Adaptive Multiscale Feature Extraction in a Distributed System for Semantic Classification of Airborne LiDAR Point Clouds

Satendra Singh, and Jaya Sreevalsan-Nair, *Senior Member, IEEE*

Abstract

Multiple spatial scales have been used extensively for feature extraction from LiDAR point clouds. These features have been used for semantic classification, segmentation, and other data analysis methods. There is a gap in the adaptive methodology for the effective use of multiple scales here. This stems from determining the best strategy to aggregate the information or features gathered from different scales. The widely used multiscale method is feature extraction at an optimal scale, which is in itself an adaptive method. However, the success of identifying the optimal scale depends on the set of scales used in its determination, as it must include the scale where the global minimum of eigenentropy occurs. An alternative method is to average features across multiple scales, which works in specific scenarios. In order to improve the flexibility of using different methods in the same workflow, we propose an adaptive method for the selection of multiscale feature extraction for semantic classification of LiDAR point clouds, with a focus on airborne laser scans. Our decision-making process for finding the best multiscale method exploits spatial locality of the features. We show how such a control strategy can be implemented in an Apache Spark–Cassandra distributed system for processing large-scale point clouds using voxelization for preserving spatial locality, and binomial logistic regression for selecting voxels to implement a specific multiscale method at. Our results show significant improvement in classification accuracy in the Dayton Annotated Laser Earth Scan (DALES) data, implemented using Spark MLlib in our distributed system.

Index Terms

Airborne LiDAR point clouds, semantic classification, Apache Spark, Cassandra, Connector, distributed system, multiscale, optimal scale, voxelization, random forest classifier, binomial logistic regression.

I. Introduction

SEMANTIC classification is one of the routinely implemented data processing operations on Light Detection and Ranging (LiDAR) data. The 3-dimensional (3D) point clouds acquired using aerial laser scanning (ALS) usually span across large topographic regions, thus including urban settlements, forests, etc. Here, we focus on 3D point clouds of urban regions, with relevant semantic classes such as, buildings, roads, vegetation, etc. Semantic classification can be done using supervised or unsupervised machine learning, which require handcrafted features, or using deep learning.

Many of the features that are significant for classification are derived from the local neighborhood. Use of multiple spatial scales for gathering salient information has been practiced for geometric analysis of LiDAR point clouds to characterize spatial locality effectively [1]. There are different ways of aggregating the information across scales, which influence the value of the extracted features. Using a scale where global minimum of entropy occurs as optimal and subsequently for feature extraction [1] has been widely practiced for LiDAR point cloud classification [2]. However, this method is fraught with the appropriate choice of initial scales which must contain the scale with global minimum value of entropy. As an alternative to optimal scale determination, the average of relevant features across scales has been used for geometric reconstruction and semantic classification has been explored to a limited extent for airborne LiDAR point clouds [3]. However, averaging works only if there is limited (statistical) variation in the feature values across different scales. Yet another way of aggregating scales is to use the features from all scales as a long vector [4], which requires neighborhood approximation with high point density. The state-of-the-art implementation of long vectors is significantly different from those of optimal scale and averaging approaches. Optimal scale as well as scale-averaged features have shown good results in semantic classification of airborne LiDAR point clouds [2], [3]. Given the limitations of both methods, an adaptive choice of the multiscale feature extraction at each point between the two methods is bound to improve the classification accuracy, to be tested on a selected classifier.

The pointwise computation of multiscale features using is in itself a compute-intensive problem. The compute requirements become exacerbated for an adaptive multiscale method for large-scale point clouds. Recently, we have proposed an integrated distributed system [5], [6] using Apache Spark [7] and Cassandra [8] for processing large-scale point clouds. Hence, we propose a control strategy of reuse of computed features that leads to efficient implementation of the adaptive method on the distributed system.

