Accuracy Assessment and Error Analysis of Labelling in Point Cloud Data: A Comparative Study

Tenzing Pema Thungon
231030063
GEO-INFORMATICS
tenzing23@iitk.ac.in

1. Introduction

Geographical information systems (GIS) and remote sensing play vital roles in diverse fields such as urban planning, environmental monitoring, and forestry management. Accurate labelling and classification of point cloud data are essential for effective decision-making in these domains. This report presents a comparative study aimed at evaluating the accuracy of labelling in point cloud data, specifically focusing on data generated in lab 7. Leveraging LiDAR (Light Detection and Ranging) technology, which offers high-resolution three-dimensional mapping capabilities, the study investigates the process of individual tree segmentation using LiDAR360 and CloudCompare software tools. Terrestrial Laser Scanning (TLS) data processing in forest environments involves several critical steps, including outlier removal, ground and non-ground point classification, normalization, and tree segmentation. By examining these processes and comparing the labelled data with reference datasets, this study seeks to provide insights into the reliability and effectiveness of labelling techniques in point cloud analysis for forestry applications. Additionally, the report will delve into the computation of metrics such as the confusion matrix and kappa value to quantify the accuracy and agreement between the labelled and reference data, providing a comprehensive analysis of error sources and performance metrics in point cloud labelling.

2. Objectives

The primary objective of this report is to evaluate the accuracy of individual tree segmentation derived from TLS (Terrestrial Laser Scanning) data processed using LiDAR360 and CloudCompare software. Specifically, we aim to assess the precision and reliability of the segmented trees through a comprehensive analysis employing confusion matrix and kappa analysis techniques. Additionally, we will explore the integration of MATLAB for advanced data processing and analysis. This report aims to provide insights into the effectiveness of TLS data processing methodologies in forestry applications and the reliability of resulting segmented tree datasets.

3. Methodology

3.1 Segmentation of tree using LiDAR360, CloudCompare

The data collection process involved the use of LiDAR360 software for processing LiDAR point cloud data obtained from terrestrial laser scanning (TLS) of forested areas. LiDAR360 offers a suite of tools for processing, visualizing, and analysing LiDAR data, including individual tree segmentation (ITS) for identifying and delineating trees within the point cloud. The raw LiDAR data files were imported into LiDAR360, and various preprocessing steps were performed to prepare the data for analysis. This included removing outliers from the point cloud data using outlier removal tools, such as the cloth simulation filter (CSF). The CSF algorithm classified points into ground and non-ground categories, essential for subsequent analysis. Normalization of the point cloud data was conducted to adjust the elevation values relative to the ground surface, facilitating further analysis and comparison of different features within the data. During normalization, the original elevation values were preserved as an additional attribute to retain the raw elevation information. The point cloud data were then segmented into individual trees using seed points marked manually or automatically using ITS tools. Seed points were identified as the centre of each tree trunk, serving as reference points for further analysis. Tree attributes such as diameter at breast height (DBH) and crown diameter were estimated based on the seed points. After segmentation, the resulting segmented trees were exported as LAS or LAZ files and imported into CloudCompare software for visualization and analysis. CloudCompare allows for the visualization of segmented trees based on their unique tree IDs, facilitating accurate segmentation and assessment of individual trees.

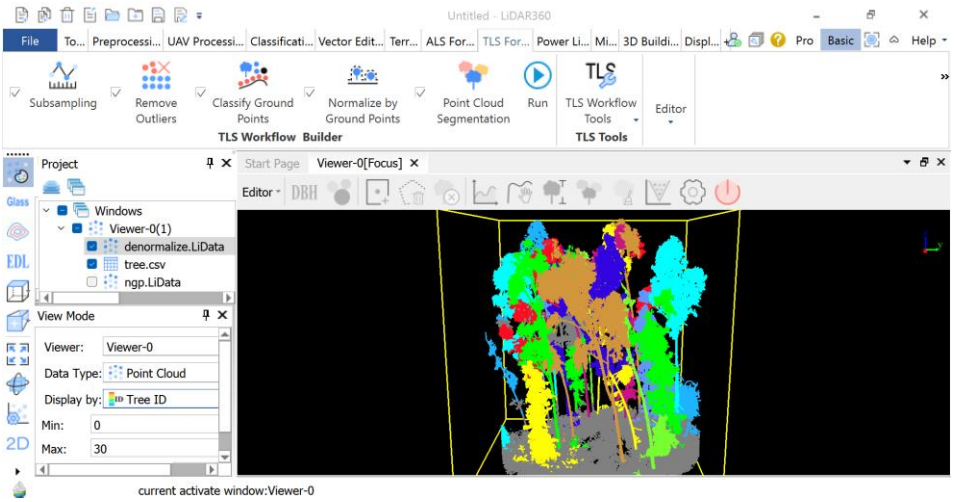

Figure 1: Side view of Denormalized point cloud obtain LiDAR360

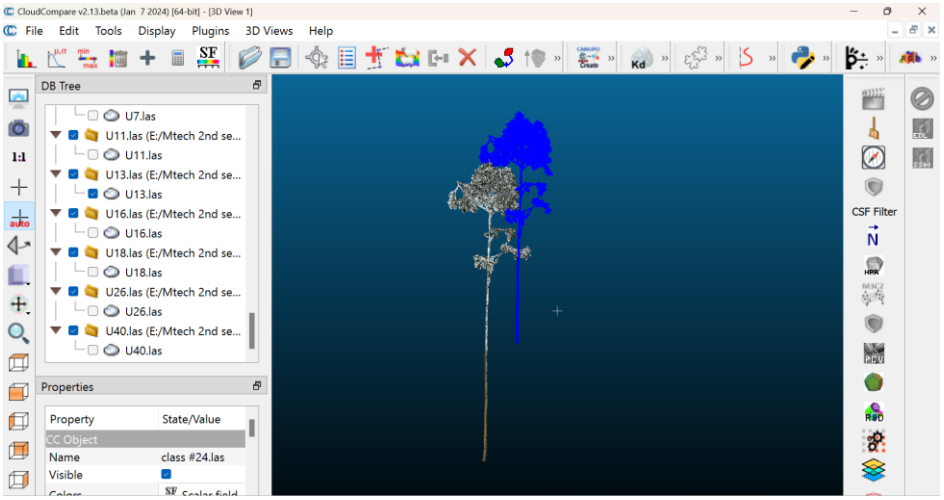

Figure 2: The reference tree with class1 as treeID is represented in RGB, whereas the segmented tree is shown in blue.

