CLASSIFICATION OF AERIAL LASER SCANNING POINT CLOUDS USING MACHINE LEARNING: A COMPARISON BETWEEN RANDOM FOREST AND TENSORFLOW

F. Pirotti [1,2*] , F. Tonion[1,2]

[1] CIRGEO - Interdepartmental Research Center of Geomatics, University of Padova, Viale dell'Università 16, Legnaro (PD), Italy - francesco.pirotti@unipd.it
[2] TESAF Department, University of Padova, Viale dell'Università 16, Legnaro (PD), Italy

Commission II, WG II/3

KEY WORDS: Deep Learning, Classification, Tensorflow, Laser Scanning, Airborne Lidar, Random Forest

ABSTRACT:

In this investigation a comparison between two machine learning (ML) models for semantic classification of an aerial laser scanner point cloud is presented. One model is Random Forest (RF), the other is a multi-layer neural network, TensorFlow (TF). Accuracy results were compared over a growing set of training data, using a stratified independent sampling over classes from 5% to 50% of the total dataset. Results show RF to have average F1=0.823 for the 9 classes considered, whereas TF had average F1=0.450. F1 values where higher for RF than TF, due to complexity in the determination of a suitable composition of the hidden layers of the neural network in TF, and this can likely be improved to reach higher accuracy values. Further study in this sense is planned.

1. INTRODUCTION

Machine Learning (ML) is the branch of Artificial Intelligence (AI) that concerns the automated detection of meaningful patterns in data (Shalev-Shwartz and Ben-David, 2014). The ML approach to data analysis was born in the second half of 20th century, when mathematical theories (such as Least Square Method, Bayes theorem and Markov Chains) met the growth of modern informatics. Starting from the last decade ML techniques were adopted in complex data intensive field such as astronomy, biology, climatology, finance and economy (Is et al., 2015). Thanks to a large number of algorithms, today ML find application in a lot of fields related to ever day human life (internet, industrial production, medicine..ecc).

In general modern ML algorithms allow to use different kind of learning methods from data. In particular it is possible to distinguish the following kind of learning approaches:

- Unsupervised. When unknown patterns and structures are uncovered within the dataset (no a priori knowledge).
- Supervised. When it is defined a model capable to connect explanatory variables to the responsive one (a priori knowledge).
- Semi supervised learning. When there is no training dataset and model is refined basing on the experience acquired.

The use of the supervised approach is common for classification problems. Two well known and common classifiers are Random Forest (RF) and Tensorflow (TF); these algorithms are non parametric and provide excellent results with multi modal data distribution.

The RF model is based on an ensemble of decision trees (forest) that grows through training towards best combinations. In fact an ensemble consist of a set of individual trained classifier (decision trees), which are combined for classify new instances (Kulkarni, 2013). The RF model requires the definition of two parameters, that are the number of trees to generate (Ntree) and the number of variable (Mtry) to be selected and tested for the best split when growing trees (Belgiu and Drăgu, 2016).
The number of trees can be different and depends on the computational efficiency and the overfitting risk; for example the default number of trees value in R package "Random Forest" is 500. Once all parameters are defined the model builds all the trees. About variable selection the following approaches are used (for more info see the work by Abellán et al., 2017):

- Random Forest Using Random Input Selection (Random Forest- RI): It is the most common. In this approach m variables are selected at random out of the available attributes and the best split on these m is used. The number of attributes used in random selection by the author were 1 and the first integer less than log 2 (M) + 1 .
- Random Forest Using Linear Combination of Inputs (Random Forest RC): Before the selection of the best variable to split, more attributes are created by taking random linear combinations of L variables (using L = 3).

However during the building of the forest, the number of variables is held constant. Before running the model the dataset is divided into two part: a training set, which can reach two third of the initial dataset, and the other part that is the validation one (or Out Of Bag – OOB). During the training phase the dataset is run down by each tree. Each tree perform a classification of the features of dataset, which is counted as a vote. The final classification is assumed with the majority vote criterion. The OOB features are used for error estimation. Moreover RF algorithm can also compute Variable Importance (VI) and proximity, which are important for understanding the behaviour of the features in the model (Breiman, 2001).

The TF is a framework created by Google's artificial intelligence team and released in 2015 with an open source license. TF uses graph to represent both the computation in an algorithm and the state on which the algorithm operates (Abadi et al., 2016). TF is based on Convolutional Neural Network (CNN); this kind of approach is similar to Neural Network (NN) one, but it uses

convolutions and pooling to reduce the number of trainable parameters. Reducing parameters reduces computation cost and improves the ability to generalize (Hemmes, 2018).