This document is a preprint, as on July 20, 2021. The authors are with Graphics-Visualization-Computing Lab (GVCL), International Institute of Information Technology, Bangalore, 26/C Electronics City, Hosur Road, Bangalore 560100, India. *Corresponding author: Jaya Sreevalsan-Nair.* e-mail: jnair@iiitb.ac.in | {satendra.singh}@iiitb.org

This work was supported by the Early Career Research Award from Science and Engineering Research Board, awarded to J. Sreevalsan-Nair, by the Department of Science and Technology, Government of India, and the Intel India Research Fellowship.

Apache Spark is a unified data analytics engine that uses in-memory processing using a programming abstraction, called as Resilient Distributed Data (RDD). Spark is often integrated with storage systems, *e.g.,* Cassandra. Spark implements the partitioning of the data in Cassandra, to be stored in tabular format in the nodes (or systems) in a cluster. Both Spark and Cassandra provide the advantage of horizontal scaling, thus accommodating large-scale data. They jointly parallelize data analytics. A Datastax Spark-Cassandra-connector [9] is used for efficient integration that leverages data locality to reduce network latency. The connector is essential for querying in Cassandra from Spark. We use this integrated system to exploit its inbuilt Spark MLlib (Machine Learning Library), for feature extraction and semantic classification [5], [6]. The persistent data storage enables interactive visualization of large-scale point clouds [6].

Multiscale methods exploit local dependencies. In order to exploit the spatial locality in the adaptive multiscale method, we propose voxelization of the point cloud. We then use these voxels as smallest units to determine the *local* strategy for choice of multiscale method. We also choose voxelization to be a part of the implementation on the distributed system, as a process *integratable* with the spatial partitioning and the parallelism in the system.

Ground truth is an ideal metric to make the decision between the two methods at each point. However, in a real-world implementation, the ground truth is usually unavailable. Hence, we propose a novel approach where the voxels are classified based on its better performance in the semantic classification. Posing this as a binary classification problem, we propose the use of a binomial logistic regression model (LRM). We choose a random forest classifier (RFC) for semantic classification owing to its suitability for ALS point cloud analysis [2] and the availability of its implementation in Spark MLlib.

Adaptive strategies in analyzing airborne LiDAR point cloud data have been studied in the recent past. A collection of spherical and cylindrical neighborhood shapes, including optimal scale for the former, has been used for feature extraction [10]. These features after normalization have been used in an RFC. The classification results show improvement in overall accuracy, when using the hybrid features. We use cubical neighborhood in our distributed system owing to the trade-off between computation and accuracy, and its approximation to spherical neighborhood [5]. Since computing features using different neighborhood shapes is not an efficient strategy for large-scale point cloud processing, we explore approaches where the same features can be reused in different ways. This is the overall motivation behind our hybrid strategy.

Our contributions include:

- a novel adaptive control strategy of combining multiscale feature extraction methods, namely, optimal scale and averaging operation, to improve accuracy of semantic classification of ALS point clouds,
- an effective application of voxelization for preserving spatial locality of implementation of selected multiscale method,
- a novel use of logistic regression to predict the classification outcome using widely used optimal scale approach in the random forest classifier, in order to select the voxels for changing to the multiscale averaging method.

II. Adaptive Multiscale Featue Extraction

We propose a control strategy of adaptive multiscale feature extraction using optimal scale and averaging operation for improving accuracy of semantic classification of ALS point clouds. For assessing our proposed strategy, we use ground truth to confirm our hypothesis. We then propose an efficient data management strategy to implement multiscale aggregation using point cloud voxelization and logistic regression.

Multiscale Feature Extraction and Classification: There are up to 21 pointwise features that are found to be significant for semantic classification of LiDAR point clouds [2], where several of them characterize the local neighborhood. The local neighborhood is defined using a local geometric descriptor. These significant features are either height- or eigenvalue-based. The eigenvalue-based ones are computed from the eigenvalues of the local geometric descriptor, $\lambda_0 \geq \lambda_1 \geq \lambda_2$. The commonly used descriptor is the covariance matrix, that is computed as the sum of tensor (outer) products of distance vector with each neighbor. The local neighborhood size is considered to be the scale for feature extraction. For instance, the radius of the spherical neighborhood or the edge length of a cubical neighborhood is used as the scale [5]. Owing to the uncertainty in environmental data, multiple scales are routinely used to extract these locality-based features.

