Improved Supervised Learning-Based Approach for Leaf and Wood Classification From LiDAR Point Clouds of Forests

Sruthi M. Krishna Moorthy, Kim Calders, Matheus B. Vicari, and Hans Verbeeck

***Abstract*— Accurately classifying 3-D point clouds into woody and leafy components has been an interest for applications in forestry and ecology including the better understanding of radiation transfer between canopy and atmosphere. The past decade has seen an increase in the methods attempting to classify leaves and wood in point clouds based on radiometric or geometric features. However, classification purely based on radiometric features is sensor-specific, and the method by which the local neighborhood of a point is defined affects the accuracy of classification based on geometric features. Here, we present a leaf-wood classification method combining geometrical features defined by radially bounded nearest neighbors at multiple spatial scales in a machine learning model. We compared the performance of three different machine learning models generated by the random forest (RF), XGBoost, and lightGBM algorithms. Using multiple spatial scales eliminates the need for an optimal neighborhood size selection and defining the local neighborhood by radially bounded nearest neighbors makes the method broadly applicable for point clouds of varying quality. We assessed the model performance at the individual tree- and plot-level on field data from tropical and deciduous forests, as well as on simulated point clouds. The method has an overall average accuracy of 94.2% on our data sets. For other data sets, the presented method outperformed the methods in literature in most cases without the need for additional postprocessing steps that are needed in most of the existing methods. We provide the entire framework as an open-source python package.**

***Index Terms*— Leaf versus wood separation, LiDAR, machine learning, python package, tropical forests.**

I. Introduction

RECENT advances in remote sensing have enabled us to observe the forest structure in 3-D and in unprecedented detail. Terrestrial laser scanning (TLS) is revolutionizing the field of forestry by enabling accurate measurements of various forest structural parameters. Parameters like vertical distribution of plant material and aboveground biomass (AGB) of trees have been successfully estimated from TLS data [1]–[5]. However, separating leaf from woody elements in a forest canopy is necessary for quantifying the vertical distribution of leaf area index (LAI) and for studying the woody component of evergreen vegetation. A recent study showed that woody materials in TLS data led to an overestimation of LAI between 3% and 32% when tested in the Bavarian Forest National Park with deciduous, coniferous, and mixed plots [6]. Accurately quantifying the distribution of leaves in forest canopy has important implications in understanding the radiation regime, photosynthetic processes, and carbon and water exchange between the canopy and the atmosphere [7]–[9]. In addition, the presence of leaves has shown to result in the overestimation of AGB of trees estimated from TLS using quantitative structure modeling (QSM) [5].

In the past decade, there have been significant advancements in the methods for leaf and wood separation from TLS data. The methods developed were either based on the radiometric features [10] or geometric features [8], [11]–[16] or a combination of both [17]. Since the radiometric features mainly depend on the wavelength used by a particular sensor, methods based on radiometric features become sensor-specific. However, the methods based on geometric features require only the x, y, and z coordinates of the points and, hence, are broadly applicable across data collected by different sensors. Geometric methods mainly use features based on eigenvectors and normal vectors calculated for each point. Both the eigenvectors and normal vectors are calculated for each point from its local neighborhood. The local neighborhood of a point can be defined either by specifying a fixed number of nearest neighbors for that point [11], [15] or by specifying a sphere of fixed radius with that point at the center called radially bounded nearest neighbors [8]. So far, the most common way of defining the neighborhood has been to specify the K-nearest neighbors, with K often ranging between 10 and 100 [14], [15], as it is more computational and memory-efficient than the radially bounded neighbors. As the geometric features for a point depend on the value of K, different studies have used different approaches to estimate the optimal K value for calculating these features. Wang *et al.* [14] showed that the method with an optimal adaptive neighborhood size performed better than the one using a fixed neighborhood size. To choose the optimal neighborhood

Manuscript received April 23, 2019; revised July 3, 2019, July 30, 2019, and August 22, 2019; accepted October 3, 2019. Date of publication October 31, 2019; date of current version April 22, 2020. This work was supported by the European Research Council (TREECLIMBERS) under Grant 637643. The work of K. Calders was supported by Belgian Science Policy Office (BELSPO) in the frame of the STEREO III Program—Project 3D-FOREST under Grant SR/02/355. The work of M. B. Vicari was supported by CNPq (National Council of Technological and Scientific Development—Brazil) through the Program Science Without Borders under Grant 233849/2014-9. *(Corresponding author: Sruthi M. Krishna Moorthy.)*

S. M. Krishna Moorthy, K. Calders, and H. Verbeeck are with the Computational and Applied Vegetation Ecology Lab, Department of Environment, Ghent University, 9000 Ghent, Belgium (e-mail: sruthi.krishnamoorthyparvathi@ugent.be; kim.calders@ugent.be; hans.verbeeck@ugent.be).

M. B. Vicari is with the Department of Geography, University College London, London WC1E 6BT, U.K. (e-mail: matheus.vicari.15@ucl.ac.u.k.).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2019.2947198

size, Wang *et al.* [14] minimized the eigen-entropy across different values of K ranging from 9 to 99 with an increment of 9. Recently, Vicari *et al.* [15] generated eigenvalue-based features that were proposed in literature [8], [18], for every point using different neighborhood sizes (K value ranging between 10 and 250, for example) and predicted leaf or wood class for every point in each of the neighborhood sizes. The final class for each point was then picked based on majority voting.

The other approach is radially bounded nearest neighbors, which defines the neighborhood using neighbors within a certain radius of a point. This approach has some advantages over using K-nearest neighbors of the point [8]. Unlike radially bounded nearest neighbor, geometry of a point defined by its K-nearest neighbors is influenced by the point cloud density. For instance, point density in the canopy is often lower than the points closer to the ground. Hence, the geometry of the leaves defined by its K neighbors in the understorey will be different from the geometry of the leaves defined by the same K neighbors in the canopy. Although this problem could partly overcome by downsampling the point cloud with a known maximum point spacing and/or by combining the features extracted from multiple neighborhood sizes [15], radially bounded nearest neighbors of a point are much less sensitive to variation in point density and, hence, broadly applicable to the point clouds of varying quality. In addition, LeGall *et al.* [19] demonstrated that radially bounded nearest neighbors result in features with consistent geometrical meaning compared with K-nearest neighbors.

Most of the existing methods based on geometric features have relied on using fixed-neighborhood size [8] or choosing an optimal neighborhood size using some heuristics like eigen-entropy as in [14] be its K-nearest neighbors or radially bounded neighbors. The use of multiple neighborhood sizes, both multiple K-nearest neighbors [20] and radially bounded neighbors [19], [21], has been successfully used for point cloud classification in other fields (mainly in urban or other man-made structurally less complex environments). Belton *et al.* [13] showed that combining the geometrical features derived from multiple search radii (between 0.1 and 0.5 m) had significant advantage over the geometrical features derived from a single search radius for the leaf-wood classification of giant red tingle trees. Moreover, almost all the existing methods based on the geometric features used eigenvalues or the features derived from eigenvalues for leaf-wood classification. While eigenvalues represent the magnitude of variance present in data, eigenvectors represent the direction of that variance. Though directional and orientation features are seldom used, a recent study improved the leaf-wood classification accuracy by including the orientation information (inclination angle in this case) of every point in its local neighborhood space along with radiometric and eigenvalue-based geometric features [6]. In addition, it is not clear how the existing methods would perform for tropical forest sites both at the individual tree-level and plot-level as the validation on tropical forest sites is limited (except for [15]). As most of these existing methods are not made available as open-source libraries, it makes the intercomparison of these methods difficult.

The main objective of the study is to develop a new method building up on existing works that is broadly applicable across different forest types for classifying leaf from wood points in 3-D data, purely based on the geometrical features of the points at local spatial scales. To overcome the above-described issues, in our study, we defined the local neighborhood of a point using radially bounded nearest neighbors. We opted for radially bounded nearest neighbors for points, as it enables the method to be broadly applicable for point clouds of varying point cloud density and quality. Since the orientation of leaf and wood points varies in the local neighborhood space, we also explored the features based on eigenvectors along with the normalized eigenvalues at multiple neighborhood sizes for classification. We defined the local neighborhood of a point at multiple spatial scales, as it eliminates the need for the selection of optimal neighborhood size for a data set. Most of the existing methods have used classical machine learning algorithms like RF [11], support vector machines (SVMs) [11], and Gaussian mixture models (GMMs) [8], [15] to classify leaf from wood points. Recent advancements in gradient boosting machines (GBMs) have made them the most popular choice for classification problems, and they have shown to outperform most of the other classical machine learning algorithms [22]. In this article, we compared the performances of three different supervised machine learning algorithms for the classification: the most commonly used RF and two other boosting methods, namely, LightGBM and XGBoost. In addition, the performance of the presented method is assessed at both the plot-level and the individual tree-level from simulated to temperate and tropical trees and tropical plots. We tested the predictive performance of the models using a cross-validation approach with the data from two tropical rainforest sites in this study. The model was tested on independent data sets that were not used for training the model from the study in [15]. The independent test data sets consisted of simulated data sets of four trees of different species and field data of nine trees (including both tropical and temperate). In addition, we compared the presented method with three other methods from the literature. We provide the entire framework for the leaf-wood classification as an open-source python package.

The specific objectives are the following.

1) To develop a new method for leaf-wood classification based on a supervised machine learning algorithm building up on the existing works.
2) To compare the performance of the presented method with some of the existing methods for leaf-wood classification.

