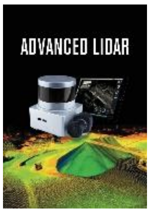
Advanced LiDAR

http://publish.mersin.edu.tr/index.php/lidar/index

e-ISSN 2791-8572



Evaluation of Machine Learning Classifiers for 3D Mobile LiDAR Point Cloud Classification Using Different Neighborhood Search Methods

Mahmoud Mohamed[*1], **Salem Morsy**[2], **Adel El-Shazly**[2]

[1]*Fayoum University, Faculty of Engineering, Civil Engineering Department, Eqypt*
[2]*Cairo University, Faculty of Engineering, Public Works Department, Eqypt*

Keywords
Mobile LiDAR,
Road Features,
Machine Learning,
Neighborhood Selection.

ABSTRACT

Mobile LiDAR systems are distinguished with large and highly accurate point clouds data acquisition for road environments. Road features extraction is becoming one of the most important applications of LiDAR point cloud, and is used largely in road maintenance and autonomous driving vehicles. The main step in Mobile LiDAR processing is point classification This classification relies on the geometric definition of the points and their surroundings, as well as the classification methods used. The neighbors of each point is helpful to find more meaningful information than the raw coordinates. On the other hand, machine learning algorithms have proved their efficiency in LiDAR point cloud classification. This research compares results of using three machine learning classifiers, namely Random Forest, Gaussian Naïve Bayes, and Quadratic Discriminate Analysis along with using three neighborhood search methods, namely k nearest neighbors, spherical and cylindrical. A part of the pre-labelled benchmark dataset (Paris Lille 3D) with about 98 million points was tested. Results showed that the using Random Forest classifer with the cylindirical neighborhood search method acheived the highest overall accuracy of 92.39%.

1. INTRODUCTION

Laser scanning systems are widely used as remote sensing techniques based on Light Detection and Ranging (LiDAR). These systems have been involved into surveying market, providing high accuracy measurements as well as efficient data collection, especially in the 3-Dimensional environments. LiDAR technology is non-contact active measuring to get information of the scanned 3D surfaces with less dependence on illuminations. LiDAR scanning systems have also the ability to record point cloud actively and precisely at a high speed in real time (Pu & Vosselman, 2009). The rapid acquisition of high 3D information of objects has been more realized due to the most recent advances in the technology (Yu et al., 2014).

3D point clouds could be obtained in precise format using one of main laser scanning systems types; Airborne, Terrestrial or Mobile (Hyyppä et al., 2013). In the past decades, the market has a high demand for utilizing data acquired from Mobile Laser Scanning (MLS) systems. The applicability of MLS from moving platforms allows for the complete coverage of complex urban environments. 3D point data acquired from MLS systems are distinguished with their high accuracy level and points density, with an average of about 1000-2000 pts./m^2. The bottleneck in any work is the transmission from field data acquisition to the processing step with a large amount of data that sometimes represent hindrances and thus need to be managed effectively. As MLS systems often provide high dense point clouds, their processing will be labor intensive (Guarnieria et al., 2009), and may last for days to handle those data that were collected in a very short time.

The MLS system consists mainly of a laser scanning sensor, global navigation satellite system (GNSS) and inertial navigation system (INS) unit. Laser scanning sensor is responsible for the emission of laser beams and reception the reflected rays. Because the laser beams have a constant speed (i.e., speed of light, S) and with measuring the time elapsed, t, from the emission and reception of the beams using a precise interval timer, the distance between the sensor and the object can be measured according to the first Newton's rule Equation (1). It should be noted that the distance, D, calculated using Newton's rule is twice the distance between the system and any measured object.

$$D = S*t \tag{1}$$

*Corresponding Author

*(maa58@fayoum.edu.eg) ORCID ID 0000-0002-8690-3192
(morsy@eng.cu.edu.eg) ORCID ID 0000-0002-1683-2050
(adel.shazly@eng.cu.edu.eg) ORCID ID 0000-0003-4332-9386

Cite this;

Mohamed M., Morsy, S. & Shazly, A. (2022). Evaluation of machine learning classifiers for 3D mobile LiDAR point cloud classification using different neighborhood search methods. Advanced LiDAR, 2(1), 01-09.

Received: 20/04/2022; Accepted: 20/06/2022

GNSS is another main component in MLS system. It is used to determine the accurate position of the system instantly. This would help to geo-reference any measured object to the global system supported by the GNSS unit. INS has a role to determine the system values of roll, pitch and heading, which helps to determine the relative position between any scanned object and the system at the measurement time.

There are additional components that could be mounted on the system such as digital cameras and distance measurement instruments (DMIs). Digital cameras are optional choice, and their videos/images are not included in the process of LiDAR data except for some methods of features extraction or 3D reconstruction that integrates the LiDAR point clouds with imagery. DMIs are used to continuously measure the distance passed by the vehicle. This helps in case of the integration with INS to determine the position of the whole system in case of the disability of the GNSS unit due to instant interruptions.

The development in MLS systems is scaled with two main topics: how much the accuracy of collected data is increased, and the availability of developing software packages that make the processing of the point cloud easier, faster and more precise. The former is improved through the MLS system itself and its internal components, either hardware or software. The latter is divided into three main steps and they are improved individually. The three steps are removal of the outliers within the dataset, detection of the required features and the modeling of the extracted features to produce CAD models. The features' detection step is still under continuous development for the purpose of evaluating its level of automation and identifying different features simultaneously.

Automatic 3D point clouds processing is an important topic in most cases related to remote sensing, photogrammetry and computer vision because of the time consumed and cost of user-assisted analysis. Current researches aim to decrease the human involvement in the point clouds processing. In the past years, machine learning classifiers have had a great contribution to 3D point clouds processing, and covered all types of LiDAR systems; airborne, terrestrial, or mobile (Chehata et al., 2009; Mohamed et al., 2021a; Nguyen et al., 2020).