A typical TF application is divided into the following two phases:
- Program definition. In this phase a CNN to be trained is defined and represented as graphical dataflow.
- Optimization. In the second phase the Neural Network defined is executed and optimized basing on the data available.

In general these classification techniques can found interesting application in point cloud classification. In fact Laser Scanning techniques allows to acquire spatial information as 3D point clouds. Before performing the classification of points cloud it is possible to do a "segmentation" of 3D points. This process groups points into different cluster, using different approaches, such as Edge Segmentation, Regional Growing Segmentation, Segmentation by Model Fitting, Hybrid Techniques Segmentation and Machine Learning Segmentation (Grilli et al., 2017).

The result of the Segmentation phase (the cluster) can be used as variable for points cloud classification phase. The classification process involves initially the features extraction phase. Initially this process involves the recovery of a local neighbourhood for each 3D point, and then the extraction of geometric features based on all 3D points within the local neighbourhood.
The radius of neighbourhood analysis can be different for shape (spherical, cylindrical..) and for length (fixed or variable) (Weinmann et al., 2015). Basing on these analysis, features are extracted. Finally the classification is performed using a classifier (RF, ANN, TF… etc). The multitude of applications for machine learning approaches for spatial data is shown in the recent numerous publications that focus on using these algorithms for image analysis (Pirotti et al., 2016) and also support raster-based spatial predictions (Piragnolo et al., 2019). In this work we compared point clouds classification accuracy of Random Forest and Tensorflow.

2. MATERIALS AND METHODS

2.1 Datasets

The data used for testing and validation in this investigation is the benchmark dataset from the ISPRS benchmark on urban object detection and 3D building reconstruction (Rottensteiner et al., 2014b). The labelled laser scanner dataset of the city of Vaihingen (Germany) was used. Points in this dataset are labelled with eight classes, indexed respectively from 0 to 8 as in figure 1. The number of points in the dataset used is not balanced (Table 1); as can be expected, power-lines, cars and fences are represented by ~100x less points.

Label	Class	N. points
0	Powerline	600
1	Low vegetation	98690
2	Impervious surfaces	101986
3	Car	3708
4	Fence/Hedge	7422
5	Roof	109048
6	Facade	11224
7	Shrub	24818
8	Tree	54226

Table 1. Number of points per class.

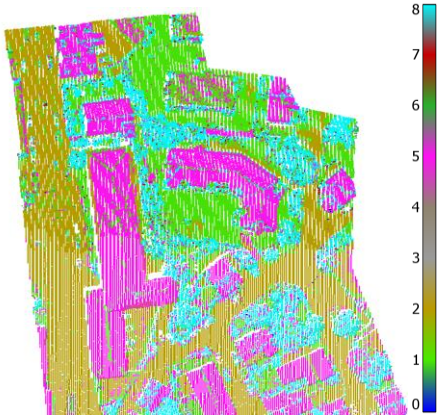

Figure 1. Colour legend for labelled classes in dataset.

2.2 Methods

The objective of this investigation is to test RF and TF methods over a set of observations (points) using a feature vector where features are predictors. Predictors here are derived not only from data present as point attribute (e.g. intensity of laser backscatter), but also from geometric characteristics extracted by analysis of neighbouring points. The set of vectors with features is then used as input to RF and TF.

2.2.1 Feature extraction. Feature vector with predictors are calculated by considering spatial context. Context has to be defined by evaluating a number of nearest neighbours (*nn*) that is large enough to represent a class, but also small enough to avoid including points that belong to other classes. Therefore the cardinality of *nn* is not fixed, but determined with a method that maximizes geometric consistency. This is done by selecting the number of *nn* which result in the lowest Shannon entropy index value, as in (Weinmann et al., 2015, 2017), calculated using normalized eigenvalues of the 3D covariance matrix:

$$min\{\sum_i^3 \lambda_i(nn) \cdot \ln[\lambda_i(nn)]\} \quad (1)$$

where nn_{min} = 15 and nn_{max} = 100 and λ_i are the normalized eigenvalues of the 3D tensor tensor matrix and i = {1,2,3}. Twenty-three predictors are extracted this way using 3D, 2D features. Similar work using these features has been done by (Thomas et al., 2018; Weinmann et al., 2015). Below the list and description of these predictors.