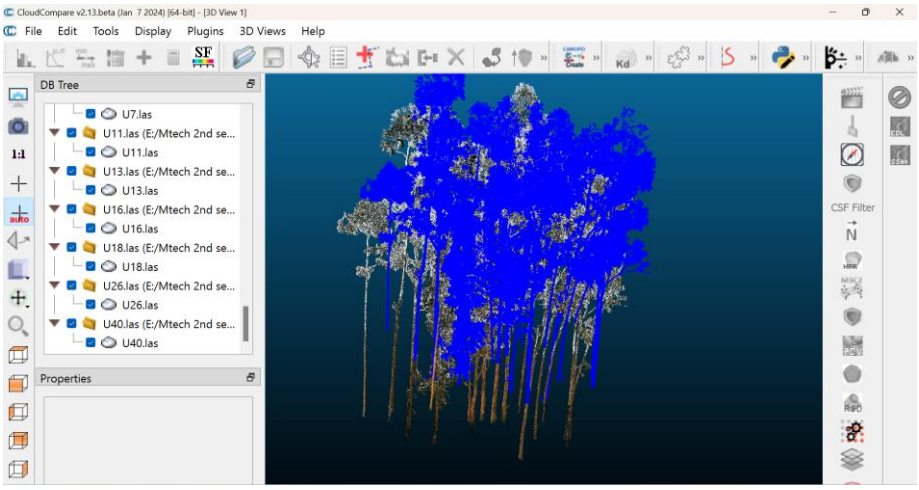

Figure 3: overlaying all the segmented trees and the reference trees

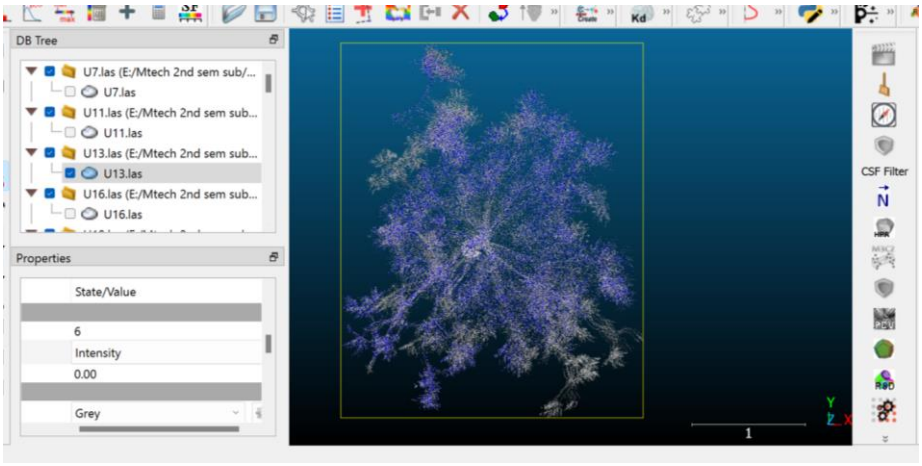

Figure 4: Overlaying both the segmented and the reference tree

After conducting accuracy assessment of the segmented trees, it was observed that while visually similar, the coordinates of the reference tree and our segmented tree did not perfectly align. The coordinates of the reference tree were noted as -687162 and -3394900, whereas our segmented tree had coordinates of -687169.92 and -3394906.49. Although both trees appeared visually similar, they did not overlay precisely. To address this discrepancy, the coordinate of our segmented tree was adjusted to match that of the reference tree. After aligning the coordinates, it was observed that both the segmented and reference trees overlaid perfectly, ensuring accurate comparison and analysis. The aligned trees were then saved as separate CSV files, which would be utilized for generating confusion matrices and conducting further analysis to assess the accuracy of the segmentation process. This step ensured that the data used for accuracy assessment was properly aligned and standardized, enhancing the reliability and validity of the subsequent analysis.

3.2 Accuracy Assessment of Results:

Method 1: Coding Using MATLAB

Confusion matrix

The reference files provide ground truth data against which the segmented trees are compared by Creating confusion metrics for each tree through checking pointwise classification and report the following metrics. We can do it by writing a code.

Formula for the Confusion Matrix:

		Classified by you	
		Tree	*Non-Tree*
Reference data	*Tree*	TP (True Positive)	FN (False Negative)
	Non-tree	FP (False Positive)	TN (True Negative)

$$OA = \frac{TP+TN}{TP+FP+TN+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Comparing segmented trees with reference data helps evaluate the accuracy of tree segmentation algorithms and assess the quality of the segmented trees.

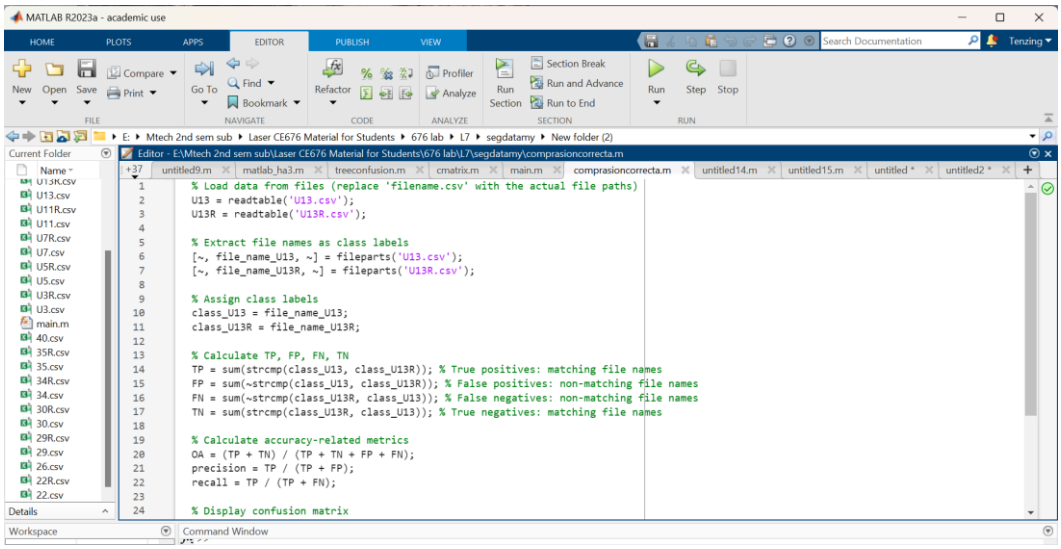

Figure 5: Code is presented for comparing U13 segmented and reference tree data.

