Academic Year: 2022 – 2023
Laboratory: OSUR
Research Unit: Plateforme LIDAR
Supervisor(s): Mathile Letard & Paul Leroy

MASTER ISC

3DMASC : Objective Evaluation on Benchmarked Urban Environment LiDAR Data

Pierre Lague

February 20, 2023

Abstract

This report presents the work undertaken during a 1 month internship with the "Plateforme LiDAR" team at the "Observatoire des Sciences de l'Univers de Rennes" (OSUR) under the supervision of Mathilde Letard and Paul Leroy. The goal was to **Compare a recently developped machine learning algorithm based on classical shallow learning approaches (random forests and handcrafted features), called 3DMASC, to state-of-the-art deep learning methods on bench marked datasets for the classification of 3D Point Cloud Data.** The objectives for this internship are to evaluate 3DMASC on a benchmarked dataset and submit our results to assess the performances of 3DMASC and compare it to other methods. In addition to this, the objective is to present and return scientific results and experiments according to standard redaction of a report.

The work will first present a state-of-the-art in 3D point clouds classification and benchmarked dataset of 3D point clouds. Then, the data engineering process on the training and validation sets, a methodology based on 3 approaches. Following, the description of the optimization routine improving the models performances, deploying the model on the benchmarked dataset and submitting our results, and discussing the results. Finally, concluding on the experimentation and opportunities 3DMASC represents.

Our results show that 3DMASC has the capability to compete with state-of-the-art deep-learning methods in terms of compactness and interpretability, but that it performed poorly because of reasons related to feature optimization and data engineering.

This internship represented a very good opportunity for me to take part into a scientific project related to artificial intelligence during the semester break. It proved itself very instructive and allowed me to manipulate algorithms related to classification and objective evaluation.

Keywords: LiDAR, Machine Learning, 3D Data, state-of-the-art, Multi-scale classification, Multi-cloud classification, Random Forests

Contents

1 Presentation of the Laboratory

1.1 The OSUR

The Observatoire des Sciences de l'Univers de Rennes (OSUR) is a public laboratory afiiliated to the University of Rennes, the CNRS and the University of Rennes 2(Bretagne). It has four main missions: i) environmental observation, ii) sharing of analytical and technical means, iii) animation of interdisciplinary research and iv) training at undergraduate and graduate level. OSUR is a Research and Support Unit between the University of Rennes, the University of Rennes 2, CNRS (INSU), INRAE and Institut Agro AGROCAMPUS OUEST.

1.2 The "Plateforme LiDAR"

The "Plateforme LiDAR" is a research and development structure of the OSUR. It designs methods to exploit massive 3D data (classification, segmentation, change detection, etc.) from airborne or ground-based LiDAR sensors in relation to various scientific challenges in environmental sciences: natural hazards (landslides, floods, ...), environmental monitoring (soil erosion, geomorphological analysis ...). They ensure the dissemination of these methods, in particular through the Cloudcompare software and the training of users. They have a terrestrial LiDAR and an airborne topo-bathymetric lidar sensor, unique in France, allowing simultaneous 3D measurement of topography, vegetation and shallow submerged surfaces.

Following more than 10 years of development in 3D data processing at OSUR and in hyperspectral data processing at OSUNA, the platform was created in collaboration between the Universities of Rennes 1 and Nantes, which acquired in 2015, the first topo-bathymetric LiDAR in France: an Optech Titan DualWavelength.

The LiDAR platform is specialized in LiDAR topo-bathymetric acquisition in a river context: geomorphological/hydrological expertise, feasibility study of lidar acquisitions and flight plans. The platform has developed its own tools for processing river topo-bathymetric LiDAR data and their applications: soil/water classification, refraction correction, exploitation of the full wave return, coupling with hydraulic simulations. The platform is also developing numerous algorithms for the exploitation of 3D time series to automatically extract changes of very low amplitude (< 1 cm in terrestrial lidar, < 10 cm in airborne lidar) such as 3DMASC.

1.3 The workspace and team

My two tutors Paul Leroy and Mathile Letard are both from the "Plateforme LiDAR" and actively contribute to the on-going project that is 3DMASC (presented later) and its evaluation. Mr Leroy is a CNRS research engineer and is working on the implementation and optimization of the 3DMASC plugin for CloudCompare in C++. Mrs Letard is a second year PhD student working on the implementation of the underlying algorithm for 3DMASC and the optimized feature selection in python. During the internship, the scripts used for feature optimization come from the "plateforme_lidar" repository on git.

2 Introduction

2.1 Context

We are recently witnessing an increasing availability of not-interpreted point clouds (Fig. 1) and 3D models, often shared online using point-based rendering solutions (e.g. CloudCompare, PoTree). A Point Cloud (PC) is a set of 3D points in which each element is a vector of coordinates (x,y,z) with associated

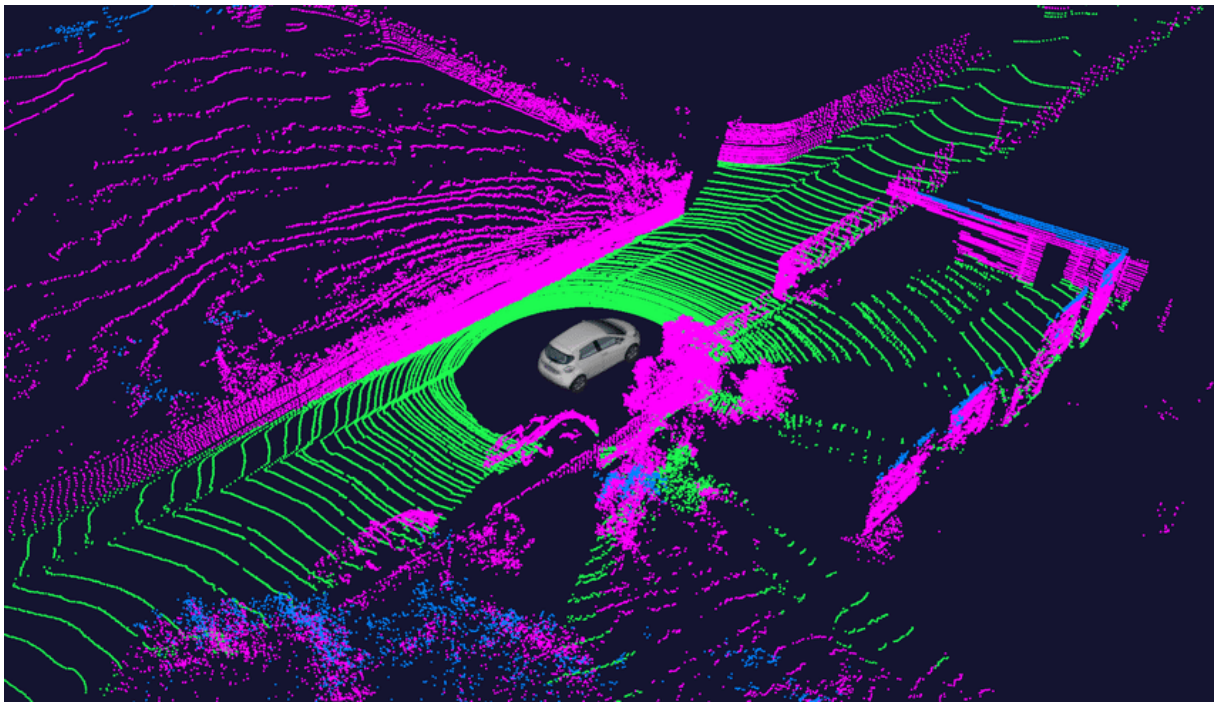

Figure 1 – Illustration of 3D Point cloud segmentation following the road slope. Ground points are green, obstacles are pink.

point based features : intensity, multi-echo characteristics, RGB color. If we focus on point clouds, there is a growing need of innovative methods for the treatment and analysis of these data and for their classification in order to fully harness their informative value. 3D point clouds are the simplest but at the same time powerful collection of elementary geometrical primitives able to represent shape, size, position and orientation of objects in space. This information may be enhanced with additional contents obtained from other sensors or sources, such as colours, multispectral or thermal information, etc. For a successful exploitation of point clouds and to better understand them, we must first proceed with segmentation and classification procedures. The former refers to a group of points in subsets (normally called segments, Fig. 1) characterized by having one or more characteristics in common (geometric, in a neighbourhood etc.) whereas classification means the definition and assignment of points to specific classes ("labels") according to different criteria.

The complexity and variety of point clouds caused by varying density, irregular sampling and different types of objects, etc., has led point cloud classification and segmentation to become very active research topics (Zhang et al. 2018 proposing a Graph-CNN model for 3D point cloud classification, Liu et al. 2021 proposing a robust model resisting to malicious input or data modification). Multiple research studies related to these two topics, are driven by specific needs provided by the field of application (robotics, autonomous driving, underground mapping, etc.).

Many algorithms require a fine-tuning of different parameters depending upon the nature of data and applications. Supervised methods are the most common. They require a mandatory training phase, necessary to guide the successive machine learning classification solution. The results are generally affected by noise and density of the cloud as well as by the quality of the training data (labelling approximation and class choice). Different benchmarks were proposed in the research community, the most comprehensive study being carried out by the previous ISPRS Working Group III/3 "3D Reconstruction from Airborne Laser Scanner and InSAR Data"' with the aim to segment and classify points in bare earth and object classes (Sithole and Vosselman, 2003; Sithole and Vosselman, 2004). The study was initiated to compare the performance of various automatic filters with the purpose of (i) determining the comparative performance of existing filters, (ii) understanding the influence of point

density on the filter performance and (iii) identifying directions for future research on point clouds filtering algorithms. A more recent benchmark is the “Semantic Kitti : A Dataset for Semantic Scene Understanding using LiDAR Sequences" (http://www.semantic-kitti.org/) that provides a sequence of labelled terrestrial 3D point cloud data (urban environment) on which people can test and validate their algorithms (Fig. 1).