Computing features at the *optimal* scale is an adaptive multiscale method. The scale where entropy is the global minimum is commonly used as the optimal scale owing to the reduction in uncertainty [1]. While averaging features across scales is an alternative strategy [3], they are contingent upon the averaged features being the optimal representative value across scales. Most of the height- and eigenvalue-based features can be averaged across scales to give representative values. The height-based features are computed from the local neighborhood. The local geometric descriptors are positive semidefinite second-order tensors [11], whose tensor invariants are the eigenvalue-based features. Thus, averages can be representative values if there is less statistical variance across scales. Averaging certain features also implies stochastic modeling. The saliency map $\{C_l, C_s, C_p\}$, when averaged, represents the likelihood values of the point belonging to the geometric classes of line, surface, point-type features, respectively. The different spatial scales may be considered as mutually exclusive events [11], and the joint probability given by the saliency map is the overall likelihood across scales.

Here, we consider the union of the feature subset of height-based H_{3D}, and 3D eigenvalue-based features $S_{\lambda,3D}$ [2] as the feature vector. The choice of features is primarily made based on their relevance for classification [12], relatively high classification accuracy with the use of $S_{\lambda,3D}$ at the optimal scale in an RFC [2], and the integrability of the feature computation in our distributed system [5], [6].

The features we use are: (a) eigenvalue-based features, namely, eigenentropy E_λ, omnivariance O_λ, sum of eigenvalues Σ_λ, change of curvature Δ_c, (spherical) anisotropy A_λ, saliency map $\{C_l, C_s, C_p\}$, and (b) height-based features using corrected height from ground z, namely, standard deviation σ_z, and range Δ_z (difference between maxima and minima). The eigenvalue-based features at each scale are:

$$E_\lambda = -\sum_{i=0}^{2} \lambda_i \ln(\lambda_i); \quad O_\lambda = \sqrt[3]{\lambda_0.\lambda_1.\lambda_2}; \quad \Sigma_\lambda = \sum_{i=0}^{2} \lambda_i; \quad \Delta_c = \frac{\lambda_2}{\Sigma_\lambda};$$

$$A_\lambda = \frac{\lambda_0 - \lambda_2}{\lambda_0}; \quad \{C_l, C_s, C_p\} = \{\frac{\lambda_0 - \lambda_1}{\Sigma_\lambda}, \frac{2.(\lambda_1 - \lambda_2)}{\Sigma_\lambda}, \frac{3.\lambda_2}{\Sigma_\lambda}\}.$$

The feature vector for each scale s, $\mathrm{F}_s = H_{3D} \cup S_{\lambda,3D}$. Thus,

$$\mathrm{F}_s = \{E_\lambda, O_\lambda, \Sigma_\lambda, \Delta_c, A_\lambda, C_l, C_s, C_p, \sigma_z, \Delta_z\}.$$

Optimal scale s_{opt} is the scale at which E_λ is the minimum, in which case the pointwise feature vector is $\mathrm{F}_{opt} = \mathrm{F}_{(s=s_{opt})}$. The averaged feature vector is $\mathrm{F}_\mu = \frac{1}{n_s}\sum_s \mathrm{F}_s$, for n_s scales.

We use the extracted features in an RFC for LiDAR point cloud classification, that is implemented using Spark MLlib. The classification accuracy for the point cloud is measured using intersection over union (IOU), averaged across all classes, and overall accuracy (OA). For a point cloud with N_c classes,

$$\mathrm{IOU}=\frac{1}{N_c}\sum_C \mathrm{IOU}_C, \text{ where for a class, } \mathrm{IOU}_C = \frac{TP}{TP+FP+FN},$$

$$\text{and } \mathrm{OA}=\sum_C \frac{TP+TN}{TP+TN+FP+FN},$$

where TP, TN, FP, and FN are class-wise counts of true-positive, true-negative, false-positive, and false-negative classification outcomes, respectively, for the point cloud.

