Aalto University
School of Science

Master's Programme in Mathematics and Operations Research

Unsupervised wood and leaf classification from LiDAR point clouds utilising adaptive radius search and Gaussian Mixture Models

Rami Echriti

Master's Thesis
2025

Aalto University
School of Science

Author Rami Echriti

Title Unsupervised wood and leaf classification from LiDAR point clouds utilising adaptive radius search and Gaussian Mixture Models

Degree programme Mathematics and Operations Research

Major Systems and Operations Research

Supervisor Prof. Antti Hannukainen

Advisor Dr Mariana Campos

Collaborative partner Finnish Geospatial Research Institute

Date 21 July 2025 **Number of pages** 69 **Language** English

Abstract

Accurate classification of wood and leaf components is essential for understanding forest structure and estimating biophysical parameters. Classifying these components remains challenging due to variations in species characteristics and differences in data acquisition geometry. This thesis presents an unsupervised method for classifying wood and leaf points in LiDAR point clouds, developed using an adaptive radius search algorithm for geometric feature extraction combined with a Gaussian mixture model classifier.

The objective of this thesis was to automate the selection of the optimal local neighbourhood for each point to improve geometric feature extraction compared to previously applied fixed neighbourhood sizes. The adaptive radius search algorithm, originally introduced for urban point clouds, identifies the optimal neighbourhood by minimising Shannon's entropy, thus adapting the scale to capture the geometric structure of each point's neighbourhood. The extracted geometric features were further classified independently using a two-component Gaussian mixture model. Final classification was conducted as a weighted sum of the individual feature classifications.

Since the algorithm performs without manually labelled datasets, it is well suited for application to large-scale forest datasets and different data acquisitions. The performance of the unsupervised wood and leaf classification algorithm developed in this thesis was evaluated on three different LiDAR datasets consisting of Terrestrial Laser Scanning (TLS), Unmanned Aerial Vehicle (UAV), and Airborne Laser Scanning (ALS). The unsupervised wood and leaf classifier achieved average overall accuracies of 85% and 91% on Scots pines of TLS and UAV datasets, respectively, and the performance on TLS datasets was comparable to supervised methods. In the sparser ALS data, classification performance degraded, with lower precision and recall due to reduced geometric detail, indicating a point density threshold below which the unsupervised classification of this thesis becomes unreliable.

Overall, the results confirmed that adaptive neighbourhood optimisation enhances wood and leaf classification by improving feature quality and classification accuracy. The method demonstrated robust performance across different tree species as well as data acquisitions.

Keywords Unsupervised wood and leaf classification, adaptive radius search algorithm, Gaussian Mixture Model, TLS, UAV, ALS, laser scanning

Aalto-yliopisto
Perustieteiden
korkeakoulu

Tekijä Rami Echriti

Työn nimi Laserkeilauspistepilvien ohjaamaton luokittelu puu- ja lehtipisteiksi soveltaen mukautuvaa naapurustohakua ja Gaussin sekoitemalleja

Koulutusohjelma Mathematics and Operations Research

Pääaine Systems and Operations Research

Työn valvoja Prof. Antti Hannukainen

Työn ohjaaja TkT Mariana Campos

Yhteistyötaho Paikkatietokeskus

Päivämäärä 21.7.2025 **Sivumäärä** 69 **Kieli** englanti

Tiivistelmä

Puiden runko- ja lehvästöpisteiden luokittelu laserkeilauspistepilvistä on keskeinen osa metsäekosysteemien tutkimusta ja niiden biofysikaalisten tunnusten, kuten biomassan ja lehtialaindeksin, arviointia. Tässä työssä kehitettiin ohjaamatonta oppimista hyödyntävä algoritmi, joka luokittelee puiden runko- ja lehvästöpisteet käyttäen mukautuvaa sädehakualgoritmia yhdessä Gaussin sekoitemallien kanssa.

Työn tavoitteena oli automatisoida optimaalisen paikallisen naapuruston määrittäminen jokaiselle pisteelle, jotta pistepilven geometrisen rakenteen kuvaaminen tehostuisi verrattuna aiemmin käytettyihin vakiokokoisiin naapurustoihin. Alun perin rakennusten pistepilviaineistoille kehitetty mukautuva naapurustohaku määrittää kullekin pisteelle parhaiten soveltuvan mittakaavan minimoimalla naapuruston Shannonin entropian, mikä mahdollistaa pisteen lähiympäristön tarkemman mallintamisen. Pisteille lasketut geometriset piirteet luokiteltiin Gaussin sekoitemalleja hyödyntäen ja lopullinen puu- tai lehtipisteluokitus muodostettiin yhdistämällä useiden binääriluokitusten tulokset painotetun summan avulla. Menetelmä soveltuu laajojen metsäpistepilvien käsittelyyn ja monipuolisten kaukokartoitusaineistojen hyödyntämiseen, sillä se ei vaadi manuaalisesti luokiteltua opetusaineistoa.

Kehitetyn luokittelualgoritmin suorituskykyä arvioitiin kolmella eri laserkeilausopetusaineistolla: maa- (TLS), drooni- (UAV) ja lentolaserkeilausaineistolla (ALS). TLS- ja UAV-aineistoissa kehitetty luokitusmenetelmä saavutti mäntyjen osalta keskimäärin 85 %:n ja 91 %:n kokonaistarkkuudet ja erityisesti TLS-datassa menetelmän suorituskyky oli verrattavissa muihin ohjattuihin opetusalgoritmeihin. Pistetiheydeltään harvemmissa ALS-aineistoissa luokitustarkkuus heikkeni selvästi geometristen yksityiskohtien puutteen vuoksi.

Tulokset osoittivat, että luotettavan luokitustarkkuuden saavuttaminen edellyttää tietyn pistetiheyden ylittymistä. Tulokset vahvistivat mukautuvan naapurustokoon optimoinnin merkityksen rungon ja lehvästön erottelussa, sillä dynaaminen naapurustokoko paransi sekä piirteiden laskennan laatua että luokituksen tarkkuutta. Menetelmä osoitti myös vahvaa suorituskykyä eri puulajeilla ja laserkeilausaineistoilla, mikä korostaa sen monipuolista soveltuvuutta eri metsäaineistoissa.

Avainsanat Ohjaamaton runko- ja lehvästöpisteiden luokittelu, mukautuva naapurustokoon optimointialgoritmi, Gaussin sekoitemalli, TLS, UAV, ALS, laserkeilaus

Preface

This thesis was written as part of my Master's studies at Aalto University. The research presented here was carried out in collaboration with the Finnish Geospatial Research Institute.

I would like to express my gratitude to Professor Antti Hannukainen for his guidance throughout the thesis process and for the engaging discussions on methodological aspects related to this work.

I am especially grateful to Dr Mariana Campos for her consistent support, pertinent feedback, and thorough guidance at every stage of the project.

I would also like to thank the Finnish Geospatial Research Institute for the opportunity to work as a researcher and to contribute to an active and meaningful research environment. The experience of being part of a professional research setting greatly enriched my learning and development.

Finally, I am deeply thankful to my friends and family for their continued encouragement and support throughout my studies and the writing of this thesis.

Otaniemi, 21 July 2025

Rami Echriti

Contents

Symbols and abbreviations

Symbols

A_{3D}	Anisotropy
C	Covariance matrix
C_{3D}	Curvature
D_{3D}	Local point density
E_f	Shannon entropy function
L	Likelihood function
L_{3D}	Linearity
n	Number of points in local neighbourhood
$\mathcal{N}$	Normal distribution
P_{3D}	Planarity
Q	Expected complete data log-likelihood function
r	Radius of a local neighbourhood
r^*_{Ef}	Optimal radius minimising entropy
$\mathbb{R}$	The set of real numbers
S_{3D}	Sphericity
V_{3D}	Verticality
$\mathcal{V}^r_x$	Neighbourhood of point x with radius r
w_i	Mixture weight of Gaussian component i
$\hat{y}_k$	Predicted label of data point k
α_{1D}	Linearity feature ratio
α_{2D}	Planarity feature ratio
α_{3D}	Scattering feature ratio
β	Lagrangian multiplier
γ_{ki}	Responsibility of component i for data point k
λ_i	ith eigenvalue of covariance matrix
μ_i	Mean vector of Gaussian component i
σ_i	Standard deviation along principal component i
σ_1	First principal component (standard deviation along first principal direction)
Σ_i	Covariance matrix of Gaussian component i
θ	Set of model parameters

Abbreviations

3D	Three-dimensional
AdaBoost	Adaptive boosting
AGB	Above-ground biomass
ALS	Airborne laser scanning
CANUPO	Caractérisation de nuages de points
CC	CloudCompare
DBH	Diameter at breast height
DBSCAN	Density-based spatial clustering of applications with noise
E-step	Expectation step
EM	Expectation-maximisation
FGI	Finnish Geospatial Research Institute
GMM	Gaussian mixture model
k-NN	K-nearest neighbours
LAI	Leaf area index
LDA	Linear discriminant analysis
LiDAR	Light detection and ranging
LeWoS	An automated leaf and wood separation method
M-step	Maximisation step
ML	Maximum likelihood
NB	Naive Bayes
PAI	Plant area index
PCA	Principal component analysis
RF	Random forest
SP	Shortest-path
SVM	Support vector machine
TLS	Terrestrial laser scanning
UAV	Unmanned aerial vehicle

1 Introduction

Laser scanning technology, providing Light Detection and Ranging (LiDAR) data, is applied across various fields, including forestry, robotics, and automated vehicles. Within forestry, Terrestrial Laser Scanning (TLS) has emerged as a new tool for capturing 3D measurements that were not previously possible with field surveys, enabling non-destructive monitoring of vegetation structures and dynamics. In action, TLS emits low-energy infrared laser pulses with high reflectance over vegetation to capture XYZ coordinates of tree points (Campos et al., 2021). Due to the laser beam's high penetrability, multiple returns per pulse are recorded, resulting in dense, high-resolution three-dimensional point clouds of the forest environment (Wehr & Lohr, 1999). Point clouds provide valuable information on forest structures and tree growth. Each 3D point in the cloud is defined by its geometric coordinates as well as the intensity of the emitted laser beam, allowing for the computation of the tree's structural parameters, such as height, canopy area, Diameter at Breast Height (DBH), Above-Ground Biomass (AGB), Plant Area Index (PAI) and Leaf Area Index (LAI).

Tree height and DBH can often be estimated with centimetre-level precision using structural analysis techniques such as fitting cylinders to stems or estimating canopy surfaces (Boucher et al., 2021). In contrast, directly estimating metrics such as AGB and LAI is more challenging, since both require accurate separation of wood and leaf components. In practice, AGB indicates forest productivity and carbon storage, whereas LAI represents the total leaf surface area per unit ground area and determines the amount of light intercepted by plants for photosynthesis.

Separating wood and leaf points accurately remains a challenge due to the similar geometric structure of leaves and wood parts of the tree. Wood and leaf classification is typically divided into supervised and unsupervised methods. Supervised machine learning techniques have demonstrated high classification accuracy, but require manually labelled datasets. Creating such datasets is labour-intensive and time-consuming, limiting the benefits of supervised approaches for large-scale forest applications. In contrast, unsupervised methods rely solely on the geometric and radiometric properties of the LiDAR point cloud and do not require manually labelled data, thus providing a more efficient alternative for processing large datasets.

A key challenge in unsupervised wood and leaf classification is the tuning of algorithm parameters, such as neighbourhood size and feature distribution thresholds for class separation. Selecting an optimal neighbourhood size, and thus the search radius applied for feature extraction, is particularly difficult due to variations according to multiple factors, including point density, tree architecture, and species-specific structural characteristics. A radius that is too small may fail to capture the local neighbourhood structure, while an excessively large radius can include features from unrelated structures. Therefore, a suboptimal radius can significantly reduce classification accuracy. Although radius is one of the most important parameters in wood and leaf classification, a fixed radius remains commonly applied. Tree architecture further complicates generalisation, as coniferous and deciduous species exhibit distinct structural traits, making it difficult for a non-adaptive algorithm to perform consistently across various tree species.

This thesis aims to improve the parametrisation of an unsupervised wood and leaf separation method. In consequence, the objective is to implement a classification algorithm that automatically adapts to varying point cloud characteristics, thus reducing the need for manual tuning and enhancing classification accuracy across different datasets. The central hypothesis focuses on how an adaptive radius search method could improve unsupervised wood and leaf classification when applied to forest LiDAR data. In order to demonstrate the consistence of the adaptive radius approach, the study particularly investigates an adaptive neighbourhood search method originally developed for building point cloud classification by Demantké et al. (2011), and proposes its adaptation to tree point clouds. The method iteratively identifies the neighbourhood size that minimises local geometric entropy, therefore capturing the relevant structural scale. Although this approach has demonstrated success in urban datasets, it has not been applied to cases in forestry. The thesis thus supports that integrating adaptive radius selection with an unsupervised classification, specifically a Gaussian Mixture Model (GMM) classifier using geometric features, can yield a robust algorithm capable of accurately separating wood and leaf points across various species and point cloud densities. The study evaluates the method on different geometries and tree species acquired from several LiDAR datasets. In addition, the density threshold beyond which the proposed unsupervised wood and leaf classification becomes unreliable is examined.

To provide context for the proposed approach, the thesis is organized as follows. Section 2 reviews the relevant literature on wood and leaf classification, covering prior supervised and unsupervised methods and their known advantages and limitations. Section 3 presents the theoretical background for the proposed method, including an overview of the adaptive radius search algorithm, the set of geometric features employed, and the Gaussian mixture model formulation for unsupervised classification. Section 4 describes the research materials and methods; it introduces the terrestrial laser scanning, Unmanned Aerial Vehicle (UAV) and Airborne Laser Scanning (ALS) datasets used in the study and details the implementation of the adaptive classification algorithm. Section 5 reports the results of experiments, and compares the algorithm's classification performance across the different data sources and tree species. Section 6 discusses the findings, by examining the algorithm's performance in context, outlining its limitations and drawing comparisons with existing methods. Finally, Section 7 concludes the thesis by summarising the contributions of the work and suggesting directions for future research.

2 Literature review

Accurate classification of wood and leaf components presents a significant challenge due to difficulties from variations in data acquisition and different tree species. The most common approaches for wood and leaf separation are supervised and unsupervised machine learning methods. Supervised methods require training datasets consisting of manually classified wood and leaf point clouds, whereas unsupervised methods rely solely on the radiometric and geometric properties of the tree point cloud.

Research on wood and leaf classification began to gain attention in 2006, when Lalonde et al. (2006) proposed an unsupervised approach based on a Gaussian mixture model. The research was further advanced by studies such as Xu et al. (2007) and Livny et al. (2010), applying Shortest-Path (SP) analysis to achieve point-wise classification. In 2012, Brodu and Lague (2012) introduced supervised machine learning to the field, achieving an average overall accuracy of up to 98%. Since then, both supervised and unsupervised approaches have been researched, each presenting strengths and limitations. More recently, in 2018, deep learning methods have been employed. For instance, Xi et al. (2018) applied deep learning to wood and leaf classification, further advancing supervised methods.

2.1 Supervised methods

Extensive research has been conducted to propose supervised learning methods, which rely on four well-known machine learning algorithms: Random Forest (RF), Support Vector Machine (SVM), Gaussian mixture model, and Naive Bayes (NB). RF is a decision tree machine learning algorithm, constructing an ensemble of trees by training each on randomly selected subsets of the original data (Wang et al., 2017). The method provides accurate and robust classification, achieving high performance even when applied to noisy datasets (Geiß et al., 2015). The SVM algorithm is based on finding a hyperplane that maximises the distance between the nearest vectors in the wood and leaf classes (Wang et al., 2017). The hyperplane is positioned to achieve the largest possible margin between these classes, resulting in wood and leaf point separation. Both GMM and NB algorithms utilise probability distributions to separate wood and leaf components. The Gaussian mixture model estimates the likelihood of observing a given feature value by modelling each class as a Gaussian distributed random variable. In binary classification, the overall continuous probability density function is mathematically expressed as a linear combination of two Gaussian components, each representing a single class (Ma et al., 2016). Similarly, naive Bayes relies on Bayes' theory and assumes conditional independence between features within a given class. It assigns a label to a point based on the class with the highest probability (Wang et al., 2017).

Wang et al. (2017) compared the performance of the four well-known machine learning algorithms. The study was conducted on two isolated trees, with 26 features extracted from the point clouds. Results indicate that the RF classifier achieved the highest accuracy, with 98% accuracy using 26 features and 94% accuracy using 5 features. NB, in contrast, performed the worst, with 87% accuracy using 26 features

and 90% accuracy using 5 features. The other algorithms exhibited similar performance to RF, demonstrating less than 1% accuracy. Since the study was conducted on two isolated trees, the performance of these algorithms should be further validated by using different species and more complex ecosystems. Ma et al. (2016) applied an improved salient feature-based approach to separate wood and leaf classes. In the method, the Gaussian mixture model parameters were estimated by application of the Expectation-Maximisation (EM) algorithm, assigning iteratively posterior probabilities to train points from each class based on their spatial distributions. The resulting GMM was further utilised to distinguish wood from leaf components. After the separation, six filters were applied to improve the results. With this method, the accuracy achieved without filters ranged from approximately 65% to 73%, while with filters, it increased to between 84% and 98%. Despite Ma et al. (2016) achieving accuracies above 80%, the radius for identifying neighbouring points was determined empirically.

Various studies extract tree features from point clouds by employing fixed nearest neighbour search radius methods. Zhu et al. (2018) introduced a variation of the adaptive radius search algorithm for feature extraction and compared its performance with the conventional fixed radius approach. The optimal radius is chosen from a set of three predetermined radii, based on the tree's DBH, as the radius minimising Shannon entropy. Several search radii were tested across different forest plots to compute geometric and radiometric features, which were then compared with the results obtained by the fixed radius method. The adaptive radius method demonstrated more accurate classification and greater robustness. The key advantage of the adaptive near-neighbour radius search is the ability to adjust to different scenarios, such as variations in the dimensionality of local point clusters. For instance, depending on the search radius, the stem of the tree could be classified as either a two-dimensional or a one-dimensional structure, demonstrating the limitations of the fixed radius near-neighbour search (Zhu et al., 2018).

Zhu et al. (2018) further researched the classification accuracy of geometric and radiometric features, both individually and in combination. The accuracy was assessed by applying a random forest classifier with the geometric and radiometric features. The study concluded that radiometric features alone provided an average overall accuracy of 45.8%, whereas geometric features achieved an accuracy of 70.4%. Combining both geometric and radiometric features improved the overall accuracies to between 84% and 90%. Radiometric features provide additional information for the classification due to the different spectral properties of leaf and wood components, thereby increasing separation accuracy. However, the difference is not significant enough for the radiometric features alone to separate effectively wood from leaf components.

Moorthy et al. (2020) introduced an alternative method that avoids using a neighbourhood radius, instead relying on different neighbourhood sizes for computing the features. The applied classifiers were random forest, XGBoost and LightGBM. Both XGBoost and LightGBM algorithms construct each decision tree based on the outcomes of the previous trees (Moorthy et al., 2020). The machine learning classifiers were trained with 60 decision trees, as the study´s findings indicated that utilising more than 60 decision trees would not yield any improvements in the results. The

performance of the three methods was found to be similar. However, the RF classifier slightly outperformed the other algorithms, displaying an average accuracy of 94.2%. Despite achieving accurate results, Moorthy et al. (2020) concluded that no single algorithm is universally suitable for all datasets and that the proposed method failed for some individual trees.

Xi et al. (2020) compared the performance of nine deep learning classifiers: FCN-VGG, ResNet-50, ResNet-152, Inception-ResNet-v2, UNet, DenseNet, PSPNet, Superpoint Graph, and PointCNN, with six machine learning algorithms, consisting of k-Nearest Neighbours (k-NN), Support-Vector Machines, Adaptive Boosting (AdaBoost), random forest, naive Bayes and Linear Discriminant Analysis (LDA). Each point cloud plot scan was divided into blocks of 96x96x96 voxels with a spatial resolution of 0.06 m (Xi et al., 2020) and was provided as input to the deep learning classifier. The algorithm was trained on 623 blocks and tested on 190 blocks, with each voxel characterised by three attributes: point occupancy, laser intensity, and height. The deep learning model produced an output block of 96x96x96 labels, which were further transformed into point-wise labels to produce the final classification.

The machine learning algorithms were trained using 56 geometric and radiometric features. Among the nine deep learning classifiers tested, overall accuracies ranged from 83% to 94%, with the highest accuracy achieved by UNet and the lowest by PointCNN. For the machine learning algorithms, accuracies ranged from 78% to 92%, with NB achieving the lowest precision and RF the highest. These machine learning results were consistent with those reported by Wang et al. (2017).

Supervised machine learning algorithms are robust classification approaches, achieving high accuracy in wood and leaf separation. Despite the effectiveness of supervised methods, these approaches preserve issues. Machine learning, along with recent deep learning algorithms, requires significant computational time and manually labelled training data. Generating manually labelled training data is labour-intensive and prone to human error. Unsupervised methods bypass these challenges by separating wood and leaf components, relying on the radiometric and geometric properties of the point cloud.

2.2 Unsupervised methods

Unsupervised methods employ the geometric and radiometric features of the point cloud for wood and leaf separation. Several studies have demonstrated that among radiometric features, only intensity provides meaningful results. Zhu et al. (2018) validated this further by examining the performance of various radiometric features, including intensity, Riegl's pulse deviation, as well as red, green and blue channels from a digital camera. The study concluded that while intensity was the only feature providing significant results, other features had an importance of less than 2%.

