*k*CV-B: BOOTSTRAP WITH CROSS-VALIDATION FOR DEEP LEARNING MODEL DEVELOPMENT, ASSESSMENT AND SELECTION

A. Nurunnabi*[1], F. N. Teferle[1], D. F. Laefer[2], F. Remondino[3], I. R. Karas[4], J. Li[5]

[1]Geodesy and Geospatial Engineering, Faculty of Science, Technology and Medicine, University of Luxembourg, 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg,−(abdul.nurunnabi, norman.teferle)@uni.lu
[2]Center for Urban Science and Progress; Department of Civil and Urban Engineering, Tandon School of Engineering, New York University, – debra.laefer@nyu.edu
[3]3D Optical Metrology (3DOM) unit, Bruno Kessler Foundation (FBK), Trento, Italy, – remondino@fbk.eu
[4]Department of Computer Engineering, Karabuk University, Karabuk, Turkey, – ismail.karas@karabuk.edu.tr
[5]Geography and Environmental Management, University of Waterloo, Waterloo ON N2L 3G1, Canada−junli@uwaterloo.ca

KEY WORDS: Classification, Cross-Validation, Neural Network, PointNet, Semantic Segmentation, Supervised Machine Learning

ABSTRACT:

This study investigates the inability of two popular data splitting techniques: train/test split and *k*-fold cross-validation that are to create training and validation data sets, and to achieve sufficient generality for supervised deep learning (DL) methods. This failure is mainly caused by their limited ability of new data creation. In response, the bootstrap is a computer based statistical resampling method that has been used efficiently for estimating the distribution of a sample estimator and to assess a model without having knowledge about the population. This paper couples cross-validation and bootstrap to have their respective advantages in view of data generation strategy and to achieve better generalization of a DL model. This paper contributes by: (i) developing an algorithm for better selection of training and validation data sets, (ii) exploring the potential of bootstrap for drawing statistical inference on the necessary performance metrics (e.g., mean square error), and (iii) introducing a method that can assess and improve the efficiency of a DL model. The proposed method is applied for semantic segmentation and is demonstrated via a DL based classification algorithm, PointNet, through aerial laser scanning point cloud data.

1. INTRODUCTION

Supervised deep learning (DL) is a non-linear machine learning (ML) approach that has been shown to successfully learn very complex patterns and rules used in many areas that include image understanding, point cloud classification, speech recognition, and natural language processing (Bishop, 2006; Goodfellow et al., 2016; Montavon et al., 2018). This technique constructs a deep artificial Neural Network (NN) architecture, and develops a model based on a given set of examples (data) associated with inputs and outputs. Usually, a model developer splits the given data mainly into two parts: training and validation. The required model is developed based on the training set and is evaluated on the validation set that is used for tuning the model hyper-parameters. The final step involves learning the pattern of the hold out test data (if available) and/or the data to be available in future. The efficacy of such models is highly hampered by an absence of statistical considerations regarding the resulting hyper-parameters and evaluation metrics used in developing the model (Taylor, 2005; Montavon et al., 2018). Recent works show that selection process of training and validation data has a significant impact on the model performance (Majgaonkar et al., 2021; Weidner and Walton, 2021). A first choice of getting training and validation sets is the split-and-training (train/test split) approach, but this approach results in only a single training and validation set pair, which hinders the initial learning and cannot achieve sufficient generalization power (Harrington et al., 2017; Nurunnabi and Teferle, 2022). A popular workaround of this problem for ML/DL approaches is the *k*-fold Cross-Validation (*k*CV) approach, which selects a group of training and validation sets. A common belief is that since *k*CV splits the data several times, the model generality can be improved as the final model is the average of using multiple pairs training and validation data sets (Wainer and Cawley, 2021). Many interesting works comprehensively discuss the prospects and problems of using cross validation (CV; Daszykowskio et al. 2002; Puzyn et al., 2011). In this paper, we investigate that both the train/test split and *k*CV fail to achieve sufficient generality for the test (and future) data. Another often overlooked issue is the proper evaluation of the developed model performance from the different training sets (Tuia et al., 2016; Becker et al., 2018; Nurunnabi and Teferle, 2022). Understanding the efficiency of a supervised DL model is vital for not only tuning the model hyper-parameters but also to estimate its generalization capacity. However, this task is complex and challenging mainly due to the black box nature of DL approaches (Taylor, 2005; Montavon et al., 2018).

