study are intersected with ICESat-2 data. After filtering, around 3% of intersected footprints remain. The performance of the prediction model demonstrates that this is insufficient to build an accurate model.

2. *Which ML method should be used to do the prediction? And what attributes should be considered?*

   From literature review, the RFR method is used to do the prediction, and geometric features show in Table 2.2 are used. This selection is based on previous research. However, since the model did not perform well in general, it may be appropriate to test other models and feature combinations in future work. For example, the non-geometric features.

3. *What's the accuracy of estimated building height and model performance? Where are those errors from?*

   The performance of the prediction model is not good in general. The maximum error is between 53 - 55m, and the R^2 score is between 0.03 - 0.05. Even if remove the significant outliers manually, the R^2 score of all models are changed from 0.03 - 0.05 to 0.13 - 0.17. When analyze the error within different building height groups, it was found the building with height between 5-10m has the best performance (MAE 1.1267m).

   The sources of errors are analyzed in detail in Section 5.3. It can be summarized into four reasons: not enough valid ICESat-2 data, influence from irregular roof shape, influence from surrounding objects and effect of significant outliers.

6.2. Contribution

This thesis used BAG and ICESat-2 data build prediction ML model to estimate building height. This project's contributions are summarized as follows:

- Investigated the idea of using ICESat-2 data and footprints to calculate building height. Got the conclusion that for buildings with heights in the range of 5-10m there is an opportunity to use ICESat-2 in a country scale to obtain its height. Some properties of the icesat2 data (sparsity, non-uniformity, possession of a 17 m footprint) were verified, and their influence on predicting building heights was investigated and studied.
- In the computation of ground elevation, many combinations of interpolation algorithms and grid sizes were investigated. Different percentiles are also evaluated in the estimation of roof elevation. These experimental results can aid in understanding of the ICESat-2 data and give information for future investigations.
- Built the RFR model with only geometric features to predict building height. The model was found to be suitable for buildings ranging in height from 5 to 10 meters.

6.3. Future Work

Although the method proposed in this project can partially provide a estimate of the building height, there are a number of limitations outlined as follows:

- **Data cleaning.** The existing method cannot remove some outliers, such as the outliers in the Maastricht data set.
- **Unexplained phenomena.** There are also some unexplained phenomena: Why do significant outliers only appear in the Maastricht? Why does using the same interpolation method perform much less well in the Maastricht dataset than in the other two?
- **Feature selection.** Only use geometric features to build prediction model. And the amount of features is only 11.
- **Model training.** The dataset used to generate the model is not representative. Only 10% of the data for heights greater than 10 meters.

Based on the limitations, several recommendations are expected as extensions for the future work.

For the outliers, we hope them can be detected and removed by the method in the future. The model performance needs to be improved, increase the amount of available training data can alleviate this problem. In the future, a mixture of ICESat-2 and GEDI data can be tried to reduce the impact of the sparsity of the data itself. In this project, three data sets were used to analyzed the feasibility of ICESat-2 in building height estimation area. The result shows that building higher than 10m do not have enough training data. Thus, could try to increase the number of data sets used in the future. It is also possible to try other machine learning methods, and feature combinations, such as cadastral and statistical (census) data, to see if these methods can get better results.

A. Reproducibility self-assessment

A.1. Marks for each of the criteria

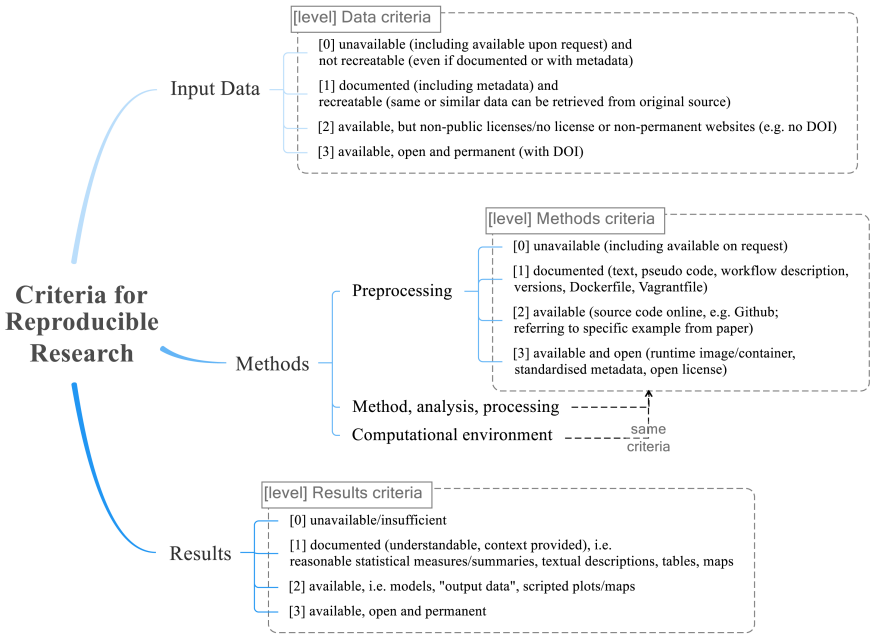


Figure A.1.: Reproducibility criteria to be assessed.

Grade/evaluate yourself for the 5 criteria (giving 0/1/2/3 for each):

Table A.1.: Evaluation of the five criteria

Criteria	Score	Comments
Input data	3	ICESat-2 and BAG data are avaliable, open and permanent.
Preprocessing	2	Source code is available on GitHub.
Method, analysis, processing	2	Source code is available on GitHub.
Computational environment	3	Source code is available on GitHub.
Results	2	Models, ”output data”, scripted plots/data are available.

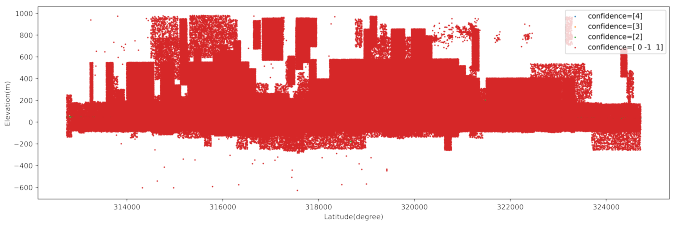
A.2. Self-reflection

The data used in this research are all open. All source codes are available in GitHub, including preprocess, methods, and analysis. The computational environment used in this research is also open-sourced.

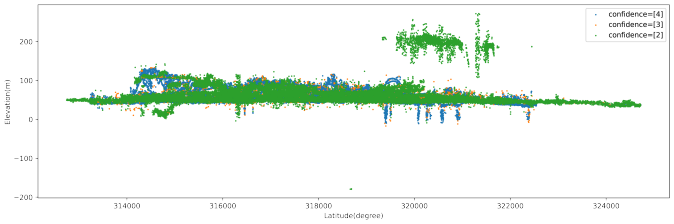
B. Additional Results

B.1. Filter Results

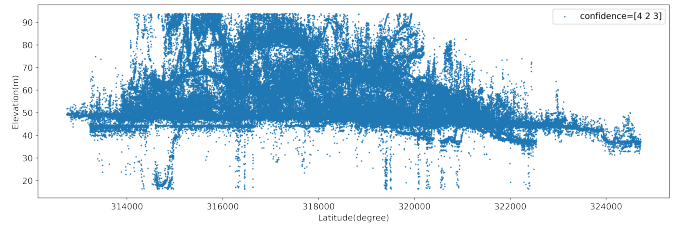
These plots display the changes in the three data sets during the data cleaning step using latitude as the x-axis and elevation as the z-axis. The original data of three regions, after the confidence filter, and the final result after the boxplot filter are shown respectively.



