MSc thesis in Geomatics

Automatic water detection using ICESat-2 measurements

H.B. Rotteveel

June 2026

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:	Ir. Maarten Pronk
	Dr. Hugo Ledoux
Co-reader:	Dr. Rajashree Tri Datta

Abstract

Accurate data on inland water surface elevations are becoming increasingly important as climate change intensifies extreme weather events worldwide. Classic data collection methods on water are often expensive and both spatially and temporally limited, making water data largely concentrated in the global north. Ice, Cloud and Land Elevation Satellite 2 (ICESat-2) and its advanced Light Detection And Ranging (LiDAR) instrument are, however, well-suited for collecting data as it is capable of measuring elevation with centimeter-level accuracy across the planet. Its existing inland water data products, ATL13 and ATL22 are, however, constrained by their reliance on water masks that exclude lakes and reservoirs smaller than 0.1 km^2 and rivers narrower than 50 m.

The goal of this thesis is therefore to create a Random Forest (RF) model using ICESat-2 data that can predict the presence of inland water, specifically for bodies smaller than 25 m. A set of window-based features that encode the interaction between photons and water was derived at multiple window radii. These features try to quantify and characterize the presence of afterpulses, bottom reflectances, a low slope, a low distribution of photon elevation, and a high photon density. The features were identified by first de-correlating all features using Ward's linkage clustering and then selecting the best using a Mean Decrease in Impurity (MDI) and permutation importance score. Photon density proved to be the best predicting feature, contributing 41.2% of the total mean decrease in impurity in the model.

The final random forest model used a 2.5 m window and was trained on 7 million points from water segments smaller than $\leq$ 25 m and an equal number of land points in the Netherlands. The model was evaluated on approximately 520 million ICESat-2 photons across the country and achieved a recall above 80.0% for water segments longer than 6 m and up to 87.1% for water bodies between 10 m and 25 m. Potential improvements of using the features from multiple windows or selecting only windows with a minimum number of photons present proved to be ineffective in gaining better results.

The main sources of misclassifications are currently the presence of snow, uncertainty near water edges, and incorrect ground truth data. Manually validating results in the Swiss Alps, Greenland, and a Mexican mangrove forest showed promising results. The RF performs reasonably well outside its training environment, suggesting that the model identified general photon-water interactions rather than region-specific characteristics. Future work should focus on expanding the training dataset to include geographically diverse data to improve performance and combining the classified photons using clustering to create line segments of water surface elevation.

Acknowledgements

I would like to express my gratitude towards Maarten and Hugo, who made for relaxed supervisors that I could both brainstorm and laugh with. The same goes for Yair and Luc, who made every lecture of the MSc worth going to. I also want to thank Anneke for all her support and encouragement; she helped me with so much and inspired me to continue my academic journey with a PhD. Lastly, of course, to my girlfriend and family, who actively listened (or at least pretended to) while I was working through complex problems: I hope you actually read the thesis and find that it was worth it.

Heiko Rotteveel
Delft, June 2025

Contents

List of Figures

List of Tables

Acronyms

1. Introduction

Monitoring of inland water is becoming more important as climate change intensifies extreme weather events worldwide. Properly managing inland freshwater reserves requires high-quality data for decision-making, which is often unavailable due to the high costs of data collection (Papa et al., 2023). This limits the availability of water data both temporally and spatially. Satellite Earth Observation is frequently used for measuring water bodies, as it provides both a large spatial extent and a decent temporal resolution. Radio Detection And Ranging (RADAR) satellites are often relied on for water detection, due to the clear return signal when radio waves interact with water.

However, as these satellite missions were primarily designed to observe ocean surface topography, they are not always able to observe inland water due to the smaller water body sizes and limitations in the surrounding topography, like high mountains (Biancamaria et al., 2018). Although Synthetic Aparture RADAR (SAR) has a long historical record, up until the more recent missions, SAR satellites also had large spatial resolutions (30 m), which excluded smaller water bodies due to the difficulty of delineating them in an image. Recent studies have therefore started using the LiDAR instrument aboard the Ice, Cloud and Land Elevation Satellite 2 (ICESat-2) as an alternative. ICESat-2 provides a high-resolution footprint (Magruder et al., 2020) that is (partially) capable of making observations through vegetation. NASA provides the ATL13 Inland Surface Water and the ATL22 Mean Inland Surface Water products to give users easy access to inland water levels, but both products are limited by the water masks they rely on. The datasets only contain data on lakes and reservoirs larger than 0.1 km^2 and rivers wider than 50 m (Jasinski et al., 2025b). Data on smaller water bodies is thus currently still unavailable in most parts of the world.

This thesis, therefore, aims to create a model that can identify photons that reflected off water, specifically photons from water bodies smaller than 25 m in width. It approaches

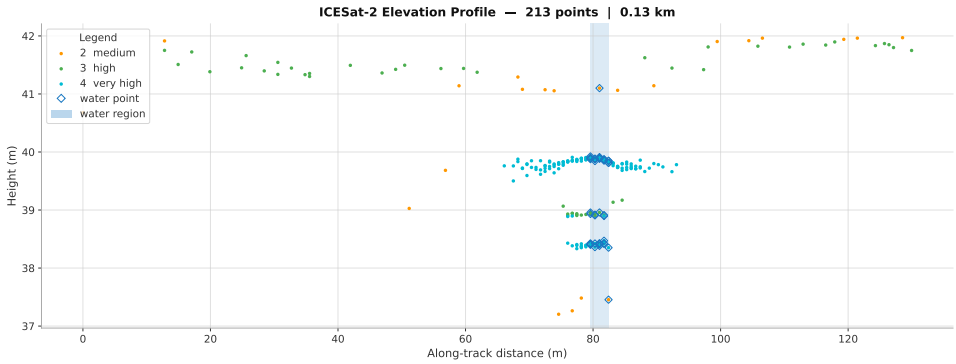


Figure 1.1.: Example of ICESat-2 obeservations over a 3 m wide water body (ATL03). Water is highly reflective, making the signal visible beyond the true edges

water detection differently than the ICESat-2 data products. Instead of asking *"Is this photon inside a known water polygon?"*, it asks *"Does this photon look like it bounced off water?"*. The 25 m width was chosen because it is the resolution at which most satellite-based methods begin to struggle to accurately delineate water body edges. For optical imagery, Radoux et al. (2016) noted that small water bodies are only reliably separable in Sentinel-2 imagery when their diameter exceeds 11 m, while 20 m was noted as the limit for SAR data from Sentinel-1/2 (Schmitt, 2020). As a global water mask resolution of 30 m is the highest resolution available (Li et al., 2013; Pekel et al., 2016), the focus was put halfway on 25 m; the edge of the reliable detection limit. The main research question was formulated as follows:

To what extent can ICESat-2 geo-referenced photon data be used to automatically detect inland water bodies smaller than 25 *m in width?*

To identify water bodies smaller than 25 m, a Random Forest (RF) machine learning model has been trained on the low-level ATL03 data product from ICESat-2, which detects patterns in the LiDAR data that encode the presence of water. Previous studies (Ma et al., 2024b; Datta and Wouters, 2021; Neuenschwander and Magruder, 2019; Neumann et al., 2025a) have already found a set of unique photon patterns when ICESat-2 observes water, but no study has assessed or combined these different features and used them specifically on smaller water bodies. The first subquestion, therefore, was:

What photon-water interaction features from ICESat-2 ATL03 data are most informative for distinguishing water from land at small spatial scales?

The model has been trained on data from the Netherlands observed over a 7-year timespan. The Netherlands was chosen due to its publicly available, high-quality water mask, which includes data even for small channels and ditches. The second subquestion was formulated as:

To what extent can a Random Forest classifier, trained on Dutch water body data, accurately detect water bodies smaller than 25 *m?*

To assess whether the model has only identified Dutch-specific water characteristics, the model was also evaluated in different geographical areas around the world in the third research question:

To what extent does the model generalize beyond its training environment (the Netherlands) to geographically diverse regions?

The rest of this thesis is structured around the scientific paper presented in Chapter 3. Though this chapter can be read as a standalone piece of work, and all the relevant information is present, a more extensive theoretical background is provided in Chapter 2. It is recommended to read Chapter 2 first, as it provides a general introduction to the different methods and technologies used in the main article. Chapter 2 starts with an introduction of the ICESat-2, how it works, and what data products are available from it. Section 2.2 will dive into Random Forest Classification, explaining how decision trees work and how combining many of them in a Random Forest (RF) can improve performance. Lastly, Section 2.3 explains how to assess the performance of a RF. The thesis will conclude in Chapter 4 with a closing discussion that expands further on the one presented in the main scientific paper, adding more specific recommendations for future work.

2. Theoretical background and related work

This chapter aims to lay the theoretical foundation needed to understand the scientific paper presented in Chapter 3. Background information is provided on the ICESat-2 mission to provide context on how the data used in this research was collected. Section 2.2 discusses RF Classification, a machine-learning technique that will be used to create a prediction model. Finally, in Section 2.3, the metrics to evaluate machine-learning performance will be addressed to understand how the final presented model will be evaluated.

2.1. ICESat-2

2.1.1. The ATLAS instrument

The ICESat-2 mission of NASA launched in 2018 to measure ice sheet elevation, sea ice thickness, land topography, vegetation characteristics, and clouds. It does so using a single photon-counting laser altimeter called the Advanced Topographic Laser Altimeter System (ATLAS). ATLAS measures the travel time of laser pulses to calculate the distance between the satellite and the surface of the Earth (Neumann et al., 2019). The laser has a wavelength of 532 nanometers, resulting in a bright green light that can penetrate water. With 10,000 pulses fired per second, ATLAS is capable of taking measurements every 2.3 feet ($\approx$ 70 cm) (Magruder et al., 2020). With each pulse, around 300 trillion photons are sent through a series of lenses and mirrors, which split the laser into six beams (Neumann et al., 2019). These six beams are organized as three pairs of two beams, one weak and one strong, separated by 90 m, where each pair in turn is separated by approximately 3.3 km in the cross-track direction (see Figure 2.1). The footprint of each laser has a diameter of 10.9 ± 1.3 m, meaning that photons can be returned to the instrument from within a circle of around 10.9 meters around the track (Magruder et al., 2020).

Only a few dozen of the photons sent out actually return to the satellite. These photons are sent through a series of filters which only let light through at precisely 532 nanometers (Neumann et al., 2019). This is to prevent sunlight reflected from the Earth from overwhelming the detectors. When a photon makes it through the filters, it falls on one of six fiber optic cables, corresponding to one of the six send-out beams. When this happens, a timer, which started when the laser left the satellite, stops (Neumann et al., 2019). This timing information for each photon is then used together with the satellite position and speed of light to determine the distance the photon has traveled. One of these data points is, however, not sufficient to determine elevation, since ATLAS also picks up a lot of background photons coming from sunlight, clouds, and particulates in the air, which could skew the ground data. That is why photon or point clouds are created, showing thousands of data points

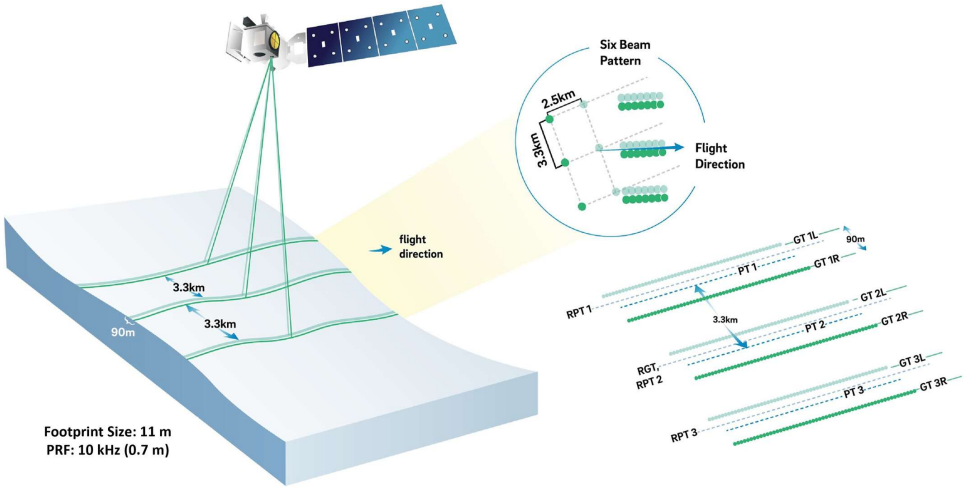


Figure 2.1.: The ATLAS instrument onboard the ICESat-2 platform obtains data using green, photon-counting LiDAR that is split into six beams. Adapted from Neumann et al. (2025c)

that returned to the instrument. Only by applying further processing and filtering can the elevation be determined (Neumann et al., 2025b).

2.1.2. The ICESat-2 data products

The data from ATLAS and data collected from ancillary systems are telemetered to the ground and processed into several data products (see Figure 2.2) (Neumann et al., 2025b). Multiple pre-processing steps are taken to process the raw data, but these are outside the scope of this thesis. The ATL03 Global Geolocated Photons data product forms the foundation for higher-level data products. It contains information per beam on all the photons that have been measured and their estimated geolocation (longitude, latitude, and elevation) with an accuracy of 3.5 m ± 2.1 m (Magruder et al., 2020). The product also provides a label for each photon representing the likelihood of the photon being a signal photon, i.e. it being a photon that actually represents a point on the surface. These points are classified as `Noise`, `Low`, `Medium`, `High`, `Very High`. Other ancillary data is also present in the product, but less relevant to this work.

The higher-level products (level 3+) use the ATL03 product as a basis to work from. An example is ATL13, which creates along-track water surface heights and descriptive statistics for inland water bodies (Jasinski et al., 2025b). The model uses ATL03 photon data and combines both physical and statistical modeling of physical processes related to open water surface dynamics and light propagation to determine surface water height statistics and

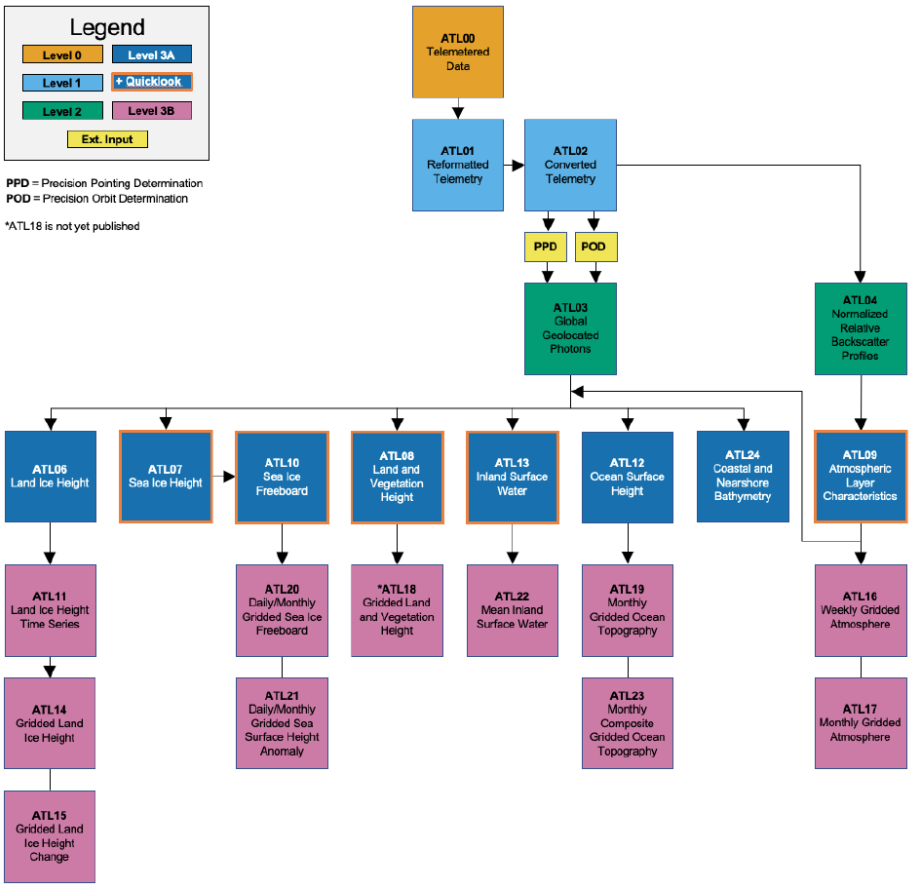


Figure 2.2.: Schematic of ICESat-2 data processing and data products. Adapted from Neumann et al. (2025c)

.

related parameters. An example of what the resulting data might look like is shown in Figure 2.3. Though the primary data source for the ATL13 products is ATL03, external sources are also used, including meteorological data like wind speed and water body datasets.

There are also many other data products coming from the ICESat-2's mission of measuring ice sheet elevation, sea ice thickness, land topography, vegetation characteristics, and clouds (see Figure 2.2). All of these products process the Geolocated Photons using their own algorithms to identify relevant characteristics.

2.2. Random Forest Classification

2.2.1. Decision trees

A Decision tree is a predictive modeling tool that is commonly used to classify data into a number of classes (Breiman, 2001). It does so by asking questions about features of the data

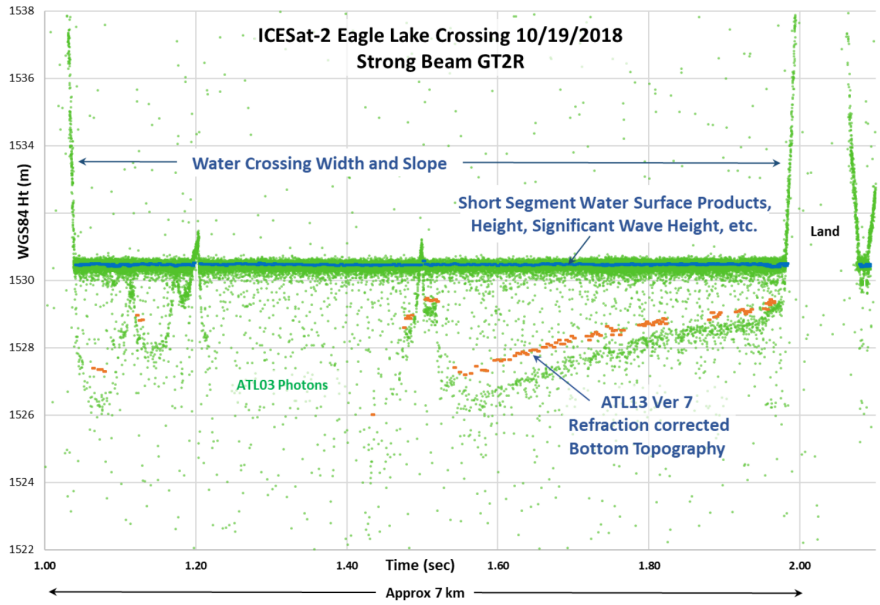


Figure 2.3.: Example of ATL13 Ver 07 short segment surface water and bathymetry products for ICESat-2 beam GT2R crossing over Eagle Lake, CA on October 19, 2018. Adapted from Jasinski et al. (2025b)

.

and splitting the data depending on the results. A decision tree is built up of nodes and branches. There are three types of nodes: the root node, the split node, and the leaf node. The root node is the origin of the decision tree. A split node represents a test on a feature of the data. Each of the outcomes of this test is represented by a branch. A leaf node represents the final class label or a final decision made. The pathways from the root to the leaves can be seen as `if-then` rules (Song and Lu, 2015).

Figure 2.4 provides a simple example of a decision tree. Depending on the answers given at each node, the result (or branch) will lead to the next node until arriving at a leaf node, at which point the data is classified. All nodes have precisely one incoming branch; loops are thus not possible (Criminisi et al., 2012).

Feature selection

Though it is possible to select the parameters and split the tree by hand for simple datasets, this gets increasingly more difficult when the data becomes more complex and the number of features required to make a proper classification increases. To help make a decision on which features to pick, we look at the Information Gain.

Information Gain explains how useful a feature (or question) is for splitting the data into groups. It does so by measuring how much uncertainty decreases after a split. A good question will create clearer groups and thus leave the least amount of uncertainty. Entropy is one approach to measure uncertainty. It can be calculated at a node N by

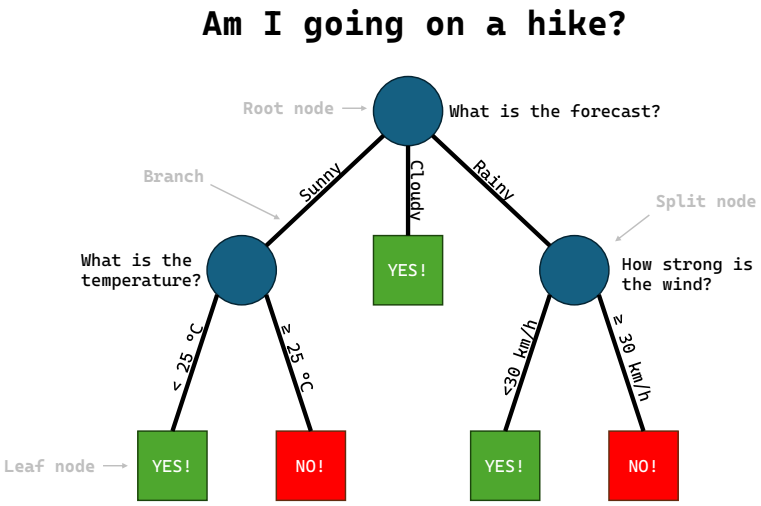


Figure 2.4.: Example of decision tree. Variant of example by Quinlan (1986)

$$Entropy(N) = -\sum_{i=1}^{k} p_i \cdot log_2(p_i), \tag{2.1}$$

where p_i represents the proportion of class i in the node, and k represents the total number of classes (Quinlan, 1986). Zero entropy corresponds to a perfectly pure split. Higher entropy, on the other hand, indicates greater disorder among class labels. By comparing the entropy of the child nodes to the parent node, it is possible to see if the feature (or question) helps to better differentiate between classes. The total information gain is thus the difference between a parent node's entropy and the weighted sum of its child node entropies. The information gain of an attribute A relative to a collection of data S is defined as

$$Gain(S, A) = Entropy(p) - \sum_{v \in Values(A)} \frac{[S_v]}{[S]} Entropy(S_v), \tag{2.2}$$

where $Values(A)$ is all the possibles values for attribute A, and S_v is the subset of S for which attribute A has value v.

A popular alternative to Entropy is the Gini Index. The Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified. A lower value is thus preferred. The Gini Index can be calculated by

$$Gini = 1 - \sum_{i=1}^{k} p_i^2. \tag{2.3}$$

As can be observed from the formula in Equation 2.3, the values of the Gini Index lie within the interval of $[0, 0.5]$. The maximum possible value of 0.5 corresponds to the highest impurity of a node, while a value of 0 corresponds to a node containing only elements of the same class. The information gain of the Gini Index is calculated similarly to that of entropy by

$$Gain(S, A) = Gini(p) - \sum_{v \in Values(A)} \frac{[S_v]}{[S]} Gini(S_v). \tag{2.4}$$

Computationally, entropy is more complex because of its use of logarithms. The Gini Index is thus the faster option, and the metric that is most commonly used in practice.

Stopping rules

A natural question that now arises is when to decide to stop splitting a node and declare it as a leaf. An extreme solution would be to stop only when all the leaf nodes are 100% pure. This would however result in a tree that is overfitted to the existing records, making it unreliable when providing new data (see Figure 2.5). Complexity and robustness are in this case competing with each other; the more complex a model is, the less reliable it will be when exposed to new data (Song and Lu, 2015). Common methods are to stop (a) at a certain depth (i.e., steps from the root node), (b) when a the change of best possible information gain is less than a threshold, (c) when a subset of all data is small enough or pure, (d) when a certain amount of features has been selected.