The most common assessment practice for choosing the best ML/DL model is the well-known hold-out protocol (Tsamardinos et al., 2018). Apart from the training and validation sets, this approach holds a portion of the available data to serve as an independent test set. Then the performance of the models from different pairs of training and validation sets are checked with the test set, and finalize the model that is the best performing one. Problematically all the available training and validation sets are samples, just parts (subsets) of an unknown larger data set that can be defined as the population. Hence, knowing the performance of the developed model on a or some specific subset(s) of the full data set may not be statistically representative or reasonable. The statistical way to know about the quality of an estimator is to study its sampling distribution. Bootstrap is a statistical resampling technique that can estimate the parameters of a model and serves as an inference tool for characterizing the sampling distributions of estimators of the model. It assesses the quality of estimators in terms of their means, standard errors, confidence intervals (CIs), etc. (Efron and Tibshirani, 1993; Davison and Hinkley, 1997; Basiri et al.,

*Corresponding author

2017). Other resampling approaches include randomization, jackknife, and Monte Carlo. The reader can see comprehensive discussion on the resampling algorithms, and more benefits of bootstrap over jackknife and others in the literature (James et al., 2015; Tsamardinos et al., 2018; Manly, 2020). Wainer and Cawley (2021) conducted a comprehensive empirical study of different flat and nested cross-validation algorithms. Nurunnabi and Teferle (2022) discussed the potential of repeated *k*CV and Monte-Carlo CV in DL based classification. As will be explained in the next two sections, this paper introduces *k*CV-based bootstrap (*k*CV-B) approach for producing multiple validation sets to improve the generalization power of a DL model. This is done by coupling *k*CV and bootstrap for better selection of training and validation sets and used with the DL algorithm PointNet (Qi et al., 2017), which is simple and fast for per point classification of large-scale point clouds (Nurunnabi et al., 2021b). Finally, the bootstrap inferential procedures are used to estimate Mean Square Error (MSE) based evaluation metrics such as mean, standard error and CIs of MSE, and to assess the final model. Scientific contributions of this paper include: (i) a study of the potential of bootstrap resampling algorithm to generate many validation sets from a given validation set that can increase the generalization capability of a DL algorithm used in pointwise point cloud classification, (ii) the development of a process of appropriate selection of training and validation sets for a supervised DL modelling, (iii) devise a new algorithm that couples *k*CV and bootstrap to improve a DL based classification algorithm in large-scale outdoor point clouds, and (iv) introduce a learning process to improve, select and assess the efficiency of a DL model. The remainder of the paper is arranged as follows. Section 2 comprises the relevant ideas and principles of train/test split, *k*CV, bootstrap, point cloud and PointNet. Section 3 proposes the methodology. Section 4 demonstrates the new method through the PointNet classification algorithm using aerial laser scanning (ALS) point clouds, Section 5 makes a brief discussion, and Section 6 concludes the paper.

2. RELATED PRINCIPLES AND METHODS

This section presents a brief discussion about methods and principles that are used in the new algorithm and for comparison.

2.1 Train/test split and *k*-fold cross-validation (*k*CV)

The train/test split is a simple and common approach for generating training and validation data sets, that behaves like random sampling. Usually, first it shuffles the available data, and then splits into two parts. One part is separated at the beginning as the hold out test set to test the final model that is developed based on the other part. The other part is split into two disjoint sets: training set and validation set. The training set is used to train a model and the validation set is to fix a model, i.e., tuning hyper-parameters and validating the trained model.