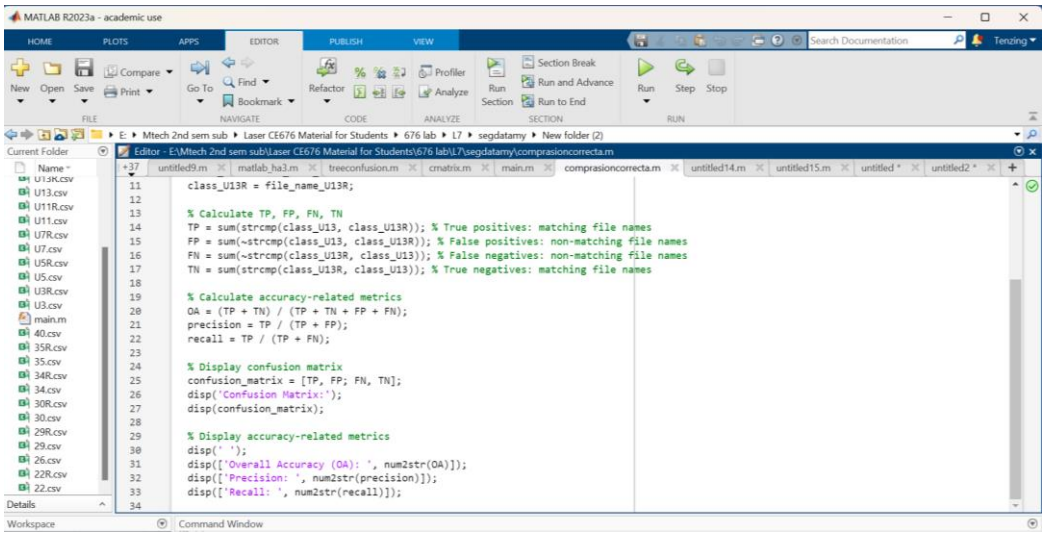

Figure 6: Code is presented for comparing U13 segmented and reference tree data.

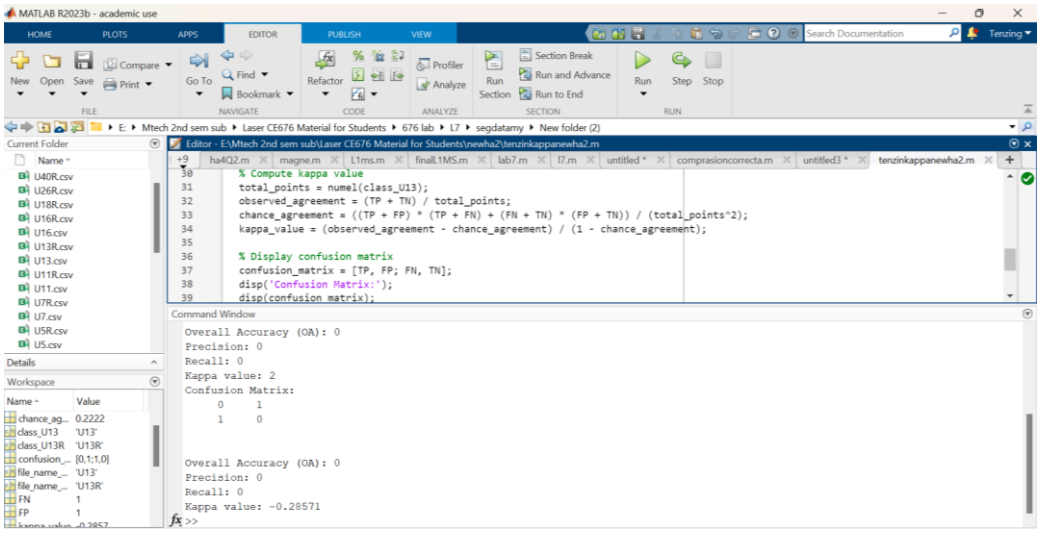

Figure 7: Confusion matrix for comparison of 1 reference and the segmented tree

Confusion matrix was generated by comparing TreeID U13 of the segmented and reference data same can be carried out for all the other points.
The confusion matrix generated indicates that there are two classes being evaluated, labelled as 0 and 1.
Overall Accuracy (OA): The overall accuracy is calculated as the ratio of correctly classified points to the total number of points. An OA of 0 means that none of the points were correctly classified, indicating that the segmentation did not accurately distinguish between the two classes (0 and 1).
Precision: Precision is a measure of the accuracy of the positive predictions. It is calculated as the ratio of true positives (correctly classified points labelled as 1) to the total number of points predicted as positive (both true positives and false positives). In this case, precision is also 0, indicating that there were no true positives.
Recall (Sensitivity): Recall is a measure of the ability of the classifier to correctly identify positive instances. It is calculated as the ratio of true positives to the total number of actual positive instances (true positives and false negatives). A recall of 0 means that none of the actual positive instances were correctly identified by the classifier. The results suggest that the segmentation process did not successfully classify any points correctly, as evidenced by the absence of true positives. This could indicate issues with the segmentation algorithm, insufficient feature representation, or other factors affecting the accuracy of the classification process. Further analysis and refinement may be necessary to improve the segmentation results.
Specificity (True Negative Rate): Specificity measures the proportion of true negative predictions among all actual negative instances in the dataset. It is calculated as the ratio of true negatives (TN) to the sum of true negatives and false positives (FP).
Specificity = TN / (TN + FP)
Specificity is useful when the focus is on correctly identifying negative instances.