II. Leaf-Wood Classification Model Development

A. Training and Validation Data Preparation

We collected TLS data from multiple plots of two different tropical forest sites: Gigante Peninsula, Panama (February–April, 2016) and Nouragues, French Guiana (August–October, 2017). The detailed description of the plot size, maximum canopy height, and data collection setup is given in Table I. The Gigante Peninsula is a

TABLE I
DESCRIPTION OF THE PLOTS WHERE THE TLS DATA WERE COLLECTED

Site	Name	Max. height (m)	Plot size	Scanner used	No. of scans per plot	Angular resolution used (deg)
Gigante Peninsula, Panama	GIG_1	26.5	15 m by 15 m	RIEGL VZ-400	4	0.02
	GIG_2	33				
	GIG_TREE	25.2	Isolated tree		-	
Nouragues, French Guiana	NOU_1	34.2	10 m (circle)	RIEGL VZ-1000	6	0.04
	NOU_2	43.6	10 m (circle)		4	
	NOU_3	41.6	10 m (circle)		6	
	NOU_TREE	31.4	Isolated tree		-	

≈60-year old secondary seasonal tropical forest with high liana density [2544 lianas ≥ 1-cm diameter at breast height (DBH) per ha] [23] and an estimated plant area index (PAI) of 6 [24]. Nouragues is an undisturbed old-growth lowland moist tropical forest with the medium liana density (1200 lianas ≥ 1 cm DBH per ha) [25] and an estimated PAI of 7 [26]. The scan locations were chosen in such a way as to get better visibility of the central tree and the surrounding lianas. We randomly distributed 20–35 retroreflective targets in the field to coregister the data from all the scan locations to get a single high-resolution 3-D point cloud with minimal occlusion [27]. All the data were collected either using RIEGL VZ-400 (Gigante Peninsula) or RIEGL VZ-1000 (Nouragues) scanners (RIEGL Laser Measurement Systems GmbH, Horn, Austria). Both are multiple return time-of-flight-based scanners using a narrow infrared laser beam of wavelength 1550 nm and a beam divergence of 0.35 mrad. Based on the manufacturer's specifications, the data from the two scanners are interoperable [28]. The registration of data from all scan positions for each plot was done using the RISCAN Pro software (version 2.5.3, RIEGL Laser Measurement Systems GmbH, Horn, Austria) provided by RIEGL.

Both GIG_TREE and NOU_TREE (as in Table I) were manually extracted using CloudCompare (version 2.8.1, CloudCompare, GPL software) [29] from a plot in Gigante and Nouragues, respectively.

The first step to build a leaf-wood classification model based on supervised machine learning is to label training data from which a classifier can learn. The data should be as representative of the real problem as possible to create a robust classifier. As indicated in Table I, all the coregistered TLS data from five plots and two isolated trees were manually labeled into two different classes: 0 being leaf points and 1 being wood points. The manual labeling was the most time-consuming part of the machine-learning pipeline. All the manual labeling was done in CloudCompare by an experienced person [29] and took almost 150 person-hours.

After manual labeling, we downsampled the TLS data using a voxel grid filter of size 0.02 m. We chose 0.02 m as the size for the voxel grid filter to make a tradeoff between retaining the important information in the point cloud and reducing the computational complexity [30]. Downsampling ensures a uniform distribution of points by reducing the number of redundant points, especially the points closer to the scanner. Though it is not absolutely necessary to perform downsampling, it is highly recommended as it would improve the computational and memory efficiency of the presented method by few-folds. In addition to downsampling, we also removed the ground points in the plot-level TLS data using the cloth simulation filter (CSF) implemented in CloudCompare and proposed in [31].

The classes are slightly imbalanced in the leaf-wood classification problem with the points belonging to wood class being fewer than the points belonging to leaf class. In our training data set, the wood points contributed to an average of only ≈15% of all the points. Sample labeled plot-level and tree-level data from these two different sites are shown in Fig. 1(a) and (b).

B. Predictor Variables

We used the normalized eigenvalues and zenith angle of the eigenvectors of every point in the TLS data from a plot, across multiple spatial scales, as the predictor variables to classify leaf and woody points. We defined the local neighborhood for each point by setting a sphere around each point with the following radii: 0.1, 0.25, 0.5, 0.75, and 1.0 m. We did not consider spheres with radii below 0.1 m, as it will render some of the points without sufficient neighbors, especially at the top of the canopy where there is minimal point density. For instance, in a dense tropical forest, the average nearest neighbor distance between points could be as much as 5 cm at 20 m from the ground [32]. The reason for not choosing the neighborhood size beyond 1 m is to minimize the memory requirements of the proposed method. As the number of neighbors a point could have within 1-m radius could sometimes exceed 100 000, especially closer to the ground, this could easily lead to memory error when computing neighbors for the whole point cloud.

The eigenvalues and vectors for each point along with its neighbors at every spatial scale are calculated using the steps in [13]. First, we compute the covariance matrix of every point with its neighbors for all the five spatial scales. Then, the eigenvectors and the corresponding eigenvalues are calculated for the covariance matrix of every point at all the spatial scales. We then normalized the eigenvalues between 0 and 1 by dividing each eigenvalue by the sum of all three eigenvalues. The normalized eigenvalues for each spatial scale

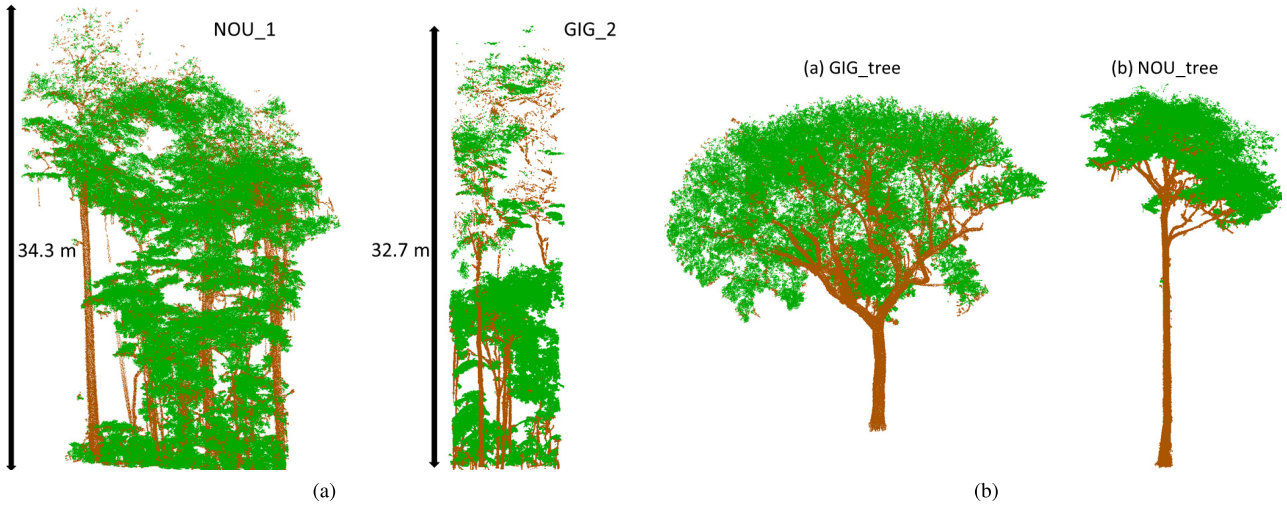

Fig. 1. Illustration of manually labeled point cloud at (a) plot-level data and (b) tree-level data from two different sites from Gigante Peninsula and from Nouragues. Green: leaf points. Brown: woody points.

are represented as λ_1, λ_2, λ_3 (x m), where $\lambda_1 > \lambda_2 > \lambda_3$ and $x \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$. We also calculated the zenith angle of all the three eigenvectors for every point at all five spatial scales that are represented as θ_1, θ_2, θ_3 (x m), where $\theta_1 > \theta_2 > \theta_3$ and $x \in \{0.1, 0.25, 0.5, 0.75, 1.0\}$. As a result, we had 30 features in total to classify leaf from wood points.

The eigenvalues and zenith angle of the eigenvectors adequately capture the different spatial distributions of trunk, branch, and leaf points at a local spatial scale, as shown in Fig. 2(a) and (b) for GIG_TREE data in Table I. This difference in spatial distributions enables the separation of leaf from wood points. As shown in Fig. 2(a), trunk and branch points have a dominant direction of variance indicated by a relatively higher λ_1 than λ_2 and λ_3. Leaf points on the other hand have similar values for both λ_1 and λ_2. The zenith angle of the eigenvectors varies among trunk, branch, and leaf points in its local neighborhood [Fig. 2(b)]. For instance, the zenith angle of the first eigenvector for tree points is close to 0° or 180°, while it is not the case for the leaf and branch points. In addition, it is evident from the figure that the leaf and branch points have different zenith angle compositions [see also Fig. 2(c)].

The main reason for choosing the radially bounded neighbors is to make the method broadly applicable for point clouds of varying point cloud density and quality compared with K-nearest neighbors (see [19] and Fig. 3). The figure shows the eigenvalue λ_1 computed using K-nearest neighbors and radially bounded neighbors for the GIG_TREE data in Table I before and after downsampling. The eigenvalues were computed at a spatial scale of 0.5 m for the radially bounded neighbor method and with a value of 150 for the K-nearest neighbor method. We downsampled with a voxel grid filter of size 0.04 m. As shown in Fig. 3(a), the λ_1 remains unchanged before and after downsampling when radially bounded neighbors were used, whereas the λ_1 changes (especially at $z = 101$ m) between the two point clouds when K-nearest neighbors are used. As a result, the optimal value of K would depend on the point cloud density and quality and has to be chosen carefully to derive the local geometrical properties of the points.