Control Strategy – Rationale and Implementation: While theoretically the global minimum of entropy measure is an effective indicator of optimal scale [1], the global minimum may not be always present in the selected scales, in practice. This leads to the case where not all regions in the point cloud are best represented using optimal scale. In practice, determining optimal scale requires an exhaustive search for the global minimum entropy across multiple scales. Especially in large-scale point clouds, a trade-off between efficiency and accuracy is used, where a local minimum is often chosen.

As a result, we hypothesize that regions with low classification accuracy owing to this trade-off, can improve the accuracy using averaged features. We use the ground truth to test this hypothesis. Ground truth identifies points misclassified using the optimal scale approach, but correctly classified using the averaging approach. The accuracy is then improved adaptively by replacing the optimal scale features with averaged ones.

Definition: The *adaptive multiscale feature extraction* is the control strategy of choosing the best multiscale features at each point in a point cloud to improve the classification outcome.

1) Voxelization: While we have incorporated spatial locality in local geometric descriptors, we also expect that the spatial contiguity (locality) influences the multiscale approach for the region. Using spatial locality for decision-making can be achieved using our distributed system, designed for managing and processing large-scale point clouds. We have shown in our previous work that an Apache Spark-Cassandra integrated distributed system can be effectively used for semantic classification of large-scale airborne LiDAR point clouds, using both averaged multiscale features [5] as well as optimal scale ones [6]. The spatial partitioning used in the distributed system is along either x- or y-axis. This partitioning is used for distributing the data to different Spark nodes. For region-wise locality, we now voxelize each spatial partition along both x- and y- axes in a gridded format, thus forming $v_x \times v_y$ cuboidal voxels. Now, we propose to apply a chosen multiscale approach to all points in a voxel, *i.e.,* a subset of the point cloud, thus implementing a region-wise application of a chosen method. For the sake of simplicity, we keep $v_x = v_y$ here. The voxelization of the point cloud also improves system efficiency through increased parallelism, as the voxelization and regional analysis are implemented in the same Spark node.

2) Voxel Selection Using Logistic Regression: We first rank voxels based on their average IOU values when classified using the optimal scale features. We select the low-ranked voxels, whose points are then classified using the averaged features.

We use ground truth analysis to establish the improvement in classification accuracy. However, in practice, an automated method is needed for selecting voxels for applying the change in the multiscale method, *i.e.,* in the absence of ground truth. Given that there is no single feature that gives clear clusters for LiDAR point cloud classification, we need a combination of features in the feature vector that can *predict* the classification outcome for a voxel. We now pose this problem as that of binary classification, given the voxels are to be grouped based on its apt multiscale aggregation method, from our two selected methods. Hence, a binomial logistic regression model (LRM) is applicable where the outcome (dependent) binary variable is the success of classification using our chosen classifier. We use the following pointwise feature vector for the logistic regression, which is derived from the same used in the classifier:

$$\mathrm{F}_{LRM}=(\{z\} \cup \mathrm{F}_s) \setminus \{C_s, \Delta_z\} \quad .$$

We remove C_s and Δ_z from the feature vector, as they are linearly dependent on remaining saliency map, $\{C_l, C_p\}$, and height z, respectively. The choice between z and Δ_z has been decided based on our results. The LRM is computed and applied pointwise.