Wang et al. (2020) developed an automated wood and leaf separation method relying only on the geometric features of the point cloud. The Leaf and Wood Separation algorithm (LeWoS) comprises three primary steps consisting of graph-based segmentation, class probability estimation and class regulation. With graph-based segmentation, the problem is transformed into a search for connected graph components, where the graph is constructed by utilising point cloud density and point-wise feature information (Wang et al., 2020). In the class probability estimation, wood and leaf components are identified from the connected graph by the parameters linearity and size, which are generated from each segment of the graph-based segmentation. Both linearity and size require threshold values for separation, which traditionally are manually fixed and fine-tuned. However, Wang et al. (2020) employed another approach by applying class probability to determine the threshold values, which resulted in spatially smoothed final classification. The advantage of the LeWoS algorithm is the requirement of only one defined parameter, allowing universal applications with different datasets. The accuracies achieved by Wang et al. (2020) were between 87.7% and 91.5% for different threshold values.

Tao et al. (2015) performed wood and leaf separation by applying shortest-path algorithms. The method consists of dividing tree point clouds into circles, arcs and line segments. Thresholding is further applied to the circles and line segments to eliminate shapes with a radius or length smaller than the threshold value. Skeleton wood points are extracted by using Dijkstra's shortest-path algorithm to connect central points derived from 2D projections of circular and linear features across horizontal segments. The final classification is achieved by applying KDTree range searching with a constant radius. The shortest-path method demonstrated effectiveness in wood and leaf separation, especially in detecting larger branches, achieving an average kappa coefficient of 0.83. However, the fixed search radius resulted in the misclassification of smaller branches in reason for their similar size to leaves, hence resulting in their elimination in the thresholding.

TLSeparation is a similar approach, developed by Vicari et al. (2019), which expands the classification further by combining pointwise and path-analysis classification in the geometric approach. Four different algorithms were used in total, two of which rely on geometric features computation at the point level, while the other two employ shortest-path algorithms. By applying geometric features from literature, Vicari et al. (2019) aimed to obtain distribution models from GMM. Expectation-maximisation algorithms were then combined with GMM to classify distribution models into a set of classes. Shortest-path algorithms were further applied to represent the tree structure

as a network. Henceforth, SP analysis was employed in the study to observe paths with a high frequency of occurrence and classify them as wood parts. On average, Vicari et al. (2019) achieved accuracies between 77% and 89%. The advantage of combining multiple methods is to provide a fully automated tool that is also robust and transferable.

Employing the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, Ferrara et al. (2018) differentiates from using geometric features by separating the point cloud into voxels. Voxels are further categorised as active or inactive depending on the point counts. The DBSCAN algorithm is subsequently applied to cluster these voxels, and the cluster containing the largest number of voxels is identified as wood. One of the main benefits of the DBSCAN algorithm is its efficiency when dealing with large datasets. The authors achieved accuracies varying from 94% to 96%. Despite achieving high accuracies, the algorithm still requires fine-tuning of two parameters, consisting of the radius Eps, representing the Eps-neighbourhood of a point and MinPts corresponding to the minimum number of points in an Eps-neighbourhood (Ferrara et al., 2018).

Hui et al. (2021) developed a method for separating wood and leaf components utilising mode points evolution. Initially, mean shift segmentation segments the point cloud into clusters, each with its centre defined as a mode point. The mode points further form the graph structure, with nodes representing these points and edges weighted by their pairwise distances. Within this graph, wood nodes are identified using analysis of path occurrence frequency, with leaf nodes determined by retracing the paths. To enhance detection accuracy, an evolution process is applied to the nodes, refining the separation of wood and leaf components. The final classification is achieved by merging the segments based on the mean shift results, leading to an average overall accuracy of 89%.

Shcherbcheva et al. (2023) conducted a study in which wood and leaf separation was performed by a Gaussian mixture model with two components. The study extracted geometric features across eight different radii for classification. The geometric features consisted of linearity, anisotropy, sphericity, surface variation, verticality and the first principal component. Each feature was used as an input to the GMM for classification. A point was labelled as wood if it was classified as such by any feature at any radius, resulting in a method that is prone to overclassification. Following classification, Shcherbcheva et al. (2023) applied DBSCAN for post-processing, achieving an average overall accuracy of 85%.

Generally, unsupervised methods demonstrate lower classification accuracy in comparison to supervised methods. Ali et al. (2024) compares the performance of the random forest algorithm with three fully automated wood and leaf classifiers: LeWoS (Wang et al., 2020), TLSeparation (Vicari et al., 2019), and CAractérisation de NUages de POints (CANUPO). Geometric features have demonstrated high performance in wood and leaf classification, leading automated classifiers to utilise these features from point clouds. CANUPO combines supervised learning with geometric features for wood and leaf separation. The geometric components, including Principal Component Analysis (PCA) and point density, are provided as inputs for a support vector machine to learn the relationship between geometric features and the labelled wood and leaf

parts of the tree (Ali et al., 2024). To enhance the accuracy, semi-supervised learning incorporates information from unlabelled points. Additionally, the method optimises computation by performing multiscale feature analysis on subsampled core points, balancing computation time with spatial resolution (Ali et al., 2024).

Applying the four leaf and wood classification algorithms resulted in accuracies of 95% for RF, 90% for LeWoS, 89% for CANUPO, and 81% for TLSeparation. Ali et al. (2024) observed that the accuracy obtained for RF was lower than Wang et al. (2017) but higher than Zhu et al. (2018), possibly due to differences in neighbourhood radius and the selection of multi-scale features (Ali et al., 2024). The accuracy of the LeWoS algorithm was similar to that reported by Wang et al. (2020). However, the accuracy of TLSeparation was lower than that achieved by Vicari et al. (2019), which might be a result of differences in parameter optimisation.

Unsupervised methods demonstrated high performance with wood and leaf classification, relying solely on the properties of tree point clouds without the need for manually labelled training data. Methods applying the search radius tend to underperform compared to voxel-based segmentation. Nonetheless, unsupervised methods offer the advantage of handling larger datasets with more efficient running times compared to supervised methods. However, supervised methods generally achieve higher classification accuracies than unsupervised approaches.

3 Background

The literature review in Section 2 highlights the strengths and limitations of both supervised and unsupervised methods for wood and leaf classification. While supervised methods generally yield higher accuracy, they require labour-intensive manually labelled datasets. In contrast, unsupervised methods are suitable for large-scale forest datasets since manually labelled datasets are not required. Gaussian mixture models have demonstrated robust performance in separating wood and leaf components as indicated by Shcherbcheva et al. (2023) and Vicari et al. (2019), resulting in GMM representing an effective choice for unsupervised wood and leaf classification in this study.

This section presents the theoretical basis of wood and leaf separation algorithm. It begins by describing the adaptive radius search method applied to determine optimal local neighbourhoods for feature computation. Following this, the Gaussian mixture model is introduced as the probabilistic approach applied to classify points into wood and leaf classes based on their geometric features. Thus, the proposed unsupervised method consists of three main steps, computation of an optimal neighbourhood radius, introduced in Section 3.1, extraction of geometric features, introduced in Section 3.2, and classification of the data using a Gaussian mixture model with two components, each corresponding to either the wood or leaf part of the tree, introduced in Section 3.3.

3.1 Adaptive radius search method

The adaptive radius search method, introduced by Demantké et al. (2011), determines an optimal neighbourhood radius for each point in a point cloud by minimising Shannon's entropy. This method enables the adaptive selection of local neighbourhoods, improving the accurate capture of geometric features in point cloud data.

Let $P \in \mathbb{R}^3$ denote the set of all points in the point cloud and $\mathbf{x} = (x, y, z)^T \in P$ be an arbitrary point. The neighbourhood of $\mathbf{x}$ with radius r, denoted as $\mathcal{V}_x^r$, is defined as the set of all points within distance r from $\mathbf{x}$,

$$\mathcal{V}_x^r = \{\mathbf{x_i} \in P \mid \|\mathbf{x} - \mathbf{x_i}\| \leq r\}\,.$$

To analyse the spatial distribution within this neighbourhood, the covariance matrix is computed based on the centred coordinates of all points in $\mathcal{V}_x^r$ as

$$C = \frac{1}{n} K^T K,$$

where $K = (\mathbf{x_1} - \overline{\mathbf{x}}, \mathbf{x_2} - \overline{\mathbf{x}}, \ldots, \mathbf{x_n} - \overline{\mathbf{x}})^T$ is the matrix of centred neighbourhood points. The centre of gravity of the neighbourhood $\mathbf{x}$, with n points, is computed as the mean of these n points by

$$\overline{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x_i}, \quad \mathbf{x_i} \in \mathcal{V}_x^r.$$

As the covariance matrix C is symmetric and positive semi-definite, all its eigenvalues are non-negative. Therefore, it can be diagonalised through eigenvalue decomposition,

$$C = R\Lambda R^T,$$

where R is an orthogonal matrix with columns representing the eigenvectors of C and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ is a diagonal matrix containing the corresponding eigenvalues, ordered such that $\lambda_1 \geq \lambda_2 \geq \lambda_3$. The eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ describe the variance of the neighbourhood along the principal component directions, with larger eigenvalues indicating greater dispersion in that direction. To characterise this variation, the standard deviations along the principal directions are derived in Equation (1) by

$$\sigma_i = \sqrt{\lambda_i}, \quad i \in \{1, 2, 3\}. \tag{1}$$

The standard deviations σ_i, defined in Equation (1), measure the distribution of points along the principal directions and form the basis for computing various geometric features. Three of such geometric features, proposed by West et al. (2004) and adapted by Demantké et al. (2011), characterise the local structure of the neighbourhood. The first feature α_{1D} represents the likelihood of the local points having a linear shape, the second α_{2D} quantifies planarity, and the third α_{3D} measures scatter behaviour within the neighbourhood. The geometric features are defined in Equation (2) as

$$\alpha_{1D} = \frac{\sigma_1 - \sigma_2}{\sigma_1 + \sigma_3}, \quad \alpha_{2D} = \frac{\sigma_2 - \sigma_3}{\sigma_1 + \sigma_3}, \quad \alpha_{3D} = \frac{\sigma_3}{\sigma_1 + \sigma_3}. \tag{2}$$

The optimal search radius is determined by evaluating Shannon entropy over the discrete probability distribution $\{\alpha_{1D}, \alpha_{2D}, \alpha_{3D}\}$ (Demantké et al., 2011). A lower entropy value indicates a well-defined structure, as one geometric characteristic dominates over the others (Demantké et al., 2011). The entropy function is given in Equation (3) by

$$E_f(\mathcal{V}_x^r) = -\sum_{i=1}^{3} \alpha_{iD} \ln(\alpha_{iD}), \tag{3}$$

The optimal neighbourhood radius $r^*_{E_f}$ for each point, given in Equation (4), is determined as the value minimising the entropy function.

$$r^*_{E_f} = \underset{r \in [r_{min}, r_{max}]}{\arg\min} \; E_f(\mathcal{V}_x^r). \tag{4}$$

Minimising entropy allows the method to adaptively determine a neighbourhood size that best captures the local geometric structure of the point cloud. The search range $[r_{min}, r_{max}]$ defines the minimum and maximum allowable radii, and its selection depends on the characteristics of the dataset. The procedure for selecting the neighbourhood radius is described in further detail in Section 4.2.

3.2 Geometric features

Geometric features are selected to capture the structural properties of the local neighbourhood. Various studies (Shcherbcheva et al., 2023; Xi et al., 2020; Vicari et al., 2019; Wang et al., 2017) suggest the application of curvature, linearity, anisotropy, verticality, local point density, the first principal component, sphericity and planarity for classifying wood and leaf components of the tree point cloud.

In particular, Wang et al. (2017) found local point density to be beneficial in classification, while both Shcherbcheva et al. (2023) and Wang et al. (2017) utilised curvature, linearity, anisotropy, verticality, sphericity, and planarity in differentiating between wood and leaf structures. Ali et al. (2024) and Shcherbcheva et al. (2023) demonstrated the advantages of including the First Principal Component to provide additional information about the tree structure. Previous studies (Shcherbcheva et al., 2023; Wang et al., 2017; Weinmann et al., 2014) further suggest that these geometric features exhibit distinct patterns between wood and leaf components. For example, branches tend to be more linear, tree stems generally have lower curvature, stem and branch points form anisotropic neighbourhoods due to their continuous surfaces, whereas canopy regions exhibit more scattered neighbourhoods (Shcherbcheva et al., 2023).

Each geometric feature characterises specific properties of the local neighbourhood. Curvature is computed as the ratio of the smallest eigenvalue to the sum of all eigenvalues, corresponding to the amount of variance explained by the first principal component. Linearity and anisotropy describe the elongation of a neighbourhood. Linearity is measured as the difference between the largest and second-largest eigenvalues, and anisotropy as the spread between the largest and smallest eigenvalues. Verticality measures how well the normal vector of the local neighbourhood aligns with the vertical axis, making it effective for detecting the tree trunk. Local point density calculates the number of points within a sphere of the optimal radius, distinguishing solid wood structures from more dispersed leaves. The First Principal Component represents the standard deviation along the first principal direction, and sphericity measures how isotropically distributed the points are within the neighbourhood.

These features were selected based on their ability to capture key geometric characteristics of the local neighbourhoods, and the linear combination of these features is utilised to provide accurate separation of wood and leaf components. The geometric features are defined by the eigenvalues of the neighbourhood covariance matrix, introduced in Equation (3.1). Table 1 summarises the mathematical form of the geometric features applied in this study for wood and leaf classification.

Feature	Notation	Formula
Curvature	C_{3D}	$\frac{\min(\lambda_i)}{\sum_i \lambda_i}$
Linearity	L_{3D}	$\frac{\lambda_1 - \lambda_2}{\lambda_1}$
Anisotropy	A_{3D}	$\frac{\lambda_1 - \lambda_3}{\lambda_1}$
Verticality	V_{3D}	$1 - \lvert N_z \rvert$
Local Point Density	D_{3D}	$\frac{n}{\frac{4}{3}\pi r^3}$
First Principal Component	σ_1	$\sqrt{\lambda_1}$
Sphericity	S_{3D}	$\frac{\lambda_3}{\lambda_1}$
Planarity	P_{3D}	$\frac{\lambda_2 - \lambda_3}{\lambda_1}$

Table 1: The table presents the geometric features and their definitions. The variable λ_i is the ith eigenvalue of the local covariance matrix, where $i \in 1, 2, 3$, n is the number of points in the local neighbourhood and N_z is the z-component of the normal vector.

3.3 Gaussian Mixture Model

A Gaussian mixture model is a probabilistic clustering algorithm representing data as a combination of multiple Gaussian distributions. Unlike traditional clustering methods, GMM estimates the probability that a data point belongs to a specific cluster, allowing for a more flexible data representation, particularly in scenarios where clusters overlap.

The model is mathematically defined as a weighted sum of Gaussian components, each characterised by its own mean and covariance matrix, approximating the overall data distribution. For classification, GMM computes the posterior probability of a data point belonging to each component by applying Bayes' rule, and assigns the point to the most likely one. The assignment relies on the model parameters — weights, means and covariances — of the components. To determine the optimal parameters for the model, GMM employs the expectation-maximisation algorithm, where the likelihood of the observed data is maximised iteratively.

3.3.1 Mathematical definition

In GMM, a probability density function of a D-dimensional feature vector $\mathbf{x} \in \mathbb{R}^D$ is represented as a weighted sum of N Gaussian distributions, defined in Equation (5) as

$$p(\mathbf{x} \mid \theta) = \sum_{i=1}^{N} w_i \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \tag{5}$$

where w_i denotes the mixture weights, satisfying $\sum_{i=1}^{N} w_i = 1$ and $\theta = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}$, for $i = 1, \ldots, N$ represents the set of model parameters. The multivariate Gaussian distribution for each component i, $\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, is defined in Equation (6) as

$$\mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\right). \tag{6}$$

Each Gaussian component is defined by a mean vector $\boldsymbol{\mu}_i$ and a covariance matrix $\boldsymbol{\Sigma}_i$. Together with the mixture weights, these parameters characterise the overall structure of the Gaussian mixture model.

The likelihood of the data $\mathbf{X} = \{\mathbf{x_1} \ldots \mathbf{x_K}\}$, given K observations, under the GMM, is defined by the product of the individual densities, as expressed in Equation (7).

$$L(\mathbf{X} \mid \theta) = \prod_{k=1}^{K} p(\mathbf{x} \mid \theta). \tag{7}$$

Since the likelihood function depends non-linearly on the parameters of θ, direct maximisation is not feasible. Instead, Maximum Likelihood (ML) estimates for the parameters are determined iteratively by applying the expectation-maximisation algorithm.

3.3.2 Expectation-Maximisation algorithm

The expectation-maximisation algorithm is an iterative method applied to estimate ML parameters in models with incomplete or unobserved data, referred to as latent variables. These latent variables are influenced by the observed data but are not directly measurable. In mixture models, for instance, latent variables indicate the specific mixture component to which each data point is assigned.

The expectation-maximisation algorithm consists of two primary steps. In the Expectation step (E-step), the expected value of the likelihood function is computed based on the current parameters. This is followed by the Maximisation step (M-step), during which the parameters are updated to optimise this expected value. The iterative process is continued until convergence, resulting in a stationary point where the derivative of the likelihood approaches zero. While effective, the algorithm may converge to local maxima or yield singular solutions, such as a mixture component collapsing to zero variance.

Expectation step

A latent variable z_k is introduced to indicate which Gaussian component generated each observation $\mathbf{x}_k$, defined by

$$z_k = \{1, 2, \ldots, N\}.$$

An indicator function is defined to indicate whether observation $\mathbf{x}_k$ was generated by component i, as

$$I(z_k = i) = \begin{cases} 1, & \text{if } z_k = i \\ 0, & \text{otherwise.} \end{cases}$$

Let $Z = \{z_1, \ldots, z_K\}$ denote the vector of all latent variables. Given the dataset X, the complete data likelihood function is expressed as the product of the likelihoods across all observations and components. The function is defined in Equation (8) as

$$L(X, Z \mid \theta) = \prod_{k=1}^{K} \prod_{i=1}^{N} [w_i \mathcal{N}(\mathbf{x_k} \mid \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})]^{I(z_k=i)} \tag{8}$$

Taking the logarithm of the complete data likelihood function in Equation (8) yields the complete data log-likelihood function, given in Equation (9) by

$$l(X, Z \mid \theta) = \log L(X, Z \mid \theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} I(z_k = i) \log \left(w_i \mathcal{N}(\mathbf{x_k} \mid \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i})\right). \tag{9}$$

Since the latent variables Z are unobserved, the expectation of the complete data log-likelihood is computed with respect to the posterior distribution of Z given the observed data and current parameter estimates $\theta^{(k)}$. The expected value yields the function $Q(\theta \mid \theta^{(k)})$, defined in Equation (10) by

$$
\begin{aligned}
Q(\theta \mid \theta^{(k)}) &= \mathbb{E}_{Z|X,\theta^{(k)}} \left[l(X, Z \mid \theta) \right] \\
&= \mathbb{E}_{Z|X,\theta^{(k)}} \left[\sum_{k=1}^{K} \sum_{i=1}^{N} I(z_k = i) \log \left(w_i \mathcal{N}(\mathbf{x_k} \mid \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}) \right) \right] \\
&= \sum_{k=1}^{K} \sum_{i=1}^{N} \mathbb{E}_{Z|X,\theta^{(k)}} \left[I(z_k = i) \right] \log \left(w_i \mathcal{N}(\mathbf{x}_k \mid \mu_i, \Sigma_i) \right).
\end{aligned}
\tag{10}
$$

The expected value of the indicator variable $I(z_k = i)$ corresponds to the posterior probability that data point $\mathbf{x}_k$ was generated by component i, under the current parameter estimates $\theta^{(k)}$, as shown in Equation (11).

$$
\mathbb{E}[I(z_k = i)] = P(z_k = i \mid \mathbf{x}_k, \theta^{(k)}). \tag{11}
$$

By applying Bayes' theorem to Equation (11), the expected value evaluates to as follows in Equation (12).

$$
\begin{aligned}
\gamma_{ki} = \mathbb{E}[I(z_k = i)] &= \frac{p(\mathbf{x}_k \mid z_k = i, \theta^{(k)}) p(z_k = i \mid \theta^{(k)})}{p(\mathbf{x}_k \mid \theta^{(k)})} \\
&= \frac{w_i^{(k)} \mathcal{N}(\mathbf{x}_k \mid \mu_i^{(k)}, \Sigma_i^{(k)})}{\sum_{j=1}^{N} w_j^{(k)} \mathcal{N}(\mathbf{x}_k \mid \mu_j^{(k)}, \Sigma_j^{(k)})}
\end{aligned}
\tag{12}
$$

The resulting responsibility γ_{ki} represents the probability that observation $\mathbf{x_k}$ belongs to component i, given the parameter estimates. If $\mathbf{x_k}$ is close to the mean μ_i, the corresponding responsibility is high, indicating that $\mathbf{x_k}$ is classified as belonging to component i.

The expectation step of the EM algorithm computes the expected value of the latent variables by evaluating their posterior probabilities given the observed data and current parameter estimates, providing the responsibility equation. These responsibilities are further applied in the maximisation step to update the parameters iteratively until convergence.