Feature	3D	2D	Feature	3D	2D
nnn	x	x	Verticality (V)	x	
radius	x	x	Omnivariance (O)	x	x
density	x	x	Anisotropy (A)	x	x
dz	x		eigenvaluesSum (ES)	x	x
stddevz	x		curvatureChange (CC)	x	
Linearity (L)	x	x	Eigenentropy (EE)	x	x
Planarity (P)	x		Dimensionality (D)	x	
Scattering (S)	x				

Table 2. List of features extracted from a context of nn nearest neighbours.

The value of *nnn* represents the number of nearest neighbours that can range from 15 to 100. The 3D structure tensor C_{nn}- 3D covariance matrix - is calculated from the *nn* points.

$$\mathrm{cov}(i,j)=\frac{\sum_{n=1}^{nn}\left(i_n-\bar{i}\right)\left(j_n-\bar{j}\right)}{N-1}$$
$$i=j=\{x,y,z\}$$
$$C_{nn}=\begin{bmatrix}\mathrm{cov}(x,x) & \mathrm{cov}(x,y) & \mathrm{cov}(x,z)\\ \mathrm{cov}(y,x) & \mathrm{cov}(y,y) & \mathrm{cov}(y,z)\\ \mathrm{cov}(z,x) & \mathrm{cov}(z,y) & \mathrm{cov}(z,z)\end{bmatrix} \quad (2)$$

$$C_{nn}\in\mathbb{R}^{3x3} \quad (3)$$

Radius and density are respectively the radius of the minimum-bounding sphere (3D) and circle (2D) of the point set. Dz and Stddevz are respectively range and standard deviation of height values (Z).

$$L_i=\frac{\lambda_1-\lambda_2}{\lambda_1}$$
$$P_i=\frac{\lambda_2-\lambda_3}{\lambda_1}$$
$$S_i=\frac{\lambda_3}{\lambda_1}$$
$$V_i=1-n_z$$
$$O_i=\sqrt[3]{\lambda_1\cdot\lambda_2\cdot\lambda_3}$$
$$A_i=\frac{\lambda_3-\lambda_1}{\lambda_1}$$
$$ES_i=\lambda_1+\lambda_2+\lambda_3$$
$$EE_i=-\sum_i^3\lambda_i\cdot\ln(\lambda_i)$$
$$CC_i=\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3} \quad (4)$$

Dimensionality: $D_i = - L_i \ln(L_i) - P_i \ln(P_i) - S_i \ln(S_i)$. (5)

Intensity of return pulse was added to the above 23 features providing a total of 24 descriptors.

2.2.2 Random forest (RF). Random forest (Breiman, 2001) is a widely used method for data mining in many disciplines, and has recently been often tested on classification of imagery (Piragnolo et al., 2017; Pirotti et al., 2016) and point clouds (Guo et al., 2011; Thomas et al., 2018; Weinmann et al., 2015).

The RF classifier is an ensemble of decision trees created through a bagging approach. The set of trees is referred to as a forest. Bagging implies selecting samples from the training subset and train the trees. Internal cross-validation technique, e.g. using Gini index, measures the performance of RF and selects the best ensemble. Two parameters require tuning for best results in RF: the number of trees (Nt) and the number of features (Nf). Iterative methods to find best Nt and Nf combinations were applied using K-fold cross-validation over a subset of 20% of the data and testing 3x3 combinations of Nf∈{4,8,12} and Nt ∈{50,100,200,500}. Best combination resulted in Nf=8 and Nt=200. Each node in a tree is split by randomly selecting Nf features from the d-dimensional input feature space. The splitting function in this case uses Gini index as a measure node purity. In the prediction step, class probability is voted by each tree – maximum probability determines the class given to the point.

2.2.3 TensorFlow (TF). The TF classifier is a convolutional neural network (CNN), thus uses a deep learning approach that can consider spatial context. A set of hidden layers process input vectors of predictors. The model is trained by feeding training data with multiple samples of each class. The model learns by going through all samples and updates node weights after each iteration. Best weight combination is defined with an accuracy metric calculated from training data. Predictor vectors are converted to tensors before being used as input in the neural network. Input is then passed sequentially to a number of hidden layers with user-defined number of nodes.

Hidden layers can be customized depending on the data. Many applications use TF over feature vectors represented by semantics (text pool) or reflectance values (images). Some implementations of TF for point clouds have been developed recently, such as PointNet (Qi et al., 2017).

