JEODEZİ VE JEOİNFORMASYON DERGİSİ
JOURNAL OF GEODESY AND GEOINFORMATION
ISSN : 2147-1339
e-ISSN: 2667-8519

Araştırma Makalesi / *Research Article*
Yıl / *Year:* 2025 **Cilt** / *Volume:* 12 **Sayı** / *Issue*: 2
Sayfa / *Page*: 179-193
Dergi No / *Journal No*: 122
Doi: 10.9733/JGG.2025R0013.E

Comparison of machine learning algorithm performances in digital terrain model generation

Abdullah Can Ozen[1], Ozgul Vupa Cilengiroglu[2,*]

[1]*Dokuz Eylul University, The Graduate School of Natural and Applied Sciences, İzmir, Türkiye.*

[2]*Dokuz Eylul University, Faculty of Sciences, Statistics Department, İzmir, Türkiye.*

Abstract: *LiDAR technology enables precise distance measurements by emitting laser pulses that reflect off surface objects, allowing for the calculation of spatial coordinates. Alongside spatial data associated color values of LiDAR points can be extracted from images captured by onboard cameras. As the laser beams reflect upon their initial contact with surfaces, the resulting point cloud must be appropriately classified to support specific analytical or operational objectives. This study uses different machine learning methods to sort and label LiDAR point cloud data into ground and non-ground points, then compares how well each method works. For this purpose, a dataset acquired by an unmanned aerial vehicle over the Democratic Republic of Congo was utilized. The dataset comprises 114,557 points, each described by three geometric features (DeltaH, Verticality, 3rd Eigenvalue) and two normalized color attributes (Red and Green Ratios), derived from RGB values. A total of ten machine learning algorithms were implemented and assessed. Among them, the XGBoost algorithm demonstrated the highest classification accuracy at 84.1%, while the Naive Bayes algorithm yielded the lowest accuracy, at 72.4%.*

Keywords: Remote sensing, LiDAR, Photogrammetry, Machine learning, Classification

Sayısal arazi modeli oluşturmada makine öğrenme algoritma performanslarının karşılaştırılması

Öz: *LiDAR teknolojisi, yüzey nesnelerinden yansıyan lazer darbeleri göndererek hassas mesafe ölçümleri yapılmasına olanak tanır ve bu sayede mekânsal koordinatların hesaplanması mümkün olur. Mekânsal verilerin yanı sıra, LiDAR noktalarına ait renk bilgileri de araç üzerindeki kameralarla çekilen görüntülerden elde edilebilir. Lazer ışınları yüzeylerle ilk temas ettikleri anda yansıdığından, ortaya çıkan nokta bulutunun belirli analizsel veya operasyonel amaçlara hizmet edebilmesi için uygun şekilde sınıflandırılması gerekmektedir. Bu çalışmada, LiDAR nokta bulutu verilerini sıralamak ve analiz etmek için çeşitli makine öğrenmesi yöntemleri kullanılmış ve her bir yöntemin performansı karşılaştırılmıştır. Bu amaçla, insansız hava aracı ile Demokratik Kongo Cumhuriyeti'nde elde edilen bir veri seti kullanılmıştır. Veri seti, üç geometrik özellik ve iki renk bilgisi içeren toplam 114 557 noktadan oluşmaktadır. On farklı makine öğrenmesi algoritması uygulanmış ve değerlendirilmiştir. Bu algoritmalar arasında XGBoost, %84.1 ile en yüksek sınıflandırma doğruluğunu gösterirken, Naive Bayes algoritması ile %72.4 ile en düşük doğruluğa ulaşılmıştır.*

Anahtar Sözcükler: Uzaktan algılama, LiDAR, Fotogrametri, Makine öğrenmesi, Sınıflandırma

* Sorumlu Yazar/Corresponding Author: Tel: +90 232 301 8562

Geliş Tarihi/Received: 04.06.2025
Kabul Tarihi/Accepted: 29.09.2025

0009-0005-9285-1447, abdullahcan.ozen@ogr.deu.edu.tr (Ozen A. C.)
0000-0003-0181-8376, ozgul.vupa@deu.edu.tr (Cilengiroglu O. V.)*

1. Introduction

With the rapid advancement of remote sensing technologies, their adoption has become increasingly widespread across various sectors, including engineering, architecture, industry, and construction (Almohsen, 2024). These innovations have made the measurement and analysis process faster, more precise, and more efficient, enabling significant developments in different sectors. Today, modern remote sensing methods like laser scanning, Interferometric Synthetic Aperture Radar (InSAR), photogrammetry, Synthetic Aperture Radar (SAR), and LiDAR (Light Detection and Ranging) are essential tools in engineering. LiDAR is especially important in many projects because it provides highly accurate and high-density three-dimensional terrain data, which is essential for generating detailed and reliable digital elevation models (DEMs) (Liu et al., 2007). This study specifically focuses on binary classification to distinguish ground from non-ground points in LiDAR data.

Accuracy is becoming increasingly important in creating a Digital Terrain Model (DTM) in three-dimensional (3D) modeling, especially in map production. Advances in technology have made it more feasible to meet these needs. The improved capabilities of measurement devices and the increased processing capacity of computers and algorithms enable more efficient processing of collected data. However, using rule-based algorithms and data reduction processes to summarize the point cloud leads to data loss, resulting in a loss of accuracy in DEM production.