(a) Original data

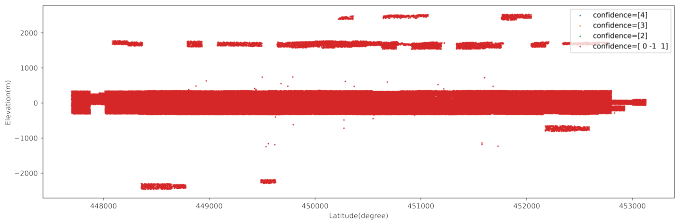


(b) After confidence filter

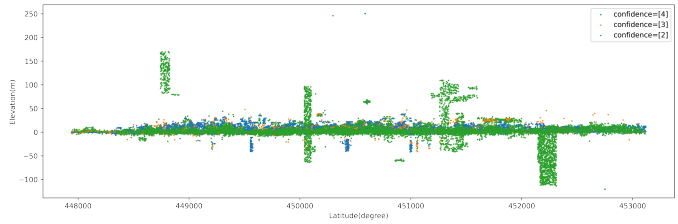


(c) After boxplot filter

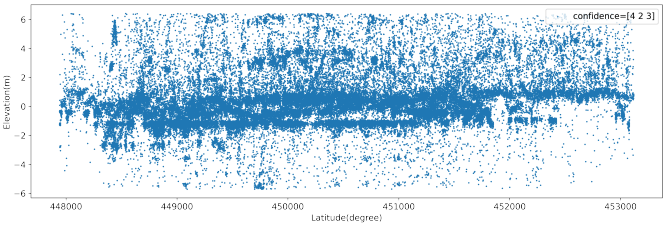
Figure B.1.: Filter results in Maastricht



(a) Original data



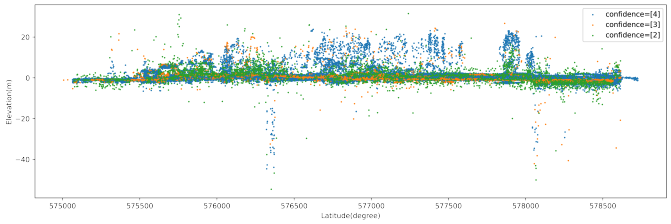
(b) After confidence filter



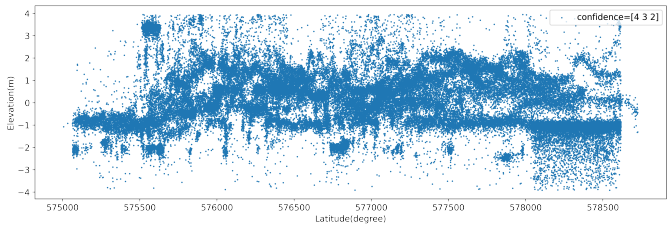
(c) After boxplot filter

Figure B.2.: Filter results in Rijswijk

(a) Original data



(b) After confidence filter

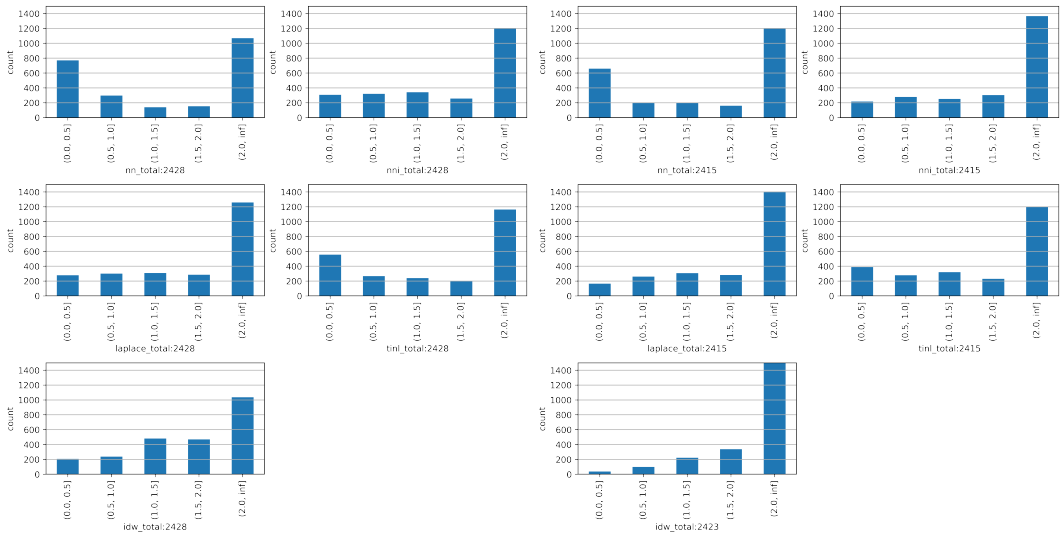


(c) After boxplot filter

Figure B.3.: Filter results in Zuidbroek

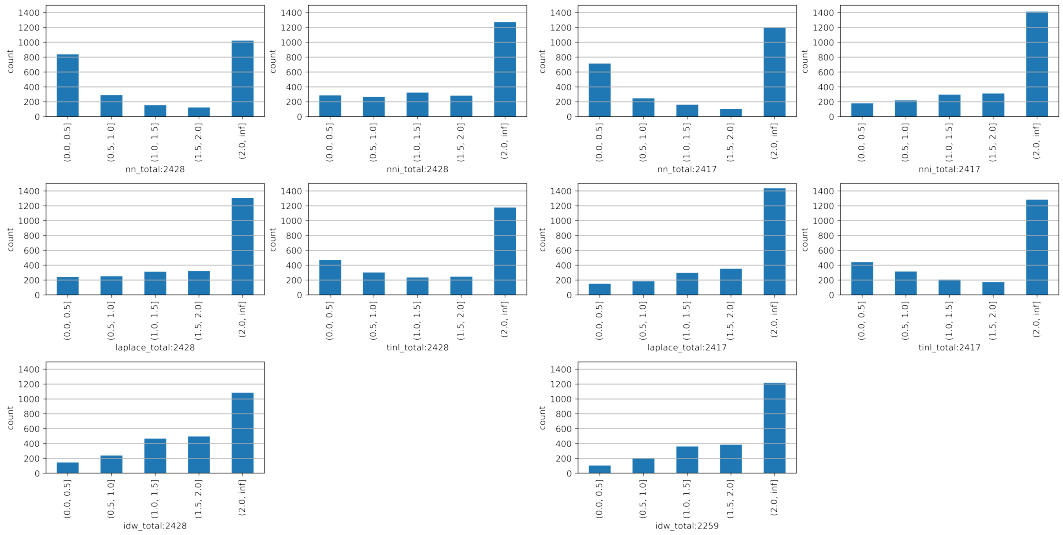
B.2. Interpolation results

Method 1 is left one in Figure 3.6. Method 2 is right one in Figure 3.6.



