RESAMPLING METHODS FOR A RELIABLE VALIDATION SET IN DEEP LEARNING BASED POINT CLOUD CLASSIFICATION

A. Nurunnabi *, F. N. Teferle

Geodesy and Geospatial Engineering, Faculty of Science, Technology and Medicine, University of Luxembourg,
6, rue Richard Codenhove-Kalergi, L-1359 Luxembourg − (abdul.nurunnabi, norman.teferle)@uni.lu

KEY WORDS: Bootstrap, Cross-Validation, Machine Learning, Monte Carlo, PointNet, Semantic Segmentation, Supervised Method

ABSTRACT:

A validation data set plays a pivotal role in tweaking a machine learning model trained in a supervised manner. Many existing algorithms select a part of available data by using random sampling to produce a validation set. However, this approach can be prone to overfitting. One should follow careful data splitting to have reliable training and validation sets that can produce a generalized model with a good performance for the unseen (test) data. Data splitting based on resampling techniques involves repeatedly drawing samples from the available data. Hence, resampling methods can give better generalization power to a model, because they can produce and use many training and/or validation sets. These techniques are computationally expensive, but with increasingly available high-performance computing facilities, one can exploit them. Though a multitude of resampling methods exist, investigation of their influence on the generality of deep learning (DL) algorithms is limited due to its non-linear black-box nature. This paper contributes by: (1) investigating the generalization capability of the four most popular resampling methods: *k*-fold cross-validation (*k*-CV), repeated *k*-CV (R*k*-CV), Monte Carlo CV (MC-CV) and bootstrap for creating training and validation data sets used for developing, training and validating DL based point cloud classifiers (e.g., PointNet; Qi et al., 2017a), (2) justifying Mean Square Error (MSE) as a statistically consistent estimator, and (3) exploring the use of MSE as a reliable performance metric for supervised DL. Experiments in this paper are performed on both synthetic and real-world aerial laser scanning (ALS) point clouds.

1. INTRODUCTION

Automatic and efficient classification of point clouds contributes in many applications including visualization, three-dimensional (3D) city modelling, construction health monitoring, autonomous driving, road furniture and assets management, urban planning, and augmented reality (Nurunnabi et al., 2015; Li et al., 2021; Su et al., 2022). Machine learning (ML) methods based on hand-crafted features such as decision trees (Tran et al., 2015), support vector machines (Secord and Zakhor, 2007) and random forests (Guo et al., 2011) have been used for automatic classification. However, these methods are heavily dependent on the representation ability of the features. Consequently, the generalization abilities of the developed models are restricted because of their limitations of using shallow architectures (Li et al., 2021). Recently, deep learning (DL) methods with supervised training have been showing increasing performance in 3D point cloud processing such as classification, segmentation, and identification (Goodfellow, et al., 2016; Nurunnabi et al., 2021a). It is a common belief that supervised learning methods need sufficiently large data sets, and their success strongly depends on the available data quality.

Generalization capability, i.e., having the ability to perform for unseen data, is one of the most important characteristics of DL models. We should follow an appropriate data splitting strategy to get reliable training and validation data sets. Several ways exist for validating a ML/DL model; one of which is to train a model on a big chunk of an existing data set and validate it on the remaining part. This approach is common in DL based point cloud classification (Qi et al., 2017a). Train/test split is another approach, when a part of available data, the validation data set, is separated before developing a model and is used to evaluate the model, when it is estimated. A common tenet is that the data used for training should not be used for validation (Ramezan et al., 2019). The argument is that the overlapping data used for training and validation may increase the likelihood of overfitting because of the possible autocorrelation among the spatially close points (Becker et al., 2017). Stehman (2009) developed a systematic sampling design strategy to minimize autocorrelation in sample sets.

In many cases, given data are limited, even sometimes getting sufficiently large data is impossible. It is a common phenomenon in medical diagnostic data; in the case of rare diseases such as when dealing with autistic adults (Vabalas et al., 2019). We know that resampling methods are good for hand-crafted feature-based ML techniques and when data sets are small. Lyons et al. (2018) investigated and justified resampling methods for large-scale remote sensing data classification. Resampling algorithms have been used for classification accuracy assessment, and to see the potential of selection of training and validation data sets in object-based classification, within the remote sensing framework (Weber and Langille, 2007; Zhen et al., 2013). When dealing with large-scale point clouds, still we can take the advantage of using resampling methods, in several ways, e. g., (i) to evaluate the generality of the model, (ii) since DL algorithms are data-hungry, large and more data generated by resampling methods are beneficial to train a better model, and (iii) collecting large data always involves more time, cost and human effort and may be unavailable, so it saves time, cost and labour. Furthermore, in case of insufficient data, with the help of resampling methods, we can generate new data, develop a model and validate with more data that can achieve sufficient generality and robust results for the test (unseen) data.

One of the most popular ways of data splitting is Cross-validation (CV) has been used frequently when the available data are limited as well as to improve the statistical reliability of the results (Ramezan et al., 2019). For the CV, no need of getting extra data to have a validation data set. Only once, the given data are split into several parts, and then every part is used as a

*Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-617-2022 | © Author(s) 2022. CC BY 4.0 License.

validation set one after another. CV methods have many variants including leave-one-out and *k*-fold. Resampling methods repeatedly draw random samples from the available data. So, they allow us to create more data from the available data. Reasonably, data splitting based on resampling methods has better generalization power for the test data. Despite a multitude of methods that exist in the literature (Efron and Tibshirani, 1993; Boos and Stefaski, 2013; Tsamardinos et al., 2018), investigation of their performance for achieving the generality of DL models is very limited due to their non-linear nature and underlying complexities of DL based methods. In this study, we focus on the four most common resampling methods: *k*-fold cross-validation (*k*-CV), repeated *k*-CV (R*k*-CV), Monte Carlo CV (MC-CV) and bootstrap that have been frequently used in ML algorithms for getting training and validation data sets. We employ a DL algorithm, PointNet (Qi et al., 2017a), for per point classification in point clouds, and to investigate the influence of the resampling methods on the developed DL models by studying the values of an error metric used for training and to evaluate the model.