However, many of these algorithms and methods remain out of reach to most people due to their difficulty to be explained, interpreted and undertaken. In this report, we will compare a newly developed algorithm (3DMASC, 3D Multiple Attributes, Scales, and Clouds) to the state-of-the-art methods used to classify 3D Data of urban environments. 3DMASC has proven itself very performant in natural environments

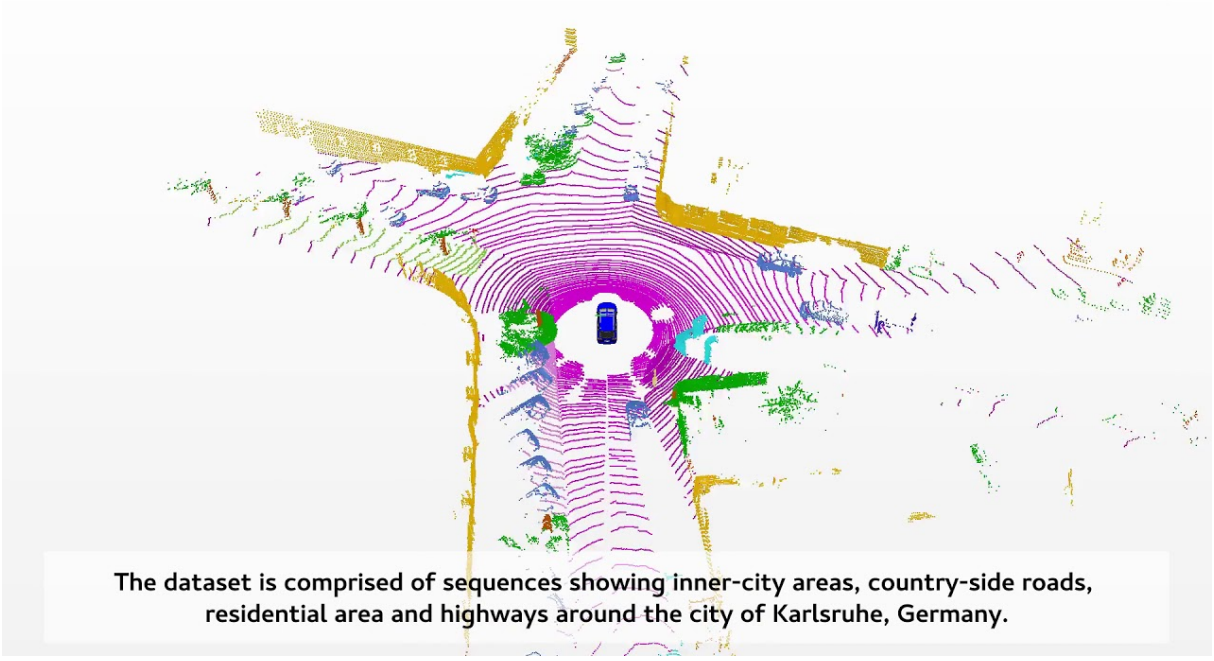

Figure 2 – Example of a classified point cloud from Semantic Kitti Dataset (https://www.youtube.com/watch?v=3qNOXvkpK4I)

2.2 3DMASC

3DMASC is a framework for 3D points classification with Multiple Attributes, Multiple Scales and Multiple Clouds (Letard et al. 2023, sub). It is a supervised approach using handcrafted features and a random forest algorithm. So far it has been applied to coastal and fluvial TB (Topo-Bathymetric) airborne lidar datasets. It combines proven classical elements of single point cloud semantic classification, such as geometric feature extraction from multi-scale spherical neighborhoods or k-nearest neighbors (Thomas et al., 2018) and a random forest machine learner (Breiman, 2001). It’s contribution consist in :

- 3DMASC can use 2 different point clouds (different wave length clouds or two successive clouds) allowing it to compute different features based on the joint-cloud local geometry.
- Screening and selecting features and scales truly contributing to 3D semantic point classification in order to develop an optimal classifier (efficiency, capability and interpretability).
- Using limited data (<2000 points per class), it is able to achieve significant score on TB lidar datasets (OA (overall accuracy) >0.95 Fig. **??**).

We will try to evaluate 3DMASC on bench-marked datasets such as Hessigheim 3D, SimKITTI64, the simulated version of SemanticKitti (Fig. 1) (see Additional Work section). The goal is to show that 3DMASC can generalize itself to other environments than natural environment where it was developped (urban environments) and is at least as efficient as the state-of-the-art deep learning methods. 3DMASC

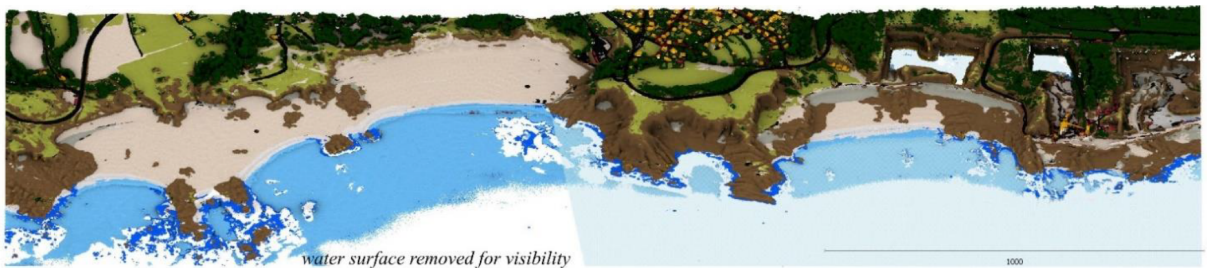

Figure 3 – Classified point cloud by 3DMASC of the "Cap Frehel" region of study.

is available as a plugin in CloudCompare and was also implemented in Python. CloudCompare (Daniel Giradeau-Montault, 2011, see Appendix B), is an open source 3D point cloud (and triangular mesh) editing and processing software and is one of the most used tool in the environmental science community using 3D point clouds.

3 Related Work

The progress of computer vision has always been driven by the development of new algorithms and bench-marked datasets to test them on. Even in 2D, milestones such as the availability of the large-scale image dataset *ImageNet* was crucial for deep learning to develop itself. Regarding the application in which a study is made, task-specific datasets appeared such as *Citiscapes* dataset (Cordts et al. 2016), which is the first dataset for self driving cars providing a considerable amount of pixel-wise labelled images. And for generalized models, the *Mapillary Vistas* dataset (Neuhold et al. 2017) provides the amount and the diversity that *Cityscapes* could not. However, regarding 3D point-based labelled datasets, the development of technologies such as LiDAR (Li et al., 2020) has made it possible for researchers to build considerable datasets, providing a variety of classes and samples in different environments.

3.1 Benchmarked Datasets

For this internship, we have chosen the Hessigheim 3D (Kölle et al., 2021) benchmark as an example of airborne LiDAR data. It was easy to get the data and submit our results to compare our results. THe Heissgheim 3D data provides :

- UAV-based (aerial scene) simultaneous data collection of both LiDAR data and imagery from the same platform.
- High density LiDAR data of 800 points/m² enriched by RGB colors of on board cameras incorporating a GSD of 2-3 cm.
- Manually set labels for the LiDAR point cloud with 11 classes.

We also aimed at using the SimKItti64 (Richa et al. 2022) datasets as it is obtained from a mobile vehicle and typically pertains to challenges related to autonomous vehicle. It proposes :

- A dataset created simulating a Velodyne HDL-32 (terrestrial scene) inside a scene modeled from the SemanticKITTI (Behley et al., 2021) dataset (acquired using a Velodyne HDL-64).
- Sequence labelled point clouds recorded at a rate of 10 Hz. Enables the use of temporal information and aggregation of information on mulitple-cloud analysis.
- Moving and non-moving objects are annotated with distinct classes (cars, trucks, pedestrians and bicyclists), a total of 34 classes.

Both of these datasets allowed multiple methods to be developed and tested on.

3.2 Classification on Point Clouds

Attempts to classifying point clouds were developed by adapting ideas from deep learning on images, e.g., using multiple view images (Hang Su et al. 2015, Tan Yu et al. 2018), or applying convolutions on 3D voxel grids (Zhirong Wu et al. 2015, Maturana et al. 2015). Then, appeared the transfert of convolution operations from 2D to 3D, it is shown that performing convolutions on a point cloud is not an easy task (Manzil Zaheer et al. 2017). The difficulty comes from the fact that a point cloud has no order of points on which convolutions can be performed. Alternatively, some other methods proposed to learn local features from convolutions, e.g., (Qi et al. 2016, Li et al. 2018, Hua et al. 2018, Dominguez et al.) or from autoencoders (Yang et al. 2018). It is also possible to treat point clouds and views as sequences (Han et al. 2018, 2019), or to use unsupervised learning. Recent works demonstrate very competitive and compelling performances on standard datasets. For example, the gap between state-of-the-art methods such as SpecGCN (Wang et al. 2018), SpiderCNN (Xu et al. 2018), DGCNN (Yang et al. 2018), PointCNN (Li et al. 2018) is less than ModelNet40 dataset (Wu et al. 2015). In the online leaderboard maintained by the authors of ModelNet40, the accuracy of the object classification task is reaching perfection, with 92% for point cloud methods (Li, WangXu et al. 2018). Notorious methods such as SPGraph (Landrieu et al., 2018) are tackling problems related to the large scale of the data and the lack of clear structure in the point clouds. KPConv (Thomas et al., 2019) is very efficient in 3D Convolution, being the first algorithm to propose a flexible and variant convolution kernel for 3D points achieving significant scores and tackling problems linked to complex structures classification. And finally PointNet++ (Qi et al., 2017).