2.2.4 Accuracy metrics. Accuracy assessment was carried out by providing fundamental accuracy metrics for each class. Precision, recall and the Jaccard index i.e. intersection of union and F1 overall index.

$$A=\begin{bmatrix}a_{11} & a_{12} & \dots & a_{1n}\\ a_{21} & \dots & \dots & a_{2n}\\ a_{n1} & a_{n2} & \dots & a_{nn}\end{bmatrix} \qquad TP_i=a_{ii}$$

$$FP_i=\sum_j^J a_{ji}-TP_i \qquad FN_i=\sum_j^J a_{ij}-TP_i$$
$$\mathrm{Pr}=\frac{TP}{TP+FP} \quad \mathrm{Re}=\frac{TP}{TP+FN}$$
$$Ji=\frac{TP}{FP+FN+TP} \quad F1=2\cdot\frac{\mathrm{Pr}\cdot\mathrm{Re}}{\mathrm{Pr}+\mathrm{Re}} \quad (6)$$

Where A is the error matrix, TP=true positives, FN=false negatives, FP=false negatives, Pr=precision, Re=recall, Ji=Jackard index, F1=F1 score.

The two machine learning models, TF and RF, are trained over a stratified sample of the total point cloud grouping by class. To assess accuracy over different sizes of training samples, 10 runs, using from 5% to 50% of the total number of points for training, where used.

All the functions used for applying the methods were implemented using modules in R-cran framework, with *rminer* and *Keras* package.

3. RESULTS AND DISCUSSION

In the next sections insights on neighbourhood size and features extracted are presented. Then accuracy of TF and RF are compared and the effect of training set size is evaluated. As stated in materials' section, the data used are from the city of Vaihingen, Germany, and belong to the ISPRS Benchmark on urban classification and 3D reconstruction. The benchmark ended in 2018 and results are available in the webpage and discussed in a special issue (Rottensteiner et al., 2014b). Results will be discussed comparing also results from that benchmark.

3.1 Neighbourhood size

Result of optimal neighbourhood size selection as described in (Weinmann et al., 2015) is described in Figure 2. The histogram distribution is very similar to results in (Weinmann et al., 2015).

Distribution of all 9 classes sees a bimodal distribution, with most values in the lower part, from 10 to 25, and the second mode in the higher part, between 90 to 100.

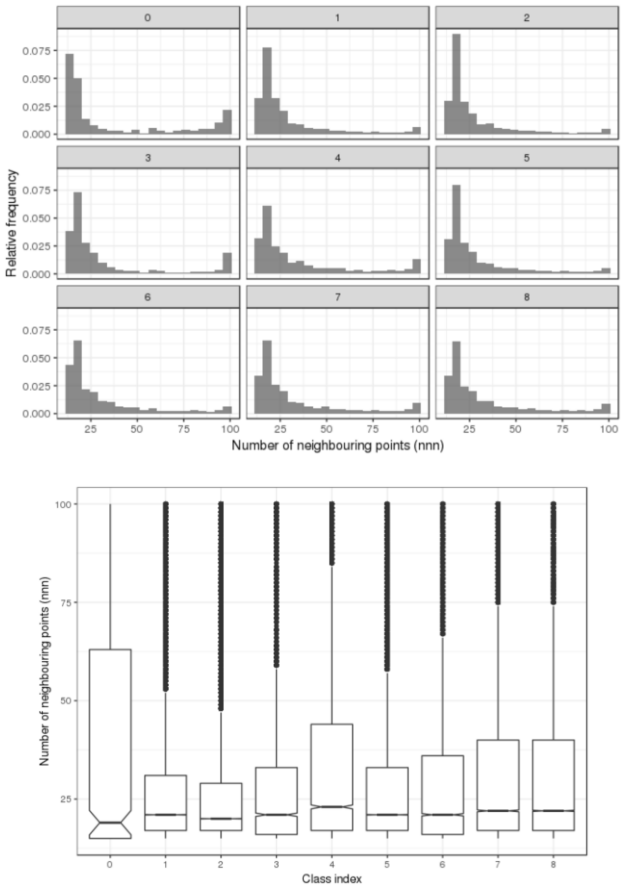

Figure 2. Frequency distribution (top) and box plot (bottom) of *nnn* values, found with minimum entropy, for each class (see Table 1 for class labels).

3.2 Feature vector of predictors

The features used, listed in Table 2 ideally have to provide significant information to separate required classes. The resulting frequency distribution of values for each feature can give an initial idea of the power that each feature has to separate classes. The training data provides this information and the distribution of values for three most important features is shown in Figure 3 , which are eigenentropy and two shape indices, linearity and planarity.