Unlike the train/test split approach, *k*-fold cross-validation (*k*CV) splits the available samples (data) into *k* (user defined) distinct groups (folds) of approximately equal size (James et al., 2015; Wainer and Cawley, 2021). Before splitting data, they can be shuffled or just split into specific spatial regions following some arrangement or in a systematic order. For the *k*CV, each time, a training set of *k*-1 folds is used to train a model, and the model is evaluated using the remaining fold. Hence, the *k*CV based models are developed *k* times, so that each fold can be part of the validation sets. Raschka (2020) noted that the main advantage of using cross-validation (CV) is that each observation of the given data set has the opportunity of appearing in both the training and validation. The average performance of the developed *k* models is considered as the performance of the final model. That can also be expressed as the generalization power of the final model. This paper also demonstrates the insufficiency to generalize a model just by evaluation once or few times with a validation set(s).

2.2 Bootstrap

Bootstrap is a widely used resampling technique for statistical decision-making w. r. t. sample estimators, to know the distributions of the sample estimators, and for better understating about population parameters. It draws *B* (a data dependent, prespecified large number, e.g. 100 or 500) random samples (data sets called bootstrap samples) of same size with replacement from a given data set. That means samples come with equal probability. Bootstrap helps to draw statistical inference on the learning model and associated evaluation metrics (estimators) based on many bootstrap samples. The basic principle that follows nonparametric bootstrap uses bootstrap samples to approximate the sampling distributions for estimating confidence interval and to test the statistical hypotheses designed for an estimator. A major benefit of using bootstrap is that it is not reliant on following the Central Limit Theorem (Boos and Stefanski, 2013) to understand population. To know more about bootstrap, its principles and properties, the reader is referred to Efron and Tibshirani (1993) and Davison and Hinkley (1997).

2.3 Point cloud and PointNet

Point clouds can be represented as a type of spatial structure usually represented by a tuple (a trio of *x, y*, *z*) coordinates and may include colour, intensity, return number, and other meta data. Point clouds can provide geometric detail such as shape, size, and orientation of objects at sufficient level of detail for various tasks such as surface reconstruction (Nurunnabi et al., 2012), normal estimation (Nurunnabi et al. 2015), and for 3D geometric primitives such as cylinders fitting (Nurunnabi et al., 2019). However, their inherent 3D complicates the use of DL approaches such as Convolutional Neural Networks (CNNs; LeCun et al., 1989) that are regularly employed for image processing (Krizhevsky et al., 2012). Direct application of such CNNs is stymied by a point clouds' unstructured and irregular data format. Any transformation of a point cloud may entail losing data information or metadata attributes. PointNet (Qi et al., 2017) is the first end-to-end DL algorithm that was successful for segmentation and classification of indoor point clouds without any transformation of the raw data. Although PointNet does not compete to the state-of-the-art DL algorithms (e. g., Boulch, 2020; Hu et al., 2020; Su et al., 2022) for point clouds classification; many researchers use it as a fast and readily available approach (e.g., Nurunnabi et al., 2021b) and many others have adopted its basic structure. Nurunnabi et al. (2021b) showed that it is promising for large-scale outdoor point clouds classification. Excluding, the spatial transformer network, T-Net (Jaderberg et al., 2015), the basic PointNet consists of only two modules: (1) max pooling (a symmetry function) that makes global point cloud features, and (2) local and global aggregation that joints local and global point feature information. PointNet ingests each point independently and learns points' features using a set of multilayer perceptrons (MLPs) followed by max pooling (see Qi et al., 2017 for additional details).

3. PROPOSED METHODOLOGY

This section proposes an algorithm to develop, assess and select a DL model that has better generalization power (e.g., reducing

the well-known overfitting problem). The new algorithm builds on a 4-step sequence for the selection of training and validation data sets and for model generation. This paper also proposes a means to evaluate its generalization capability for pointwise classification (semantic segmentation) in aerial LiDAR point clouds.

3.1 Step 1: Bootstrap couples with *k*CV

This paper couples bootstrap with *k*CV to generate a multiplicity of combinations of training and validation sets. To improve generalizability of a model, first *k*CV is employed. This splits the available data into *k* (user defined) distinct folds, and groups *k*-1 folds together to make a training set. The remaining set (fold) is considered as the validation set. This process repeats for all the *k* folds and, thus, results in *k* pairs of training and validation sets. Next, independent *B* (user defined number) bootstrap samples are drawn from each of the *k* validation sets generated by *k*CV (see Fig. 1). This results in *k* training sets, and *B* validation sets for each of the *k* training data set. Sizes of the bootstrap samples are the same as the respective validation set. Since larger size of *k* and *B* will take more time for the model building process, we fix k = 5, and B = 100 to make a balance between time and desired level of performance. Larger size of *B* is suggested in bootstrap literature to obtain more accuracy for estimating sampling distributions.