Machine learning (ML) is a subfield related to computer science that is mainly concerned with constructing useful algorithms which rely on a collection of given examples of some phenomenon. ML can also be defined as the process of solving a practical problem. This is conducted through gathering required dataset, algorithmically build a statistical model based on that dataset, and the statistical model is somehow expected to solve the practical problem (Burkov, 2019).

Supervised machine learning is one of ML algorithms that uses the dataset to produce a model which takes a feature vector x as input and output information and allows deducing the label for this feature vector (Burkov, 2019). There are various supervised learning algorithms which differ according to their mathematical definition such as k Nearest Neighbor, Logistic Regression, Naïve Bayes, Discriminant Analysis, Decision Tree, Random Forest, Support Vector Machine, and Neural Network. Figure 1 shows an example of a supervised learning algorithm whereas it predicts a yellow edge boundary between two classes (Red and Blue classes) according to inputs attributes of X and Y.

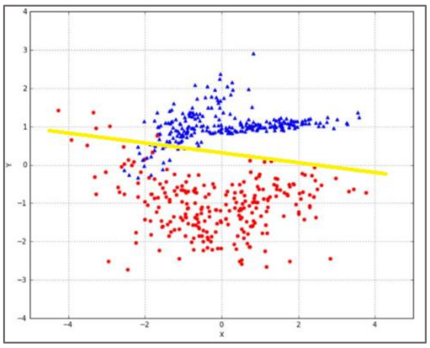


Figure 1. An example of a supervised learning algorithm (Bonaccorso, 2017)

1.1. Mobile LiDAR Data Classification Using Machine Learning

ML classifiers require input data that are distinguishable to categorize each class. However, MLS point cloud in its raw format consisting of 3D coordinates (i.e., X, Y and Z) and sometimes intensity values are not sufficient for ML to differentiate between different classes. Coordinates are meaningless for ML classifiers except for Z coordinate which may somehow be used to extract some classes such as ground (i.e., the lowest points within any point cloud dataset). In addition, intensity values may differ for points of same class according to weather conditions or how far the point from the sensor is.

Due to the ability of ML classifiers to distinguish between multiple classes, most researches that applied ML classifiers aimed at multi-classification (Hackel et al., 2016; Mohamed et al., 2021b; Weinmann et al., 2013), but there were also some researches that focused on one class such as rail track detection in (Elberink et al., 2013) out of all other classes. Generally, the classification process is divided into three steps; neighborhood search method, features extraction and ML classifier.

Neighborhood search method is defined as the predetermined scale around each point. The neighborhood may take various types according to the geometric definition such as K nearest neighbors (KNN), spherical and cylindrical neighborhoods as illustrated in Figure 2. KNN method is defined as the most nearest k number of points to the point of interest x according to the Euclidean distance (Linsen & Prautzsch, 2001). The determination of how much k neighbors has been studied in (Hackel et al., 2016; Weinmann et al., 2013) where a fixed number of points for all points was applied. Others applied a different and changing k number for each individual point according to a specific condition (Demantké et al., 2011; Weinmann et al., 2014; Weinmann, Jutzi, et al., 2015; Weinmann, Urban, et al., 2015).

Spherical neighborhood method of point x is defined by a sphere with a radius r and centered with the point of interest x (Lee & Schenk, 2002). Cylindrical neighborhood is determined by the cylinder centered with the point of interest x, and its neighbors are all points within that cylinder. The cylinder is defined with a 2D radius r and its height may be a specific value above and below the point of interest, or may be infinite (Filin & Pfeifer, 2005). This neighborhood step allows getting most of common used features in the classification process, from Eigenvalues and Eigenvectors derived from their covariance matrix, or from heights of the neighborhood points.

Spherical and cylindrical neighborhoods have not been widely used in 3D point neighborhood search. (Li et al., 2021) used a spherical neighborhood method around the points within 0.5 m radius. (Weinmann et al., 2017) used the spherical and cylindrical neighborhood in a comparison with KNN in its two ways; fixed and optimal k value. They used four individual neighborhoods, cylindrical with 1.0 m radius, spherical with 1.0 m radius, and two KNN with k = 50 and optimal k of each point according to Eigen entropy defined in (Weinmann, Jutzi, et al., 2015). Other approaches proposed a multi-scale neighborhood with different features extracted from different neighborhoods at the same time. (Hackel et al., 2016) proposed a multi-scale neighborhood using (k = 10) of KNN. (Blomley et al., 2016) used a cylindrical multi-scale neighborhood of radius (1m, 2m, 3m, and 5m) as well as KNN with the optimal value of k according to the Eigen entropy defined by (Weinmann et al., 2014) with the normalized Eigenvalues. (Zheng et al., 2017) used a cylindrical neighborhood with radius r_C = 0.25 m, (Zheng et al., 2018) used other values (0.45, 0.6, 0.75, 0.9, 1.05) m, but without any significant effect on the results.

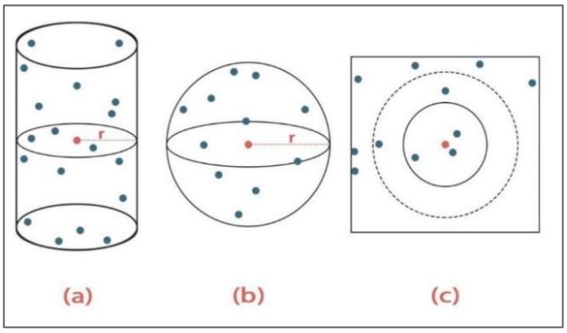


Figure 2. A definition of neighborhood is presented where (a) the cylindrical neighborhood, (b) the spherical neighborhood, and (c) the k nearest neighborhood. Also, the point in red is the point of interest, and r is the predefined radius for the neighborhood.