4 Methodology

In this section we will describe the methodology we used to evaluate 3DMASC and engineer the different benchmark datasets. We will provide details of procedure and convention to use 3DMASC in CloudCompare and in Python (see Appendix A for a functional representation of the methodology). The study proposes 3 different approaches : the first one is without a context cloud, it allows us to see how the model performs from its basic configuration. The second approach is with a context cloud (needing to build a model capable of creating a context), allowing us to assess the contribution of context-based features. This second approach will end by generating a context cloud which will be filtered according to the classification confidence in order to keep only high confidence classified points, making the context cloud reliable. Finally, the third approach is with a context cloud and feature optimization to have the final model for the test inference.

4.1 Computing Features and Scales

The underlying algorithm of the 3DMASC plugin takes a 3D point cloud as input. After the inference, it produces a classified cloud. In addition to point-based features, 3DMASC computes neighbourhood-based features defined by a spherical neighbourhood or KNN search. 4 different entities can be involved in the extraction of these features :

- 1-2) two point clouds. The originality of 3DMASC lies in using up to two PCs to characterize the scene of interest. For urban environment classification applications, they originate from different sequences of clouds. We refer to them as PC1 and PC2, respectively. For this study, only 1 cloud will be used (PC1) on the Hessigheim 3D dataset.
- 3) A set of core points, denoted PCX, that 3DMASC classifies at the end of the process. Usually a subset (regular subsampling) of PC1 or PC2.
- 4) An optional context PC, denoted CTX. Contains spatial and contextual information, at a much lower resolution than PC1 or PC2. A typical CTX would be previously classified ground points at 2 m spatial resolution.

Additionally, the features are all specified in a parameter file following a particular syntax. Features of geometric, contextual, intensity and color nature can be specified and later used (or not) in the model. As previously mentioned, the neighbourhood-based features are computed according to a certain scale (eg. 0.5, 1, 1.5, 4). A scale is defined as the sphere diameter of the spherical neighbourhood. 3DMASC uses a multi-scale classifier computing multiple neighborhoods for each core point. The computed features are used in the random forest classifier to predict a label for each point of the input PC, using the predictor vector used to describe the many aspects of 3D objects. At the end of the training phase, the plugin provides metrics on feature importance and model performance on the validation set.

4.2 Hessigheim 3D Dataset

The benchmark datasets all differ from one another by their data format, the way they are organized, the way they were labelled and other caracteristics. We decided to visualize them first and try to understand how they were arranged using CloudCompare.

The first benchmarked dataset studied was Hessigheim 3D (Kölle et al., 2021) proposing 3 point clouds organized according to Fig. 4 :

— A training set (79 Million points).
— A validation set (19 Million points).
— A test set (51 Million points).

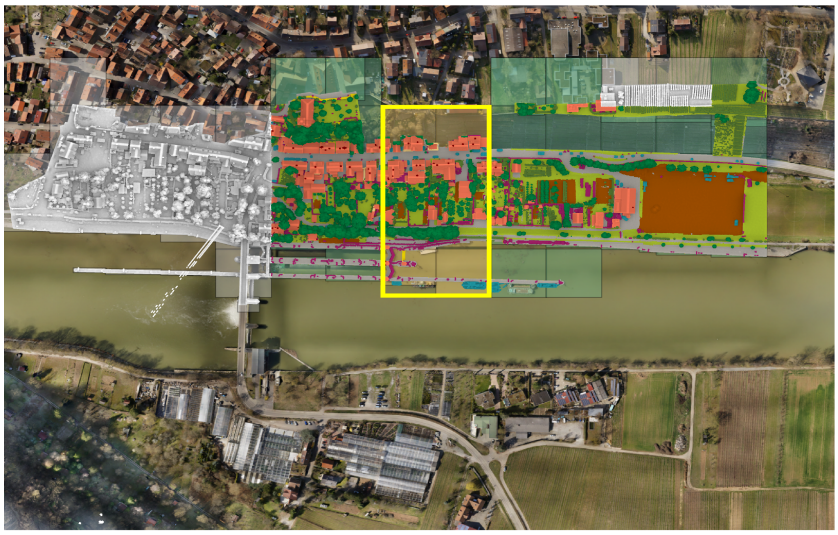

Figure 4 – Partition of the H3D(PC) dataset (epoch March 2018) into training (colored by class colors), validation (colored by class colors and marked by yellow box) and test set (grey).

The dataset is composed of 11 un-balanced classes : Low Vegetation, Impervious Surface, Vehicle, Urban Furniture, Roof, Facade, Shrub, Tree, Soil/Gravel, Vertical Surface and Chimney (numbered from 0 to 11). We spotted some inconsistencies in the labelling, especially for the classes related to the groud (approximate labelling of low vegetation and miss-labelling of ground volumes). Because the data is very dense, the first step was to subsample the clouds to limit the inference and computing time. We've balanced the classes to 7000 samples per class in the training and validation set. The 3DMASC plugin in CloudCompare provided the first results and gave an estimation of the relevant features and scales.

4.2.1 First Approach : no context clouds

Initial tests on the training and validation sets returned OA scores around 0.7. During those tests, no context clouds were used. The confusion matrix (Fig. 5) extracted from the training session in Table 1. shows that ground-related classes like Impervious Surfaces (class 1), Low Vegetation (class 0) and

Class	Precision	Recall	F1-Score
Low Vegetation	**0.54**	0.66	0.6
Impervious Surface	0.68	0.69	0.69
Urban Furniture	0.68	0.67	0.67
Roof	**0.4**	0.53	0.46
Facade	0.87	0.87	0.87
Shrub	0.6	0.76	0.67
Tree	0.68	0.55	0.61
Soil/Gravel	0.84	0.85	0.84
Vertical Surface	0.66	0.41	0.51
Chimney	0.9	0.79	0.84
Vehicle	0.94	0.85	0.89

Table 1 – First test : 7000 points per class, no context cloud. OA : 0.69

Soil/Gravel (class 8) are miss classified and lead to a less robust and discriminant model. The "urban furniture" class is also miss classified due to it's spatial position and density (few points per objects and few objects in the scene). These problems can be solved by making our model more discriminant between spatially close classes. Hence the use of a context point cloud.

	Low Vegatation	Impervious Surface	Vehicle	Urban Furniture	Roof	Facade	Shrub	Tree	Soil/Gravel	Vertical Surface	Chimney
Low Vegatation	4587	285	56	540	16	106	295	97	1014	4	0
Impervious Surface	681	4753	207	389	96	310	24	0	448	92	0
Vehicle	3	406	4795	832	373	503	3	12	0	18	55
Urban Furniture	154	76	716	3736	143	963	493	235	7	432	45
Roof	123	157	36	175	6105	245	37	28	20	15	59
Facade	72	129	312	510	144	5550	99	71	1	37	75
Shrub	126	19	50	1714	36	460	3995	580	3	14	3
Tree	63	0	10	178	14	36	709	5984	5	1	0
Soil/Gravel	2672	984	22	289	0	21	131	0	2881	0	0
Vertical Surface	0	113	613	149	0	620	0	0	0	5505	0
Chimney	1	0	8	469	152	628	50	63	0	63	5566

Figure 5 – Confusion Matrix from first test (Table 1). It shows that ground-related classes are miss classified. Other spatially independent classes like the chimneys (class 10) are very well classified.

4.2.2 Second Approach : generating the context cloud

To build a context cloud we decided to implement a "3-class" strategy. The cloud will be composed of 3 classes : Roof (class 4), Ground (class 0, 1, 8), Others. We first operated with a "known" context, meaning we built CTX from the labelled data. We later trained a model to build a context cloud from a rasterized sumbsampled cloud (0.5cm rasterization, 10 cm spatial subsampling). The rasterization (scening) was done according to the Z axis (less effective than a Numerical Terrain Mapping), retrieving the lowest point in a regular grid of 0.5 by 0.5 m, leading to a mapping of the roofs, the grounds and other horizontal surfaces (Fig. 6). The subsampling reduced the number of points for practical reasons. To favor the generalization of the model we decided to combine the training and validation set (Fig 4.) in order to have a more diverse dataset. We used 20% and 80% of the dataset for the validation phase and the training phase respectively.

After going through the inference phase, our context cloud was generated. Being a context cloud it needs to be reliable, as a result, we set a confidence threshold on the cloud in a way that the context cloud has enough points to be useful(189 955 points) and the classified points are reliable (70% confidence). This way we were able to :

— Build a 3 class model able to generate a CTX cloud for our training/validation and benchmark dataset.

— A context cloud on which important features will be computed in order for our model to better contextualize classes and their surroundings.

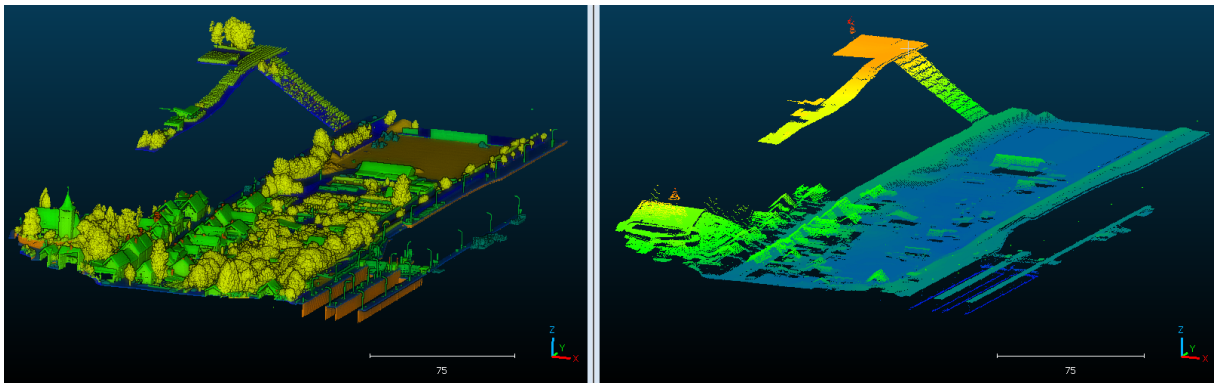

Figure 6 – On the left, the rasterized sample (PCX cloud for the CTX classifier). On the right, the subsampled version of PC1 for the training sessions.