Maximisation step

The maximisation step updates the model parameters θ by maximising the function $Q(\theta \mid \theta^{(k)})$. In the expectation step, the function $Q(\theta \mid \theta^{(k)})$ was defined according to Equation (10). Replacing the expected value with γ_{ki}, as defined in Equation (12), yields Equation (13).

$$
\begin{aligned}
Q(\theta \mid \theta^{(k)}) &= \sum_{k=1}^{K} \sum_{i=1}^{N} \gamma_{ki} \log \left(w_i \mathcal{N}(\mathbf{x}_k \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right) \\
&= \sum_{k=1}^{K} \sum_{i=1}^{N} \gamma_{ki} \left[\log w_i + \log \mathcal{N}(\mathbf{x}_k \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right].
\end{aligned}
\tag{13}
$$

The logarithmic Gaussian probability density function of Equation (13) is expanded in Equation (14).

$$\begin{aligned}\log \mathcal{N}(\mathbf{x}_k \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) &= \log\left[\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)\right)\right] \\ &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i).\end{aligned} \tag{14}$$

Substituting Equation (14) into Equation (13) yields the expanded form of the function $Q(\theta \mid \theta^{(t)})$, as given in Equation (15).

$$\begin{aligned}Q(\theta \mid \theta^{(k)}) &= \sum_{k=1}^{K}\sum_{i=1}^{N} \gamma_{ki}\left[\log w_i - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)\right] \\ &= \sum_{k=1}^{K}\sum_{i=1}^{N} \gamma_{ki}\left[\log w_i - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\mathbf{x}_k - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)\right].\end{aligned} \tag{15}$$

Since the constants d and $\log(2\pi)$ of Equation (15) do not affect the optimisation, they can be omitted from the maximisation step.

To find the parameters maximising $Q(\theta \mid \theta^{(k)})$ under the constraint that the mixture weights sum to one, the Lagrange multiplier method is utilised. The optimisation problem is formulated in Equation (16) as

$$\begin{aligned}&\max\ Q(\theta \mid \theta^{(k)}) \\ &\text{subject to } \sum_{i=1}^{N} w_i = 1,\end{aligned} \tag{16}$$

where the function $Q(\theta \mid \theta^{(k)})$ is defined in Equation (15). The Lagrangian function, of the optimisation problem of Equation (16), is defined in Equation (17) by

$$\mathcal{L} = Q(\theta \mid \theta^{(k)}) + \beta\left(\sum_{i=1}^{N} w_i - 1\right), \tag{17}$$

where β represents the Lagrangian multiplier. The optimal parameters are located at the stationary points of the Lagrangian, obtained by solving the system of equations defined in (18).

$$\begin{aligned}&\nabla\mathcal{L} = 0 \\ &\nabla Q(\theta \mid \theta^{(k)}) + \beta\,\nabla\left(\sum_{i=1}^{N} w_i - 1\right) = 0.\end{aligned} \tag{18}$$

The parameter updates are obtained by differentiating the Lagrangian with respect to each of the model parameters w_i, $\mu_\mathbf{i}$ and $\boldsymbol{\Sigma}_\mathbf{i}$ as presented in Equations (19), (20) and (21) respectively.

Maximising w_i
Derivative of the Lagrangian with respect to w_i yields

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial w_i} = \frac{\partial}{\partial w_i}\left[Q(\theta \mid \theta^{(k)}) + \beta\left(\sum_{i=1}^{N} w_i - 1\right)\right] = \sum_{k=1}^{K}\frac{\gamma_{ki}}{w_i} + \beta = 0 \\ w_i = -\frac{1}{\beta}\sum_{k=1}^{K}\gamma_{ki}\end{aligned} \tag{19}$$

Applying the constraint $\sum_{i=1}^{N} w_i = 1$ provides

$$w_i^* = \frac{1}{K}\sum_{k=1}^{K}\gamma_{ki}.$$

Maximising $\boldsymbol{\mu}_i$
Derivative of the Lagrangian with respect to $\boldsymbol{\mu}_i$ yields

$$\begin{aligned}\frac{\partial Q}{\partial \boldsymbol{\mu}_i} = \frac{\partial}{\partial \boldsymbol{\mu}_i}\left[Q(\theta \mid \theta^{(k)}) + \beta\left(\sum_{i=1}^{N} w_i - 1\right)\right] = \sum_{k=1}^{K}\gamma_{ki}\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i) = 0 \\ \sum_{k=1}^{K}\gamma_{ki}\boldsymbol{\Sigma}_i^{-1}\mathbf{x}_k - \sum_{k=1}^{K}\gamma_{ki}\boldsymbol{\Sigma}_i^{-1}\boldsymbol{\mu}_i = 0 \\ \boldsymbol{\mu}_i^* = \frac{\sum_{k=1}^{K}\gamma_{ki}\mathbf{x}_k}{\sum_{k=1}^{K}\gamma_{ki}}.\end{aligned} \tag{20}$$

Maximising $\boldsymbol{\Sigma}_i$
Derivative of the Lagrangian with respect to $\boldsymbol{\Sigma}_i$ yields

$$\begin{aligned}\frac{\partial Q}{\partial \boldsymbol{\Sigma}_i} = \frac{\partial}{\partial \boldsymbol{\Sigma}_i}\left[Q(\theta \mid \theta^{(k)}) + \beta\left(\sum_{i=1}^{N} w_i - 1\right)\right] \\ = -\frac{1}{2}\sum_{k=1}^{K}\gamma_{ki}\left[\boldsymbol{\Sigma}_i^{-1} - \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}\right] = 0 \\ \boldsymbol{\Sigma}_i^* = \frac{\sum_{k=1}^{K}\gamma_{ki}(\mathbf{x}_k - \boldsymbol{\mu}_i)(\mathbf{x}_k - \boldsymbol{\mu}_i)^T}{\sum_{k=1}^{K}\gamma_{ki}}.\end{aligned} \tag{21}$$

The parameters maximising the function $Q(\theta \mid \theta^{(k)})$, derived in Equations (19), (20) and (21), are summarised in Equation (22).

$$w_i^* = \frac{1}{K}\sum_{k=1}^{K}\gamma_{ki}, \quad \boldsymbol{\mu}_i^* = \frac{\sum_{k=1}^{K}\gamma_{ki}\mathbf{x}_k}{\sum_{k=1}^{K}\gamma_{ki}}, \quad \boldsymbol{\Sigma}_i^* = \frac{\sum_{k=1}^{K}\gamma_{ki}(\mathbf{x}_k-\boldsymbol{\mu}_i)(\mathbf{x}_k-\boldsymbol{\mu}_i)^T}{\sum_{k=1}^{K}\gamma_{ki}},$$

where (22)

$$\gamma_{ki} = \frac{w_i^{(k)}\mathcal{N}(\mathbf{x}_k \mid \mu_i^{(k)}, \Sigma_i^{(k)})}{\sum_{j=1}^{N} w_j^{(k)}\mathcal{N}(\mathbf{x}_k \mid \mu_j^{(k)}, \Sigma_j^{(k)})}.$$

The E-step and M-step are repeated iteratively until convergence. Convergence is achieved when the log-likelihood function no longer changes significantly between iterations, satisfying the condition

$$|\ell(\theta^{(k+1)}) - \ell(\theta^{(k)})| < \epsilon,$$

where ϵ denotes a predefined threshold value.

Once the parameters of the GMM have been estimated with the expectation-maximisation algorithm, classification is performed by computing the posterior probability of a point $\mathbf{x}_k$ belonging to each component i, as defined in Equation (12). The predicted class label $\hat{y}_k$ is further assigned to the component with the highest posterior probability, as presented in Equation (23).

$$\hat{y}_k = \arg\max_i \gamma_{ki}. \tag{23}$$

Each point $\mathbf{x}_k$ is assigned to the most probable Gaussian component according to the computed responsibilities γ_{ki}. In the case of wood and leaf classification, where a two component GMM is applied, each component represents either the wood or the leaf class, resulting in a binary classification.

4 Research material and methods

This section presents the data and methods applied to classify wood and leaf components in individual tree point clouds. Section 4.1 introduces the three types of point cloud datasets applied in the study, consisting of terrestrial laser scanning, unmanned aerial vehicle and airborne laser scanning. Section 4.2 describes the classification methodology, which incorporates adaptive radius search, Gaussian mixture models and post-processing denoising, while Section 4.3 introduces the classification metrics applied in this study for result analysis.

4.1 LiDAR datasets

The study involves point cloud data collected with three different data acquisition geometries, TLS, UAV and ALS data. These datasets were selected to evaluate how acquisition geometry, point cloud density and viewing perspective affect the performance of wood and leaf classification. TLS data offers the highest point density and detailed coverage of tree structures. UAV data, captured from an aerial perspective at close range, provides a balance between canopy visibility and ground coverage. ALS data, acquired from a helicopter, represents a sparse and top-down scanning geometry, typically resulting in reduced penetration into dense foliage. However, this type of data collection enables large-scale analysis with relatively low acquisition time. Different datasets allow for an evaluation of the algorithm's performance for different data qualities and perspectives. In addition, the point clouds consist of diverse tree species, providing a basis for assessing the classification of different tree architectures.

Pre-processing was performed on all datasets to segment individual trees, resulting in individual tree point clouds acting as input for the proposed method. Each point within the tree point cloud contains information of the x, y, and z coordinates, along with intensity and manual classification labels. The classification label is a binary variable, with wood points labelled as 1 and leaf points as 0.

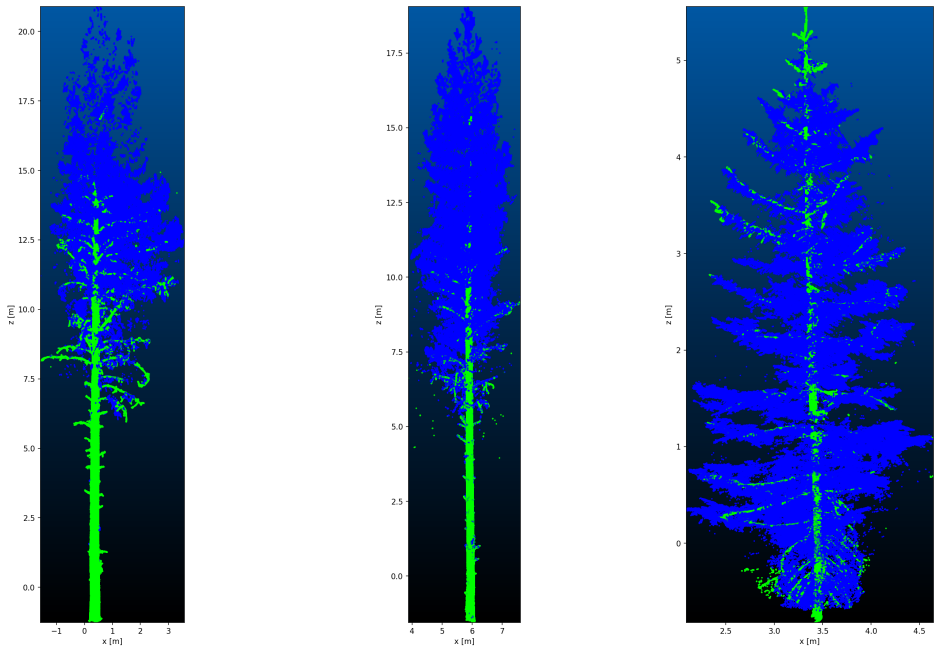
4.1.1 Terrestrial Laser Scanning

Liang et al. (2018) conducted terrestrial laser scanning surveys in April and May 2014 using a Leica HDS6100 scanner with a laser wavelength between 650–690 nm, achieving a spatial resolution of ±2mm at 25 m. The scanner's field of view was $360° \times 310°$, and data were collected in high density mode with an angular resolution of 0.036°, resulting in a point spacing of 15.7 mm at 25 m in both horizontal and vertical directions. Each full scan took approximately three minutes to complete (Liang et al., 2018). Data collection followed a multi-scan approach, with each sample plot scanned from five different positions: one at the centre and four in the quadrants, each spaced 11.3 m from the centre. Scanner positions were adjusted based on forest structure to avoid obstructions such as nearby tree stems (Liang et al., 2018).

The dataset includes TLS point clouds from three test plots representing different levels of forest complexity (easy, medium, and difficult). Each label contains two plots comprising single and multi-scan datasets. The TLS perspective provides strong

returns from lower branches and stems but is often incomplete in capturing the upper canopy structure. In less dense plots, tree visibility was high in both single and multi-scan data, whereas in dense forests, significant occlusion reduced tree completeness even in multi-scan datasets (Liang et al., 2018). The TLS dataset was made publicly available for non-profit research purposes (Liang et al., 2018).

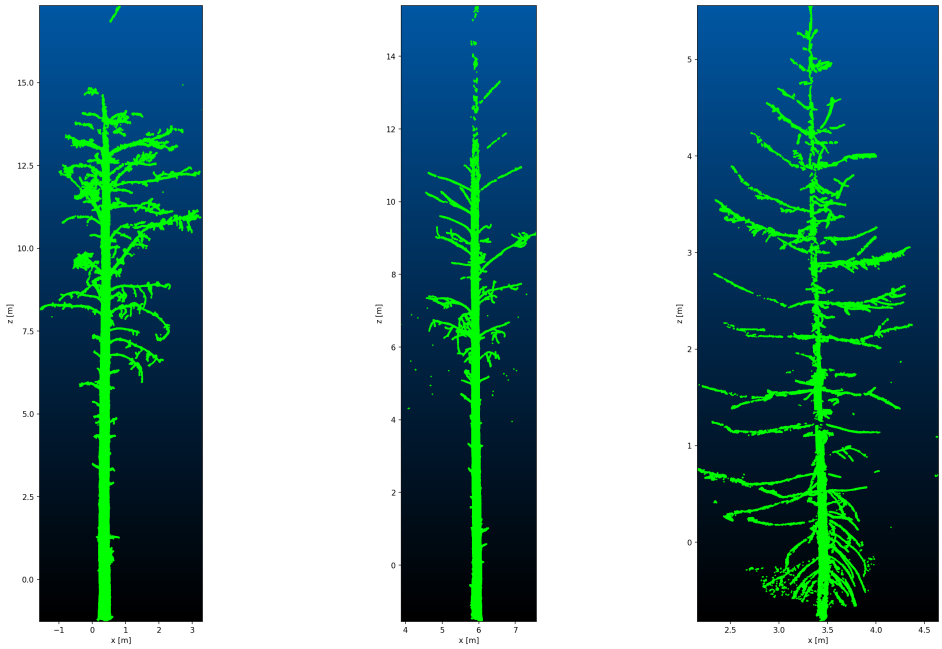
For this thesis, three test plots from each complexity category were selected as datasets for wood and leaf classification, all belonging to multi-scan TLS data. Three sample trees from each test plot are presented in Figure 1, alongside representations of the manually classified tree point clouds in Figure 2.



(a) Scots pine from plot 1 **(b)** Scots pine from plot 3 **(c)** Norway spruce from plot 5

Figure 1: TLS data of tree point clouds from three different forest plots

The TLS data includes trees with incomplete canopy or stem structures, as well as noise introduced by the scanner, visible in Figure 1a. Figure 1 shows manually classified trees, revealing missing sections in both canopy and stem, which become more evident after examining the classified trees presented in Figure 2. The missing stem shown in Figure 2a, together with the misclassified leaf points at the top of the Norway spruce in Figure 2c, demonstrate the challenges involved in wood and leaf classification. Further imperfections in the datasets, including missing points, mislabelled references and scanner-induced noise, may compromise the accuracy of the classification process.



(a) Scots pine from plot 1 **(b)** Scots pine from plot 3 **(c)** Norway spruce from plot 5

Figure 2: Manually classified tree point clouds from three different forest plots

4.1.2 Unmanned Aerial Vehicle

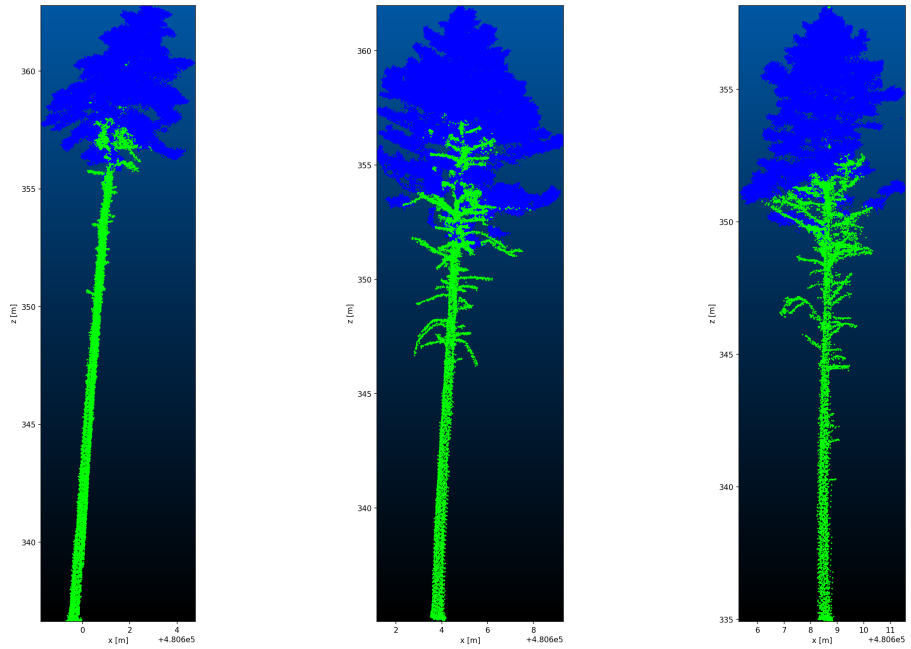
The performance of the wood and leaf separation algorithm on UAV point clouds is particularly important given the increasing use of UAV-based laser scanning for forest monitoring, as UAV scanners can cover larger areas faster than TLS scanners. Accurate functioning of the classification algorithm with UAV data supports the application of non-destructive AGB estimation, as demonstrated by Brede et al. (2022), provided that the resolution of the individual tree point clouds is sufficient.

Puliti et al. (2023) introduced the FOR-instance dataset, a curated and machine learning-ready dataset designed for benchmarking semantic and instance segmentation of forest point clouds. The dataset consists of five UAV-based laser scanning collections obtained from Norway, the Czech Republic, Austria, New Zealand, and Australia, representing different forest types. Data acquisition was conducted using Riegl VUX-1 UAV and Riegl MiniVUX-1 UAV sensors, capturing high-density 3D point clouds with varying sampling designs, including plot-based and wall-to-wall scanning approaches. The UAV flights followed two distinct patterns: a parallel flight pattern, where flight lines were aligned in a single direction and a double-grid flight pattern, where perpendicular flight lines were used to increase coverage and minimise occlusion effects (Puliti et al., 2023).

To ensure high-quality annotations, the dataset was manually labelled using CloudCompare (CC). Individual trees were assigned unique identifiers and further categorised into stems, woody branches, live branches, terrain, and low vegetation. The annotation process was conducted over six months by multiple annotators and later reviewed for accuracy. Additionally, reference spheres with a fixed 198 mm

radius were deployed in each plot for data registration, achieving an average alignment accuracy of 2.1 mm. The dataset is divided into development and test subsets, allowing researchers to evaluate segmentation methods (Puliti et al., 2023).

A Czech Republic dataset, referred to as the CULS dataset, was selected for this study due to its sparse forest structure and relatively low point density. The point density was approximately 2600 pts/m^2, allowing for more efficient processing (Puliti et al., 2023). The CULS dataset, referred to as the UAV dataset within this thesis, consisted of three forest plots containing a total of 47 Scots pines. Each tree was manually classified into stem, branch or leaf points. For the purpose of this study, stem and branch points were combined into a single wood class to enable wood and leaf classification. Examples of three trees from different plots in the UAV dataset are presented in Figure 3.

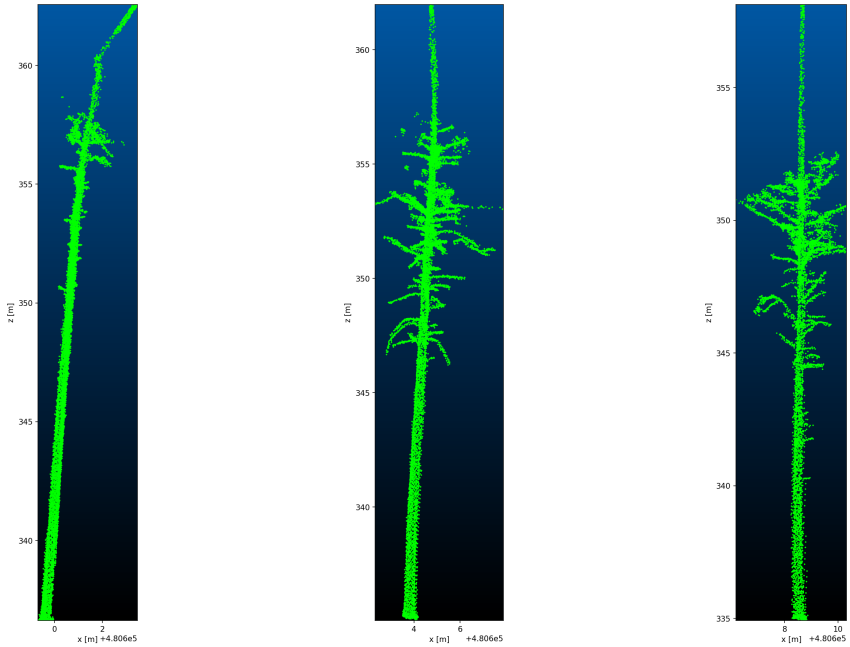


(a) Scots pine from plot 1 **(b)** Scots pine from plot 2 **(c)** Scots pine from plot 3

Figure 3: UAV data of Scots pine tree point clouds from three different forest plots of the CULS dataset

Figure 3 illustrates that the UAV dataset better captures the canopy details in comparison to the TLS dataset shown in Figure 1, with a greater focus on the canopy than the stem. This allows evaluation of the robustness of the method across different geometries. Figure 3c further demonstrates that the point cloud is noticeably sparse. This lower point density may enable faster and more efficient feature computation, although it could potentially reduce the level of geometric detail. Furthermore, a larger neighbourhood radius may be required to capture the geometric structure in sparser point clouds. The neighbourhood search radius will be discussed further in Section 4.2.