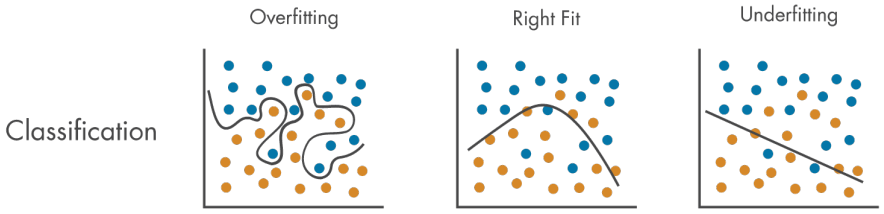


Figure 2.5.: Example of an overfitted (left) and underfitted (right) classification model that memorized the training data too well or not well enough in comparison with a correctly fitted model (middle). Adapted from Mathworks.com

Pruning

Stopping rules do, however, not always work well. Threshold values can result in trees that are not the right size, either because the tree growing is stopped too late or too early. An alternative way to create a decision tree is therefore to first grow a large tree and then prune it to the optimal size by removing nodes that provide less additional information (Dehghani et al., 2022).

There are two types of pruning: pre-pruning (forward pruning) and post-pruning (backward pruning) (Song and Lu, 2015). Pre-pruning restricts the growth of the decision tree during the training phase itself, by checking a variety of constraints (like those discussed

in Section 2.2.1). Post-pruning is a strategy in which the decision tree first grows to its full depth first, after which the unnecessary or weak branches are removed.

One of the simplest post-pruning algorithms is the reduced error pruning method. This method tries to prune a decision tree by removing the subtree at a node, making it a leaf and assigning it the most common class at that node. If the resulting decision tree performs no worse than the original, the change is kept. The nodes are removed iteratively, where the pruning continuous until further pruning becomes harmful. While somewhat naive, the reduced error pruning algorithm has the benefit of being simple and fast.

The minimal Cost-complexity pruning algorithm is another effective post-pruning algorithm. It considers the number of misclassifications (cost) and the number of nodes (complexity) to drop sub-trees that provide a minimal increase in classification cost and a maximum reduction of complexity. Simply said, if two sub-trees lead to a similar increase in misclassifications, it removes the tree with more nodes. The algorithm is parameterized by $\alpha \geq 0$ known as the complexity parameter. The complexity parameter is used to define the cost-complexity measure $R_\alpha(T)$ of a given tree T:

$$R_\alpha(T) = R(T) + \alpha \left|\tilde{T}\right| \tag{2.5}$$

where $\left|\tilde{T}\right|$ is the number of leaf nodes in T and $R(T)$ is the total misclassifications rate of the leaf nodes (Breiman et al., 2017). The cost complexity of a single node is $R_\alpha(t) = R(t) + \alpha$. The branch, T_t, is defined to be a tree where node t is its root. In general, the impurity of a node is greater than the sum of impurities of its leaf nodes, $R(T_t) < R(t)$. However, the cost complexity measure of a node t, and its branch T_t, can be equal depending on α. The effective α of a node is the value where they are equal $R_\alpha(T_t) = R_\alpha(t)$ or $\alpha_{eff}(t) = \frac{R(t)-R(T_t)}{|T|-1}$. Thus, a leaf node with the smallest value of α_{eff} is the weakest link and will be pruned. This process will stop when the pruned tree's minimal α_{eff} is greater than a threshold.

2.2.2. Random Forests

A drawback associated with tree classifiers is their high variance. In practice, it is common that a small change in the data set results in a very different tree. Decision trees are hierarchical, meaning that the question being asked depends on the previous answer. If one of the answers changes, then that results in a change of all nodes thereafter. This high variance can have a large impact on the performance of the decision tree when never before seen data is introduced. Random forests try to mitigate this effect by creating an ensemble of multiple decision trees.

During the construction of each tree in a forest, the tree is trained on a random sample of the data (bootstrap sampling) and considers only a random subset of features (feature randomization). By introducing these two sources of randomness, the forests yield decision trees with somewhat decoupled prediction errors. By taking an average of those predictions, some errors cancel out. While individual trees thus might make errors, the collective process averages out these mistakes to arrive at more reliable predictions.

Bootstrapping

When training a predictive model, it is almost impossible to have a clear picture of all the data (the population) that is available. Bootstrapping is a method used to infer characteristics

of the population from sample data. It does so by resampling from the same dataset multiple times to get an estimate of what the population distribution looks like. In its simplest form, the bootstrap might sample from the dataset randomly and create a new dataset of the same size. For example, when bootstrapping the dataset `[1,2,3,4,5]`, it picks 5 times a random sample and creates a bootstrapped dataset which can be `[2,5,4,4,1]` or `[3,3,1,5,5]` or many others. If this is done thousands of times, it will eventually provide insights on the variance of data (see Figure 2.6). It is also possible to only draw a fixed number of samples from the dataset instead of it being the same size as the sample. This is called the Bagging (Bootstrap Aggregating) method.

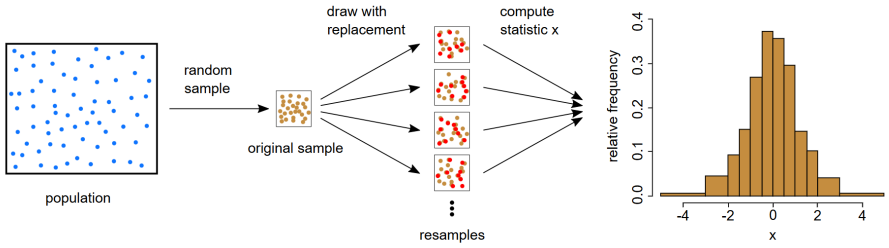


Figure 2.6.: Illustration of bootstrapping from a population sample to gain insights on the characteristics of the population. Adapted from Wikimedia commons

Combining trees

A Random Forest works by creating a unique training dataset for multiple trees, by randomly sampling from the original data with replacement (see Section 2.2.2). Each of the trees also only gets a random subset of all features available (typically the square root of the total number of features). All of the trees are then grown individually following the steps discussed in Section 2.2.1. Lastly, all trees vote for the final predictions Figure 2.7. In the case of classification, this is done by taking the most frequent class label (the mode) among the predictions of all decision trees in the ensemble. This process is known as Majority Voting, where the final prediction $\widehat{y}$ of n trees (T) is given by:

$$\widehat{y} = mode(T_1(x), T_2(x), ..., T_n(x)), \tag{2.6}$$

where $T_n(x)$ represents the prediction of the nth decision tree.

2.3. Assessing Random Forest performance

Random Forest models are powerful and versatile machine learning algorithms that can, with the right parameters, be both robust and accurate. Interpreting the results of the model can, however, be quite difficult due to the increased complexity when compared with a simple decision tree. The goal of this subsection is therefore to clarify the choices that need to be made when creating a Random Forest and how to assess the performance.

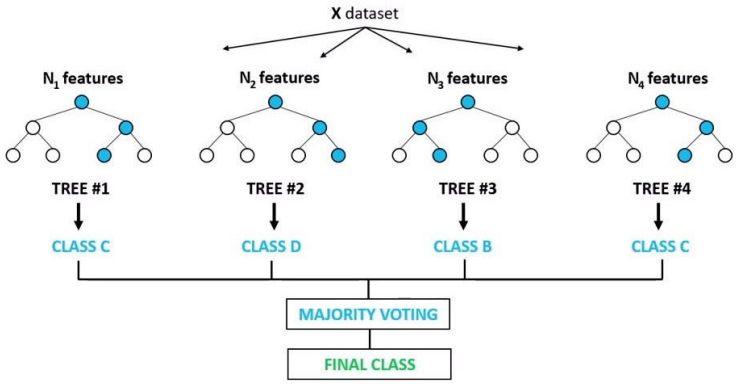


Figure 2.7.: Illustration of typical Random Forest classification using majority voting. Adapted from Abdulla et al. (2023)

2.3.1. Splitting up datasets

When training a predictive model, it is almost impossible to have all the data of the population available. A model can only be trained on a sample. To ensure a robust decision tree that will work on the entire population, it is important to split up the available data (sample) into a training and a test dataset. The training set will be used to create a decision tree, while the test dataset will be used to assess whether this tree also works when never-before-seen data is introduced. The train-test split should be balanced to ensure the model works as intended. A small test set will yield an unbiased, but unreliable, accuracy estimate for a well-trained classifier, while a large test set will yield unbiased and reliable accuracy estimates for a badly trained classifier. Train-to-test ratios of 7:3, 6:4, and 5:5 are therefore commonly used to strike the balance.

When evaluating different settings (hyperparameters) for a decision tree, like the maximum depth, for example (see Section 2.2.1), there is still a risk of the model becoming overfitted. That is, because these values can be tweaked and changed until the model performs optimally. This way, knowledge about the test set can leak into the model. The evaluation metrics will then no longer show generalization performance. To solve this problem, the dataset can be split into yet another part: a validation set. A validation set is used during the training and tweaking of the model settings, while the test set will be used as the eventual metric to test general performance.

By splitting the available data into yet another subset, the number of samples that can be used for training the model is reduced while also introducing the risk that model results can depend on a random choice of the training and validation split. A solution to this problem is to introduce a cross-validation. In the basic approach called k-fold cross-validation, the training set is split into k smaller sets (see Figure 2.8). A model is then trained using $k-1$ of the folds as training data and validated on the remaining part. The performance measure is then the average of the values of all splits. This approach is computationally expensive, but it does not waste too much data.

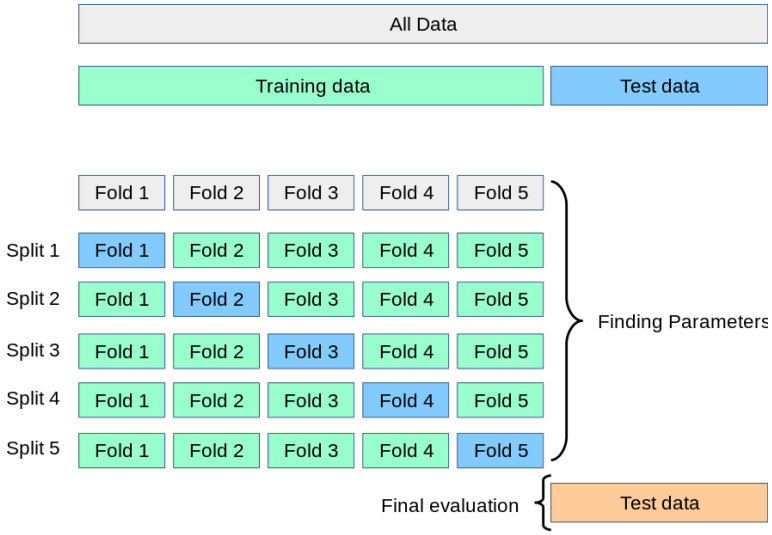


Figure 2.8.: Example of how a dataset is split up for k-fold cross-validation. Adapted from from Scikit-learn documentation (Pedregosa et al., 2011)

2.3.2. Hyperparameters

Several hyperparameters can be tuned to improve the performance of a random forest. The first is the number of trees in the forest. More trees typically improve model performance, but increase the computational cost (Zhang et al., 2023). The second hyperparameter is the maximum number of features that are considered when splitting a node. These parameters help to prevent overfitting and are often set as the square root of the total number of features. Third is the maximum allowed depth of the tree; a shallow tree may underfit, while a deep tree may overfit. Fourth is the maximum allowed number of leaves. By limiting this, the complexity and size of the trees are controlled. Fifth is the maximum sample size to determine how much of the full dataset is given to each tree. This is especially relevant when working with large datasets. Lastly, the minimum number of samples required to split a node.

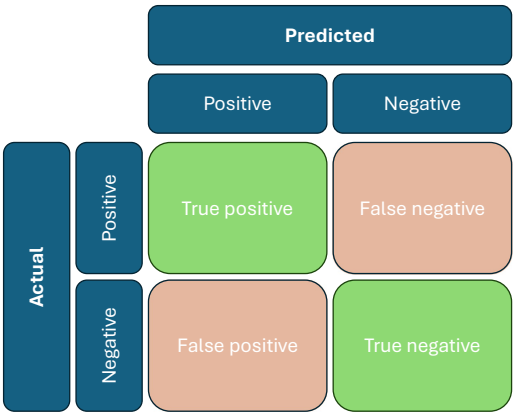
2.3.3. Performance metrics

Confusion Matrix

A confusion matrix is a simple table used to measure how well a classification model is performing. It compares the predictions made by a model with the actual results and shows where the model was right or wrong. This can help improve the model by knowing where mistakes are made. The predictions are split up into four categories (see Figure 2.9) (Powers and Ailab, 2011):

- **True Positive (TP)**: The model correctly predicted a positive outcome
- **True Negative (TN)**: The model correctly predicted a negative outcome

- **False Positive (FP)**: The model predicted a positive outcome, but the actual outcome is negative. This is also known as a Type I error.
- **False Negative (FN)**: The model predicted a negative outcome, but the actual outcome is positive. This is also known as a Type II error.



$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Figure 2.9.: Confusion matrix and formulas of its relevant metrics

A few key metrics can be calculated using the confusion matrix, which gives a better idea of performance, especially when the data is imbalanced (adapted from GeeksforGeeks). Accuracy, calculated by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

shows how many predictions the model got right out of all predictions. This gives an idea of overall performance, but can be misleading when one class is more dominant than the other. Recall measures how good a model is at predicting positives and is calculated by

$$Recall = \frac{TP}{TP + FN}. \tag{2.8}$$

High recall is essential when missing positive cases has significant consequences like in medical tests. Precision on the other hand focuses tells how many of the positive predictions were actually correct. This is especially important when false positives need to be minimized, such as detecting spam emails. The formula for precision is given by

$$Precision = \frac{TP}{TP + FP}. \tag{2.9}$$

Specificity is another important metric, particularly in binary classification. It measures the ability of a model to correctly identify negative instances. The formula is given by:

$$Specificity = \frac{TN}{TN + FP}. \tag{2.10}$$

Lastly is the F1-score, which combine precision and recall into a single metric to balance their trade-offs. This provides a better sense of a model's overall performance, particularly

for imbalanced datasets. This is helpful when both false positives and false negatives are important. The F1-score can be calculated with:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{2.11}$$

3. Scientific Article

3.1. Abstract

Accurate data on inland water surface elevations are becoming increasingly important as climate change intensifies extreme weather events worldwide. ICESat-2's advanced LiDAR instrument is well-suited for collecting this data as it is capable of measuring elevation with centimeter-level accuracy across the planet. The existing inland water data products using ICESat-2 (ATL13/ATL22) are, however, constrained by their reliance on water masks that exclude lakes and reservoirs smaller than 0.1 km^2 and rivers narrower than 50 m. To address this problem, we present a Random Forest classifier model trained on ICESat-2 ATL03 data that can identify the photons intersecting small inland water bodies ($<$ 25 m). A set of window-based features that encode the interaction between photons and water was derived at multiple window radii. These features try to detect water-specific features, including a high photon density, subsurface reflectance, the presence of afterpulses, a low photon height standard deviation, and a low slope. The number of high-confidence photons within a 2.5 m radius window proved to be the best predicting feature, contributing 41.2% of the total mean decrease in impurity. The final model relies on features calculated for a 2.5 m window and was trained on 7 million points from water segments $<$ 25 m and an equal number of land points observed in the Netherlands, for which we have good ground-truth of small water bodies. The model was evaluated on approximately 520 million ICESat-2 data points and achieved a recall above 80% for water segments longer than 6 m and up to 87.1% for water bodies between 10 m and 25 m. Potential improvements of using the features from multiple windows or selecting only windows with a minimum number of photons present proved to be ineffective in gaining better results. Key sources of misclassifications are currently the presence of snow, uncertainty near water edges, and incorrect ground truth data. Manual validation in the Swiss Alps, the Greenland ice sheet, and a Mexican mangrove forest shows promising results of the models functioning beyond the Dutch environment, suggesting that the model identified general photon-water interactions rather than region-specific characteristics. Future work should focus on expanding the training dataset to include a diverse set of environments to improve performance and combining the classified photons using clustering to create line segments of water surface elevation.

3.2. Introduction

Monitoring inland water surface elevation is becoming more important as climate change intensifies extreme weather events worldwide. There is increasing evidence that there are large differences, both spatially and temporally, in lake area, water level, and volume worldwide caused by climate change and increased human activities in recent decades (Madani, 2026; Lv et al., 2024; Feng et al., 2022). Accurate water level data is, therefore, becoming more important. This data is used to estimate freshwater availability, to forecast floods, and to

create risk assessment models. It not only influences market prices, but also affects policies and decision-making that affect future water availability and natural ecosystems (Karsten Rinke et al., 2019). However, traditional measurement methods like in-situ gauge stations and airborne Light Detection And Ranging (LiDAR) remain both spatially and temporally limited, expensive, and largely concentrated in the global north (Papa et al., 2023; Lv et al., 2024). In addition, even when water data is collected, it is not always available for scientists due to restrictions by governmental agencies or political situations (Chawla et al., 2020; Papa et al., 2010). Satellite Earth Observation, therefore, offers the only feasible approach to monitor the majority of the world's estimated 1.4 million lakes larger than 0.1 km^2, most of which lie in remote, high-latitude regions (Jasinski et al., 2025b).

The Advanced Topographic Laser Altimeter System (ATLAS) LiDAR instrument onboard the ICESat-2 is well-suited for inland water monitoring. Its high-resolution footprint (diameter 10.9 m ± 1.3 m), dense along-track sampling (every 0.7 m), and geolocation accuracy (3.5 m ± 2.1 m) make it capable of observing water bodies in great detail. Alternatives like RADAR altimetry missions can often miss smaller water bodies due to their coarser resolution and sensitivity to wind, rain, and surrounding vegetation (Magruder et al., 2020; Biancamaria et al., 2018). Because of the potential of ICESat-2, NASA already provides two data products: ATL13 Inland Surface Water (Jasinski et al., 2025b) and ATL22 Mean Inland Surface Water (Jasinski et al., 2025a). These products provide processed inland water levels, but are constrained by their reliance on water masks. The water masks exclude lakes and reservoirs smaller than 0.1 km^2 and rivers narrower than 50 m (Jasinski et al., 2025b), missing many of the smaller water bodies around the world. The masks are also too static, which causes both products to miss capturing dynamic changes in water extent during seasonal changes, drought, or flooding. As a result, many studies resort to the lower-level ATL03 Geolocated Photons product (Neumann et al., 2025b) combined with independently derived water masks (Kaya et al., 2025; Song et al., 2023).

Deriving such water masks independently introduces its own limitations. The most common approach to creating these masks is by using optical satellite imagery, which cannot see through vegetation. Furthermore, these observations may not temporally coincide with an ICESat-2 overpass, meaning the observed water extent may not reflect actual conditions at the time of measurement (Yang et al., 2024). This motivates a fundamentally different approach: using ICESat-2 photon data itself to both detect and measure water bodies simultaneously, without relying on water masks. Ma et al. (2024a) has done so for areas in Guangdong in China and Borneo in Malaysia, while Datta and Wouters (2021) used ICESat-2 data to identify water bodies and measure bathymetry in Greenland. Both papers have demonstrated that this concept works for larger water bodies in their specific environments, but the generalizability to smaller and globally distributed water bodies remains undemonstrated.

In this paper, we present a Random Forest (RF) classifier trained on ATL03 data that identifies photons reflected from small inland water bodies, relying exclusively on features derived from the ICESat-2 data itself. Specifically, the focus is on water bodies with an extent < 25 m, which is around the edge at which satellites can currently delineate water bodies (Radoux et al., 2016; Schmitt, 2020; Li et al., 2013; Pekel et al., 2016). We describe the photon-water interaction features underlying the model in Section 3.4 and present training, evaluation, and validation results in Section 3.5. The final model was trained on 14 million points from the Netherlands with features calculated using a 2.5 m radius window around each photon. The Netherlands was chosen as the training area because of its high-quality water masks, including detailed coverage of very small water bodies. The RF was trained on only data from water bodies < 25 m and evaluated on approximately 520 million ICESat-2

observations. It achieved a recall above 80% for water segments longer than 5 m, reaching 87.1% for water bodies between 10 m and 25 m. Manual validation in the Swiss Alps, the Greenland ice sheet, and a Mexican mangrove forest suggests the model captures general photon-water interactions rather than region-specific characteristics, indicating potential for global applicability. External data sources are used only during training; the final model operates without water masks or other auxiliary inputs, making it applicable to unmapped and dynamically changing water bodies worldwide.

3.3. Background and related work

Multiple interactions between water and photons can be used to identify water bodies. Previous studies have used different types of these interactions for classification purposes. Ma et al. (2024a) based their method on the difference in the elevation standard deviation between land and water. Photons that interact with water have a smaller elevation standard deviation and a higher density standard deviation than land photons do.

In their algorithm to identify supraglacial lake bathymetry, Datta and Wouters (2021) uses more general characteristics of water, taking the flatter slope when compared to land and the presence of bottom reflectance as key features for classification. By using a kernel density estimate, they were able to differentiate peak returns on different elevations. This could help in identifying overgrowth, ice coverage, and bottom reflectance.

Another interesting interaction observed in the ICESat-2 data is a *ringing* effect beneath water surfaces (Neuenschwander and Magruder, 2019). Still standing, non-turbulent water exhibits this characteristic, also when there is vegetation present (see Figure 3.1). Neumann et al. (2025a) explains that this *ringing* effect occurs in the data because the LiDAR instrument on board of ICESat-2 (Advanced Topographic Laser Altimeter System (ATLAS)) cannot handle the strong return signal created by water. ATLAS thus detects multiple surface returns, with echoes spaced by either one or two times ATLAS' dead time. Dead time is the time required for a single detection element to detect a photon and reset itself to be capable of observing the next. The second return will be observed around 0.5 m below the surface with another, tertiary, return beneath that (Neumann et al., 2025a).

On top of these, there are also echoes present at around 2.3 m and 4.2 m below the primary surface return when the surfaces are relatively flat (Neumann et al., 2025a). These are likely present due to small after-pulses in either the ATLAS transmitted pulse or a small amount of electronic noise following the primary surface return. Multiple surface echoes are typically seen in granules containing very smooth open water surfaces (such as inland water or leads in sea ice) when surface winds are negligible.

3.4. Methodology

3.4.1. Data acquisition and preprocessing

The main data source for our classifier model is the ATL03 Geolocated Photons data product. Each granule of this product contains, among other fields, the height above the WGS 84 ellipsoid (ITRF2014 reference frame), latitude, longitude, and time for all photons observed

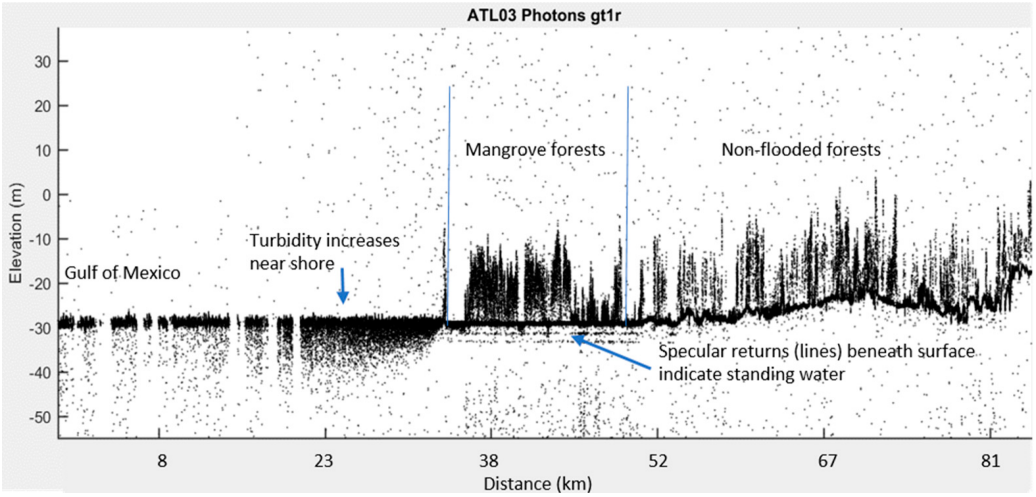


Figure 3.1.: Profile of ICESat-2 over a mangrove forest in Mexico highlights *ringing* effect (specular return lines) beneath the surface, indicating the presence of water. Adapted from (Neuenschwander and Magruder, 2019)

by the ATLAS instrument (Neumann et al., 2025b). All photons also have a confidence value ranging from -2 to 4, where everything ≤ 1 indicates that a photon is noise, and higher values represent a higher confidence of the point being a true signal. The data was downloaded within a spatial extent of `[3.0 - 7.5]` longitude and `[50.5 - 53.7]` latitude, corresponding to a bounding box around the Netherlands. The Netherlands was chosen as the training area because of its high-quality water masks, including detailed coverage of very small water bodies. A total of 1087 granules were downloaded, amounting to 1.68 TB of raw data (collected up to 19/11/2025).