The second step in the classification process of XYZ point cloud is to extract much more meaningful information from those data than the XYZ coordinates. One of the most used set of features are covariance features, those covariance features are derived from the covariance matrix of each point's neighborhood (Pauly et al., 2003; West et al., 2004). Those features are computed using the Eigenvalues $(\lambda_1, \lambda_2, and\ \lambda_3)$ of the covariance matrix. According to (Weinmann et al., 2013), for a linear (1D) structure; λ_1 is the largest between the three Eigenvalues. For a (2D) planar structure, $(\lambda_1\ and\ \lambda_2)$ are much larger than λ_3, while a (3D) volumetric structure has similar Eigenvalues (Dittrich et al., 2017). In (Weinmann et al., 2014; Weinmann, Jutzi, et al., 2015; Weinmann, Urban, et al., 2015), they replaced the Eigenvalues with their normalized values $(e_1, e_2, and\ e_3)$ where $(e_i = \lambda_i / \sum_{i=1} \lambda_i)$.

Another common set of features is the moment features which implemented previously is (Hackel et al., 2016), those features were derived from the dot product of the coordinates' array and the Eigenvectors of the covariance matrix. (Demantké et al., 2011; Hackel et al., 2016) had also added another feature which may be considered as a covariance feature, namely verticality. This feature was derived from the vertical component of the normal vector. The Eigenvalues were used in a previous work of (Chehata et al., 2009) and (Wang et al., 2020), where the Eigenvalues were added as features in addition to waveform features.

The last set of features is the height features. These features are derived from the Z-coordinate of the points within the local neighborhood of each point. (Weinmann et al., 2013) used the neighborhood of each point to calculate the standard deviation of the points' heights as well as the maximum difference of the heights. (Hackel et al., 2016) also used the heights of the points to calculate the maximum height difference, the maximum height below the point of interest, and the maximum height above the point of interest. Another height feature is the height above ground, but was used in Airborne LiDAR Scanning classification. The definition of the ground was set to the lowest point within a cylindrical neighborhood according to (Chehata et al., 2009) and (Mallet et al., 2011). They used a cylindrical neighborhood around the point of interest with a 15 m and 20 m radius, respectively.

1.2. Research Objectives

This article aims to investigate the effectiveness of machine leaning classifiers for the sake of road features extraction, also the importance of choosing appropriate neighborhood method and its direct impact on the classification results. The two main objectives of this research are 1) evaluate the effectiveness of using three neighborhood selection methods for MLS data and 2) evaluate the application of three machine learning algorithms for MLS data classification.

2. METHOD

The methodology of this research is divided into four stages as shown in Figure 3. First, the pre-processing stage which contains data subsampling and data slicing. Second, the neighborhood search method to find the neighbors of each point, it includes three alternatives (section 2.2). Third, geometric features extraction that will replace the XYZ coordinates as input to ML classifiers. Finally, three ML classifiers are applied to learn and classify the dataset (section 2.4).

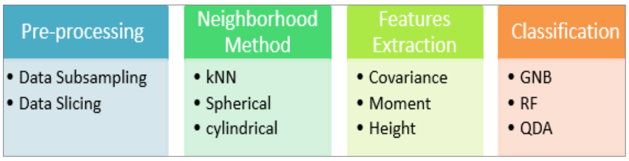


Figure 3. 3D point clouds classification workflow

2.1. Pre-processing Stage

2.1.1. Data Subsampling

Due to the high points' density of MLS systems, data processing means time. Hence, different researches suggested various scenarios for the reduction of the dataset. For instance, (Zheng et al., 2017, 2018) removed the ground points as their research's aim was to classify non-ground points. Another research, (Weinmann et al., 2014), removed any class with points' count less than 0.05% of the whole dataset. However, this could result in losing significant information of removed points. On the other hand, the huge amount of MLS point cloud may be more than the amount of information required to detect the urban road objects. Thus, removal of some points in a specific manner would improve the classification processing time, which is a major evaluation factor of any method. This is applicable if the reduction in the dataset does not harm the information and the classification results are acceptable compared to results of the whole dataset (Mohamed et al., 2021a).

Reducing the dataset may be through variant manners according to the organization of the dataset and the differences in the point density. As much as the dataset is organized and equally distributed, the point reduction may be more applicable by a high percentage. We used the dame subsampling method implemented in (Mohamed et al., 2021a), the subsampling was by the minimum spacing between points, and the reduction in the dataset was by about the half but without low reduction in the results, whereas the overall accuracy was 92.39% and 90.26% for the full and subsampled datasets.

2.1.2. Data Slicing

Figure 4 shows the effect of slicing on finding the neighbors of any point within the black points on the left side of the dataset. It has no meaning to search for neighbors within the whole dataset including the white points as there is no way to have neighbors from the white points. Therefore, if the search of the neighborhood of any point is within a small slice of the dataset, the processing time will be more efficient. Slicing of the dataset could be by distance or equal number of points along the dataset. In this research, we divided the dataset into slices with same amount of points along the road and added two overlaps before and after each slice to best calculate the neighbors of edge points.