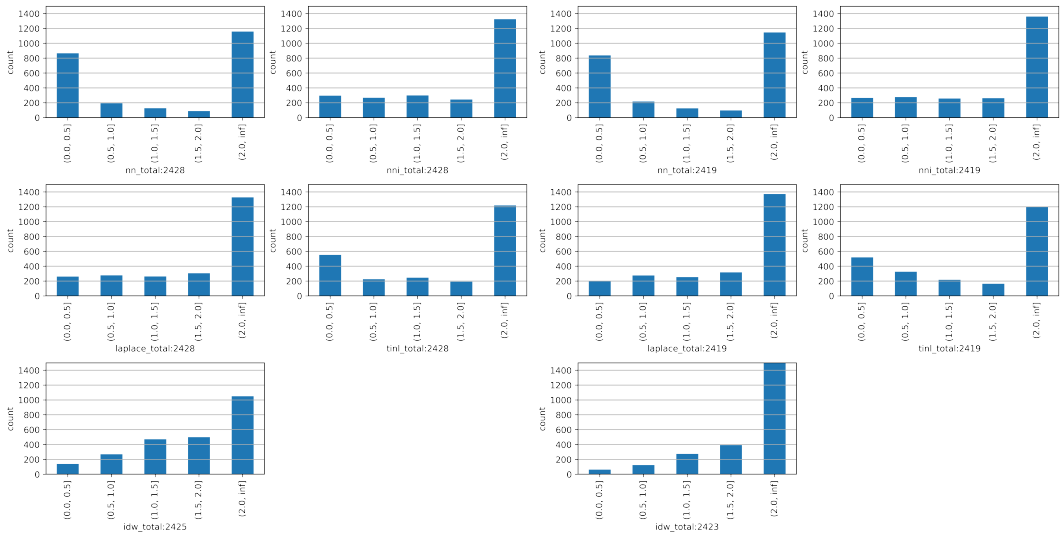
(a) Grid size 150m (method1)

(b) Grid size 150m (method2)



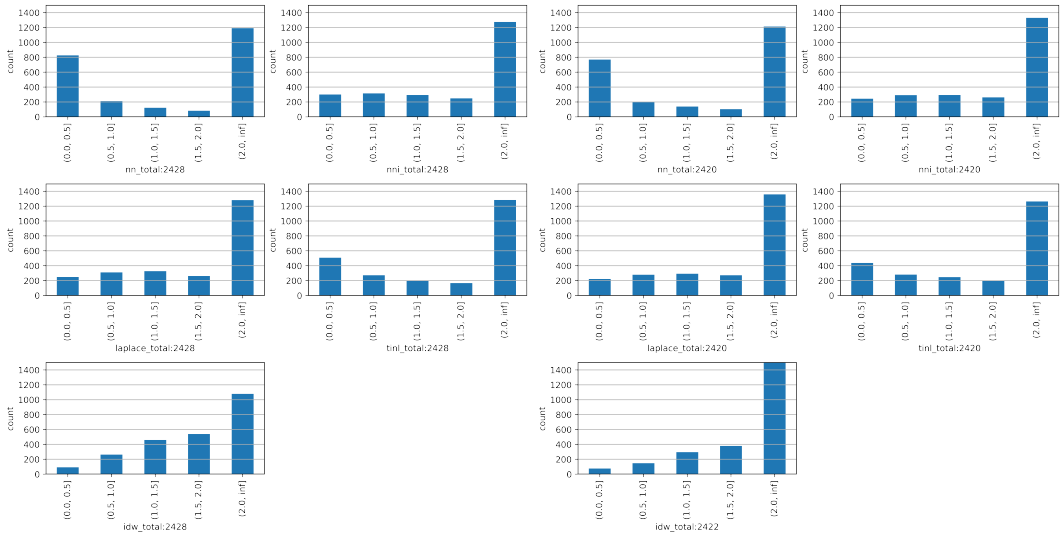
(c) Grid size 100m (method1)

(d) Grid size 100m (method2)



(e) Grid size 50m (method1)

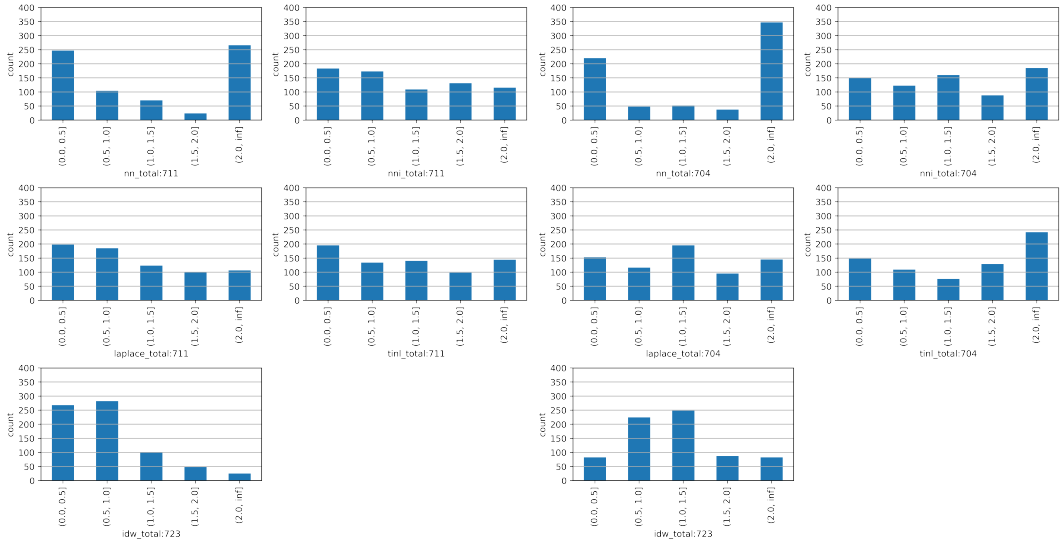
(f) Grid size 50m (method2)



(g) Grid size 25m (method1)

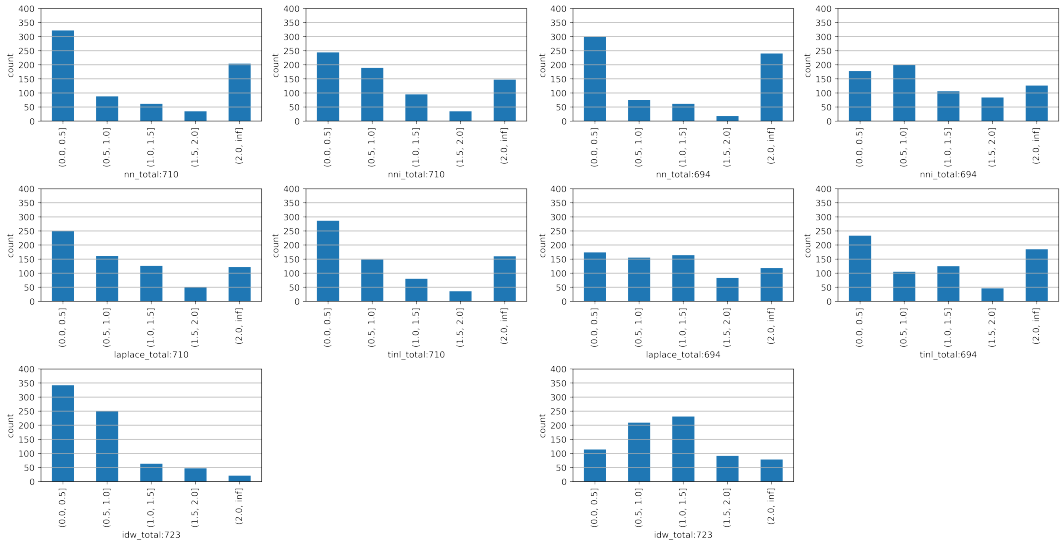
(h) Grid size 25m (method2)