Kappa Coefficient: The kappa coefficient, also known as Cohen's kappa, is a statistical measure of inter-rater agreement for categorical items. In the context of classification, it measures the agreement between the observed and expected classification results, while accounting for the agreement that would occur by chance.

$$\text{Kappa} = \frac{N\sum_{i=1}^{k} c_{ii} - \sum_{i=1}^{k}(c_{i+} \times c_{+i})}{N^2 - \sum_{i=1}^{k}(c_{i+} \times c_{+i})}$$

N = This represents the total number of cases in our dataset

Kappa Statistic Formula:

$$\kappa = \frac{(p(a) - p(e))}{(1 - p(e))}$$

- Kappa statistic value (ranges from -1 to 1)
- P(a): Observed agreement between the model's predictions and the ground truth
- p(e): Expected agreement by chance
- $\kappa < 0$: No agreement (worse than chance)
- $\kappa = 0$: Poor agreement or Agreement equal to what would be expected by chance alone.
- $0.81 \leq \kappa \leq 1.00$: Almost perfect agreement
- A Kappa value less than 0 indicates less agreement than expected by chance. In this case, the negative Kappa value (-0.28571) suggests that the agreement between the classifier's predictions and the actual classes is even worse than what would be expected by random chance. This indicates a poor performance of the classifier.
- This method is very time consuming and computationally challenging.

Method 2: Machine learning and deep learning using MATLAB

The analysis conducted on the confusion matrix and Kappa value for a single reference tree and a single segmented tree serves as a demonstration of the evaluation process. However, when dealing with multiple trees, the same methodology would require significant computational resources and time due to the large volume of data. To address this challenge, a more efficient approach was adopted using machine learning and deep learning techniques with MATLAB. Specifically, the Classification Learner app was utilized to handle the extensive point cloud data, which comprised nearly 100,000 data points stored in CSV files. By leveraging machine learning algorithms available in MATLAB, such as decision trees, support vector machines, and neural networks, the classification process could be automated and scaled to handle the complexity and size of the dataset. This approach not only reduces computational burden but also offers the potential to improve accuracy and efficiency in classifying multiple trees within the point cloud data.

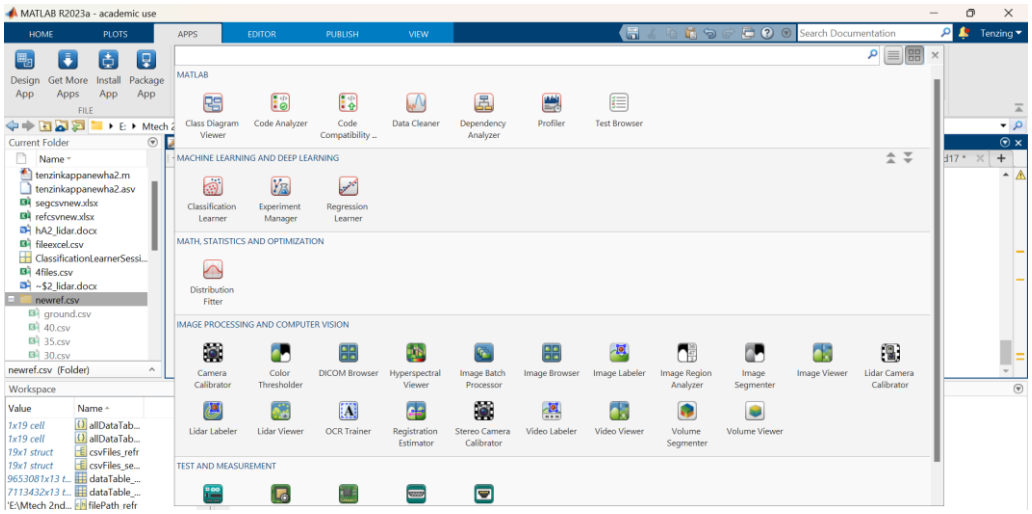

Figure 8: MATLAB interface showing Machine learning and deep learning tools

E:\Mtech 2nd sem sub\Laser CE676 Material for Students\newha2\newnameseg.csv

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes	Folder Path
Binary	11.csv	.csv	12-03-2024 10:28:47 AM	11-03-2024 11:24:36 PM	11-03-2024 11:24:35 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	13.csv	.csv	12-03-2024 10:28:48 AM	11-03-2024 11:25:36 PM	11-03-2024 11:25:35 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	16.csv	.csv	12-03-2024 10:28:49 AM	11-03-2024 11:26:01 PM	11-03-2024 11:25:59 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	22.csv	.csv	12-03-2024 10:28:50 AM	11-03-2024 11:29:53 PM	11-03-2024 11:29:52 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	23.csv	.csv	12-03-2024 10:28:51 AM	11-03-2024 11:32:10 PM	11-03-2024 11:32:09 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	24.csv	.csv	12-03-2024 10:28:52 AM	11-03-2024 11:32:46 PM	11-03-2024 11:32:43 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	25.csv	.csv	12-03-2024 10:28:53 AM	11-03-2024 11:33:15 PM	11-03-2024 11:33:14 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	26.csv	.csv	12-03-2024 10:28:53 AM	11-03-2024 11:33:56 PM	11-03-2024 11:33:53 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	28.csv	.csv	12-03-2024 10:28:55 AM	11-03-2024 11:13:53 PM	11-03-2024 11:13:52 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	29.csv	.csv	12-03-2024 10:28:56 AM	11-03-2024 11:14:31 PM	11-03-2024 11:14:28 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	3.csv	.csv	12-03-2024 10:28:58 AM	11-03-2024 11:22:15 PM	11-03-2024 11:22:14 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	30.csv	.csv	12-03-2024 10:28:58 AM	11-03-2024 11:18:15 PM	11-03-2024 11:18:13 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	34.csv	.csv	12-03-2024 10:28:59 AM	11-03-2024 11:18:48 PM	11-03-2024 11:18:44 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	35.csv	.csv	12-03-2024 10:29:02 AM	11-03-2024 11:19:19 PM	11-03-2024 11:19:16 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	4.csv	.csv	12-03-2024 11:12:05 AM	11-03-2024 11:22:44 PM	11-03-2024 11:22:41 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	40.csv	.csv	12-03-2024 10:29:05 AM	11-03-2024 11:19:51 PM	11-03-2024 11:19:49 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	5.csv	.csv	12-03-2024 10:29:06 AM	11-03-2024 11:23:26 PM	11-03-2024 11:23:25 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	7.csv	.csv	12-03-2024 10:29:06 AM	11-03-2024 11:24:01 PM	11-03-2024 11:24:00 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...
Binary	ground.csv	.csv	12-03-2024 10:29:07 AM	11-03-2024 11:21:38 PM	11-03-2024 11:20:48 PM	Record	E:\Mtech 2nd sem sub\Laser CE676 Material for Studen...