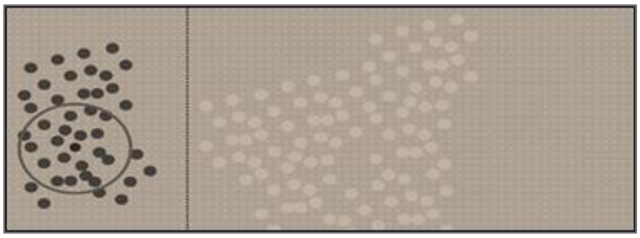


Figure 4. Data slicing for neighborhood search

The disadvantage in the slicing concept is that points on the edge will find their neighbors from one side only and this may affect the results. The more slices we have, the more edges we get, and hence the effect will be increased. In order to avoid this effect, each slice will be extended with an overlap from each edge to find the neighbors of the edge points effectively.

2.2. Neighborhood Search Stage

Neighborhood method is used to find the neighbors of any point and derive extract features from its surroundings. As aforementioned, there are three common types of neighborhood; KNN which is based on the Euclidian distance, and spherical and cylindrical neighborhoods which are based on a predefined radius. The choice between those three types depends on the uniformity and points' density in the used dataset. Searching by a fixed radius (i.e., spherical and cylindrical neighborhoods) in dataset with uniformly point density is usually suitable as it preserves the objects in a fixed geometry scale. However, the strongly varying point density requires a fixed number of points; hence, the KNN is an effective choice (Hackel et al., 2016).

K nearest neighborhood is determined by the nearest (k) points to the point of interest. The change in the number of k points by increasing or decreasing has a high linearly proportional relation with the processing time. In addition, it affects the classification results. More k points enhance the results but with more processing time. Spherical and cylindrical are determined by a radius r. A spherical neighborhood definition is the sphere with radius in 3D (r_{3D}) around the query point while cylindrical neighborhood is implemented in the 2D projection of points neglecting the height of points when searching for neighbors. The cylinder contains all points around the query point within 2D radius (r_{2D}) above and below the point.

2.3. Features Extraction Stage

For each point, we replace its coordinates with three sets of features; covariance, moment and height. Those features have had the most occurrence in previous researches. Figure 5 shows the structure of features extraction.

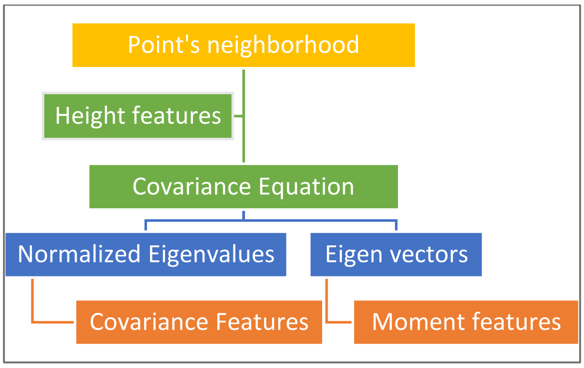


Figure 5. workflow of points' features extraction

A total of fifteen features are derived and used in this research as listed in Table 1. From the neighborhood of each point, the covariance matrix is constructed using the three coordinates' arrays as shown in Equation (2).

$$\mathrm{C} = \frac{1}{n} \sum_{i \in N} (p_i - \bar{p})(p_i - \bar{p})^T \tag{2}$$

Where n is the number of points within the neighborhood of each point x, $p_i$is representing each point in the neighborhood, $\bar{p}$ is the centroid of n points in the neighborhood.

The result of the covariance matrix is three Eigenvalues $(\lambda_1, \lambda_2, and\ \lambda_3)$ and three Eigenvectors$(v_1, v_2, and\ v_3)$. The first subset of features, covariance features, are derived from the normalized Eigenvalues $(\mathrm{e}_1, \mathrm{e}_2, and\ \mathrm{e}_3)$ where $\mathrm{e}_i = \lambda_\mathrm{i}/(\lambda_1 + \lambda_2 + \lambda_3)$. The covariance features are similar to what have been previously used in the research of (Weinmann, Urban, et al., 2015), except for the "Sum" feature that is derived from the summation of the three Eigenvalues, not the normalized ones. Another feature is added to the covariance set is the verticality which has been driven before in the research of (Demantké et al., 2011).

The second set of features (moment features) were first used in (Hackel et al., 2016). Those features are driven from the dot product of coordinates' arrays and the first two Eigenvectors. Those moment features are helpful in identifying the crease edges as well as occlusion boundaries. Height features are also used in this research. Those features include Δz: the max difference in height between all points within the neighborhood and σz: the standard deviation of z - coordinate of points.

Table 1. Geometric features

Items	Feature	Formula
Covariance features	L_λ: Linearity	$(\mathrm{e}_1 - \mathrm{e}_2)/\mathrm{e}_1$
	P_λ: Planarity	$(\mathrm{e}_2 - \mathrm{e}_3)/\mathrm{e}_1$
	S_λ: Scattering	$\mathrm{e}_3/\mathrm{e}_1$
	O_λ:Omni variance	$\sqrt[3]{\mathrm{e}_1\mathrm{e}_2\mathrm{e}_3}$
	A_λ: Anisotropy	$(\mathrm{e}_1 - \mathrm{e}_3)/\mathrm{e}_1$
	E_λ: Eigen entropy	$-\sum_{\mathrm{i}=1}^{3} \mathrm{e}_\mathrm{i}\ln(\mathrm{e}_\mathrm{i})$
	C_λ:Change of curvature	e_3
	$\sum_\lambda$: Sum	$\lambda_1 + \lambda_2 + \lambda_3$
	V: Verticality	$1 - \langle(0,0,1), v_3\rangle$
Moment features	1st order, 1st axis (f11)	$\sum_{\mathrm{i} \in \mathrm{p}} \langle(\mathrm{p}_\mathrm{i} - \mathrm{p}), v_1\rangle$
	1st order, 2nd axis (f22)	$\sum_{\mathrm{i} \in \mathrm{p}} \langle(\mathrm{p}_\mathrm{i} - \mathrm{p}), v_2\rangle$
	2nd order, 1st axis (s11)	$\sum_{\mathrm{i} \in \mathrm{p}} \langle(\mathrm{p}_\mathrm{i} - \mathrm{p}), v_1\rangle^2$
	2nd order, 2nd axis (s22)	$\sum_{\mathrm{i} \in \mathrm{p}} \langle(\mathrm{p}_\mathrm{i} - \mathrm{p}), v_2\rangle^2$
Height Features	Δz	$\mathrm{Z}_{\mathrm{max}} - \mathrm{Z}_{\mathrm{min}}$
	σz	Standard deviation of z coordinate within the neighborhood