DL based classifiers use several metrics such as the well-known cross-entropy, Mean Square Error (MSE), and overall accuracy (Michelucci, 2018). Practically, the analytical representation of the error metrics is complex. In this paper, we show that MSE is a statistically reliable and consistent estimator that has the potential for measuring the performance of a DL algorithm. We also explore that the well-known Central Limit Theorem (CLT; Boos and Stefansky, 2013) can play a crucial role in defining MSE as an asymptotically normally distributed estimator. The contributions of this paper are: (1) investigating and comparing the resampling methods for creating training and validation data sets used in training and validating a DL based point cloud classifier (PointNet; Qi et al., 2017a), and comparing also with the conventional train/test split approach, (2) justifying MSE as a statistically consistent estimator, and (3) exploring the use of MSE as a statistically reliable performance metric for a supervised DL algorithm. Our experiments are performed on both synthetic and real-world aerial laser scanning (ALS) point clouds.

The remaining part of the paper is organized as follows. Section 2 briefly presents the basic ideas of resampling methods, and state-of-the-art DL algorithms for point cloud classification. In Section 3, we propose the process of investigating the significance of using resampling methods for selecting training and validation data sets, and the potential of using MSE as an error and evaluation metric for the generalization power of a DL algorithm used in point cloud classification. Section 4 demonstrates the proposed method through synthetic and real-world ALS data sets. Section 5 concludes the paper.

2. RELATED PRINCIPLES AND METHODS

This section presents a quick discussion about related resampling methods, and DL algorithms that are used in point cloud classification.

2.1 Resampling

Resampling is the process of creating a multitude out of a given sample. From each of these samples, we can estimate an estimator function and later plot those outcomes in a distribution. Thus, studying the location, scatter or other necessary statistical moments of the resultant distribution can reveal the true estimator of the population. In the sense of model fitting, a model is refitted based on each sample to serve the purpose of learning more about the adapted model (Wang et al., 2021). Many resampling methods (e. g., randomization, bootstrap, and Monte Carlo) are available in existing literature (Good, 2010; James et al., 2015; Manly, 2020). Some are briefly discussed below.

Fisher (1935) introduced the well-known permutation test also called randomization test or randomization sampling. This is to represent resampling without replacement, i.e., drawing observations from a given sample, randomly, without replacement. Quenouille (1949) developed the Jackknife resampling algorithm to estimate bias and standard error. A distinctive feature of this method is that a different observation is excluded every time of sampling. Bootstrap is perhaps the most widely used resampling method for deriving the distribution of an estimator that was developed by Efron (1979). Sooner, after developing bootstrap it was realized that bootstrap is more flexible than the Jackknife (Efron, 1982; Beasley and Rodgers, 2009). In this method, data are sampled repeatedly with replacement, so data can be occurred time and again with the same probability in each sampling. Several variants of bootstrap are available in the literature those can be classified as parametric and non-parametric bootstrap (Tsamardinos et al., 2018). CV is another most applied resampling approach that has been used in ML algorithms. *k* fold CV (*k*-CV) and leave-one-out CV (LOO-CV) are the two popular variants of CV. *k*-CV randomly divides the data set into *k* sections (folds), one of the *k* folds is used for validation while others go for training. The advantage of CV is that all observations are used for both training and validation purposes. The LOO-CV is the extreme case of CV, where *k* is the number of total observations. The LOO-CV is appropriate for small data sets as it has the highest computational cost, hence it is not appropriate for large data sets.

2.2 Deep learning (DL) in point cloud, and PointNet

Deep learning methods use artificial neural networks composed of many processing layers to learn representations of data with multiple levels of abstraction (LeCun et al., 2015). DL has achieved tremendous success in computer vision for 2D image data, but it is hard to get the desired level of success with point clouds in a similar fashion, especially for convolutional neural network (CNN; LeCun et al., 1989) type representations. This is because, point cloud data are irregular (variable point density), unstructured and have no order. CNNs that have unprecedented success in image representation are designed to process data in the form of multiple structured arrays (Krizhevsky et al., 2012; Hackel et al., 2017). Alternatively, researchers do some transformations of point clouds into regular grids and/or multiple image collections that can carry the possibility of information loss as well as need extra time for processing.

Qi et al. (2017a) proposed the first DL architecture, named PointNet that does not use convolution operators, rather consists of fully connected layers. PointNet generates features using shared multi-layer perceptrons (MLPs), and aggregates them by a symmetric function called max-pooling. Max-pooling works as the global signature (maximal response) among all the points. Per-point features are obtained by local and global information aggregation. To aid classification, a tiny spatial transformer network (T-Net; Jaderberg et al., 2015) is implemented to transform the data into a canonical form that increases invariance to input permutation. Nurunnabi et al. (2021b) showed that PointNet is simple, computationally efficient, and has potential for large-scale outdoor point cloud classification. In PointNet, the point coordinates (*x*, *y*, and *z*) of points in a point cloud are used as the raw inputs with the potential use of additional

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-617-2022 | © Author(s) 2022. CC BY 4.0 License.

features e.g., colours (*R*, *G*, *B*). The authors (Nurunnabi et al., 2021b) also showed that using LiDAR (Light Detection and Ranging) features such as intensity (*I*) and return numbers (*RN*s) with the point coordinates produces better results than using colour information as the input vectors.