Figure 9: Merging all the .csv file into one file in using excel

Classification Learner:

The Classification Learner is a powerful tool in MATLAB designed for building and training classification models using machine learning techniques. Unlike the Regression Learner, which is used for predicting continuous values, the Classification Learner focuses on predicting categorical outcomes. In our case, since we are dealing with classifying trees (categorical), the Classification Learner is the appropriate choice. We start by adding the segmented tree LAS file to a CSV file. Our CSV file contains features such as X, Y, Z coordinates, and tree IDs, which serve as inputs to the classification model. We choose a suitable algorithm for our classification task. Decision trees are a popular choice due to their interpretability and ability to handle both numerical and categorical data.

Then we select the relevant features (X, Y, Z, tree ID) from our CSV file to be used as input features for training the model. With the selected features, we train the decision tree model using the available data. The model learns patterns and relationships between the input features and the target variable (tree class). Once trained, we evaluate the performance of the model. This includes generating visualizations such as scatter plots to visualize the predicted classes (red and blue dots) compared to the actual classes. The cross marks indicate misclassifications. By clicking "Train All," the model is evaluated on the training data, and a confusion matrix is generated. This confusion matrix provides a detailed breakdown of the model's performance, showing the predicted classes (X-axis) against the true classes (Y-axis). The numbers in each cell represent the count of data points classified into each class.
The presence of 6 classes in the confusion matrix could indeed indicate that the segmented tree data includes not only the target tree but also other trees and ground points from the surrounding environment. When processing LiDAR point cloud data, it is common to capture not just the target object (in this case, a single tree) but also surrounding objects and features.