We also defined the local geometry of all the points at five different spatial scales between 0.1 and 1 m. Fig. 4 shows the distribution of eigenvalue λ_1 at three different spatial scales (0.25, 0.5, and 0.75 m) calculated for GIG_TREE. As shown in the figure, λ_1 at 0.25-m spatial scale has the highest value for small branches (< 10 cm) and λ_1 at 0.5- and 0.75-m spatial scales peaks at big branches (> 10 cm), while the λ_1 for leaves remains smaller and relatively the same across all the three spatial scales. This shows that using multiple spatial scales might have advantages over single or an optimal spatial scale. All three eigenvalues at all the five spatial scales are shown in Appendix A.

C. Model Development

We built a leaf-wood classification model using a supervised machine-learning algorithm. We chose three different ensemble-based machine-learning classifiers to develop the leaf-wood classification model. Ensemble classifiers combine the output from multiple weak learners, as opposed to individual classifiers that depend on one strong learner, and have proved to be more beneficial than the individual classifiers in a variety of fields including remote sensing [33]. Empirical studies have shown the bagging and boosting methods, using decision trees as base learners, to outperform single classifiers most of the time, making them very popular for myriad of applications [34]. In this study, we chose a bagging method named RF and two boosting methods such as lightGBM and XGBoost.

1) Random Forest: One of the main reasons for choosing RF for building the leaf-wood classification model is that RF is robust to outliers, and hence avoids overfitting by building multiple parallel decision trees using random subsets of data and random subsets of features for each tree [35]. A recent study has shown that RF outperformed other individual machine-learning classifiers like Naive Bayes, neural networks, and so on, for leaf and wood classification

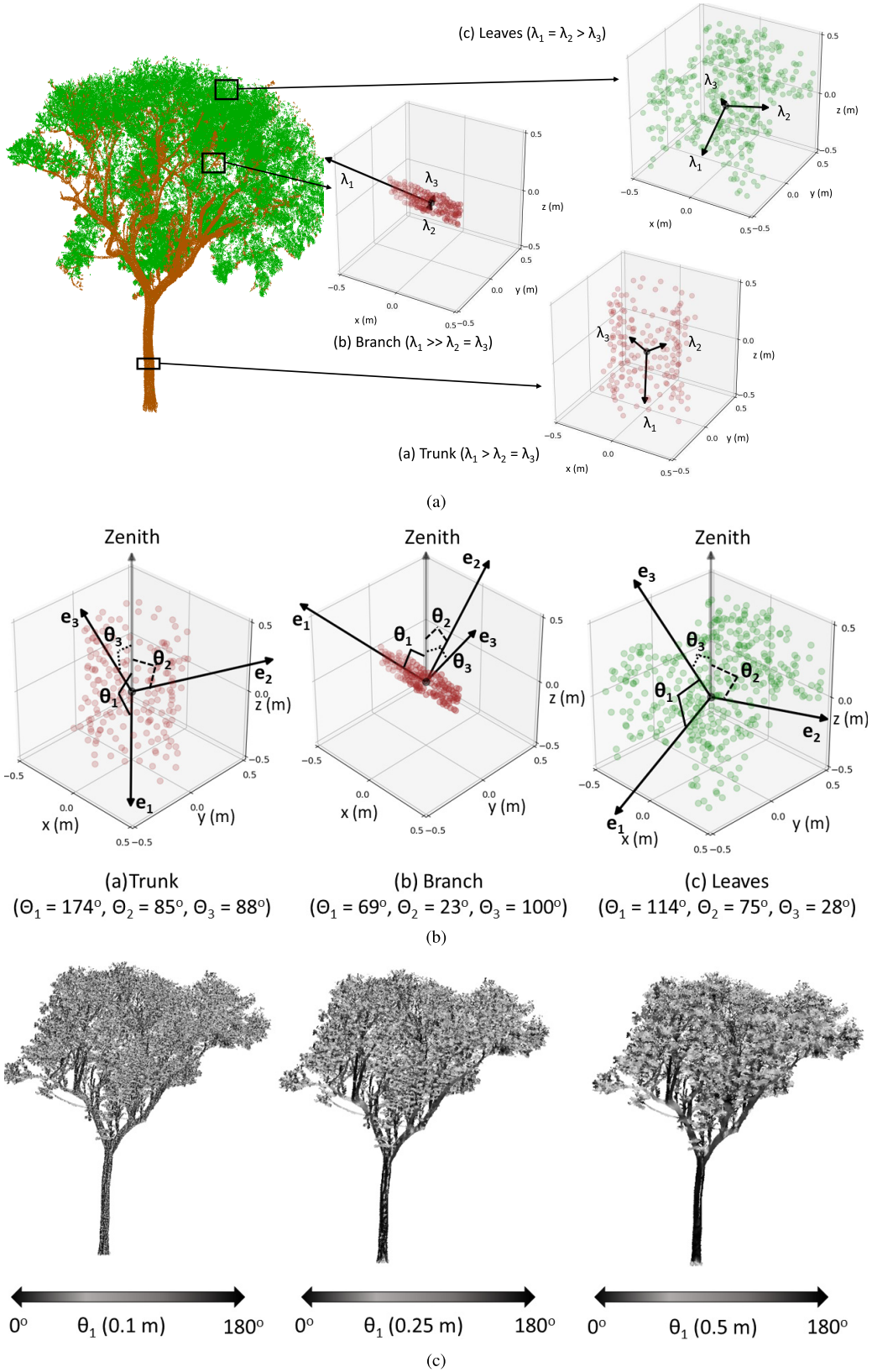

Fig. 2. Illustrating (a) eigenvalues and (b) zenith angle of the eigenvectors to represent the different spatial distribution of (a) trunk, (b) branch, and (c) leaf points at a local spatial scale (0.5 m in this case). (c) Zenith angle of the first eigenvector θ_1 calculated for the spatial scales 0.1, 0.25, and 0.5 m for GIG_TREE in Table I. The color scales at the bottom of each figure indicate the range of θ_1.

of individual trees based on geometric features in temperate forests [11]. To choose the optimal number of decision trees to build the model, we analyzed the performance of the models built with 10–80 decision trees by cross-validation. We used the default value of $\sqrt{30}$ as in [35] for the number of features to be used for building each decision tree.

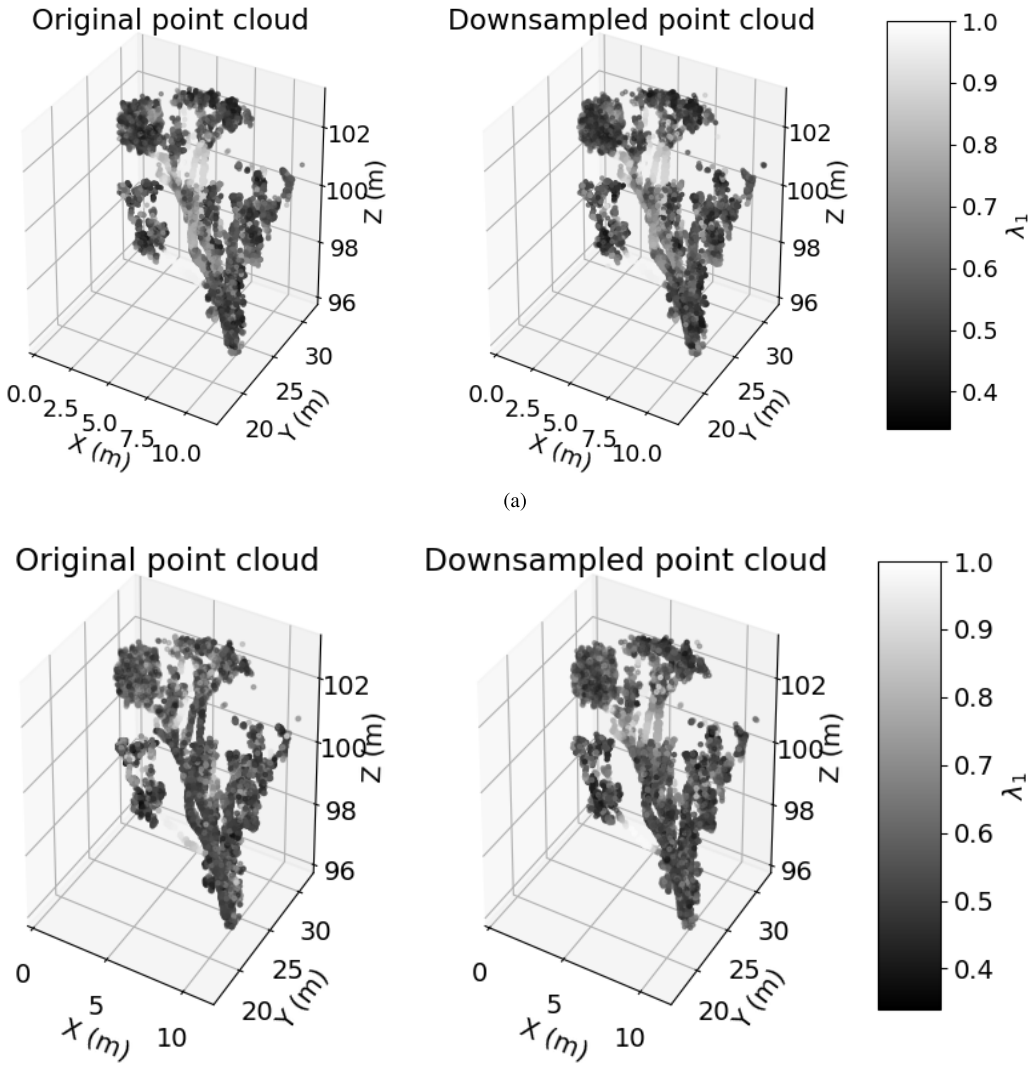

Fig. 3. Illustrating the difference between the eigenvalues derived for a point with its local neighborhood defined based on (a) radially bounded neighbors and (b) K-nearest neighbors. Both (a) and (b) indicate the values for λ_1 derived at a spatial scale of 0.5 m for the radially bounded neighborhood in (a) and derived with 150 nearest neighbors in (b).