A multitude of point-based DL algorithms have been introduced in recent years for point cloud classification (Li et al., 2018; Guo et al., 2020; Hu et al., 2020). To capture points' local structure, Qi et al. (2017b) improved the PointNet algorithm, and developed a hierarchical network PointNet++, inspired by the 2D-CNN, where input captures features at progressively larger scales (increasing neighbourhood size) along a multi-resolution hierarchy. To address the problems of irregular and unordered data format, Li et al. (2018) proposed PointCNN, a generalized version of CNN that makes convolution on X-transformed points, for the *k* Nearest Neighbours (*k*NN). It was developed on PointNet++ using an MLP network. Among the others, the most noteworthy are PointConv (Wu et al., 2019), KPConv (Thomas et al., 2019), and RandLA-Net (Hu et al., 2020). The reader can see Guo et al. (2020) and Li et al. (2021) to know more about DL algorithms for point cloud classification. Point-based methods do not consider explicit information loss and becoming popular day by day. We employ PointNet, because of its simplicity to understand, and it is sufficiently fast to implement our method on large-scale outdoor point clouds.

3. METHODOLOGY

In this section, we propose a methodology to: (i) show MSE is a statistically consistent estimator, and to demonstrate its potential for measuring the generalization power of a DL classifier, (ii) investigate the consequences of using resampling methods for selecting training and validation data sets. We fulfil the objectives in three steps; first, we show MSE is a statistically consistent estimator and it follows an asymptotically Normal distribution. In the second step, we describe how the sampling methods work to get training and validation data sets. In the third step, we present how the PointNet algorithm works and uses the training and validation sets from the second step, and develops a classifier. Finally, we investigate the performance of the DL models that are developed based on the data sets from different resampling and train/test split methods.

3.1 Use of MSE as an error metric

In this step, we find an error metric (estimator) to train and assess a DL model. To do that, we find an estimator $\hat{\theta}$, which is statistically consistent, effectively unbiased (Eq. 1) with high precision (Eq. 2), and is also asymptotically normally distributed (Eq. 6). $\hat{\theta}$ is a consistent estimator if it satisfies the following conditions,

$$\text{(i) } bias\left(\hat{\theta},\theta\right) = 0 \text{ as } n \to \infty \text{, and} \tag{1}$$

(ii) standard error, $se\left(\hat{\theta}\right) = 0$ as $n \to \infty$, i.e. high precision. (2)
Equivalently,

$$MSE(\hat{\theta},\theta) = 0 \text{ as } n \to \infty. \tag{3}$$

The standard error (*se*) is defined as the standard deviation (*sd*) divided by the square root of *n* (the sample size), where

$$sd = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i-\bar{x})^2}, \tag{4}$$

$$\text{sample average, mean, } \bar{x} = \frac{1}{n}\sum_{i}^{n} x_i, \tag{5}$$

and x is a random variable. Following the well-known CLT (Boos and Stefansky, 2013), we can show that the distribution of a mean of an independently and identically distributed (i.i.d) random variable is asymptotically normal (Eq. 6), i.e.,

$$\hat{\theta} \sim N(\theta, se\,(\hat{\theta})^2) \tag{6}$$

for sufficiently large sample size, *n*. More mathematical descriptions for developing an asymptotic distribution of an estimator are available in Efron and Tibshirani (1993) and Good (2013). We explore this fact empirically, and show that a synthetic data set following an exponential distribution (Fig. 1a) has a mean ($\bar{x}$) which gradually converges to the shape of a Gaussian (Normal) distribution with increasing sample sizes (see Fig. 1 b-e). Fig. 1(f) the boxplots show that with the increasing sample size, all that means ($\bar{x}$) are concentrated to their same central locations (i.e., mean and median are almost the same). The MSE of an estimator $\hat{\theta}$ of a parameter θ can be defined as the average squared difference between the estimator and its parameter, i.e.,

$$MSE\left(\hat{\theta}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{\theta}-\theta\right)^2. \tag{7}$$

It is understandable that the MSE function defined in Eq. 7 is nothing but a mean, and follows a Normal distribution (Fig. 2d; Experiment 1). The use of MSE has at least two benefits; one, it is analytically tractable, and it can be interpreted in terms of both bias and variance. We estimate the mean of MSE (m_{MSE}) and standard deviation of MSE (sd_{MSE}) for the models that are trained with the help of the employed DL algorithm, PointNet. We use resampling methods to get the training and validation data sets as described in Section 3.2, and decide on a reliable validation set that has the least m_{MSE} and sd_{MSE}. That validation set with the least m_{MSE} (i.e., tends to zero) and/or sd_{MSE} rationally provides more generalization power for a test (unseen) data set.

3.2 Performing resampling methods to get training and validation data sets

We perform the above mentioned (in Section 1) four most common resampling methods using standard procedures (Efron and Tibshirani, 1993; James et al., 2015; Hastie et al., 2017). Suggested by the reviewers for the earlier version of this paper, besides the resampling methods, we also consider the train/test split approach to compare. For the train/test split, first, we divide the available data into two parts. One is separated at the beginning to use for the test of the final model, and the rest part is used to apply resampling approaches to get training and validation sets. Later, the training and validation part is split again into two disjoint (training and validation) sets. Resampling methods are described below in brief to follow this paper.

(i) For the *k* cross-validation (*k*-CV), we shuffle the seen data and divide them into *k* folds (parts). One fold is held out as the validation set while the rest *k*-1 folds are merged to use for training. The process of selecting training and validation data lasts *k* times. DL model is trained and validated every *k* time. The performance of the final model is the average (arithmetic mean) performance of the *k* models. Usually, researchers make 5-fold or 3-fold to get a reasonable portion (%) of data for training and validation sets. In this paper, we fix *k* = 5 to have 20% of the data for validating a trained model.