Figure B.4.: Distribution of ground elevation errors in Maastricht



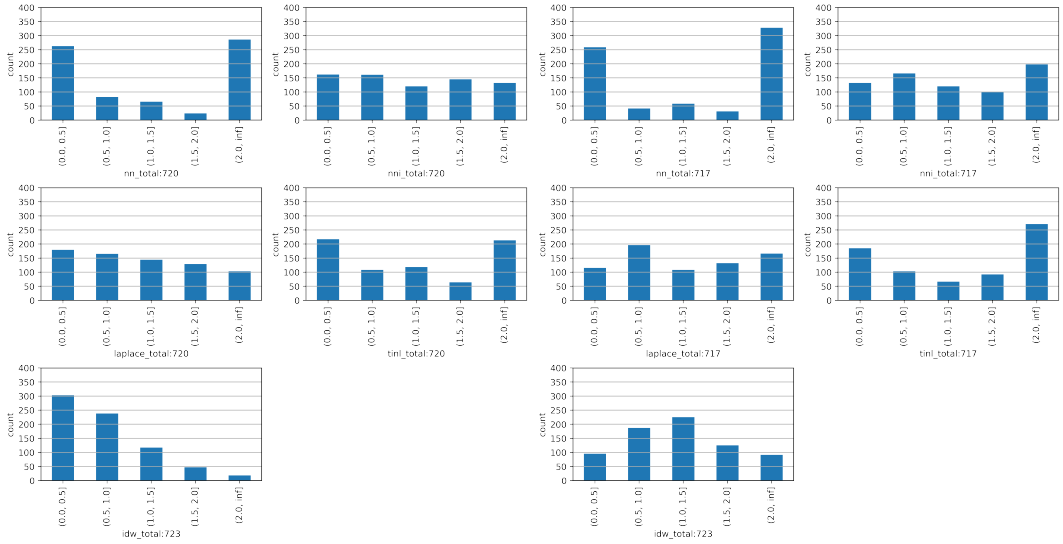
(a) Grid size 150m (method1)

(b) Grid size 150m (method2)



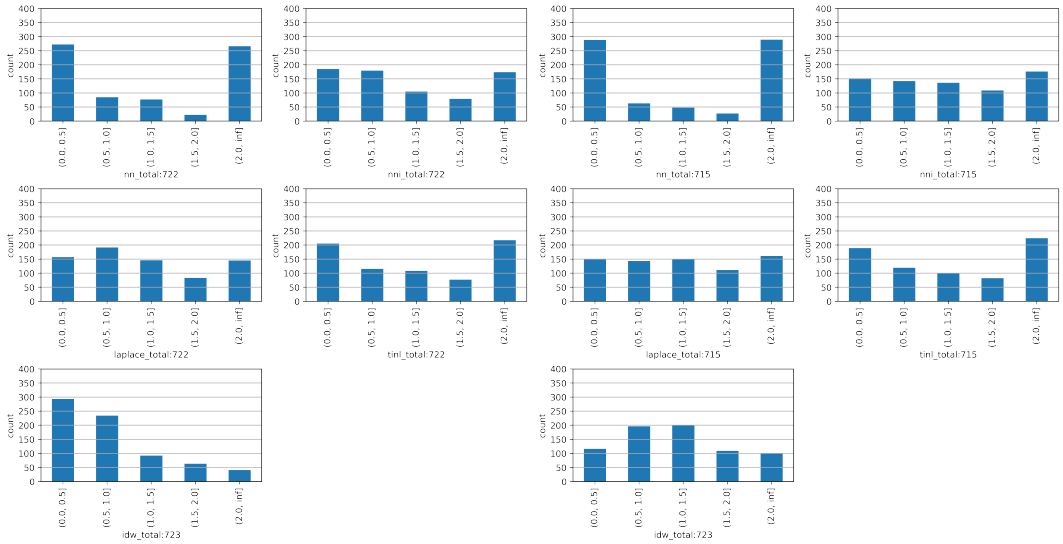
(c) Grid size 100m (method1)

(d) Grid size 100m (method2)



(e) Grid size 50m (method1)

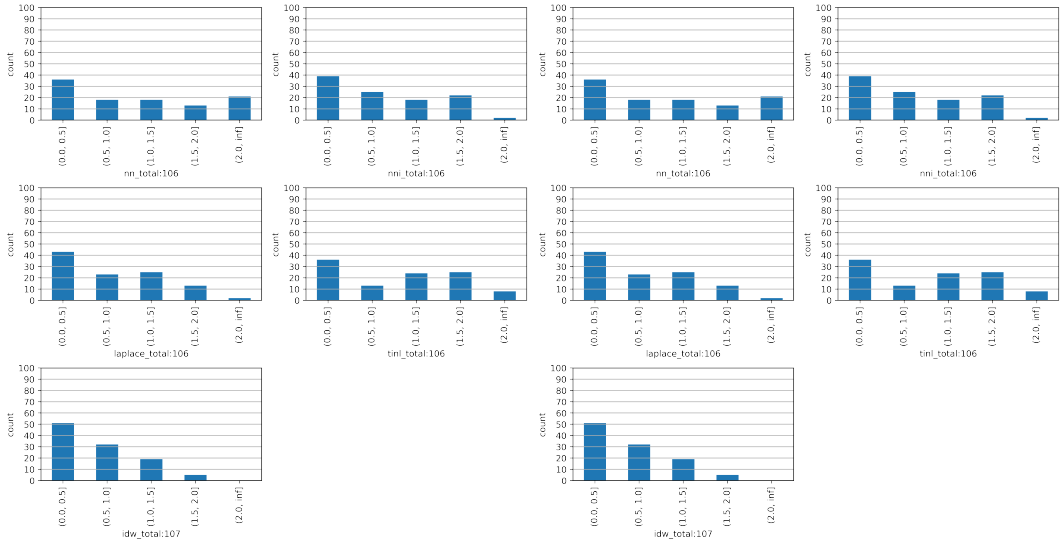
(f) Grid size 50m (method2)



(g) Grid size 25m (method1)

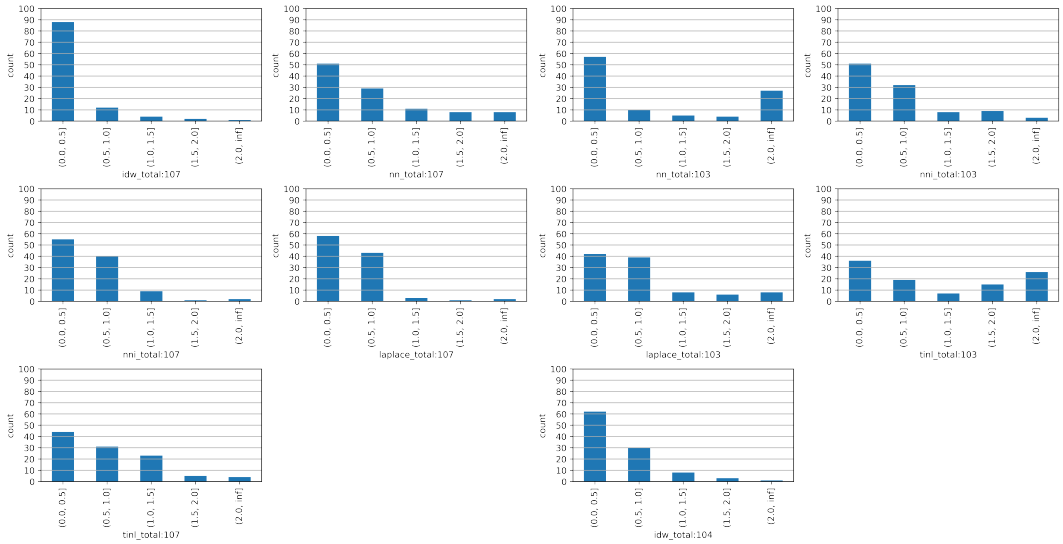
(h) Grid size 25m (method2)