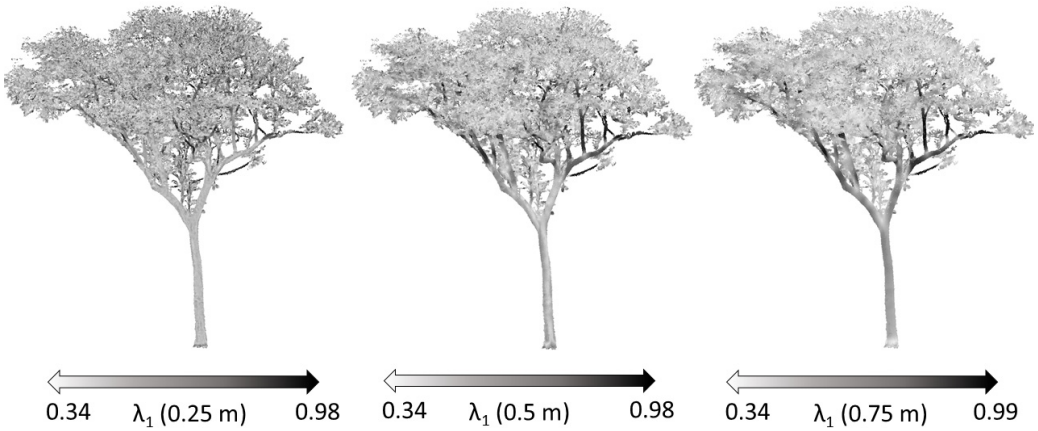

Fig. 4. Eigenvalue λ_1 calculated for the spatial scales 0.25, 0.5, and 0.75 m for GIG_TREE in Table I. The color scales at the bottom of each figure indicate the range of λ_1.

2) XGBoost and LightGBM: XGBoost [22] and lightGBM [36] are two of the most popular choices based on boosting for different machine-learning applications. In this study, we used decision trees as our base learner for both the boosting methods. Unlike RF, which is based on bagging, boosting trees are built sequentially. At any given instant, the current decision

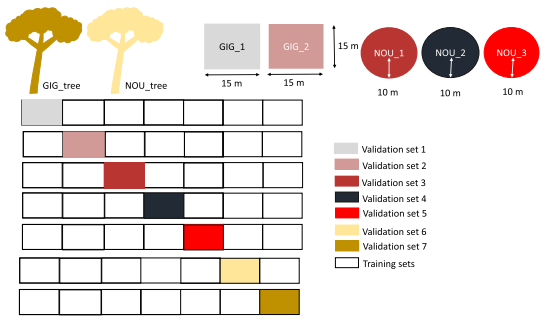

Fig. 5. Sevenfold spatial cross-validation approach for assessing the predictive performance of the leaf-wood classification model.

tree is built based on the outcomes from the previously built decision trees. While the implementation of both XGBoost and lightGBM are similar, lightGBM improves on XGBoost in its training speed and the data set size it can handle. However, both the methods require extensive parameter tuning to improve the performance over other methods like RF. We used randomized search for parameter tuning and chose the optimal parameter combination by cross-validated search over the different parameter settings. More details on the hyperparameter optimization for XGBoost and lightGBM is given in Appendix B.

The model performance to choose the optimal hyperparameters for all the three classifiers was assessed using the performance metrics described in Section II-D. We then built the final models using the optimal parameter combination and compared the predictive performance of these models using a sevenfold spatial cross-validation approach.

D. Model Predictive Performance

We assessed the performance of all the three leaf-wood classification models using a sevenfold spatial cross-validation approach [37]. In the sevenfold spatial cross-validation approach, we trained the model on six of the seven data sets described in Table I and tested on the seventh data set. As a result, the performance of the model is assessed on all the available data sets while still efficiently using all the data for building the model, as shown in Fig. 5

In addition to assessing the performance by a cross-validation approach, we used two independent test data sets: simulated data and field data from the study in [15]. The sevenfold cross-validation approach required building seven different models, each built on different combinations of training data as explained above. The final model, whose performance is assessed on independent data sets, is built on all seven training data sets mentioned in Table I. All the independent test data sets are at the individual tree-level. The simulated data sets are from four 3-D tree models simulated using the Monte Carlo ray-tracing library, librat [38]–[40]. Four 3-D tree models used are *Acer platenoides* (ACPL), *Almus glutinosa* (ALGL3), *Betula pendula* (BEPE2), and *Tilio cordata* (TICO2). All nine field trees were scanned using the RIEGL VZ400 laser scanner. Of the nine trees, three trees are from Alice Holt, U.K., represented as alice1, alice2, and alice3, two trees are from Caxiuana, Brazil, represented as caxiuanaA117 and caxiuanaA21, two trees are from Nouragues, French Guiana, represented as nouraguesH20 108 and nouraguesH20 13, and two trees are from Ankasa Forest Reserve, Ghana, represented as tree 13 and tree 2. The height of the trees ranged from 18 m (alice 3) to 45 m (nouraguesH20 108). For more details on both of these data sets, refer to the original article of [15].

We use accuracy, F_1 score for leaf and wood for assessing the performance of the model as in [15]. We defined leaf as the positive class to estimate the F_1 score for leaf class and wood as the positive class to estimate the F_1 score for wood class.

Accuracy quantifies the percentage of correctly classified points.

F_1 score is the harmonic mean of both precision and recall with 1 as its best value and 0 as its worst

$$F_1 \text{ score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{1}$$

Precision is the fraction of correctly classified positive class points [true positives (TPs)] out of all the classified positive points [TP and false positives (FPs)]

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2}$$

Recall is the fraction of correctly classified positive class points (TP) out of all the positive class points in the data [TP and false negative (FN)]

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{3}$$

The positive class in this case would be leaf for F_1 score (leaf) and would be wood for F_1 score (wood) as mentioned above.

E. Intercomparison With Existing Methods

We compared the performance of the presented method with three other methods in literature [8], [13], [15] on the independent data sets from [15] described in Section II-D. While the result of the method proposed in [15] on these data sets is readily available from the manuscript, we had to implement the two other methods based on the descriptions in their original manuscripts to be able to compare their performance against our method. We described a wide range of methods available for leaf-wood classification in Section I. Of all the methods, some of the methods relied on intensity values for separating leaf from wood points. Since the intensity values are sensor-specific and we had access to neither their methods nor data sets, we chose to compare our methods only against the ones based on geometric features [10], [17]. Of all the geometric-based methods, we only implemented those methods for which we could gather all the necessary information. For instance, it was not clear how and what neighborhood size was chosen for some of the methods [11], [12]. This left us with two methods that could be used for comparison, and these two methods that we finally chose were both based on the GMMs [8], [13]. However, it was not very clear how the GMMs were initialized in

TABLE II

DETAILED INFORMATION ON THE FEATURES USED BY THE THREE EXISTING METHODS FOR LEAF-WOOD CLASSIFICATION

Reference	Features	Neighborhood type	Neighborhood sizes	Method
[8]	λ_2, $\lambda_0 - \lambda_1$, $\lambda_1 - \lambda_2$	Fixed, radial neighbor	0.45 m	GMM
[13]	λ_0, λ_1, λ_2, $\lambda_1 - \lambda_0$, $\lambda_2 - \lambda_1$	multiple, radial neighbors	0.1, 0.2, 0.3, 0.4 m	GMM
[15]	λ_2, $\lambda_0 - \lambda_1$, $\lambda_1 - \lambda_2$, $\lambda_1 - \lambda_2/\lambda_1$, $\lambda_0 - \lambda_1/\lambda_0$, $\sum_{i=0}^{2} \lambda_i * \log \lambda_i$	multiple, K nearest neighbors	10 to 250 in steps of 10	GMM

TABLE III

COMPARISON OF RF, XGBOOST, AND LIGHTGBM MODELS USING ACCURACY, F_1 SCORE (LEAF), AND F_1 SCORE (WOOD) FOR ALL THE PLOTS AND TREES SHOWN IN TABLE I

Tree name	Random Forest model			XGBoost model			LightGBM model		
	Accuracy	F_1 score (leaf)	F_1 score (wood)	Accuracy	F_1 score (leaf)	F_1 score (wood)	Accuracy	F_1 score (leaf)	F_1 score (wood)
GIG_1	95.6	0.98	0.78	95.7	0.98	0.79	95.5	0.98	0.78
GIG_2	93.6	0.96	0.8	92.3	0.95	0.74	92.2	0.95	0.74
GIG_TREE	90.1	0.94	0.77	90.2	0.94	0.78	90.3	0.94	0.78
NOU_1	95.4	0.97	0.82	95.3	0.97	0.82	95.1	0.97	0.82
NOU_2	96.3	0.98	0.89	96.3	0.98	0.89	95.9	0.98	0.88
NOU_3	92.2	0.95	0.74	92.4	0.96	0.76	92.3	0.95	0.75
NOU_TREE	95.9	0.98	0.88	95.8	0.98	0.88	95.7	0.97	0.88
Average	**94.2**	**0.97**	**0.81**	**94**	**0.97**	**0.81**	**93.8**	**0.96**	**0.80**

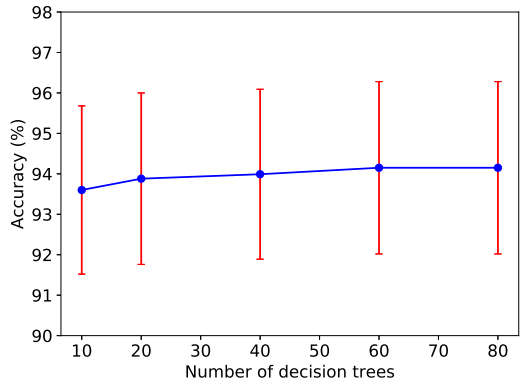

Fig. 6. Performance of the model built with varying number of decision trees assessed by average accuracy from the sevenfold cross validation approach explained in Fig. 5. The models assessed for varying number of decision trees were built with all the 30 features.

both the cases. To avoid any bias, we generated our own machine learning model based on RF (with 60 decision trees) with the features and neighborhood sizes used in those two studies. The RF model was built on the training data set explained in Section II-A using the features in Table II. This way, we empirically compare the performance of the features presented in this article with the features presented in the two chosen studies using the metrics mentioned in Section II-D. Detailed information on the features and neighborhood type and sizes used by all the three methods is given in Table II.