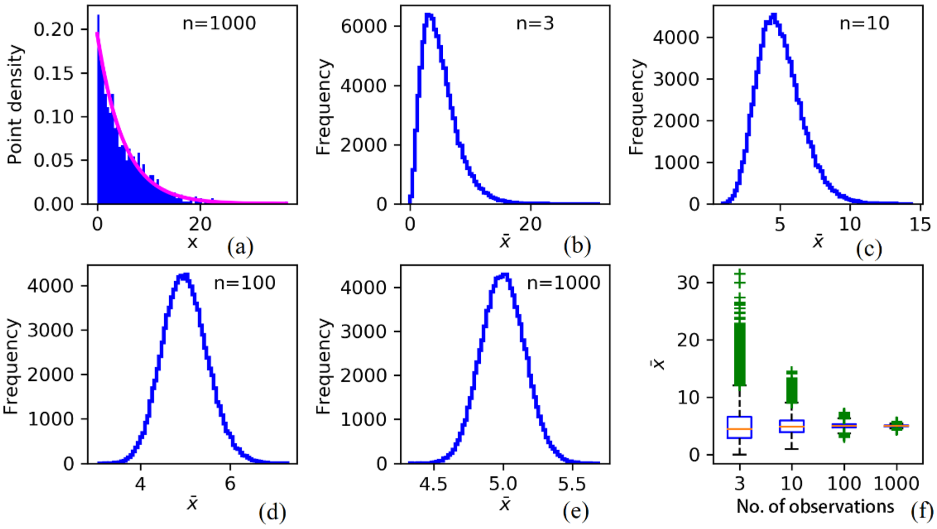

Figure 1. Demonstration of the Central Limit Theorem (CLT): (a) histogram of a 1D synthetic data set, follows an Exponential distribution with mean 5. 10^5 samples of sizes 3, 10, 100 and 1,000 are drawn, means are $\bar{x}$; line diagrams of frequency versus $\bar{x}$ for: (b) n =3, (c) n =10, (d) n =100, and (e) n =1000, plots show that $\bar{x}$ with an increasing number of points follows Normal distribution, (f) boxplots for the means ($\bar{x}$) of different sizes of samples.

(ii) Repeated k-CV (Rk-CV) has the opportunity of making data R times larger than the given data set, it checks the validity of the developed model with R×k validation sets. It repeats the k-CV, R times. However, before performing k-CV, the data are shuffled every time for R times. We repeat k-CV 20 times (i. e., R = 20).

(iii) Monte Carlo CV (MC-CV) randomly draws a prefixed (user-defined) size subset (a portion without replacement) from the seen data, uses it as a training set while the rest of the seen data are used as a validation set. It repeats this process several (m) times defined by the user, and the performance of the model is the average performance of the repeats. We repeat 100 times.

(iv) The main idea behind bootstrap approach is generating new data sets from a given one by resampling with reiteration. In our case, bootstrap draws samples (e.g., B times with replacement) of the same size from a validation set. This approach trains a model only once on the training set and validates B times with the bootstrap samples. We fix B =100.

3.3 Implementation of a DL based classification algorithm

We use PointNet as a DL classifier for the pointwise classification of laser scanning point clouds. The input to the PointNet network can be arranged as a matrix array of size $N \times M$, where N is the number of points and M is the number of associated features for each point. Unlike PointNet, instead of colour values, we use related LiDAR features (e. g., point intensity (I) and return numbers (RN)) with the point coordinates. The *ReLU* (Rectified Linear Unit) is used for the hidden layers and the Softmax activation functions are used for the output layers. Instead of multiple cross-entropy, MSE is used as the loss function, and the *Adam* (a stochastic optimizer) with a learning rate of 0.001 is used to train the model. To reduce the influence of vanishing and exploding gradients, the '*He initialization'* strategy is used with the *ReLU* activation function. *Batch Normalization* (Ioffe and Szegedy, 2015) is used for all the layers, and the dropout layers are installed only for the last MLP. Interested readers are referred to Goodfellow et al. (2016) and Qi et al. (2017a) for more details on the DL and PointNet algorithm, respectively.

4. EXPERIMENTS, RESULTS AND EVALUATION

We perform three experiments in this section. One experiment is on a synthetic data set, and the other two are through real-world ALS data sets. We investigate the use of MSE as a consistent estimator for error estimation at the model training stage, and to evaluate the developed model with the validation data set(s). MSE is also used to measure the influence of above mentioned four resampling methods for generating training and validation data. To understand the developed model, and the classification results for the ALS data, we use five common evaluation metrics: F_1-score (F_1), mean F_1 (mF_1), Intersection over Union (IoU), mean IoU (mIoU), and the Overall Accuracy (OA). The reader is referred to Nurunnabi et al. (2022) for detail about the metrics.

4.1 Experiment 1: Synthetic data set

In the first experiment, we show that MSE is a statistically reliable error metric that follows Normal distribution for the large data. We generate 500 2D points that portray five different spirals (Fig. 2a) representing five classes. We train a 5-layer Neural Network (NN) with the well-known stochastic *Adam* optimizer, the *ReLU* and Softmax activation functions for the input and the output layers, respectively. MSE is used as the loss function. Plots b and c in Fig. 2 portray the predictions of the model. We derive, then, 1,000 samples of 100 points, repeat the training of the NN for each sample and calculate MSE values. Fig. 2d shows that MSE follows a Normal distribution. Next, resampling methods are executed 100 times for 100, 500, 1,000 and 5,000 sample points, and m_{MSE}, and sd_{MSE} are computed for each sample size. Rk-CV is performed with R=20 and k = 5, and MC-CV is performed with 100 repetitions. Table 2 explores that with the increasing sample sizes, m_{MSE} and sd_{MSE} decrease, thus justifying MSE as a consistent estimator. k-CV and bootstrap produce the least m_{MSE} and sd_{MSE}, respectively. We draw boxplots and line diagrams for the results of Rk-CV, MC-CV and bootstrap those who are evaluated 100 times with the validation data sets. Fig. 3, both the boxplots and line diagrams show that bootstrap produces more robust results with just one outlying value (Fig. 3a) of MSE. It is worth noting that bootstrap takes significantly less time (12.13 s) than the others (Table 2).