Figure B.5.: Distribution of ground elevation errors in Rijswijk



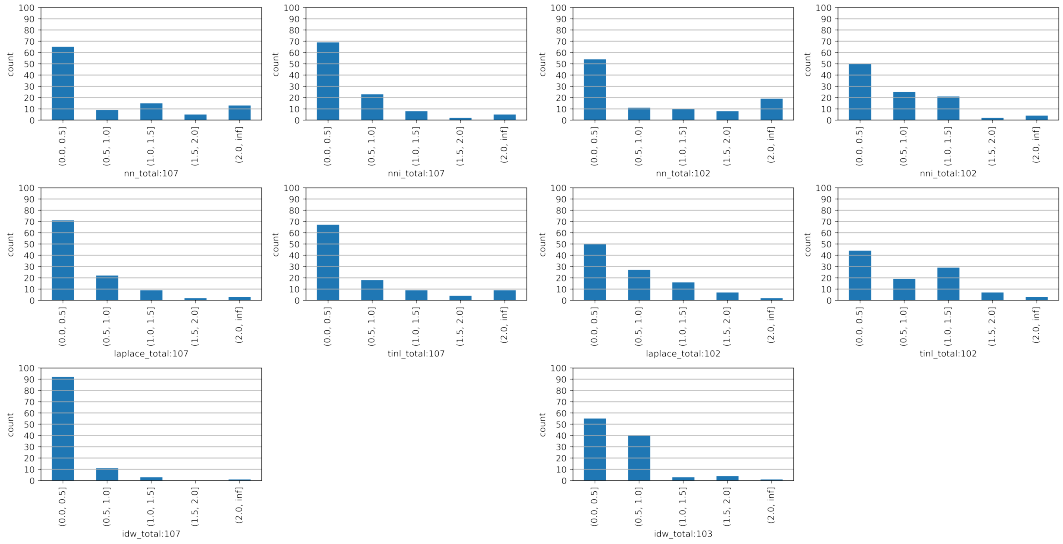
(a) Grid size 200m (method1)

(b) Grid size 200m (method2)



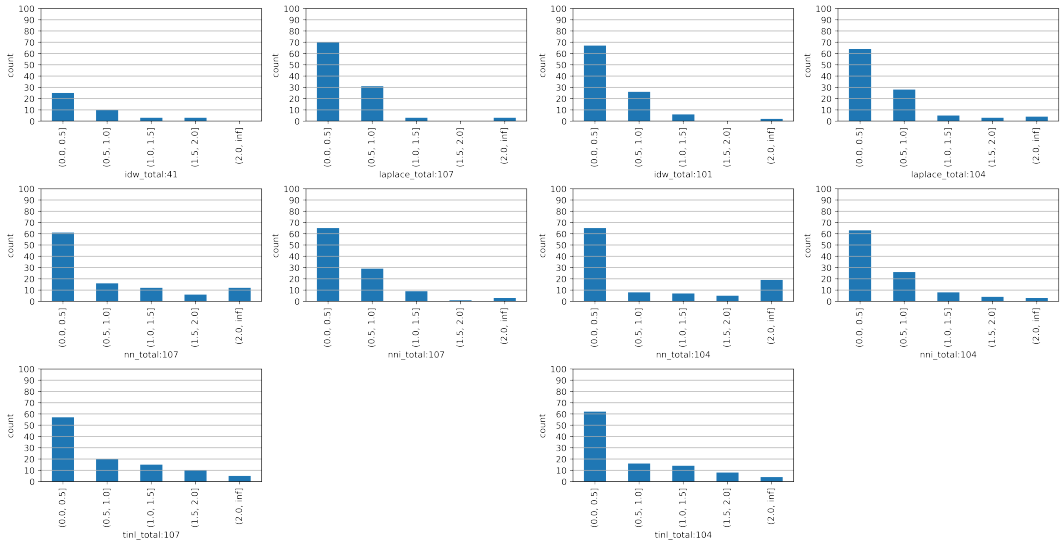
(c) Grid size 150m (method1)

(d) Grid size 150m (method2)



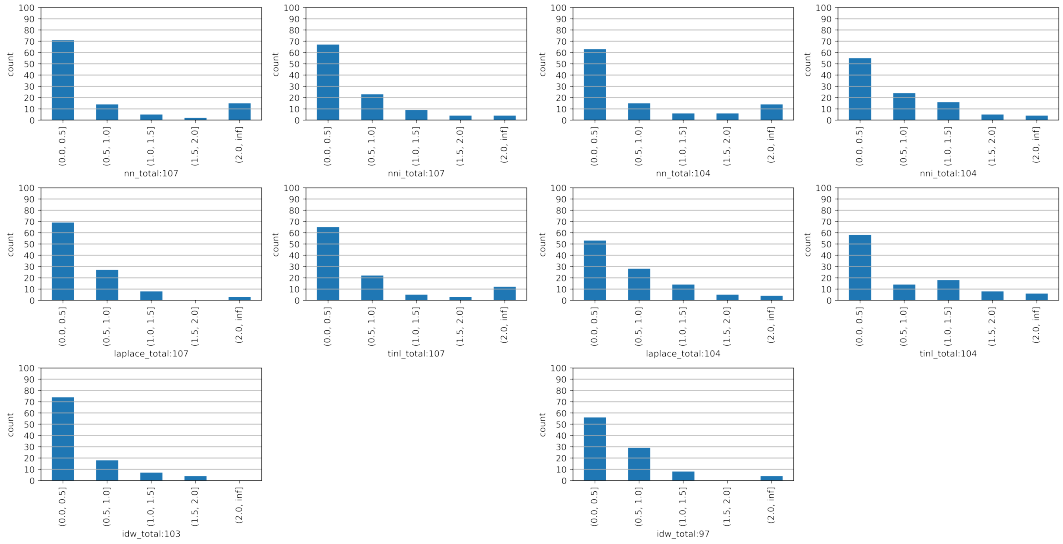
(e) Grid size 100m (method1)

(f) Grid size 100m (method2)



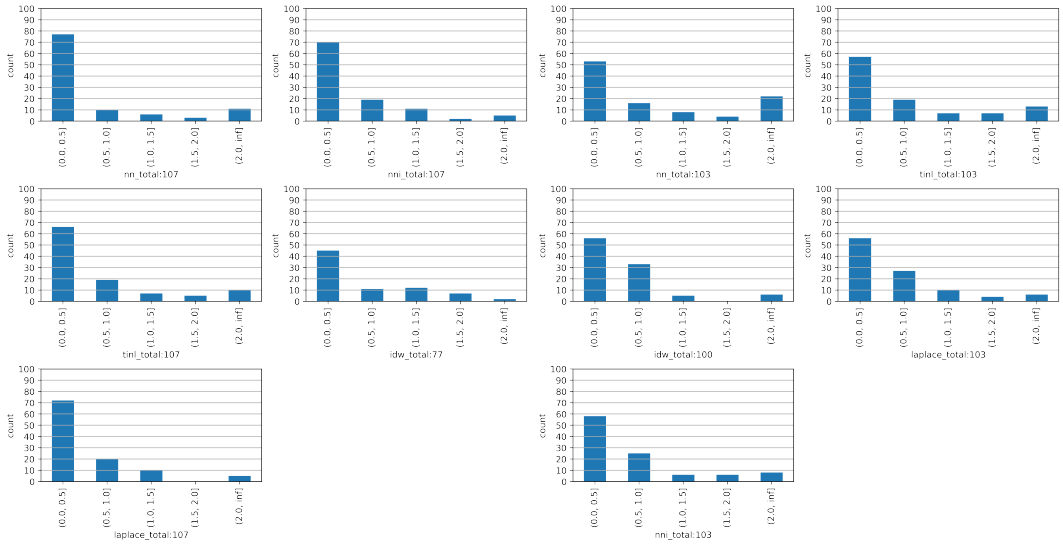
(g) Grid size 50m (method1)