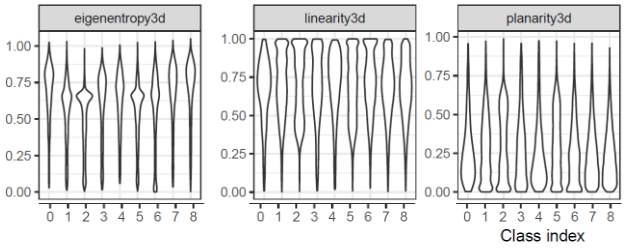

Figure 3. Frequency distribution of the three features that have most importance for both RF and TF methods (see Table 1 for class labels).

Finding ideal predictors is an important step; relevance of each feature is also defined in the training step of RF or TF, and is discussed in the next sections. Interest in segmentation and classification of irregular point clouds has gained much attention lately, resulting in many investigations. Features can be implicit, as point characteristics (e.g. intensity of reflected pulse, return number ratio), extracted from neighbours in the 3D context (e.g. index of linearity, planarity, scattering – see methods section) and also be assigned from segment-based characteristics (Vosselman, 2013). In this work only the first two types are used, but further work must include also segment-based features.

3.3 Results of RF classification

Table 3 shows commission and omission errors for best RF classification, with the four accuracy metrics reported in equation 6. Visual representation of misclassified points from RF is presented at top figure in Figure 7. The table provides insight on what classes are hard to unmix and which are more easily separated from the others.

	Powerline	**Low vegetation**	**Impervious surfaces**	**Car**	**Fence/ Hedge**	**Roof**	**Facade**	**Shrub**	**Tree**
0	**419**	7	1	1	2	23	9	11	127
1	6	**86728**	4720	25	77	2542	84	711	3797
2	0	3544	**97229**	45	13	780	11	53	311
3	0	580	123	**2316**	14	195	4	93	383
4	0	1140	64	9	**4588**	205	34	243	1139
5	3	4849	2216	10	20	**98685**	95	272	2898
6	7	608	82	6	25	267	**8383**	115	1731
7	2	2929	256	8	43	731	99	**15302**	5448
8	12	2179	139	9	53	1255	288	856	**49435**
F1	0.799	0.862	0.940	0.755	0.749	0.924	0.829	0.721	0.827
Re	0.698	0.879	0.953	0.625	0.618	0.905	0.747	0.617	0.912
Pr	0.933	0.846	0.927	0.953	0.949	0.943	0.931	0.867	0.757
Ji	0.285	0.301	0.320	0.274	0.272	0.316	0.293	0.265	0.293

Table 3. Error matrix for best RF and accuracy indices (eq. 6). column headers are class labels, row names are respective class indices.

Random forest can be tuned by setting the number of features (Nf) selected for each ensemble and number of trees, (Nt). Optimal parameters resulted in Nf=8 and Nt=200.This resulted from a limited choice of of Nf∈{4,8,12} and Nt ∈{50,100,200,500}, due to time limitations those 3x3 combination grid was tested, and further tuning is possible, but the differences in accuracy between the best combination and the second best combination was not particularly high (0.911 and 0.892 respectively), leading to the conclusion that further tuning does not provide significant advantage.

3.4 Results of TF classification

Table 4 shows commission and omission errors for best TF classification, with the four accuracy metrics reported in equation 6. Classes “powerline”, “impervious surfaces”, “low vegetation”, “roof” and “tree” had F1 index above 0.5, whereas classes “car”, “fence”, “shrub” had very low F1 values. The latter three classes were not able to be detected well by TF. More discussion is presented in the next section, which compares results between the two methods tested.

	Powerline	Low vegetation	Impervious surfaces	Car	Fence/ Hedge	Roof	Facade	Shrub	Tree
0	**265**	12	1	1	0	29	42	0	250
1	2	**63634**	13124	26	74	8806	247	342	12435
2	1	7294	**87276**	22	11	5889	18	9	1466
3	3	1345	421	**219**	7	742	33	56	882
4	3	2784	232	5	**119**	519	246	115	3399
5	19	14350	6927	19	11	**77313**	238	122	10049
6	18	1051	230	9	28	812	**4612**	60	4404
7	15	5866	602	20	43	2174	319	**391**	15388
8	21	4065	457	30	30	3631	1033	258	**44701**
F1	0.560	0.639	0.827	0.108	0.031	0.740	0.512	0.030	0.607
Re	0.442	0.645	0.856	0.059	0.016	0.709	0.411	0.016	0.824
Pr	0.764	0.634	0.799	0.624	0.368	0.774	0.679	0.289	0.481
Ji	0.219	0.242	0.292	0.051	0.015	0.270	0.204	0.015	0.233

Table 4. Error matrix for best TF and accuracy indices (eq. 6) – column headers are class labels, row names are respective class indices.