2.4. Machine Learning Classification Stage

Machine Learning has its ability to differentiate between different classes without any preprogramming. The results of any classification vary with respect to different classifiers and their suitability with the used dataset and given features. Three ML classifiers are evaluated in this research, including Random forest (RF), Gaussian Naïve Bayes (GNB), and Quadratic Discriminate Analysis (QDA).

2.4.1. Random Forest

RF classifier is an ensemble algorithm containing multiple tree decisions (Breiman, 2001). It combines multiple weak learners for the sake of a stronger one (Weinmann, Urban, et al., 2015). For each decision tree, the classifier makes nested relations between the input features and the output class according to specific conditions in the inputs. The more estimators (decision tress) are, the better results will be but with an increase in processing time. The optimization of best fitted RF model depends on various parameters of the algorithm that should be well tuned. In addition to the number of trees (estimators), there are other important parameters such as 'max_depth' and 'min_samples_split' parameters, both determine how far each tree will go down.

The mechanism of RF is simplified as following. For each decision tree; a sample of points represented in its geometric features are trained into that tree to find the different relations between points' features and corresponding outputs. This process is repeated for other decision trees used. The process of classifying any unknown point is to implement that point in each decision tree to reach a class of that tree. A voting step is then applied between the results of all trees to classify that point to the most occurrence class among the trees.

2.4.2. Gaussian Naïve Bayes

Naïve Bayes classifiers are simple probabilistic classifiers. They are based on the Bayes' theorem but with strong independence assumptions inside the features. GNB is a variant of Naïve Bayes that follows Gaussian normal distribution and supports continuous data. (Bishop, 2006). The main steps of GNB are calculating the probability of each class which equals to the portion of class' points in the whole dataset, and constructing a Gaussian distribution of each class for each feature. To classify any unknown point, a value is calculated of the point for each class according to Equation (3), and the point is classified to the class of highest value.

$$\log P(class_i) + \sum_{j \in F} \log L\left(feature_j | class_i\right) \tag{3}$$

Where $P(class_i)$ is the probability of each class, and $L\left(feature_j | class_i\right)$ is the likelihood of each $feature_j$ of $class_i$, and F is the number of features.

2.4.3. Quadratic Discriminate Analysis

A quadratic classifier is a statistical classifier that uses a quadratic decision surface to separate measurements of two or more classes of objects or events (Bishop, 2006). The algorithm represents the points according to their attributes (features) on multi-dimensional graph. Then, it builds a quadratic boundary between classes that enclose each class entire a single boundary. To classify any point, it is located on the graph according to its features and is classified to the class boundary it is inside.

3. Study Area and Dataset

To evaluate the proposed methodology, we used a pre-labelled benchmark dataset (Paris Lille 3D). It is a part of NPM3D Benchmark Suite research project (Roynard et al., 2018). It was acquired using a MLS system of robotics center of Mines Paris Tech (L3D2). The dataset consists of two parts; a longitudinal section of about 1500 m length with about 98 million points in Lille and another part in Paris with 450 length and about 45 million points.

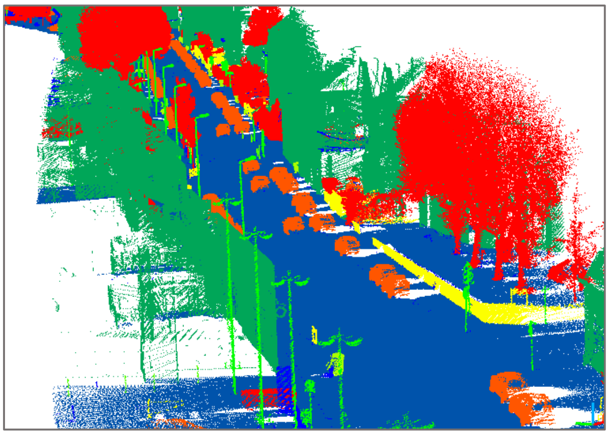


Figure 6. Paris-Lille-3D dataset (Lille Part)

The dataset contains mainly nine coarse classes in addition to some unclassified points, the classes are ground, building, pole, bollard, trashcan, barrier, pedestrian, car and vegetation. Figure 7 shows the data portion of different classes, some classes (both ground and building represent about 90% of the dataset)

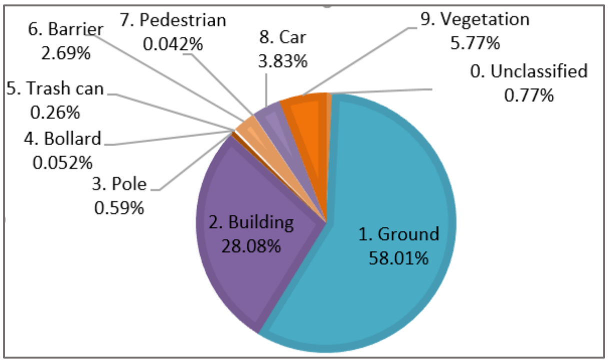


Figure 7. Dataset portion of classes

4. RESULTS

This research aimed at the comparison between three ML classifiers with three different neighborhood methods. According to (Zheng et al., 2017, 2018) ,the change and the increase in the radius of the cylinder did not have a remarkable effect, hence we used the radius of the cylinder to equal 0.20m, and similarly we choose the sphere radius to be 0.20m. For the kNN method, we chose k = 10, as in the research of (Weinmann et al., 2015), the most occurrence value as an optimal value. For each point in the dataset, its neighbors were defined and the geometric features, were derived as presented in Table 1.