The raw granules were preprocessed in three steps. First, all photons with a signal confidence < 2 were discarded to reduce noise in the calculated features and reduce computing time. Although photons with a confidence of 2 and 3 are often still noisy when compared to the points with confidence 4, these were still included as personal exploration of the data indicates that sub-water-surface points often have lower confidences. Second, a land mask of the Netherlands was used to remove all points observed in other countries or at sea. Photons intersecting with large inland lakes like the IJsselmeer and Markermeer, however, were kept. Third, the remaining points were split into six separate files, one for each of the ICESat-2 beams, and saved as GeoParquet for its efficient compression and fast querying capabilities. Besides the coordinates of the points, the `quality`, `uncertainty`, `sun_angle`, `detector_id` and `height_reference` were also saved (see Appendix C for more information). After preprocessing, approximately 530 million points remained. Of these, 86.7% were classified as high confidence (4), with medium- (3) and low-confidence (2) photons accounting for 5.5% and 7.8% respectively.

3.4.2. Feature engineering

Since a single data point cannot be used to classify something as water or land, all features were computed within a local window centered around each photon to add contextual in-

formation. To determine the appropriate spatial scale at which to calculate the features, all were calculated at six window radii: 1 m, 2.5 m, 5 m, 10 m, 20 m, and 25 m. Smaller radii are better for identifying sharp transitions between water and land and for capturing fine-scale patterns in smaller water bodies. Larger radii, on the other hand, include more context points, which can be used to calculate more stable statistics. Window radii larger than 25 m were not considered since a window wider than the target water body will extend beyond both edges and include land photons. This will weaken the water signal and reduce the discriminative power of the features. Since this paper targets water bodies < 25 m, a 25 m radius already represents the upper limit at which a window can be centered on a small water body without being dominated by surrounding land returns. The candidate features can be grouped according to the photon-water interaction, identified in Section 3.3, which they try to encode.

High photon density and strong surface reflectance. Water surfaces produce a strong specular return that results in a high density of high-confidence photons within a window. To capture this, the number of photons at each confidence level was recorded, alongside statistics on along-track distances between photons (`mean`, `median`, and `standard deviation`). These features try to encode the concentration and spread of photons along the track.

Flat surface and low slope. Water surfaces are most often flatter than land and produce returns with a lower elevation standard deviation and a smaller along-track slope. Along-track statistics were therefore computed, including the `slope`, the `residual` (deviation from a linear surface fit), and a set of standard height statistics: `mean`, `median`, `standard deviation`, `range`, `inter-quartile range`, `skewness`, and `kurtosis`. The fractions of photons within 0.1 m and 0.2 m of the median elevation (`frac_01m`, `frac_02m`) were also included as a direct measure of surface flatness.

Subsurface reflectance and bathymetric returns. Photons can penetrate the water surface and return from the bottom, producing secondary peaks in the height distribution below the primary surface return. A 50-bin histogram was, therefore, used to identify peaks in the height distribution, with the number of peaks and the height of the most prominent peak saved as features. These peaks can provide information about bathymetric returns, overgrowth, or specular returns.

Ringing effect at dead-time spacing. Still-standing, non-turbulent water produces a characteristic *ringing* effect beneath the surface return, caused by the ATLAS instrument being unable to handle the strong returning signal from water (Neumann et al., 2025a). The instrument detects multiple surface returns spaced by one or two times the ATLAS dead time, with a second return at approximately 0.5 m below the surface and additional echoes at around 2.3 m and 4.2 m below the primary return. To capture this interaction, the number of photons beneath the prominent surface peak at dead-time-like spacing (in the range $[0.3\,\mathrm{m} - 0.7\,\mathrm{m}]$) was calculated together with their mean distance and standard deviation.

Many of the features identified above require a minimum number of data points to provide meaningful statistics. We therefore decided to give all windows with < 5 points *NONE* values for all features, to prevent noisy data from entering the model.

3.4.3. Model selection

We identified five requirements for our classification model based on the dataset, features, and research question. First, the model must be capable of capturing non-linear relationships. Second, it must be robust to noise, given that the ATL03 product contains low- and medium-confidence photons that are noisy. Third, it must be non-parametric, as the dataset combines binary, continuous, and discrete features with unclear distributions. Fourth, it must be able to handle missing values, which are unavoidable when features such as dead-time spacing statistics cannot be computed when there are no sub-surface photons in a window. Fifth, it must be computationally lightweight, as the model is evaluated on hundreds of millions of points and training is repeated for six window radii.

A Random Forest (RF) classifier satisfies all five requirements. Logistic regression assumes a linear decision boundary and cannot capture the nonlinear relationships present in the data. Support vector machines can approximate nonlinear boundaries through kernel functions, but are computationally expensive at this scale and do not natively handle missing values. Deep learning approaches are capable of learning complex spatial patterns, but are sensitive to hyperparameter choices and are computationally expensive to train. The RF, by contrast, can train efficiently on large datasets by using parallelization, handles mixed data types and `NULL` values natively, and makes no assumptions about the input features. As an additional benefit, it produces MDI importance scores as a by-product of training, providing direct insight into which photon-water interactions are most predictive, making the model more interpretable.

3.4.4. Feature selection

The full candidate feature set contains 49 features (see Appendix C). Reducing this to a target of 15 served three purposes: (i) it lowers the computational cost of training the RF, (ii) it ensures only the most relevant features are retained to reduce noise in the model, and (iii) it reduces the risk of overfitting on irrelevant features. We aimed for 15 features as a target, as this is large enough to represent each of the photon-water interactions identified in Section 3.3 with multiple features, while being small enough to keep the models' computational cost manageable and avoid fitting on redundant predictors. Through allowing the features from multiple identified water-photon interactions, the hope is that model robustness will increase and that the model will be scalable to environments that differ from the Dutch landscape.

To select features, we settled on an embedded approach using Mean Decrease in Impurity (MDI) scores, which are computed as part of the RF training process. This allows feature importance to be assessed based on interactions between features rather than in isolation. It also can handle mixed data types. MDI scores carry two known limitations, however: (i) bias towards high-cardinality features and (ii) sensitivity to overfitting, since importances are computed on training set statistics (Breiman, 2001). We addressed this by combining MDI with permutation importance, which directly measures the degradation in model performance on the test set when a single feature is randomly shuffled, providing a metric that identifies the potential of a model to generalize (Breiman, 2001).

Correlated features further complicate importance scoring, as they split their MDI scores between each other and produce near-zero permutation scores when one can substitute for the other. The selection procedure, therefore, was done in two steps. First, hierarchical

clustering using Ward's linkage (Ward, 1963) with a threshold distance of 0.4 grouped correlated features and reduced the set to one representative per cluster. Binary features were excluded from this, as Spearman's rank correlation is unreliable on them. Second, a RF was trained on the decorrelated set, and the 15 features with the strongest combined MDI and permutation importance scores were selected. The feature selection procedure was repeated independently for each of the six window radii.

3.4.5. Ground truth

The *Basisregistratie Grootschalige Topografie (BGT) waterdeel* dataset contains polygons of all waterbodies in the Netherlands with a minimum accuracy of 60 cm (Geonovum, 2020). These include rivers, canals, streams, ditches, lakes, ponds, and fens (see Figure 3.2). Dry ditches and mudflats are also included, as there is sometimes water present. The data is collected through a variety of methods, including land surveying and mapping from stereo aerial photos. All definitions are available at Geonovum and data can be downloaded via PDOK (accessed 16/02/2026).

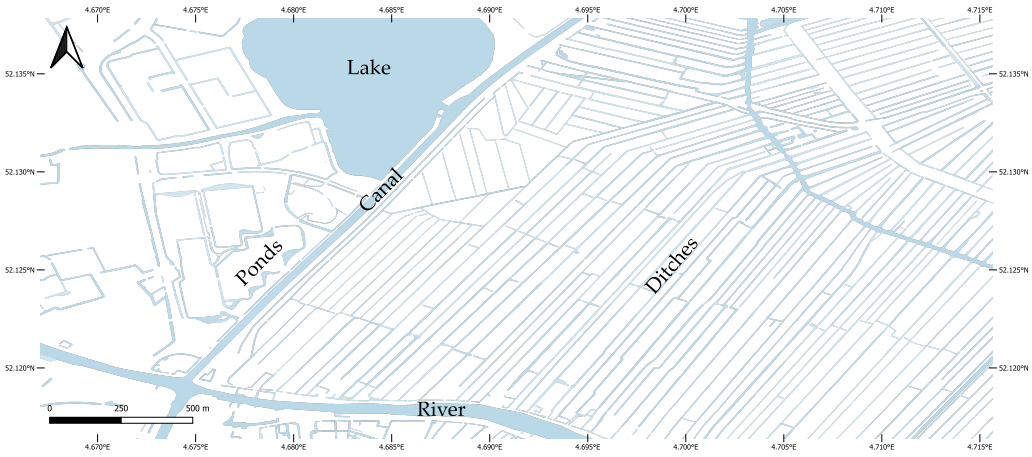


Figure 3.2.: The BGT waterdeel contains polygons of all waterbodies in the Netherlands, including rivers, canals, streams, ditches, lakes, ponds, and fens.

Ground truth was added to all data in the dataset by computing the intersection between each photon point and the water polygons in the BGT; points that intersected with water were assigned a positive Boolean value. A column was also added containing the length of the intersection along-track with a water body as presented in Figure 3.3. This length indicates how large a water body is; when trying to detect water bodies < 25 m, we thus refer to an along-track water segment length < 25 m. Approximately 165 million of all points (30.9%) intersect with a water body polygon in the Netherlands. Of these points 11.2% have an intersection length < 10 m, while 9.7% have an intersection between 10 m and 25 m.

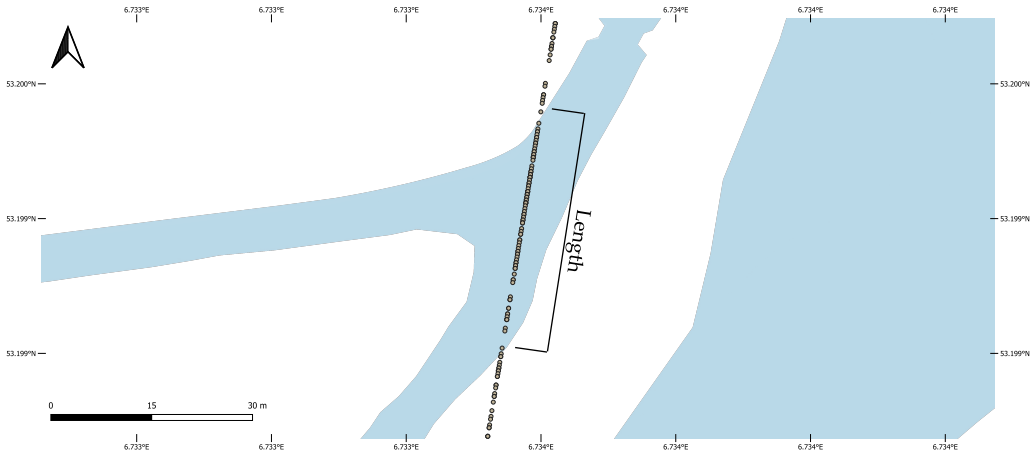


Figure 3.3.: The *size* of a waterbody is calculated by determining the intersection length of the polygon and the ICESat-2 beam

3.4.6. Model training

Six RF models were trained in parallel, one for each window radius, to enable a direct comparison of their predictive performance. We binned the water points into 7 different water body sizes to assess how performance will differ based on size [$<$ 10 m, 10 - 25 m, 25 - 50 m, 50 - 100 m, 100 - 250 m, 250 - 500 m, $\geq$ 500 m]. This way, the model can also be assessed for general water detection performance as opposed to just smaller water bodies. The difficulty of classification is expected to vary between water body sizes, and separating the bins makes this variation in performance visible rather than obscuring it in an average. The bins were chosen to all have around the same percentage of points, except the $\geq$ 500 m bin, which was selected to capture all large water bodies.

Since the primary goal of this research is to identify smaller water bodies, each model was trained exclusively on water bodies $<$ 25 m. The training set was constructed by selecting 3.5 million points from each of the two target bins ($<$ 10 m and 10 - 25 m), resulting in 7 million water points in total. An equal number of land points was then added, giving a 1 : 1 water-to-land ratio. To ensure that the same points were selected across all six window radii to isolate the effect of window width from random variation in sampling, we used a hash function based on each photon's `latitude`, `longitude`, and `height`. This deterministic approach made it possible to draw identical subsets from all files without loading the full dataset into memory.

3.4.7. Evaluation design

A total number of 14 million points was selected for the training set, as that was near the edge of what our computing power could handle. This means the model was trained on only around 2.6% of all points, which is significantly lower than a normal train-evaluation split of 80% - 20%. We, however, believe this is a benefit, as this small training sample can test

how quickly the model generalizes and thus whether the selected features actually capture water-photon interactions.

Each model was evaluated on the full remaining Dutch dataset of approximately 520 million points after removing the training samples. Recall was used as the primary metric for assessing per-bin performance, as it only requires the true positives and false negatives within each bin and is therefore not influenced by the water-land class imbalance present in the evaluation data. Missing a small water body entirely is a more consequential error than incorrectly flagging a land point as water, since false positives can be filtered in post-processing steps, whereas undetected water bodies are lost entirely. The overall accuracy and precision are reported alongside recall, but both aggregate counts across the full dataset and are therefore sensitive to the overrepresentation of large water bodies. These metrics are therefore less informative for assessing performance on the smaller bodies.

To assess whether the model generalizes beyond the Dutch environment, manual validation was performed in three environments with distinct characteristics. First, ICESat-2 granules intersecting small streams near Gordevio in the Swiss Alps were inspected to test performance on steep terrain and on narrow streams that vary in size between seasons. Second, granules intersecting the Sermeq Kujalleq area in Greenland during the melt season of 2019 were processed, following the area and period used in Datta and Wouters (2021), to assess performance on supraglacial meltwater lakes and streams. Third, beams passing through the mangrove forest of Parque Nacional Lagunas de Chacahua in Mexico were evaluated to test the detection of water beneath a dense tree canopy. As high-quality water masks for small water bodies are unavailable in most of these areas, validation in all three environments was qualitative, based on manual inspection of elevation profiles and comparison with available satellite imagery.

3.4.8. Hyperparameter tuning

Five hyperparameters control the learning process of the RF and determine the extent to which the model fits the training data: (i) the maximum depth of each decision tree, (ii) the maximum number of features considered at each split, (iii) the minimum number of samples required in a leaf node, (iv) the minimum number of samples required to split an internal node, (v) and the number of decision trees in the forest.

Each hyperparameter was tuned independently using a validation curve, in which all other parameters were held constant while one was varied across the range specified in Table E.1. Model performance was assessed using 5-fold cross-validation with the F1-score as the evaluation metric, which combines precision and recall into a single value; higher scores reflect better overall performance. These curves are not fully representative of the combined impact on model performance, since the dependence between hyperparameters is not considered, but they provide a reliable initial indication of where performance saturates.

Final values were chosen conservatively, as the dataset used for hyperparameter tuning consisted of only 1.4 million points (0.26% of all available data). Selecting aggressive values on such a limited subset risks overfitting to the tuning data, particularly for parameters that control tree depth and minimum sample requirements. The chosen values are listed in Table E.1.

3.4.9. Method implementation

The methodology was implemented using both Python and Julia. The following libraries were used during the process:

1. SpaceLiDAR (Pronk and Gardner, 2026) was used to download and process parts of the ATL03 data in Julia.
2. Scikit-learn (Pedregosa et al., 2012) was used for the RF algorithm implementations in Python.
3. Dask (Dask Development Team, 2016) was used for its ability to handle larger-than-memory datasets.
4. Joblib (Varoquaux et al., 2024) made it possible to save and re-use the RF models created by Scikit-learn.

Julia was used for its fast execution times; all data processing is implemented in Julia. The switch to Python was made for the Dask package, which can handle datasets larger than memory, and for Scikit-learn, which RF classifier handles *NULL* values natively. We created two plugins for QGIS (Dawson et al., 2026) to visualize results throughout the model creation process. One of the plugins can plot a selection of the along-track data of a single beam, with the ability to add variables layered on top (Rotteveel, 2026b) (see Section F.1). The other QGIS plugin can visualize RF classifications, by loading a Joblib file of the model (Rotteveel, 2026c) (see Section F.2). This makes it possible to see classifications, confidences, and the correctness of the decisions. All the code for reproducing the work in this paper is also publicly available at Rotteveel (2026a)

3.5. Results and evaluations

3.5.1. Final model performance

The adopted final model uses a 2.5 m window radius and was trained exclusively on water bodies < 25 m, as this window width achieved the highest recall (Table E.7). Its performance on the held-out Dutch dataset, of approximately 520 million points, is presented in Figure 3.4. The model achieves a recall of 76.7% on water bodies < 10 m and 87.1% on the 10 - 25 m bin. Looking at the performance of individual meter bins < 10 m in Figure 3.4, however, shows that the recall increases steadily with water length, reaching 80% above 6 m. Performance for the smallest water bodies is considerably lower. This size-dependent degradation is expected, as smaller water bodies contain fewer photons per window, because of the 0.7 m along-track sampling rate (Magruder et al., 2020), and these smaller bodies are more affected by edge effects, where the 2.5 m radius window is more likely to include land noise. As a consequence of training exclusively on < 25 m bodies, performance degrades substantially for larger bins, with the recall dropping to 51.0% for water segments ≥ 500 m. This is a conscious trade-off, as large water bodies can already be identified reliably using other methods (see Section 3.2); the model is intentionally optimized for the smaller water bodies.

The overall accuracy of 72.4% and precision of 53.7% are poor summaries of model performance and should not be interpreted in isolation. Both metrics aggregate counts across

Bin	Rec.	Conf.
<10 m	0.767	0.705
10–25 m	0.871	0.740
25–50 m	0.858	0.733
50–100 m	0.848	0.730
100–250 m	0.813	0.723
250–500 m	0.740	0.715
≥500 m	0.510	0.729
Land	0.745	0.751
Accuracy	0.724	
Precision	0.537	

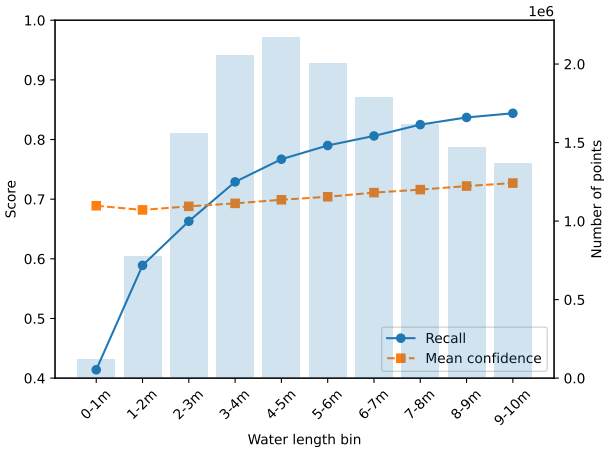


Figure 3.4.: (*Left*) Recall (Rec.) and confidence (Conf.) scores per water length bin. Smaller windows perform better on shorter segments, while wider windows perform better on longer ones. (*Right*) Model performance per water segment intersection length < 10 m. Recall is low for small water bodies but exceeds 80% above 6 m; exact values are listed in Table E.6.

the full evaluation set, which is strongly dominated by water segments ≥ 500 m and land points. The low recall on large bodies lowers the overall accuracy and precision, making these metrics not reflect the performance on the small water bodies. The recall of individual bins should therefore be used for assessing whether the model succeeds.

Confidence is defined as the mean probability assigned to the predicted class across all points in each bin, regardless of whether the prediction was correct. The confidence scores are stable across all bins, ranging from 0.705 to 0.751, suggesting the model is consistently decisive regardless of water body size. This is notable for the ≥ 500 m bin, where the model is wrong nearly half the time yet still predicts with a confidence of 0.729, indicating that it misclassifies large water bodies with high certainty rather than hesitation. The model has learned a decision boundary optimized for small water bodies, which results in overconfidence when applied in a new context. Confidence values around 0.70, however, are quite low and indicate there is room for improved calibration.

3.5.2. Selected features

The MDI importance scores of the fifteen selected features are presented in Figure 3.5, together with the importance per tree in the forest ($N = 100$). The selected features contain a mix of the water detection properties identified in Section 3.3, confirming that the model draws on multiple distinct photon-water interactions rather than relying on a single signal.

The strongest predictor is `conf_4`, the number of high-confidence photons present in each window, which accounts for 41.2% of total importance. This feature captures both the density of photons and the strength of the surface reflection simultaneously, making it the

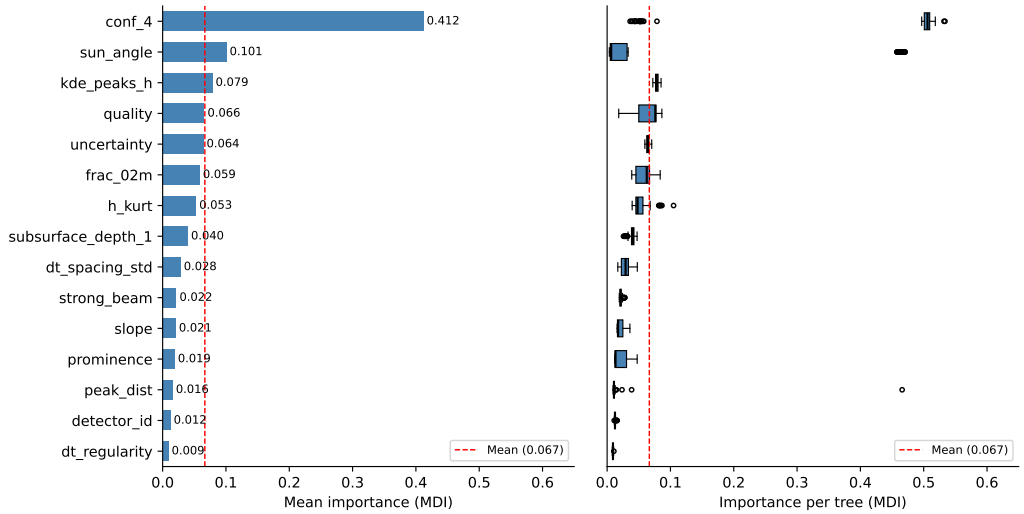


Figure 3.5.: Feature importances of the Random Forest, ranked by mean decrease in impurity (MDI). The dashed line indicates the mean importance across all features ($N = 15$). Feature `conf_4` is the most important feature accounting for 41.2% of total importance.

most informative single indicator. Surface flatness is represented by `frac_02m`, the fraction of photons within 0.2 m of the median elevation, which achieved the highest permutation importance score despite ranking lower on MDI. Subsurface reflectance is captured through `kde_peaks_h`, which identifies the number of peaks along the height axis, and `subsurface_depth_1`, which records the depth of the first subsurface return below the primary surface peak.

To further understand the predictive capability of the features, we plotted the distributions of individual features in Figure 3.6 using a random sample of land and water (< 25 m) points. Outliers were removed, and *NONE* values were ignored. These plots can highlight how discriminatory a single feature is, but do not consider potential interactions between features. The plots do, however, highlight how certain features are more prominent when water is present. For example, by looking at the n values, you can see that twice as many water points have `kde_peaks_h` values. The presence of a value alone can already be used to differentiate between water and land, independent of the actual value.