In recent years, machine learning methods have become widely used and have taken on a key role in engineering and map production. Recent studies have shown the growing relevance of machine learning-based classification in 3D point cloud analysis, not only for terrain modeling but also for complex architectural and heritage documentation tasks. For example, Teruggi et al. (2020) proposed a hierarchical multi-resolution classification approach using machine learning, which demonstrated reliable and replicable results across large-scale datasets. Özdemir et al. (2019) tested and evaluated several classification algorithms for aerial point cloud data, including both deep learning and traditional machine learning methods, demonstrating the effectiveness of these approaches in high-resolution remote sensing applications. The proposed network, Mo-Net, simplifies shape coordinates by mapping them into a smaller, moment-friendly space. This makes it easier for the model to learn efficiently, using less memory and processing power, and also helps improve classification accuracy compared to similar approaches (Joseph-Rivlin et al., 2019). Since LiDAR data processing demands high computational resources and cost, two common approaches are used: developing new machine learning models and tools that reduce reliance on command-based LiDAR data. However, it is worth noting that reducing or interpolating LiDAR data is not always the ideal choice in industrial applications (Gharineiat et al., 2022).

According to Kuçak (2022), in his work with laser scanning data, the RANSAC algorithm can produce high-precision and complete three-dimensional geometric models, resulting in reliable 3D data that is important for restoration and other engineering works. Maturana and Scherer (2015) pointed out that machine learning is valuable for understanding land shapes and surface features. These algorithms are generally grouped into two types: supervised and unsupervised. Since unsupervised learning does not use training data to build models, it is not ideal for tasks requiring high accuracy. Wu et al. (2019) showed that supervised learning gives better results than unsupervised methods. Duran et al. (2021) conducted a study to evaluate the effectiveness of various machine learning algorithms in classifying LiDAR point cloud data. In the survey, 21 geometric features were derived from both the photogrammetric model and the LiDAR point cloud. A total of nine different machine learning algorithms were assessed. The results indicated that the multilayer neural network achieved the highest classification accuracy, reaching 96%. The percentage of correctly classified points among 15 577 sample points in Columbus, Ohio, amounts to 96.5%. Random Forest (RF) method yields outcomes much closer to the ground truth than earlier classification approaches (Park & Guldmann, 2019). In their study, Jakovljevic et al. (2019) showed that the model

well processed both classes by obtaining recall values above 0.70 and precision values above 0.85 for both ground and non-ground points in the classification of point clouds based on deep learning using an artificial neural network in pixel-based evaluation (Jakovljevic et al., 2019). Kang et al. (2017) showed, experiments demonstrate that the Bayesian Network classifier can effectively distinguish four types of basic ground objects, including ground, vegetation, trees, and buildings, with a high accuracy of over 90%.

The novelty of this study lies in the integration of selected geometric and normalized color features for ground classification, using a large-scale LiDAR dataset acquired over a topographically diverse area. Additionally, the study compares ten machine learning algorithms on a balanced dataset to assess performance under complex terrain conditions.

2. Materials and Methods

2.1 Materials

Various machine learning methods were used to analyze the geometric and color features taken from the LiDAR point cloud. These methods are ordered as classification tree (CT), bagging (BCT), random forest (RFC), logistic regression (LR), K-nearest neighbors (KNN), support vector machine (SVM), Naive Bayes (NB), neural networks (ANN), multilayer perceptron (MLP), and XGBoost (XGB). The analyses were done in R (version 4.2.3) and Python (version 3.11), using a 5% significance level ($\alpha = 0.05$).

The dataset utilized in this study was acquired via LiDAR technology during an aerial survey conducted over Kinshasa, Democratic Republic of Congo, in March 2023. An international construction company provided the data used in this study and dataset includes a point cloud of over 43 million points. The LiDAR system used was a RIEGL mini VUX-1UAV scanner operating at 100 kHz pulse frequency, mounted on a fixed-wing UAV flying at 150 meters above ground level with a speed of approximately 60 km/h. The scan angle was ±30°, and the data included only first returns. It has a spatial density of 15 points per square meter and covers 27 682.6 hectares. The sample points used in the study, shown in Figure 1, come from various land types, such as steep and flat areas, developed and undeveloped zones, forests, and open land, with a density of 1 point per square meter. Accordingly, 114 557 points, including 45 203 ground points and 69 354 other points, were analyzed over an area of approximately 12 hectares.

Figure 1: *Study area on Google Earth map*

Özen and Çilengiroğlu (2024), used a total of 9391-point data from March 2023 in a different region of the same city, consisting of 4766 terrain points and 4625 non-terrain points distributed over an area of approximately 1 hectar, with 1

point/m^2. The most significant difference between the two datasets is the larger sample size and the inclusion of a valley topography to challenge the prediction performance of the models (Figure 2).

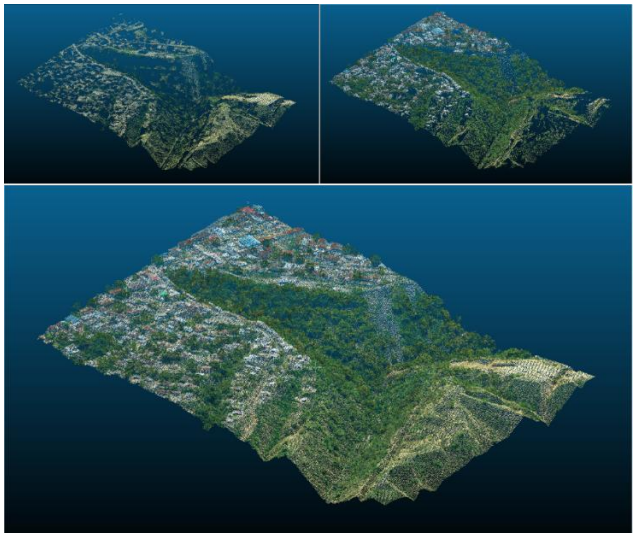

Figure 2: *LiDAR study data*

2.1.1 Variables

Geometric and color features were taken from the LiDAR point cloud. The geometric features include Verticality, DeltaH, and the 3rd Eigenvalue. The color features include red, green, blue, and brightness. In total, more than 20 different geometric variables, including Verticality and 3rd Eigenvalue variables, were found to be used in the studies in literature. A total of over 20 geometric features commonly found in the literature—such as linearity, planarity, eigenentropy, and surface variation—were initially considered. To ensure model efficiency and avoid multicollinearity, a correlation matrix was computed for all candidate features. DeltaH, Verticality, and 3rd Eigenvalue were selected based on their low correlation with one another and high predictive contribution in preliminary model runs. Features exhibiting high pairwise correlation or low variance were excluded. This selection aimed to reduce model complexity while preserving classification accuracy. In addition to all these variables, DeltaH and color-based variables for each point were also calculated and included in the models. The model contributions and model performances of all variables were evaluated, and as a result, it was decided to analyze the models with 5 variables (DeltaH, Verticality, 3rd Eigenvalue, Red Ratio and Green Ratio).