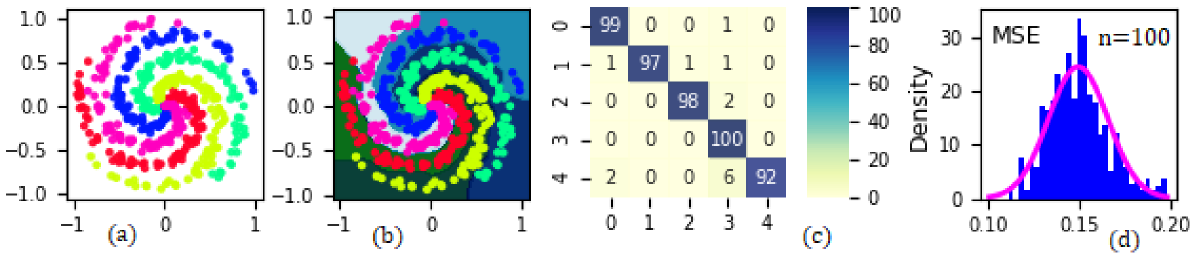

Figure 2. Classification by a neural network: (a) an artificial non-linear 2D data (500 points) set having five different spirals, (b) classification into five different regions (colours), (c) confusion matrix for the classification, (d) histograms of density versus MSE, MSE of 1,000 samples of 100 points follows a Normal distribution (magenta curve).

No. of sample points	Train/test split	*k*-CV		R*k*-CV		MC-CV		Bootstrap	
	MSE	m_{MSE}	sd_{MSE}	m_{MSE}	sd_{MSE}	m_{MSE}	sd_{MSE}	m_{MSE}	sd_{MSE}
100	0.1786	0.1348	0.0157	0.1513	0.0162	0.1500	0.0156	0.1520	0.0120
500	0.0193	0.0255	0.0072	0.0213	0.0073	0.0204	0.0078	0.0305	0.0059
1,000	0.0064	0.0076	0.0028	0.0088	0.0044	0.0096	0.0043	0.0077	0.0025
5,000	0.0042	**0.0049**	0.0015	0.0057	0.0027	0.0060	0.0031	0.0071	**0.0011**
Time (for 1,000 points)	11.73 second (s)	58.18 s		1977.52 s		1887.55 s		**12.13 s**	

Table 1. Results of different sampling methods based on 100 samples of different sizes; higher performance (i. e., lower m_{MSE} and sd_{MSE}) gains with increasing sample size.

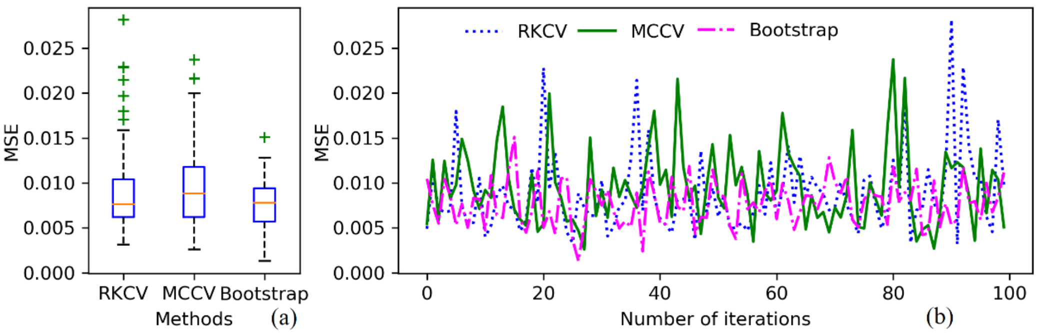

Figure 3. Results of R*k*-CV, MC-CV and bootstrap for 100 synthetic data sets: (a) Boxplots for the MSE values, (b) line diagrams for MSE versus number of iterations.

Methods	MSE results		Classification results										
	Metrics		Methods	Train/test split		*k*-CV		R*k*-CV		MC-CV		Bootstrap	
	m_{MSE}	sd_{MSE}	Class	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU
k-CV	**0.0839**	0.0528	uC	56.66	39.54	72.68	57.08	74.71	59.63	74.34	59.15	78.82	65.04
R*k*-CV	0.0749	0.0187	Ground	93.35	87.52	85.79	75.12	88.60	79.54	88.49	79.36	93.14	87.16
MC-CV	0.0802	0.0470	Building	83.44	71.58	92.07	85.30	91.70	84.67	91.46	84.27	92.11	85.37
Bootstrap	**0.0452**	**0.0020**	mF_1/mIoU	77.82	66.21	83.51	72.50	85.00	74.61	84.76	74.26	88.02	79.19
Train/test	**0.0599** (MSE)		OA	**81.95**		**84.03**		85.69		85.47		**89.13**	

Table 2. Results of train/test split and resampling methods (*k*-CV, R*k*-CV, MC-CV and bootstrap). Results show significant impact on error metrics, and classification performance metrics (in percentage, %) for the AHN test data set.

4.2 Experiment 2: AHN data set

The second experiment is performed on real-world ALS data from *Actueel Hoogtebestand Nederland*, version 3(AHN3). This open access data set has a point density of 15-20/m^2. The points are labelled in three classes (ground, building, and unclassified (uC) that includes vegetation). We choose two small parts from the AHN3 data. One part (Fig. 4b) is separated for the final test, and the other part (Fig. 4a) is used to get training and validation data sets following the train/test split and resampling methods. These training and validation data are then used to train and validate the DL, i.e., PointNet (Qi et al., 2017a), model.

We follow the specified guidelines of the PointNet. The network is performed with the training data having blocks of the size of 10m×10m, sampled with 2,048 points per block, and batch size of 32. We use the spatial coordinates as well as the LiDAR features (*I* and *RN*), and heights as the raw inputs. The heights are the differences between the *z* values of a point of interest and the lowest point in the local neighbourhood of the interest point (Nurunnabi et al., 2022). Hence, individual points are characterized by their coordinates (*x*, *y*, *z*), *I*, *RN*, and heights. Besides resampling methods, we perform train/test split approach. For the train/test split approach, training and validation sets are separated in a way so that they have no overlap.