III. RESULTS

A. Model Development

The choice of the number of decision trees had little impact on the performance of the RF model with the average accuracy of the model from the sevenfold spatial cross validation merely increasing from 93.6% to 94.15%. We chose 60 decision trees as the RF model's accuracy stayed at 94.15% beyond 60 decision trees (see Fig. 6).

Since the number of hyperparameters optimized were eight for both the XGBoost and lightGBM models, it is hard to visualize the performance of the optimal model compared with the others. The final optimal parameter combination for both the models is given in Appendix B.

We compare the accuracy of three models for each plot and tree in Table I using the sevenfold cross validation approach (see Table III). As indicated in the table, while all the three models on average performed well, the RF model had a slightly better accuracy than XGBoost and XGBoost (and in turn RF) performed slightly better than lightGBM across all performance metrics. As a result, we chose RF as our predictive model to perform further analysis of its performance to classify leaf from wood points.

B. Predictive Performance of Random Forest Model

1) Performance of Eigenvector-Based Features: We compared the performance of the RF model with and without the features based on eigenvectors. Most of the existing studies have used features based on eigenvalues but not eigenvectors. However, as shown in Fig. 7, zenith angles derived from the eigenvectors increased the F_1 score (wood) by 3%.

Our RF model with features based on both eigenvalues and vectors at multiple spatial scales showed an overall average accuracy of 94.2% with individual accuracy ranging between 90.1% and 96.3% (Table III). F_1 score for the leaf class of the model was high with an average of 0.97. On the other hand, the F_1 score for the wood class of the model was relatively low with an average value of 0.81. An example of this classification for NOU_2 and GIG_TREE can be seen in Fig. 8.

As shown in Fig. 8 and Table III, the method performed equally well at both the individual tree- and plot-level.

2) Intercomparison of Methods on the Independent Test Data Sets From [15]: We evaluated and compared the performance of our method with three methods from the literature on the independent data sets from [15] (Table IV).

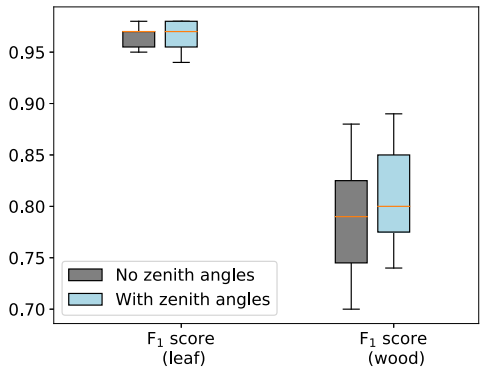

Fig. 7. Performance of the RF model built with and without zenith angles of the eigenvectors. F_1 score (leaf) and F_1 score (wood) are from the sevenfold cross validation approach explained in Fig. 5.

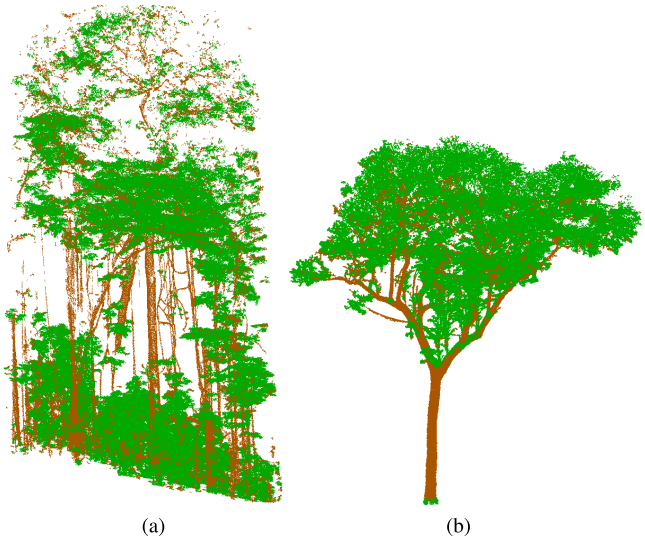

Fig. 8. Illustration of the model predictions for (a) NOU_2 plot and (b) GIG_TREE.

The independent data sets consist of both simulated and field data sets, as explained in [15].

The presented method outperformed the methods implemented based on the features proposed in [8] and [13]. Though [8] and [13] used similar features based on eigenvalues, they both differed in the number of local neighborhoods chosen to derive the geometric features. As the results indicate, our RF method based on [13], which combined the features from multiple neighborhoods, clearly outperformed the method based on [8], which used fixed, single neighborhood size.

For the simulated data sets, the performance of the proposed method was consistent with that of the method proposed in [15]. The method performed better for ACPL and TICO2 than for ALGL3 and BEPE2. The proposed method has higher F_1 score for leaf class and a lower F_1 score for wood class than the study in [15]. The lower F_1 score for the wood points of ALGL3 and BEPE2 could be mainly attributed to the geometrical properties of the leaves, which are similar to thin long branches, as shown in Fig. 9. The F_1 score for the leaf points still remains high as wood points belong to the minority class and misclassifications in the minority class do not reflect in the performance metrics of the majority class as much as they do in the minority class. It is clear from Fig. 9 that we need additional information compared with just geometric properties to distinguish between leaf and wood for such tree species.

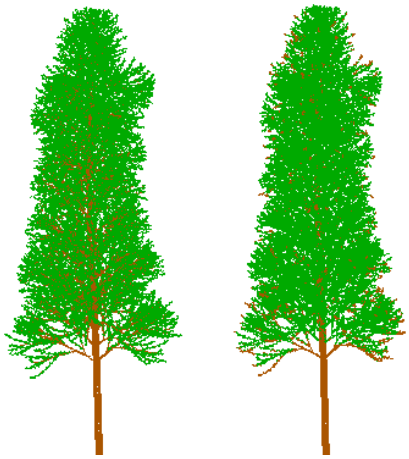

Fig. 9. Ground truth and model predictions for the simulated point cloud ALGL3, using Monte Carlo ray tracing. (Left) Ground truth for the whole tree with green representing the actual leaf points and brown representing the actual woody points. (Right) RF model predictions for the whole tree with green representing the predicted leaf points and brown representing the predicted woody points.

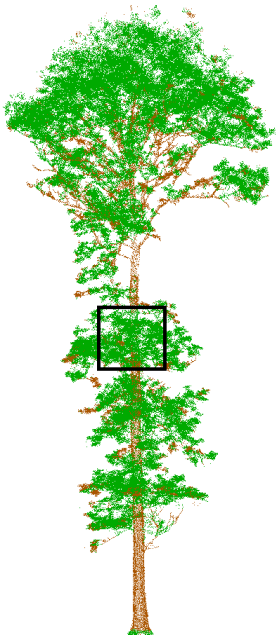

Fig. 10. Illustration of the model predictions for alice 3 tree in Table IV. Black square: part of the trunk misclassified as leaf points by the proposed method.

For the field data, the proposed method outperformed the method presented in [15] even without any additional post-processing steps for most of the trees except the Alice Holt trees from U.K. (alice 1, alice 2, and alice 3 in Table IV). Though the F_1 score for wood points were much lower for all three Alice Holt trees, our visual inspection revealed that the method was able to detect most of the big and small wood structures except for few regions surrounded by dense vegetation (indicated by the black square in Fig. 10).

TABLE IV
COMPARISON OF ACCURACY, F_1 SCORE (LEAF), AND F_1 SCORE (WOOD) FOR BOTH THE SIMULATED AND FIELD DATA SETS FROM [15]

Type	Tree name	Our method			[8] method			[13] method			[15] method		
		Accuracy	F_1 score (leaf)	F_1 score (wood)	Accuracy	F_1 score (leaf)	F_1 score (wood)	Accuracy	F_1 score (leaf)	F_1 score (wood)	Accuracy	F_1 score (leaf)	F_1 score (wood)
Simulated	ACPL	84.5	0.88	0.78	77.8	0.84	0.62	72.1	0.78	0.61	89	0.92	0.84
	ALGL3	85.5	0.91	0.55	87.2	0.93	0.54	84.1	0.91	0.5	77	0.81	0.70
	BEPE2	73.5	0.8	0.62	74	0.81	0.59	72.6	0.8	0.59	79	0.76	0.81
	TICO2	84.8	0.90	0.67	82.8	0.89	0.59	80.1	0.87	0.58	89	0.93	0.79
	Average	**82.1**	**0.87**	**0.66**	**80.45**	**0.87**	**0.58**	**77.2**	**0.84**	**0.57**	**83**	**0.85**	**0.78**
Field	alice 1	86	0.92	0.6	84.5	0.91	0.47	86.5	0.92	0.57	90	0.94	0.74
	alice 2	85	0.91	0.4	79	0.88	0.05	83	0.9	0.29	90	0.94	0.69
	alice 3	79	0.87	0.46	76.5	0.86	0.25	82	0.89	0.53	86	0.90	0.69
	caxiuanaA117	89	0.93	0.71	86.5	0.92	0.63	85.5	0.91	0.65	89	0.93	0.67
	caxiuanaA21	89	0.93	0.76	79	0.88	0.34	84	0.9	0.62	85	0.91	0.64
	nouraguesH20 108	87	0.91	0.73	81.5	0.89	0.45	88	0.92	0.74	93	0.95	0.82
	nouraguesH20 13	90.5	0.94	0.76	87	0.92	0.55	88	0.93	0.68	89	0.93	0.69
	tree 13	92	0.93	0.92	68	0.79	0.33	87	0.9	0.81	90	0.92	0.86
	tree 2	91	0.92	0.89	66.5	0.77	0.36	89	0.91	0.86	91	0.92	0.89
	Average	**87.6**	**0.92**	**0.69**	**78.7**	**0.87**	**0.38**	**87**	**0.91**	**0.64**	**89**	**0.92**	**0.74**
	Average without alice sp.	**89.75**	**0.93**	**0.8**	**78.08**	**0.86**	**0.44**	**88.6**	**0.91**	**0.73**	**89**	**0.92**	**0.76**