Verticality refers to the angular deviation of the normal vector of a point from the vertical axis (Z-axis) (Figure 3). This measurement is used to analyze how vertical or horizontal surfaces or points are. It is particularly useful in LiDAR data for detecting steep surfaces such as walls of buildings, tree trunks, etc. Verticality is usually expressed as a value, not as an angle. This value is scaled between 0 and 1.

If the normal vector of a point is called $\vec{N} = (N_x, N_y, N_z)$ and the verticality of this point is called V, it is calculated as:

$$V = 1 - |N_z| \tag{1}$$

Here N represents the normal vector of the point, and V denotes verticality. Here, it can be considered as:

- $V = 1$, fully vertical (e.g., wall)
- $V = 0$, fully horizontal (e.g., floor)

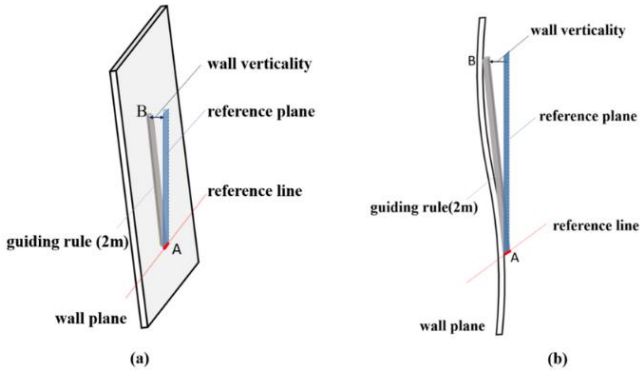

Figure 3: *Visual representation of verticality (Tan et al., 2023)*

DeltaH is a variable that represents the vertical deviation of a point relative to its surrounding surface. It is calculated as the difference between a given LiDAR point's Z-value (elevation) and the neighboring triangles' average Z-value that share this point as a vertex (Figure 4). These neighboring triangles are generated through Delaunay triangulation based on Voronoi diagrams constructed from the entire LiDAR point set. This metric provides insight into the relative height of the center point with respect to the local surface geometry defined by its neighbors. Considering the number of neighbor points as n, the height (Z) values of the points as H, and the DeltaH value at the center point as Δ_{H_m} it is written as follows:

$$\bar{H} = \sum_{i=1}^{n} H_i \tag{2}$$

$$\Delta_{H_m} = H_m - \bar{H} \tag{3}$$

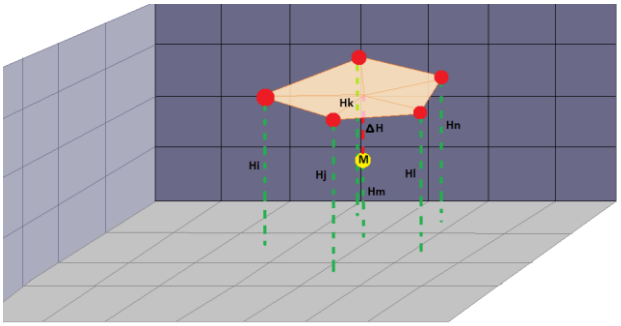

Figure 4: *Visual representation of DeltaH*

3rd Eigenvalue is a value calculated in the neighborhood analysis of a point that describes the local structure of the point cloud geometry (Figure 5). This value helps us understand the size and direction of the distribution around a point. The smallest eigenvalue refers to the distribution perpendicular to the surface. In an area with a radius of 3 m, neighbor points are selected, the number of these neighbor points is N, the coordinates of these points are $p(x, y, z)$, and the covariance matrix (C) is calculated using the neighbor points as follows:

$$C = \frac{1}{N}\sum_{i=1}^{N}(p_i - \bar{p})(p_i - \bar{p})^T \tag{4}$$

In Equation 4, C represents the covariance matrix calculated from the 3D coordinates of N neighboring points. Each p_i is the position vector of a neighboring point, and $\bar{p}$ is the mean position vector. The term $(p_i - \bar{p})(p_i - \bar{p})^T$ captures the deviation of each point from the mean, and their average defines the local geometric distribution around the central point. After solving this covariance matrix with eigenvalues $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, $\boldsymbol{\lambda}_3$ for the 3rd Eigenvalue;

- $\boldsymbol{\lambda_3}$ is the smallest eigenvalue and indicates how flat the local surface is.
- If $\boldsymbol{\lambda_3}$ is very small, the points are concentrated on a flat surface.
- A large value of $\boldsymbol{\lambda_3}$ indicates a coarser, rougher, or curved surface.

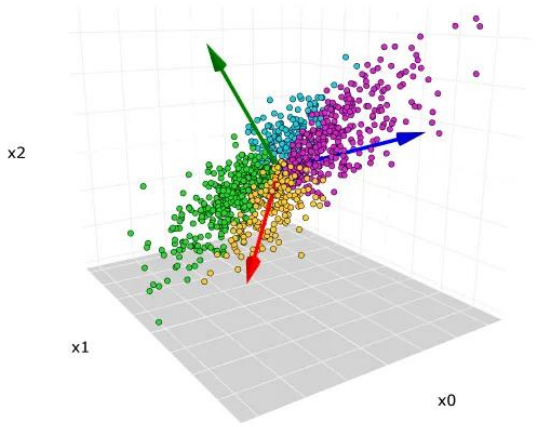

Figure 5: *Visual representation of 3rd Eigenvalue (URL-1)*

Every point in the LiDAR point cloud has a color value based on the Red-Green-Blue (RGB) model, which was recorded when the data was collected. The RGB values for each point have an 8-bit depth per channel, allowing for intensity values ranging from 0 to 255 for each color component.