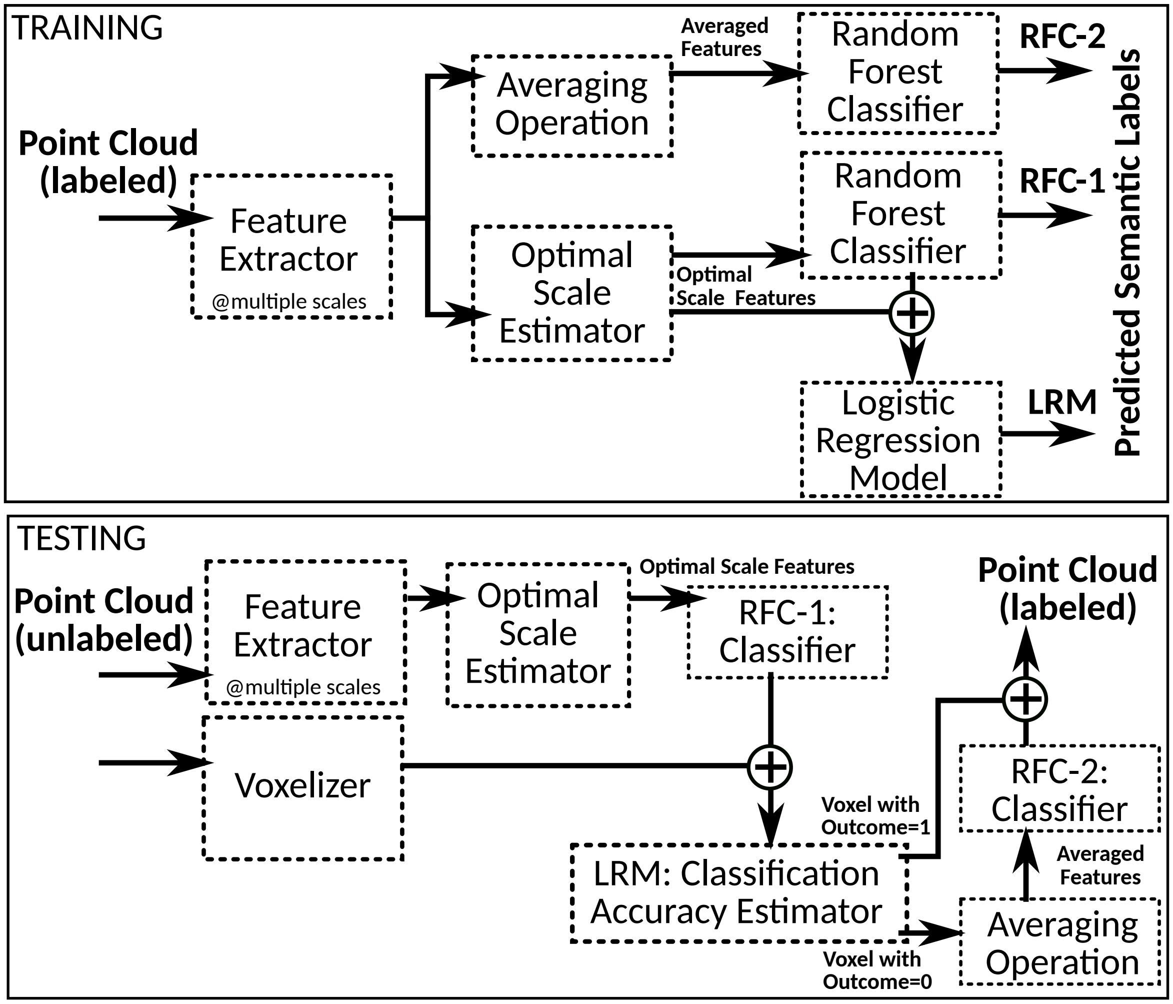

Fig. 1: Our proposed workflow involves the hybrid strategy of multiscale aggregation and a logistic regression model for improving classification accuracy in large-scale point clouds using an Apache Spark-Cassandra integrated distributed system [5], [6]. Models RFC-1, RFC-2, and LRM are trained using Apache Spark MLlib.

Since the number of points in each voxel is not a constant, the feature size varies at the voxel level. Hence, performing logistic regression directly for voxel classification based on a long vector of pointwise features is inefficient and impractical. Hence, we perform pointwise analysis using logistic regression, and then we determine the *success rate* for each voxel. We use a threshold for the success rate τ_s, lower than which, we select the voxel to change its multiscale method for feature extraction. We expect the voxel size and voxel-wise point density to influence the improvement in accuracy. We study the influence of voxel size in our results, here. The voxels could be further merged or split to achieve a desired point density in each voxel for effective implementation. This leads to adaptive voxel sizes, which is to be studied in future.