IV. DISCUSSION

In this article, we presented a method for classifying leaf from wood points purely based on the geometrical properties of the points at different local spatial scales. We also compared various state-of-the-art machine learning classification algorithms and found that the model built by the RF algorithm yielded the best results. As indicated in Table III, the model performs uniformly across the two different tropical sites included in this study. The two sites included plots with their canopy heights ranging from 25 to 44 m and also included trees belonging to multiple species. The performance of the proposed method on the data sets from [15] further confirms the broader applicability of the method to trees from the sites that were not part of the training data set. For instance, the model performed with an average accuracy of 87.6 when tested on trees from temperate forests (in U.K.) and three different tropical forest sites (one from Brazil, French Guiana, and Ghana).

A. Field Versus Simulated Data Set

We tested our method on both the simulated and field data. The simulated data consisted of 3-D tree models from four different species, simulated using the Monte Carlo ray tracing library, librat [38]–[40]. One of the main advantages of the simulated data is that we know the true class values, and as a result, they could serve as true validation data sets for evaluating an algorithm. Though we could carefully label all the points collected by a scanner in the field, it is a tedious task and it is easy to mislabel points especially in the areas of low point density. As a result, simulated point clouds could serve as a true validation set when available. An additional benefit of having simulated data when using supervised machine learning is that they can also be used as the training data set. For instance, as shown in Fig. 9, our method failed to accurately separate leaf from wood points for the species ALGL partly due to the lack of similar examples in our training data set. These simulated data sets hold promise for generating new reliable training data sets for improving the predictions from the machine learning algorithms. However, to simulate realistic virtual 3-D tree models, intense field data collection is necessary to parameterize and construct them [38]. The more complex the nature and structure of an environment is, the more difficult it is to simulate realistic models. Considering the complex nature of the tropical forest structure, field testing is necessary to test the performance of the method in tropical environments truly.

B. Impact of Scanner and Scanning Protocol on the Method

The quality of the 3-D point cloud could have an impact on the performance of the presented method. Data quality is highly dependent on the scanner and the scanning protocol used in the field. Therefore, it is important to understand how the scanner type and the scanner protocol used can have an impact on the leaf-wood classification results. Scanner's beam divergence and resolution can influence the quality of the point cloud. All the data used in this study were collected using RIEGL VZ-400, which is a high-resolution multireturn TLS. As indicated in Table I, we used two different resolutions to collect the data from the same scanner, and our method yielded similar results for both these resolutions. In the case of TLS, minimum size of the object distinguishable by the scanner increases with the increasing distance from the scanner. Since it is not possible to resolve an object (leaf or a small branch) smaller than the beam divergence of the scanner used, it is important to choose a similar high-resolution scanner to get reproducible results [41]. While beam divergence determines

the minimum size of the object resolvable at a given distance, multireturn for a given pulse is capable of detecting several objects within a single outgoing pulse. Studies have demonstrated the advantages of multireturn LiDAR over single- or first-return LiDAR in complex forests [32].

Data quality is dependent not only on the scanner type but also on the scanning protocol used. The quality of the point cloud can be assessed by using the vertically resolved average nearest neighbor distance as a proxy [27]. As shown in [27], the average nearest neighbor distance (point spacing) increased, especially in the mid and upper canopies, with lower sampling densities. This effect of occlusion is expected to be amplified when scanning in forests with dense understory vegetation. Similarly, higher sampling density should be used when using a first-return LiDAR to reduce the impact of occlusion. Therefore, scanning protocols should be adapted according to the forest and scanner types, following the guidelines published for using TLS in forest plots [27], [41]. These guidelines on the scanning protocol are already being implemented by the ecosystem research and monitoring networks like the terrestrial ecosystem research network (TERN) in Australia, to enable data acquisition of desirable quality from TLS [41].

Dense understory vegetation not only amplifies the impact of occlusion away from the scanner but also affects the performance of the method closer to the scanner, as shown in Fig. 10. Additional postprocessing steps proposed in other studies might aid in improving the performance for this type of trees [8], [15].

C. Intercomparison With Existing Methods

In addition, we also compared our method with the three state-of-the-art leaf-wood classification methods proposed in [8], [13], [15]. Eigenvector-based features proposed in this method have clear advantages over the eigenvalue-based features used in [13], and multiple local spatial scales clearly outperform the geometric features derived from a single spatial scale as evident from the results in [8]. Intercomparison of the results between our method and the method in [15] indicates that there is no one solution that fits for all when it comes to classifying leaf and wood points in natural environments like forests. While the presented method outperformed the method in [15] for most of the trees, it clearly failed for few trees. In addition, the presented method has an average accuracy comparable with the existing methods without the need for additional postprocessing steps, which is a part of most of the existing methods. As a result, the misclassified woody points (as in Figs. 8 and 10) can be corrected using the postprocessing steps proposed in literature [8], [15]. This means that the existing postprocessing steps in literature would have a considerably more accurate starting point when fused with the presented classification method. This might lead to even better results than the ones currently reported, which needs further confirmation. The combination of current method with multiple existing approaches could be the best way to move forward. However, the nonavailability of most of the existing methods and standardized reference data sets make it difficult to take the leap forward, as discussed in [15].

D. Broader Applicability of the Proposed Method

One of the main challenges of point cloud classification in natural environments like forests is the data heterogeneity, which is a consequence of forest structural complexity, varying scanner configurations, varying scanning protocols such as multi or single scanning setups, and so on. As a result, it is challenging to develop generalizable and transferable methods in natural environments compared with other computer vision methods, which are mostly developed and applied in structurally less complex man-made environments. Nevertheless, the proposed method based on multiscale radially bounded neighbors, to some degree, mitigates this problem by being more robust to point clouds of varying density and quality than the methods based on K-nearest neighbors. Our method is based on a machine learning model trained on data sets from two tropical forest sites. However, the independent test data sets included broad-leaved trees from one temperate and three other tropical forest sites that were not part of the training set. This indicates the broader applicability of the presented method. However, there are other forest types (e.g., boreal, savannah, and so on), where the method is yet to be evaluated. Though there is no guarantee that the method will work on these forest types, we make our model available to the community to incrementally train and test it on their own data sets. We provide the entire leaf-wood classification method as an open-source python package to enable the community to perform the intercomparison of different methods and fusion of our method with other methods. Extracting eigenvalues at multiple spatial scales on a small plot with approximately 500 000 points can be completed within 10 min even on a single-core machine. The main specifications of the machine we used to run the algorithm are the following: dual boot (Ubuntu 16.04 LTS and Windows 10), Intel Xeon E5-1650 v4, 6 cores, and 128-GB RAM. The python package has been tested on both the Ubuntu and Windows operating systems. Although the current implementation of our method has high computational efficiency, it comes at the cost of high memory requirements as we keep the neighbors of all the points in the point cloud in memory. One of the ways to overcome the memory issue is to downsample the point cloud. As the proposed method is more robust to point clouds of varying density than existing methods based on K-nearest neighbors, downsampling should not have an impact on the final outcome (see Fig. 7). In addition, memory requirements would be higher when working with plot-level data than the individual tree-level data. As a result, one practical solution to overcome the memory issue would be to split the plot-level data into smaller subplots and calculate the eigenvalue and vectors for each of the subplots. More details on the practical implementation and tips to deal with the memory issue can be found in the github page: https://github.com/sruthimoorthy/leaf_wood_clf.

V. Conclusion

We present a robust method for leaf-wood classification from the 3-D data of tropical rainforest plots. The presented method combines the geometric features based on radially bounded nearest neighbors for points at multiple spatial scales