The dataset was divided into two equal parts after replacing the coordinates of each point with their corresponding extracted geometric features. The two parts are training/validation part as well as the testing part. The first part was divided into four partitions, from which four ML models were created to find out the best fitting model according to the overall accuracy, after that our method could be evaluated with the remaining testing part. This procedure was implemented in all our classification scenarios whether the change in the classifier or the neighbourhood search method.

Overall accuracy was used as a primary score for the models as shown in Figure 8. In the three neighborhood methods, cylindrical neighborhood was the best between the three neighborhood methods, regardless the ML classifier used. The overall accuracy of the cylindrical method for the RF, GNB and QDA was 92.39%, 78.47% and 78.18%, respectively. On the other hand, a great difference does exist between RF and other classifiers which makes RF is the most suitable classifier for the tested dataset.

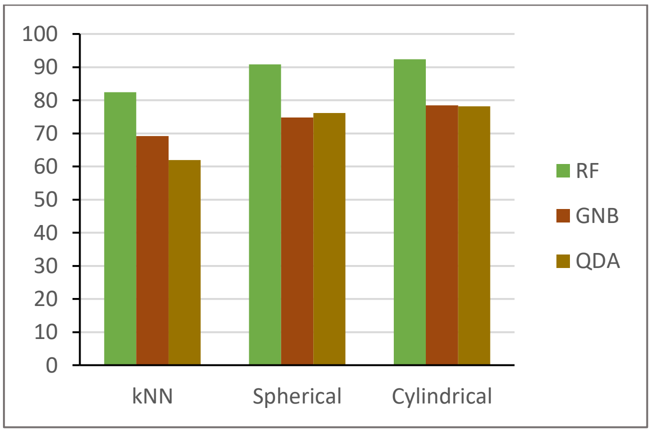
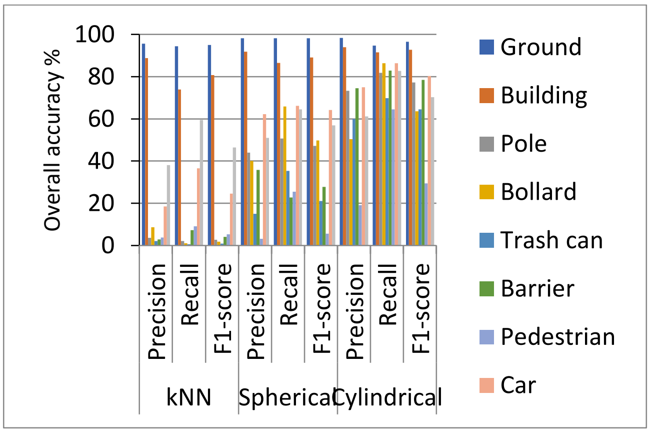


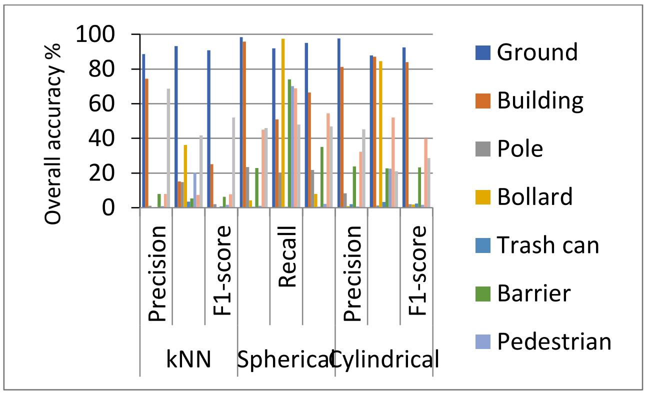
Figure 8. Overall accuracy percent of RF, GNB, and QDA for KNN, spherical, and cylindrical neighborhood.

Other scores, including precision, recall and f1-score were used to individually evaluate the results of each class. Mobile LiDAR point clouds data of road environments are usually imbalanced, and two or three classes may consist more than 90% of the whole dataset. Consequently, any ML model may be confused and tend to classify most of the points to the major class(s). As shown in Figure 7, the ground and building classes contain about 90% of the dataset, whereas the ML classifier could classify the whole points to one of these two classes, and hence the overall accuracy could reach up to 90% but with misleading classification. Therefore, the role of using precision, recall and f1-score is to evaluate the classifiers with datasets that have low percentages of classes.

Figure 9 shows the calculated precision, recall and f1 score of each class using to different classifiers and neighborhood search methods. The ground and building classes achieved high results in the three classifiers. This comes from the large amount of points in the dataset of both classes. Therefore, any classifier will easily detect classes with large portions such as ground and building, and hence they could not be a measure for a good classifier. For instance, the precision of ground class was 98.11%, 96.66% and 97.68%, while the recall was 91.35%, 88.1% and 87.95% and f1-score was 94.61%, 92.18 and 92.56% for RF, GNB and QDA, respectively. For building class, the precision was 95.60%, 76.27% and 81.21%, while the recall was 92.65%, 90.25% and 87.14% and f1-score was 94.1%, 82.67 and 84.07% for RF, GNB and QDA, respectively.