The red ratio (RR) shows how much red there is compared to the total amount of red, green, and blue ($R + G + B = T$). The green ratio (GR) is calculated in the same way, showing the amount of green. The blue ratio (BR) represents the share of blue in the overall color. In Equations 5 to 7, R represents red value, G represents green value, B represents blue value and T represents sum of color values ($R + G + B$). These ratios represent the contribution of the color components to the overall color distribution over each point and take a value between 0 and 1.

$$RR = R/T \tag{5}$$

$$GR = G/T \tag{6}$$

$$BR = B/T \tag{7}$$

Brightness is calculated by dividing the sum of the red, green, and blue values by the highest possible color value. This means the brightness shows how much light there is at a point. A higher brightness means more light is present. For example, a shaded spot high up will have lower brightness than a spot in direct sunlight.

Since color values are perceived in 8-bit depth, each color component (Red, Green, Blue) ranges from 0 to 255. This allows the total color value ($R + G + B$) to be between 0 and 765. The brightness value of a spot is calculated by the ratio of this total value to the maximum value (765) on the color scale.

$$Brightness = \frac{T}{3 \times 255} \tag{8}$$

Due to their very low contribution to the model, BR and brightness were excluded in order to reduce model complexity, and

only RR and GR were used as color-related variables.

2.2 Methods

The LiDAR system, as shown in Figure 6, sends laser beams to the surface with the help of a sensor and determines the distance between the surface and the sensor with great precision by measuring the arrival time of the returning beams. This technology is not only limited to measuring distances but also allows for creating a detailed three-dimensional model of the surface. LiDAR devices, often used with cameras, record each point's color information (red, R, green, G, and blue, B), allowing for more comprehensive and visually enriched data. LiDAR data allows the precise location of each point to be determined using a three-dimensional Cartesian coordinate system (X, Y, Z). In addition, the surface color information provides a detailed representation of visual features in the digital environment. These features make LiDAR widely used in various fields, such as terrain modeling, building design, infrastructure planning, environmental analysis, and archaeological research. The superior accuracy and versatility of LiDAR technology offer innovative solutions in engineering projects and urban planning, forest management, disaster risk assessment, and autonomous vehicle development. With its advantages and opportunities, this technology contributes to transforming modern engineering and design processes.

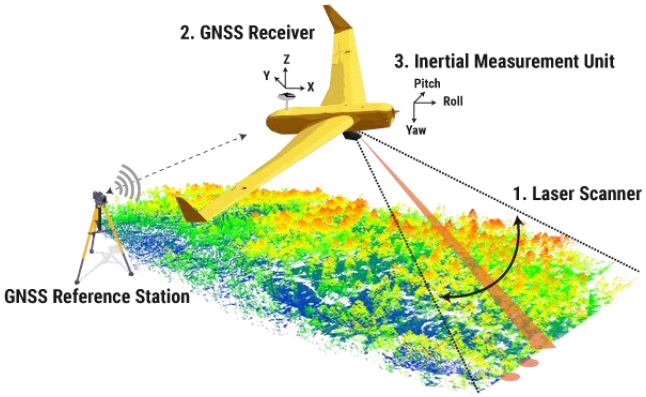

Figure 6: *LiDAR scanning (URL-2)*

LiDAR sensors send out laser beams that record the location of the first surface they hit. In addition to spatial coordinates, LiDAR systems record other properties such as intensity (reflectance strength), scan angle, and multiple return levels (e.g., first and last return). These features can significantly impact classification outcomes. However, the dataset used in this study contains only single returns and does not include intensity or scan angle attributes. Also, it needs to be organized and appropriately classified to use this point cloud data effectively in a specific area. For this purpose, traditionally used rule-based algorithms provide a fast solution, but their accuracy is considerably lower than that of machine learning algorithms. Several studies have demonstrated that machine learning methods outperform rule-based approaches, particularly in complex terrain and object classification tasks. According to Gharineiat et al. (2022), machine learning techniques provide better results than rule-based methods in LiDAR data classification. Machine learning algorithms are more successful in this field because they can evaluate various variables together on the training data while analyzing complex point cloud data. A DTM is a 3D model of the ground's surface, created using data from different measurement techniques. It is used in many areas, like engineering and graphic design. The main difference between a Digital Surface Model (DSM) and a DTM is that a DTM shows only the bare ground. This distinction is essential for this study, as the primary goal is to generate a DTM by classifying LiDAR point cloud data into ground and non-ground points. Correctly identifying and removing non-ground elements is fundamental for accurate terrain modeling. Figure 7 illustrates this difference. Surface objects like buildings, trees, and poles are not included in a DTM.

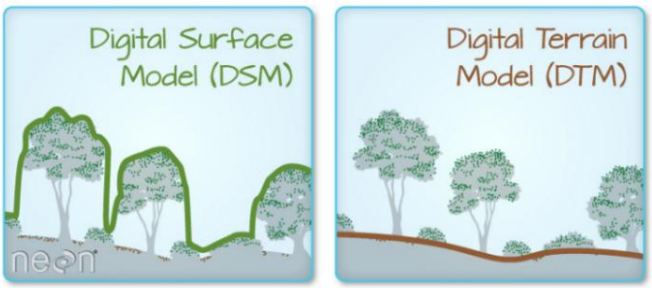

Figure 7: *Digital surface and terrain model (URL-3)*

2.2.1 Statistical Analysis

The classification task in this study is binary, aiming to distinguish ground points from all other objects. To define the surface geometry based on point cloud data, neighboring points for each LiDAR point were identified using Delaunay triangulation—one of the most widely adopted triangulation methods—constrained by a maximum neighbor distance of 3 meters. This step was essential for calculating point-specific geometric attributes, which require accurately determining adjacent points. Once the neighbors were identified, three geometric variables (Verticality, DeltaH, and 3rd Eigenvalue) were computed for each point using curvature angle and surface roughness formulas. In the same way, red, green, and blue ratios and brightness values were calculated using their specific color and brightness formulas.