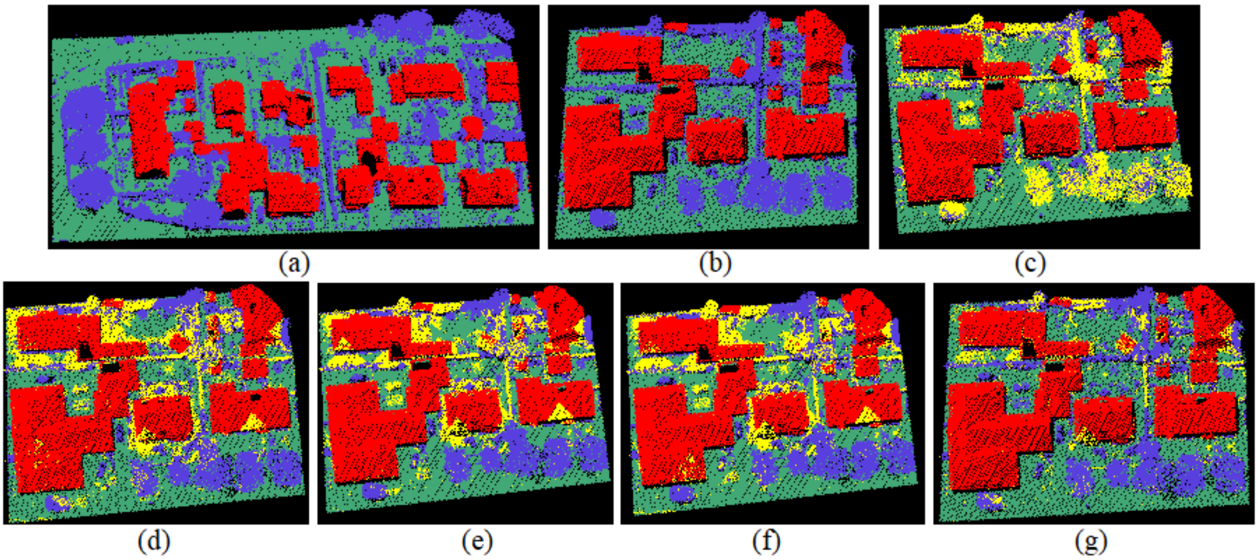

Figure 4. (a) AHN training and validation data, (b) AHN test data, classification results based on: (c) train/test split, (d) *k*-CV, (e) R*k*-CV, (f) MC-CV, and (g) bootstrap.

Table 2 and Fig. 4 present the performance of the concerned resampling and train/test split methods. Results in Table 2 show that bootstrap achieves the least values of m_{MSE} (0.0452) and sd_{MSE} (0.002). As desired, bootstrap gets the highest OA of 89.13% for the test set. Train/test split and *k*-CV get MSE = 0.0599 and m_{MSE} = 0.0839, respectively. It reveals that although the MSE (0.0599) of split/train is lower than the m_{MSE} (0.0839) of *k*-CV, train/test split gets significantly less OA of 81.95% than *k*-CV (OA = 84.03%), because train/test split evaluates its model just once with a validation set and does not get sufficient generality for the new (test) data set. Significantly better results are achieved for all the resampling methods for the test data than the non-resampling train/test split method.

4.3 Experiment 3: Vaihingen data set

We consider the ISPRS (International Society for Photogrammetry and Remote Sensing) benchmark Vaihingen data set (Niemeyer et al., 2014) that has been used frequently for the development of DL algorithm. This data set was collected by using a Leica ALS50 scanning system. It has an average point density of around 4-6/m^2. It is separated as training and test sets. The training set covers mainly a residential area of 399m×421m that consists of 753,876 points, and the test set covers two scenes of 389m×419m urban area of 411,722 points. Along with the point coordinates, each point has *I*, *RN*, the number of returns, and the respective class labels of power lines (PL), low vegetation (LV), impervious surface (IS), car, fence, roof, facade, shrub, or tree. This is an imbalanced data set that has significant disparity by the number of points in the classes. For example, the group PL has only 546 and 600 points, whereas IS has 193,723 and 101,986 points for training and test, respectively. To see the influence of imbalanced data on the classification results, this time, we perform simple random sampling method, and also stratified random sampling method to avoid the possible absence of points from the smaller groups. Training and validation sets were generated according to the resampling algorithms. The PointNet algorithm was run on the training set using a block size of 10m ×10m, and a batch size of 32. Each block contains 2,048 sample points. We feed the same input (feature) vectors and the same hyper-parameters to train the network that were used for the second experiment on the AHN3 data. The training was finished with 100 epochs.

Table 3 summarizes and explores the results for the Vaihingen test data set. Results in the table show that stratified sampling based resampling methods produce better classification rates (%) than their corresponding item from simple random sampling. We see, train/test split approach produces OA of 68.04% and 70.24% when it is based on the samples from simple random sampling and stratified sampling, respectively. For both simple random sampling and stratified sampling, train/test split method produces less correct classification (OA) rate than any of the concerned resampling approaches. For example, for stratified sampling, train/test split achieves OA of 70.24% and bootstrap achieves OA of 71.72%. All four resampling methods get almost similar success rates. With the least value of m_{MSE} (0.0248), R*k*-CV achieves the highest values of mF$_1$, mIoU and OA (71.74%). Bootstrap achieves the least sd_{MSE} with an OA of 71.72%.