4.2.3 Third Approach : Optimized features and scales + context cloud

The possible number of features with our setup is over 200. Each of these features having their own contribution to the model (positive or negative). They are calculated at 8 different scales from 0.5 m to 8 m and for KNNs with K in 1;3;5. In order for our model to be explainable and compact, it is important to select the features that are positively contributing to the model i.e. features that make our model more discriminant and efficient. In this section we explain how we proceeded to select those features, on CloudCompare and using Python.

4.2.4 On CloudCompare

A common problem in every classification problem is the preliminary class engineering to overcome "class imbalance". In our point clouds, many classes were over-represented (ground classes 15 + Mo points) and some had only few instances and samples (class C10 "Chimneys" 25 000 points). Even after balancing the classes at around 7000 samples per class, our model was biased by the density of certain classes in the point cloud. To tackle this problem we decided to de-densify accurate areas of the cloud and over-sample miss classified areas in order for our model to take into consideration miss classified points first (boosting). After doing so, we entered multiple training sessions and manually discarded features and scales based on their importance to the model, taking into consideration the inference time as well. We observed that the model didn't require more than 35 features (out of 200+ possible features) to achieve significant scores (Fig. 7). This allows the model to have low computational cost in contrast to some state-of-the-art deep learning methods.

4.2.5 In Python

Various Python scripts were implemented by Mathilde Letard for her thesis. These scripts propose a feature selection routine (Dash and Liu, 1997) to improve the explainability and the complexity of the trained algorithm. Although information redundancy supposedly does not impact RF performances, it disrupts the explainability of the feature importance values, since if two features bring similar information, their relative importance will be underrepresented. The scripts keep only a set of uncorrelated features, by using a bivariate feature selection (Dash and Liu, 1997; Guyon and Elisseeff, 2003), incorporating an assessment of the features such as Information Gain (IG) (Dash and Liu, 1997) and the Pearson linear correlation coefficient of attribute pairs. The correlation threshold and the scale at which each feature is evaluated are user-defined, and determined after an empirical investigation. The same bivariate procedure allows the selection of scales. After executing the feature selection, the scripts decided to promote small scales to limit the computation cost of the classifier. The selection process relies on a majority voting procedure. Since it is impossible to consider a scale independently from its application

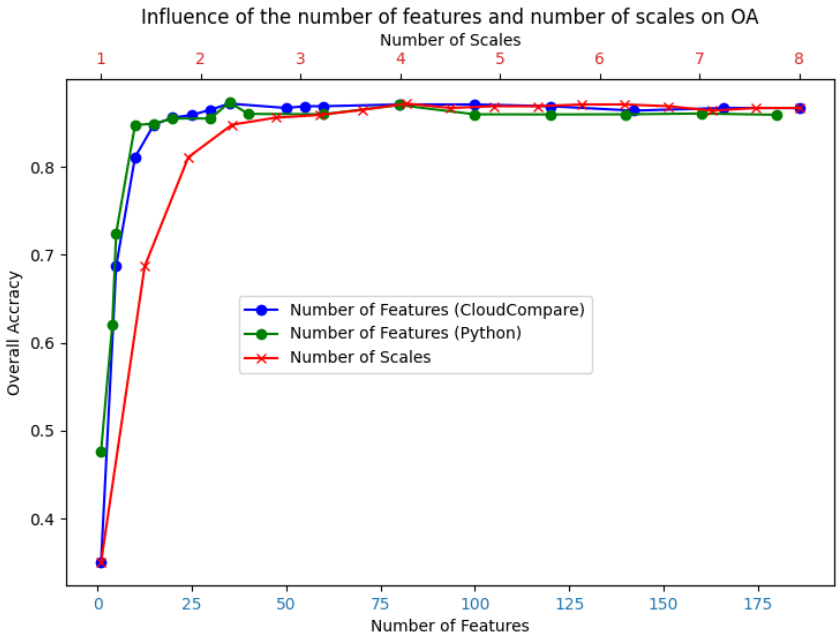

Figure 7 – OA vs Number of Features / Number of Scales. The model doesn't need a lot of features to perform well and not a lot of scales. The final number of features selected is 35, the final number of scales is 6.

to a feature, they retain the scales that are the most often selected when they are evaluated for each feature independently. If we only consider the correlated features, only few will be discarded leaving the model still computationally demanding and a final predictor vector of around 100 features, making the model harder to explain.

To further reduce the dimension of the predictor vector, Mrs Letards work considered a feature ranking depending on the IG. However, defining a fixed number of features and scales is highly task- and site-dependent, and filter-based selection would not consider internal synergies between features. Consequently, they use an embedded backward feature selection, relying on the RF feature importance, as detailed in (Aggarwal, 2014; Dash and Liu, 1997). This selection is performed on the uncorrelated set previously obtained. The optimal predictor vector is then identified through automatic OA monitoring, using a sliding window and keeping the last best iteration before OAs start to drop (methodology as described in Letard et al. 2023 sub.).

5 Results on Hessigheim Dataset

This section first presents the overall classification results obtained in the urban environments and the impact of feature preselection and optimization on classification. All results presented are obtained on a validation dataset strictly different than the training dataset. The model will be deployed on the test dataset (also strictly different from the two previous datasets) and the results will be sent to the IFP institute (creators of Hessigheim 3D) to compare classified points with officiel labels.

The starting set of features contains 200+ features, which include all possible features of 3DMASC computed on PC1. To determine the scale to use for feature evaluation – i.e. IG assessment – we analyzed the OA obtained when selecting features based on their IG at scales varying from 0.5 m to 8 m. Figure 7 shows that the optimal number of scales is 6 in terms of best OA.

Figure 8 show that multi-echo characteristics (multi-echo measurement such as Intensity of EchoRatio can provide information about object contours and semi-transparent surfaces which can be used to better

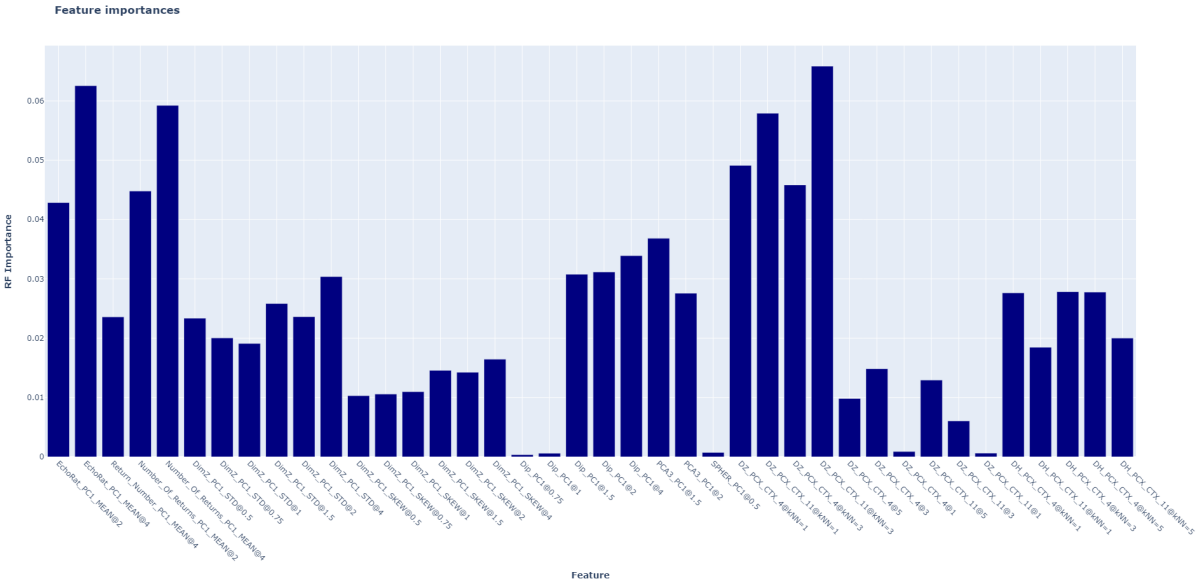

Figure 8 – Histogram showing the importance of each features in the model. The more a feature will contribute in improving the model by improving its classification ability, the more it will be important. The contextual features denoted by $D\{H,Z\}\_PCX\_CTX\_\{Class\}@\{KNN\} = \{K\}$ with $K \in \{1,3,5\}$