(h) Grid size 50m (method2)



(i) Grid size 25m (method1)

(j) Grid size 25m (method2)

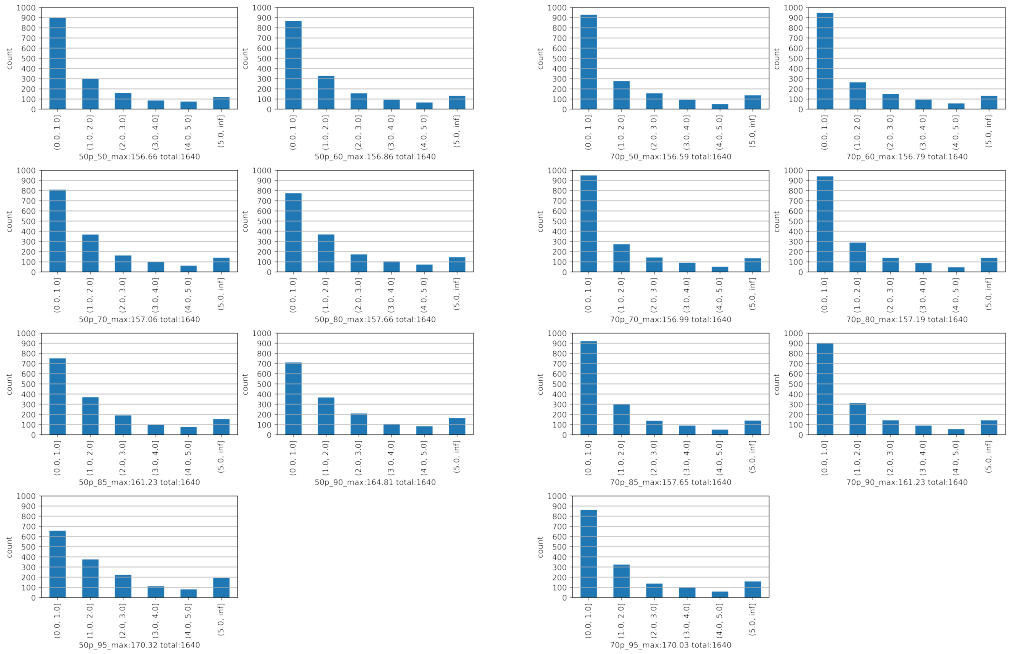


(k) Grid size 10m (method1)

(l) Grid size 10m (method2)

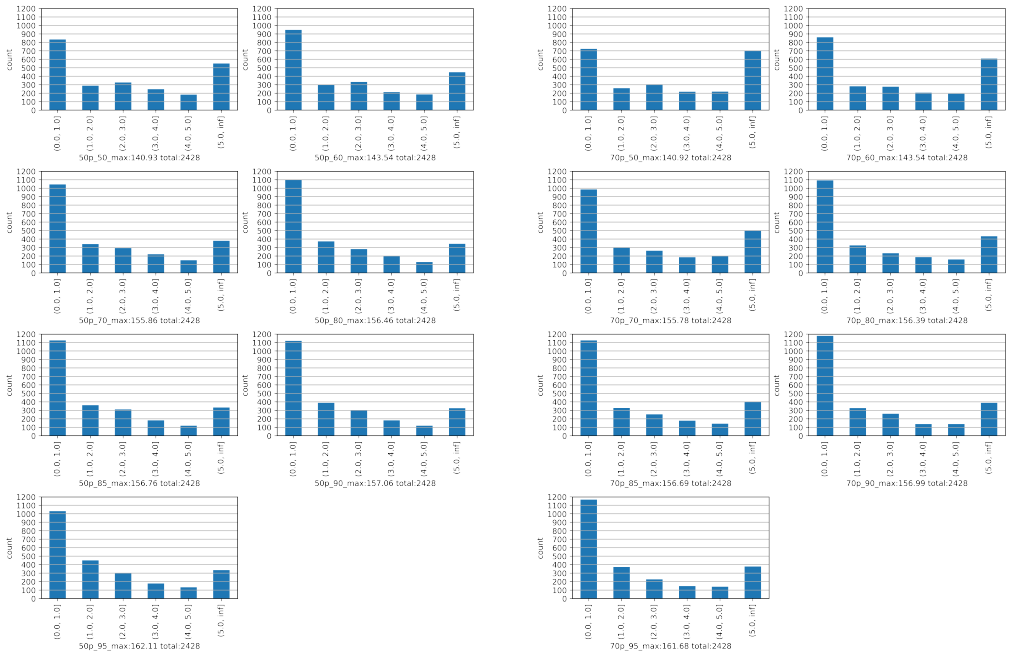
Figure B.6.: Distribution of ground elevation errors in Zuidbroek

B.3. Roof elevation errors with different percentiles



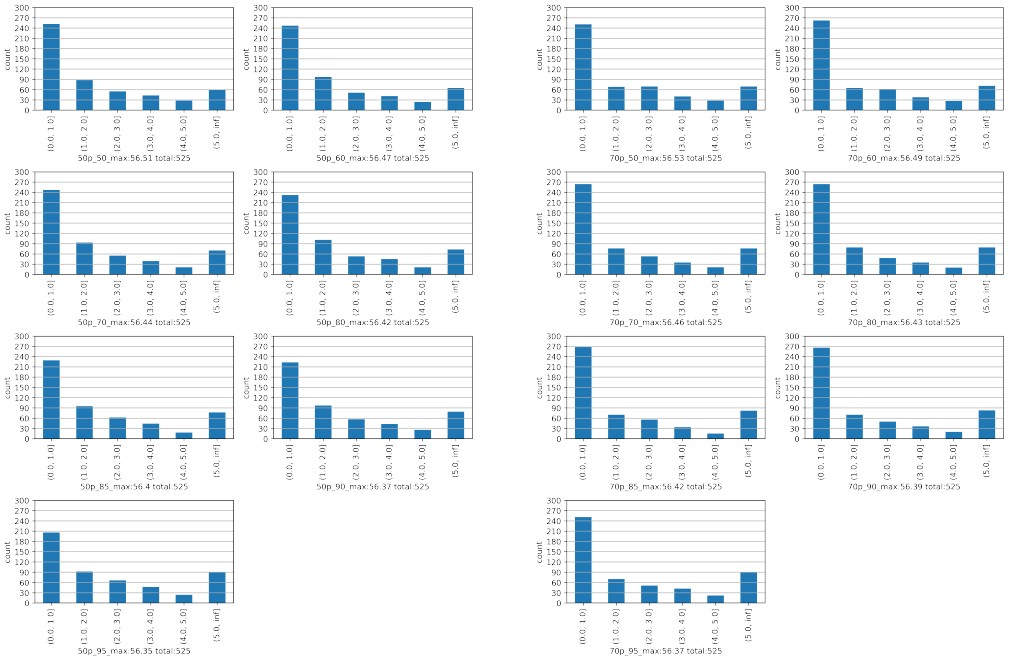
(a) Compared with h_dak_50p (b) Compared with h_dak_70p