Two features specifically warrant attention. The `uncertainty` and `quality` features are both strong predictors in Figure 3.5 despite describing properties of individual photons rather than aggregates over all photons in the window. The distribution of the `quality` feature in Figure 3.6 shows that it can be clearly used to separate water, while `uncertainty` appears to have a distribution that is similar between water and land, making it non-discriminatory on its own. When combined with other features like `prominence` in Figure 3.7, its predictive power becomes more apparent. The `sun_angle` feature also has a distribution overlap in Figure 3.6, while showing a high MDI score that is strongly influenced by a small number of trees with outlier importances. The outliers most likely are caused by strong interactive prediction with the number of high confidence points in a window (`conf_4`) highlighted in Figure 3.7. The `sun_angle` might, however, be acting as a proxy for geographic or seasonal patterns. The sun angle varies across the globe and between seasons, meaning this could be a form of overfitting to the Dutch conditions.

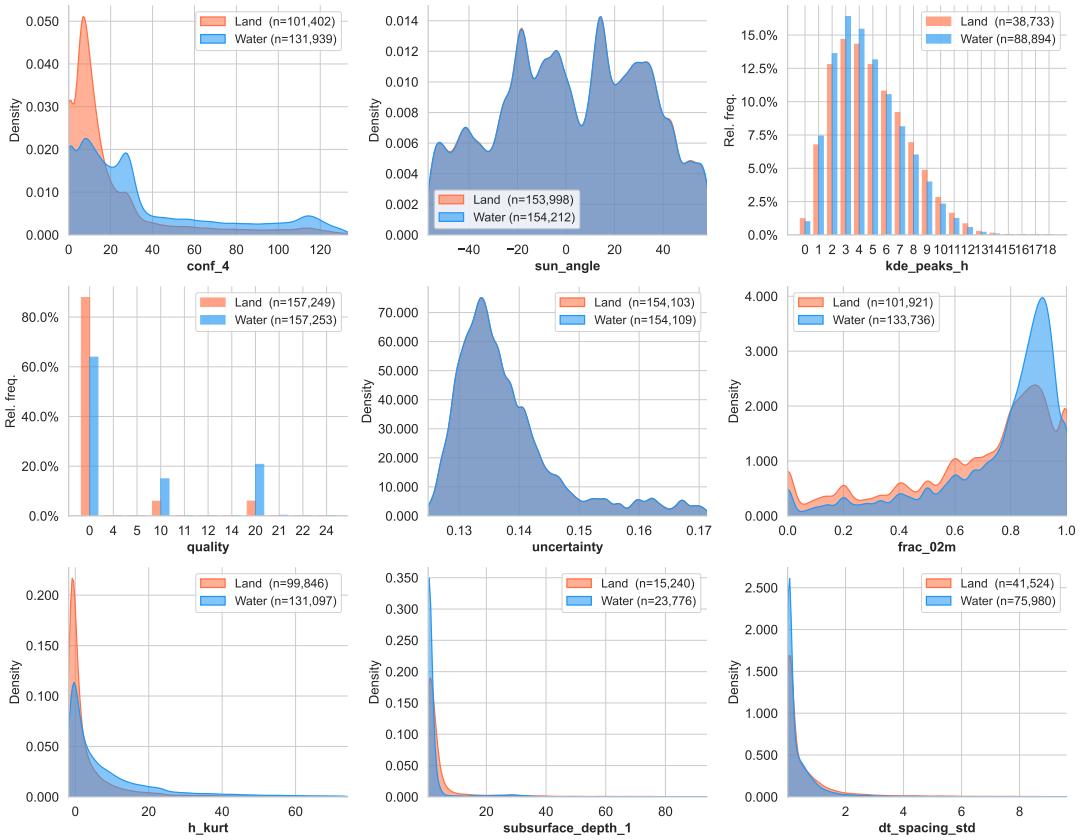


Figure 3.6.: Distribution of the 9 highest ranked features compared between water and land data. Outliers at the 1% extremes are removed. Parts of the distribution that do not overlap show the discriminatory potential of a single feature.

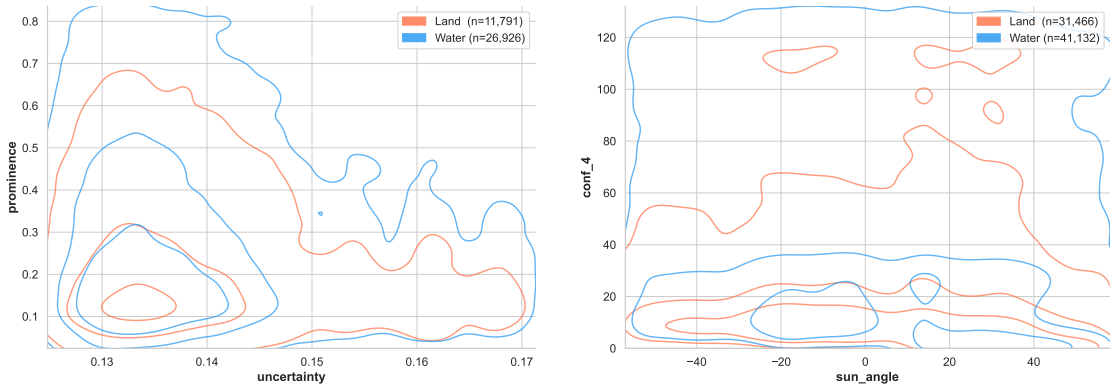


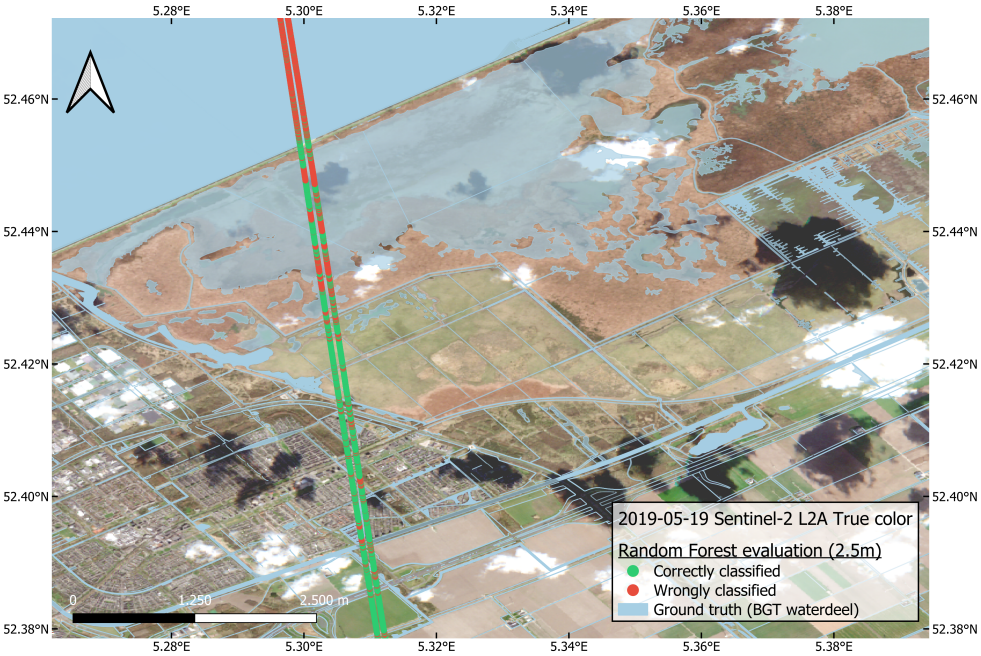
Figure 3.7.: A bi-variate distribution of the `uncertainty` x `prominence` (*left*) and `sun_angle` x `conf_4` (*right*), where the hue represents the distribution based on the class. The difference between the outer land border and the outer water border indicates the water points that can be discriminated.

The dead-time and ringing features are absent from the final selection. Although specular returns at dead-time spacing beneath the surface are a definite physical indicator of water presence (see Figure 3.1), the dead-time statistics did not achieve sufficient importance across any of the six window radii tested (see Section E.1). The pulses beneath the surface only appear clearly enough in the data at spatial scales larger than the windows considered in this thesis. Turbidity, furthermore, also suppresses the ringing effect, making dead-time spacing a viable indicator only for still-standing water of larger water bodies (Neuenschwander and Magruder, 2019).

3.5.3. Model limitations

Although per-bin recall quantifies how often the model detects water, it does not reveal where, when, or why the model fails. Four sources of misclassifications are worth discussing: (i) limitations of the ground truth, (ii) reduced confidence near water edges, (iii) misclassifications under highly reflective conditions, and (iv) sensitivity to snow cover.

Ground truth limitations. The BGT water polygons used as labels were collected over a seven-year period and may not reflect the actual water extent at the time of each ICESat-2 overpass. A concrete example is the Oostvaardersplassen nature reserve, a marshland area classified as water in the BGT even though it is sometimes dry. Validation against Sentinel-2 imagery captured one day after the ICESat-2 observation confirms that the model correctly identified parts of this area as land, yet these predictions are recorded as false negatives because the ground truth labels them as water. The reported false positive and negative rates are therefore not fully representative of real-world performance. Figure 3.7 also highlights that the model struggles with large water bodies: all points over the IJsselmeer were classified as land, consistent with the low recall of 51.0% for the $\geq 500\,\text{m}$ bin (Figure 3.4) and the model having been trained exclusively on water bodies $< 25\,\text{m}$.



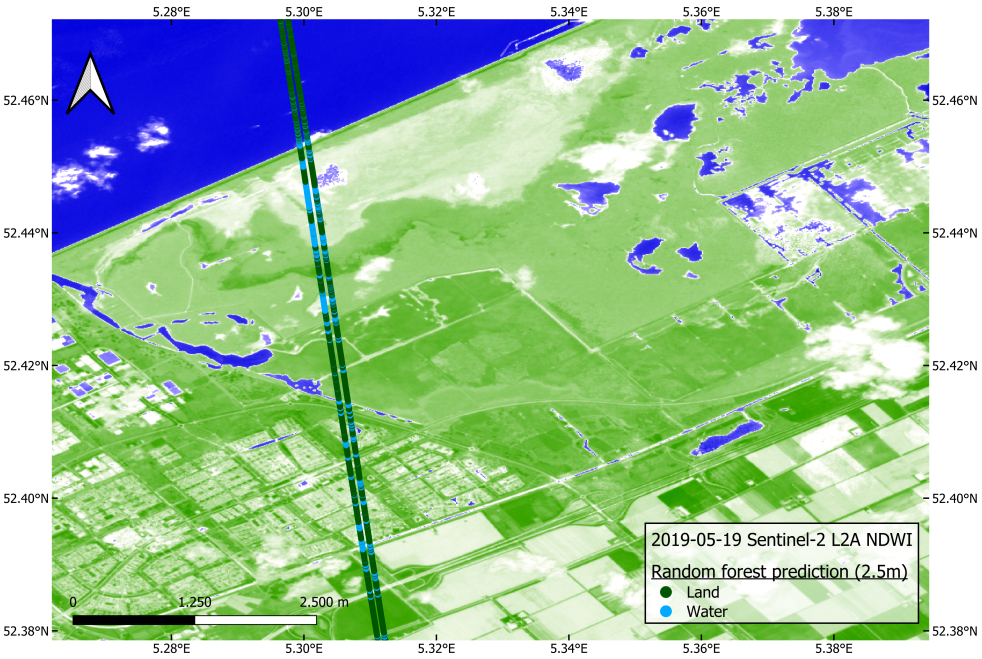


Figure 3.7.: Ground-truth at the Oostvaardersplassen (*top*) is not representative of the real scenario (*bottom*), resulting in predictions that are recorded as false negatives because the ground truth labels them as water. Observed ICESat-2 data on 18-03-2025. Normalized difference water index data observed by Sentinel-2 L2A on 17-03-2025.

Reduced confidence near water edges. The model shows systematically lower confidence near the edges of water bodies, as shown in Figure 3.7. This is partly a physical effect: the strong reflectance of water produces photon returns up to half of the ICESat-2 beam footprint (7 m to 8 m) away from the actual edge (Pronk et al., 2024), causing water-like signals to appear beyond the true boundary. This overestimation noise is visible in the elevation profile of Figure 3.7. A contributing factor is that the ground truth polygons do not always precisely delineate channel edges, meaning some genuine water points fall outside the labeled polygons.

Highly reflective surfaces. Since the model relies on the high reflectance of water as its primary signal, other highly reflective surfaces pose a potential misclassification risk. The greenhouse roofs in Pijnacker produce noisy photon returns (see Figure 3.8), but do not lead to widespread misclassification, with most points correctly identified as land. The exceptions are small water channels passing through the area, which the model failed to detect. This is consistent either with the general difficulty of identifying water in the smallest size bins (see Figure 3.4), or with the greenhouse noise contaminating the window features sufficiently to prevent a correct classification.

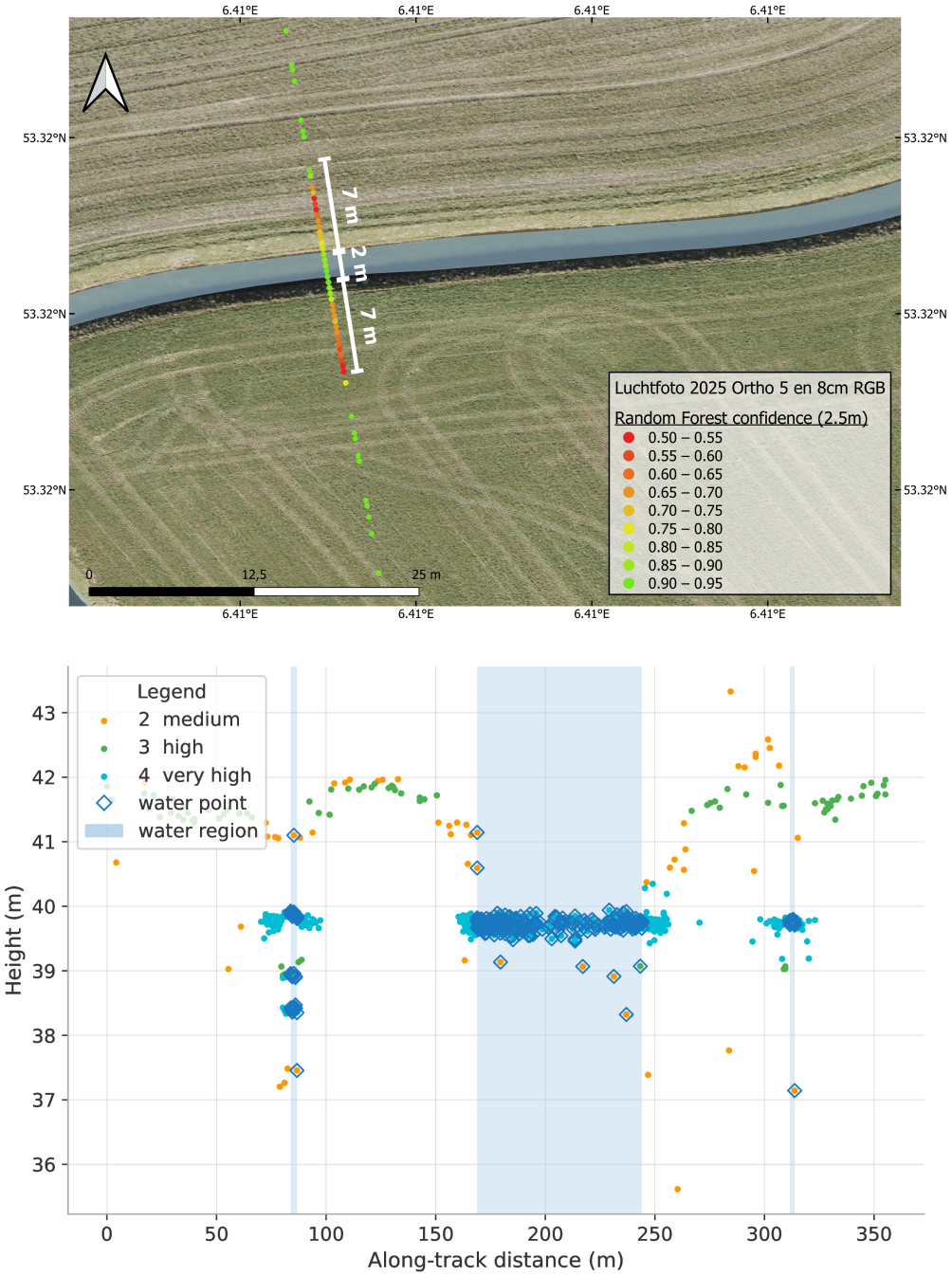


Figure 3.7.: The model has a lower confidence near water edges up to about 7 m (half of the ICESat-2 beam footprint) (*top*). Due to the strong reflective property of water, water signals are still present past the water body edges (*bottom*). Observed ICESat-2 data (beam gt3l/r) on 23-11-2018.

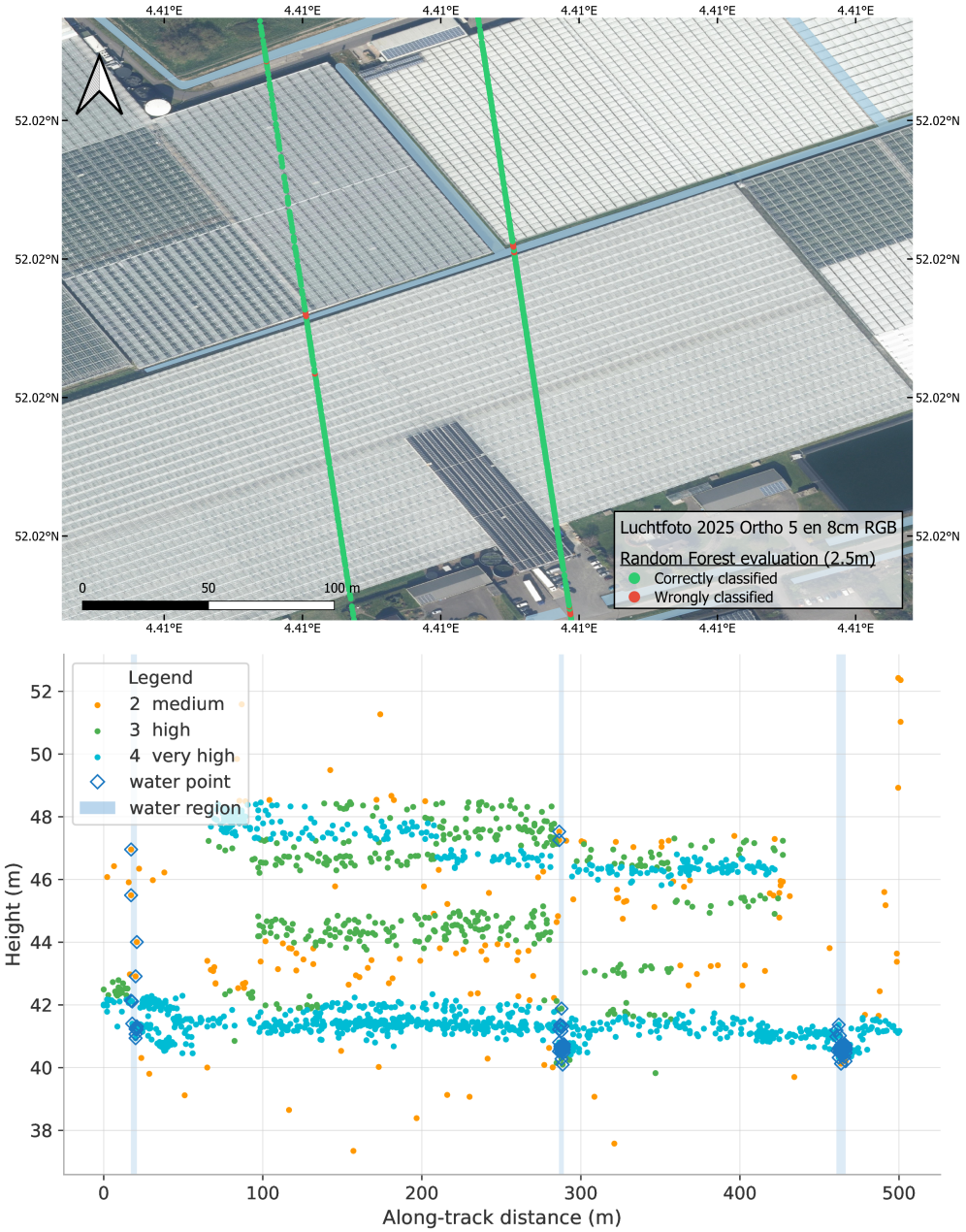


Figure 3.8.: The highly reflective surfaces of greenhouses in Pijnacker (the Netherlands) result in noisy data, but not in misclassifications by the model. Observed ICESat-2 data (beam gt1l/r) on 03-03-2020.

Snow cover. Snow presents a more severe challenge. As shown in Figure 3.8, the photon return pattern of snow closely resembles that of water reflectance, resulting in highly inconsistent classifications and many misclassified points along both beam tracks. Confidence values for snow-covered areas fall between 0.5 and 0.7, indicating that the model is uncertain but still commits to incorrect predictions. This is likely because too few snowy acquisitions are present in the training data for the model to learn to distinguish snow reflectance from liquid water.