On the other hand, the other classes are varying in their classification results, between the three classifiers; the RF revealed the highest scores between the three classifiers for detecting those low portion classes. Generally, the results showed that RF is much more effective than other classifiers. RF is also suitable to classes with less number of points such as poles, barriers, and trashcans. Therefore, classes with variant geometric characteristics require large scale of features to best distinguish between them. However, not all classifiers are able to handle all classes with a huge number of samples as well as many features. GNB and QDA classifiers were not able to achieve high scores for all classes. Only ground and building classes revealed close results for different classifiers. Ground and building were clearly distinguished due to their geometric shapes (i.e., 2D planes) which were determined using Verticality feature. Thus, this is helpful for any classifier to best find most of points that belong to those classes.





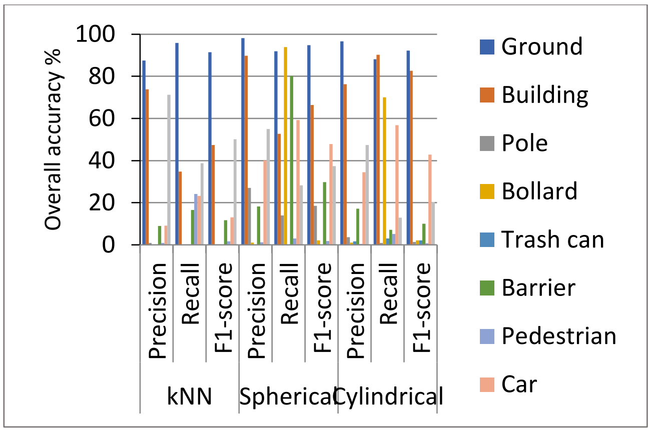


Figure 9. Classification results of RF, GNB, and QDA.

The classes could represent an obstacle and mislead the classification process. It may be a class which misleads the algorithm for the purpose of finding other classes. For example in Figure 10, the white points were wrongly classified due to the similar geometric properties with other classes.

Bottom base of vegetation (class 9 in our dataset) is an example, as nearly all bottom bases were classified wrongly. The similarity between points in different classes makes it not trivial task to classify those points. Literality, points on tree base are very similar to ground and curb points. They all have the same orientation and nearly the same height but different classes, and that make it difficult for any classifier.

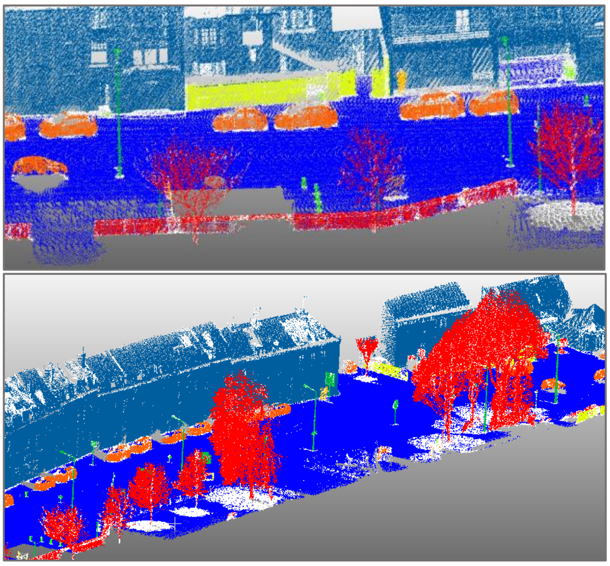


Figure 10. Example of wrongly classified points

5. CONCLUSION

Nowadays, the growth in the utilization of mobile LiDAR scanning systems is very rapid in many road corridors' applications. They have an advantage of there is no required direct contact with any road features in the data acquisition. That makes the MLS systems preferable for applications such as infrastructure and surveying assets. They capture huge amount of point clouds which describe road scenes with high details. The detection of different road features such as light poles, curb, and road signs from MLS point cloud is important to be automatic. The extracted information could be used in various applications such as quantity and volume surveys, right-of-way asset inventory and GIS applications.

In this research, three neighborhood search methods were studied and compared, namely KNN, spherical and cylindrical neighborhoods. Their results were varying and have revealed that cylindrical neighborhood method was the most suitable for the tested dataset. Although the overall accuracy of both the spherical and cylindrical was close using different classifiers with imbalanced dataset, the overall accuracy could not be a final judgment. However, and according to precision, recall and f1-score, the cylindrical neighborhood method was much more effective than the spherical one.

ML classifiers are various and differ in their mathematical models. According to the case study and type of dataset, the appropriate classifier could be selected. Three ML were applied for road features' classification from mobile LIDAR dataset to evaluate which classifier is more suitable. Three models were trained on the dataset using the three classifiers and their overall accuracy were 92.39%, 78.47% and 78.18% for RF, GNB and QDA, respectively with cylindrical neighborhood method. From the overall accuracy and the detailed scores (precision, recall and f1-score), the RF was the most suitable for the classification process. In addition, RF achieved high scores for each class including classes with low portion while those classes may be considered non-existent for the other two classifiers due to their very low scores.

Acknowledgement

The authors would like to thank the editor and the anonymous reviewers for their valuable comments and suggestions, which helped to improve this work.

Author contributions

Mahmoud Mohamed: Methodology, Software, Data curation, Validation, Visualization, Writing-Original draft preparation. **Salem Morsy:** Conceptualization, Methodology, Visualization, Writing-Reviewing and Editing. **Adel El-Shazly:** Conceptualization, Writing-Reviewing and Editing.

Conflicts of interest

There is no conflict of interest between the authors.

Statement of Research and Publication Ethics

The authors declare that this study complies with Research and Publication Ethics.

REFERENCES

Bishop, C. M. (2006). Pattern recognition. Machine Learning, 128(9).

Blomley, R., Jutzi, B., & Weinmann, M. (2016). 3D semantic labeling of ALS point clouds by exploiting multi-scale, multi-type neighborhoods for feature extraction.