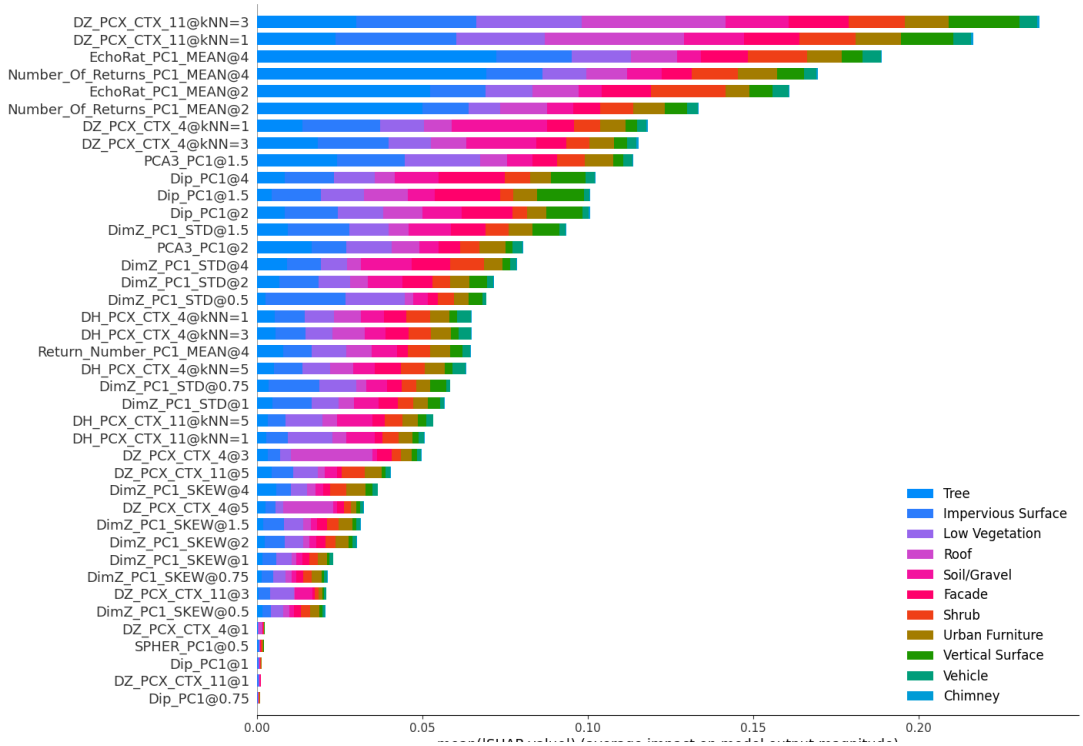

Figure 9 – Shap plot showing the contribution of each feature for each class.

identify and locate objects) and context-based features are predominant in the feature selection. In opposition, geometry-based features are a minority. Figure 9 allows us to see which feature contributed the most in the classification of each classes. The first two features $DZ\_PCX\_CTX\_11@KNN = 3$ and $DZ\_PCX\_CTX\_11@KNN = 1$ are context-based features, and contribute to the classification of trees, impervious surfaces, low vegetation and roofs. The use of a previously classified ground PC as contextual feature allows to improve the classification of points at the limit between ground and above

ground features, namely house walls and lower tree branches, explaining the improvement observed when they are included. Point-based and multi-echo characteristics (3rd feature in Fig. 9) allow to discriminate the nature of different structures, having an important impact on the classification of trees and classes that last send a return echo. The use of statistical operators seems particularly informative and able to decuplate the informative power of point-based characteristics, in particular multi-echo attributes. These features help the model to consider spatial relationships between points since it's not an inherent ability of the random forest.

Class	Precision	Recall	F1-Score
Low Vegetation	0.76	0.77	0.77
Impervious Surface	0.83	0.82	0.82
Vehicle	0.96	0.97	0.97
Urban Furniture	0.77	0.72	0.74
Roof	0.95	0.97	0.96
Facade	0.83	0.86	0.84
Shrub	0.74	0.75	0.75
Tree	0.94	0.85	0.91
Soil/Gravel	0.91	0.41	0.92
Vertical Surface	0.94	0.96	0.95
Chimney	0.98	0.99	0.99

Table 2 – Optimized Model : 7000 points per class, with context cloud. OA : 0.88

The confusion matrix 2 associated to our last training session with our optimized model, shows important improvements in ground class classification and urban furniture classification.Contextual-based features constitute the majority of the final predictor. The rest of the predictor is composed of point-based features.

	Low Vegetation	Impervious Surfa	Vehicle	Urban Furniture	Roof	Facade	Shrub	Tree	Soil/Gravel	Vertical Surface	Chimney
Low Vegetation	2 114	224	6	64	1	26	88	15	165	28	0
Impervious Surface	302	2 325	6	41	3	51	18	1	68	23	0
Vehicle	5	14	2 798	18	0	17	19	3	0	2	0
Urban Furniture	79	66	44	2 006	37	175	280	28	12	52	5
Roof	0	0	0	12	2 732	47	3	5	0	1	21
Facade	25	77	15	129	65	2 318	53	2	0	10	15
Shrub	93	33	33	252	18	80	2 082	115	14	43	0
Tree	63	0	5	39	6	25	196	2 557	2	15	3
Soil/Gravel	99	39	0	28	0	11	20	0	2 642	0	0
Vertical Surface	6	30	2	15	0	26	39	2	2	2 592	0
Chimney	0	0	0	0	8	4	0	0	0	0	2 802

Figure 10 – Confusion matrix of the training session of our optimized model. The average accuracy is 0.88, the scores have improved for each class.

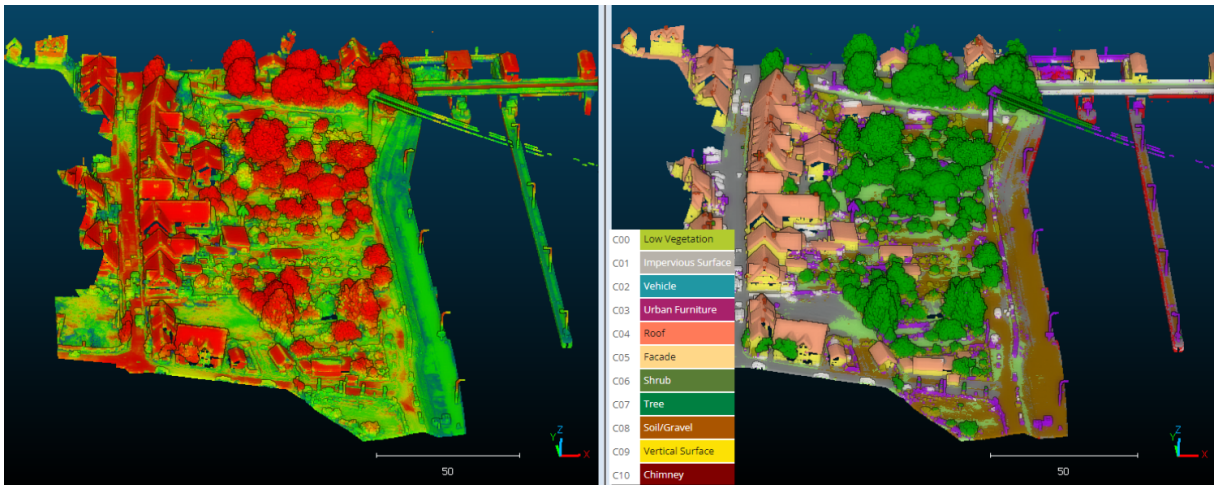

Figure 11 – Final Classification of the Benchmarked Dataset. On the right : classified point cloud. On the left : confidence of classification (green to red = low to high confidence). Total points classified : 51 Million. Total inference time : 27 hours running on 70+ CPUs.

After submitting the results to the IFP institute, the creators of the H3D dataset, we obtained further details on the results of the classification detailed in Figure 11. 3DMASC achieved an overall accuracy of 66.7, placing itself 13th out of 15 (OA ranking) on the benchmark leaderboard. The table 3 shows a few other methods applied to the dataset.

Participant	mF1	OA
ifp-RF	74.85	87.43
ifp-SCN	76.67	88.42
Gao-KPCONV	72.90	87.69
Gao-PN++	41.20	68.50
Letard(ours)	59.13	66.70

Table 3 – Extract of the results of benchmarked methods on the H3D Dataset.

6 Discussion on Hessigheim Dataset

The previously introduced method and results show an optimized and compact model of at most 35 predictors computed on 6 scales and resulting in a good and reliable training results. However, this model did not perform as well as expected when applied to the benchmark test set. This section aims to discuss those results taking into consideration the existing work on PC classification with contextual, point and neighbourhood-based features.

6.1 Classifier characteristics

Using 3DMASC, we were able to achieve significant scores of urban scenes classification with OAs up to 88% (during training phase), using a lightweight classifier trained on 7000 points per class and using 35 features. However, during the inference of the test set for the benchmark competition, the model performed poorly achieving an OA of 66% (close to the scores we had on non-optimized models). The metrics provided by the IFP institute after evaluating our classification (Fig. 12) show a high rate of miss-classification of the "Soil/Gravel" class with classes like "Impervious surface" and "Low Vegetation". These 3 classes represent the "Ground" class of our context cloud whose features showed themselves very important during the training session representing 4 out of the 8 most important features for our

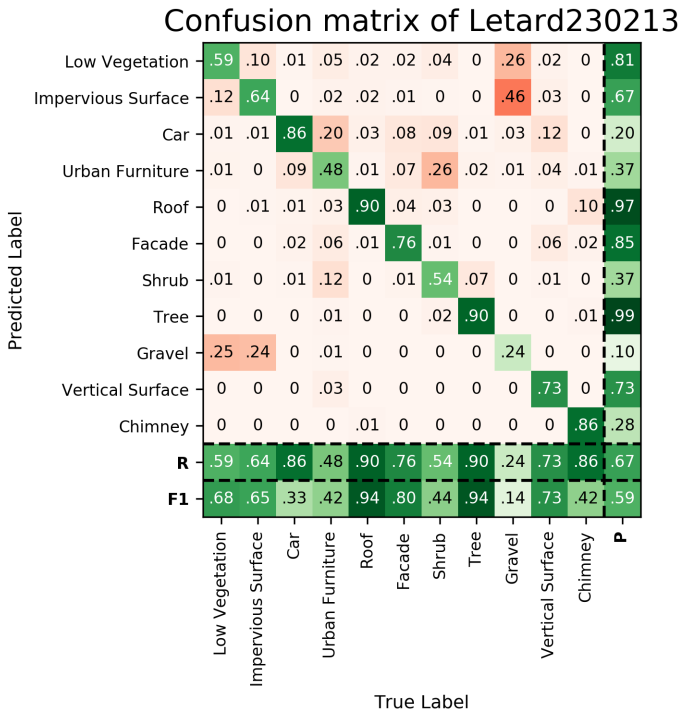

Figure 12 – Confusion matrix of the inference detailed on Fig 11. The matrix was computed by the IFP institute.

classifier(Fig. 8). The confusion between these classes could be due to the fact that the "Intensity" feature, commonly used to differentiate the nature of surfaces, was not used in the final predictor. In fact the "Intensity" feature was very quickly discarded by the feature selection protocol. This could also be due to the confidence threshold set on the CTX cloud, which was too high, leading "Ground" classes to be less represented or simply that the contextual features made the model too specific and did not allow it to properly discriminate ground classes. It illustrates that there is necessarily a balance between the importance of a feature and its inherent usefulness to the model and to the environment in which the model is performing.