Methods	Stratified sampling					Simple random sampling				
	MSE results		Classification results			MSE results		Classification results		
	m_{MSE}	sd_{MSE}	mF$_1$	mIoU	OA	m_{MSE}	sd_{MSE}	mF$_1$	mIoU	OA
Train/test split	0.0257	---	41.53	31.53	**70.24**	0.0269	---	39.44	29.81	**68.04**
k-CV	0.0284	0.0019	41.39	31.12	71.22	0.0269	0.0004	41.19	29.81	69.69
R*k*-CV	**0.0248**	0.0016	42.57	32.52	**71.74**	0.0273	0.0018	42.44	32.08	70.33
MC-CV	0.0275	0.0016	41.47	31.73	71.69	0.0267	0.0009	41.35	31.59	70.69
Bootstrap	0.0309	0.0003	42.41	32.15	**71.72**	0.0304	0.0003	40.55	30.84	71.21

Table 3. Results based on train/test split and resampling methods (*k*-CV, R*k*-CV, MC-CV and bootstrap). Outcomes reveal impact on error metrics, and classification performance metrics (in %) for the ISPRS benchmark Vaihingen test data set (Scene 1 and 2).

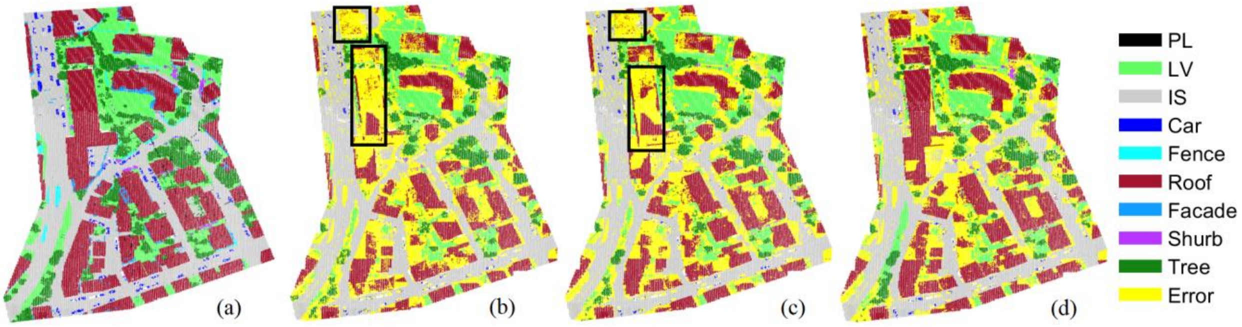

Figure 5. classification results for the Vaihingen test data set (Scene 1). (a) ground-truth, results of: (b) simple random sampling based train/test split, (c) stratified sampling based train/test split, and (d) stratified sampling based R*k*-CV. Many points are misclassified in the black rectangles for plots b and c.

Figure 5 portrays the classification results of Scene 1 of the Vaihingen test data set. To accommodate within the space, we portray results of three selected methods only for Scene 1. R*k*-CV produces significantly better results than the train/test split approach based on both simple random sampling and stratified sampling. Compared with the stratified sampling based R*k*-CV results in Fig. 5d, many more roof points in the black rectangles of Fig. 5b (simple random sampling based train/test split) and Fig. 5c (stratified sampling based train/test split) are misclassified as the false negative.

5. CONCLUSIONS

This paper showed that MSE is a statistically consistent estimator, it does work as a reliable cost function. Moreover, it can work as an error metric to assess a DL algorithm and to evaluate the generality of the model. Results on synthetic data sets showed that bias and standard deviation of an MSE tend to zero for a large and increasing sample size. For large data sets, it tends asymptotically to a shape of a Gaussian (Normal) distribution. Hence, MSE with the least values of mean and standard deviation (error) has the potential for appropriate selection of resampling methods that finds a reliable validation set to generalize a DL classifier.

Experiment showed that a specific resampling method does not always produce the best results for all data sets. The investigation explored that whatever method is used for data splitting, we should check its performance with several validation data sets to understand the generality of the developed model. In one experiment, we see that despite having lesser MSE, train/test split did not achieve better results than a resampling method. This is because the train/test split approach evaluates just with a single validation data set. Since bootstrap draws samples with replacement, it has more possibility of getting autocorrelation between points that can produce more bias to the estimators, however, bootstrap has the opportunity of resampling with the same probability for the observations and has the potential for the estimation of the distribution of a statistical estimator, e.g., MSE. Bootstrap has more potential for small data sets. Future studies will investigate the potential of using MSE and resampling methods to understand the generality of the other supervised machine learning methods, such as decision trees and random forests when they deal with large-scale point clouds.

ACKNOWLEDGEMENTS

This study is with the Project 2019-05-030-24, SOLSTICE – Programme Fonds Européen de Developpment Régional (FEDER)/Ministère de l’Economie of the G. D. of Luxembourg. We appreciate the reviewers’ comments on the earlier version that improved the quality of the paper.

REFERENCES

AHN3: *Actueel Hoogtebestand Nederland*. Available at: https://app.pdok.nl/ahn3-downloadpage/.

Beasley, W. H., Rodgers, J. L., 2009. Re-Sampling methods. *Millsap: The SAGE Handbook of Quantitative Methods in Psychology*, 362–386.

Becker, C., Hani, N., Rosinskaya, E., Angelo, E. D., Strecha, C., 2017. Classification of aerial photogrammetric 3D point clouds., *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, Vol. IV-1/W1, 2017.

Boos, D. D., Stefanski, L. A., 2013. *Essential Statistical Inference: Theory and Methods*. New York: Springer.

Efron, B., 1979. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7: 1–26.

Efron, B., 1982. *The Jackknife, the Bootstrap, and Other Resampling Plans*. Philadelphia: Society for Industrial and Applied Mathematics.

Efron, B., Tibshirani, R. J. 1993. *An Introduction to the Bootstrap*. Springer-Science + Business Media.

Fisher, R. A., 1935. *Design of Experiments*. Edinburgh, Oliver and Boyd.

Good, P. I., 2010. *Permutation, Parametric and Bootstrap Tests of Hypotheses*. Springer, New York.

Good, P. I., 2013. *Introduction to Statistics through Resampling Methods and R*, Wiley & Sons, NJ.

Goodfellow, I., Yoshua B., Aaron, C., 2016. *Deep Learning*. MIT press.