Nevertheless, the UAV dataset offers several advantages. Trees are captured from an aerial perspective, with minimal background noise and high-quality manual classifications. This perspective is possible due to the increasing capability of UAV platforms to fly at low altitudes while acquiring data with sufficiently high resolution. The manually classified trees from UAV forest plots are illustrated in Figure 4. Even though the point clouds are well captured and accurately classified, Figure 4 reveals that the smallest branches are not properly detected and are therefore misclassified as leaf points.



(a) Scots pine from plot 1 **(b)** Scots pine from plot 2 **(c)** Scots pine from plot 3

Figure 4: Manually classified Scots pine trees of three forest plots of the CULS dataset

4.1.3 Airborne Laser Scanning

Ruoppa et al. (2025) utilised a high-resolution airborne laser scanning dataset collected on June 22, 2021, at the scan forest test site near Evo, Finland. The data was acquired using the HeliALS-TW system, an advanced laser scanning system developed by the Finnish Geospatial Research Institute (FGI). This system integrates three Riegl LiDAR scanners—VUX-1HA, miniVUX-3UAV, and VQ-840-G—referred to as scanners 1, 2, and 3, respectively. To ensure accurate positioning and orientation, the system is equipped with a GNSS-IMU navigation system comprising a NovAtel ISA-100C IMU, a NovAtel PwrPak7 GNSS receiver and a NovAtel GNSS-850 antenna (Ruoppa et al., 2025).

The helicopter flew at a low altitude of approximately 80 metres above ground along predefined circular flight paths over multiple 55 m diameter test sites to capture detailed point clouds from two perpendicular directions. Scanners 1 and 2 were mounted at a 15-degree forward tilt, providing cross-track profiles, while scanner 3 used a conical

scanning approach with a 40-degree cross-track and 28-degree flight direction angle. The system's trajectory was processed using Waypoint Inertial Explorer, with a single virtual GNSS base station positioned near the Evo research forest. The raw LiDAR scans were then georeferenced using Riegl RiPROCESS software, incorporating GNSS and IMU data to ensure precise spatial alignment (Ruoppa et al., 2025).

The ALS dataset consists of two forest plots containing a total of 356 trees. Like the UAV data, the ALS data were captured using an airborne scanner, but scanning was performed directly from above. Although this top-down perspective effectively records the position and general shape of trees, the resulting point clouds are relatively sparser compared to the UAV data, which could present further challenges for extracting geometric features. Moreover, the three scanners of the helicopter-mounted system operated with distinct intensities, requiring wood and leaf separation to be performed individually for each intensity. Figure 5 presents two example trees from different ALS forest plots, while Figure 6 shows the same trees with manual classification, demonstrating accurate capture of tree structure while highlighting point cloud sparsity.

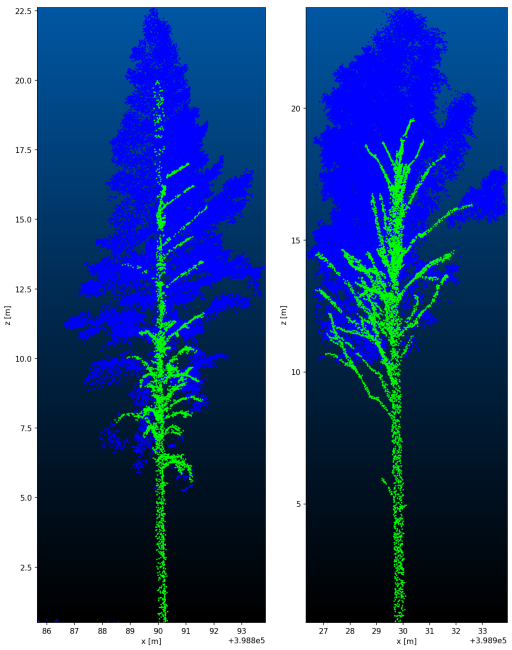


Figure 5: Two different tree point clouds of ALS dataset

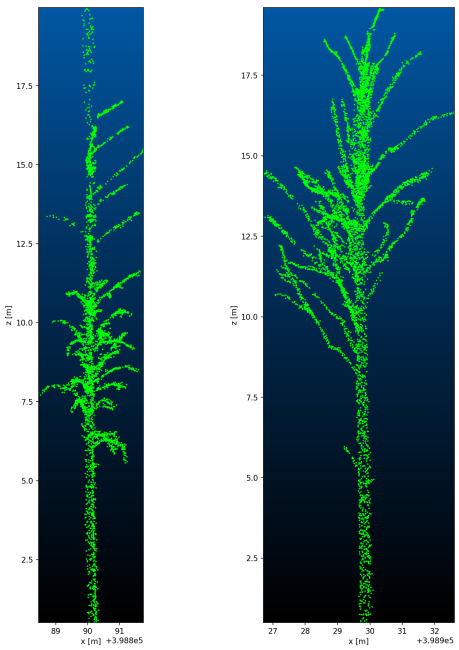


Figure 6: Manual classifications of the two different tree point clouds of ALS dataset

4.2 Methods

Figure 7 represents the overall workflow of the wood and leaf separation algorithm proposed in this thesis. The workflow consists of four main steps, beginning with an adaptive radius search for feature extraction, followed by classification using a two-component Gaussian mixture model, then computation of a weighted sum of feature classifications to obtain the final label, and concluding with post-processing denoising to eliminate residual noise. Each of these steps is described in detail in Sections 4.2.1, 4.2.2, 4.2.3, and 4.2.4, respectively.

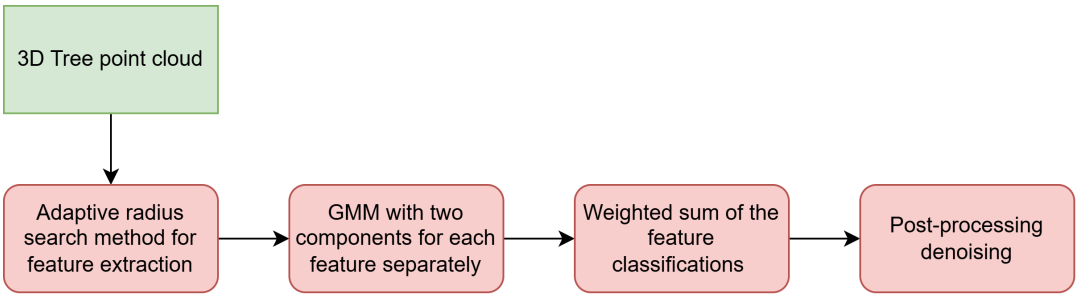


Figure 7: Flowchart of the wood and leaf separation algorithm

4.2.1 Optimal radius search

The selection of the optimal radius follows the approach of Demantké et al. (2011), where the optimal radius is determined as the one minimising Shannon's entropy. Demantké et al. (2011) originally developed the adaptive radius method as an approach for extracting geometric features from building point clouds, and the method has not previously been applied to forest applications such as wood and leaf classification at a tree level. A related method by Zhu et al. (2018) implemented a variation that selects an optimal radius from three fixed values (0.2 m, 0.3 m, and 0.4 m) by identifying the radius that minimises entropy.

In this thesis, the adaptive radius method aims to determine the optimal local neighbourhood for each point by identifying the radius that minimises entropy within a specified search range, as described in the function *ProcessSinglePoint* of Algorithm 1. For each point in the dataset, a k-nearest neighbour search is performed using a KDTree to determine a local neighbourhood of ten neighbours. The maximum distance to the ten nearest neighbours defines the initial search radius, which serves as the lower bound $r_{\text{min}}^{\text{lim}}$, ensuring principal component analysis (Demantké et al., 2011). However, any initial search radius found lower than a threshold of 10 cm is set to the latter to ensure that neighbourhoods effectively capture geometric structures (Shcherbcheva et al., 2023; Zhu et al., 2018).

The upper bound $r_{\text{max}}^{\text{lim}}$ is defined in relation to the point cloud density. For TLS data with higher point density, an upper bound of 0.5 m is sufficient. However, due to differences in point cloud resolution described in Section 4.1, the upper bound varies between datasets. Sparser point clouds from UAV and ALS require a larger bound of 1.5 m to better capture the local geometric structure of the neighbourhood.

To determine the optimal neighbourhood size, the algorithm iterates through potential radii within the defined range, incrementing by a predefined step size s. The selection of the step size represents a trade-off between accuracy and computational efficiency. Although parallel computation was implemented, processing large datasets remains computationally intensive and time-consuming.

For each candidate radius, the covariance matrix of the local neighbourhood is computed, and its eigenvalues are obtained. These eigenvalues are applied to compute Shannon's entropy, and the radius minimising entropy is selected as the optimal neighbourhood radius r^*. Once the optimal neighbourhood is determined, the covariance matrix is recomputed for the optimised neighbourhood, and its eigenvalues are obtained again. From these values, a set of geometric features is derived, consisting of curvature (C_{3D}), linearity (L_{3D}), anisotropy (A_{3D}), sphericity (S_{3D}), planarity (P_{3D}), and verticality (V_{3D}). Additionally, the 3D point density (D_{3D}) is computed based on the number of neighbours within the selected radius. The final output of Algorithm 1 is a vector containing these computed features for a single point.

The overall feature computation for the entire dataset follows Algorithm 2. This process initialises an empty feature set and iterates over all points in the dataset, calling the ProcessSinglePoint function in Algorithm 1 for each point to extract geometric features with the optimal search radius. The resulting feature vectors are stored in a data frame, where each row corresponds to a point in the dataset and each column represents a different geometric feature. The final output is a dataset containing point-wise geometric features, serving as input for further classification. Due to its point-by-point analysis, the adaptive radius search algorithm is computationally intensive; thus, parallelisation was required to improve computation speed.

Algorithm 1 Pseudocode for ProcessSinglePoint function

Require: A query point $\mathbf{x_i} = [x_i, y_i, z_i]^T$, dataset $\mathbf{x} = [(x_1, y_1, z_1), \ldots, (x_n, y_n.z_n)]^T$, $\mathcal{T}$ = KDTree, s = Step size

$r_{\min}^{\lim} \leftarrow r_1^*$, $r_{\max}^{\lim} \leftarrow r_2^*$
$n \leftarrow$ Length of the dataset $\mathbf{x}$
$d \leftarrow$ Distances to n neighbours by applying $\mathcal{T}$
$r_{\min} \leftarrow \max(d)$
$r^* \leftarrow \infty$, $E^* \leftarrow \infty$
if $r_{\min} \leq r_{\min}^{\lim}$ **then**
    $r_{\min} \leftarrow r_{\min}^{\lim}$
end if
if $r_{\min} < r_{\max}^{\lim}$ **then**
    $r_{\max} \leftarrow r_{\max}^{\lim}$
    for r in range$[r_{\min},\ r_{\max},\ s]$ **do**
        inds_$r \leftarrow$ Indices of the neighbouring points within the radius r
        $X \leftarrow \mathbf{x}[\text{inds\_}r]$
        $k \leftarrow$ Length of X
        $C \leftarrow \frac{1}{n}[(X - \frac{1}{k}\sum_{i=1}^{k} X_i)^T (X - \frac{1}{k}\sum_{i=1}^{k} X_i)]$
        $\lambda \leftarrow$ Sorted eigenvalues of C
        $\sigma \leftarrow \sqrt{\lambda}$
        $\alpha \leftarrow \left[\frac{\sigma_1 - \sigma_2}{\sigma_1 + \sigma_3}, \frac{\sigma_2 - \sigma_3}{\sigma_1 + \sigma_3}, \frac{\sigma_3}{\sigma_1 + \sigma_3}\right]$
        $E \leftarrow -\sum_{i=1}^{3} \alpha_i \log \alpha_i$
        if $E < E^*$ **then**
            $E^* \leftarrow E$
            $r^* \leftarrow r$
        end if
    end for
else
    $r^* \leftarrow r_{\min}$
end if
inds_$r^* \leftarrow$ Indices of the neighbouring points within the optimal radius r^*
$X^* \leftarrow \mathbf{x}[\text{inds\_}r^*]$, $k^* \leftarrow$ Length of X
$C^* \leftarrow \frac{1}{n}[(X^* - \frac{1}{k}\sum_{i=1}^{k} X_i^*)^T (X^* - \frac{1}{k}\sum_{i=1}^{k} X_i^*)]$
$\lambda^* \leftarrow$ Sorted eigenvalues of C^*, $\sigma_1 \leftarrow \sqrt{\lambda_1}$
$C_{3D} \leftarrow \frac{\min(\lambda_i)}{\sum \lambda_i}$
$L_{3D} \leftarrow \frac{\lambda_1 - \lambda_2}{\lambda_1}$
$A_{3D} \leftarrow \frac{\lambda_1 - \lambda_3}{\lambda_1}$
$S_{3D} \leftarrow \frac{\lambda_3}{\lambda_1}$
$P_{3D} \leftarrow \frac{\lambda_2 - \lambda_3}{\lambda_1}$
$V_{3D} \leftarrow 1 - |N_z|$
$D_{3D} \leftarrow \frac{n}{\frac{4}{3}\pi(r_{\min})^3}$
return $[C_{3D}, L_{3D}, A_{3D}, S_{3D}, P_{3D}, V_{3D}, D_{3D}, \sigma_1]$

Algorithm 2 Pseudocode for computing features with adaptive radius

Require: dataset $\mathcal{X} = [(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)]^T$, s^*=Step size

$n \leftarrow 11$, $s \leftarrow s^*$
$\mathcal{T} \leftarrow$ KDTree from dataset **x**
$\mathcal{F} \leftarrow \emptyset$, Initialisation of the feature set
for all $\mathbf{x_i} \in$ **x do**
    $\mathcal{F}_i \leftarrow$ ProcessSinglePoint$(x_i, \mathcal{X}, \mathcal{T}, n, s)$
    Append $\mathcal{F}_i$ to $\mathcal{F}$
end for
Define column names C_{names}
$\mathcal{D} \leftarrow$ DataFrame$(\mathcal{F}, C_{\text{names}})$
return $\mathcal{D}$

4.2.2 Classification utilising Gaussian Mixture Model

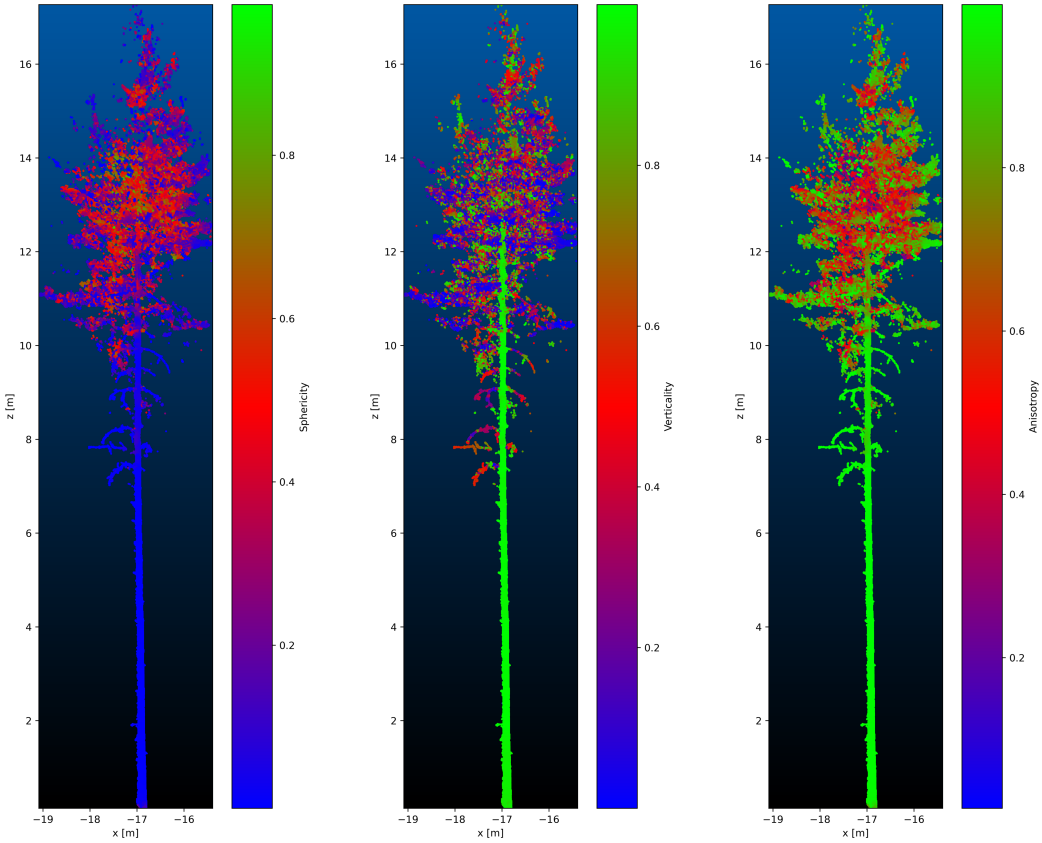
After the geometric features have been extracted, classification is performed by applying an independent cross-scale salience method, where each feature, including intensity, is considered independently to assess whether a point belongs to the wood or leaf class. For each feature type, a Gaussian mixture model with two components is fitted to model the distribution of feature values across the point cloud. Points are further classified based on whether the feature value is above or below the mean of the two GMM component means. As each feature contributes with varying influence to the classification, a weighted sum approach is applied to obtain the final classification.

4.2.3 Weighted sum of features

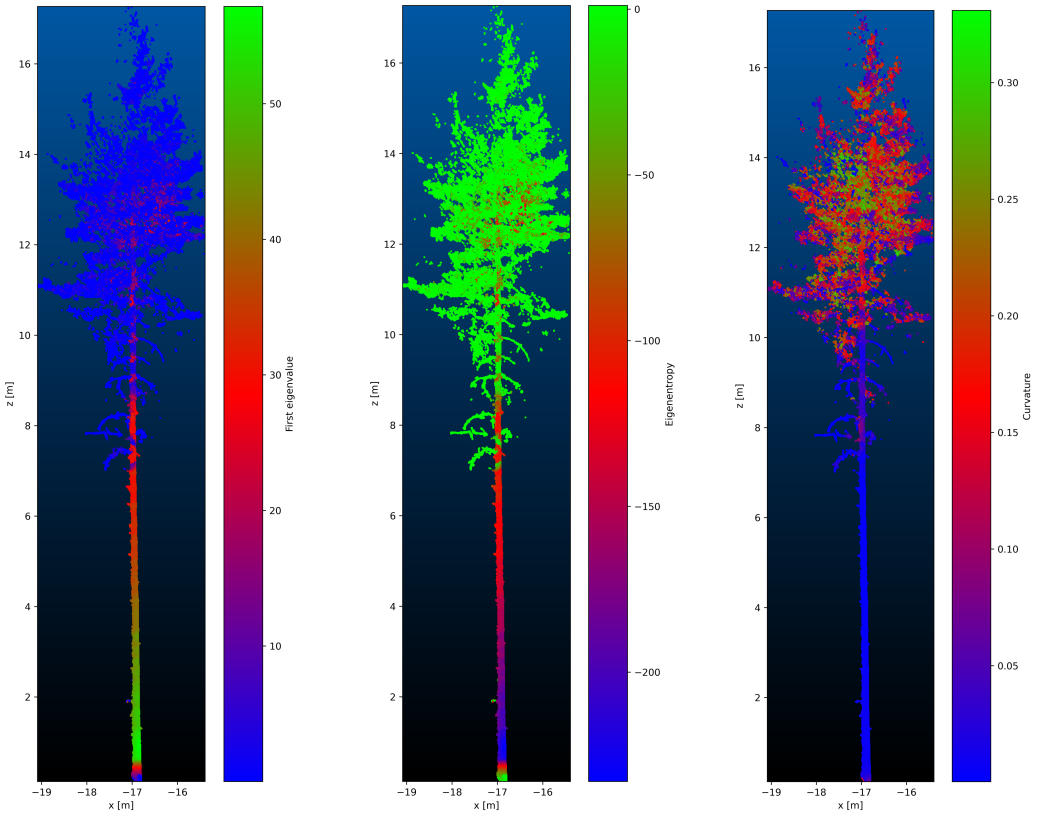
The weighted sum approach is applied to assign each feature a weight based on its significance in distinguishing wood and leaf components, additionally allowing the algorithm to adapt to different geometries and data acquisitions. Feature weights depend on the dataset as well as its resolution. The significance of each feature was determined based on its distribution and whether it contributes to distinguishing wood from leaf more than other features. The weighted sum is implemented in Equation (24) as

$$S = w_1 F_1 + w_2 F_2 + \ldots + w_n F_n, \tag{24}$$

where F_i denotes the feature classification and w_i the corresponding weight assigned to it. Several features were tested, and those with poor performance were excluded from the analysis. Experimentation showed that features such as eigenvalues, the sum of eigenvalues, eigenentropy, and omnivariance did not significantly contribute to the separation and were therefore removed, consistent with the findings of Shcherbcheva et al. (2023). A selection of some included and excluded features is illustrated in Figure 8 to demonstrate the selection process.



(a) Sphericity **(b)** Verticality **(c)** Anisotropy



(d) First eigenvalue **(e)** Eigenentropy **(f)** Curvature

Figure 8: Geometric features of Scots pine tree of TLS dataset

Figure 8 presents example features of a Scots pine tree from the TLS dataset. The first eigenvalue and eigenentropy, shown in Figures 8d and 8e, respectively, contributed little to the classification. In both of these features, the lower part of the stem exhibited different feature values compared to the upper part. Additionally, there was no clear distinction between the leaves in the canopy and the branches, leading to the exclusion of these features from the wood and leaf classification.