Workflow: In our proposed workflow (Figure 1), we first train the RFC models for both multiscale methods, namely, RFC-1 and RFC-2 for the optimal scale and averaged features, respectively. We then use the accuracy of RFC-1 outcomes to train the binomial LRM, now using *normalized* features. For testing an unlabeled point cloud, we first run the RFC-1, to get the pointwise class labels. We run the voxelizer and the LRM, after storing the class labels from the RFC-1. If the LRM gives outcome ‘1’ for a voxel, *i.e.,* success in running RFC-1 for the voxel is above the threshold τ_s, then we retain the labels from RFC-1 for all the points in the voxel. If the outcome for the voxel is ‘0’, then we compute averaged features for only those points and implement RFC-2 on the voxels. The class labels of all points in the voxel are now the outcomes of RFC-2. Our workflow finally outputs labeled point clouds.

An alternate to the strategy of *correcting* optimal scale features using averaged features, is to *choose* between the two approaches. However, choosing between two approaches requires implementation of both approaches for each point, which is inefficient for large-scale point clouds. This step is unavoidable for training but can be avoided for testing, and using pre-trained models for testing can keep it as a one-time process. Thus, we adopt a sequential implementation of one method, followed by correction of selected regions using the second method, which is more efficient for big data.

TABLE I: Tiles in DALES dataset [13] used in Random Forest Classifier (RFC) and Logistic Regression Model (LRM)

Tile ID	#points	Tile ID	#points	+Roles (LRM)
RFC Training		RFC Testing		
`5110_54320`	17,747,769	`5150_54325`	11,882,667	Train
`5110_54460`	13,784,200			
`5110_54475`	11,981,458	`5080_54400`	12,219,779	Test
`5110_54495`	11,930,713			
Total: 55,444,140 points		Total: 24,102,446 points		

TABLE II: Intersection Over Union (IOU) and Overall Accuracy (OA) measures of Semantic Classification by RFC

Granularity Level	Average IOU	OA
Pointwise	Using Optimal Scale	
	0.2782	74.18%
	Using Multiscale Averaging	
	0.3312	85.48%
Proposed Adaptive Multiscale Feature Extraction *(Voxelization along x- and y-axes gives $v_x \times v_y$ voxels)*		
Point/Voxel Selection based on Ground-truth Analysis		
Pointwise	0.3333	85.611%
Voxel-wise		
10×10 voxels (50.00m×50.00m)	0.3380	85.49%
20×20 voxels (25.00m×25.00m)	0.3395	85.44%
40×40 voxels (12.50m×12.50m)	0.3452	85.53%
80×80 voxels (6.25m×6.25m)	0.3476	85.62%
160×160 voxels (3.12m×3.12m)	0.3514	85.77%
320×320 voxels (1.56m×1.56m)	0.3520	85.79%
Voxel Selection based on Logistic Regression Model		
Voxel-wise		
10×10 voxels (50.00m×50.00m)	0.3378	85.22%
20×20 voxels (25.00m×25.00m)	0.3395	85.61%
40×40 voxels (12.50m×12.50m)	0.3381	85.59%
80×80 voxels (6.25m×6.25m)	0.3384	85.68%
160×160 voxels (3.12m×3.12m)	0.3391	85.60%
320×320 voxels (1.56m×1.56m)	0.3520	85.72%

The metric size of each voxel is given in parentheses in the leftmost column.

III. Experiments, Results, and Discussion

For our experiments, we have used the recently published Dayton Annotated Laser Earth Scan (DALES) dataset [13], which is of the City of Surrey, Canada. The dataset with 0.5 billion points, covering 10 km^2 of area. It contains 40 tiles of dense, labeled point data, including urban regions, of which 29 are training and remaining, testing files. Each tile is of size 0.5km, with a point density of 50ppm (points per metre) as average, and 20ppm as minimum. The eight semantic classes available in DALES are: ground, vegetation, cars, trucks, poles, power lines, fences and buildings. We split the data for training and testing for the Random Forest Classifier (RFC) in the Spark MLlib (Table I). We split the testing data used for the RFC further, as training and testing the binary logistic regression model (LRM), also implemented using Spark MLlib. We have used four tiles for training the RFC, and two tiles for RFC testing, which are used again for LRM training and testing (Table I). We have used a ∼70/30 split for training and testing on the RFC. For multiscale feature extraction, we have used ten uniformly distributed scales with cubical neighborhood sizes l=1m, at minimum, l=10m at maximum, and Δl=1m as the scale increment.