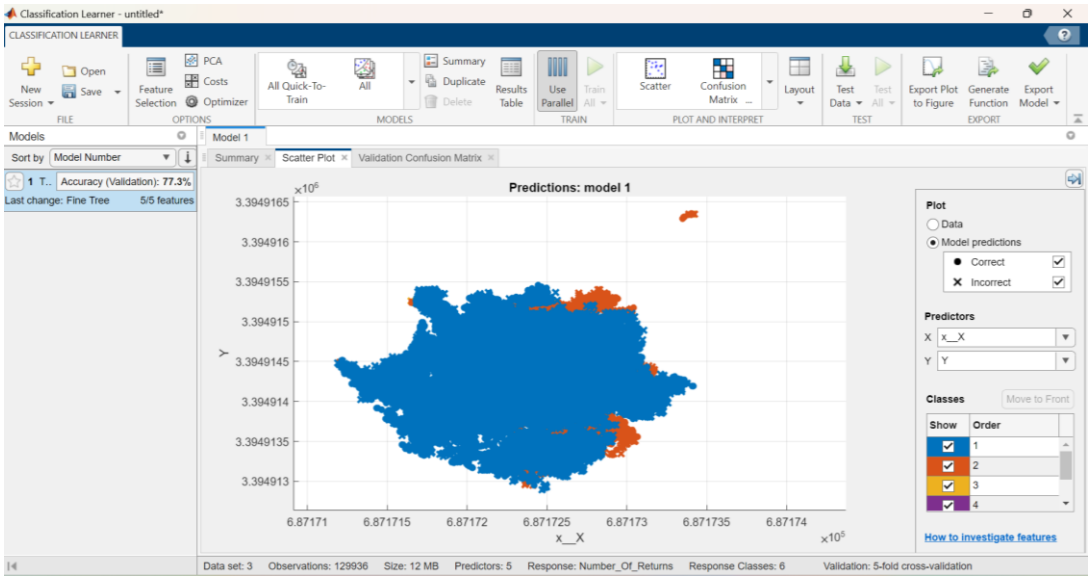

Figure 10: Scatter plot of a single segmented tree

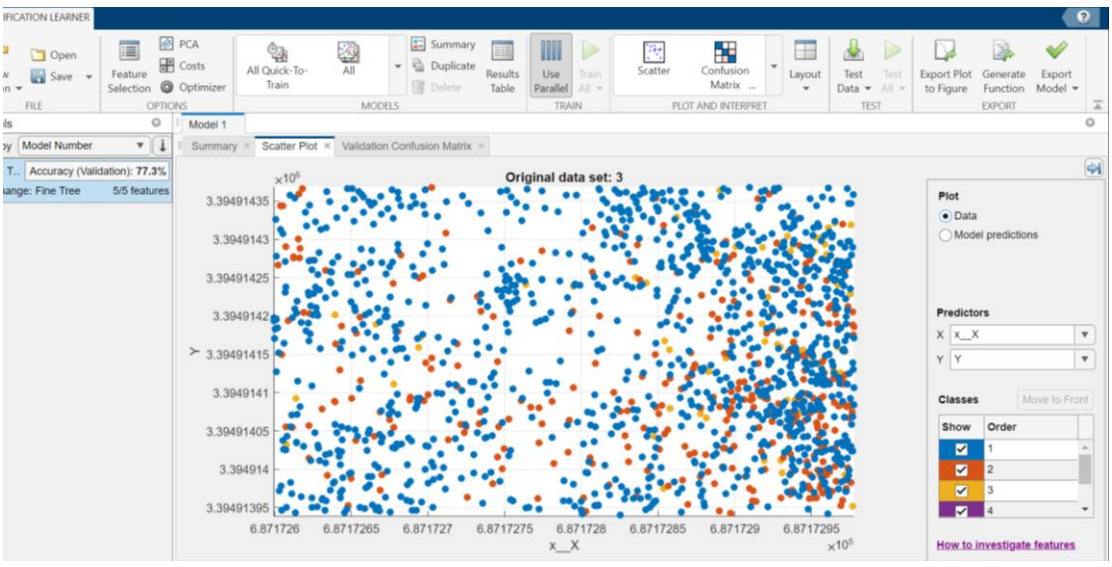

Figure 11: scatter plot showing original data set

The original dataset points are plotted on the scatter plot, each coloured or symbolized according to its true class or label. These points represent the ground truth or actual data points obtained from the LiDAR scan.

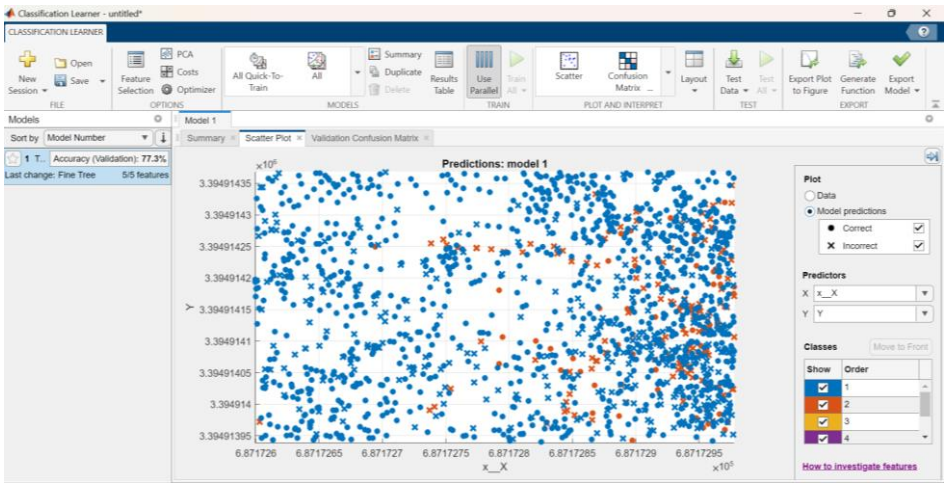

Figure 12: prediction model

The prediction model generates predictions for each data point in the dataset. The scatter plot also includes points representing the predicted class for each data point, typically indicated by a different colour or symbol. This allows us to visually compare the predicted classes with the actual classes.

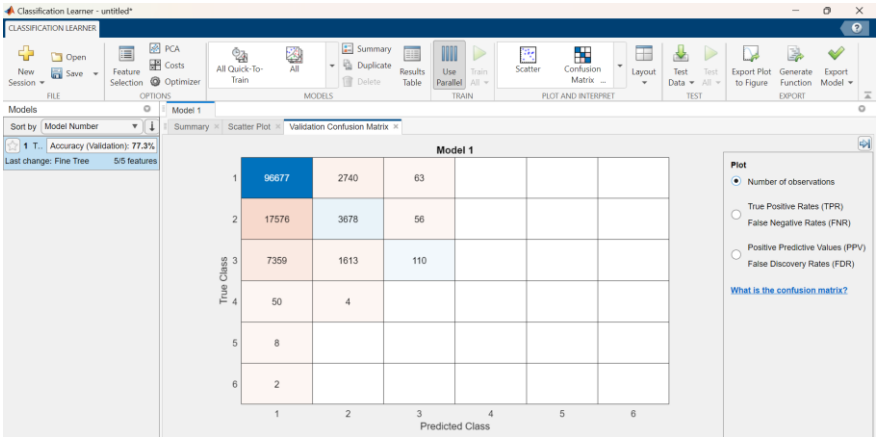

Figure 13: Number of observations

The confusion matrix gives crucial information about the model's performance. Each class is assessed for true positive rates (TPR), false negative rates (FNR), positive predicted values (PPV), and false discovery rates (FDR). These metrics assist evaluate the model's ability to accurately classify examples of each class as well as the rate of misclassifications. The model is trained using the available data without explicitly providing reference tree data. Instead, the model learns patterns and relationships within the input features to make predictions. The training process involves adjusting the parameters of the decision tree algorithm to minimize the classification error on the training data.