Curvature, presented in Figure 8f, did not effectively distinguish between leaves and branches in the upper part of the tree. Nonetheless, it was included in the analysis given that it successfully captured the stem structure along with the lower branches. Verticality, as seen in Figure 8b, captured the stem accurately up to the top of the tree but was less effective in detecting smaller branches. Since the tree stem is generally difficult to capture, verticality provided valuable information for the classification, resulting in its inclusion in the feature set.

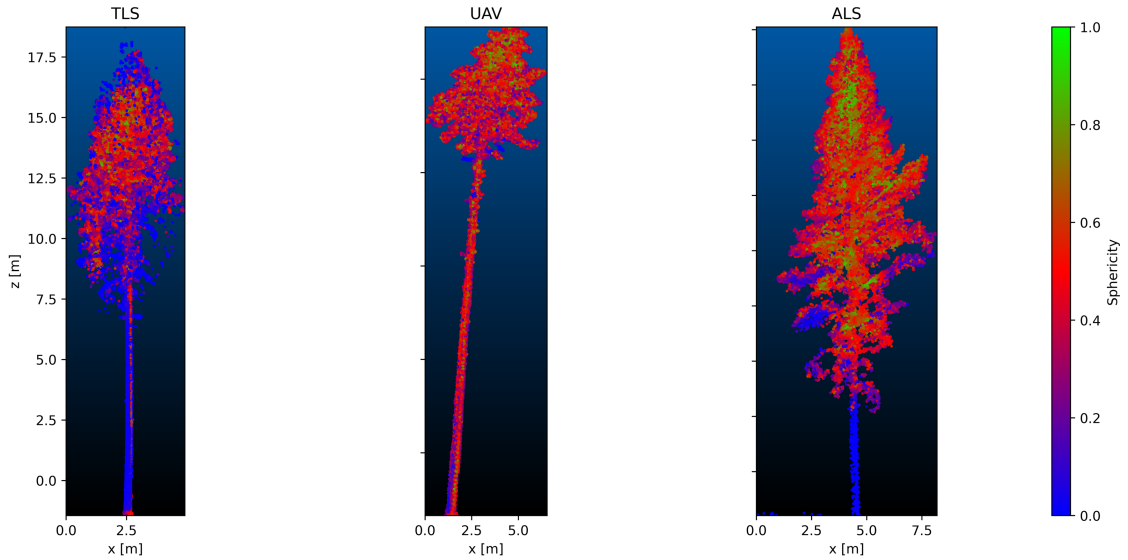
Sphericity and anisotropy, illustrated in Figures 8a and 8c, respectively, were able to capture both the stem and the branches in the upper parts of the canopy and were therefore included in the wood and leaf separation.

In general, the selection of features was implemented by evaluating the classification performance of each feature individually. If a feature alone achieved a classification accuracy of approximately 70% or higher, it was included for the final classification, and its weight was determined based on its distribution. Feature weights were adjusted independently for each feature and varied between datasets depending on the point cloud density. The assigned weights for each data type are summarised in Table 2.

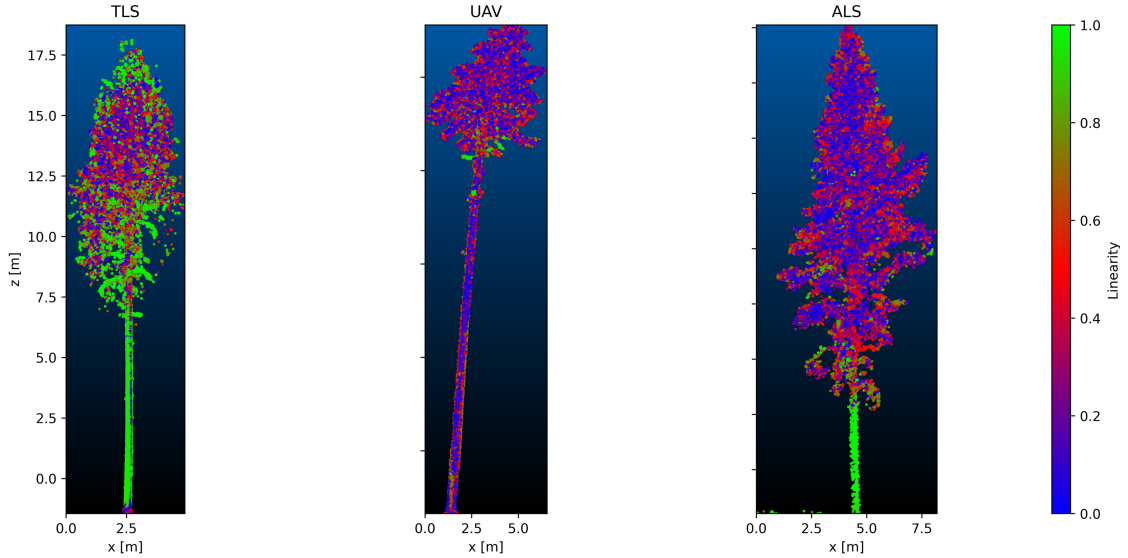
Weights	TLS	UAV	ALS
Curvature	1.0	0.5	1.0
Linearity	0.0	1.5	1.0
Anisotropy	3.0	1.5	1.0
Verticality	2.0	3.0	3.5
Local point density	2.0	0.5	0.0
First principal component	2.0	1.5	2.0
Sphericity	3.0	1.0	0.5
Planarity	0.5	3.5	2.0
Sum	13.5	13.0	11.0

Table 2: Weights of each feature in each dataset

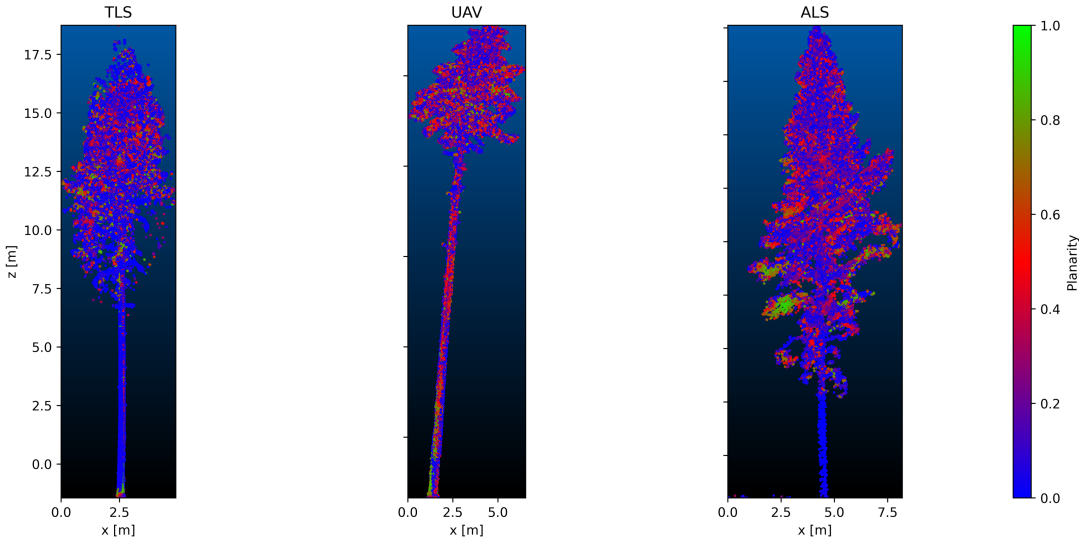
By examining Table 2, it becomes evident that the feature weights are dependent on the dataset and, more specifically, on the point cloud density. Variations in feature performance across different data types are illustrated in Figure 9. For instance, Figure 9a demonstrates that while sphericity is effective for TLS data, it fails to capture any structural components in the UAV or ALS data. Linearity and planarity, shown in Figures 9b and 9c, respectively, are more informative in the UAV and ALS point clouds than in the TLS dataset but tend to misclassify wood points in the process. Despite the misclassifications, these features are still assigned a higher weight, as the sparsity of UAV and ALS datasets limits the ability of geometric features to perfectly distinguish between wood and leaf structures. In the denser TLS dataset, linearity was assigned a weight of zero due to its tendency to misclassify parts of the trunk. Planarity was similarly assigned a smaller weight for this reason. Since distinguishing wood and leaf points is less challenging in TLS data, feature misclassifications have a larger impact than in UAV and ALS datasets.



(a) Comparison of sphericity feature



(b) Comparison of linearity feature



(c) Comparison of planarity feature

Figure 9: Comparison of Scots pine trees from the TLS, UAV and ALS datasets using different features

In general, the feature weights were determined by analysing the distribution of each feature separately for each dataset. Figure 10 illustrates the distributions of the geometric features across 50 trees in the TLS data.

In the TLS dataset, wood points outnumber leaf points; hence, features showing a strong association with wood are prioritised in the analysis. As shown in Figure 10, linearity does not exhibit a clear dominance of wood points in its distribution. Instead, it displays two distinct peaks, representing wood and leaf points, leading to the exclusion of this feature. Anisotropy and sphericity share a similar pattern, with one peak representing wood points and a smaller secondary peak indicating leaf points. This clear separation between the two classes aligns with the expected distribution of wood and leaf points and led to these features being assigned the highest weights. Verticality, local point density, and the first principal component were each assigned a weight of one, as reported in Table 2. These features successfully capture many wood points but fail to detect a sufficient number of leaf points, leading to misclassification of some leaf points as wood. This is evident from their distributions in Figure 10, which are heavily skewed toward wood-dominated values. Finally, planarity was given the lowest non-zero weight due to its distribution, which displays a clear wood peak but is otherwise relatively uniform.

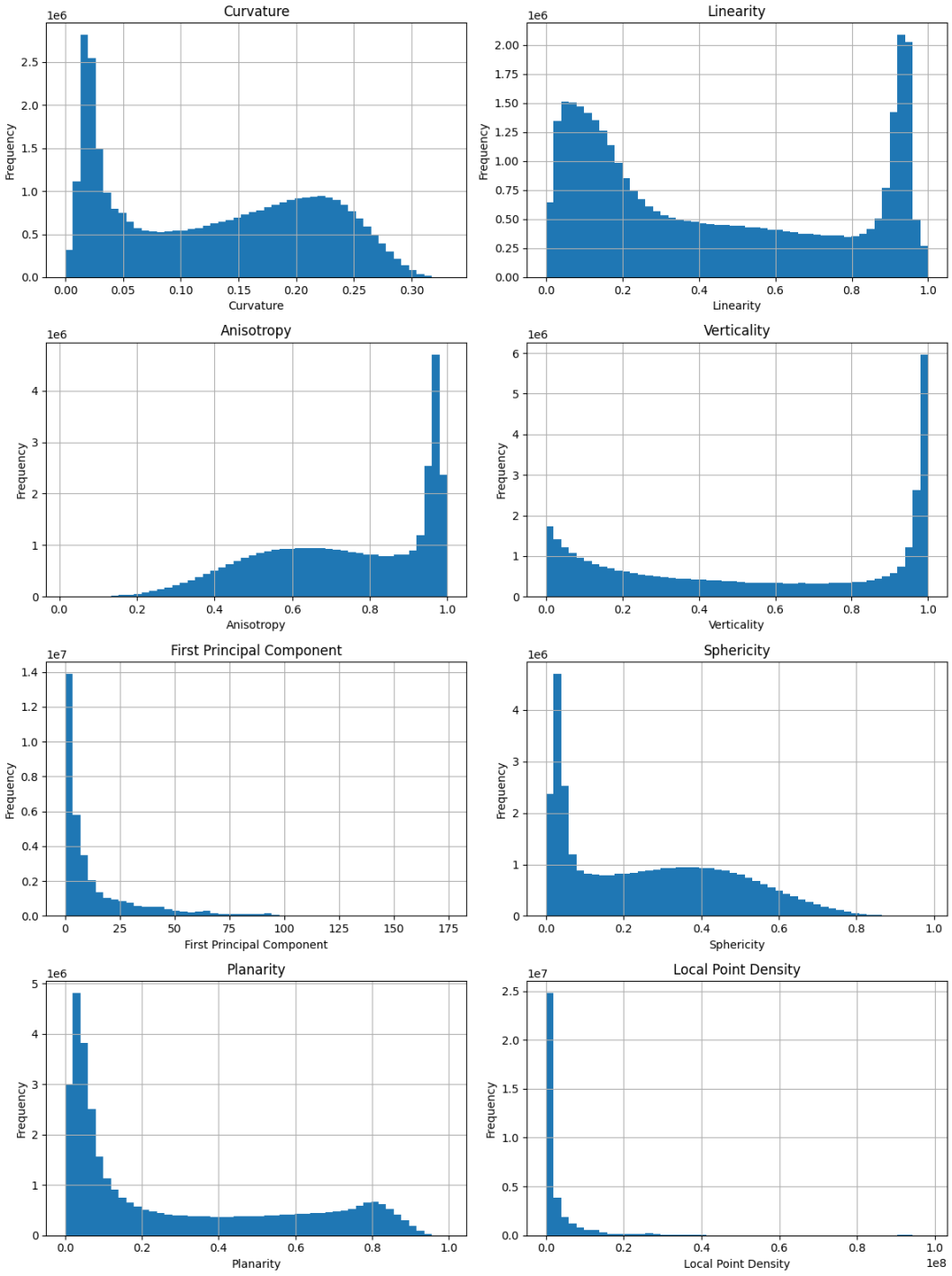


Figure 10: Histograms of geometric features of TLS data

After determining the feature weights, a weighted sum was computed for each point to classify it as wood or leaf. In the TLS dataset, points with a weighted sum of 8 or more were classified as wood. For the UAV data, the threshold was increased to 11, and for the ALS data, it was set to 9. These thresholds were determined based on the point cloud density of each dataset. Specifically, TLS, with the highest density,

required only 8 out of a possible 13.5, whereas the sparser UAV dataset required 11 out of 13. In comparison, ALS, with a similar resolution to UAV, used a threshold of 9 out of 11. This indicates that as point cloud density decreases, a higher proportion of features must agree for a point to be classified as wood. Such an approach ensures that the algorithm maintains classification reliability and avoids incorrectly labelling all points as wood in sparser datasets.

During testing, it was observed that features with left-skewed distributions occasionally caused the GMM to assign clusters in reverse, resulting in wood points being classified as leaf and vice versa. To address this issue, the classification direction for these features was adjusted manually. These adjustments varied across datasets due to differences in feature distributions between the TLS, UAV, and ALS point clouds.

4.2.4 Post-processing denoising

The final step in the wood and leaf separation algorithm was post-processing denoising. The predicted labels, indicating whether each point belonged to the wood or leaf class, were post-processed to improve the overall quality of the results. This step involved denoising the classified point cloud to eliminate noisy points that may have been misclassified. The distribution of the classified wood points was examined by computing the mean distance to each point's k-nearest neighbours. If this mean distance exceeded a specified threshold, defined as a function of the global mean and standard deviation, the point was identified as an outlier and reclassified as leaf. Specifically, a point was considered noise if it lay farther than a threshold distance from its 20 nearest neighbours, where the threshold was computed as the average of all such distances plus 1.7 times their standard deviation (Shcherbcheva et al., 2023). Following denoising, the updated point cloud was saved, including both the classification labels and the computed geometric features. Lastly, classification performance was evaluated by comparing predicted labels against the manual classification by applying accuracy, precision, recall and F1-score metrics.

4.3 Classification metrics

This section introduces the key performance metrics applied in this study to evaluate different aspects of binary classification performance. These metrics, consisting of accuracy, precision, recall, and F1-score, are essential for understanding the strengths and limitations of the separation algorithm.

Accuracy measures the overall correctness of the model and is defined as the ratio of correctly predicted instances to the total number of predictions, as shown in Equation (25).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{25}$$

In Equation (25), TP, TN, FP, and FN respectively denote true positives, true negatives, false positives, and false negatives. While accuracy provides a general measure of performance, it can be misleading in cases of class imbalance. For instance, overall accuracy may remain high even if the minority class is poorly predicted.

To better assess classification reliability, precision and recall are applied. Precision, defined in Equation (26),

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{26}$$

indicates the proportion of predicted positive instances that are truly positive. High precision implies that the algorithm produces few false positives. Recall, defined in Equation (27),

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{27}$$

measures the model's effectiveness in capturing all actual positive instances. High recall implies successful identification of most true positives. To balance the trade-off between precision and recall, the F1-score is employed. It is computed as the harmonic mean of precision and recall, as defined in Equation (28),

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}. \tag{28}$$

The F1-score is particularly beneficial for evaluating classifiers on imbalanced datasets, as it provides a single measure considering both false positives and false negatives. It is exclusively high when both precision and recall are high, yielding a robust metric for summarising classification effectiveness on the positive class.

5 Results

This chapter presents the performance of the wood and leaf classification algorithm on three LiDAR datasets — TLS, UAV, and ALS. The algorithm's effectiveness is evaluated on each dataset by applying performance metrics. For each data source, the metrics are reported along with their standard deviations, and the results are further compared. As tree species may influence the performance of the wood and leaf separation algorithm, the analysis also includes a species-wise evaluation focusing on the three most common tree species in Finland: Scots pine (Pinus sylvestris), Norway spruce (Picea abies), and silver birch (Betula pendula). The analysis concludes by examining how the point cloud density affects the performance of the proposed wood and leaf separation algorithm.

5.1 Classification results across LiDAR datasets

The overall classification results of the wood and leaf separation algorithm using three different LiDAR data sources are summarised in Table 3 along with the standard deviation of each metric. Performance is evaluated by applying accuracy, precision, recall, and F1-score metrics.

Data Source	Accuracy	Precision	Recall	F1-Score
TLS	0.794 ± 0.104	0.739 ± 0.300	0.635 ± 0.202	0.665 ± 0.250
UAV	0.916 ± 0.019	0.764 ± 0.124	0.426 ± 0.091	0.543 ± 0.097
ALS	0.875 ± 0.046	0.091 ± 0.117	0.103 ± 0.091	0.086 ± 0.094

Table 3: Classification performance metrics with standard deviations for each data source, considering all species combined

Table 3 shows TLS data achieving an average accuracy of 0.794, precision of 0.739, recall of 0.635, and F1-score of 0.665. Approximately 79.4% of the points are correctly classified, as indicated by these values. A precision of 73.9% suggests that most wood predictions are correctly labelled, while a recall of 63.5% indicates the classifier detects approximately two-thirds of actual wood points. The F1-score of 66.5% reflects this balance between precision and recall. Among the three data sources, TLS achieves the highest recall and a relatively high F1-score, indicating that a higher point density, combined with a ground-level scanning geometry, facilitates more accurate wood point detection. However, moderate precision implies that some wood points were incorrectly classified, potentially due to the complexity of certain species such as the dense foliage of Norway spruce, which can hinder laser beam penetration to the wood material.

For UAV data, the classifier achieves the highest overall accuracy of 0.916, with precision also relatively high at 0.764; however, recall drops to 0.426. This inconsistency suggests that while the classifier accurately labels wood when it is predicted as such, it fails to detect many actual wood points. The F1-score of 0.543 is

lower than that of TLS despite UAV's higher accuracy, suggesting UAV data leads to incomplete predictions by classifying only the most evident wood structures. The imbalance between wood and leaf points inflates accuracy, as wood points dominate within the point cloud. Limitations in capturing inner canopy structure and finer wood details likely contribute to the reduced recall in UAV data.

ALS results are particularly distinct. Despite an accuracy of 0.875, precision and recall are significantly lower at 0.091 and 0.103. The resulting F1-score is only 0.086, indicating a failure to classify wood points effectively. The classifier appears to label nearly all points as wood, an approach that yields high accuracy when wood points dominate the dataset but fails to identify actual wood instances. A precision of 9.1% suggests that only one in eleven wood predictions is correct, and a recall of 10.3% indicates that a vast majority of actual wood points are misclassified. The near-zero F1-score confirms the classifier does not distinguish wood and leaf components correctly in the ALS data. The outcome is likely due to the sparse ALS point cloud, lacking sufficient detail to support effective geometric feature extraction.

The standard deviations reported in Table 3 provide insight into the consistency of the classification. UAV metrics demonstrate the lowest variability across trees, with accuracy varying by only ±0.019, and precision, recall and F1-score remaining relatively stable. This consistency indicates that the classifier's performance is predictable across the UAV dataset, suggesting that the tree point clouds within the dataset exhibit similar structural characteristics. This is made clear by the exclusive presence of Scots pines. TLS metrics exhibit greater variability, with a precision standard deviation of ±0.300 suggesting notable differences in classification quality across trees. Recall also fluctuates more in TLS data, with a standard deviation of ±0.202 reflecting inconsistent wood detection. These inconsistencies likely result from the presence of different geometric structures of tree point clouds and species variation within the dataset.

Figure 11 illustrates the distribution of classification metrics for each data source. TLS box plots show a wider spread, confirming variation in performance across tree species. UAV box plots are tighter, consistent with the low standard deviations and the homogeneous composition of the UAV dataset. ALS box plots reflect low median values and substantial variability, particularly in precision and recall, indicating the classifier's inconsistency in distinguishing between wood and leaf classes across different tree species.