For the distributed system, we have used Apache Spark 2.4 and Cassandra 3.0., with five executor nodes on Spark. Of the five Spark nodes, one executor node runs on the master node. Each of the five nodes uses Intel i7 processor @2.80 GHz, 4 cores, 8 logical processors, and 8GB RAM. We have additionally used AWS (Amazon Web Services) S3 bucket for storing the model and raw point cloud. In terms of performance, our distributed system takes 2.287s for per-point processing, followed by 117μs for per-voxel processing.

Table II gives results for both point- and voxel-wise classification. We observe from the pointwise results that the averaging approach by itself significantly outperforms the optimal scale approach. Hence, we use a conservative success rate threshold τ_s= 90% here. We first explore our hypothesis of the implementation of adaptive strategy of multiscale averaging and voxelization improving the classification accuracy. The accuracy at point-level using ground truth analysis is the baseline (Table II). We see that, as the voxel size reduces the accuracy improves, but at the same time, the point density is influenced by the voxel size. At 320×320 voxelization, we have an average of ∼900 points per voxel, that describes spatial locality sufficiently. We expect that any further increase in the number of voxels will not improve results owing to the reduction in point density.

We observe that running LRM in the voxelized test data, at different voxel sizes, gives comparable results to that of the ground truth analysis, thus demonstrating that our choice of logistic regression model is effective. We have repeated the experiment after swapping the LRM training and testing tiles (Table I). However, this swap deteriorates classification to 73.45% OA with 0.3032 IOU for optimal scale, and 74.84% OA with 0.2825 IOU for averaging. We observe that the tile `5080_54400` has relatively more class imbalance, in comparison to the tile `5150_54325`. Overall, the class balance of the tile for LRM training is a determinant of the success of our proposed method.

We have also repeated these experiments for the Vaihingen benchmark dataset [14], for proof of concept. The dataset has an average point density of 5-7 ppm, with total of 1.2 billion points. The experiments could not be run for more than 20×20 voxelization owing to the high sparsification of the voxels. The voxel selection using ground truth analysis gives 69.02% OA and 0.2059 IOU; and using LRM gives 68.73% OA with 0.2078 IOU, when using Area-3 (323,895 points) as training and Area-2 (266,674 points) as testing. We have used three scales with sizes l={3.78m, 4.20m, 4.62m}, based on our previous work [3]. We observe that the high point density is indeed a determinant of the success of our proposed method.

Discussion: Our proposed workflow has revealed salient observations on the dependency of the voxel size on the classification accuracy, when using our control strategy and adaptive multiscale method. Further analysis of the voxelization for automated decision-making using logistic regression is required.

Our work opens up novel adaptive methods of extracting hand-crafted features to be used in different learning methodologies, namely, unsupervised and supervised learning. Recently, the choices made for the extraction of appropriate hand-crafted features, e.g. radial neighborhood, to be used in an RFC, have been reused for building the convolution function in deep learning methods [15] for point classification.

Our work also demonstrates how the big data technology frameworks can be harnessed effectively for solving large-scale point cloud processing problems. Our proposed workflow shows that novel engineering methods of using available features in these frameworks pave way for more complex data science workflows for spatial data.