Bonaccorso, G. (2017). Machine learning algorithms. Packt Publishing Ltd.

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

Burkov, A. (2019). The hundred-page machine learning book (Vol. 1). Andriy Burkov Canada.

Chehata, N., Guo, L., & Mallet, C. (2009). Airborne lidar feature selection for urban classification using random forests.

Demantké, J., Mallet, C., David, N., & Vallet, B. (2011). Dimensionality based scale selection in 3D lidar point clouds.

Dittrich, A., Weinmann, M., & Hinz, S. (2017). Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3D point cloud data. ISPRS Journal of Photogrammetry and Remote Sensing, 126, 195–208.

Filin, S., & Pfeifer, N. (2005). Neighborhood systems for airborne laser data. Photogrammetric Engineering & Remote Sensing, 71(6), 743–755.

Guarnieria, A., Vettorea, A., Pirottia, F., & Maranib, M. (2009). Filtering of TLS point clouds for the generation of DTM in salt-marsh areas. Laser Scanning, 1–2.

Hackel, T., Wegner, J. D., & Schindler, K. (2016). Fast Semantic Segmentation of 3D Point Clouds With Strongly Varying Density. ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, III–3(July), 177–184. https://doi.org/10.5194/isprsannals-iii-3-177-2016

Hyyppä, J., Jaakkola, A., Chen, Y., & Kukko, A. (2013). Unconventional LIDAR mapping from air, terrestrial and mobile. Proceedings of the Photogrammetric Week, 205–214.

Lee, I., & Schenk, T. (2002). Perceptual organization of 3D surface points. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(3/A), 193–198.

Linsen, L., & Prautzsch, H. (2001). Local versus global triangulations. Proceedings of EUROGRAPHICS, 1, 257–263.

Li, Q., Yuan, P., Lin, Y., Tong, Y., & Liu, X. (2021). Pointwise classification of mobile laser scanning point clouds of urban scenes using raw data. Journal of Applied Remote Sensing, 15(2), 24523.

Mallet, C., Bretar, F., Roux, M., Soergel, U., & Heipke, C. (2011). Relevance assessment of full-waveform lidar data for urban area classification. ISPRS Journal of Photogrammetry and Remote Sensing, 66(6), S71–S84.

Mohamed, M., Morsy, S., & El-Shazly, A. (2021a). Evaluation of data subsampling and neighbourhood selection for mobile LiDAR data classification. The Egyptian Journal of Remote Sensing and Space Science, 24(3), 799–804.

Mohamed, M., Morsy, S., & El-Shazly, A. (2021b). Machine Learning for Mobile LIDAR Data Classification of 3d Road Environment. ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 44, 113–117.

Nguyen, V.-T., Constant, T., Kerautret, B., Debled-Rennesson, I., & Colin, F. (2020). A machine-learning approach for classifying defects on tree trunks using terrestrial LiDAR. Computers and Electronics in Agriculture, 171, 105332.

Pauly, M., Keiser, R., & Gross, M. (2003). Multi-scale feature extraction on point-sampled surfaces. Computer Graphics Forum, 22(3), 281–289.

Pu, S., & Vosselman, G. (2009). Knowledge based reconstruction of building models from terrestrial laser scanning data. ISPRS Journal of Photogrammetry and Remote Sensing, 64(6), 575–584.

Roynard, X., Deschaud, J. E., & Goulette, F. (2018). Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. International Journal of Robotics Research, 37(6), 545–557. https://doi.org/10.1177/0278364918767506

Wang, Y., Jiang, T., Yu, M., Tao, S., Sun, J., & Liu, S. (2020). Semantic-based building extraction from LiDAR point clouds using contexts and optimization in complex environment. Sensors, 20(12), 3386.

Weinmann, M., Jutzi, B., Hinz, S., & Mallet, C. (2015). Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing, 105, 286–304. https://doi.org/10.1016/j.isprsjprs.2015.01.016

Weinmann, M., Jutzi, B., & Mallet, C. (2013). Feature relevance assessment for the semantic interpretation of 3D point cloud data. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2(5W2), 313–318. https://doi.org/10.5194/isprsannals-II-5-W2-313-2013

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

Weinmann, M., Jutzi, B., Mallet, C., & Weinmann, M. (2017). GEOMETRIC FEATURES and THEIR RELEVANCE for 3D POINT CLOUD CLASSIFICATION. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 4(1W1), 157–164. https://doi.org/10.5194/isprs-annals-IV-1-W1-157-2017

Weinmann, M., Urban, S., Hinz, S., Jutzi, B., & Mallet, C. (2015). Distinctive 2D and 3D features for automated large-scale scene analysis in urban areas. Computers and Graphics (Pergamon), 49, 47–57. https://doi.org/10.1016/j.cag.2015.01.006

West, K. F., Webb, B. N., Lersch, J. R., Pothier, S., Triscari, J. M., & Iverson, A. E. (2004). Context-driven automated target detection in 3D data. Automatic Target Recognition XIV, 5426, 133–143.

Yu, Y., Li, J., Guan, H., Wang, C., & Yu, J. (2014). Semiautomated extraction of street light poles from mobile LiDAR point-clouds. IEEE Transactions on Geoscience and Remote Sensing, 53(3), 1374–1386.

Zheng, M., Lemmens, M., & van Oosterom, P. (2017). CLASSIFICATION OF MOBILE LASER SCANNING POINT CLOUDS FROM HEIGHT FEATURES. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, 42.

Zheng, M., Lemmens, M., & van Oosterom, P. (2018). Classification of mobile laser scanning point clouds of urban scenes exploiting cylindrical neighbourhoods. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci, 42, 1225–1228.