Figure B.7.: Errors in Maastricht with valid roof points filter



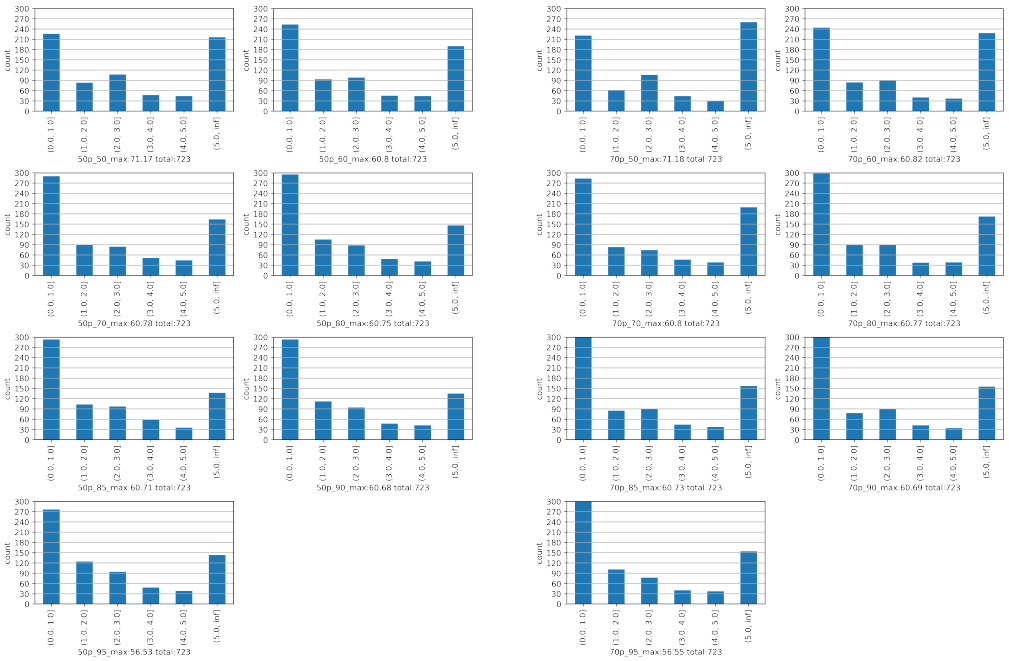
(a) Compared with h_dak_50p (b) Compared with h_dak_70p

Figure B.8.: Errors in Maastricht without valid roof points filter



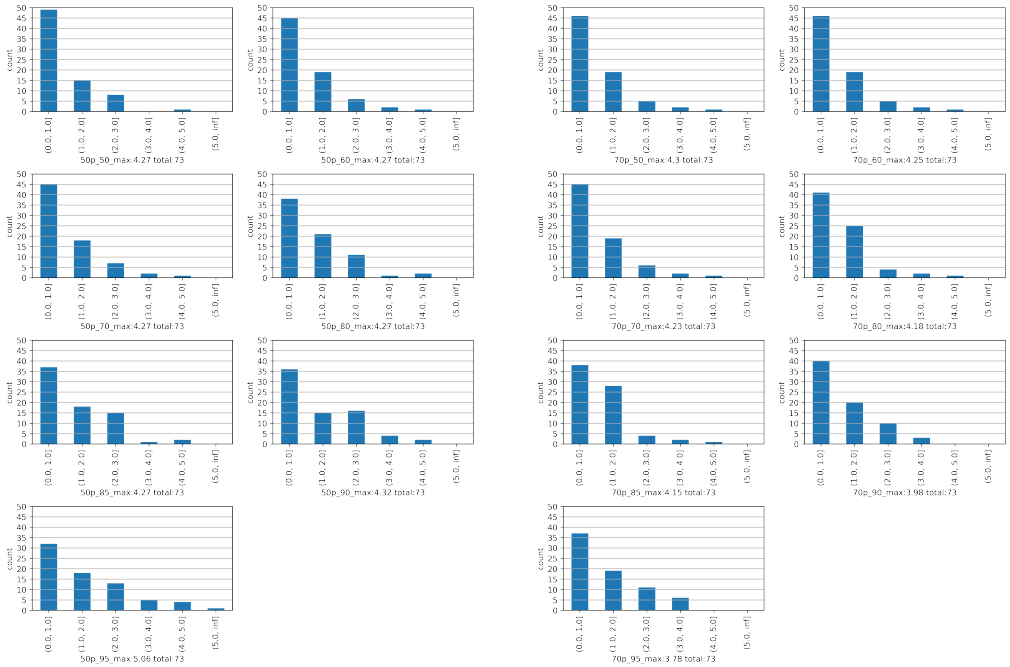
(a) Compared with h_dak_50p (b) Compared with h_dak_70p

Figure B.9.: Errors in Rijswijk with valid roof points filter



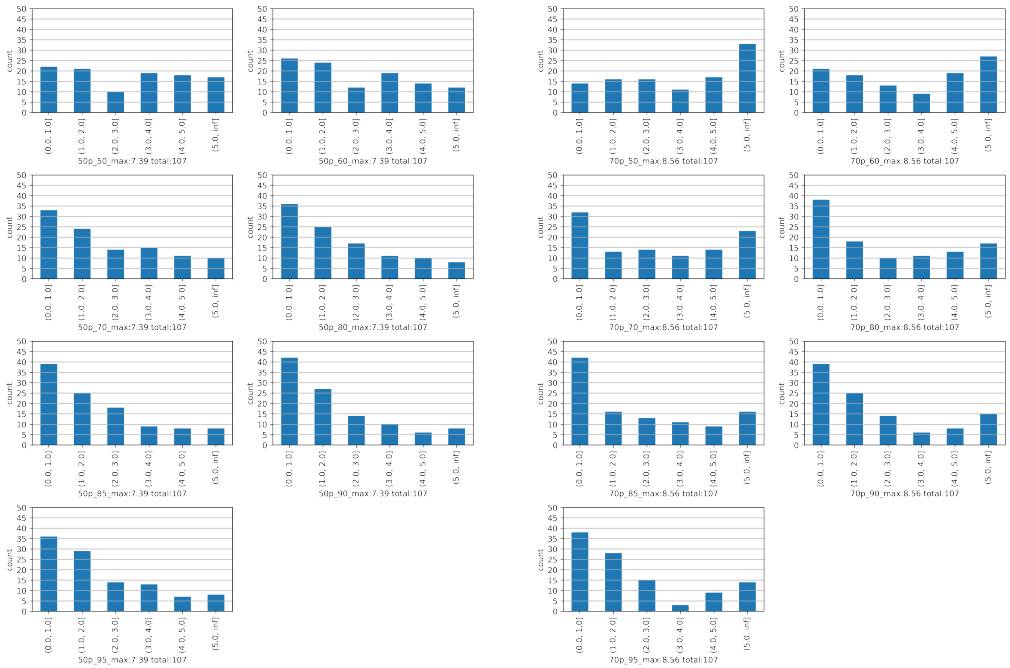
(a) Compared with h_dak_50p (b) Compared with h_dak_70p

Figure B.10.: Errors in Rijswijk without valid roof points filter



(a) Compared with h_dak_50p (b) Compared with h_dak_70p

Figure B.11.: Errors in Zuidbroek with valid roof points filter



(a) Compared with h_dak_50p (b) Compared with h_dak_70p

Figure B.12.: Errors in Zuidbroek without valid roof points filter

Bibliography

Anh, P., Thanh, N. T. N., Vu, C. T., Ha, N. V., and Hung, B. Q. (2018). Preliminary Result of 3D City Modelling For Hanoi, Vietnam. In *2018 5th NAFOSTED Conference on Information and Computer Science (NICS)*, pages 294–299.

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks*, 5(4):537–550. Conference Name: IEEE Transactions on Neural Networks.

Bergstra, J. and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(10):281–305.

Biljecki, F., Ledoux, H., and Stoter, J. (2014). Height references of CityGML LOD1 buildings and their influence on applications. *undefined*.

Biljecki, F., Ledoux, H., and Stoter, J. (2016). An improved LOD specification for 3D building models. *Computers, Environment and Urban Systems*, 59:25–37.