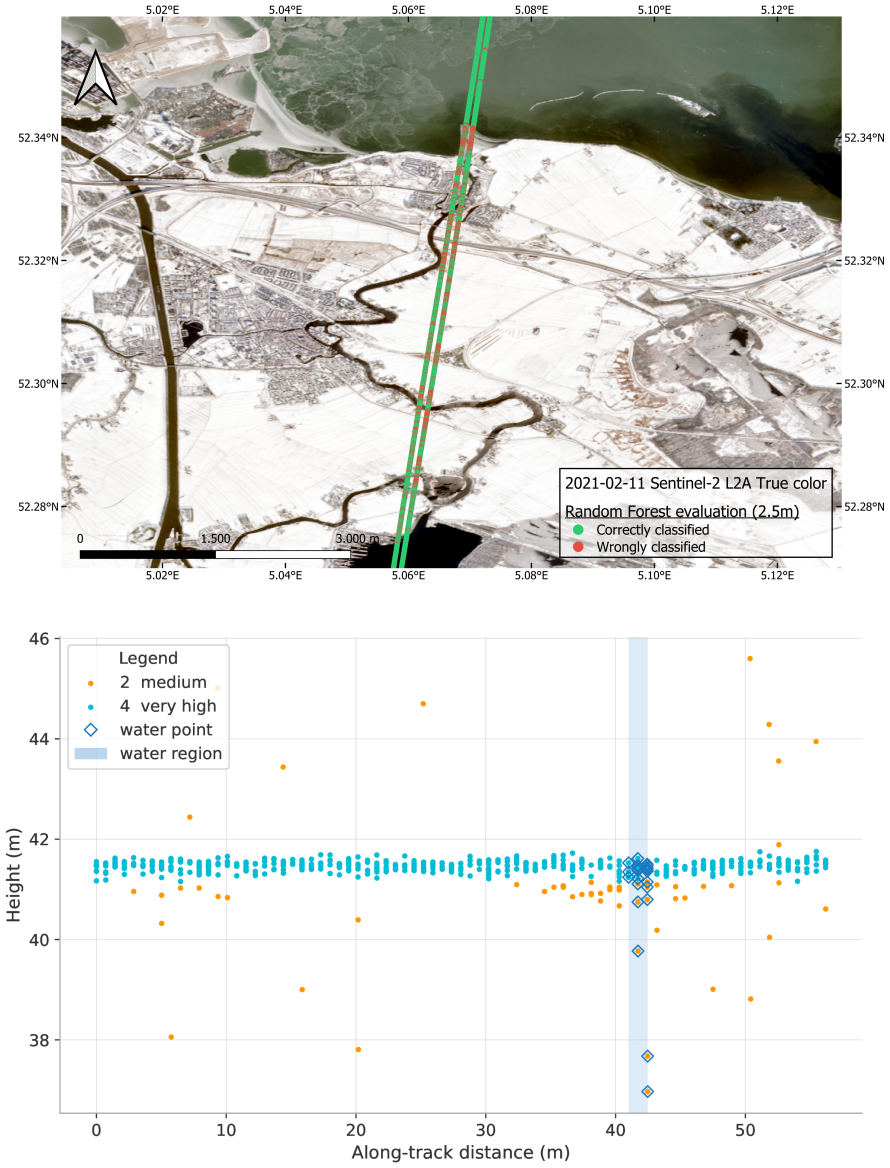
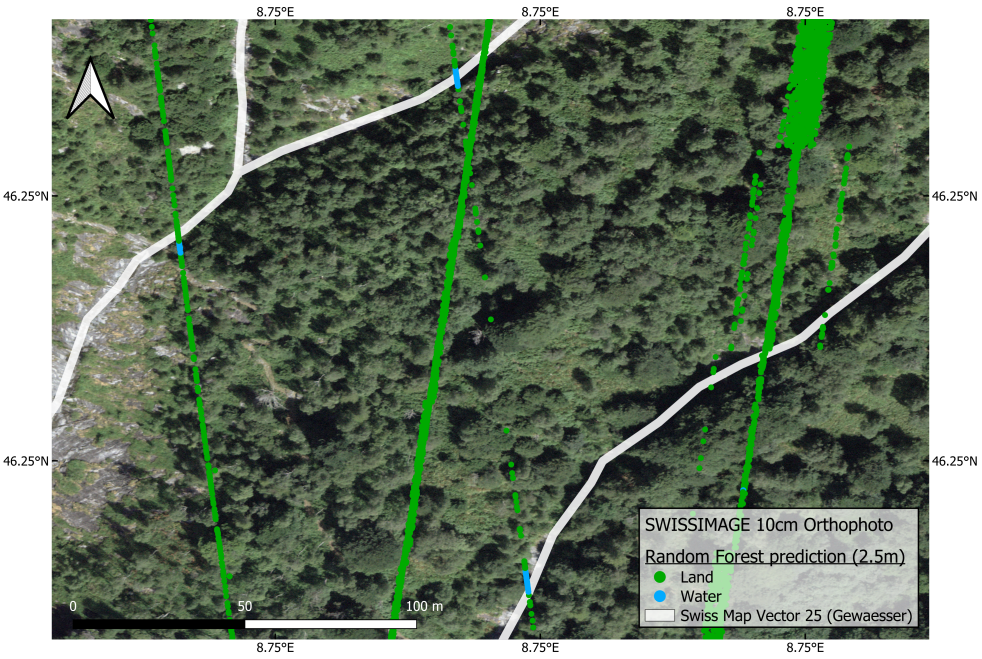


Figure 3.8.: The presence of snow gives a pattern similar to water, resulting in low confidence values between 0.5 and 0.7 where there is a lot of snow. Classification results by the model are inconsistent. Observed ICESat-2 data (beam gt1l/r) on 11-02-2021. Snow background data observed by sentinel-2 L2A on 11-02-2021.

3.5.4. Performance outside the Netherlands

Validating performance outside the training area is essential, as the Netherlands is unique in being extremely flat and having much sediment in the water. The model may therefore have learned characteristics specific to these conditions rather than general photon-water interactions. Quantitative evaluation of small water bodies is not feasible in most environments, however, as high-quality water masks often are unavailable. The results presented here are therefore qualitative, based on manual inspection of elevation profiles and comparison with available Sentinel-2 imagery, and should be considered an initial assessment of generalization potential rather than a comprehensive evaluation.

Swiss Alps. Figure 3.8 and Figure 3.9 show multiple classified tracks near Gordevio in the Swiss Alps. Reference data was derived from the Swiss Map Vector 25 Gewässer lin dataset, consisting of line segments extended with a buffer for visual clarity. These streams can be dry throughout the year and vary widely in width, meaning an absence of water detections along a reference line does not necessarily indicate a false negative; manual inspection of elevation profiles was therefore required to confirm whether water was present. The model successfully identifies small streams in this mountainous terrain: in the bottom image of Figure 3.8, two flat sections in the elevation profile correspond to confirmed water intersections, and manual validation across all beams in the scene confirmed that all points classified as water are true positives. Figure 3.9 shows a false negative, most likely caused by canopy noise above the water suppressing the photon-water signal. Larger elevation differences do not appear to significantly affect performance when the beam crosses water perpendicularly; performance on streams running parallel to a track on a steep slope could not be assessed, as no such example was found.



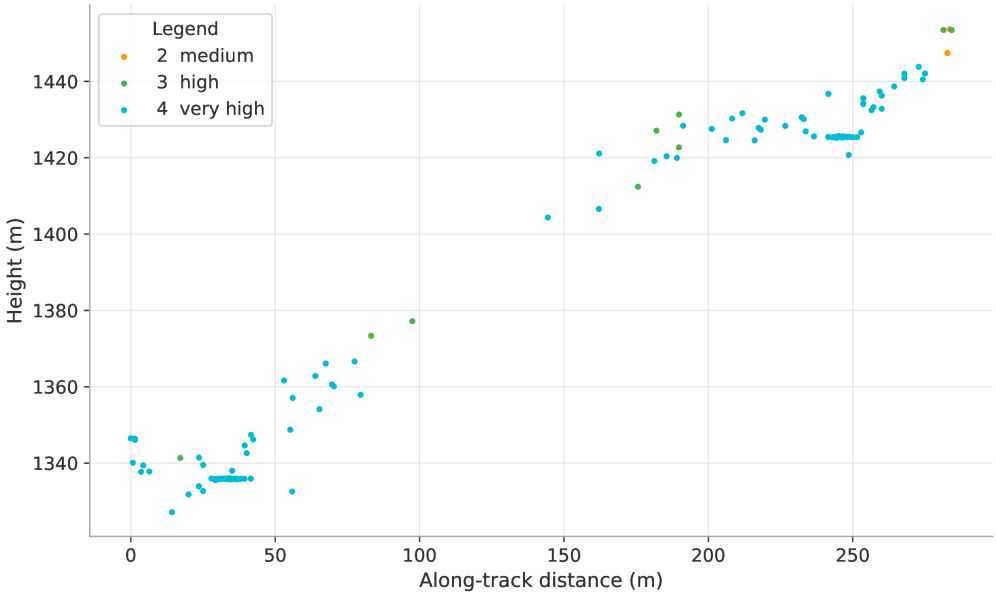
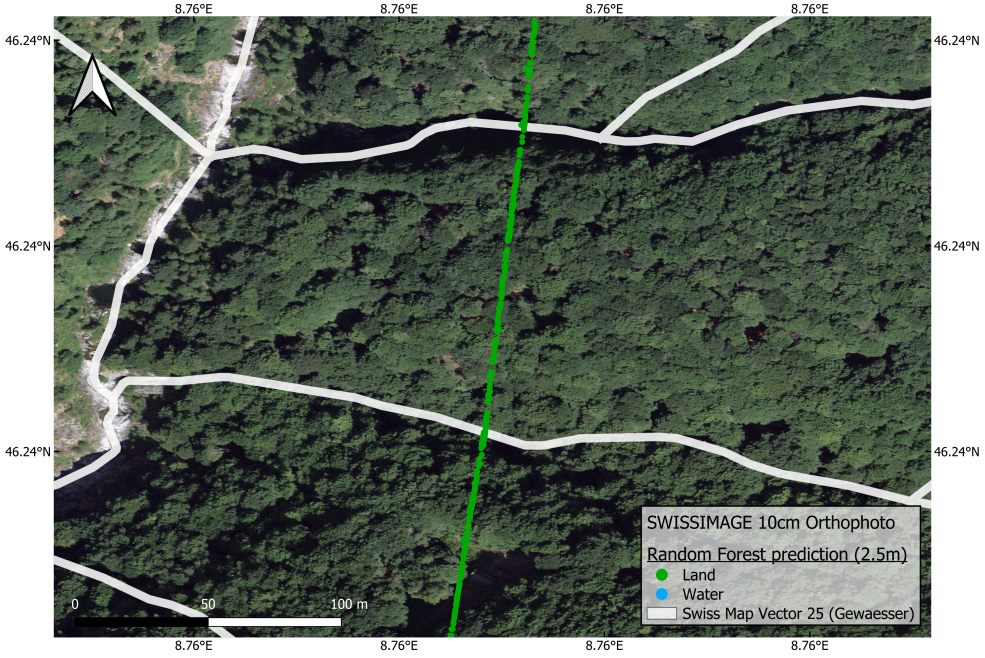


Figure 3.8.: Classification results near Gordevio in the Swiss Alps, showing how small streams can be identified when water is present. Background and ground truth data from Swisstopo.



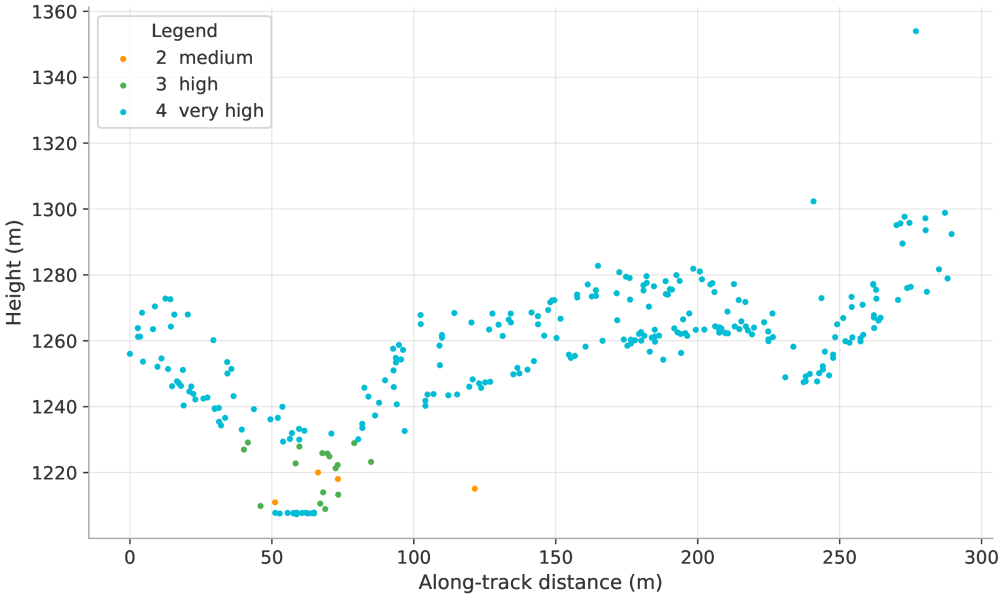


Figure 3.9.: Classification results near Gordevio in the Swiss Alps, showing how the model sometimes misses the presence of water due to overgrowth. Observed ICESat-2 data (beam gt2r) on 23-12-2019. Background and ground truth data from Swisstopo.

Greenland ice sheet. Figure 3.10 shows results for ICESat-2 granules intersecting the Sermeq Kujalleq area in Greenland during the melt season of 2019, following the area and period used in Datta and Wouters (2021). The results are ambiguous: though some points are classified as water, it is difficult to determine whether these reflect genuine detections of supraglacial meltwater or the same snow-induced misclassifications observed in Figure 3.8. The 10 m resolution of Sentinel-2 is furthermore insufficient to confirm whether water points outside the main melt lake correspond to small meltwater streams or false positives. For reliable performance in snow- and ice-covered environments, the training dataset would need to include a greater number of acquisitions under such conditions so that the model can learn to distinguish the photon return patterns of snow and ice from those of liquid water.

Mexican mangrove forest. Figure 3.11 shows results for multiple beams passing through the mangrove forest of Parque Nacional Lagunas de Chacahua in Mexico. Despite the noise introduced by the tree canopy, the model detects the presence of water beneath it, which is encouraging given that this environment was entirely absent from the training dataset. False negatives remain frequent, consistent with the general difficulty of detecting partially obscured water bodies.

Taken together, the results across the three environments suggest that the model has learned to identify the physical properties of water in general, rather than overfitting to the specific conditions of the Netherlands. Performance degrades predictably in conditions underrepresented in the training data, most notably snow and ice. Expanding the training dataset to include a greater variety of environments and acquisition conditions would be the most direct path to improving robustness at a global scale.

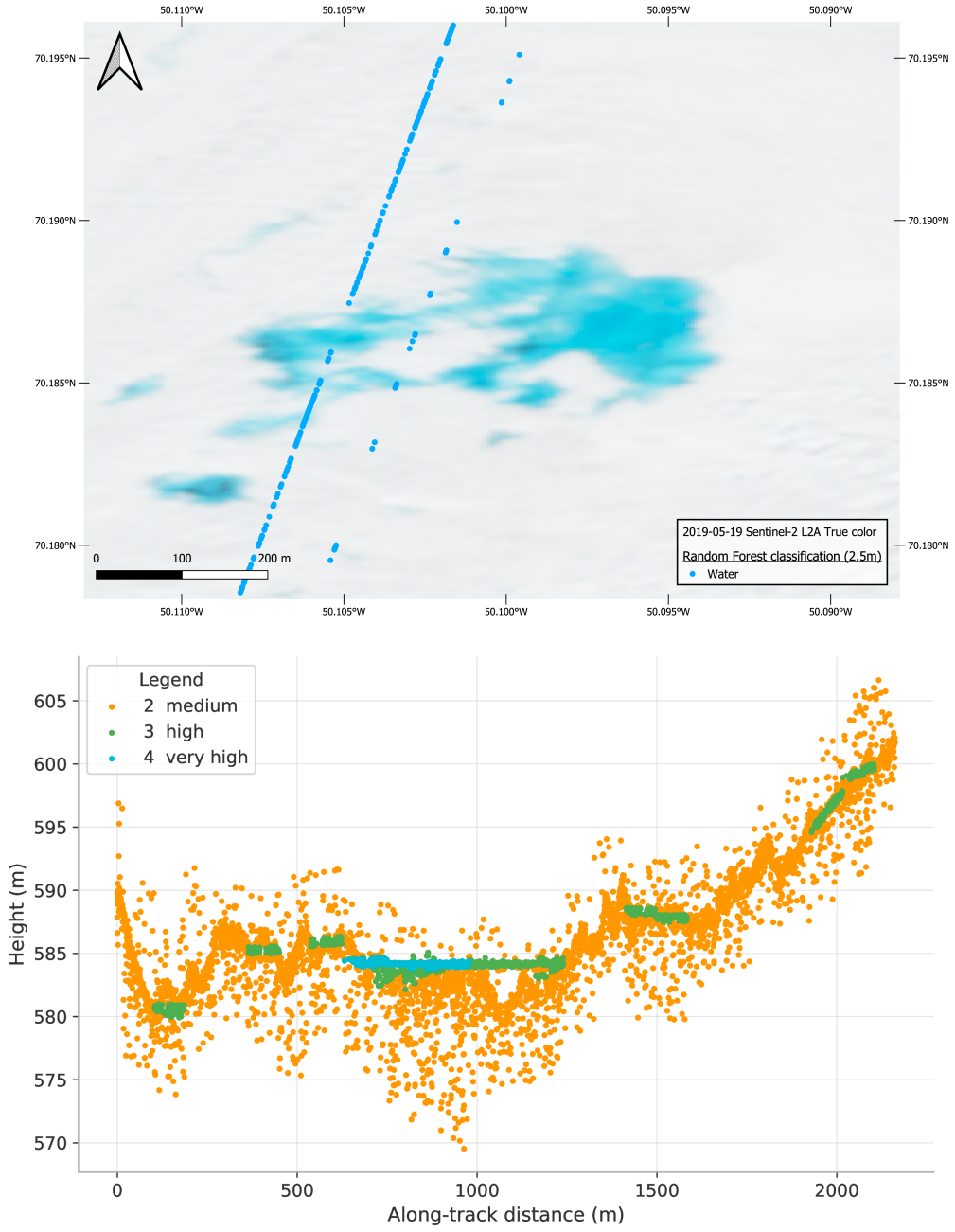


Figure 3.10.: Points over Greenland are being classified as water, but it is unclear if this is due to noise or the actual presence of small water streams not visible due to resolution constraints. Land points are not shown for clarity, but make up the majority of points for both beams (72.2% and 94.8%). Observed ICESat-2 data (beam gt1l/r) on 19-05-2019. Background data observed by sentinel-2 L2A on 19-05-2019.

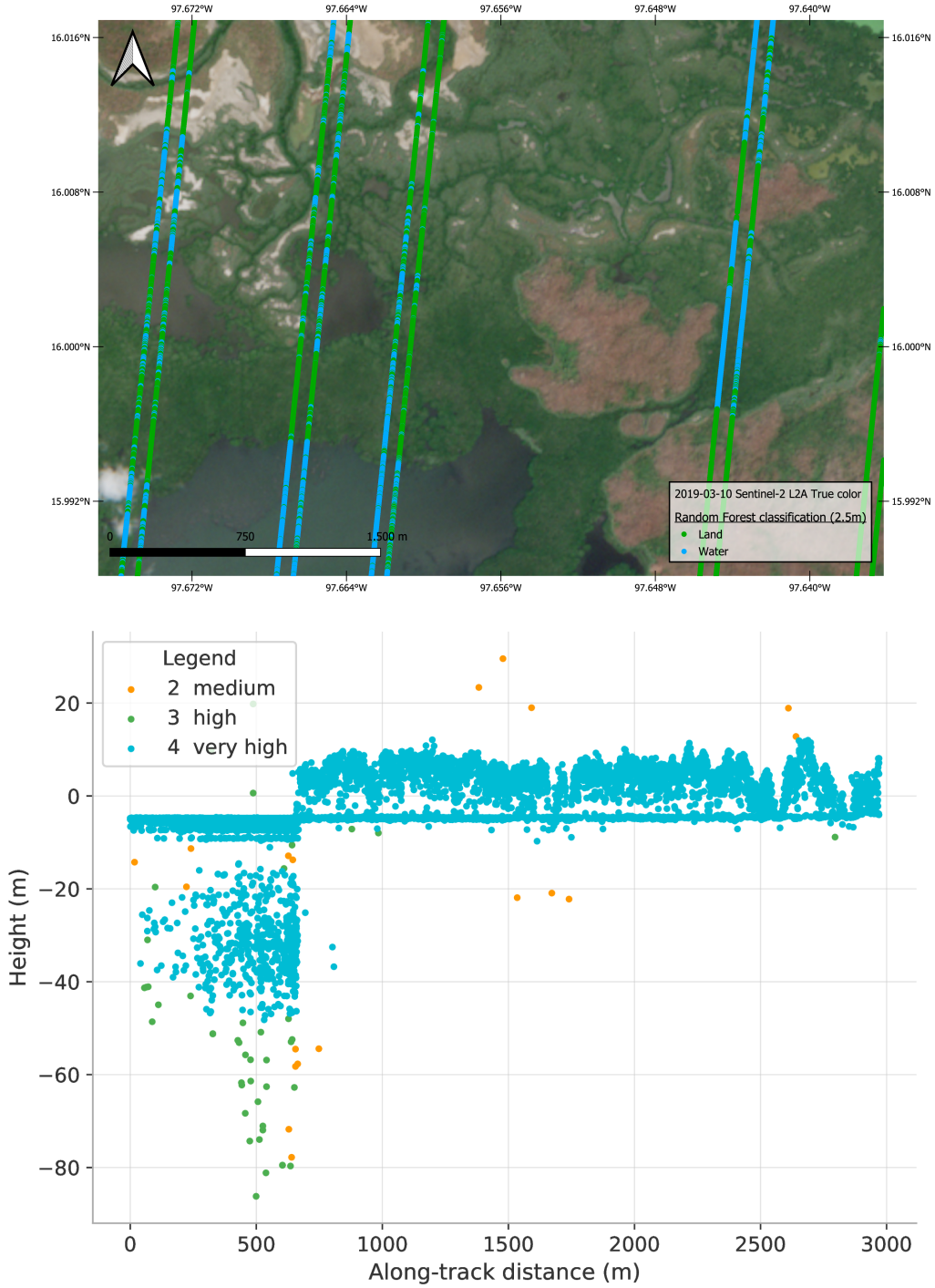


Figure 3.11.: Model results observed at the mangrove forest in Lagunas de Chacahua (Mexico) show that the model is capable of identifying water through trees. Plotted ICESat-2 data (*bottom*) from beam gt3r on 08-03-2019. Background data observed by sentinel-2 L2A on 10-03-2019

3.5.5. Model development

Impact of window width

To investigate the impact of window width, six RF models were trained in parallel, one for each window radius (see Table E.7). The overall model accuracy improves with larger window radii, but this trend is driven primarily by improved recall on larger water bodies. The 2.5 m radius achieved the best recall for small water bodies, making it the preferred window for this research.

Improvement experiments

Two strategies were identified to potentially improve performance on small water bodies. First, filtering windows to only consider those with a minimum of 5 photons could discard windows whose statistics are unreliable due to insufficient observations. And second, merging all the features of the 2.5 m, 10 m, and 25 m windows into a single feature vector. The model could then select from all the features and combine the strengths of the different scales. For example, water edges are more easily identified by smaller window widths, while larger windows can provide more stable statistics.

Filtering by a minimum of 5 points per window had little to no impact on performance, even in combination with other strategies, and produced no improvement in recall over the base case. Merging windows achieved a higher overall accuracy (83.3%) and precision (67.4%), indicating a balanced performance across all size classes, but performed worse for the water bodies < 25 m (see Table E.8).

3.6. Conclusions

In this paper, we presented a Random Forest (RF) classifier model specifically designed to identify photons of smaller water bodies (< 25 m) by assessing properties directly from ATL03 data rather than relying on pre-existing masks or auxiliary data. This approach enables global usability and detection in unmapped or dynamic environments.

Our feature analysis identified the number of high-confidence photons in a window to be the most important feature (MDI: 41.2%), followed by the sun angle, although its importance was skewed by outliers. Other relevant features included the number of peaks along the height axis, photon quality and uncertainty, and the fraction of photons within 0.2 m of the median elevation. The ringing effect beneath water surfaces seemed promising but was largely absent in the final selections, as it only appears at spatial scales larger than the largest window width considered and in non-turbulent conditions.

The final model was trained on 14 million points using a 2.5 m radius window with a $1:1$ ratio of small water segments (< 25 m) and land points and evaluated on 520 million ICESat observations in the Netherlands. For water-beam intersections longer than 6 m, recall exceeded 80%, reaching 87.1% for 10 - 25 m segments. However, performance degraded at both extremes: below 3 m, recall fell below 72%, dropping to 41.4% for 0 - 1 m segments due to the 0.7 m along-track sampling rate and windows wider than the bodies themselves. Similarly, recall declined to 74% for 250 - 500 m segments and 51% for intersections ≥ 500 m,

but for water identifying bodies of these sizes, optical imagery approaches are readily available.

Manual evaluation revealed four key challenges. First, outdated or spatially inaccurate ground truth polygons caused correct predictions to be recorded as misclassifications, affecting both reliability and training quality. Second, the model struggles at water edges, where lower confidence values reflect water signals extending 7 - 8 m beyond true boundaries. Third, snow produces return patterns resembling water, creating unreliable classifications; the training dataset lacks sufficient snow examples for the model to distinguish these cases. Finally, large open water body classification depends heavily on acquisition conditions; cloud cover, turbidity, surface clarity, and solar elevation all affect photon availability and, with it, performance on large water bodies.

Beyond the Netherlands, the model generalized well despite training exclusively on Dutch data, successfully identifying streams in the Swiss Alps and water beneath a mangrove canopy. Performance on the Greenland ice sheet remained ambiguous due to snow-water similarities, confirming that the model captures general water properties but degrades in underrepresented environments.