3.5 Comparison between TF and RF results

Figure 5 shows accuracy metrics (recall and precision) for each class and method in function of percentage of data used for training. It can be seen that RF increases steadily for some classes, for example Fences/Hedge Recall metric grows from 0.121 to 0.618 for RF, whereas TF has very low Recall for that class. For both TF and RF Recall was higher than Precision or close to it, except for shrub class which had better Precision than Recall. Shrub is very much misclassified by committing to low vegetation and tree classes, which is quite obvious. TF method results in much lower accuracy for shrub class than RF. In this investigation we did not create a specific feature containing height above ground, which could help discriminating these three vegetation classes. It is a trivial task to add such feature, and future work will take this into account.

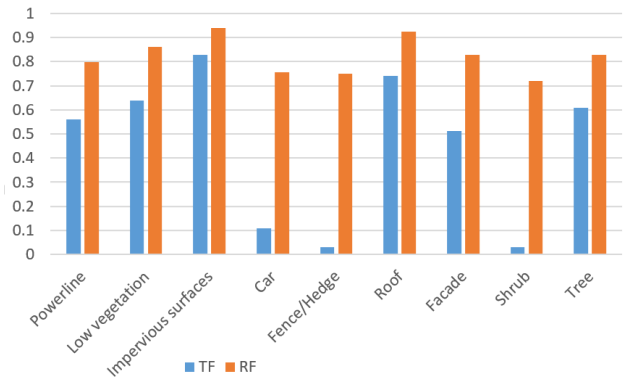

Figure 4. Comparison of best F1 values between classes and methods used.

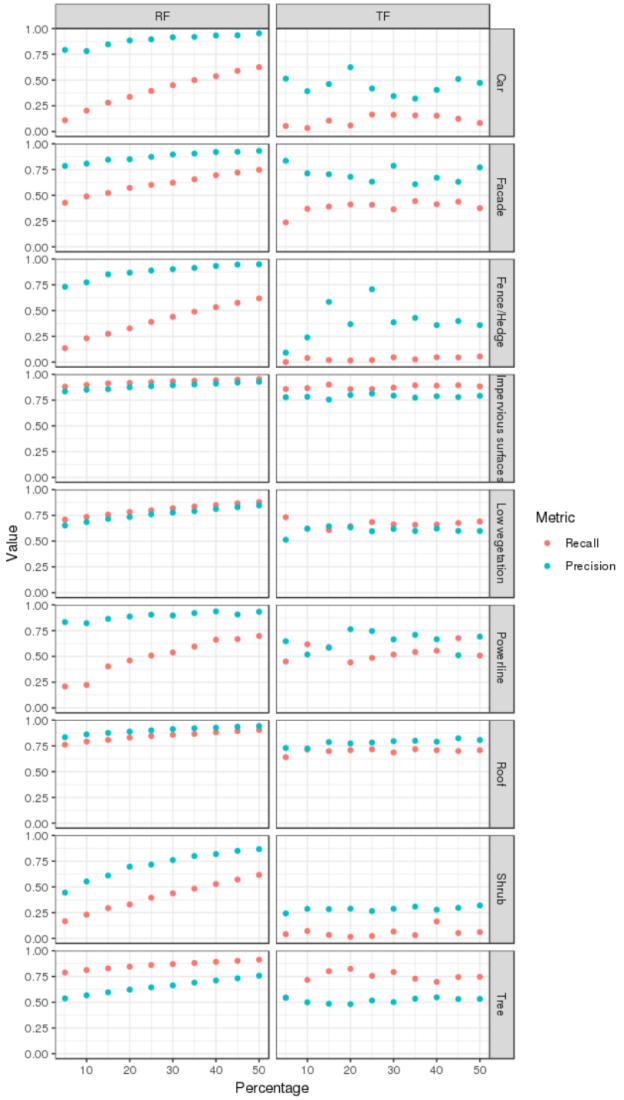

Figure 5. Comparison of accuracy metrics (recall and precision) for each class and method in function of percentage of data used for training.

It is worth noting that misclassified points are on the edges of the roof as can be seen in Figure 7, and on points with peculiar characteristics. Classes with omission/commission errors are Shrub and Tree, which can be expected considering that height above ground is the only difference that separates the two classes.