Biljecki, F., Ledoux, H., and Stoter, J. (2017). Generating 3D city models without elevation data. *Computers, Environment and Urban Systems*, 64:1–18.

Blum, A. L. and Langley, P. (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1):245–271.

Chai, T. and Draxler, R. R. (2014). Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. *Geoscientific Model Development*, 7(3):1247–1250. Publisher: Copernicus GmbH.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Chicco, D., Warrens, M. J., and Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, 7:e623.

Dandabathula, G., Sitiraju, S. R., and Jha, C. S. (2021). Retrieval of building heights from ICESat-2 photon data and evaluation with field measurements. *Environmental research*, 1(1):011003. Publisher: IOP Publishing.

Dubayah, R., Blair, J. B., Goetz, S., Fatoyinbo, L., Hansen, M., Healey, S., Hofton, M., Hurtt, G., Kellner, J., Luthcke, S., Armston, J., Tang, H., Duncanson, L., Hancock, S., Jantz, P., Marselis, S., Patterson, P. L., Qi, W., and Silva, C. (2020). The Global Ecosystem Dynamics Investigation: High-resolution laser ranging of the Earth's forests and topography. *Science of Remote Sensing*, 1:100002.

Dukai, B., Ledoux, H., and Stoter, J. E. (2019). A MULTI-HEIGHT LOD1 MODEL OF ALL BUILDINGS IN THE NETHERLANDS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W8:51–57.

Forthofer, R. N., Lee, E. S., and Hernandez, M. (2007). 13 - Linear Regression. In Forthofer, R. N., Lee, E. S., and Hernandez, M., editors, *Biostatistics (Second Edition)*, pages 349–386. Academic Press, San Diego.

Frantz, D., Schug, F., Okujeni, A., Navacchi, C., Wagner, W., van der Linden, S., and Hostert, P. (2021). National-scale mapping of building height using Sentinel-1 and Sentinel-2 time series. *Remote Sensing of Environment*, 252:112128.

Galarnyk, M. (2020). Understanding Boxplots.

Garvin, M. (2013). *An Introduction to Statistical Learning, chapter 3: Linear Regression, pages 59–128.* Springer Science+Business Media New York 2013.

Hancock, S., McGrath, C., Lowe, C., Davenport, I., and Woodhouse, I. (2021). Requirements for a global lidar system: spaceborne lidar with wall-to-wall coverage. *Royal Society Open Science*, 8(12):211166. Publisher: Royal Society.

Jović, A., Brkić, K., and Bogunović, N. (2015). A review of feature selection methods with applications. In *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1200–1205.

Kokalj, v. and Mast, J. (2021). Space lidar for archaeology? Reanalyzing GEDI data for detection of ancient Maya buildings. *Journal of Archaeological Science: Reports*, 36:102811.

Lao, J., Wang, C., Zhu, X., Xi, X., Nie, S., Wang, J., Cheng, F., and Zhou, G. (2021). Retrieving building height in urban areas using ICESat-2 photon-counting LiDAR data. *International Journal of Applied Earth Observation and Geoinformation*, 104:102596.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2018). Feature Selection: A Data Perspective. *ACM Computing Surveys*, 50(6):pp1–45.

Li, X., Zhou, Y., Gong, P., Seto, K. C., and Clinton, N. (2020). Developing a method to estimate building height from Sentinel-1 data. *Remote Sensing of Environment*, 240:111705.

Liu, A., Cheng, X., and Chen, Z. (2021). Performance evaluation of GEDI and ICESat-2 laser altimeter data for terrain and canopy height retrievals. *Remote Sensing of Environment*, 264:112571.

Lánský, I. (2020). Height Inference for all USA Building Footprints in the Absence of Height Data. *Delft University of Technology*.

Miche, Y., Bas, P., Lendasse, A., Jutten, C., and Simula, O. (2007). Advantages of Using Feature Selection Techniques on Steganalysis Schemes. In Sandoval, F., Prieto, A., Cabestany, J., and Graña, M., editors, *Computational and Ambient Intelligence*, Lecture Notes in Computer Science, pages 606–613, Berlin, Heidelberg. Springer.

Milojevic-Dupont, N., Hans, N., Kaack, L. H., Zumwald, M., Andrieux, F., Soares, D. d. B., Lohrey, S., Pichler, P.-P., and Creutzig, F. (2020). Learning from urban form to predict building heights. *PLOS ONE*, 15(12):e0242010. Publisher: Public Library of Science.

Narine, L. L., Popescu, S., Neuenschwander, A., Zhou, T., Srinivasan, S., and Harbeck, K. (2019). Estimating aboveground biomass and forest canopy cover with simulated ICESat-2 data. *Remote Sensing of Environment*, 224:1–11.

Neuenschwander, A. and Pitts, K. (2019). The ATL08 land and vegetation product for the ICESat-2 Mission. *Remote Sensing of Environment*, 221:247–259.

Neumann, T. A., Martino, A. J., Markus, T., Bae, S., Bock, M. R., Brenner, A. C., Brunt, K. M., Cavanaugh, J., Fernandes, S. T., Hancock, D. W., Harbeck, K., Lee, J., Kurtz, N. T., Luers, P. J., Luthcke, S. B., Magruder, L., Pennington, T. A., Ramos-Izquierdo, L., Rebold, T., Skoog, J., and Thomas, T. C. (2019). The Ice, Cloud, and Land Elevation Satellite – 2 mission: A global geolocated photon product derived from the Advanced Topographic Laser Altimeter System. *Remote Sensing of Environment*, 233:111325.

Pham-Gia, T. and Choulakian, V. (2014). Distribution of the Sample Correlation Matrix and Applications. *Open Journal of Statistics*, 4(5):330–344. Number: 5 Publisher: Scientific Research Publishing.

Pronk, M. (2022). SpaceLiDAR.jl.

Pronk, M., Ledoux, H., and Eleveld, M. (2022). Comparing and combining ICESat-2 and GEDI spaceborne LiDAR for reconstructing terrain. .

Ranjan, G. S. K., Kumar Verma, A., and Radhika, S. (2019). K-Nearest Neighbors and Grid Search CV Based Real Time Fault Monitoring System for Industries. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–5.

Roy, E. (2022). Inferring the number of floors of building footprints in the Netherlands. *Delft University of Technology*.

Sammut, C. and Webb, G. I. (2010). Mean Absolute Error. In *Encyclopedia of Machine Learning*, pages 652–652. Springer US, Boston, MA.

Sampath, A. and Shan, J. (2010). Segmentation and Reconstruction of Polyhedral Building Roofs From Aerial Lidar Point Clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 48(3):1554–1567. Conference Name: IEEE Transactions on Geoscience and Remote Sensing.

Swamidass, P. M. (2000). MAPE (mean absolute percentage error)MEAN ABSOLUTE PERCENTAGE ERROR (MAPE). In Swamidass, P. M., editor, *Encyclopedia of Production and Manufacturing Management*, pages 462–462. Springer US, Boston, MA.

Sánchez-Maroño, N., Alonso-Betanzos, A., and Tombilla-Sanromán, M. (2007). Filter Methods for Feature Selection – A Comparative Study. In Yin, H., Tino, P., Corchado, E., Byrne, W., and Yao, X., editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2007*, Lecture Notes in Computer Science, pages 178–187, Berlin, Heidelberg. Springer.

Wendel, J., Murshed, S., Sriramulu, A., and Nichersu, A. (2016). Development of a web-browser based interface for 3D data—A case study of a plug-in free approach for visualizing energy modelling results. *Lecture Notes in Geoinformation and Cartography*, pages 185–205. ISBN: 9783319196015.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.