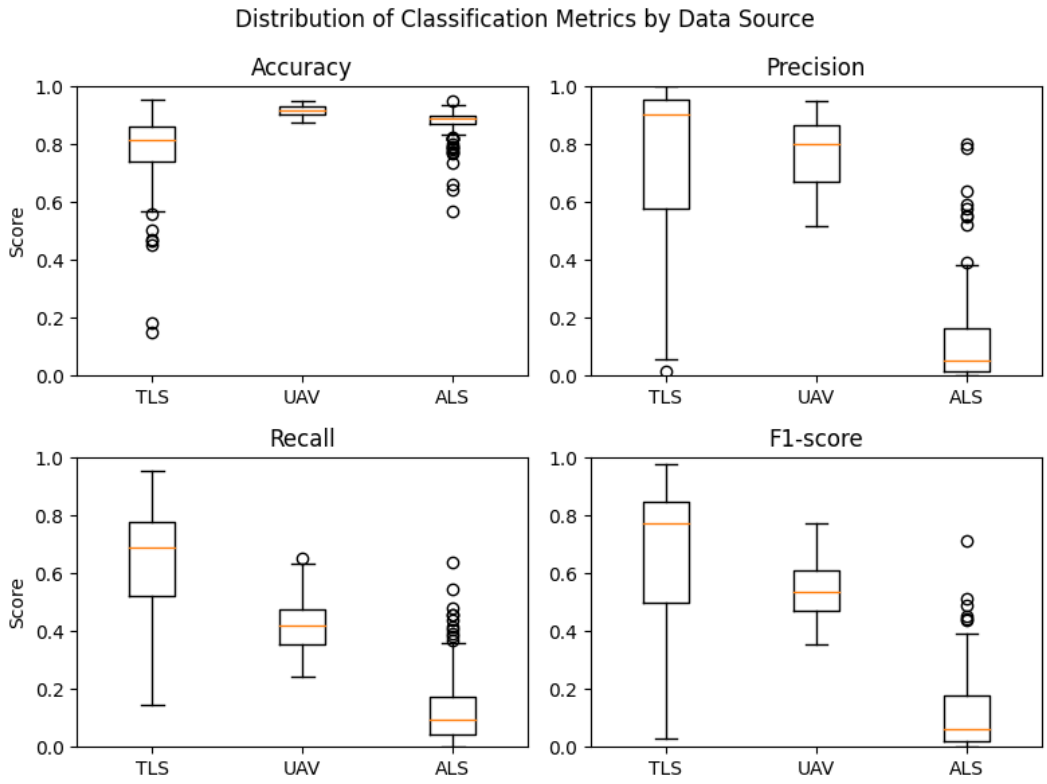


Figure 11: Overall distribution of classification metrics by data source

Results confirm that both the species and density of point cloud data significantly influence classification performance. Dense, high-quality data facilitates accurate wood and leaf separation, while sparse data leads to class imbalance issues and limited feature extraction. As point density decreases, the distance between points increases and a larger neighbourhood radius is required. As a result, the larger radius captures nearby structures and blurs the distinction between wood and leaf components in the case of a branch.

The variation observed in Figure 11 and the standard deviations in Table 3 between TLS, ALS, and UAV results can also be explained by species composition. The UAV dataset includes only Scots pines, whereas TLS and ALS also include Norway spruces and silver birches. Norway spruce, with densely packed needles and complex canopy structures, provides a more challenging classification. The high foliage density reduces laser penetration, often occluding the underlying wood and limiting the quality of geometric features extracted for classification. This structural complexity affects both the data quality and the effectiveness of the classification method, contributing to reduced consistency in performance across individual trees. Additionally, intensity offers limited contribution due to mixed values that fail to clearly separate wood and leaf components. As a result, the method relies primarily on geometric features, increasing the challenge in structurally complex species such as Norway spruce.

In contrast, Scots pine is more straightforward to separate due to its open branching and tufted needles. The presence of Norway spruces in TLS datasets likely contributes

to performance fluctuations. These observations support the importance of species-wise performance evaluation in order to enhance the understanding of algorithm behaviour across tree and data types.

5.2 Classification results across tree species

To investigate the effect of tree species on classification performance, a species-wise analysis was conducted. The number of trees representing each species in the datasets is presented in Table 4.

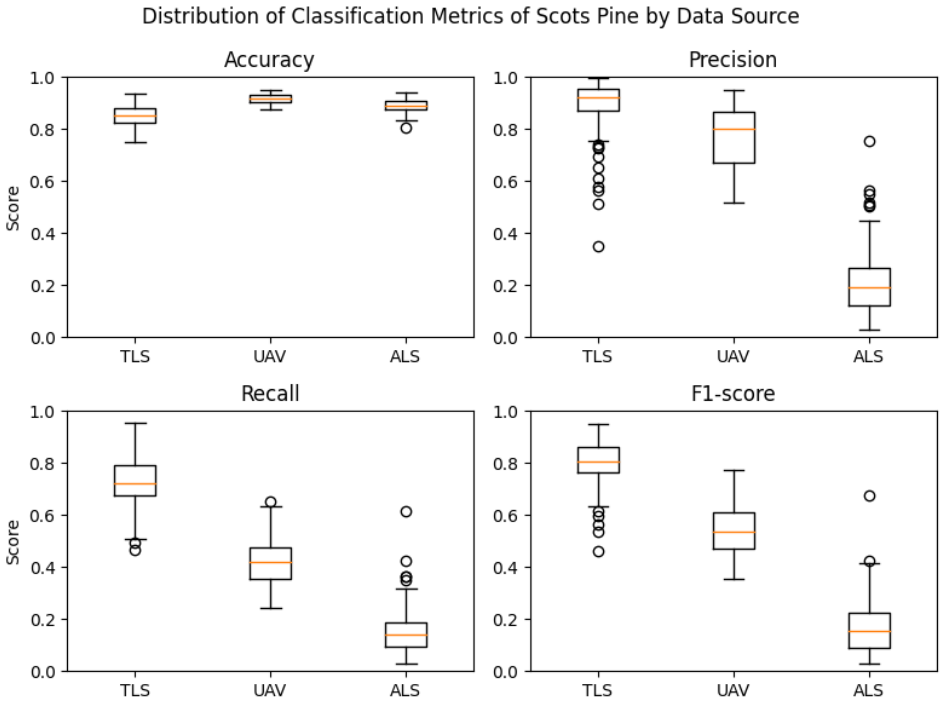
	TLS			**UAV**	**ALS**		
	1	**2**	**3**	**1**	**1**	**2**	**3**
Number of Trees	128	45	8	47	60	19	156

Table 4: Number of trees per species and data source. Scots pine, Norway spruce and silver birch are labelled as 1, 2 and 3 respectively

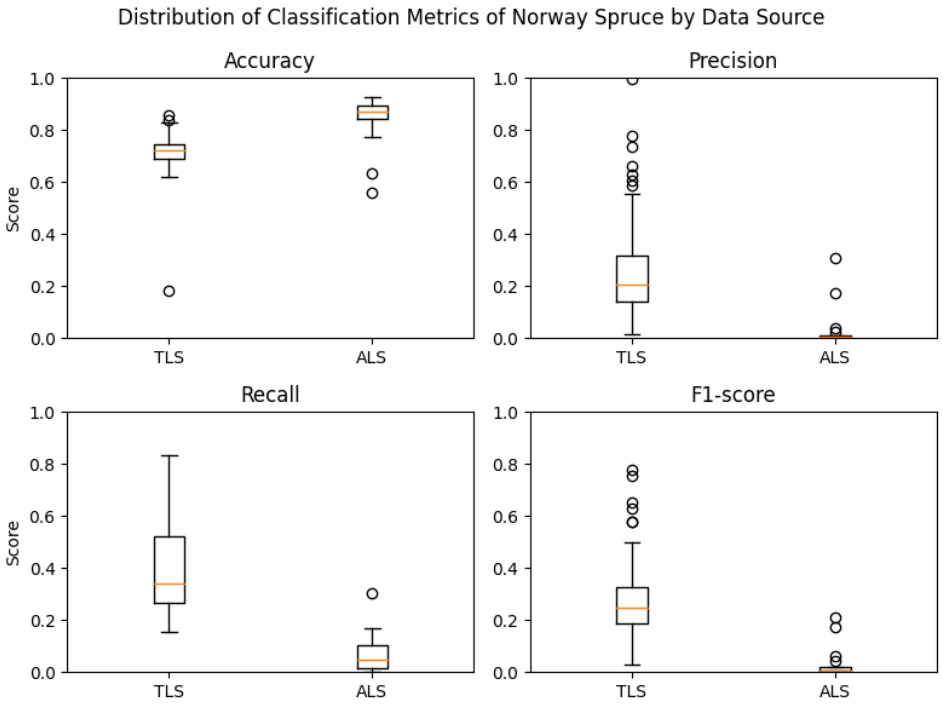
Table 5 presents the classification results for each tree species — Scots pine, Norway spruce, and silver birch — across the TLS, UAV, and ALS point cloud datasets. Figures 12a, 12b, and 12c display box plots for accuracy, precision, recall, and F1-score for each species, illustrating both average classification performance and variability across the different data sources. Each species is analysed in turn, focusing on how the metric values and their variation relate to its structural characteristics and the point cloud density of each LiDAR data source.

Data Source	**Accuracy**	**Precision**	**Recall**	**F1-Score**
TLS				
Scots pine	0.850 ± 0.040	0.893 ± 0.099	0.733 ± 0.094	0.801 ± 0.086
Norway spruce	0.715 ± 0.097	0.282 ± 0.221	0.404 ± 0.180	0.300 ± 0.175
Silver birch	0.834 ± 0.032	0.977 ± 0.014	0.826 ± 0.044	0.895 ± 0.030
UAV				
Scots pine	0.916 ± 0.019	0.764 ± 0.124	0.426 ± 0.091	0.543 ± 0.097
ALS				
Scots pine	0.899 ± 0.025	0.217 ± 0.150	0.160 ± 0.105	0.177 ± 0.114
Norway spruce	0.842 ± 0.091	0.032 ± 0.075	0.074 ± 0.075	0.032 ± 0.057
Silver birch	0.874 ± 0.041	0.050 ± 0.057	0.085 ± 0.077	0.057 ± 0.061

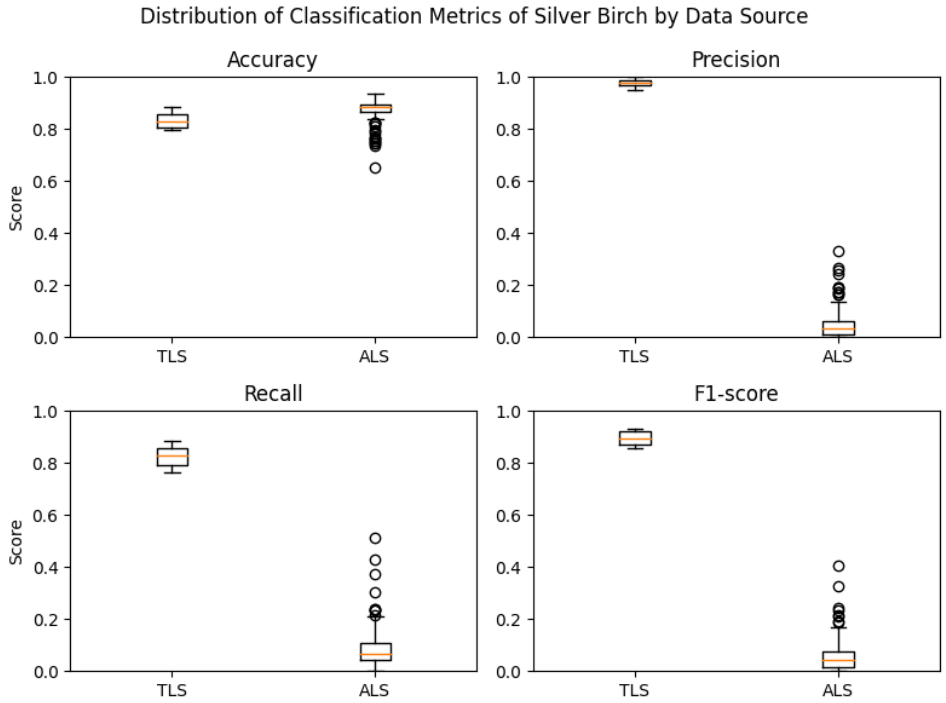
Table 5: Species-wise classification performance metrics with standard deviations



(a) Overall distribution of classification metrics of Scots pine



(b) Overall distribution of classification metrics of Norway spruce

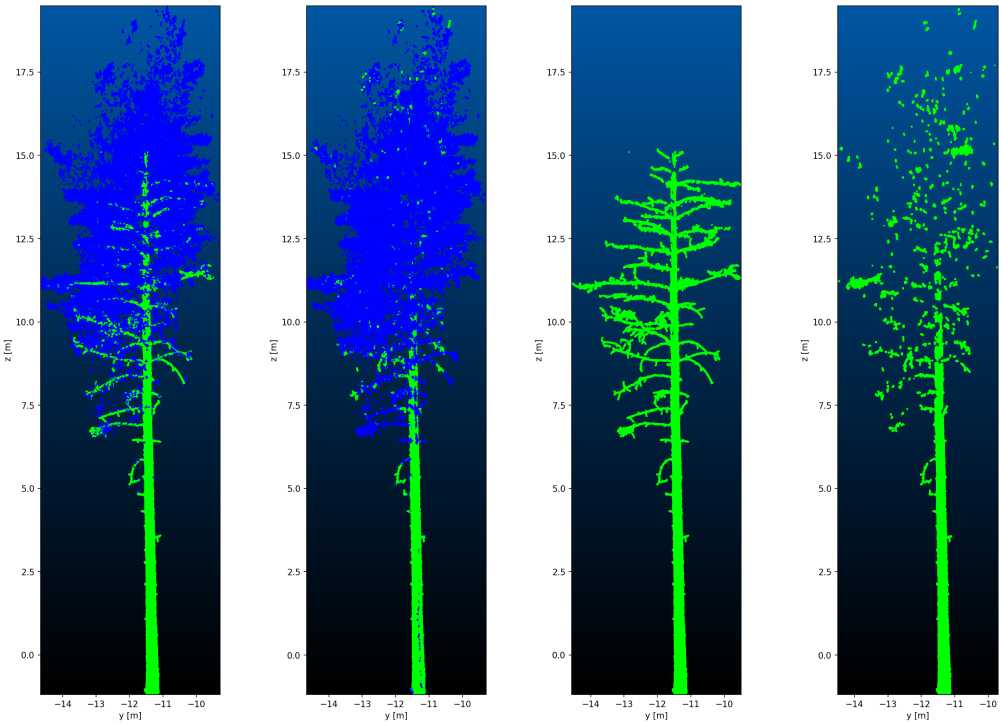


(c) Overall distribution of classification metrics of silver birch

Figure 12: Overall distribution of classification metrics species-wise

TLS data yielded the strongest overall classification performance across Scots pines and silver birches, as shown in Table 5. For Scots pine, the classifier achieved an accuracy of 0.850, a precision of 0.893, a recall of 0.733, and an F1-score of 0.801, with relatively low standard deviations. The narrow interquartile ranges in Figure 12a, with only a few outliers, further confirm a consistent performance across Scots pine trees in the TLS dataset. The presence of outliers is expected given the large sample size of 128 Scots pine trees, as shown in Table 4. These metrics reflect not only the overall proportion of correctly classified points but also how the algorithm distinguishes effectively between wood and leaf points. This overall reliability is likely attributed to the relatively sparse canopy structure of Scots pines and the clear geometric distinction between their thin needle clusters and woody branches, which can be captured effectively in dense TLS point clouds. Therefore, thresholds derived from geometric features are more feasible to detect using the GMM due to the clarity of structural separation.

An example Scots pine tree from the TLS dataset is analysed in Figure 13, where the classification accuracy is 0.87, precision is 0.96, recall is 0.79, and F1-score is 0.87. Figure 13 presents, side by side, the manual classification of the entire tree in Figure 13a, the classification result from the algorithm in Figure 13b, the manual classification of the wood component in Figure 13c, and the algorithmic wood classification in Figure 13d.



(a) Manual classification **(b)** Classification algorithm **(c)** Manual classification **(d)** Classification algorithm

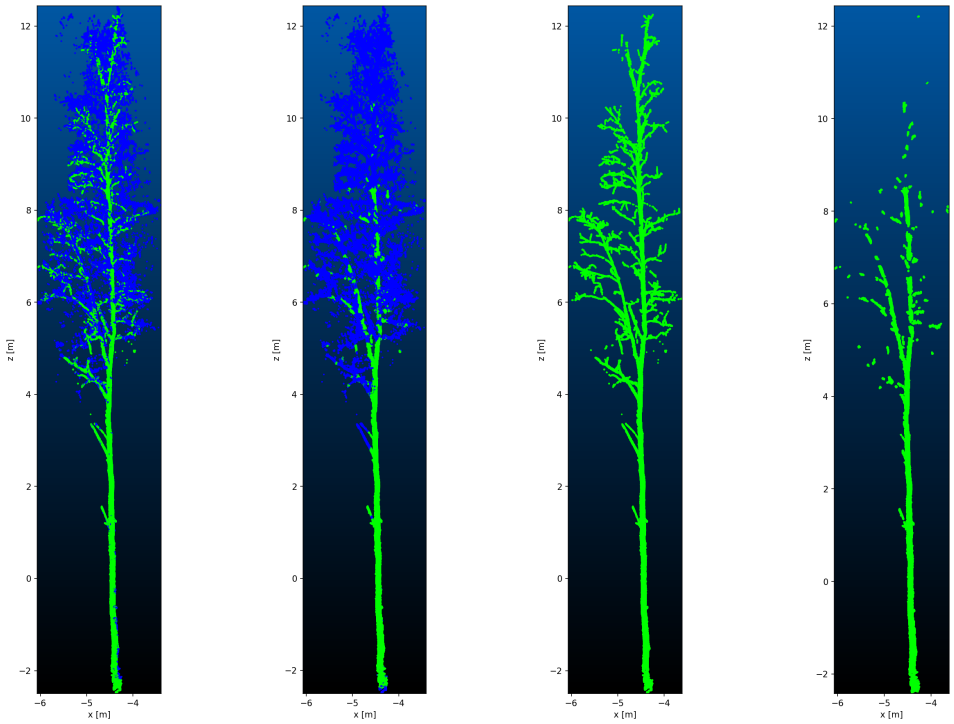
Figure 13: Classification results of Scots pine from TLS data

Figure 13 reveals that despite the example tree exhibiting high accuracy, precision, recall, and F1-score, the algorithm struggles to capture smaller branches as the geometric features of these branches tend to blend with those of the leaves, as shown when comparing Figures 13c and 13d. Additionally, the stem becomes more difficult to detect in the upper canopy region, where a significant portion is misclassified as leaf. At the very top of the tree, the stem is missing, as seen in Figure 13c, resulting in misclassification by the algorithm in that part. This misclassification is particularly visible when comparing Figures 13b and 13a. These errors are likely intensified by scanner noise and missing parts of the point cloud, both of which negatively impact the algorithm's ability to accurately separate wood from leaf structures, as illustrated in Figure 13d.

Silver birch exhibited different behaviour, partly due to the more distinct intensity separability between wood and leaf components compared to coniferous species. Although TLS accuracy for birch was slightly lower (0.834), its precision (0.977), recall (0.826), and F1-score (0.895) were all higher than those of Scots pine. The high precision indicates that when the classifier predicted a point as wood or leaf, it was correct nearly all the time, with very few misclassifications. The high recall shows the classifier also successfully identified most actual wood and leaf points. This combination of high precision and recall produces a strong F1-score, suggesting a well-balanced and effective classification. Low standard deviations across all metrics indicate high consistency among trees. This is visible in Figure 12c, where TLS box plots are tight with minimal variation and no outliers. Silver birch's broad, flat leaves

likely offer clearer geometric separation from branches, benefiting from the detail in TLS data and supporting more reliable classification. Comparing silver birch and Scots pine may not reflect the overall performance of the algorithm, since, as presented in Table 4, only eight silver birches are available in the TLS dataset compared to 128 Scots pines. The effect of this algorithm on silver birch thus requires further study, which will be discussed in Section 6.

A silver birch example tree from the TLS dataset is analysed in Figure 14. The classification accuracy for this tree is 0.82, precision is 0.97, recall is 0.80, and F1-score is 0.88. Figure 14 presents, side by side, the manual classification of the tree (Figure 14a), the classification obtained with the wood and leaf separation algorithm (Figure 14b), the manual classification of the wood component (Figure 14c), and the algorithmic wood classification (Figure 14d). The analysis of Figures 14c and 14d shows that most parts of the stem are captured by the separation algorithm. However, smaller branches remain challenging even for silver birch. The upper part of the tree is also difficult to capture due to the thin stem. Nonetheless, the metrics align with the visual classification results, as observed by comparing Figures 14a and 14b. In Figure 14d, wood and leaf parts are mostly captured, with some misclassifications occurring within the leaf regions. A substantial portion of the stem is correctly identified, but smaller branches are often misclassified as leaves because their geometric features resemble those of leaves.