Future work should focus on clustering classified points together to form water segments, which can be used to derive water surface elevation. This would make the data more accessible and usable than with individually classified points. Given the model's underperformance on water bodies $\geq 250\,\text{m}$, combining it with a complementary model trained on larger segments through weighted predictions or merged outputs could improve overall performance without sacrificing small-body recall. Using a confidence threshold could also reduce edge noise and snow-related misclassifications, but careful tuning is needed when doing this to avoid excluding small bodies that often have weaker signals and lower confidences. Expanding the training dataset geographically and in size will also most likely increase performance and robustness outside the Netherlands.

4. Conclusion and discussion

This chapter presents a more extensive discussion of the results than presented in the Scientific Article. It starts with a Research overview and contributions, summarizing the research and findings of the paper and reflecting on its main contributions and insights. Afterwards, the Limitations will be addressed, discussing where the final model and the research methodology fall short. Based on this, seven recommendations for future work are formulated in the Recommendations and future work section to address these challenges.

4.1. Research overview and contributions

4.1.1. The core problem

The goal of this thesis was to create a model that can predict the presence of inland water bodies smaller than 25 m in width. This addresses current limitations in the collection of water data, which is often expensive, lacks spatial resolution, or is limited by natural constraints like the presence of overgrowth. Previous studies have started using NASA's Ice, Cloud and Land Elevation Satellite 2 (ICESat-2) elevation data provided by a high-quality onboard LiDAR instrument. NASA provides two official Inland Water products (ATL13 and ATL22) that provide information on the surface elevation of inland water. These products are, however, limited by the quality and coverage of pre-defined water body polygons they rely on to select which photons are over water. The products only contain data on lakes and reservoirs larger than 0.1 km^2 and rivers wider than 50 m. Besides problems with size, the polygons are also static, meaning they do not adapt to seasonal flooding, drought, or gradual changes in water extent. In data-sparse regions like the global south or remote Arctic areas, that means water polygons often do not even exist at all.

4.1.2. The approach

This thesis approaches water detection differently than the ICESat-2 data products. Instead of asking *"Is this photon inside a known water polygon?"*, it asks *"Does this photon look like it bounced off water?"*. The research resulted in a Random Forest (RF) classifier that uses features that encode physical properties of the photon return signals within a small window around each data point. The model solely relies on ICESat-2 data; no polygons, masks, or other satellites are required to detect water. This approach has a few strong benefits:

- It allows for water detection and water elevation measurements simultaneously. Previously, it was required to already know where water was before its elevation could be measured, resulting in polygons that often did not properly represent real observations.

- It extends the capability of using ICESat-2 for small, unmapped, or dynamically changing water bodies. These water bodies can be important for hydrological research, but are currently invisible in the data products.
- It is self-contained, relying only on ATL03 data. This data is globally and freely available, making it usable by everyone everywhere.

4.1.3. The features

Previous work by (Ma et al., 2024b) and (Datta and Wouters, 2021) also used ICESat-2 photon properties for water detection, but both were limited to large water bodies in specific study areas. It was therefore unclear whether these methods would work in different environments and for small water bodies. Scientists of the ICESat-2 team also identified a *ringing* effect beneath water surfaces at a 0.5 m spacing, caused by the dead time of the sensors (Neuenschwander and Magruder, 2019). Dead time is the time it takes for a sensor, after observing a photon, to be able to detect the next. Besides the dead time rings, scientists also observed a ringing effect at around 2.3 m and 4.2 m beneath the surface over flat areas, which often include water bodies (Neumann et al., 2025a).

To detect a water signal in the data, a total of 49 features (see Appendix C) were created based on a window around individual photons that try to encode the (i) density of photons, (ii) presence of bottom reflectance, (iii) presence of afterpulses, (iv) low slope, and (v) low standard deviation of photon heights. To rank and select the best features, they were first decorrelated by applying hierarchical clustering. Only the best-performing feature in each cluster, ranked using MDI score, remained to be used. The features were then given to a random forest, which ranked them using both MDI and permutation importance scores. The 15 best features, having high values for both metrics, were then selected.

In the final model, the number of high confidence points proved to be the best predicting feature, accounting for 41.2% of the total importance. Although the sun angle is the second most important feature according to the MDI, this value is highly skewed by a few strong outliers, while the median performs badly. It is not understood why this feature performs well. It could be a subtle data artefact, but no indications were found that this is the case. Other important features are the number of peaks along the height axis, the quality and uncertainty of the photon, and the fraction of photon within 0.2 m of the median height.

While the specular ringing effect at dead-time spacing beneath water surfaces theoretically seemed promising to detect water with, this thesis showed that it is not a reliable feature. The specular return lines only become clearly visible in the data on larger spatial scales. Individual photons at dead-time depth are not prevalent enough to create a clear signal, even when working with a 25 m radius window. Another reason why the specular ringing effect features underperformed is that return lines only show up at still standing water. Most water bodies that are large enough for this pattern to appear have too much turbidity.

The ringing effect beneath all flat surfaces also proved to be a non-viable feature. Because the Netherlands is a flat country, it was expected that the model would pick up this pattern, but it did not. This, based on the large amount of data assessed, is caused by it either not being prevalent enough, not providing a clear enough distinction between water and land, or not showing up for windows smaller than 25 m.

4.1.4. The model

There are many different classification models available that can be used to predict the presence of water. In this thesis, a RF classifier was selected because of its ability to identify non-linear relationships, its robustness to noise, its non-parametric nature, and its ability to handle both numerical and categorical data. The data the model was trained and evaluated on includes 533 million points with a confidence of at least 2 collected in the Netherlands. The Netherlands was used because of its high-quality water polygons.

4.1.5. The results

The final model uses a 2.5 m radius window and was trained exclusively on photons intersecting small water segments (< 25 m) and land points in a $1:1$ ratio, using 14 million points in total. This approach significantly improved recall on small water bodies compared to training on the full distribution of water segment sizes. The downside is that it also worsens performance on large water bodies. Filtering windows by a minimum point count and considering features from multiple window sizes (2.5 m, 10 m, and 25 m) for one RF had little effect on the results. Merging the different window sizes could provide the model with contextual information on different scales, but this also introduces new types of noise that the model needs to deal with. Though the recall for small water bodies (< 10 m) did slightly improve, the performance on 10 - 25,m water bodies slightly decreases, resulting in the same overall performance. Window size selection proved to be a balancing act: smaller window sizes contain less data to work with, making decisions more uncertain, but they can provide a higher resolution. For the detection of small water bodies, a radius of 2.5 m proved to be balanced best, while overall water detection performs better when using larger window radii.

The model was evaluated on approximately 520 million ICESat-2 observations across the Netherlands. For water-beam intersections longer than 6 m, recall exceeds 80.0%, rising to 87.1% for segments between 10 m and 25 m. Below 3 m, where the 0.7 m along-track sampling rate limits the number of photons available for classification, performance drops to under 72.0% with $0 - 1$ m water segments having a recall of just 41.4%. Performance also drops for larger water segments (≥ 250 m), with a recall of 74.0% for segments up to 500 m and 51.0% for intersections ≥ 500 m.

Four limitations of the final model have been identified after manually inspecting the results:

1. **Faults in ground truth:** Water polygons that are outdated or do not accurately reflect the true water extent at the time of the ICESat-2 overpass cause correct model predictions to be recorded as errors. This might also affect training, where bits of noise can influence results.

2. **Uncertainty near edges:** Near the boundary of water bodies, the model has a lower confidence in the classification of points. This is caused by a partial water signal being present up to 7 m (half the beam footprint diameter) away from the true edge. The strong reflectance of water produces photon returns, even when the beam just partly overlaps a water body. Because BGT polygons do not always align precisely with actual water edges, it is sometimes difficult to separate genuine misclassifications from ground-truth inaccuracies.

3. **Confusion during snow cover:** The model's primary feature encodes the high reflectance of water, making it sensitive to some other specular surfaces. Greenhouses introduce noisy photon returns but do not cause systematic misclassifications, as the noise pattern differs enough from water. The same is not true for snow cover, however. The photon return pattern of snow could not be properly distinguished from water, leading to many uncertain or incorrect classifications. Because the training dataset contains very few observations from snowy days in the Netherlands, the model has not learned to distinguish the two surfaces.

4. **Large water body performance depends on conditions:** Performance of the model for large water bodies ($\geq$ 500 m) is low, mainly because it is highly dependent on conditions. Depending on the day, the model can either correctly classify all/most points or none/few. This indicates that the performance on large water bodies is not just a consequence of the training strategy, but also depends on the conditions. When fewer photons are returned, for example, due to cloud cover or turbidity, not enough signal is present for the model to correctly classify points.

Besides the Netherlands, the model was also manually validated in three environments around the world. In the Swiss Alps, the model successfully identified narrow mountain streams even in the presence of surrounding trees. In a mangrove forest in Mexico, the model detected water through the dense forest canopy. Both results suggest that the model has learned general photon-water interaction properties rather than Dutch-specific patterns. Performance on the Greenland ice sheet was less conclusive, as snow produces strong return patterns with lots of noise that can sometimes resemble water, leading to uncertain classifications that could not be properly assessed due to the limited resolution of satellite imagery used for validation.

4.2. Limitations

The model presented in this thesis has several limitations that should be acknowledged.

- **Problems with ground truth data quality:** The model was exclusively trained and evaluated using the polygons in the BGT waterdeel dataset. This is a semi-static dataset containing many polygons stacked on top of each other, representing the water extent at different times. These polygons change only every few years and do not reflect seasonal changes in water extent, causing a fraction of the ground truth labels to be incorrect at the time of observation. The problem is worsened by the fact that the approach in this thesis does not differentiate polygons based on time and instead merges all of them. This introduces noise both in the training and evaluation, which is difficult to quantify.

- **Limited geographic scope:** All quantitative evaluations were performed only on data from the Netherlands. The limited validation in Switzerland, Greenland, and Mexico indicates promising results for general usability, but the assessment is still limited. A more extensive evaluation is needed to properly test performance in other environments.

- **Limited volume and diversity of training data:** Due to memory constraints, only 2.6% of available ICESat-2 data was used for training. The training dataset also consists solely of points in the Netherlands, which limits robustness in environments that differ

from the Dutch Landscape. These are demonstrated by the model's struggles on the Greenland ice sheet.

- **Point-level classification:** The model currently classifies individual points rather than contiguous water segments. Individual point predictions cannot directly be used to determine water surface elevations, because they are too noisy alone. The classified points first need to be processed further to create water segments.
- **Bad performance on large water bodies:** By using only segments smaller than 25 m, the recall significantly drops for water bodies $\geq$ 250 m. While this was a design choice, it means that the model currently can not be used as an all-around inland water classifier.

4.3. Recommendations and future work

Based on the conclusions and limitations of this research, several recommendations for future work can be formulated. These are grouped into two categories: recommendations that take the current model as a starting point and expand its usability, and recommendations that target the training methodology itself to produce a better-performing model.

4.3.1. Extending the current model

- **Combining classified points into water segments:** The current model only classifies individual photons, which cannot be reliably used to derive surface elevation on their own. Future work should focus on clustering individually classified points into line segments using algorithms like Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN), which is capable of identifying clusters of varying densities. These line segments would make the output directly usable for people to do water elevation assessments, which is a primary motivation for inland water monitoring with ICESat-2.
- **Introducing a confidence threshold:** Restricting water classifications only to points above a confidence threshold can trade the recall for precision. Introducing this restriction might reduce noise near water edges and potentially reduce the number of misclassifications when observing snow cover. The threshold should be chosen carefully, however, as setting it too high risks excluding genuine water points, particularly for small water bodies where the signal is weaker. Extensive testing is required to identify a fitting threshold and to assess the influence on water identification.
- **Combining classification models:** The final RF was trained on small water segments, resulting in a model that can predict small water bodies. It, however, underperforms for water bodies $\geq$ 250 m. This thesis showed that larger windows can reliably classify specifically these points. An approach that combines the strengths of a model trained to identify smaller bodies and a model trained to identify larger bodies, using a weighted prediction, a combined confidence threshold, or merging clusters each model identifies, could improve the overall performance without reducing the recall on small water bodies.

4.3.2. Improving the training methodology

- **Expanding the training dataset:** Only 2.6% of all available ICESat-2 data in the Netherlands was used for training due to memory constraints. Increasing the training sample size is likely the easiest way to improve model performance, and should be prioritized in future work.
- **Removing low confidence photons:** The model currently uses a minimum confidence of 2 for the selection of photons. By increasing the confidence threshold, less noise will be present. This might improve performance, but can also negatively affect features that identify subsurface peaks. Future work could test how model performance changes depending on confidence thresholds.
- **Improving ground truth accuracy:** The BGT waterdeel dataset contains many different polygons representing the same water body at different epochs. Because the water polygons change over time, the accuracy of the ground truth data can be improved by using the polygon temporally closest to ICESat-2 observation. Doing this will reduce noise in both the training and evaluation data.
- **Increasing geographic diversity:** Including training data from outside the Netherlands will most likely improve performance robustness in environments that differ from the Dutch landscape. This, however, will require high-quality water polygon data for small water bodies around the world, which is difficult to obtain. A possible approach would be to use OpenStreetMap (OSM) line geometries, which are available globally. These line geometries can be transformed into polygons using a buffer up to 7 m. Water signal extends up to 7 m beyond the true water edge due to the high reflectance of water, which can be used as a proxy for water extent. This would open up a large portion of the globe as potential training regions. But, ground truth data should still be selected carefully to prevent introducing label noise caused by seasonal water bodies, such as the Swiss streams that are dry most of the year.
- **Refining feature selection for small water bodies:** The current feature selection process relies only on 700,000 points. Repeating the feature selection using a larger dataset can result in a feature set fitted to detect small water bodies.

A. Declaration of AI/LLM usage

No word in this master's thesis has been touched or thought of by a Large Language Model (LLM). No LLMs have been used to write abstracts, structure texts, do research for me, generate outlines, or do anything else related to the content presented. All ideas were thought of by me in collaboration with my supervisors or in discussions with fellow students. LLMs were, however, occasionally used in assisting with creating LaTeX tables and programming. Specifically for creating the QGIS plugins presented in Appendix F. At the recommendation of my supervisors, I used Claude.ai, as they had positive experiences with its programming capabilities. After creating the basic functionality I required for the plugins in Python, I asked Claude to wrap it in the plugin functionality, which it was highly capable of. Claude was also used to sometimes solve errors in the code that I myself was not able to identify or solve. All the usage was in full adherence to academic integrity.

B. Reproducibility self-assessment

As a large proponent of open science, it is highly important to me that all the work presented in this thesis is reproducible and accessible. All the data used in this thesis, therefore, comes from public and open sources that are available to everyone. The two datasets we used were:

1. **BGT waterdeel**: polygons of all waterbodies in the Netherlands. These include rivers, canals, streams, ditches, lakes, ponds and fens. All definitions are available at Geonovum and data can be downloaded via PDOK (accessed 16/02/2026).
2. **ATL03 Geolocated Photons**: contains height above the WGS 84 ellipsoid (ITRF2014 reference frame), latitude, longitude, and time for all photons observed by the ATLAS instrument aboard the ICESat-2 satellite (Neumann et al., 2025b). Data can be downloaded via Earthdata (data used up to 19/11/2025).

All code used in this master's thesis is also publicly available at `https://github.com/HeikoRotteveel` (Rotteveel, 2026a). There are three relevant repositories on this GitHub page. The first is the main data processing and analysis code. The second is the ICESat-2 profile viewer QGIS plugin presented in Section F.1 (Rotteveel, 2026b). Lastly is the machine learning classification visualizer presented in Section F.2 (Rotteveel, 2026c). All repositories include README files with explanations of how to install and use the content.

I rate the reproducibility of this thesis as **High**.

C. List of considered features

This appendix lists all features considered during model development, grouped by inclusion status. Each feature is identified by its source (ATL03, Calculated), a short name, and a description. Features excluded from the model also include a short description of the reason they were excluded. All calculated features are derived within a sliding window at six radii (`1 m, 2.5 m, 5 m, 10 m, 20 m, and 25 m`). More information on the precise definitions of ATL03 features can be found in Neumann et al. (2025b).

Table C.1.: Overview of all features considered for random forest model

#	Feature	Description	Source	Status
1	`quality`	Photon quality flag: 0=nominal, 1=afterpulse, 2=impulse response, 3=TEP	ATL03	Included
2	`uncertainty`	Estimated vertical uncertainty of the photon height	ATL03	Included
3	`strong_beam`	Flag indicating whether the photon belongs to a strong (1) or weak (0) beam	ATL03	Included
4	`sun_angle`	Solar elevation angle at the bounce point	ATL03	Included
5	`detector_id`	Which of the 6 ATLAS laser spots on the ground a given beam group corresponds to	ATL03	Included
6	`n_points`	Total number of photons in window	Calculated	Included
7	`conf_2`	Number of photons with confidence level 2 in window	Calculated	Included
8	`conf_3`	Number of photons with confidence level 3 in window	Calculated	Included
9	`conf_4`	Number of photons with confidence level 4 in window	Calculated	Included
10	`h_std`	Standard deviation of photon height (vertical spread)	Calculated	Included
11	`h_range`	Height range in window (total vertical extent)	Calculated	Included
12	`h_iqr`	Interquartile range of photon height (robust spread)	Calculated	Included
13	`h_skew`	Skewness of photon height distribution (asymmetry)	Calculated	Included
14	`h_kurt`	Kurtosis of photon height distribution (peakednes)	Calculated	Included
15	`frac_01m`	Fraction of photons within 0.1 m of the median height (flatness indicator)	Calculated	Included
16	`frac_02m`	Fraction of photons within 0.2 m of the median height (flatness indicator)	Calculated	Included
17	`h_ref_std`	Standard deviation of the height reference within the window	Calculated	Included
18	`kde_peaks_h`	Number of histogram peaks along the height axis	Calculated	Included
19	`peak_dist`	Mean spacing between histogram peaks	Calculated	Included
20	`fwhm`	Full width at half maximum of the dominant height peak	Calculated	Included

(continued on next page)

(continued from previous page)

#	Feature	Description	Source	Status
21	`prominence`	Relative height of dominant peak above surrounding signal	Calculated	Included
22	`slope`	Along-track surface tilt	Calculated	Included
23	`residual`	Standard deviation of photon heights from the fitted linear surface	Calculated	Included
24	`spacing_mean`	Mean along-track distance between consecutive photons	Calculated	Included
25	`spacing_median`	Median along-track distance between consecutive photons	Calculated	Included
26	`spacing_std`	Standard deviation of along-track photon spacing	Calculated	Included
27	`ap_23_count`	Number of photons at 2.3 m below surface ($\pm$0.3 m)	Calculated	Included
28	`ap_42_count`	Number of photons at 4.2 m below surface ($\pm$0.3 m)	Calculated	Included
29	`ap_23_ratio`	Ratio of photons at 2.3 m depth to total count ($\pm$0.3 m)	Calculated	Included
30	`ap_42_ratio`	Ratio of photons at 4.2 m depth to total count ($\pm$0.3 m)	Calculated	Included
31	`ap_23_present`	Binary flag: at least one photon detected at 2.3 m afterpulse depth	Calculated	Included
32	`ap_42_present`	Binary flag: at least one photon detected at 4.2 m afterpulse depth	Calculated	Included
33	`ap_depth_23_mean`	Mean depth of photons in the 2.3 m afterpulse	Calculated	Included
34	`ap_depth_42_mean`	Mean depth of photons in the 4.2 m afterpulse	Calculated	Included
35	`dt_return_count`	Number of photons at dead-time-like spacing below surface (0.3–0.7 m)	Calculated	Included
36	`dt_spacing_mean`	Mean depth gap of photons at dead-time-like spacing (0.3–0.7 m)	Calculated	Included
37	`dt_spacing_std`	Std. dev. of depth gaps at dead-time-like spacing (0.3–0.7 m)	Calculated	Included
38	`dt_regularity`	Regularity of the dead-time return pattern (coefficient of variation of spacings)	Calculated	Included
39	`dt_present`	Binary flag: dead-time return count ≥ 2	Calculated	Included
40	`n_subsurface_peaks`	Number of peaks detected $>$0.5 m below the surface peak	Calculated	Included
41	`subsurface_depth_1`	Depth of the first (strongest) detected subsurface peak	Calculated	Included
42	`subsurface_depth_2`	Depth of the second detected subsurface peak	Calculated	Included
43	`bimodal_score`	Relative height of the subsurface peak with respect to the surface peak (bimodality indicator)	Calculated	Included
44	`dh_mean_left`	Change in `h_mean` relative to the previous window	Calculated	Included
45	`dh_mean_right`	Change in `h_mean` relative to the next window	Calculated	Included
46	`dh_std_left`	Change in `h_std` relative to the previous window	Calculated	Included
47	`dh_std_right`	Change in `h_std` relative to the next window	Calculated	Included
48	`dn_points_left`	Change in `n_points` relative to the previous window	Calculated	Included
49	`dn_points_right`	Change in `n_points` relative to the next window	Calculated	Included

Table C.2.: Features not considered during model development (metadata, identifiers, or dependent variables).

#	Feature	Description	Source	Reason
50	`confidence`	Signal confidence: 0=noise, 2=low, 3=med, 4=high, −1=not applicable	ATL03	Redundant
51	`longitude`	Estimated photon longitude	ATL03	Not a physical predictor
52	`latitude`	Estimated photon latitude	ATL03	Not a physical predictor
53	`height`	Photon height relative to WGS-84 ellipsoid, including geophysical corrections	ATL03	Direct input to calculated features; not fed to model
54	`datetime`	Acquisition timestamp	ATL03	Not a physical predictor
55	`segment`	Along-track segment identifier	ATL03	Identifier only
56	`track`	Reference ground track number	ATL03	Identifier only
57	`height_reference`	Geoid/ellipsoid offset at the photon location	ATL03	Direct input to `h_ref_std`; not fed to model
58	`water`	Intersection with BGT water polygon: 0=no water, 1=water	BGT Waterdeel	Dependent variable
59	`water_length_m`	Length of intersection between water polygon and along-track beam	Calculated	Metadata only
60	`h_mean`	Mean photon height in window (WGS-84/ITRF2014)	Calculated	Direct input to gradient features; not fed to model
61	`h_median`	Median photon height in window (WGS-84/ITRF2014)	Calculated	Direct input to flatness features; not fed to model
62	`surface_peak`	Most dominant surface elevation in the window	Calculated	Direct input to afterpulse features; not fed to model

D. Feature calculation methods

This appendix describes the formulas behind the computed features. Basic calculations like calculating the `mean` or `standard deviation` are not discussed, but can be found in the source code on the author's GitHub page. All window-based features are computed within a sliding window at six radii (1 m, 2,5 m, 5 m, 10 m, 20 m, and 25 m). A window centered on photon i collects all photons j for $|x_j - x_i| \leq r$, where x is the along-track distance in meters and r is the window radius. Windows with fewer than five photons return `missing` for all calculated features. This is to prevent noise, like weird standard deviations when there are just 3 points in a window, from slipping into the data. ATL03 features (`quality`, `uncertainty`, `confidence`, `strong_beam`, `sun_angle`, `detector_id`) are read directly from the input data without modification.

D.1. Flatness fractions

These features capture how photons cluster around the median height $\tilde{h}$ of all points in the window. When there is open water, the surface in the window should be nearly horizontal. Both fractions in that case should approach 1. When there is land or vegetation present, the distribution should be much broader. To calculate the fraction of photons (`frac_01m`) within 0.1 m of the median we use:

$$\texttt{frac\_01m} = \frac{1}{n} \left| \{ j \in W : |h_j - \tilde{h}| < 0.1 \} \right| \tag{D.1}$$

where W denotes the set of photon indices in the window, n describes the total number of photons and h_j is the height of photon j. `frac_02m` uses the same calculation but with a value of 0.2 instead of 0.1.