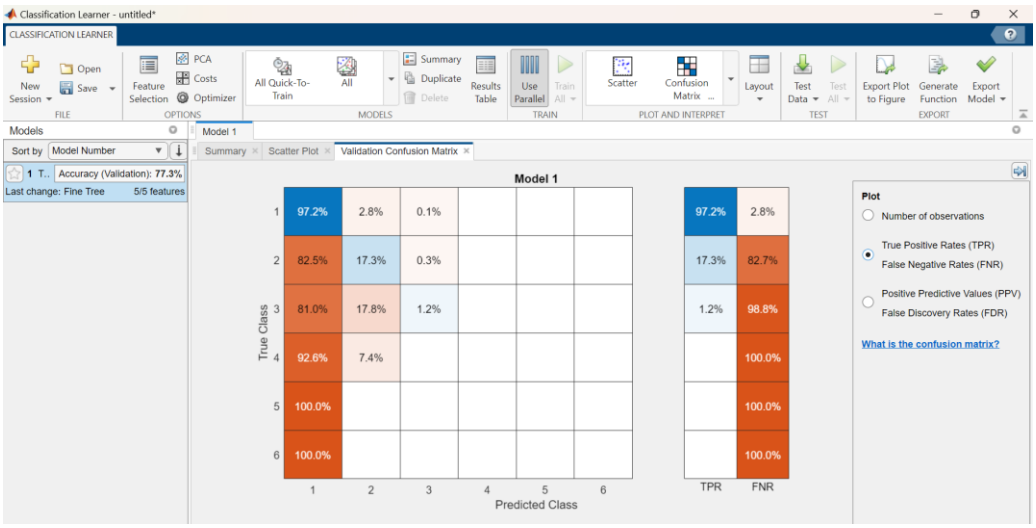

Figure 14: confusion matrix showing TPR and FNR

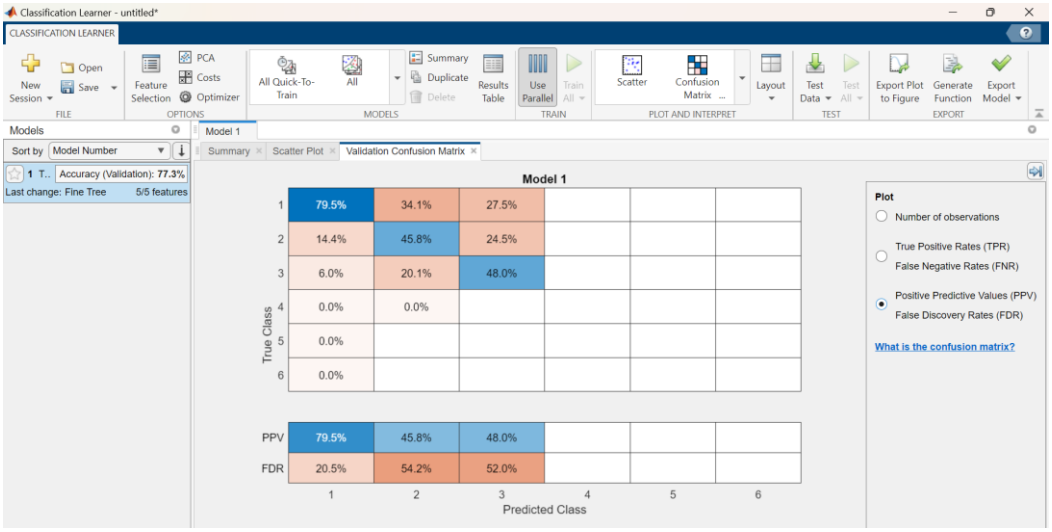

Figure 15: Confusion matrix showing PPV and FDR

An accuracy validation of 77.3% indicates the overall performance of the classification model in correctly predicting the classes of the data points. Around 22.7% of the data points were assigned incorrect classes by the model, meaning they were classified as belonging to a different class than their true labels.

Kappa Value:

$$\kappa = \frac{(p(a) - p(e))}{(1 - p(e))}$$

Convert the rates to proportions:
ΣTP = 0.972+0.173+0.012 = 1.157
ΣFP = 1.733
ΣFN = 4.83
ΣTN = 1.267

Calculate the observed agreement (*po*):

$$Po = \frac{\Sigma TP + \Sigma TN}{\Sigma TP + \Sigma FP + \Sigma FN + \Sigma TN} = 0.269$$

Calculate the expected agreement (*pe*):

$$Pe = \frac{(\Sigma TP+\Sigma FP)(\Sigma TP+\Sigma FN)+(\Sigma FN+\Sigma TN)(\Sigma FP+\Sigma TN)}{(\Sigma TP+\Sigma FP+\Sigma FN+\Sigma TN)^2} = 0.356$$

po and *pe* into the Kappa formula:

k =0.175 (Fair agreement)

A Cohen's kappa coefficient (κ) value of approximately 0.175 indicates slight agreement between the observed and expected classifications.