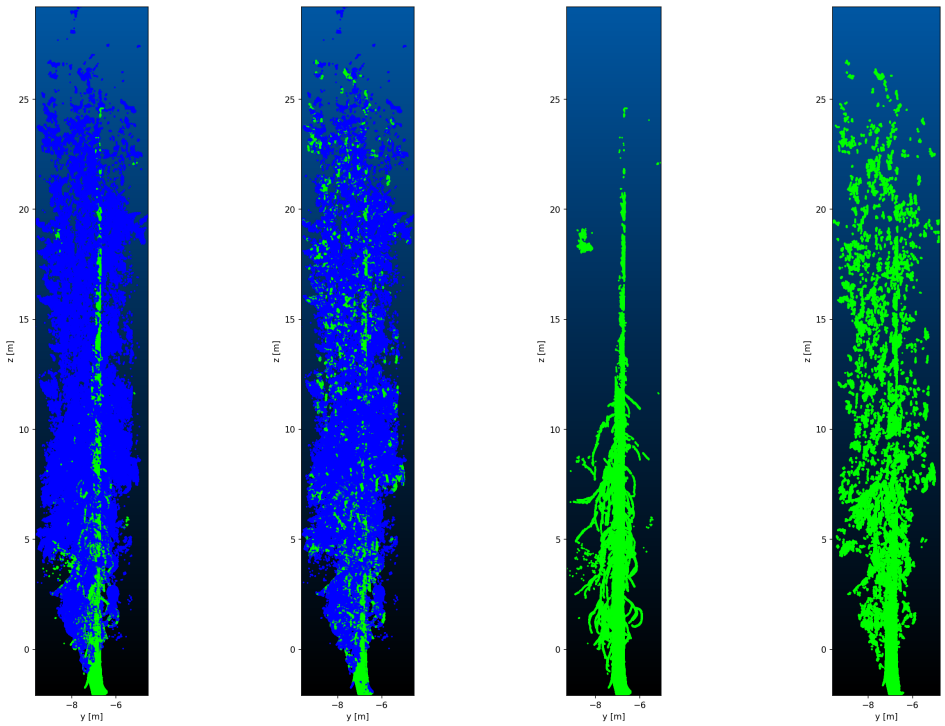


(a) Manual classification **(b)** Classification algorithm **(c)** Manual classification **(d)** Classification algorithm

Figure 14: Classification results of silver birch from TLS data

Norway spruce, by contrast, shows the weakest results with TLS. Accuracy is lower at 0.715, while precision (0.282), recall (0.404), and F1-score (0.300) are all substantially lower than those of Scots pine and birch. Low precision indicates that many points classified as wood or leaf are incorrect, and low recall reveals that many true wood or leaf points are missed. The resulting low F1-score reflects a poor balance between detecting relevant classes and avoiding misclassification. Moreover, these metrics have the largest standard deviations, indicating significant variation between individual trees. This is confirmed in Figure 12b, with TLS box plots showing broad spreads and several outliers, particularly for recall and F1-score. This variability is likely due to the dense and conical structure of spruces, with the geometric distinction between wood and leaf reduced by fine and evenly distributed needles, therefore impeding classification even with high-resolution TLS data.

Further analysis of Norway spruce is conducted by analysing an example tree from the TLS dataset, shown in Figure 15. The classification accuracy for this tree is 0.80, precision is 0.66, recall is 0.64, and F1-score is 0.65. Figure 15 presents a comparative view of the manual classification of the tree (Figure 15a), the classification obtained with the wood and leaf separation algorithm (Figure 15b), the manual classification of the wood component (Figure 15c), and the algorithmic wood classification (Figure 15d).



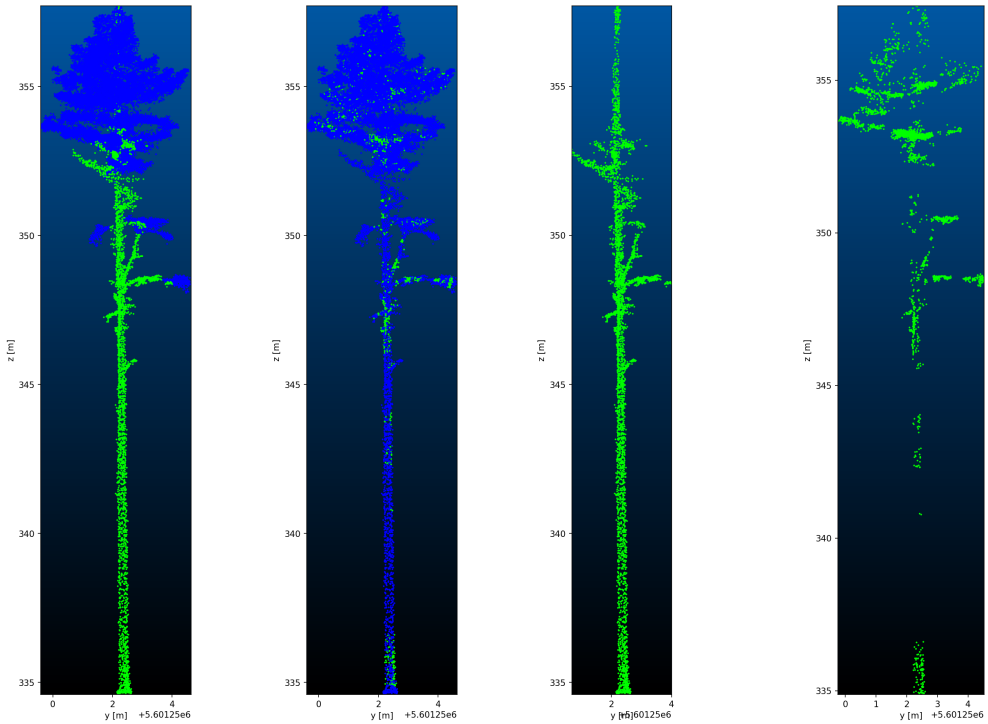
(a) Manual classification **(b)** Classification algorithm **(c)** Manual classification **(d)** Classification algorithm

Figure 15: Classification results of Norway spruce from TLS data

Figure 15 shows that although the metrics for spruce are above average, the classification includes many misclassified points, as observed in Figure 15d. However, given the structure of Norway spruce, even manual classification from point cloud data is extremely challenging and prone to errors, as seen in Figure 15c, by the missing branches in the upper half of the tree. Additionally, the scanner exhibits difficulties in penetrating the dense canopy structure. Examining Figures 15a and 15c reveals that most of the upper part of the tree is classified as wood, resulting in the misclassification of many needles. Due to misclassifications in the reference data, assessing the algorithm's true performance in wood and leaf separation for Norway spruce is difficult.

In the UAV dataset, only Scots pine is included. It achieves the highest accuracy across all species and data types at 0.916, indicating that most points are classified correctly. However, precision (0.764), recall (0.426), and F1-score (0.543) remain lower compared to TLS. The low recall indicates that many actual wood or leaf points are missed. Although a moderately high precision suggests that the model often produces correct predictions, it is less reliable than TLS. This imbalance is reflected in the lower F1-score. Standard deviations for precision and F1-score are slightly higher than those in TLS, indicating greater variability between trees. This variability is evident in the box plots in Figure 12a, demonstrating wider spreads and several outliers, particularly in precision. High accuracy likely results from the predominance of wood points in the UAV data, as predicting most points as wood yields a high proportion of correct labels. However, lower recall demonstrates the classifier's limited ability to detect all relevant wood points.

A Scots pine example tree from the UAV dataset is analysed in Figure 16. The classification accuracy for the tree is 0.92, precision is 0.95, recall is 0.46, and F1-score is 0.62. Figure 16 provides a visual comparison between the manual classification of the tree (Figure 16a), the classification obtained with the wood and leaf separation algorithm (Figure 16b), the manual classification of the wood component (Figure 16c), and the algorithmic classification of the wood component (Figure 16d).



(a) Manual classification **(b)** Classification algorithm **(c)** Manual classification **(d)** Classification algorithm

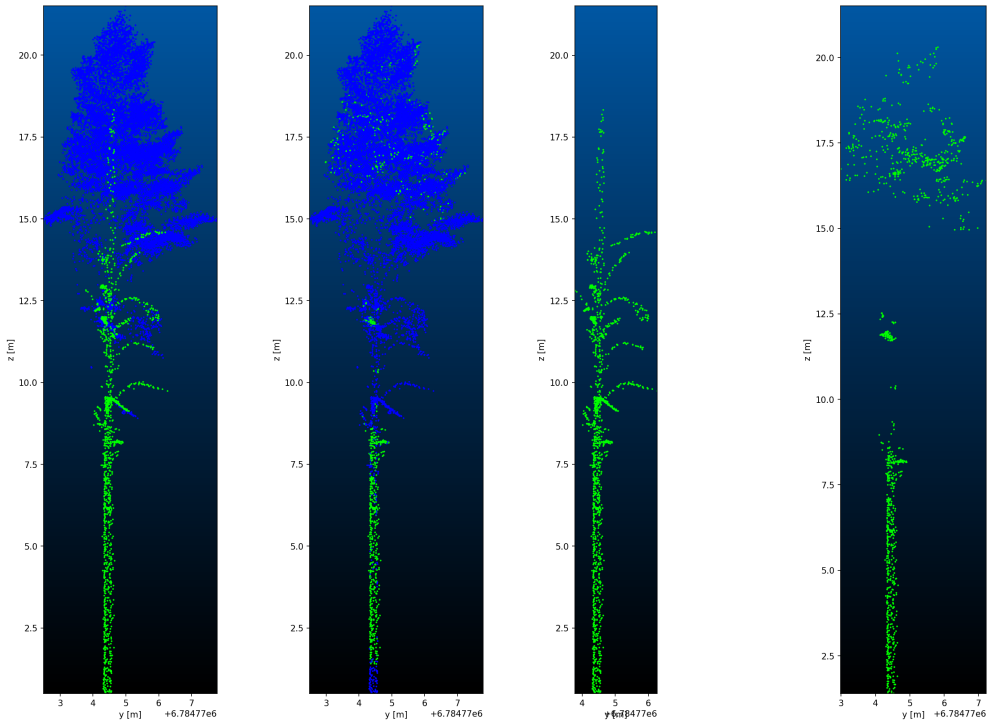
Figure 16: Classification results of Scots pine from UAV data

Analysis of Figure 16 demonstrates that high accuracy and precision alone do not sufficiently assess the quality of wood and leaf separation. Figures 16c and 16d show that despite high precision and accuracy, the low recall of 46% indicates the model captures less than half of the actual wood points, underscoring the importance of analysing recall. Figure 16b illustrates the difficulty in capturing parts of the stem due to the sparsity of the point cloud, where the algorithm successfully identified a lot of the stem in TLS data, as demonstrated in Figures 13d and 14d. In addition to the model's difficulty in capturing most wood points within the UAV point cloud, data sparsity also complicates manual classification of wood and leaf components, as shown in Figures 16a and 16c. The upper part of the tree in Figure 16c lacks labelled branches connecting to the canopy visible in Figure 16a. Furthermore, many wood points are misclassified as leaves in the reference data.

The ALS data demonstrates the weakest overall performance, particularly in terms of precision, recall, and F1-score, although accuracy remains relatively high. For Scots pine, ALS achieves an accuracy of 0.899, but precision (0.217), recall (0.160), and F1-score (0.177) are significantly lower compared to TLS and UAV. Low precision and recall indicate both a large number of misclassified points and many missed wood or leaf points. These metrics are reflected in Figure 12a, displaying wide box plots and several outliers, particularly in precision and recall.

Figure 17 presents a classified Scots pine tree. The classification accuracy for this tree is 0.92, the precision is 0.59, the recall is 0.42, and the F1-score is 0.49. Figure

17 presents the tree's manual classification (Figure 17a), the classification obtained with the wood and leaf separation algorithm (Figure 17b), the manual classification of the wood component (Figure 17c), and the algorithmic classification of the wood component (Figure 17d).



(a) Manual classification **(b)** Classification algorithm **(c)** Manual classification **(d)** Classification algorithm

Figure 17: Classification results of Scots pine from ALS data

Similar to the UAV Scots pine results in Figure 16d, Figure 17d demonstrates the algorithm's difficulty in capturing the full stem of the tree in ALS data. The Scots pine example contains many missed and misclassified points, especially when comparing Figures 17a and 17b. The sparsity of the dataset is particularly evident in Figure 17c. When examining the stem region, the point density is insufficient to facilitate the reliable extraction of geometric features indicating structural components or orientation. Similar difficulties are observed in Norway spruce and silver birch.

For Norway spruce in the ALS dataset, the algorithm yields an accuracy of 0.842, but precision (0.032), recall (0.074), and F1-score (0.032) are extremely low. These values indicate the classifier rarely makes correct predictions and often entirely misses relevant points. High variability is evident in both metric standard deviations and the box plots in Figure 12b, showing multiple outliers. The low point density of the ALS data, combined with the complex geometry of spruce trees, limits the detection of meaningful separation between wood and leaves.

Similarly, silver birch in ALS exhibits poor results. Accuracy is 0.874, while precision (0.050), recall (0.085), and F1-score (0.057) remain low, with large standard

deviations. The classifier often fails to distinguish between wood and leaf components. This is illustrated by broad box plots and numerous outliers in Figure 12c, especially in recall and F1-score. Although accuracy remains relatively high, it is a result of the dominance of the wood class rather than true classification success.

Overall, the results confirm that denser point clouds improve wood and leaf separation. TLS consistently outperforms the other data types in precision, recall, and F1-score, particularly for Scots pine and silver birch. UAV achieves the highest accuracy for Scots pine but shows lower recall and F1-score, indicating a substantial number of relevant points were missed despite classifying most points correctly. ALS data produces the poorest results in precision, recall, and F1-score for all species, particularly for Norway spruce and silver birch. The high accuracies observed in the ALS dataset are a result of class imbalance, as wood points dominate the data. By predicting all points as wood, a high accuracy would be achieved. This underlines the importance of evaluating all metrics beyond accuracy when assessing classifier performance. Nonetheless, this study aims to evaluate the algorithm's limitations and determine the point cloud densities at which its performance begins to decline.

5.3 Effect of point density on classification performance

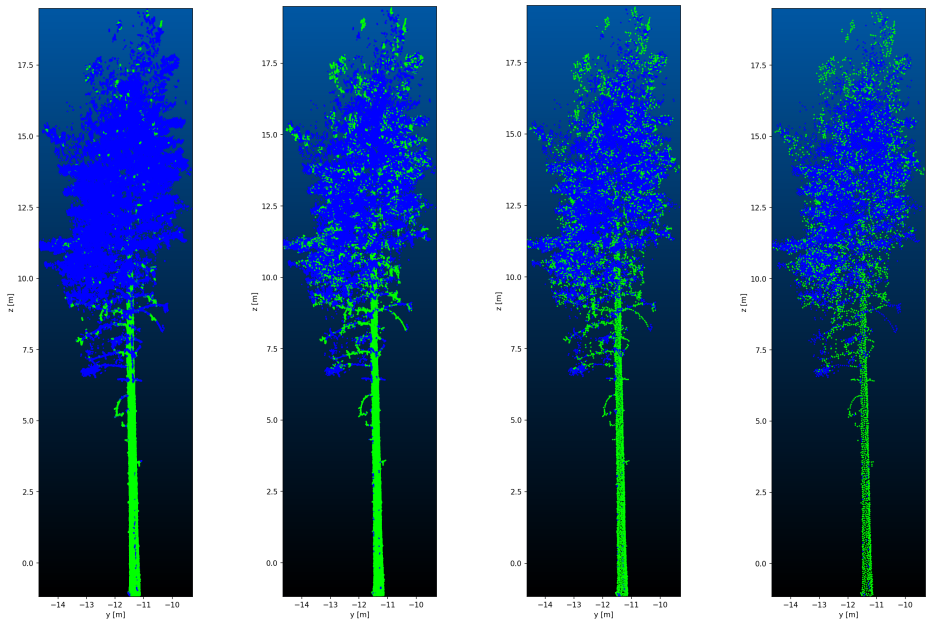
The ALS data achieved worse classification metrics than both the TLS and UAV data. The poor performance is directly attributed to the sparser point cloud. In a dense dataset, the algorithm accurately captures the geometric features that describe local structures. However, as the dataset becomes sparser, the geometric characteristics of leaves and wood start to resemble each other more closely, making it harder to distinguish between them. This causes the classification algorithm's effectiveness to collapse, yielding much lower precision and recall for the ALS data.

To further examine how point cloud density affects classification, the density of a high-resolution point cloud is reduced to observe the changes in performance metrics. As an example, the Scots pine tree from the TLS dataset (Figure 13) is analysed for the experiment. Initially, with the full-density point cloud (approximately 87 600 points/m^2), the classification metrics were high: 0.87 accuracy, 0.96 precision, 0.79 recall, and 0.87 F1-score. The point cloud of this tree was further downsampled to lower densities by removing points within a specific distance from each other, thereby increasing the spacing between points. The classification performance was re-evaluated at each level. The resulting metrics, along with the corresponding average point densities, are presented in Table 6.

Point density (points / m^2)	Accuracy	Precision	Recall	F1-Score
87 609	0.8702	0.9623	0.7888	0.8669
6 665	0.7956	0.4217	0.6047	0.4969
2 145	0.7954	0.3257	0.5637	0.4129
1 218	0.7737	0.2662	0.5362	0.3558

Table 6: The affect of point cloud resolution in wood and leaf separation of a Scots pine from TLS data

At the original resolution of roughly 87 600 points/m^2, the classification performed well. Reducing the density to about 6 700 points/m^2 caused all metrics to decline, with precision dropping the most. Notably, the accuracy remained relatively high across all densities, as visible in Table 6, likely due to wood points dominating the point cloud. When the density was further reduced to approximately 2 100 points/m^2, comparable to the UAV dataset's density, the metrics fell to a level similar to those observed for actual UAV data. However, the precision in this downsampled case was much lower than the precision obtained from the real UAV dataset, since the feature weights have not been adjusted accordingly, demonstrating the importance of adjusting the weights depending on the dataset's density. Finally, at the lowest density of approximately 1 200 points/m^2, the classification metrics degraded to a level approaching the performance on the ALS data. The classification outcomes at these different point densities are visualised in Figure 18, which demonstrates the Scots pine point cloud and its classification results at each resolution, including both the entire tree classification and the wood points classification for clarity.

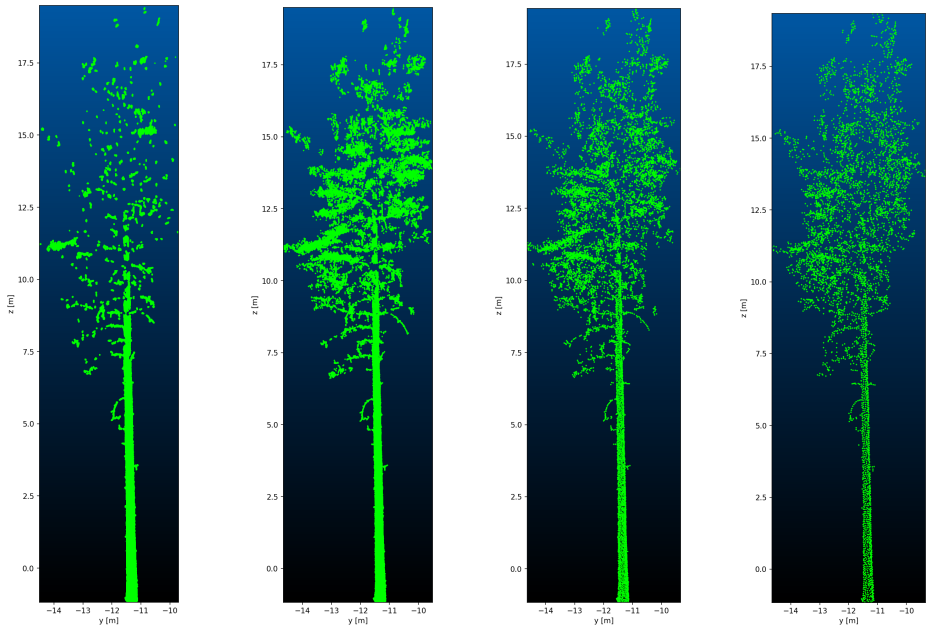


(a) Classification of the tree with original point density

(b) Classification of the tree with point density of 6700 points / m^2

(c) Classification of the tree with point density of 2100 points / m^2

(d) Classification of the tree with point density of 1200 points / m^2



(e) Classification of the wood with original point density

(f) Classification of the wood with point density of 6700 points / m^2

(g) Classification of the wood with point density of 2100 points / m^2

(h) Classification of the wood with point density of 1200 points / m^2

Figure 18: Classification visualisations with different point densities

As shown in Figures 18c and 18d, the tree at the two lowest densities of 2,100 and 1,200 points/m^2 resembles the typical point density and visual sparsity of UAV and ALS datasets, respectively. Correspondingly, the classification results in Figures 18g and 18h indicate that the algorithm labels nearly all points as wood. This occurs because the classifier is still using feature weightings optimised for the high-density TLS data, which do not transfer well to much sparser point clouds.

Overall, it is evident that as the point cloud resolution decreases, the classification performance worsens significantly since the method relies on geometric feature extraction and the feature values no longer reflect the local structure as the distance between points increases. However, this experiment revealed that the number of points does not affect the algorithm's performance. It is rather the spacing between them, indicating that the distance between points in the cloud has a crucial role in the performance.

6 Discussion

6.1 Wood and leaf separation algorithm

The thesis explored an alternative unsupervised approach for wood and leaf classification. Since the adaptive radius yields one optimal neighbourhood radius and, subsequently, an optimal set of geometric features, an experiment was conducted combining information from two optimal radii at different scales. This approach was inspired by Shcherbcheva et al. (2023), who performed wood and leaf classification by combining features obtained with multiple radii. In the thesis experiment, each point had two sets of optimal geometric features derived from both a smaller and a larger neighbourhood radius. Results were merged using logical OR and AND operators. The assumption was that a second set of geometric features could capture additional wood points missed by a single radius approach. However, this method degraded accuracy compared to using a single optimal adaptive radius, achieving average overall accuracies of 40 to 60%. Through a series of trials, it became evident that applying one optimal radius per point yielded the best performance. Therefore, the combination of two optimal radii was not further explored or reported in this thesis. This finding supports the idea that an appropriately chosen single scale can capture the key geometric structure of a neighbourhood, whereas combining scales may lead to misclassification and overfitting.

A key aspect of this study's wood and leaf separation algorithm was the computation of geometric features using the adaptive radius search method. Rather than applying a single radius for all points or using a small set of discrete radii, the algorithm finds an optimal radius for each point by minimising Shannon's entropy of the local neighbourhood. As described in Section 4.2, this approach, based on Demantké et al. (2011), was originally developed for building point clouds and had not been explored in the case of tree point clouds, even though the search radius is one of the most important parameters for computing geometric features. Implementing the adaptive radius search for tree point clouds allowed the features to adjust to the local geometry of the neighbourhood. For instance, a large branch or stem was assigned a larger neighbourhood, whereas smaller branches retained a small optimal radius.