Another reason for the poor performance of the model, could be the labelling inconsistencies of the training dataset being corrected in the test dataset, hence the miss-classification of certain classes (ground related classes see Fig 13). The two datasets being strictly different, a learning mistake made during the training will penalize the model during the test inference. In addition to this, the fact that the training set was boosted, by over-sampling and sub-sampling areas of points, might have influenced the robustness of the model when applied to areas in the test set where those particular classes were sub-sampled or over-sampled respectively. Although our model did not perform as well as the other methods such as Gao-KPCONV (Thomas et al. 2019), it did not require a lot of samples to train on (7000 samples per class) and still achieved scores close to certain deep-learning methods, requiring much more samples to be trained on, like Gao-PN++ (Qi et al. 2017). Training 3DMASC with more than 7000 samples per classes (eg. 30 000 samples per classes) has proven to make a difference in classification quality. By training a model with 30 000 samples per class we achieved 92% OA after the training phase and the quality of the classification on the benchmarked dataset was much better (Fig. 14). Additionally, the Intensity feature was left in the final predictor even if it wasn't showed as an important feature, this way the model was forced to take into consideration the intensity of each point. It shows that the Intensity feature was indeed important to differentiate the ground classes. Unfortunately, we did not have time to start a new inference before the end of the internship and to submit the new result.

Compared to standard deep-learning methods, the model remains a lot more explainable than neural-

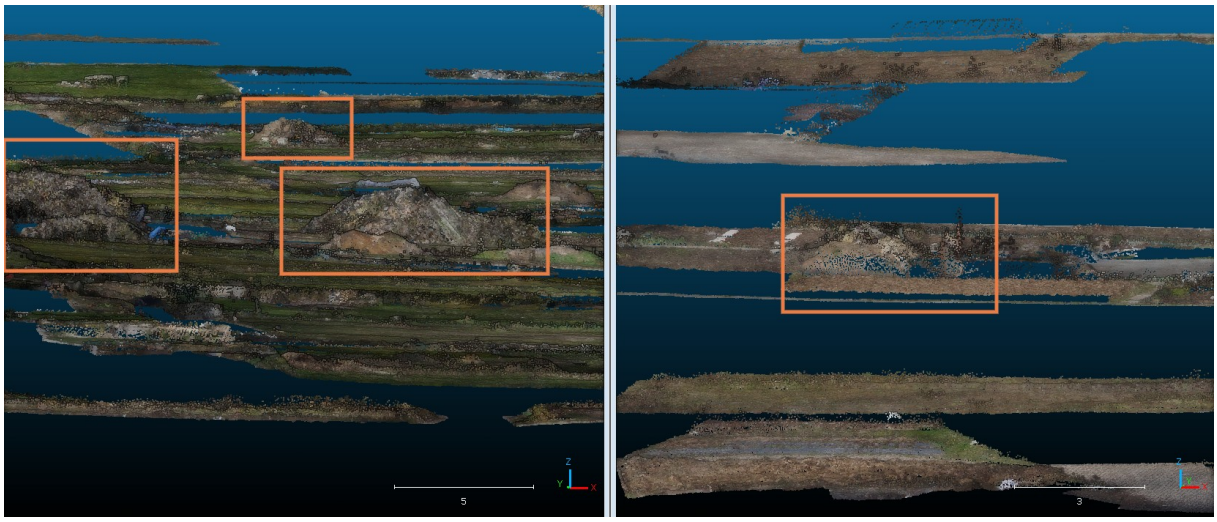

Figure 13 – RGB visualization for the training set. On the left class "Low Vegetation". "Low" is clearly questionable and the small vegetation hills should be labelled as "Shrub". On the right is class "Soil/Gravel", the quality of the labelling can be questioned, there are points that obviously do not correspond to gravel or soil.

network based methods. In fact, we were able to fully explain the model with features and their corresponding importance (Fig. 8) and Shapley values (measures a feature's average effect on a model's prediction, the average marginal contribution of a feature value across all the possible combinations of features.) (Fig. 9), showing how relevant contextual features are. The best model for the benchmark competition is IFP's random forest classifier. The gap between the result of two similar techniques (ours and theirs) is quite surprising, it could be due to the fact that they used features we didn't use (they did not describe their methodology in their paper) and different neighbourhoods that we didn't think of. They might've used more sample to train their model on. The second best model for the benchmark competition is also IFP's work, it is a SCN classifier (deep-learning model).

6.2 Dominant Scales

The method's explainability allows us to identify standard characteristics of OMS classification. In first place, a set of scales emerges from the feature and scale optimization technique (Letard et al. 2023, sub). These scales are rather grouped into a same range (0.5m to 4m), the environment being adequate to this kind of scales. The introduction of larger scales could be interesting with a parallel introduction of PC2 : a subsampled version of our original point cloud on which large scale features could be computed and then combined with the small scale features. This could allow to model to have a better understanding of the scenery and still be rather efficient in feature computing. Smaller scales could also be introduced to favor the discrimination between ground classes and tackle the miss-classification of the "Urban Furniture" and "Shrub" classes (Fig. 12).

6.3 Dominant Features

As previously mentionned, the final predictor is composed of 35 features in total, each of them of different nature. The features are : *Point-based features (Echorat, Number of Returns), Contextual features (horizontal and vertical distance to the center of the spherical neighbourhood of the context point), Neighbourhood-based features (PCA3, Dip, Point Height)* (see Appendix C). However, classical features of 3D Data interpretation allowing the model to delineate shapes of local PCs (Brodu and Lague, 2012; Gross and Thoennessen, 2006; Vandapel et al., 2004; Weinmann et al., 2013) are almost unused. This is certainly linked to the fact we analyze airborne lidar data, while these features were designed in

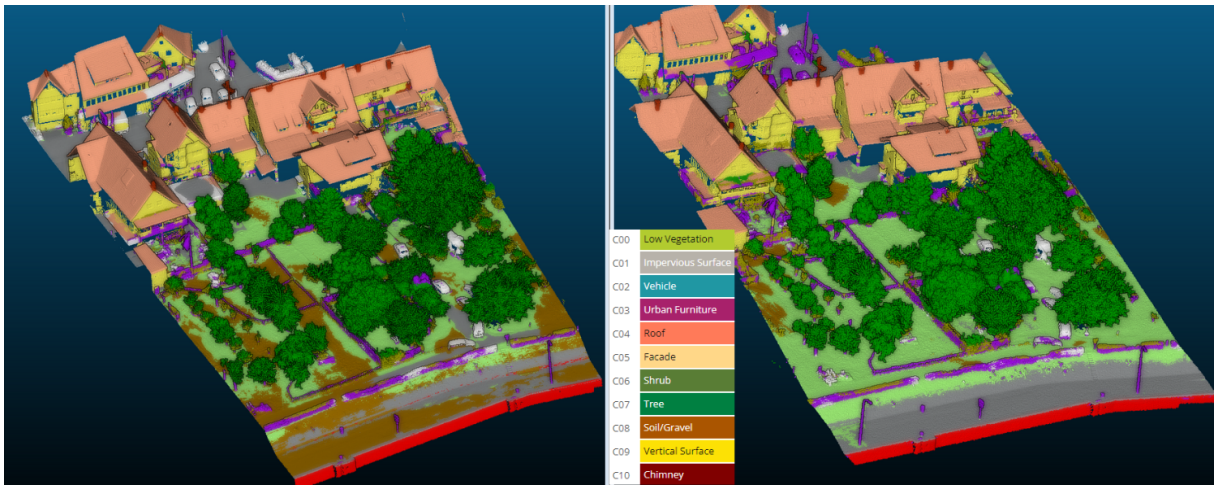

Figure 14 – Classification on an extract of the benchmark dataset. On the left : classification with model trained on 7000 samples per class, 35 features, 6 scales. On the right : classification with model trained on 30 000 samples per class, 35 features, 6 scales. There is less confusion between class "Soil and Gravel" (brown) and "Low Vegetation" (light green).

priority to describe terrestrial and mobile laser scanning, with more surface orientations. Another feature that was not used at all during the study because of the setup is the color of each point (RGB). Due to the complexity and height variation of the scene, a lot of shadowing was present, which made RGB features non representative of the point they were associated to. Surely LiDAR data classification of rather plane surfaces would use RGB feature because they represent a good indicator of belonging to a certain class nearby.

6.4 Task Driven Application for 3DMASC

3DMASC offers new perspectives on 3D data classification : new features, indicators, methodology and characteristics. All the former accessible and understandable to non-specialists. 3DMASC also offers the opportunity for lightweight ML models to be introduced on unmanned aerial vehicles, or on-board systems without important hardware capacity. In addition to this, the dual-cloud features could be used in point cloud time series analysis, to detect changes at different scales. Finally, the previous subsections show that with more data to train on, 3DMASC competes with state of the art methods and can be generalized to multiple environments.