3.6 Variable importance

Variable importance is an interesting aspect to assess when applying data mining techniques (Cortez and Embrechts, 2013). In this case the predictors were many, but some had similar and likely collinear information. To see which variable had more weight importance can be computed based on corresponding reduction of predictive accuracy when that variable is not used. Using permutation, the decrease of node impurity can be assessed. In RF the Gini importance index is defined as the averaged Gini decrease in node impurities over all trees in the forest. Results showing most important and least important input variables is shown in Figure 6 below. The most important variables for prediction are the CC (curvature change), the eigen-entropy in 2D and the height differences – standard deviation and

range - in the cluster of neighbours used to calculate the predictors.

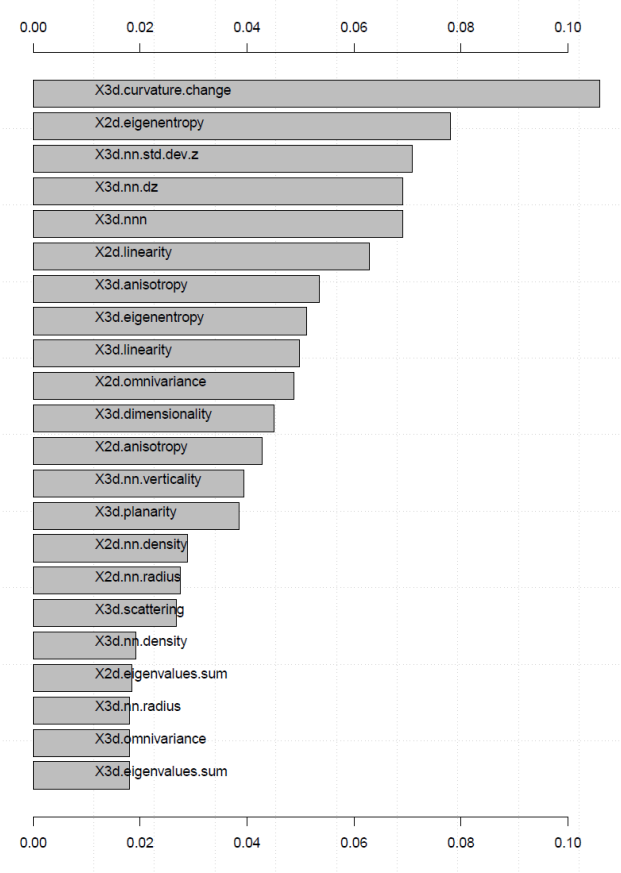

Figure 6. Variable importance for RF.

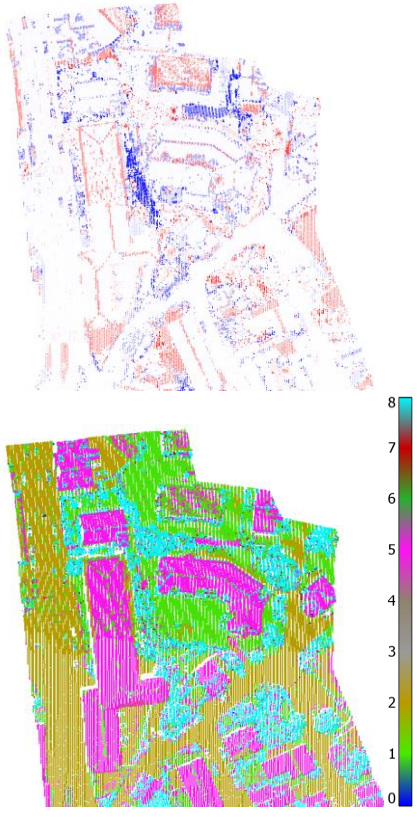

Figure 7. Top and bottom are misclassified points in RF and TF respectively: Middle is the final product from TF.

4. CONCLUSIONS

In this investigation a comparison between two machine learning (ML) models for semantic classification of an aerial laser scanner point cloud is presented. One model is Random Forest (RF), the other is a multi-layer neural network, TensorFlow (TF). Accuracy results were compared over a growing set of training data, using a stratified independent sampling over classes from 5% to 50% of the total dataset. Results show RF to have average F1=0.823 for the 9 classes considered, whereas TF had average F1=0.450. F1 values where higher for RF than TF, due to complexity in the determination of a suitable composition of the hidden layers of the neural network in TF, and this can likely be improved to reach higher accuracy values. Further study in this sense is planned.