The complete dataset was split into 70% training and 30% testing subsets, as shown in Table 1. After the split, all models were trained using the training subset, and predictions were made on the test subset. Confusion matrices were generated based on these predictions, and performance metrics such as accuracy, precision, recall, specificity, and F1 score were calculated accordingly. While a separate validation phase or k-fold cross-validation was not applied in this study, future research will consider incorporating such approaches to enhance model robustness and ensure unbiased performance estimation. Machine learning methods (CT, BCT, RFC, LR, KNN, SVM, NB, ANN, MLP, XGB) were trained using the training data. The model settings were adjusted as needed, and their performance was tested on the test data.

Table 1: *Train-test dataset distribution*

	Ground	Non-Ground	Total
Train (70%)	31642	48548	80190
Test (30%)	13561	20806	34367
Total	45203	69354	114557

The performance metrics were calculated with the following formulas:

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (9)$$

$$Precision = TP/(TP + FP) \quad (10)$$

$$Sensitivity\ (Recall) = TP/(TP + FN) \quad (11)$$

$$Specificity = TN/(TN + FP) \quad (12)$$

$$F1\ Score = 2 \times (Precision \times Recall)/(Precision + Recall) \quad (13)$$

Upon analyzing the results from models utilizing all seven variables, it was observed that the blue ratio and brightness consistently exhibited negligible importance scores across all algorithms. To reduce model complexity and enhance computational efficiency, the study used only the five most informative variables: Verticality, DeltaH, 3rd Eigenvalue, RR, and GR.

Machine learning algorithms offer various methods for data classification and prediction problems. DT decomposes the dataset based on features, while BCT and RF combine multiple trees to produce more stable and accurate results. LR is particularly effective in binary classifications by adopting a probabilistic approach. KNN classifies based on neighborhood relations, while SVM aims to find hyperplanes with the broadest margin separating classes. As a probabilistic model, NB is widely used in text analysis. ANN and MLP perform strongly on nonlinear problems. In contrast, XGB provides high accuracy with a fast and streamlined structure. Each algorithm offers different advantages and provides various solutions depending on the problem type. Decision trees classify data by recursively splitting it based on feature values, creating a tree-like structure for decision-making (Breiman et al., 1984). Bagging reduces variance by training multiple decision trees on bootstrapped subsets and aggregating their results (Breiman, 1996). Random Forest is an ensemble of decision trees where each tree is trained on a random subset of data and features, improving accuracy and controlling overfitting (Breiman, 2001). Logistic regression is a probabilistic model used for binary classification by modeling the relationship between input features and a binary target using the logistic function (Hosmer & Lemeshow, 2000). KNN is a non-parametric method that classifies a data point based on the majority class among its k closest neighbors (Cover & Hart, 1967). SVM finds the optimal hyperplane that maximizes the margin between classes in a high-dimensional space (Cortes & Vapnik, 1995). Naive Bayes is a probabilistic classifier based on Bayes' theorem assuming feature independence (McCallum & Nigam, 1998). ANN is a network of interconnected nodes (neurons) that learn complex patterns through backpropagation and weight adjustment (Schmidhuber, 2015). MLP is a class of feedforward neural network consisting of multiple layers of neurons with nonlinear activation functions (Bishop, 1995). XGBoost is a scalable, gradient-boosted decision tree algorithm designed for speed and performance, using regularization to prevent overfitting (Chen & Guestrin, 2016).

The algorithm's results were checked using an error matrix, and a detailed review was done using different statistics, such as specificity, precision, accuracy, sensitivity, and F1 score. In this process, the effects of each criterion on the classification success of the algorithms were analyzed and compared to see which algorithm gave the most effective results. Thus, it has been revealed in detail how successful each algorithm is compared to the others and which criteria affect the performance more.

Özen and Çilengiroğlu (2024) tested fewer machine learning models (CT, RFC, LR, KNN, SVM, NB, ANN, MLP) using only three features (Angle of Curvature, Roughness, and RR). The ANN model gave the best F1 score of 0.836.

The main differences between the two studies are that a more challenging data set was used, different highly correlated variables were included in the models, and the model richness was increased with two different machine learning algorithms (BCT and XGB).

3. Results

The LiDAR point cloud data was analyzed with different machine learning methods using the training data. Then, the models were tested on separate data to measure their performance. The outcomes of these evaluations are summarized in Table 2. Based on the results, while the differences in performance among the algorithms are not statistically significant, the highest

accuracy rate of 81.1% was achieved by both the XGB and SVM algorithms. This conclusion is supported by a Kruskal-Wallis H-test applied to the classification accuracies, which yielded a p-value above 0.05, indicating no statistically significant difference among the models. In contrast, the NB algorithm yielded the lowest accuracy at 70.8%. Notably, the NB model demonstrated a relatively higher sensitivity of 63.9% compared to the other models. The results of the SVM and XGB models show less than 1% difference. XGB has the highest F1 score with 84.1%, followed by RFC, SVM, and ANN with 84%.