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 J. Photogramm. Remote Sens*., 66(1): 56–66.

Guo, Y., et al., 2020. Deep learning for 3D point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1–27.

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-617-2022 | © Author(s) 2022. CC BY 4.0 License.

Hackel, T., Savinov, N., Ladicky, L., Wegner, J. D., Schindler, K., Pollefeys, M., 2017. Semantic3d.net: A new large-scale point cloud classification benchmark. arXiv:1704.03847.

Hastie, T., Tibshirani, R., Friedman, J., 2017. *The Elements of Statistical Learning*. New York: Springer.

Hu, Q., et al., 2020. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. *IEEE CVPR*, 11108–11117.

Ioffe, S., Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Int. Conf. Mach. Learn. ICML*, 448–456.

Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K., 2015. Spatial transformer networks. ArXiv :1506.02025.

James, G., Witten, D., Hastie, T., Tibshirani, R. 2015. *An Introduction to Statistical Learning*. New York: Springer.

Krizhevsky, A., Sutskever, I., Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.*, 25.

LeCun et al., 1989. Backpropagation applied to handwritten zip code recognition, *Neural Comput*, 1(4), 541–551.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. *Nature*, 521(7553): 436–444.

Li, N., Kähler, O., Pfeifer, N., 2021. A comparison of deep learning methods for airborne lidar point clouds classification. *IEEE J Sel Top Appl Earth Obs Remote Sens*, 14: 6467–6486.

Li, Y., et al., 2018. PointCNN: Convolution on χ-transformed points. *Adv. Neural Inf. Process. Syst.*, 31: 820–830.

Lyons, M. B., Keith, D. A., Phinn, S. R., Mason, T. J., Elith, J. (2018). A comparison of resampling methods for remote sensing classification and accuracy assessment. *Remote Sens. Environ*, 208, 145–153.

Manly, B., 2020. *Randomization, Bootstrap and Monte Carlo Methods in Biology*. Boca Raton, FL: Chapman and Hall/CRC.

Michelucci, U., 2018. *Applied deep Learning – A Case Based Approach to Understanding Deep Learning Networks*. APRESS Media, LLC: New York.

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS J. Photogramm. Remote Sens*. 87, 152–165.

Nurunnabi, A., West, G., Belton, D., 2015. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern Recognit.* 48(4), 1404–1419.

Nurunnabi, A., Teferle, F. N., Li, J., Lindenbergh, R., Hunegnaw, A., 2021a. An efficient deep learning approach for ground point filtering in aerial laser scanning point clouds. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci*., Vol. XLIII-B1-2021, 31–38, XXIV- ISPRS Congress, 5-9 July.

Nurunnabi, A., Teferle, F. N., Li, J., Lindenbergh, R., Parvaz, S., 2021b. Investigation of PointNet for semantic segmentation of large-scale outdoor point clouds. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci.*, Vol. XLVI-4/W5, 397–404.

Nurunnabi, A., Teferle, F. N., Laefer, D. F., Lindenbergh, R., Hunegnaw, A., 2022. A two-step feature extraction algorithm: application to deep learning for point cloud classification. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci.*, Vol. XLVI-2/W1, 401–408.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *IEEE CVPR*, 652–660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. PointNet++ : Deep hierarchical feature learning on point sets in a metric space. ArXiv:1706.02413.

Quenouille, M. H., 1949. Approximate tests of correlation in time-series. *J R Stat Soc Series B, Stat Methodol*, 11: 68–84.

Ramezan, C. A., Warner, T. A., Maxwell, A. E., 2019. Evaluation of sampling and cross-validation tuning strategies for regional-scale machine learning classification. *Remote Sens.*, 11, 1–21.

Secord, J., Zakhor, A., 2007. Tree detection in urban regions using aerial lidar and image data. *IEEE Geosci. Remote. Sens. Lett.*, 2: 196–200.

Stehman, S.V., 2009. Sampling designs for accuracy assessment of land cover. *Int J Remote Sens.*, 30, 5243–5272.

Su, Y., et al., 2022. DLA-Net: Learning dual local attention features for semantic segmentation of large-scale building facade point clouds. *Pattern Recognit.*, 123, 108372.

Thomas, H., Qi, C. R., Deschaud, J. E., Marcotegui, B., Goulette, F., Guibas, L. J., 2019. KPConv: Flexible and deformable convolution for point clouds. *IEEE Int. Conf. Computer Vision*, 6411–6420.

Tran, G., Nguyen, D., Milenkovic, M., Pfeifer, N., 2015. Potential of full waveform airborne laser scanning data for urban area classification-Transfer of classification approaches between missions. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci.*, 40(7): 1317.

Tsamardinos, I., Greasidou, E., Borboudakis, G., 2018. Bootstrapping the out-of-sample predictions for efficient and accurate cross-validation. *Mach Learn*, 107(12):1895–1922.

Vabalas, A., Emma G., Poliakoff, E., Casson, A., J., 2019. Machine learning algorithm validation with a limited sample size. *PLoS One*, 14(11): e0224365. doi: 10.1371/journal.pone.0224365.

Wang, F., et al., 2021. Applying different resampling strategies in machine learning models to predict head-cut gully erosion susceptibility. *Alex. Eng. J.*, 60(6): 5813–5829.

Weber, K. T., Langille, J., 2007. Improving classification accuracy Assessments with statistical bootstrap resampling techniques. *GIsci Remote Sens*, 2007, 44(3), p. 237–250.

Wu, W., Qi, Z., Fuxin, L., 2019. PointConv: Deep convolutional networks on 3d point clouds. *IEEE CVPR*, 9621–9630.

Zhen, Z., Quackenbush, L.J., Stehman, S.V., Zhang, L., 2013. Impact of training and validation sample selection on classification accuracy and accuracy assessment when using reference polygons in object-based classification. *Int J Remote Sens*., 34 (19), 6914–6930.