7 Conclusion

The work undertaken during this internship has proposed multiple things. First, the evaluation of a newly developed 3D point clouds classification framework : 3DMASC, able to predict a label for each point taking into consideration the importance of each feature in the predictor. 3DMASC stands out from other 3D point clouds classification methods with his explainability and compactness. However, in terms of performance on a benchmarked dataset, 3DMASC has proven itself quite mediocre. The previous results allow us to draw the following conclusions : (i) the number of features the model is trained on is important (few samples will lead the model to miss-classify related classes), (ii) the feature selection protocol may not be the best selection for a given environment (the inherent usefulness of a feature is always to be considered even after the feature selection), (iii) Context-based features represent an important part of the final predictor and are a good way to allow the model to spatially contextualize objects and class instances, neighbourhood-based features allow the model to delineate shapes and point-based features allow the model to grasp the nature of certain surfaces allowing it to be more discriminant. Overall

3DMASC has a lot of potential thanks to its explainability and compactness. Further studies on urban environments should be explored and undertaken to fully harness its potential.

8 Additional Work

The initial objective of the internship was to take part in 2 benchmark competition. However the Hessigheim Dataset study took more time than anticipated but allowed me to provide a more complete analysis of the study. The other benchmark competition was supposed to be on the SimKITTI64 dataset (detailed in later section). The goal was to use the dual cloud feature computing ability of 3DMASC on terrestrial mobile LiDAR data. It would've provided insight on how features and scales are selected given the environment of the study and how 3DMASC is able to perform for autonomous driving classification.

8.1 The SimKITTI64 Dataset

This dataset was chosen to assess the performance of 3DMASC in a context of autonomous driving using LiDAR mapping. SimKITTI64 is a dataset created by simulating a Velodyne HDL-64 inside a scene modeled from the SemanticKITTI dataset (acquired using a Velodyne HDL-64 on top of a car). The goal from the dataset is to test sensor domain generalization of 3D semantic segmentation methods in the same environment, which tests if a method designed to semantically segment point clouds acquired using one sensor, e.g., a Velodyne HDL-64, can generalize to point clouds acquired using another sensor, e.g., a Velodyne HDL-32. An interesting point with this simulated dataset is that it simulates semantic data up to 80m whereas in SemanticKITTI (Fig 3) the data is annotated up to 50m. It simulates the HDL-32 sensor inside the modeled scene of sequence 08 in the SemanticKITTI dataset. This sequence was chosen because it is used as validation sequence for 3D semantic segmentation methods. All point cloud files are in ply format.

The dataset is manually labelled and there are 34 classes (8 moving object classes and 26 static classes). The particularity of the dataset is that the labelling takes into consideration instance classification (moving and non moving objects, eg. class 10: "car", class 252: "moving-car" etc.). This way, we plan on exploiting the dual cloud features computing with 3DMASC and assess its proficiency in ground urban areas classification and instance segmentation.

Each sequence of the dataset is composed of multiple attributes :

- (X, Y, Z) - the coordinates of a point.
- intensity - point based attribute, intensity of the beam during the sampling.
- semantic - the label of a point.
- instance - moving or non moving instance of an object.

During the sampling, the odometry was not recorded, this causes the frames to have different referentials making it impossible to correctly align the frames like in Figure 15. We developed a python script reading ply files and poses.txt, a file containing the transformation matrix for the file in order to adjust them to the original trajectory. Once the point clouds were transformed we were able to build different datasets to train our model using the 3DMASC plugin.

8.2 Other participation to the project

Throughout the internship, I helped debug the 3DMASC plugin for CloudCompare and give insights to possible improvements and new functionalities. I also helped generalize Mrs Letard's python scripts in order to make them more accessible to different setups.

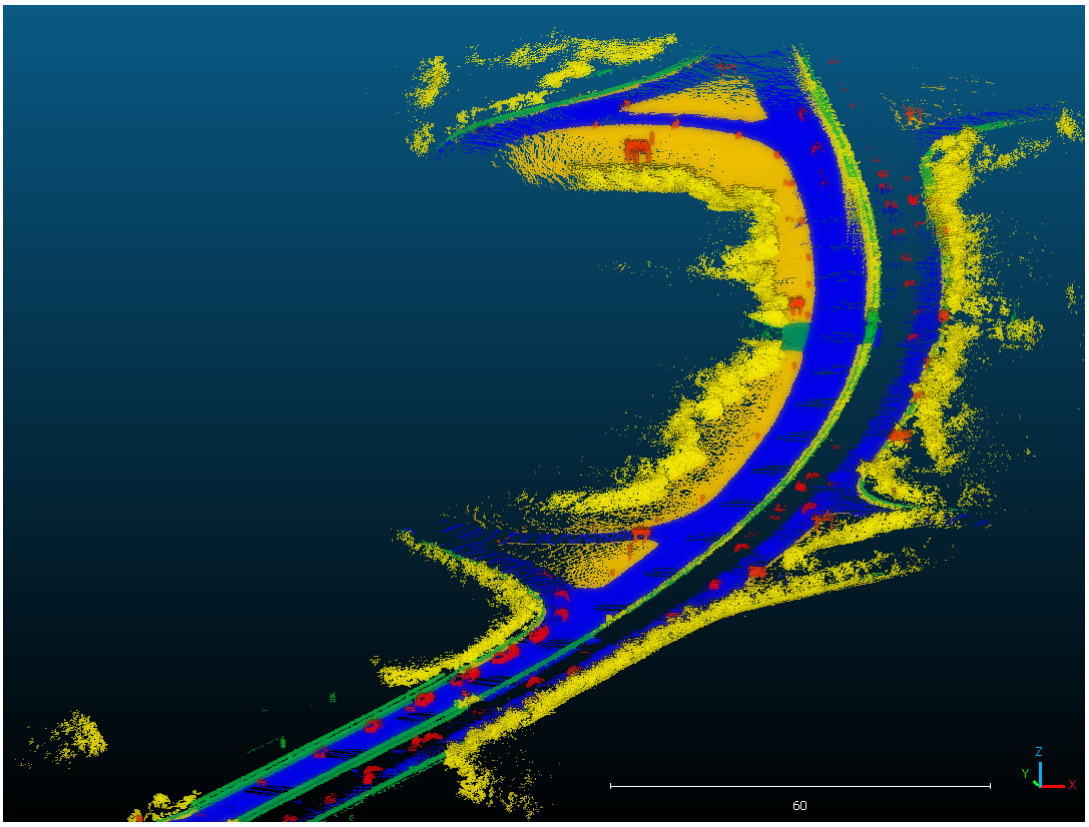

Figure 15 – Transformed SimKitti64 dataset in order to retrieve the original trajectory of the SemanticKitti Dataset. Each point color corresponds to a different class

9 Acknowledgements

During this internship I've had the opportunity to be overlooked by two great tutors : Mathilde Letard and Paul Leroy. Their knowledge and expertise on the subject have allowed me to improve in many ways and fulfill the mission I was given. This internship was also a great opportunity for me to build experience in the field of "Objective Evaluation" and application to artificial intelligence. I've gained new abilities and have familiarized myself even more with the field of research and its methodology. Finally I would like to thank the UTC for allowing me to undertake an internship outside their facility and give me the opportunity to take part into a project that I became passionate about.

10 References

Mathilde Letard, Dimitri Lague, Arthur Le Guennec, Sébastien Lefevre, Baptiste Feldmann, Paul Leroy, Daniel Girardeau-Montaut and Thomas Corpetti. Univ Rennes, Geosciences Rennes, UMR 6118 CNRS, France, 2023, submitted.

Kölle, M., Laupheimer, D., Schmohl, S., Haala, N., Rottensteiner, F., Wegner, J. D., Ledoux, H. (2021a). The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo. ISPRS Open Journal of Photogrammetry and Remote Sensing, 1, 11. doi:10.1016/j.ophoto.2021.100001

G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kontschieder. The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes. In International Conference on Computer Vision (ICCV), 2017.

Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Gall, J., Stachniss, C. (2021). Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI Dataset. The International Journal on Robotics Research, 40(8–9), 959–967. doi:10.1177/02783649211006735

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., Pollefeys, M. (2017). Semantic3d.net: A new large-scale point cloud classification benchmark. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-1-W1, 91–98.

Li, Y., Ibanez-Guzman, J. (2020). Lidar for Autonomous Driving: The Principles, Challenges, and Trends for Automotive Lidar and Perception Systems. IEEE Signal Processing Magazine, 37(4), 50–61. doi:10.1109/MSP.2020.2973615

Landrieu, L., Simonovsky, M. (2018). Large-scale point cloud semantic segmentation with superpoint graphs. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4558–4567.

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L. J. (2019). KPConv: Flexible and Deformable Convolution for Point Clouds. doi:10.48550/ARXIV.1904.08889

Qi, C. R., Yi, L., Su, H., Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. doi:10.48550/ARXIV.1706.02413

Richa, J. P., Deschaud, J.-E., Goulette, F., Dalmasso, N. (2022). AdaSplats: Adaptive Splatting of Point Clouds for Accurate 3D Modeling and Real-Time High-Fidelity LiDAR Simulation. Remote Sensing, 14(24). doi:10.3390/rs14246262

Kölle, M., Laupheimer, D., Schmohl, S., Haala, N., Rottensteiner, F., Wegner, J. D., Ledoux, H. (2021b). The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo. ISPRS Open Journal of Photogrammetry and Remote Sensing, 1, 100001. doi:10.1016/j.ophoto.2021.100001

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

Zhirong Wu, Shuran Song, Aditya Khosla, Xiaoou Tang and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In CVPR, 2015. 1, 2

Tan Yu, Jingjing Meng, and Junsong Yuan. Multi-view harmonized bilinear network for 3d object recognition. In CVPR, 2018. 2

Daniel Maturana and Sebastian Scherer. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. In IROS, 2015. 2

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In Advances in Neural Information Processing Systems, 2017. 2, 3

Charles R. Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In CVPR, 2016. 2

Miguel Dominguez, Rohan Dhamdhere, Atir Petkar, Saloni Jain, Shagan Sah, and Raymond Ptucha. General-purpose deep point cloud feature extractor. In WACV, 2018. 2

Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In CVPR, 2018. 2

Haoxuan You, Yifan Feng, Rongrong Ji, and Yue Gao. Pvnet: Ajoint convolutional network of point cloud and multi-view for 3d shape recognition. In Proceedings of the ACM International Conference on Multimedia, 2018. 2

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds.

Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In ECCV, 2018. 2, 3, 5, 6, 7, 8

Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in Neural Information Processing Systems, 2017. 2, 3, 5, 6, 7, 8

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012, Stanford University —Princeton University — Toyota Technological Institute at Chicago, 2015. 2

Girardeau-Montaut, D., 2022. CloudCompare (version 2.12.4) [GPL software]. (2022). Retrieved from http://www.cloudcompare.org/.

Dash, M., Liu, H., 1997. Feature Selection for Classification. IDA ELSEVIER Intelligent Data Analysis 1, 131–156

Y. Zhang and M. Rabbat, "A Graph-CNN for 3D Point Cloud Classification," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 6279-6283, doi: 10.1109/ICASSP.2018.8462291.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

A Appendix

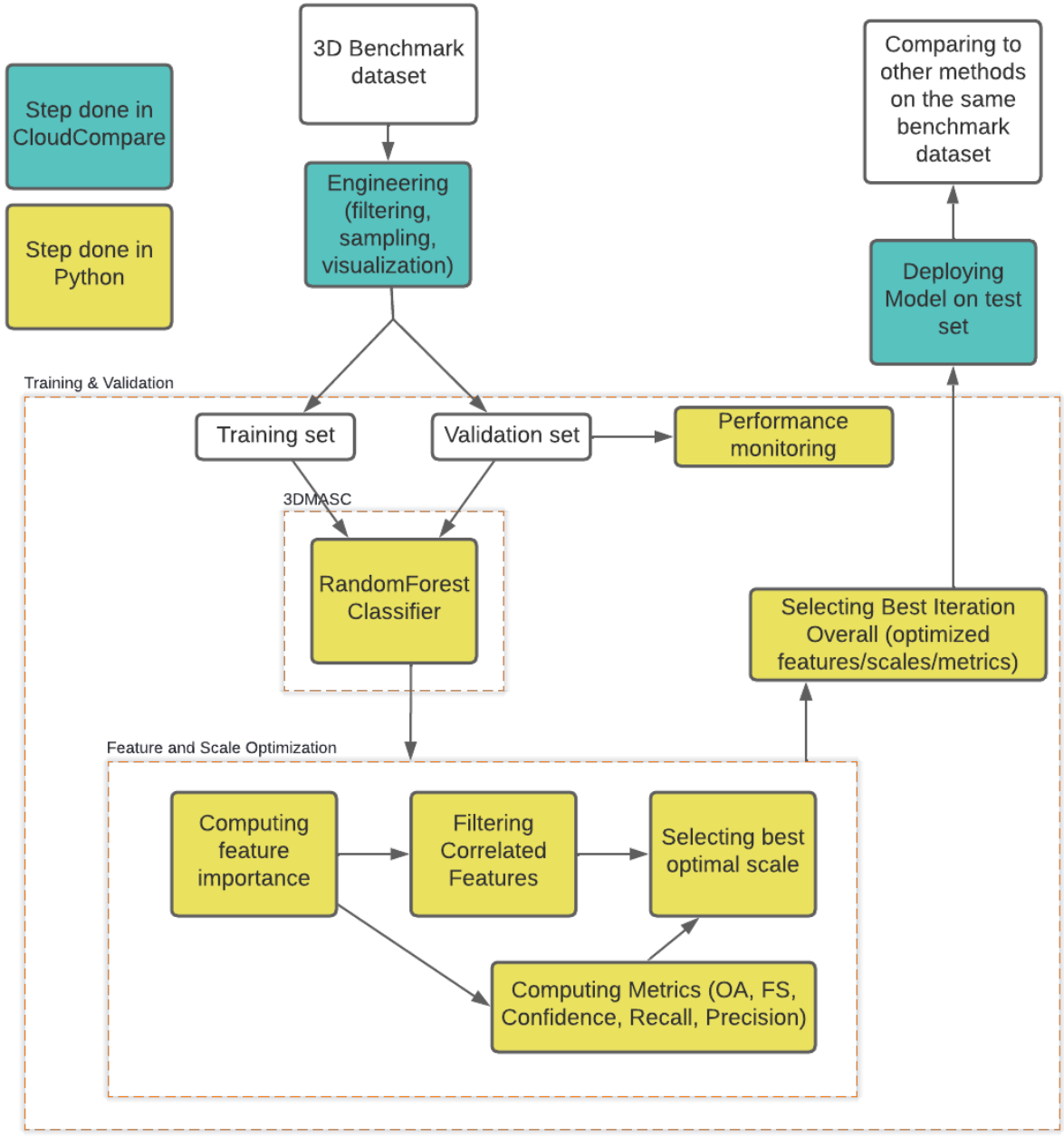

Figure 16 – Methodology process used during the study

B Appendix

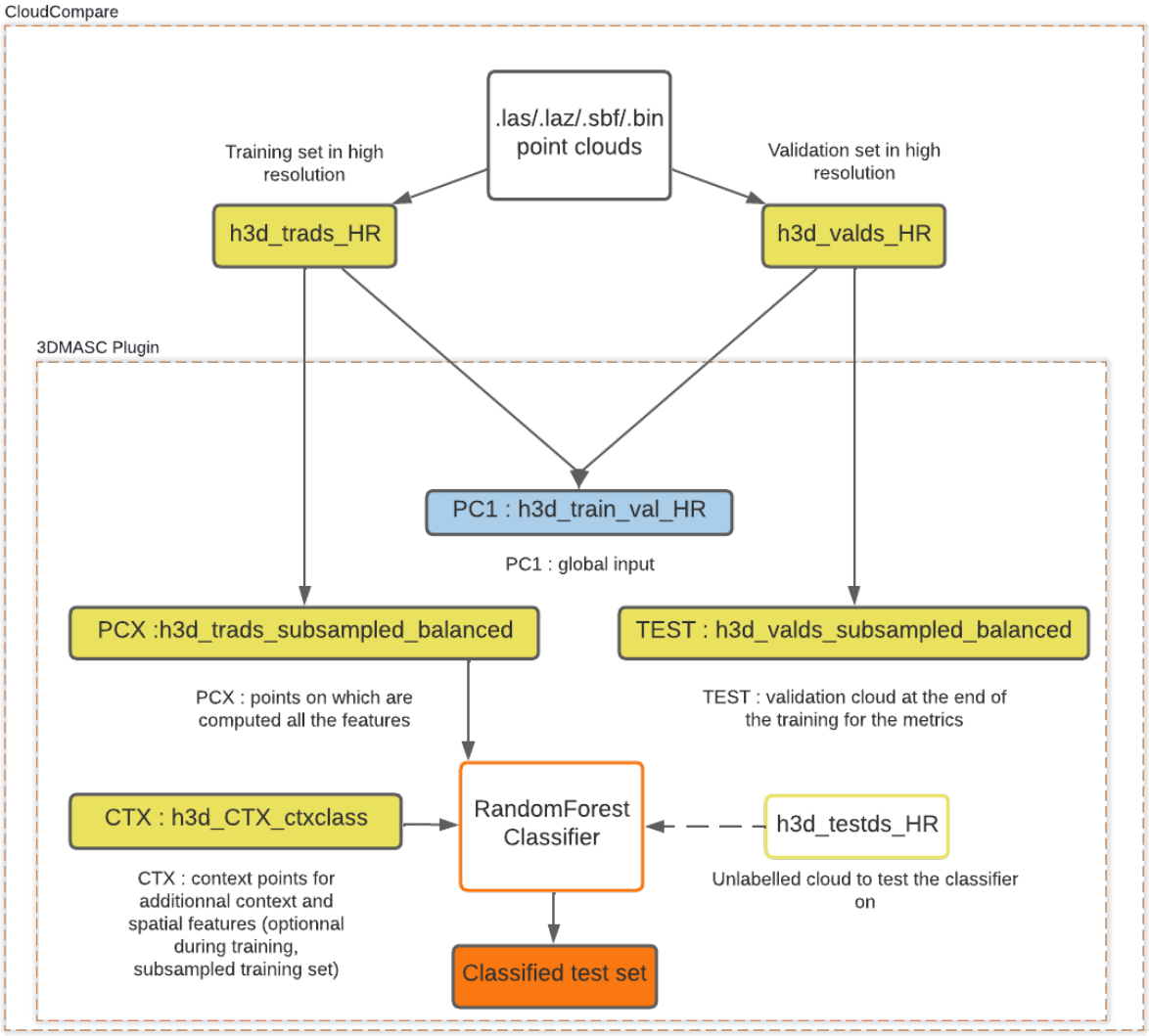

Figure 17 – 3DMASC plugin functional representation

C Appendix

Name	**Description**	**Nature of feature**
Echo Ratio	Measure for local transparency and roughness	Point-based
Number of Returns	Total number of returns for a given pulse	Point-based
DH to KNN	Mean horizontal distance to k nearest neighbor	Contextual Feature
DZ to KNN	Mean vertical distance to k nearest neighbor	Contextual Feature
PCA3	3 component of principal component analysis in a neighbourhood	Neighbourhood-based
Dip	Orientation of the beam on a surface	Neighbourhood-based
DimZ	Height of the point	Neighbourhood-based

Table 4 – Name, description and nature of the features used in the optimized predictor.