Table 2: *Performance results of algorithms*

	Accuracy	Sensitivity	Specificity	Precision	F1 Score
CT	0.789	0.877	0.652	0.793	0.832
BCT	0.800	0.832	0.746	0.833	0.833
RFC	0.808	0.840	0.762	0.843	0.840
LR	0.753	0.782	0.705	0.801	0.792
K-NN	0.769	0.798	0.728	0.817	0.805
SVM	0.811	0.827	0.785	0.854	0.840
NB	0.708	0.639	0.818	0.842	0.724
ANN	0.809	0.842	0.754	0.839	0.840
MLP	0.807	0.781	0.848	0.886	0.837
XGB	0.811	0.836	0.777	0.851	0.841

In Figure 8, contour lines derived using linear interpolation from DTMs generated using ten different machine learning algorithms are visually compared with those of the reference (baseline) terrain model by overlaying them. In each visualization, black lines represent the reference terrain model, while red lines correspond to the terrain model produced by the respective algorithm for the given segment. These visualizations provide insight into how accurately each model captures topographic details. As illustrated in the sample figure above, the contour lines produced using XGBoost closely match those of the reference model, particularly in areas with significant elevation changes. However, minor deviations are observed in some contours, which can be attributed to terrain complexity and local point density variations.

In the XGB model, the LR model was used as the objective function; the error was used as the evaluation metric, and the error threshold was set as 0.01, which was obtained by performing 100 iterations in total. LR was chosen because it is a function that gives good results, especially in binary classifications. For each observation, the total score of the trees is calculated, and this score is converted into probability by passing it through a sigmoid function;

$$P(y = 1) = \frac{1}{1+e^{-z}} \tag{14}$$

Here, $P(y = 1)$ represents the probability that a given input belongs to class 1. The value z is the linear combination of input features, and the sigmoid function $\frac{1}{1+e^{-z}}$ maps this value into a probability range between 0 and 1, enabling binary classification. The threshold value is set as 0.5, and a positive result is obtained if $P \geq 0.5$, and a negative result is obtained if $P \leq 0.5$.

For the XGBoost model, training was conducted on the 70% training subset using the xgboost package in R. The model was trained with 100 boosting rounds (nrounds = 100), using the “binary:logistic” objective for binary classification and “error” as the evaluation metric. No manual hyperparameter tuning was applied; instead, default settings were used for all other parameters: maximum tree depth (max_depth = 6), learning rate (eta = 0.3), subsample ratio (subsample = 1), column

sampling ratio (colsample_bytree = 1), and L2 regularization term (lambda = 1). These defaults provided a baseline model configuration for performance comparison.

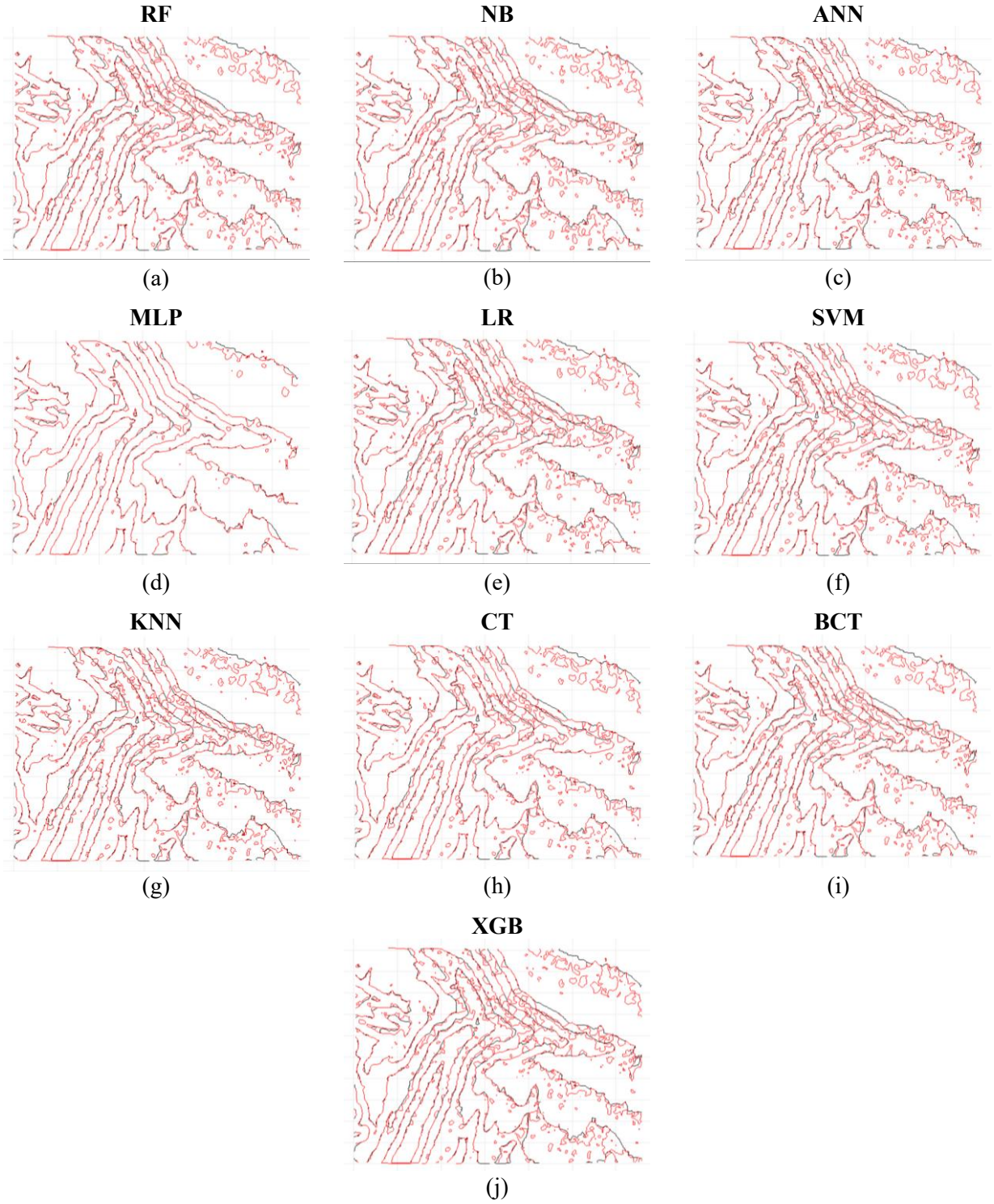

Figure 8: *Comparison of contours (a) RF, (b) NB, (c) ANN, (d) MLP, (e) LR, (f) SVM, (g) KNN, (h) CT, (i) BCT, (j) XGB*