D.2. Histogram peak features

These features are derived from a 50-bin histogram of the photon heights in the window. The calculations require multiple points for the features to be meaningful. We therefore decided that windows with fewer than 10 points return `missing`. The features are most informative at the 20-25 m scale, where photon counts are large enough to provide distinct peaks. The histogram uses equal-width bins spanning $[\min(h), \max(h)]$. A bin b is a peak if its count strictly exceeds both neighbors: $\text{count}(b) > \text{count}(b-1)$ and $\text{count}(b) > \text{count}(b+1)$.

The idea of these features is to detect which points represent the water surface. This is especially useful when, for example, there is overgrowth present. By detecting multiple peaks, it might also be possible to identify subsurface returns, which can be a great indicator of the presence of water.

`kde_peaks_h` is calculated by counting the number of histogram peaks detected in the window. `peak_dist` then determines the mean distance between consecutive peak centers (in meters), if more than two peaks are detected. The `fwhm` (full width at half maximum) of the dominant peak (the peak with the highest count) is the width of the region around the dominant peak where the number of points is more than half its maximum count (or `missing` if the half-maximum crossing points cannot be found). The `prominence` of the dominant peak is defined as the peak count minus the minimum bin count, normalized by the total number of photons:

$$\texttt{prominence} = \frac{\text{count}_{\max} - \text{count}_{\min}}{n} \tag{D.2}$$

Subsurface peak features

Using the same 50-bin histogram as above, it is also possible to identify more localized peaks. Peaks located more than 0.5 m below the surface peak h_s are identified as subsurface peaks. A bin is a local peak if its count strictly exceeds both its neighbors. We can use this to identify the number of peaks detected below the surface (`n_subsurface_peaks`). `Subsurface_depth_1` and `subsurface_depth_2` are then the depth below h_s of the first (strongest) and second subsurface peaks, ranked by bin count. If fewer than one or two subsurface peaks exist, respectively, the values are `missing`. Lastly, we calculate the ratio of the dominant subsurface peak count to the surface peak count:

$$\texttt{bimodal\_score} = \frac{\text{count}(b_{\text{sub},1})}{\text{count}(b^*)} \tag{D.3}$$

A value approaching 1 indicates a subsurface return nearly as strong as the surface return. When no subsurface peak is detected, or the surface peak bin is empty, then the feature is `missing`.

D.3. Along-track features

The along-track features can help to provide insights into the density of points and the slope of the points. We can define the along-track distances between consecutive photons by $\Delta x_j = x_{j+1} - x_j$, for $j = 1, \ldots, n-1$. The `spacing_mean`, `spacing_median` and `spacing_std` are calculated based on the resulting values. At least two photons are required for these features.

To calculate the slope of the points, we fitted a least-squares line to the photon heights h_j as a function of along-track position x_j using the following equation

$$\texttt{slope} = \frac{\sum_j (x_j - \bar{x})(h_j - \bar{h})}{\sum_j (x_j - \bar{x})^2}. \tag{D.4}$$

At least three photons with non-zero variance in x are required for the slope to be calculated. Finally, we also calculated the `residual` (Standard deviation of the deviations from the fitted line) using

$$\texttt{residual} = \text{std}\big(h_j - \big(\bar{h} + \texttt{slope} \cdot (x_j - \bar{x})\big)\big) \tag{D.5}$$

Gradient features

To help detect water edges and to highlight the contrast between water and land, we also computed the difference between two neighboring windows along-track. These values were calculated in a second pass over the full data after all other features were calculated. In order for this to work, the data must be ordered along the track direction. Let f_i be the value of a window statistic (e.g. `h_mean`) at photon i.

$$\texttt{dh\_mean\_left}_i = f_i^{h_{\text{mean}}} - f_{i-1}^{h_{\text{mean}}}, \qquad \texttt{dh\_mean\_right}_i = f_{i+1}^{h_{\text{mean}}} - f_i^{h_{\text{mean}}} \tag{D.6}$$

The same calculation can also be done for the `h_std` and `n_points`. The first and last photons in each file receive `missing` for all gradient features. A gradient is also `missing` whenever the underlying statistic is `missing` for either of the two photons involved.

D.4. Afterpulse and dead-time features

Neumann et al. (2025a) highlighted multiple return patterns in the ATL03 photon data that can be used to detect the presence of water. The LiDAR instrument aboard ICESat-2 (ATLAS) is briefly insensitive to photons after registering one. Photons that follow shortly after the first can therefore not be detected until the sensor has reset. This short period is called dead time and results in after pulses at approximately 0.5 m below the first return. This creates a "ringing" effect, especially over flat water where the returning laser will be (nearly) specular. Over relatively flat surfaces, even without water, there can often also be multiple returns at around 2.3 m and 4.2 m below the true surface. These "afterpulses" are most likely the result of small after-pulses in either the ATLAS transmitted laser pulse, or a small amount of electronic noise following the arrival of the primary surface return (Neumann et al., 2025a). Both these patterns become clearer when there are more points available, making them most informative at the 25 m window scale.

The first step in detecting these below-surface returns is to detect the surface. This point is calculated using the same 50-bin histogram of photon heights approach as in Section D.2. The `surface_peak` h_s is the center of the bin with the highest count, where $w = (h_{\max} - h_{\min})/50$ is the bin width:

$$h_s = h_{\min} + (b^* - 0.5) \cdot w, \quad b^* = \arg\max_b \text{count}(b) \tag{D.7}$$

Dead-time features

All photons more than 0.1 m below the surface we just calculated are collected as subsurface photons k with depths d_{sub}. Their depths are sorted, and consecutive spacings $\delta_k = d_{k+1} - d_k$ are computed. A spacing is classified as a dead-time spacing if $0.3 < \delta_k < 0.7$ m. We used these wider margins around the 0.5 m to capture as many of the subsurface points as possible. Some basic descriptive features are then calculated (`dt_present`, `dt_return_count`, `dt_spacing_mean`, `dt_spacing_std`). The `dt_regularity` is the coefficient of variation of the subsurface depth spacings.

$$\texttt{dt\_regularity} = \frac{\texttt{dt\_spacing\_std}}{\texttt{dt\_spacing\_mean}} \tag{D.8}$$

A low value indicates that spacings are uniform, consistent with a regular dead-time pattern. `missing` if `dt_spacing_mean` is zero or `dt_spacing_std` is `missing`.

Afterpulse features

A photon is counted in the 2.3 m afterpulse band if $2.0 < d_j < 2.6$, and in the 4.2 m band if $3.9 < d_j < 4.5$, where the depth below the surface is $d_j = h_s - h_j$. We use this to calculate some basic descriptives for both afterpulse band (`ap_XX_present`, `ap_XX_count`, `ap_depth_XX_mean`). After this, we can calculate the ratio between afterpulse photons and the total number of photons in the window using for both the 2.3 m and 4.2 m depth:

$$\texttt{ap\_XX\_ratio} = \frac{\texttt{ap\_XX\_count}}{n} \tag{D.9}$$

D.5. Water intersection length

The dataset we created not only contains a label indicating the presence of `water`, but also provides information on the size of the water body. The `water_length_m` column contains, per photon, the intersection length (in meters) of the overlap between the ICESat-2 beam and the BGT-waterdeel polygon containing the photon. In case a photon does not lie within a water polygon, the value is set to `missing`. The calculation is done in five steps.

Step 1: Labeling water points All points in the beam are tested to see if it is contained within a BGT water polygon. Each photon receives a binary `water` flag (1 = inside a water polygon, 0 = outside).

Step 2: Identifying water clusters The photons are ordered in along-track order. All consecutive photons that all lie in water (`water` = 1) are grouped into a water cluster $[i_s, i_e]$, where i_s and i_e are the start and end indices of the cluster, respectively.

Step 3: Fitting a line After all points are clustered together, we approximate a straight line through all the points in a cluster using Principal Component Analysis (PCA). Let $\mathbf{C} \in \mathbb{R}^{m\times 2}$ be the matrix of longitude-latitude coordinates of the photons used for fitting, where m represents the number of fit points. The centroid $\bar{\mathbf{c}}$ is subtracted and a Singular Value Decomposition (SVD) is applied to the matrix. The first right-singular vector $\mathbf{v}_1$ gives the dominant direction of the points. The ray is extended by $\Delta = 0.02$ beyond the projected extremes of the cluster:

$$\mathbf{p}_{\text{start}} = \bar{\mathbf{c}} + \mathbf{v}_1\,(\hat{p}_{\min} - \Delta), \qquad \mathbf{p}_{\text{end}} = \bar{\mathbf{c}} + \mathbf{v}_1\,(\hat{p}_{\max} + \Delta) \tag{D.10}$$

where $\hat{p}_{\min}$ and $\hat{p}_{\max}$ are the minimum and maximum scalar projections of the cluster points onto $\mathbf{v}_1$. When the cluster contains fewer than 10 photons, the fit is extended with up to 10

neighboring photons on each side of the cluster to gain a more accurate estimation of the direction.

Step 4: Polygon intersection All BGT water polygons with bounding boxes that intersect with the cluster's bounding box are retrieved and merged with a union operation. This is because the BGT is a large file (4.5 GB) with hundreds of thousands of polygons. This first query is required to significantly reduce computation time. The polygons are merged because the dataset contains multiple polygons of the same type, most of which are layered on top of each other. By merging them into one, double counts are prevented. The PCA line is then intersected with this merged polygon. The result is one or more line segments, each representing a portion of the beam that passes through water.

Step 5: Length assignment and fallback Each intersection segment is assigned to the photons whose latitude falls within the latitude range of that segment. The geodesic length of the segment is computed on the WGS-84 ellipsoid and stored in `water_length_m` for all assigned photons.

Any water photon not covered by an intersection segment (e.g. because the ray missed a narrow polygon) receives a fallback value equal to the geodesic distance between the first and last photon of the cluster:

$$\ell_{\text{fallback}} = d_{\text{WGS84}}((x_{i_s}, y_{i_s}),\ (x_{i_e}, y_{i_e})) \tag{D.11}$$

E. Model metrics

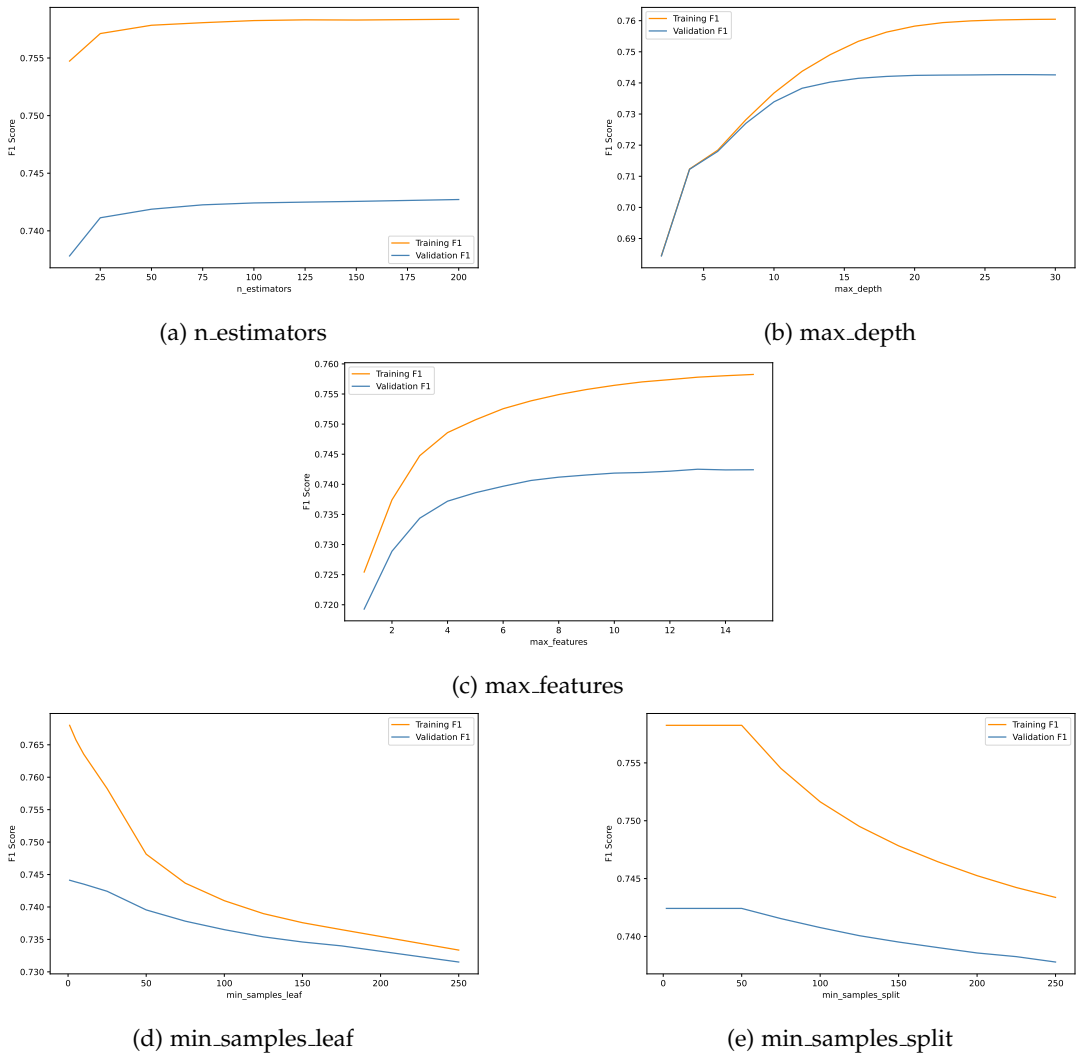
This appendix presents detailed model performance metrics that expand on the results reported in the Scientific Article. The sections below cover hyperparameter tuning curves, feature importance rankings across all six spatial window radii, per-segment performance tables, and the results of the improvement experiments described in Section 3.5.5.

E.1. Hyperparameter validation curves

Each hyperparameter was tuned independently by sweeping its value over the range shown in Table E.1 while holding all other parameters constant. Model performance was assessed using 5-fold cross-validation with the F1-score as the evaluation metric. The gap between the training curve (orange) and the validation curve (blue) is an indicator of overfitting: a large gap suggests the parameter value allows the model to memorize training examples, while convergence of the two curves indicates that the model generalizes well. The chosen value for each parameter corresponds to the point at which the validation performance flattens. It is better to be careful in the selection of hyperparameters, as extreme parameter values provide no meaningful improvement and increase the risk of overfitting.

#	Hyperparameter	Min.	Max.	Chosen
1	max_depth	2	30	20
2	max_features	2	15	12
3	min_samples_leaf	1	250	50
4	min_samples_split	1	150	50
5	n_estimators	10	200	100

Table E.1.: Tuning ranges and chosen values for each hyperparameter, derived from the validation curves in Figure E.1. Values are chosen conservatively because the tuning dataset comprised only 1.4 million points (0.26% of all available observations); more aggressive settings would risk overfitting to this limited subset.



(a) n_estimators

(b) max_depth

(c) max_features

(d) min_samples_leaf

(e) min_samples_split

Figure E.1.: Validation curves for all five Random Forest hyperparameters `n_estimators` controls the size of the forest. The curve shows that performance does not increase when there are more than 75 trees, confirming that additional trees contribute negligible new information. `max_depth` limits how deep each decision tree may grow, which reduces the risk of overfitting. `max_features` determines the number of features considered at each split, which introduces randomness between trees and improves the diversity in the forest. `min_samples_leaf` and `min_samples_split` control the minimum number of samples required at the leaf and split levels, respectively. By increasing these parameters, the tree depth is limited, which prevents overfitting on small subsets of the training data.

E.2. Feature importances per window radius

The feature selection method described in Section 3.4.3 was used for each of the six window radii. Tables E.2 and E.3 list the top-15 features ranked by Mean Decrease in Impurity (MDI) at each scale. Features that appear in multiple radii can be considered robust predictors of water presence, as they contribute to the model regardless of the width.

Table E.2.: Top-15 feature importances by window radius (1 m – 5 m)

1 m		**2.5 m**		**5 m**	
Feature	MDI	Feature	MDI	Feature	MDI
conf_4	**0.438**	**conf_4**	**0.420**	**spacing_mean**	**0.448**
quality	0.139	frac_02m	0.105	quality	0.118
sun_angle	0.084	quality	0.082	frac_02m	0.097
h_skew	0.066	sun_angle	0.079	h_kurt	0.060
frac_01m	0.061	kde_peaks_h	0.078	sun_angle	0.053
spacing_median	0.056	uncertainty	0.061	uncertainty	0.044
uncertainty	0.053	h_kurt	0.037	h_ref_std	0.041
h_std	0.052	prominence	0.034	h_std	0.038
strong_beam	0.029	strong_beam	0.021	strong_beam	0.024
prominence	0.009	subsurface_depth_1	0.020	bimodal_score	0.022
slope	0.006	peak_dist	0.020	prominence	0.017
kde_peaks_h	0.003	dt_spacing_std	0.014	subsurface_depth_1	0.013
dh_mean_right	0.002	detector_id	0.012	dt_regularity	0.010
dh_std_left	0.001	slope	0.011	slope	0.010
dh_mean_left	0.001	dt_regularity	0.007	kde_peaks_h	0.005

Table E.3.: Top-15 feature importances by window radius (10 m – 25 m)

10 m		**20 m**		**25 m**	
Feature	MDI	Feature	MDI	Feature	MDI
spacing_median	**0.400**	**spacing_median**	**0.373**	**quality**	**0.317**
quality	0.137	quality	0.152	spacing_median	0.211
h_ref_std	0.096	h_ref_std	0.123	h_ref_std	0.138
frac_02m	0.093	frac_02m	0.085	frac_02m	0.079
frac_01m	0.061	frac_01m	0.063	frac_01m	0.075
bimodal_score	0.036	bimodal_score	0.033	bimodal_score	0.036
sun_angle	0.034	residual	0.030	residual	0.029
h_std	0.027	sun_angle	0.028	strong_beam	0.022
prominence	0.027	uncertainty	0.023	h_skew	0.021
strong_beam	0.024	strong_beam	0.022	dt_spacing_std	0.014
slope	0.016	h_skew	0.021	prominence	0.014
dt_regularity	0.015	dt_spacing_std	0.012	subsurface_depth_1	0.013
dt_spacing_std	0.013	subsurface_depth_1	0.012	dt_regularity	0.013
subsurface_depth_1	0.012	prominence	0.012	subsurface_depth_2	0.009
n_subsurface_peaks	0.010	dt_regularity	0.011	n_subsurface_peaks	0.008

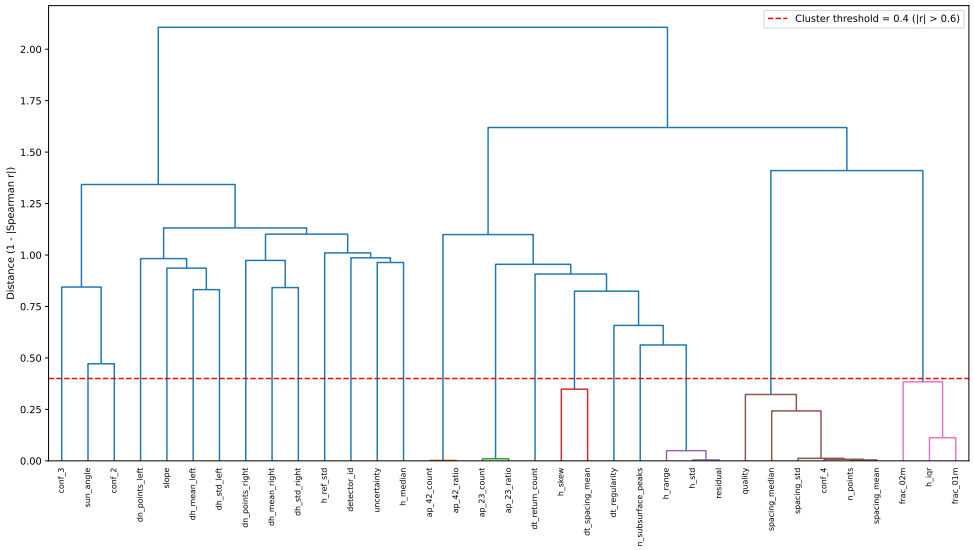


Figure E.2.: The higher up the y-axis of the dendrogram two features join, the less correlated they are. Closely correlated features are colored the same. Only non-binary features were considered, and a clustering threshold of 0.4 was used.

E.3. Selected feature importance statistics of final model

The final model uses a 2.5 m window radius trained only on water bodies < 25 m. Figure E.2 displays the results of decorrelation, while Table E.4 shows the distribution of MDI importance scores across all 100 decision trees in the forest for each of the 15 selected features. The Standard Deviation (SD) and interquartile range (Q1 - Q3) indicate how consistently a feature contributes across trees. A high SD relative to the mean, as seen for `conf_4` and `sun_angle`, suggests that a small number of trees assign that feature a disproportionately high importance. This probably reflects that they have interaction with each other, as can be seen in the bi-variate distribution in Figure 3.7. The other option is that these features are sensitive to specific subsets of the training data. Features with low a SD (e.g. `dt_regularity`, `uncertainty`) contribute more uniformly across all trees, which indicates that these are more stable in their relevance. Figure E.3 shows the MDI and permutation importance side by side, illustrating cases where the two metrics diverge.

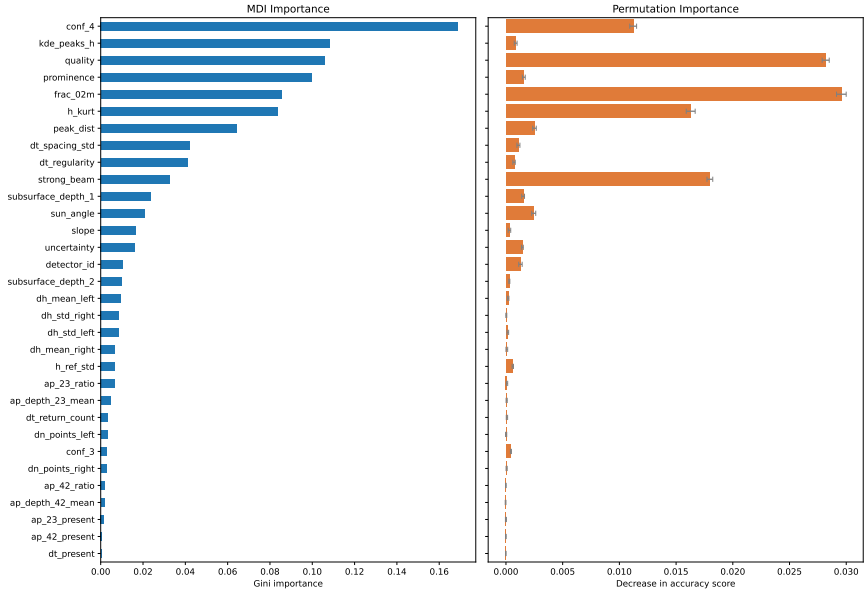


Figure E.3.: MDI (Gini) and permutation importance scores for the 2.5 m window. `conf_4` dominates on MDI but performs comparatively worse on permutation importance, suggesting its influence is partly shared with correlated features, most likely the `sun_angle`. By contrast, `frac_02m` achieves the highest permutation score, indicating that shuffling this feature degrades generalization performance more than any other, making it the strongest indicator of robustness.

Rank	Feature	Mean	SD	Median	Q1	Q3
1	`conf_4`	0.412	0.187	0.506	0.501	0.509
2	`sun_angle`	0.101	0.181	0.006	0.005	0.032
3	`kde_peaks_h`	0.079	0.003	0.078	0.076	0.080
4	`quality`	0.066	0.017	0.076	0.050	0.077
5	`uncertainty`	0.064	0.002	0.064	0.063	0.066
6	`frac_02m`	0.059	0.013	0.063	0.045	0.067
7	`h_kurt`	0.053	0.012	0.048	0.045	0.057
8	`subsurface_depth_1`	0.040	0.004	0.041	0.038	0.042
9	`dt_spacing_std`	0.028	0.006	0.029	0.023	0.034
10	`strong_beam`	0.022	0.002	0.021	0.020	0.022
11	`slope`	0.021	0.005	0.017	0.016	0.025
12	`prominence`	0.019	0.010	0.013	0.013	0.031
13	`peak_dist`	0.016	0.045	0.011	0.011	0.011
14	`detector_id`	0.012	0.001	0.012	0.012	0.013
15	`dt_regularity`	0.009	0.000	0.009	0.009	0.009

SD = standard deviation; Q1 / Q3 = first and third quartile of the importance distribution.