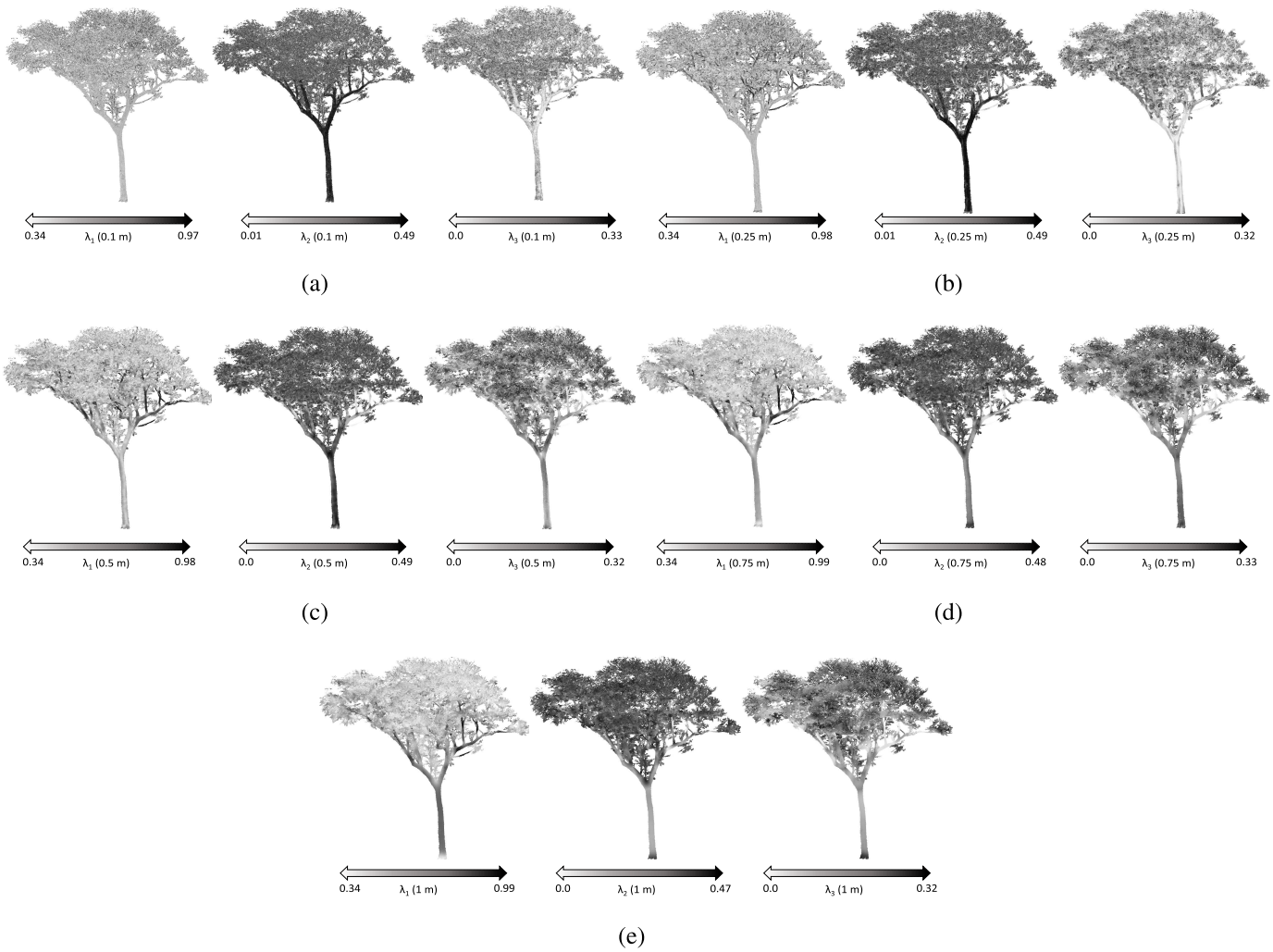

Fig. 11. Illustrating the effectiveness of eigenvalues at multiple scales for classifying leaf and wood points in tropical forests (GIG_TREE in Table I). The ground truth for GIG_TREE is shown in Fig. 1(b). The color scales at the bottom of each figure indicate the range of λ_1, λ_2, and λ_3 at (a) 0.1, (b) 0.25, (c) 0.5, (d) 0.75, and (e) 1 m spatial scale from left to right, respectively.

in a machine learning model. The main advantage of the method is that it eliminates the need for the selection of optimal neighborhood size, which is a requirement for most of the state-of-the-art methods, thus making it completely automated and userfriendly. The choice of defining the local neighborhood using radially bounded nearest neighbors makes the method broadly applicable to the 3-D data of varying point cloud density and quality. We show the robustness of the method across a wide range of test data sets from simulated to temperate trees and tropical forest trees and plot. In addition, we also provide the entire leaf versus wood classification model as an open-source python package for the scientific community to explore and contribute for future improvements of the model.

APPENDIX A
EIGENVALUE AT MULTIPLE SPATIAL SCALES FOR GIG_TREE

The eigenvalues λ_1, λ_2, and λ_3 for the GIG_TREE at the spatial scales of 0.1, 0.25, 0.5, 0.75, and 1 m are shown in Fig. 11.

APPENDIX B
XGBOOST AND LIGHTGBM: HYPERPARAMETER OPTIMIZATION

A. XGBoost

We tuned the following set of parameters for XGBoost. The range of values and the optimal value chosen for each of the parameters is given next to each parameter.

1) *max_depth:* (Range = [3, 5, 7, 9], Optimum = 9). This parameter defines the maximum depth of a decision tree.
2) *min_child_weight:* (Range = [1, 2, 3, 4, 5, 6], Optimum = 3). This parameter defines the minimum sum of weights of all observations in a leaf node.
3) *gamma:* (Range = [0, 0.1, 0.2, 0.3, 0.4, 0.5], Optimum = 0.1). This parameter defines when to make a split.
4) *subsample:* (Range = [0.6, 0.7, 0.8, 0.9], Optimum = 0.7). The parameter defines the fraction of the data to be randomly sampled to build each tree.
5) *colsample_bytree:* (Range = [0.6, 0.7, 0.8, 0.9], Optimum = 0.8). The parameter defines the fraction of all the features to be randomly sampled to build each tree.

6) alpha = (Range = [1e-5, 1e-2, 0.01, 0.05, 0.1, 1, 2, 10, 20], Optimum = 0.1). L1 Regularization term.
7) lambda = (Range = [1e-5, 1e-2, 0.01, 0.05, 0.1, 1, 2, 10, 20], Optimum = 1). L2 Regularization term.
8) learning_rate = (Range = [0.001, 0.01,0.1], Optimum = 0.01). This parameter controls the learning rate of the model.

B. LightGBM

We tuned the following set of parameters for lightGBM. The range of values and the optimal value chosen for each of the parameters is given next to each parameter.

1) *num_leaves:* (Range = 20 randomly generated values between 6 and 50, Optimum = 45). This parameter defines the maximum number of leaf nodes in a decision tree.
2) *min_child_samples:* (Range = 20 randomly generated values between 100 and 500, Optimum = 211). This parameter defines the minimum number of data allowed in a leaf node.
3) *min_child_weight:* (Range = [1e-5, 1e-3, 1e-2, 1e-1, 1, 1e1, 1e2, 1e3, 1e4], Optimum = 1). This parameter defines the minimum sum of weights of all observations in a leaf node.
4) *subsample:* (Range = [0.6, 0.7, 0.8, 0.9], Optimum = 0.6). The parameter defines the fraction of the data to be randomly sampled to build each tree.
5) *colsample_bytree:* (Range = [0.6, 0.7, 0.8, 0.9], Optimum = 0.8). The parameter defines the fraction of all the features to be randomly sampled to build each tree.
6) *alpha:* (Range = [1e-5, 1e-2, 0.01, 0.05, 0.1, 1, 2, 10, 20], Optimum = 2). L1 Regularization term.
7) *lambda:* (Range = [1e-5, 1e-2, 0.01, 0.05, 0.1, 1, 2, 10, 20], Optimum = 20). L2 Regularization term.
8) *learning_rate:* (Range = [0.001, 0.01,0.1], Optimum = 0.001). This parameter controls the learning rate of the model.

The number of estimators or decision trees in both the cases was set to 1000 with an early stopping condition to stop the training, when the performance on an independent validation set did not increase in 30 iterations.

ACKNOWLEDGMENT

The authors would like to thank the Smithsonian Tropical Research Institute (STRI) for their support during the fieldwork in Gigante and J. Chave, P. Gaucher, and other members of the Nouragues Ecological Research Station managed by CNRS Guyane, for permission with collecting data from the plots. They would also like to thank A. Mastelinck, P. Gaucher, and F. Jeanne for their help during their stay at the Nouragues Research Station. They declare no conflict of interest.

S. M. Krishna Moorthy and H. Verbeeck conceived and designed the experiments with inputs from K. Calders and M. B. Vicari; S. M. Krishna Moorthy and M. B. Vicari collected the data; S. M. Krishna Moorthy performed the analysis; S. M. Krishna Moorthy wrote this article with critical contributions from all the authors.

The entire framework for leaf-wood classification is implemented as an open-source python package and is available in the following github page: https://github.com/sruthimoorthy/leaf_wood_clf. The codes are written in Python scripting language and have been tested in both Windows and Ubuntu. The data used for building the classifier can be obtained by sending an e-mail to the corresponding author (sruthi.krishnamoorthyparvathi@ugent.be).