Figure 9 presents the variable importance scores and the model graph associated with the final version of the algorithm. Figure 10 presents the simplified structure of the decision trees generated using the XGBoost algorithm with a logistic regression objective function, where the error threshold was set at 0.01. To prevent overfitting, the model was limited to 100 iterations. Examination of the decision structure reveals that the first split occurs on the DeltaH variable, confirming its role as the most informative feature. As shown in Figure 9, the variable contributions were determined as 40% for DeltaH, 19% for Rratio, 18% for Gratio, 13% for the 3rd Eigenvalue, and 10% for Verticality. While geometric variables accounted for

63% of the total contribution, color-based variables contributed 37%. This distribution indicates that the model primarily relies on geometric criteria in its decision-making process, with color ratios serving a complementary role.

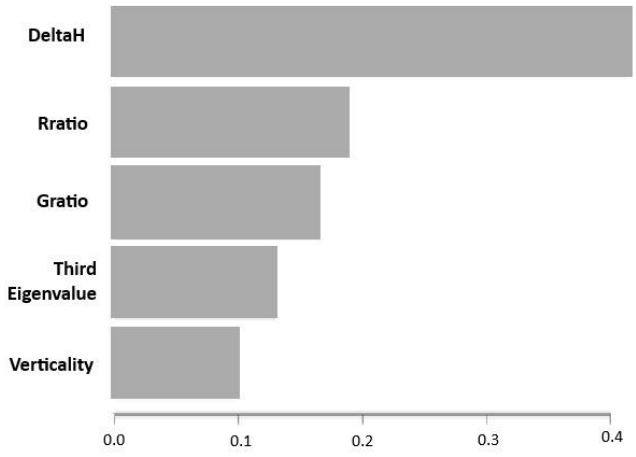

Figure 9: *Variable weight distribution chart*

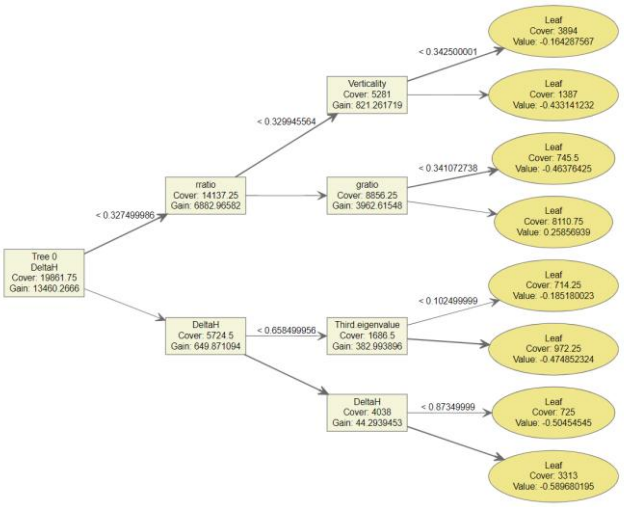

Figure 10: *XGB model diagram*

An examination of the confusion matrix (Table 3) reveals that the XGB model achieves higher performance in the "non-ground" class, whereas its accuracy in the "ground" class remains comparatively lower. This discrepancy can be primarily attributed to the tendency of the model to misclassify ground points as non-ground, particularly in areas characterized by complex topography and dense vegetation. The model's heavier reliance on geometric variables (DeltaH, Verticality, and the 3rd Eigenvalue) may further amplify this effect, as above-ground objects exhibit more distinctive geometric and color features, making the non-ground class easier to discriminate. Additionally, the unequal distribution of ground and non-ground points in the dataset (with non-ground points being more numerous) likely biases the model toward the majority class, contributing to the reduced prediction accuracy for the ground class. These findings indicate that, although the overall performance of the model aligns with comparable studies in the literature, improvements such as addressing class imbalance and enhancing the contribution of color-based variables may further increase its effectiveness in classifying ground points.

The model results show that geometric variables (DeltaH, RR, GR, 3rd Eigenvalue, and Verticality) dominate the prediction process, contributing 63%. The DeltaH variable has the highest contribution of 40% and is decisive in the model's accuracy. Its selection as the first split point in the decision tree confirms the high information gain of this variable. The total

contribution of color-based variables such as RR and GR is 37%, indicating that these variables play an auxiliary role. The analysis reveals that the model is more sensitive to geometric features and suggests that these features should be improved. Alternatively, methods to increase the contribution of color-based variables should be investigated to improve the model.

Table 3: *Confusion matrix of XGB model results*

	Predicted: Ground	**Predicted: Non-ground**
Actual: Ground	6992	7048
Actual: Non-ground	6621	13389

4. Discussion

In this study, the LiDAR point cloud was divided into ground and non-ground points using the XGB method, reaching an accuracy of 81.1%. This performance level is considered sufficient for professional applications, especially given the inherent measurement uncertainties in geospatial sciences. In literature, achieving 100% accuracy is widely acknowledged as unrealistic due to the complexities and variabilities in terrain modeling. Thus, the results presented here align with or exceed those of similar studies.

A distinguishing aspect of this research is the inclusion of the DeltaH geometric variable (representing the height differences among neighboring points) combined with color-based attributes. This variable contributed 40% to the model's predictive power, surpassing the combined contribution (37%) of red and blue color ratios. The deliberate exclusion of photogrammetric modeling steps also resulted in a more efficient workflow by reducing time consumption, which can be advantageous in time-sensitive projects.

However, it is essential to recognize that none of the original positional (X, Y, Z) or color (R, G, B) values were directly used as input features. Instead, all independent variables were derived through mathematical and geometric transformations. While this approach enhances the interpretability and abstraction of features, it also introduces variability in model outcomes depending on the choice of algorithm and variable engineering methods.