IV. Conclusions

In this work, we have demonstrated that adaptive use of different multiscale feature extraction methods within a single point cloud yields effective feature vectors for classification using supervised learning. We have designed a control strategy to implement the adaptive method on an Apache Spark – Cassandra integrated distributed system, building on our previous work [5]. Our results show that the use of voxelization to exploit spatial locality of multiscale aggregation, and logistic regression to predict the success of the RFC have been effective. Our proposed strategy demonstrates how extracted features can be used in multiscale methods using optimal scale approach or averaging, to improve classification accuracy. Our contributions pave way for novel workflows to be implemented on the big data framework for large-scale point cloud analysis. Generalizing our adaptive method to terrestrial and mobile LiDAR datasets requires further study.

References

[1] J. Demantké, C. Mallet, N. David, and B. Vallet, "Dimensionality based Scale Selection in 3D LiDAR Point Clouds," *The Intl. Arch. of the Photogramm., Remote Sens. and Spatial Info. Sc.*, vol. 3812, pp. 97–102, Sept. 2011.

[2] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS J. of Photogramm. and Remote Sens.*, vol. 105, pp. 286–304, 2015.

[3] J. Sreevalsan-Nair and P. Mohapatra, "Influence of Aleatoric Uncertainty on Semantic Classification of Airborne Lidar Point Clouds: A Case Study with Random Forest Classifier Using Multiscale Features," in *2020 IEEE Intl. Geosc. & Remote Sens. Symp. (IGARSS)*. IEEE, 2020, pp. 1066–1069.

[4] T. Hackel, J. D. Wegner, and K. Schindler, "Fast semantic segmentation of 3D point clouds with strongly varying density," *ISPRS Ann. of the Photogrmm., Remote Sens. & Spatial Info. Sc.*, vol. III-3, pp. 177–184, 2016.

[5] S. Singh and J. Sreevalsan-Nair, "A distributed system for multiscale feature extraction and semantic classification of large-scale LiDAR point clouds," in *2020 IEEE India Geosc. and Remote Sens. Symp. (InGARSS)*. IEEE, 2020, pp. 74–77.

[6] S. Singh and J. Sreevalsan-Nair, "A Distributed System for Optimal Scale Feature Extraction and Semantic Classification of Large-Scale Airborne LiDAR Point Clouds," in *Intl. Conf. on Distributed Computing and Internet Tech.* Springer, 2021, pp. 280–288.

[7] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin *et al.*, "Apache Spark: A unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[8] A. Lakshman and P. Malik, "Cassandra: A decentralized structured storage system," *ACM SIGOPS Operating Systems Review*, vol. 44(2), pp. 35–40, 2010.

[9] DataStax Analytics, "Spark Cassandra Connector," https://github.com/datastax/spark-cassandra-connector, 2014, accessed on April 14, 2021.

[10] R. Blomley and M. Weinmann, "Using Multi-scale Features for the 3D Semantic Labeling of Airborne Laser Scanning Data," *ISPRS Ann. of the Photogrmm., Remote Sens. & Spatial Info. Sc.*, vol. IV-2/W4, pp. 43–50, 2017.

[11] J. Sreevalsan-Nair and B. Kumari, "Local Geometric Descriptors for Multi-Scale Probabilistic Point Classification of Airborne LiDAR Point Clouds," in *Modeling, Analysis, and Visualization of Anisotropy*. Springer Cham, 2017, pp. 175–200.

[12] M. Weinmann, B. Jutzi, and C. Mallet, "Feature relevance assessment for the semantic interpretation of 3d point cloud data," *ISPRS Ann. of the Photogrmm., Remote Sens. & Spatial Info. Sc.*, vol. II-5/W2, pp. 313–318, 2013.

[13] N. Varney, V. K. Asari, and Q. Graehling, "DALES: A Large-scale Aerial LiDAR Data Set for Semantic Segmentation," in *Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 186–187.

[14] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf, "The ISPRS benchmark on urban object classification and 3D building reconstruction," *ISPRS Ann. of the Photogrmm., Remote Sens. & Spatial Info. Sc.*, vol. I-3, pp. 293–298, 2012.

[15] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and Deformable Convolution for Point Clouds," in *Proceedings of the IEEE/CVF Intl. Conf. on Computer Vision*, 2019, pp. 6411–6420.