Table E.4.: Feature importances of the Random Forest model ranked by (MDI) score. The statistics were calculated across all 100 decision trees in the forest.

E.4. Detailed model performance per water segment

This section provides the detailed results of the model performance in Section 3.5. Table E.6 gives the recall and mean confidence values for the final model, breaking the < 10 m group into 1 m intervals to show how performance degrades for the smallest detectable water bodies. Table E.5 summarizes the distribution of photons across the different water length bins of the full ATL03 dataset used. Table E.7 compares recall and confidence across all six window radii. Table E.8 presents the results of the seven training-configuration experiments described in Section 3.5.5.

Length range (m)	Count	Share (%)
< 10	18,419,787	11.2
10–25	15,934,480	9.7
25–50	12,890,541	7.8
50–100	13,118,185	7.9
100–250	18,606,357	11.3
250–500	14,765,031	9.0
≥ 500	71,296,781	43.2
Total	165,031,162	100.1

Table E.5.: Distribution of along-track water segment intersection lengths in the selected ATL03 evaluation dataset. Water bodies ≥ 500 m account for 43.2% of all water points, creating a strong class imbalance that suppresses overall accuracy and precision when used as summary statistics across the full dataset.

Group	Evaluated points	Trained points	Recall	Mean confidence
0–1 m	118,857	27,745	0.414	0.689
1–2 m	773,277	180,686	0.589	0.682
2–3 m	1,561,572	364,995	0.663	0.688
3–4 m	2,052,495	481,111	0.729	0.693
4–5 m	2,170,597	509,686	0.767	0.699
5–6 m	2,003,357	469,409	0.790	0.704
6–7 m	1,790,854	420,777	0.806	0.711
7–8 m	1,618,557	379,546	0.825	0.716
8–9 m	1,467,213	344,892	0.837	0.722
9–10 m	1,364,512	319,649	0.844	0.727
10–25 m	12,435,790	3,498,690	0.871	0.740
Land	361,736,349	6,997,182	0.745	0.751

Table E.6.: The recall and mean confidence scores per water bin for the final model. The < 10 m group is split into 1 m intervals to highlight how recall degrades for the smallest of water bodies.

	1 m		2.5 m		5 m		10 m		20 m		25 m	
Group	Rec.	Conf.	Rec.	Conf.	Rec.	Conf.	Rec.	Conf.	Rec.	Conf.	Rec.	Conf.
<10 m	0.577	0.701	0.599	0.697	0.573	0.691	0.507	0.687	0.483	0.689	0.475	0.692
10–25 m	0.773	0.744	0.803	0.741	0.799	0.740	0.766	0.739	0.751	0.724	0.731	0.721
25–50 m	0.831	0.779	0.859	0.777	0.863	0.780	0.850	0.790	0.859	0.790	0.852	0.784
50–100 m	0.858	0.799	0.885	0.799	0.889	0.805	0.884	0.818	0.896	0.824	0.894	0.819
100–250 m	0.865	0.814	0.892	0.814	0.899	0.821	0.901	0.839	0.914	0.848	0.914	0.844
250–500 m	0.834	0.811	0.869	0.815	0.883	0.826	0.892	0.845	0.908	0.857	0.909	0.854
≥500 m	0.733	0.775	0.743	0.782	0.817	0.819	0.913	0.882	0.949	0.914	0.951	0.913
Land	0.799	0.761	0.789	0.773	0.796	0.788	0.817	0.799	0.828	0.803	0.824	0.803
Accuracy	0.787		0.786		0.801		0.825		0.838		0.834	
Precision	0.623		0.617		0.635		0.668		0.686		0.680	

Rec. = Recall; Conf. = Mean confidence.

Table E.7.: Recall and confidence scores by water segment length group and window radius, all models were trained on a balanced subset of all points. Narrow windows (1 m and 2.5 m) achieve the best recall on the smallest water bodies, while wider windows improve recall on larger segments. The overall accuracy is dominated by the ≥ 500 m bin due to its overrepresentation in the dataset, making it a misleading summary for assessing performance on smaller water bodies.

	Base case		Run 1		Run 2		Run 3	
Min. 5 pts			✓				✓	
Merge win.					✓		✓	
Group	Rec.	Conf.	Rec.	Conf.	Rec.	Conf.	Rec.	Conf.
<10 m	0.767	0.705	0.766	0.694	0.781	0.752	0.790	0.740
10–25 m	0.871	0.740	0.866	0.729	0.851	0.767	0.845	0.755
25–50 m	0.858	0.733	0.841	0.717	0.718	0.720	0.689	0.712
50–100 m	0.848	0.730	0.826	0.714	0.606	0.700	0.567	0.696
100–250 m	0.813	0.723	0.782	0.707	0.483	0.696	0.441	0.697
250–500 m	0.740	0.715	0.702	0.701	0.395	0.701	0.354	0.704
≥500 m	0.510	0.729	0.490	0.727	0.240	0.725	0.218	0.729
Land	0.745	0.751	0.734	0.751	0.795	0.783	0.781	0.781
Accuracy	0.724		0.710		0.691		0.664	
Precision	0.537		0.554		0.491		0.497	

Rec. = Recall; Conf. = Mean confidence. All runs use a 2.5 m base window; merged windows combine 2.5 m, 10 m, and 25 m features.

Table E.8.: Recall and confidence by water length bin for four training configurations. The base case is the final model. None of the runs show improvements in the results when compared to the base case, but merging multiple windows does slightly improve recall of bins < 10 m at the cost of a lower recall for $10 - 25$ m.

F. QGIS Plugins

F.1. ICESat-2 Profile Viewer

During the entire process, it was often necessary to visualize the data we worked with. Though it is possible to visualize the points of small segments using Julia or Python, actually geo-referencing them and understanding the spatial context was extremely difficult. QGIS, on the other hand, had the opposite problem; all the points could be understood in their spatial context, but it was not possible to plot the points' verticality. We therefore decided to combine the strengths of the two and create a QGIS plugin with which it is possible to select points in QGIS and visualize them in a nice Python plot (see Figure F.1). The plugin is publicly available (Rotteveel, 2026b). This page also includes an example and download instructions to get started. This plugin was of enormous help during the entire process, especially while exploring and understanding the performance of the different features in the Random Forest. It is a useful tool that can identify what the Random Forest actually makes its decisions with.

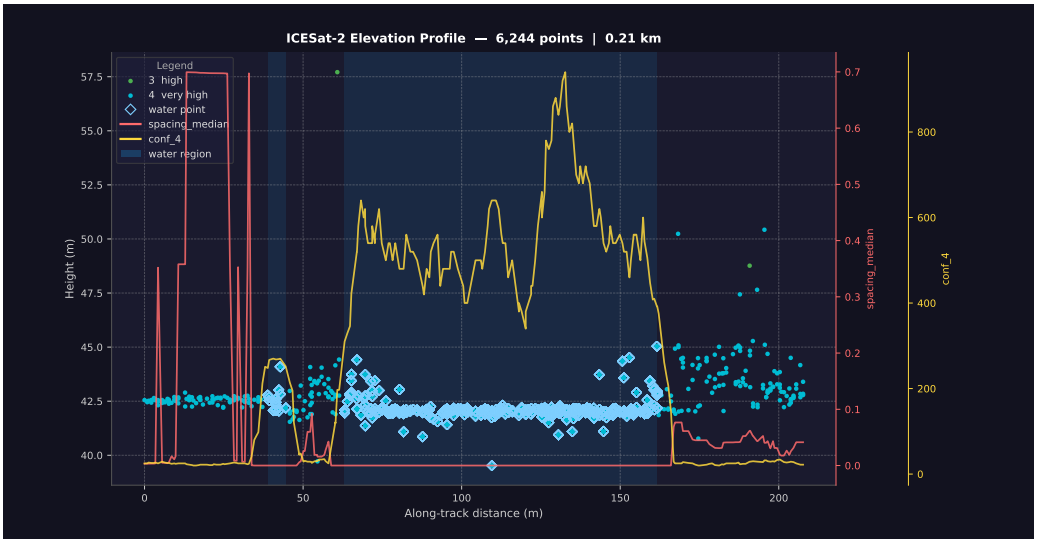


Figure F.1.: Example of plot created by our ICESat-2 profile viewer QGIS plugin

F.2. Joblib classification visualizer

After training the machine learning model, it is important to properly evaluate the results. Though metrics about the performance can provide some insights, they cannot answer the fundamental questions of *why*, *when*, and *where* the model underperforms. That is why we decided to create a QGIS plugin that takes a `.joblib` (Varoquaux et al., 2024) file of a machine learning classification model as input and visualizes the classifications made by the model on the data of choice. We tried to make the code usable for most file types and classification models, but we have not yet been able to test and validate whether this works. For the purposes of this master's thesis, we made it possible to color features according to the model's prediction, the accuracy, and confidence. This made it possible to quickly identify where the model struggled. Using the plugin presented in Section F.1, it was then possible to visualize the points with an overlay of the features to see why. Another benefit was that this plugin made it possible to download data outside the scope of the research area and visualize the results of the model. The plugin is publicly available (Rotteveel, 2026c). This page also includes an example and download instructions to get started.

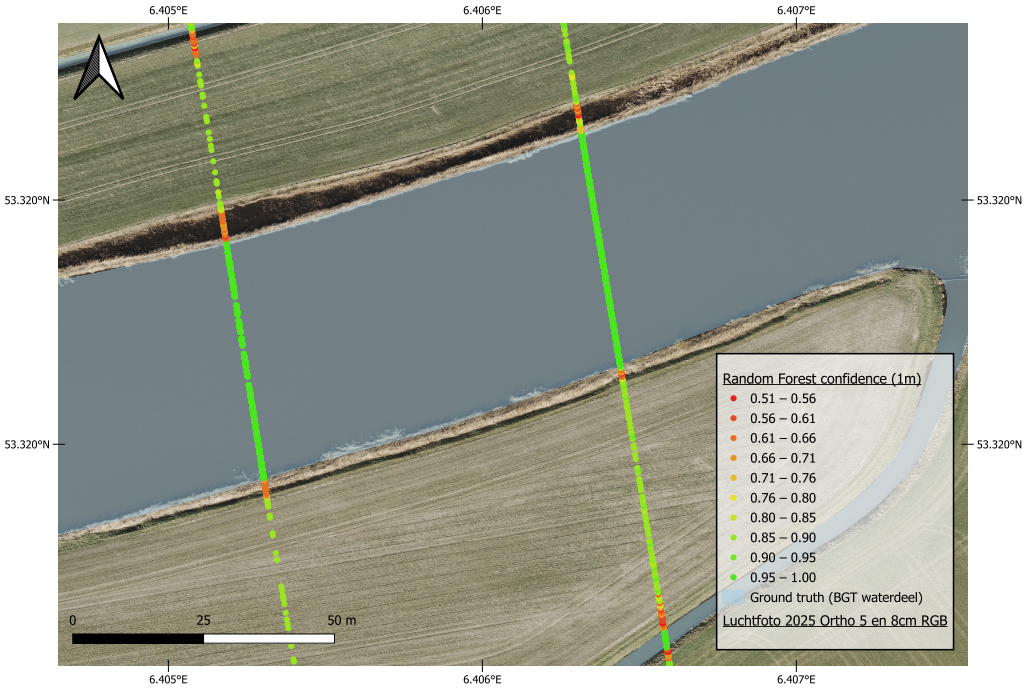


Figure F.2.: Example of confidence visualization created by an older version of our Random Forest model and our QGIS plugin. Observed ICESat-2 data (beam gt3l/r) on 23-11-2018.

Bibliography

Abdulla A, Baryannis G, and Badi I (2023). An integrated machine learning and MARCOS method for supplier evaluation and selection. *Decision Analytics Journal*, 9:100342. ISSN 27726622. doi:10.1016/j.dajour.2023.100342.

Biancamaria S, Schaedele T, Blumstein D, Frappart F, Boy F, Desjonquères JD, Pottier C, Blarel F, and Niño F (2018). Validation of Jason-3 tracking modes over French rivers. *Remote Sensing of Environment*, 209:77–89. ISSN 00344257. doi:10.1016/j.rse.2018.02.037.

Breiman L (2001). Random forests. *Machine Learning*, 45(1):5–32. doi:10.1023/a:1010933404324.

Breiman L, Friedman JH, Olshen RA, and Stone CJ (2017). *Classification And Regression Trees*. Routledge. ISBN 9781315139470. doi:10.1201/9781315139470.

Chawla I, Karthikeyan L, and Mishra AK (2020). A review of remote sensing applications for water security: Quantity, quality, and extremes. *Journal of Hydrology*, 585. ISSN 00221694. doi:10.1016/j.jhydrol.2020.124826.

Criminisi A, Shotton J, and Konukoglu E (2012). Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227. ISSN 1572-2740. doi:10.1561/0600000035.

Dask Development Team (2016). *Dask: Library for dynamic task scheduling*.

Datta RT and Wouters B (2021). Supraglacial lake bathymetry automatically derived from ICESat-2 constraining lake depth estimates from multi-source satellite imagery. *Cryosphere*, 15(11):5115–5132. ISSN 19940424. doi:10.5194/tc-15-5115-2021.

Dawson N, Fischer J, Kuhn M, Pasotti A, Rouzaud D, mhugent, Bruy A, Sutton T, Dobias M, Pellerin M, Rouault E, Olaya V, Blottiere P, Macho W, Blazek R, Bartoletti L, Sherman G, Sant-anna H, Cabieces J, Woodrow N, signedav, rldhont, Natsis S, Shaffer L, Felder J, Belgacem N, Santilli S, Larosa S, Mani S, and Jurgiel B (2026). qgis/qgis: 3.44.9. doi:10.5281/zenodo.19401570.

Dehghani AA, Movahedi N, Ghorbani K, and Eslamian S (2022). Decision tree algorithms. In *Handbook of HydroInformatics: Volume I: Classic Soft-Computing Techniques*, pages 171–187. Elsevier. ISBN 9780128212851. doi:10.1016/B978-0-12-821285-1.00004-X.

Feng Y, Zhang H, Tao S, Ao Z, Song C, Chave J, Le Toan T, Xue B, Zhu J, Pan J, Wang S, Tang Z, and Fang J (2022). Decadal Lake Volume Changes (2003–2020) and Driving Forces at a Global Scale. *Remote Sensing*, 14(4). ISSN 20724292. doi:10.3390/rs14041032.

Geonovum (2020). Basisregistratie grootschalige topografie gegevenscatalogus bgt 1.2.

Jasinski M, Stoll J, Hancock D, Robbins J, and Nattala J (2025a). Atlas/icesat-2 l3b mean inland surface water data, version 4. doi:10.5067/ATLAS/ATL22.004.

Jasinski M, Stoll J, Hancock D, Robbins J, Nattala J, Morison J, Jones B, Ondrusek M, and Parrish C (2025b). ICESat-2 Algorithm Theoretical Basis Document (ATBD) for Along Track Inland Surface Water Data, ATL13, Release 7. ICESat-2 Project. *NASA Goddard Space Flight Center*, 190:pp. doi:10.5067/46BO943W5S2X.

Karsten Rinke, Philipp Steffen Keller, Xiangzhen Kong, Dietrich Borchardt, and Markus Weitere (2019). Atlas of Ecosystem Services. In Matthias Schröter, Aletta Bonn, Stefan Klotz, Ralf Seppelt, and Cornelia Baessler, editors, *Atlas of Ecosystem Services: Drivers, Risks, and Societal Responses*, chapter 30, pages 191–195. Springer. ISBN 978-3-319-96229-0. doi:10.1007/978-3-319-96229-0_30.

Kaya Y, Balik Sanli F, and Abdikan S (2025). Refinement of ICESat-2 derived inland water surface levels with the TG20 local geoid model: In the case of Türkiye lakes. *Physics and Chemistry of the Earth*, 139. ISSN 14747065. doi:10.1016/j.pce.2025.103900.

Li S, Sun D, Goldberg M, and Stefanidis A (2013). Derivation of 30-m-resolution water maps from terra/modis and srtm. *Remote Sensing of Environment*, 134:417–430. ISSN 0034-4257. doi:https://doi.org/10.1016/j.rse.2013.03.015.

Lv Y, Jia L, Menenti M, Zheng C, Jiang M, Lu J, Zeng Y, Chen Q, and Bennour A (2024). A novel remote sensing method to estimate pixel-wise lake water depth using dynamic water-land boundary and lakebed topography. *International Journal of Digital Earth*, 17(1). ISSN 17538955. doi:10.1080/17538947.2024.2440443.

Ma W, Liu X, and Zhao X (2024a). Extraction of River Water Bodies Based on ICESat-2 Photon Classification. *Remote Sensing*, 16(16). ISSN 20724292. doi:10.3390/rs16163034.

Ma Z, Zhang S, Camps A, Park H, Liu Q, Tan P, and Wang C (2024b). A fast and efficient method to estimate inland water levels using CYGNSS L1 data and DTMs: Application to Floods, lakes and reservoirs monitoring. *Journal of Hydrology*, 645. ISSN 00221694. doi:10.1016/j.jhydrol.2024.132258.

Madani K (2026). Global Water Bankruptcy: Living Beyond Our Hydrological Means in the Post Crisis Era. Technical report, United Nations University Institute for Water, Environment and Health (UNU-INWEH), Richmond Hill, Ontario, Canada. doi:10.53328/INR26KAM001.

Magruder LA, Brunt K, Neumann T, Klotz B, and Alonzo M (2020). Passive ground-based optical techniques for monitoring the on-orbit ICESat-2 altimeter geolocation and footprint diameter. doi:10.1002/essoar.10504571.1.

Neuenschwander AL and Magruder LA (2019). Canopy and terrain height retrievals with icesat-2: A first look. *Remote Sensing*, 11(14). ISSN 2072-4292. doi:10.3390/rs11141721.

Neumann T, Hancock D, Robbins J, Gibbons A, Lee J, Brenner A, Felikson D, Harbeck K, Saba J, Luthcke S, Rebold T, Reese A, and Sutterly T (2025a). Atl03 known issues. doi:10.5067/ATLAS/ATL03.007.

Neumann T, Hancock D, Robbins J, Gibbons A, Lee J, Brenner A, Felikson D, Harbeck K, Saba J, Luthcke S, Rebold T, Reese A, and Sutterly T (2025b). Atlas/icesat-2 l2a global geolocated photon data, version 7. doi:10.5067/ATLAS/ATL03.007.

Neumann TA, Hancock D, Robbins J, Gibbons A, Lee J, Brenner A, Felikson D, Harbeck K, Saba J, Luthcke S, Rebold T, Reese A, and Sutterley T (2025c). Ice, Cloud, and Land Elevation Satellite (ICESat-2) Project Algorithm Theoretical Basis Document (ATBD) for Global Geolocated Photons ATL03, Version 7. doi:10.5067/ENBSEIJENE3U.

Neumann TA, Martino AJ, Markus T, Bae S, Bock MR, Brenner AC, Brunt KM, Cavanaugh J, Fernandes ST, Hancock DW, Harbeck K, Lee J, Kurtz NT, Luers PJ, Luthcke SB, Magruder L, Pennington TA, Ramos-Izquierdo L, Rebold T, Skoog J, and Thomas TC (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. ISSN 0034-4257. doi:https://doi.org/10.1016/j.rse.2019.111325.

Papa F, Crétaux JF, Grippa M, Robert E, Trigg M, Tshimanga RM, Kitambo B, Paris A, Carr A, Fleischmann AS, de Fleury M, Gbetkom PG, Calmettes B, and Calmant S (2023). Water Resources in Africa under Global Change: Monitoring Surface Waters from Space. doi:10.1007/s10712-022-09700-9.

Papa F, Durand F, Rossow WB, Rahman A, and Bala SK (2010). Satellite altimeter-derived monthly discharge of the Ganga-Brahmaputra River and its seasonal to interannual variations from 1993 to 2008. *Journal of Geophysical Research: Oceans*, 115(12). ISSN 21699291. doi:10.1029/2009JC006075.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, and Duchesnay E (2011). Scikit-learn: Machine Learning in Python. Technical report. doi:10.48550/arXiv.1201.0490.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E, and Louppe G (2012). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12.

Pekel JF, Cottam A, Gorelick N, and Belward A (2016). High-resolution mapping of global surface water and its long-term changes. *Nature*, 540. doi:10.1038/nature20584.

Powers D and Ailab (2011). Evaluation: From precision, recall and f-measure to roc, informedness, markedness correlation. *J. Mach. Learn. Technol*, 2:2229–3981. doi:10.9735/2229-3981.

Pronk M, Eleveld M, and Ledoux H (2024). Assessing Vertical Accuracy and Spatial Coverage of ICESat-2 and GEDI Spaceborne Lidar for Creating Global Terrain Models. *Remote Sensing*, 16(13). ISSN 20724292. doi:10.3390/rs16132259.

Pronk M and Gardner A (2026). SpaceLiDAR.jl. *Zenodo (CERN European Organization for Nuclear Research)*. doi:10.5281/zenodo.18618643.

Quinlan JR (1986). Induction of Decision Trees. Technical report. doi:10.1007/BF00116251.

Radoux J, Chomé G, Jacques DC, Waldner F, Bellemans N, Matton N, Lamarche C, D'Andrimont R, and Defourny P (2016). Sentinel-2's potential for sub-pixel landscape feature detection. *Remote Sensing*, 8(6):488. doi:10.3390/rs8060488.

Rotteveel HB (2026a). Icesat-2 automatic water detection model. doi:10.5281/zenodo.20547697.

Rotteveel HB (2026b). Icesat-2 profile viewer (qgis plugin). doi:10.5281/zenodo.20543317.

Rotteveel HB (2026c). Joblib classifier (qgis plugin). doi:10.5281/zenodo.20544547.

Schmitt M (2020). Potential of large-scale inland water body mapping from sentinel-1/2 data on the example of bavaria's lakes and rivers. *PFG*, 88:271–289. doi:10.1007/s41064-020-00111-2.

Song L, Song C, Luo S, Chen T, Liu K, Zhang Y, and Ke L (2023). Integrating ICESat-2 altimetry and machine learning to estimate the seasonal water level and storage variations of national-scale lakes in China. *Remote Sensing of Environment*, 294. ISSN 00344257. doi:10.1016/j.rse.2023.113657.

Song YY and Lu Y (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, 27(2):130–135. ISSN 10020829. doi:10.11919/j.issn.1002-0829.215044.

Varoquaux G, Grisel O, Estève L, Abadie A, Moreau T, Glaser P, Gervais P, Halchenko Y, Charras F, Self-Construct E, Berkes P, Lemaitre G, Lars, Larson E, du Boisberranger J, Chapman B, Mayner W, Niculae V, Hug N, Weyl M, Rocklin M, Arnold KC, Jerphanion J, Carvajal JMC, Neumann A, Olsson A, Besson L, Mueller A, jlopezpena, and d42 (2024). joblib/joblib: 1.4.2. doi:10.5281/zenodo.14915602.

Ward JH (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244. ISSN 01621459, 1537274X. doi:10.2307/2282967.

Yang H, Chen M, Xi X, and Wang Y (2024). A Novel Approach for Instantaneous Waterline Extraction for Tidal Flats. *Remote Sensing*, 16(2). ISSN 20724292. doi:10.3390/rs16020413.

Zhang A, Lipton Z, Li M, and Smola A (2023). *Dive into Deep Learning*. Cambridge University Press. doi:10.48550/arXiv.2106.11342.

Colophon

This document was typeset using LaTeX, using a modified the KOMA-Script class `scrbook` available at https://github.com/tudelft3d/msc_geomatics_thesis_template. The main font is Palatino.