5. Conclusion

This study demonstrates that machine learning algorithms, particularly XGB, can effectively classify LiDAR point cloud data when derived geometric and color-based features are supported. The results suggest that, although traditional photogrammetric approaches were bypassed, the model still achieved competitive accuracy levels.

Nonetheless, due to the limitations imposed by the nature of geospatial measurements, perfect modeling remains unattainable. Novel geometric features based on shape distributions demonstrate reliable performance across multiple scales and outperform covariance-based features in all tested cases (Blomley et al., 2014). Future studies will aim to enhance model accuracy by introducing new variables, exploring alternative algorithms, and testing different datasets. Furthermore, the research will focus on filling the spatial gaps between ground points and predicting elevation (Z values) using machine learning approaches, offering a novel alternative to classical interpolation techniques in DTM production. It is anticipated that such efforts will lead to improved precision and broader applicability in geospatial analysis workflows.

Acknowledgements

This research has no funding. The authors would like to thank our colleagues for their insightful feedback and discussions, which greatly contributed to the development of this study. Lastly, we appreciate the continuous support of our families during this research process.

Author Contribution

Abdullah Can Ozen: Conceptualization, Design, Data collection and processing, Analysis, Software, Literature review. **Ozgul Vupa Cilengiroglu:** Supervision, Project administration, Support and final approval.

Declaration of Competing Interests

The authors declare that they have no known relevant competing financial or non-financial interests that could have appeared to influence the work reported in this paper.

References

Almohsen, A. S. (2024). Challenges facing the use of remote sensing technologies in the construction industry: A review. *Buildings, 14*(9), 2861.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Blomley, R., Weinmann, M., Leitloff, J., & Jutzi, B. (2014). Shape distribution features for point cloud analysis–A geometric histogram approach on multiple scales. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*, 9-16.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123-140.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth International Group.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). San Francisco, California, USA.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21-27.

Duran, Z., Ozcan, K., & Atik, M. E. (2021). Classification of photogrammetric and airborne LiDAR point clouds using machine learning algorithms. *Drones, 5*(4), 104.

Gharineiat, Z., Tarsha Kurdi, F., & Campbell, G. (2022). Review of automatic processing of topography and surface feature identification LiDAR data using machine learning techniques. *Remote Sensing, 14*(19), 4685.

Hosmer, D. W., & Lemeshow, S. (2000). *Applied logistic regression* (2nd ed.). New York: Wiley.

Jakovljevic, G., Govedarica, M., Alvarez-Taboada, F., & Pajic, V. (2019). Accuracy assessment of deep learning-based classification of LiDAR and UAV point clouds for DTM creation and flood risk mapping. *Geosciences, 9*(7), 323.

Joseph-Rivlin, M., Zvirin, A., & Kimmel, R. (2019, October). Moment: Flavor the moments in learning to classify shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (pp. 4085-4094). Seoul, South Korea.

Kang, Z., Yang, J., & Zhong, R. (2017). A Bayesian-network-based classification method integrating airborne LiDAR data with optical images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 10*(4), 1651-1661.

Kuçak, R. A. (2022, June). The analysis of 3D geometric features on point clouds by using open-source software. In *Proceedings of the 4th Intercontinental Geoinformation Days* (pp. 87-90). Tabriz, Iran.

Liu, X., Zhang, Z., Peterson, J., & Chandra, S. (2007, December). The effect of LiDAR data density on DEM accuracy. In *Proceedings of the 17th International Congress on Modelling and Simulation (MODSIM07)* (pp. 1363-1369). Christchurch, New Zealand: Modelling and Simulation Society of Australia and New Zealand.

Maturana, D., & Scherer, S. (2015, May). 3D convolutional neural network for landing zone detection from LiDAR. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3471-3478). Seattle, Washington, USA.

McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, No. 1, pp. 41-48). Madison, Wisconsin, USA.

Özdemir, E., Remondino, F., & Golkar, A. (2019). Aerial point cloud classification with deep learning and machine learning algorithms. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-4/W18*, 843-849.

Özen, A. C., & Çilengiroğlu, Ö. V. (2024, May). Performance comparison of classification methods in LiDAR point clouds. In *Proceedings of the International Applied Statistics Congress (UYIK)*. Tokat, Türkiye.

Park, Y., & Guldmann, J.-M. (2019). Creating 3D city models with building footprints and LiDAR point cloud classification: A machine learning approach. Computers, *Environment and Urban Systems, 75*, 76-89.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks, 61*, 85-117.

Tan, Y., Liu, X., Shuaishuai, J., Wang, Q., Wang, D., & Xie, X. (2023). A terrestrial laser scanning-based method for indoor geometric quality measurement. *Remote Sensing, 16*(1), 59.

Teruggi, S., Grilli, E., Russo, M., Fassi, F., & Remondino, F. (2020). A hierarchical machine learning approach for multi-level and multi-resolution 3D point cloud classification. *Remote Sensing, 12*(16), 2598.

Wu, L., Zhu, X., Lawes, R., Dunkerley, D., & Zhang, H. (2019). Comparison of machine learning algorithms for classification of LiDAR points for characterization of canola canopy structure. *International Journal of Remote Sensing, 40*(15), 5973-5991.

URL-1: Kılıç, İ. Principal component analysis (PCA): A practical guide. Medium. https://medium.com/@ilyurek/principal-component-analysis-pca-a-practical-guide-58dea99dd93 (Accessed: 18 January 2025).

URL-2: Yellowscan. How does LiDAR work? Yellowscan Knowledge Base. https://www.yellowscan.com/knowledge/how-does-LiDAR-work (Accessed: 18 January 2025).

URL-3: Wasser, L. A. What is a CHM, DSM and DTM? About gridded, raster LiDAR data. NEON Science. https://www.neonscience.org/resources/learning-hub/tutorials/chm-dsm-dtm (Accessed: 18 January 2025).