REFERENCES

[1] D. L. B. Jupp, D. S. Culvenor, J. L. Lovell, G. J. Newnham, A. H. Strahler, and C. E. Woodcock, "Estimating forest LAI profiles and structural parameters using a ground-based laser called 'Echidna," *Tree Physiol.*, vol. 29, no. 2, pp. 171–181, 2009.
[2] K. Calders, J. Armston, G. Newnham, M. Herold, and N. Goodwin, "Implications of sensor configuration and topography on vertical plant profiles derived from terrestrial LiDAR," *Agricult. Forest Meteorol.*, vol. 194, pp. 104–117, Aug. 2014.
[3] K. Calders, T. Schenkels, H. Bartholomeus, J. Armston, J. Verbesselt, and M. Herold, "Monitoring spring phenology with high temporal resolution terrestrial LiDAR measurements," *Agricult. Forest Meteorol.*, vol. 203, pp. 158–168, Apr. 2015.
[4] K. Calders *et al.*, "Variability and bias in active and passive ground-based measurements of effective plant, wood and leaf area index," *Agricult. Forest Meteorol.*, vol. 252, pp. 231–240, Apr. 2018.
[5] K. Calders *et al.*, "Nondestructive estimates of above-ground biomass using terrestrial laser scanning," *Methods Ecol. Evol.*, vol. 6, no. 2, pp. 198–208, Nov. 2015.
[6] X. Zhu *et al.*, "Improving leaf area index (LAI) estimation by correcting for clumping and woody effects using terrestrial laser scanning," *Agricult. Forest Meteorol.*, vol. 263, pp. 276–286, Dec. 2018.
[7] G. B. Bonan, "Importance of leaf area index and forest type when estimating photosynthesis in boreal forests," *Remote Sens. Environ.*, vol. 43, no. 3, pp. 303–314, 1993.
[8] L. Ma, G. Zheng, J. U. H. Eitel, L. M. Moskal, W. He, and H. Huang, "Improved salient feature-based approach for automatically separating photosynthetic and nonphotosynthetic components within terrestrial LiDAR point cloud data of forest canopies," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 679–696, Feb. 2016.
[9] J. Liu *et al.*, "Variation of leaf angle distribution quantified by terrestrial LiDAR in natural European beech forest," *ISPRS J. Photogramm. Remote Sens.*, vol. 148, pp. 208–220, Feb. 2019.
[10] M. Béland, D. D. Baldocchi, J.-L. Widlowski, R. A. Fournier, and M. M. Verstraete, "On seeing the wood from the leaves and the role of voxel size in determining leaf area distribution of forests with terrestrial LiDAR," *Agricult. Forest Meteorol.*, vol. 184, pp. 82–97, Jan. 2014.
[11] D. Wang, M. Hollaus, and N. Pfeifer, "Feasibility of machine learning methods for separating wood and leaf points from terrestrial laser scanning data," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 157–164, Sep. 2017.
[12] T. Yun, F. An, W. Li, Y. Sun, L. Cao, and L. Xue, "A novel approach for retrieving tree leaf area from ground-based LiDAR," *Remote Sens.*, vol. 8, no. 11, p. 942, 2016.
[13] D. Belton, S. Moncrieff, and J. Chapman, "Processing tree point clouds using Gaussian mixture models," in *Proc. ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, Antalya, Turkey, 2013, pp. 11–13.
[14] D. Wang *et al.*, "Separating tree photosynthetic and non-photosynthetic components from point cloud data using dynamic segment merging," *Forests*, vol. 9, no. 5, p. 252, 2018.
[15] M. B. Vicari, M. Disney, P. Wilkes, A. Burt, K. Calders, and W. Woodgate, "Leaf and wood classification framework for terrestrial LiDAR point clouds," *Methods Ecol. Evol.*, vol. 10, no. 5, pp. 680–694, 2019.
[16] S. Tao *et al.*, "A geometric method for wood-leaf separation using terrestrial and simulated Lidar data," *Photogramm. Eng. Remote Sens.*, vol. 81, no. 10, pp. 767–776, 2015.
[17] X. Zhu *et al.*, "Foliar and woody materials discriminated using terrestrial LiDAR in a mixed natural forest," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 64, pp. 43–50, Feb. 2018.
[18] Z. Wang *et al.*, "A multiscale and hierarchical feature extraction method for terrestrial laser scanning point cloud classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2409–2425, May 2015.

[19] Y. LeGall, H. Thomas, F. Goulette, J.-E. Deschaud, and B. Marcotegui, "Semantic classification of 3D point clouds with multiscale spherical neighborhoods," in *Proc. Int. Conf. 3D Vis. (3DV)*, Sep. 2018, pp. 390–398.
[20] T. Hackel, J. D. Wegner, and K. Schindler, "Fast semantic segmentation of 3D point clouds with strongly varying density," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 3, pp. 177–184, Jul. 2016.
[21] K. Koenig, B. Höfle, M. Hämmerle, T. Jarmer, B. Siegmann, and H. Lilienthal, "Comparative classification analysis of post-harvest growth detection from terrestrial LiDAR point clouds in precision agriculture," *ISPRS J. Photogramm. Remote Sens.*, vol. 104, pp. 112–125, Jun. 2015.
[22] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 785–794.
[23] G. M. F. van der Heijden, J. S. Powers, and S. A. Schnitzer, "Lianas reduce carbon accumulation and storage in tropical forests," *Proc. Nat. Acad. Sci. USA*, vol. 112, no. 43, pp. 13267–13271, 2015.
[24] M. E. Rodríguez-Ronderos, G. Bohrer, A. Sanchez-Azofeifa, J. S. Powers, and S. A. Schnitzer, "Contribution of lianas to plant area index and canopy structure in a Panamanian forest," *Ecology*, vol. 97, no. 12, pp. 3271–3277, 2016.
[25] S. A. Schnitzer, S. J. DeWalt, and J. Chave, "Censusing and measuring lianas: A quantitative comparison of the common methods," *Biotropica*, vol. 38, no. 5, pp. 581–591, 2006.
[26] L. H. Emmons *et al.*, "Seasonal change in leaf-area index at three sites along a South American latitudinal gradient," *Ecotropica*, vol. 12, pp. 87–102, Jan. 2006.
[27] P. Wilkes *et al.*, "Data acquisition considerations for terrestrial laser scanning of forest plots," *Remote Sens. Environ.*, vol. 196, pp. 140–153, Jul. 2017.
[28] K. Calders *et al.*, "Evaluation of the range accuracy and the radiometric calibration of multiple terrestrial laser scanning instruments for data interoperability," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 5, pp. 2716–2724, May 2017.
[29] D. Girardeau-Montaut. (2011). *CloudCompare OpenSource Project*. [Online]. Available: http://www.danielgm.net/cc/
[30] A. Burt, M. Disney, and K. Calders, "Extracting individual trees from lidar point clouds using treeseg," *Methods Ecol. Evol.*, vol. 10, no. 3, pp. 438–445, 2018.
[31] W. Zhang *et al.*, "An easy-to-use airborne LiDAR data filtering method based on cloth simulation," *Remote Sens.*, vol. 8, no. 6, p. 501, Jun. 2016.
[32] S. M. K. Moorthy, K. Calders, M. E. Di Porcia Brugnera, S. A. Schnitzer, and H. Verbeeck, "Terrestrial laser scanning to detect liana impact on forest structure," *Remote Sens.*, vol. 10, no. 6, p. 810, 2018.
[33] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, Sep. 2006.
[34] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *J. Artif. Intell. Res.*, vol. 11, pp. 169–198, Aug. 1999.
[35] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[36] G. Ke *et al.*, "LightGBM: A highly efficient gradient Boosting decision tree," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3146–3154.
[37] J. N. Goetz, A. Brenning, H. Petschko, and P. Leopold, "Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling," *Comput. Geosci.*, vol. 81, pp. 1–11, Aug. 2015.
[38] J.-L. Widlowski *et al.*, "The fourth phase of the radiative transfer model intercomparison (RAMI) exercise: Actual canopy scenarios and conformity testing," *Remote Sens. Environ.*, vol. 169, pp. 418–437, Nov. 2015.
[39] P. Lewis, "Three-dimensional plant modelling for remote sensing simulation studies using the Botanical Plant Modelling System," *Agronomie*, vol. 19, nos. 3–4, pp. 185–210, 1999.
[40] M. Disney, P. Lewis, and P. Saich, "3D modelling of forest canopy structure for remote sensing simulations in the optical and microwave domains," *Remote Sens. Environ.*, vol. 100, no. 1, pp. 114–132, Jan. 2006.
[41] K. Calders, P. Wilkes, M. Disney, J. Armston, M. Schaefer, and W. Woodgate, "Terrestrial LiDAR for measuring above-ground biomass and forest structure," in *Effective Field Calibration and Validation Practices*. Taipei, Taiwan: TERN, 2018, ch 19.

Sruthi M. Krishna Moorthy received the bachelor's degree in computer science engineering from Anna University, Chennai, India, in 2011, the M.S. degree in artificial intelligence from KULeuven University, Leuven, Belgium. She is currently pursuing the Ph.D. degree with the CAVElab, Department of Environment, Ghent University, Ghent, Belgium, under the supervision of Prof. H. Verbeeck.

Her research interests include understanding how different ecological processes interact using innovative remote sensing and computer vision techniques.

Kim Calders received the B.Sc. and M.Sc. degrees in bioscience engineering from KULeuven, Leuven, Belgium, in 2006 and 2008, respectively, the M.Sc. degree in remote sensing from University College London (UCL), London, U.K., in 2010, and the Ph.D. degree in LiDAR remote sensing from Wageningen University, Wageningen, The Netherlands, in 2015.

He was a Post-Doctoral Researcher with the National Physical Laboratory, Department of Geography, UCL from 2015 to 2017. Since 2017, he has been a Post-Doctoral Researcher with CAVElab, Ghent University, Ghent, Belgium. His research interests include measurements of full 3-D vegetation structure and how this is related to airborne or spaceborne signals.

Matheus B. Vicari received the bachelor's degree in environmental engineering from the Universidade de Passo Fundo, Passo Fundo, Brazil, in 2013, the M.S. degree in remote sensing from Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, and the Ph.D. degree in remote sensing from University College London, London, U.K.

His research interests include remote sensing and computer vision techniques to better understand and characterize tree and forest structural variations.

Hans Verbeeck received the B.Sc. and M.Sc. degrees in bioscience engineering from Ghent University, Ghent, Belgium, in 2000 and 2002, respectively, and the Ph.D. degree in vegetation modeling from Antwerp University, Antwerp, Belgium, in 2007.

From 2007 to 2008, he was a Post-Doctoral Researcher with LSCE, and the Laboratory of Plant Ecology, Ghent University from 2009 to 2014. From 2015 to 2018, he was a Tenure-Track Research Professor with the Department of Environment, Ghent University, where he is currently an Associate Professor and also a Principal Investigator of CAVElab-Computational and Applied Vegetation Ecology. He studies vegetation dynamics and biogeochemical cycling in terrestrial ecosystems. His research interests include all types of terrestrial ecosystems, especially ecology of tropical forest ecosystems. In addition to vegetation modeling, the study of vegetation structure and biomass using terrestrial laser scanning is one of his main research lines.