The results from the ISPRS benchmark regarding this dataset are all from supervised approaches (Rottensteiner et al., 2014b, 2014a). The F1 scores are similar, sometimes higher, then the ones found in this work. Similarity of results are found also in the fact that car and trees have lower accuracies than roofs and facades. Also other CNN achieve comparable accuracies. PointNet achieves 48% average class accuracy on the indoor benchmark dataset S3DIS with 13 classes in tests by Charles et al. (2017).

Work in the direction of using advanced methods for classification of unstructured point clouds is giving promising results. It will also benefit from faster processors that can provide faster results also in big datasets. Understanding the composition of elements in point clouds automatically can lead to further applications in the realm of using point cloud data for multiple uses.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., Brain, G., Osdi, I., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D.G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow : A System for Large-

Scale Machine Learning. In: *12th USENIX Symposium on Operating Systems Design and Implementation*. 265–283. doi:10.1126/science.aab4113.4

Abellán, J., Mantas, C.J., Castellano, J.G., 2017. A Random Forest approach using imprecise probabilities. *Knowledge-Based Systems*, 134, 72–84. doi:10.1016/j.knosys.2017.07.019

Belgiu, M., Drăgu, L., 2016. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*. doi:10.1016/j.isprsjprs.2016.01.011

Breiman, L., 2001. Random Forests. In: *Machine Learning*. Kluwer Academic Publishers, 5–32. doi:10.1023/A:1010933404324

Cortez, P., Embrechts, M.J., 2013. Using sensitivity analysis and visualization techniques to open black box data mining models. *Information Sciences*. doi:10.1016/j.ins.2012.10.039

Grilli, E., Menna, F., Remondino, F., Scanning, L., Scanner, L., 2017. A review of point clouds segmentation and classification algorithms. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII, 1–3. doi:10.5194/isprs-archives-XLII-2-W3-339-2017

Guo, L., Chehata, N., Mallet, C., Boukir, S., 2011. Relevance of airborne lidar and multispectral image data for urban scene classification using Random Forests. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66, 56–66. doi:10.1016/j.isprsjprs.2010.08.007

Hemmes, T., 2018. Classification of large scale outdoor point clouds using convolutional neural networks. Delft University of Technology

Kulkarni, V.Y., 2013. Random Forest Classifiers : A Survey and Future Research Directions, 36, 1144–1153

Piragnolo, M., Grigolato, S., Pirotti, F., 2019. Planning harvesting operations in forest environment: remote sensing for decision support. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. 33–40. doi:10.5194/isprs-annals-IV-3-W1-33-2019

Piragnolo, M., Masiero, A., Pirotti, F., 2017. Open source R for applying machine learning to RPAS remote sensing images. *Open Geospatial Data, Software and Standards*, 2, 16. doi:10.1186/s40965-017-0033-4

Pirotti, F., Sunar, F., Piragnolo, M., 2016. Benchmark Of Machine Learning Methods for Classification of a Sentinel-2 Image. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 335–340. doi:10.5194/isprs-archives-XLI-B7-335-2016

Qi, C.R., Su, H., Kaichun, M., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 77–85. doi:10.1109/CVPR.2017.16

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J.D., 2014a. Theme section "Urban object detection and 3D building reconstruction." *ISPRS Journal of Photogrammetry and Remote Sensing*. doi:10.1016/j.isprsjprs.2014.04.009

Rottensteiner, F., Sohn, G., Gerke, M., Wegner, J.D., Breitkopf, U., Jung, J., 2014b. Results of the ISPRS benchmark on urban object detection and 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 93, 256–271. doi:10.1016/j.isprsjprs.2013.10.004

Shalev-Shwartz, S., Ben-David, S., 2014. Understanding Machine Learning. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, Cambridge. doi:10.1017/CBO9781107298019

Thomas, H., Goulette, F., Deschaud, J.E., Marcotegui, B., Gall, Y. Le, 2018. Semantic classification of 3d point clouds with multiscale spherical neighborhoods. In: *Proceedings - 2018 International Conference on 3D Vision, 3DV 2018*. doi:10.1109/3DV.2018.00052

Vosselman, G., 2013. Point cloud segmentation for urban scene classification. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*. doi:10.5194/isprsarchives-XL-7-W2-257-2013

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. doi:10.1016/j.isprsjprs.2015.01.016

Weinmann, M., Weinmann, M., Mallet, C., Brédif, M., 2017. A Classification-Segmentation Framework for the Detection of Individual Trees in Dense MMS Point Cloud Data Acquired in Urban Areas. *Remote Sensing*, 9, 277. doi:10.3390/rs9030277