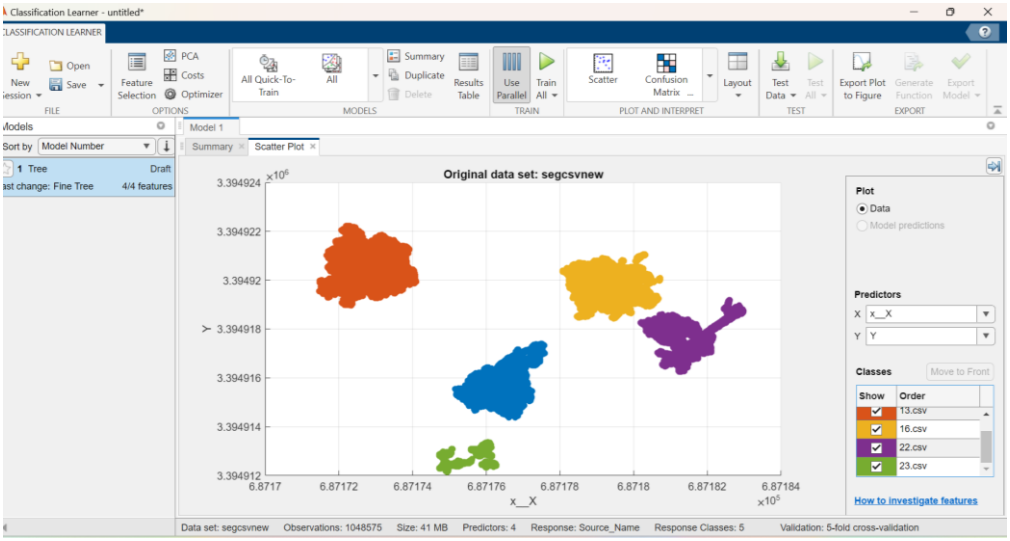

Figure 16: plotting 6 segmented trees

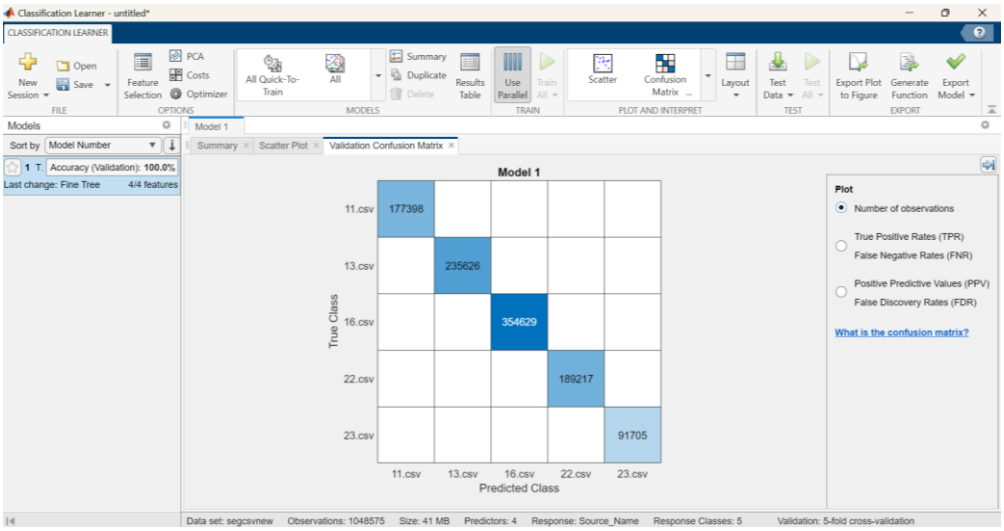

Figure 17: showing number of observations

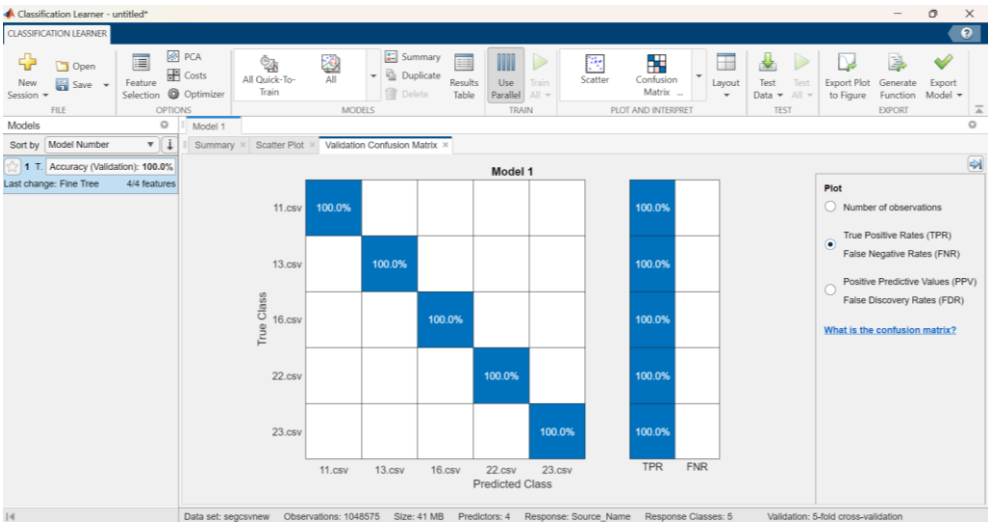

Figure 18: Showing TP and FN

Here all the true positive (TP) rates are 100% and the true negative (TN) rates are 0, along with positive predicted values at 100% and a false discovery rate of 0, it suggests that the classification model is exhibiting perfect performance. This scenario indicates that the model is correctly identifying all the positive instances (trees, in this case) in the dataset without any false positives. Essentially, every tree that should be identified is indeed classified as such, and there are no instances where non-tree points are incorrectly labelled as trees. Further investigation into the model's training process, data preprocessing steps, and validation procedures may provide insights into the reasons behind this seemingly perfect performance.

Here a kappa value of 1 indicates perfect agreement between the observed classifications and the expected classifications. In other words, there is complete agreement between the model's predictions and the true labels in the dataset. This suggests that the model's performance is flawless, with no discrepancies or errors in its predictions. However, it is essential to interpret this result cautiously and consider factors such as the size and composition of the dataset, the model's complexity, and the possibility of overfitting. While a kappa value of 1 is ideal, it is crucial to validate the model's performance across different datasets and ensure its generalizability before drawing definitive conclusions about its effectiveness.

4. Results:

The comparative study conducted to evaluate the accuracy of individual tree segmentation in point cloud data revealed valuable insights into the effectiveness of different processing methodologies and analysis techniques. Utilizing LiDAR360 and CloudCompare software tools, the segmentation process involved several critical steps, including outlier removal, ground and non-ground point classification, normalization, and tree segmentation. While visually similar, discrepancies in coordinate alignment between segmented and reference trees underscored the importance of precise data preprocessing and alignment for accurate comparison and analysis. The subsequent analysis, employing confusion matrix and kappa analysis techniques, provided quantitative measures of accuracy and agreement between segmented and reference data. Through both manual coding in MATLAB and machine learning using the Classification Learner app, the accuracy of segmentation results was assessed, revealing varying levels of agreement and performance.

The confusion matrix generated for a single segmented tree and reference tree explained the classification performance, highlighting key metrics such as overall accuracy, precision, recall, and specificity. Additionally, the kappa coefficient, calculated from observed and expected agreement rates, provided a standardized measure of agreement between predicted and actual classifications. The results indicated varying levels of agreement and performance across different methodologies, with some approaches demonstrating near-perfect agreement while others exhibited only slight agreement. The integration of machine learning techniques using MATLAB offered a more efficient and scalable approach for processing large volumes of point cloud data, enabling automated classification and evaluation of segmentation results.

5. Conclusion:

The accuracy assessment and error analysis of labelling in point cloud data carried out in this study have provided meaningful insights into the performance of the labelling process and classifying point cloud data. The use of precision, recall, overall accuracy, and Kappa value for each class has enabled a comprehensive evaluation of the labelled data. Furthermore, the separation of points into true positives, false positives, and false negatives, along with their visualization in a point cloud viewer, has facilitated a deeper understanding of the sources of errors in the labelling process. By identifying the factors contributing to false positives and false negatives, this study has laid the groundwork for improving the accuracy of labelling in point cloud data. Moreover, the integration of machine learning techniques presents promising avenues for improving efficiency and scalability in processing large-scale point cloud datasets. The findings and methodologies presented here can serve as a valuable resource for future research and practical applications in geographical information systems, remote sensing, and related fields.

6. References

Jie Shan, C. K. (2008). Topographic Laser Ranging and Scanning . *Taylor & Francis Group, LLC.*