The optimal radius was found by iterating through different possibilities within a search range. The search range relied on the resolution of the dataset and was set manually. However, future research could further optimise the search range by basing it on computed point cloud density. The radius search was bounded between a minimum of 0.10 m, and a dataset-dependent maximum was set to 0.5 m for dense TLS data and extended to 1.5 m for sparser UAV and ALS datasets.

Once the optimal neighbourhood was found for each point, a set of geometric features was computed to characterise the local neighbourhood shape. However, not all features contributed equally to the separation of wood and leaf, with variations across datasets. Multiple features were evaluated, and those demonstrating poor classification performance were excluded. Only features that individually achieved at least 70% classification accuracy were included in the analysis.

A two-component GMM classification was further applied independently to each chosen feature, resulting in binary classifications. Each feature classification was assigned a weight based on its importance in wood and leaf separation, with weights adjusted for each dataset according to feature distributions. Although the weighting process was conducted empirically in this study, future research could automate the weight assignment to enhance the algorithm's universality. The application of a weighted sum of features ensured that each feature contributed to a point's classification, with more informative features having greater influence. This approach enhanced algorithm robustness across TLS, UAV, and ALS datasets.

Implementing the adaptive radius search involved substantial computational cost. Processing millions of points across varying radii is computationally intensive, proving efficiency crucial. The step size in radius search (s in Algorithm 1) represented a trade-off between accuracy and computational efficiency. While a smaller step size increased the likelihood of accurately identifying the true entropy minimum, it significantly raised the number of iterations and runtime. A balanced step size was therefore selected to preserve classification accuracy while limiting computational cost. Manual fine-tuning demonstrated that for TLS data, increasing the step size beyond 35 yielded no improvement in results past the fourth decimal place. In contrast, for airborne data, a larger step size negatively affected accuracy due to the increased space between points. Therefore, UAV and ALS results improved when the step size was reduced to 25.

Since the algorithm requires finding an optimal neighbourhood radius for each point individually, the originally implemented brute force radius searches were computationally intensive. Parallelisation was therefore implemented by distributing the workload, reducing runtime. This parallel approach transformed the initial nested-loop algorithm into a method capable of processing entire point cloud datasets efficiently. Although further optimisations or approximation techniques may be necessary for large-scale applications, the parallelised implementation applied in this thesis was sufficient to process the full datasets within a reasonable time. Tuning algorithm parameters, along with parallelisation, ensured its practicality across varying data sources.

6.2 Performance across applied data acquisitions

The proposed wood and leaf separation algorithm utilising the adaptive radius approach demonstrated strong overall accuracy, with performance varying across the TLS, UAV, and ALS datasets. A species-wise analysis provided further insights into algorithm performance. Scots pine, with its relatively open canopy and long needles, yielded reliable classification results, aligning with the species' structure. The sparse canopy allowed the algorithm to compute geometric features that effectively distinguished between wood and leaf components. Finer branches were more likely to be detected, as their geometry was less occluded with leaf points due to limited overlap in the neighbourhood structure.

Silver birch, which exhibited the most distinct intensity between leaf and wood components among all tested species, achieved the best overall performance. However,

the sample size of silver birch was small, consisting of only eight trees compared to 128 Scots pines, suggesting that this high accuracy may not generalise without further testing. Nevertheless, the results were consistent from a structural point of view. Silver birch trees had broad, flat leaves that occupied much of the surrounding space in their local neighbourhood. In cases where the optimal neighbourhood was computed for a point located on a leaf, the radius was expected to include primarily leaf points, facilitating accurate classification by the algorithm.

In contrast, Norway spruce emerged as the most challenging species. The algorithm struggled to differentiate small spruce branches from needles, as the dense needles' neighbourhoods blended with the small branches, causing their geometric features to overlap. Additionally, the stem often remained obscured within dense foliage, complicating the identification of woody structures. Future work could test if improvement can be obtained in the classification performance for Norway spruce by using separate radius search ranges for small branches and the stem.

Most studies in the field have focused on wood and leaf separation utilising TLS datasets. However, complete 3D modelling of tree point clouds often requires the fusion of UAV and TLS data, which in turn demands different parametrisation. Therefore, this study aimed to generalise the method across various data acquisition types, given the increasing use and potential of UAV and ALS. UAV point clouds, however, tended to misclassify more wood points due to the greater distances between points, which limited geometric feature extraction and thus blurred the distinction between wood and leaf components.

ALS data presented limitations in distinguishing between wood and leaf classes. At such low point densities, the classifier tended to predict the dominant class for most points. The relatively high accuracy for ALS spruce and birch resulted from class imbalance rather than true classification performance.

These results confirm that point cloud resolution significantly influences classifier performance, even with the adaptive radius search method. The study managed to identify the point density threshold at which the unsupervised classification algorithm becomes unstable. The results confirmed that, with UAV data, performance remains adequate. However, the algorithm failed to well handle datasets as sparse as the ALS data, highlighting the limitation that no single algorithm can be universally applicable across all tree species and data acquisitions.

6.3 Comparison with existing methods

Prior studies have explored both supervised and unsupervised algorithms for wood and leaf separation. The accuracy achieved by the method presented in this thesis on TLS data aligns with existing unsupervised algorithms, though some methods outperform it. For instance, Ali et al. (2024) evaluated several classifiers on TLS datasets, reporting accuracies of 95% for the supervised random forest, 90% for the unsupervised LeWoS algorithm, 89% for the supervised CANUPO, and 81% for the unsupervised TLSeparation. In comparison, the method described in this thesis achieved an accuracy of approximately 85% on TLS datasets, excluding Norway spruce.

The unsupervised GMM approach presented here is competitive with CANUPO, a supervised classifier using similar geometric features, and approaches the accuracy of the LeWoS method. The comparable performance of this study's unsupervised algorithm to a supervised method, such as CANUPO, highlights its effectiveness. However, accuracy is not the sole consideration. A key advantage of the proposed method is its data efficiency and adaptability. Unsupervised methods handle larger datasets without requiring manually labelled data and can be more easily applied across forest sites.

Many automated wood and leaf classifiers apply similar geometric features. For instance, CANUPO applies a supervised SVM on geometric features derived from PCA and point density, whereas this study's method applies GMM to individual geometric features, combining them into a classification through a weighted sum. The success of the weighted feature sum approach suggests that, even without advanced learning algorithms, geometric features alone provide sufficient information to separate wood from leaves. The adaptive radius further enhances this by computing features at an appropriate scale for each point, improving their quality.

Ali et al. (2024) noted that variations in chosen neighbourhood radii and the use of multi-scale features could lead to differences in accuracies. Interestingly, Ali et al. (2024) observed that their random forest classifier achieved lower accuracy than that reported by Wang et al. (2017) but higher than that of Zhu et al. (2018), likely due to differences in multi-scale feature selection and radius choices. This thesis method selected a single adaptive scale rather than incorporating multi-scale features. Additionally, the geometric features applied in this study are similar to those in Wang et al. (2017), which achieved the highest accuracies among the supervised methods. However, the exceptionally high accuracy achieved by Wang et al. (2017) could be attributed to the limitations of the two-tree dataset used in their study.

In summary, the GMM with adaptive radius method performs competitively among unsupervised algorithms, approaching the lower performance range of supervised methods, while requiring no training data. This offers a significant advantage for large-scale forestry applications. The findings align with the broader understanding in the field, suggesting that supervised methods typically yield higher accuracy, while unsupervised methods offer greater flexibility and efficiency. Future developments could aim to integrate aspects of both approaches, such as applying geometric features as unsupervised methods do, while learning optimal weightings or neighbourhood searches from data. A semi-supervised extension could enhance precision, particularly for challenging cases such as Norway spruce, as also noted by Ali et al. (2024).

Overall, the adaptive radius GMM method is effective under certain conditions and compares well with existing approaches. However, it remains unclear what the tree species or the quality of point clouds utilised in previous studies were and whether their reported high accuracies resulted from exceptionally well-captured data. Factors such as data density, species complexity, and class imbalance continue to influence the algorithm's performance. These results emphasise the importance of adaptive, data-driven parameter selection in point cloud analysis. Challenges such as penetrating dense spruce canopies or processing incomplete data may require further development. With refinement and potential integration with supervised techniques, this approach

could evolve into a powerful tool for forest point cloud analysis, combining the strengths of both unsupervised and supervised methods to enhance robustness and applicability.

6.4 Recommendations for future work based on advantages and limitations

While the proposed wood and leaf separation algorithm provided accurate results, limitations of this study's unsupervised approach can be highlighted to guide further improvement. The presence of scanner noise or missing parts of the point cloud reduced the algorithm's classification performance. For instance, when tree crowns were not fully captured by the scanner, the computed geometric features failed to represent the true local structure, leading to misclassification. This issue was particularly noticeable with TLS data, where incomplete capture of the upper parts of the trees resulted in poorer classification. Although the UAV and ALS datasets captured the overall tree structure, their lower point cloud density in these datasets restricted the algorithm's ability to accurately capture the local geometry of the neighbourhood, resulting in misclassification. To address these limitations, the fusion of TLS and UAV data could provide a more complete and dense dataset, but future studies are required to evaluate the parametrisation requirements in data fusion scenarios.

Class imbalance in the data represents another challenge for evaluation metrics. In all datasets, the number of wood points exceeded the number of leaf points, and the unsupervised classifier does not account for this imbalance. This issue was most evident in the ALS results, where the classifier labelled the majority of points as wood, resulting in high overall accuracy but very low precision and recall for the wood class. This scenario highlights why accuracy alone is a misleading metric in imbalanced datasets. Although other metrics were incorporated in the evaluation, the algorithm itself does not explicitly address class imbalance.

Norway spruce presented classification difficulties, raising questions about the generalisability of the method across species. The adaptive radius method occasionally failed to identify a scale capable of distinguishing foliage points from non-foliage within Norway spruce point clouds. At small radii, the local neighbourhood of a spruce needle often included parts of adjacent small branches, whereas at larger radii, multiple branches and needles were included. Future research could explore density-specific neighbourhood radius search bounds to better capture geometric features of foliage points. While earlier supervised studies, for instance Xi et al. (2018), characterised wood and leaf separation as species-independent, the present findings challenge this view concerning unsupervised classification. Although the algorithm did not incorporate any species-specific parameters, its performance was influenced by species-specific characteristics.

In contrast to Norway spruce, the method performed well on Scots pine and broadleaved silver birch, indicating that the adaptive radius search algorithm effectively adapted to different tree structures and captured their geometric properties. However, limited sample sizes restricted the assessment of generalisability on silver birch, and conclusions regarding species effects should require validation on larger datasets.

Nonetheless, prior research indicates that leaf and wood classification tends to be more effective for deciduous species (Ferrara et al., 2018; Chen et al., 2025). The dominance of Scots pine, particularly in the UAV dataset, contributed to high classification accuracy, as the method handles this species relatively well. Further research could include a broader range of tree species to assess the method's applicability more comprehensively.

The unsupervised nature of the classifier introduced additional limitations. As the GMM is not based on manual labels, it classifies points based solely on feature distributions. While this allows application without training data and enables generalisation across datasets, it may lead to misclassification. The lack of supervision also required manual tuning of parameters, such as feature weights and radius ranges, which were dataset-specific. Application to new datasets with differing characteristics may necessitate reconfiguration. The weight tuning process, as noted earlier, could be automated in future by analysing point cloud properties. Despite these limitations, the unsupervised approach demonstrated strong performance in this context, achieving reliable accuracy without labelled data. This is particularly advantageous for large-scale datasets where manual classification is impractical. Overall, the study confirmed the original hypothesis and demonstrated that the adaptive radius search method enhances feature extraction and improves accuracy in wood and leaf classification.

7 Conclusion

This thesis demonstrated that adaptive radius search improved unsupervised wood and leaf classification in forest LiDAR data, validating the central hypothesis. By automatically determining an optimal neighbourhood radius for each point, the method enhanced geometric feature quality and improved classification accuracy across datasets. The unsupervised classifier achieved performance comparable to supervised algorithms in high-density scans without relying on manually labelled training data, highlighting its practicality and efficiency. Selecting point-specific optimal neighbourhoods reduced manual parameter tuning and ensured consistent classification performance across diverse tree species and acquisition geometries.

Achieving accuracies similar to supervised methods was particularly significant in a field demanding extensive manual labelling, a process both labour intensive and time consuming. Demonstrating such performance with a fully unsupervised approach emphasised its effectiveness and aligned with current efforts to minimise dependence on labelled data. The capability to classify wood and foliage without supervised training offered clear benefits for large-scale and long-term forest monitoring applications.

Nevertheless, some limitations persisted. This study showed that while the algorithm performed effectively with point cloud densities comparable to UAV data, performance declined with sparser datasets such as ALS. Variation in data acquisition methods and tree species prevented the universal applicability of any single classification approach. Overall, this study confirmed the objective of developing an adaptable and effective wood and leaf classification algorithm. Continued refinement and integration of emerging techniques could establish this adaptive unsupervised approach as a robust foundation for fully automated forest point cloud analysis.

References

Ali, M., Lohani, B., Hollaus, M., & Pfeifer, N. (2024). Benchmarking geometry-based leaf-filtering algorithms for tree volume estimation using terrestrial lidar scanners. *Remote Sensing*, *16*(6), 1021. https://doi.org/10.3390/rs16061021

Boucher, P. B., Paynter, I., Orwig, D. A., Valencius, I., & Schaaf, C. (2021). Sampling forests with terrestrial laser scanning. *Annals of Botany*, *128*(6), 689-708. https://doi.org/10.1093/aob/mcab073

Brede, B., Terryn, L., Barbier, N., Bartholomeus, H. M., Bartolo, R., Calders, K., ... Herold, M. (2022). Non-destructive estimation of individual tree biomass: Allometric models, terrestrial and uav laser scanning. *Remote Sensing of Environment*, *280*, 113180. https://doi.org/10.1016/j.rse.2022.113180

Brodu, N., & Lague, D. (2012). 3d terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS Journal of Photogrammetry and Remote Sensing*, *68*, 121-134. https://doi.org/10.1016/j.isprsjprs.2012.01.006

Campos, M. B., Litkey, P., Wang, Y., Chen, Y., Hyyti, H., Hyyppä, J., & Puttonen, E. (2021). A long-term terrestrial laser scanning measurement station to continuously monitor structural and phenological dynamics of boreal forest canopy. *Frontiers in Plant Science*, *11:606752*. https://doi.org/10.3389/fpls.2020.606752

Chen, S., Verbeeck, H., Terryn, L., Van Den Broeck, W. A., Vicari, M. B., Disney, M., ... Calders, K. (2025). The impact of leaf-wood separation algorithms on aboveground biomass estimation from terrestrial laser scanning. *Remote Sensing of Environment*, *318*, 114581. https://doi.org/10.1016/j.rse.2024.114581

Demantké, J., Mallet, C., David, N., & Vallet, B. (2011). Dimensionality based scale selection in 3d lidar point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XXXVIII-5/W12*, 97–102. https://doi.org/10.5194/isprsarchives-XXXVIII-5-W12-97-2011

Ferrara, R., Virdis, S. G., Ventura, A., Ghisu, T., Duce, P., & Pellizzaro, G. (2018). An automated approach for wood-leaf separation from terrestrial lidar point clouds using the density based clustering algorithm dbscan. *Agricultural and Forest Meteorology*, *262*, 434–444. https://doi.org/10.1016/j.agrformet.2018.04.008

Geiß, C., Pelizari, P. A., Marconcini, M., Sengara, W., Edwards, M., Lakes, T., & Taubenböck, H. (2015). Estimation of seismic building structural types using multi-sensor remote sensing and machine learning techniques. *ISPRS Journal of Photogrammetry and Remote Sensing*, *104*, 175–188. https://doi.org/10.1016/j.isprsjprs.2014.07.016

Hui, Z., Jin, S., Xia, Y., Wang, L., Ziggah, Y. Y., & Cheng, P. (2021). Wood and leaf separation from terrestrial lidar point clouds based on mode points evolution. *ISPRS Journal of Photogrammetry and Remote Sensing*, *178*, 219–239. https://doi.org/10.1016/j.isprsjprs.2021.06.012

Lalonde, J., Vandapel, N., Huber, D. F., & Hebert, M. (2006). Natural terrain classification using three dimensional ladar data for ground robot mobility.

Journal of Field Robotics, *23*(10), 839–861. https://doi.org/10.1002/rob.20134

Liang, X., Hyyppä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., . . . Wang, Y. (2018). International benchmarking of terrestrial laser scanning approaches for forest inventories. *ISPRS Journal of Photogrammetry and Remote Sensing*, *144*, 137-179. https://doi.org/10.1016/j.isprsjprs.2018.06.021

Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H., & El-Sana, J. (2010). Automatic reconstruction of tree skeletal structures from point clouds. *ACM Transactions on Graphics (TOG)*, *29*(6), 151. https://doi.org/10.1145/1882261.1866177

Ma, L., Zheng, G., Eitel, J. U. H., Moskal, L. M., He, W., & Huang, H. (2016). Improved salient feature-based approach for automatically separating photosynthetic and nonphotosynthetic components within terrestrial lidar point cloud data of forest canopies. *IEEE Transactions on Geoscience and Remote Sensing*, *54*(2), 679–696. https://doi.org/10.1109/tgrs.2015.2459716

Moorthy, S. M. K., Calders, K., Vicari, M. B., & Verbeeck, H. (2020). Improved supervised learning-based approach for leaf and wood classification from lidar point clouds of forests. *IEEE Transactions on Geoscience and Remote Sensing*, *58*(5), 3057-3070. https://doi.org/10.1109/tgrs.2019.2947198

Puliti, S., Pearse, G., Surový, P., Wallace, L., Hollaus, M., Wielgosz, M., & Astrup, R. (2023). For-instance: a uav laser scanning benchmark dataset for semantic and instance segmentation of individual trees. *arXiv (Cornell University)*. https://doi.org/10.48550/arXiv.2309.01279

Ruoppa, L., Oinonen, O., Taher, J., Lehtomäki, M., Takhtkeshha, N., Kukko, A., . . . Hyyppä, J. (2025). Unsupervised deep learning for semantic segmentation of multispectral lidar forest point clouds. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2502.06227

Shcherbcheva, A., Campos, M. B., Liang, X., Puttonen, E., & Wang, Y. (2023). Unsupervised statistical approach for tree-level separation of foliage and non-leaf components from point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVIII-1/W2-2023*, 1787–1794. https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-1787-2023

Tao, S., Guo, Q., Xu, S., Su, Y., Li, Y., & Wu, F. (2015). A geometric method for wood-leaf separation using terrestrial and simulated lidar data. *Photogrammetric Engineering & Remote Sensing*, *81*(10), 767-776. https://doi.org/10.14358/pers.81.10.767

Vicari, M. B., Disney, M., Wilkes, P., Burt, A., Calders, K., & Woodgate, W. (2019). Leaf and wood classification framework for terrestrial lidar point clouds. *Methods in Ecology and Evolution*, *10*(5), 680-694. https://doi.org/10.1111/2041-210X.13144

Wang, D., Hollaus, M., & Pfeifer, N. (2017). Feasibility of machine learning methods for separating wood and leaf points from terrestrial laser scanning data. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *IV-2/W4*, 157–164. https://doi.org/10.5194/isprs-annals-iv-2-w4-157-2017

Wang, D., Takoudjou, S. M., & Casella, E. (2020). Lewos: A universal leaf-

wood classification method to facilitate the 3d modelling of large tropical trees using terrestrial lidar. *Methods in Ecology and Evolution*, *11*(3), 376–389. https://doi.org/10.1111/2041-210x.13342

Wehr, A., & Lohr, U. (1999). Airborne laser scanning — an introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing*, *54*(2-3), 68-82. https://doi.org/10.1016/s0924-2716(99)00011-8

Weinmann, M., Jutzi, B., & Mallet, C. (2014). Semantic 3d scene interpretation: A framework combining optimal neighborhood size selection with relevant features. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *II–3*, 181-188. https://doi.org/10.5194/isprsannals-ii-3-181-2014

West, K., Webb, B., Lersch, J., Pothier, S., Triscari, J., & Iverson, A. (2004). Context-driven automated target detection in 3d data. *Proceedings of SPIE, the International Society for Optical Engineering*, *5426*, 133–143. https://doi.org/10.1117/12.542536

Xi, Z., Hopkinson, C., & Chasmer, L. (2018). Filtering stems and branches from terrestrial laser scanning point clouds using deep 3-d fully convolutional networks. *Remote Sensing*, *10*(8), 1215. https://doi.org/10.3390/rs10081215

Xi, Z., Hopkinson, C., Rood, S. B., & Peddle, D. R. (2020). See the forest and the trees: Effective machine and deep learning algorithms for wood filtering and tree species classification from terrestrial laser scanning. *ISPRS Journal of Photogrammetry and Remote Sensing*, *168*(3), 1-16. https://doi.org/10.1016/j.isprsjprs.2020.08.001

Xu, H., Gossett, N., & Chen, B. (2007). Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics (TOG)*, *26*(4), 19. https://doi.org/10.1145/1289603.1289610

Zhu, X., Skidmore, A. K., Darvishzadeh, R., Niemann, K. O., Liu, J., Shi, Y., & Wang, T. (2018). Foliar and woody materials discriminated using terrestrial lidar in a mixed natural forest. *International Journal of Applied Earth Observation and Geoinformation*, *64*, 43–50. https://doi.org/10.1016/j.jag.2017.09.004