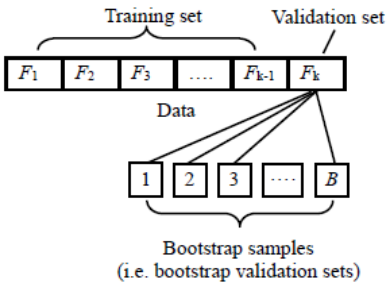

Figure 1. A schematic diagram: cross-validation couples with bootstrap to get *B* validations sets for each of *k* training sets. F_i is the *i*th (i =1,2,…,k) fold for a data set having *k* folds, where *k*-1 folds are used as a training set, and *B* bootstrap validations sets are used to validate the model developed by the training set.

3.2 Step 2: Employing PointNet and DL model development

In step 2, PointNet (Qi et al., 2017) network is employed to develop *k* distinctive DL models based on the training data sets generated by the group of *k*-1 folds. Selection of the PointNet hyper-parameters (e.g., the number of hidden layers, activations functions for the hidden and output layers, the *Adam* optimizer [Kingma and Ba, 2014]) are fixed as per the original implementation (Qi et al., 2017). Then each of the *k* models are tuned via the *B* bootstrap validation samples and used to determine the error metrics, based on the Mean Square Error (MSE), as described in Step 3.

3.3 Step 3: Calculation of evaluation and decision metrics

Step 3 defines the evaluation and decision metrics need to assess the models developed in Step 2. MSE was selected, as it is easily defined as the mean that reasonably satisfies the necessary statistical conditions to be a consistent estimator. Moreover, it behaves asymptotically normal following the Central Limit Theorem (Boos and Stefanski, 2013). Nurunnabi and Teferle (2022) demonstrated MSE as a statistically consistent estimator, and showed its potential in DL model evaluation in large-scale point clouds. MSE is used as the cost function for the model building process, and its related functions as the model evaluation metrics. The error metrics: mean (M_{MSE}), standard error (SE_{MSE}), and the 95% confidence intervals (CI_{MSE} 95%) of the MSE are calculated to evaluate the developed models. These most common estimators of signifying statistical accuracy are calculated following the standard procedures of nonparametric bootstrap (c.f., Efron and Tibshirani, 1993), where

$$SE_{MSE}(bt) = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}(MSE_b - M_{MSE})^2}\,, \qquad (1)$$

where $SE_{MSE}(bt)$ and MSE_b are the standard error of MSE for the *B* bootstrap samples, and the MSE for the *b*th bootstrap sample, respectively. The terms MSE (·) and M_{MSE} are defined as Eqs. 2 and 3, respectively:

$$MSE_b(\hat{\theta}) = \frac{1}{B}\sum_{b=1}^{B}(\hat{\theta} - \theta)^2, \qquad (2)$$

where θ is an estimator (error metric); estimated from the empirical distributions based on the *B* independent bootstrap samples, and

$$M_{MSE} = \frac{1}{B}\sum_{b=1}^{B} MSE_b. \qquad (3)$$

There are several ways to estimate bootstrap CI (e.g., Thomas and Efron, 1996). The 95% CI of the bootstrap MSE is determined statistically based on percentile values. The *B* bootstrap MSE values are arranged in an ascending order to find the 2.5th and 97.5th percentiles. The 95% bootstrap CI ($CI_{MSE_B}95\%$) can be defined as Eq. 4.

$$MSE_b^{(0.025\times B)} \le MSE_b \le MSE_b^{(0.975\times B)};\ b=1, 2, \ldots, B. \qquad (4)$$

3.4 Step 4: Model assessment and selection

In step 4, the apparently best model is selected and then assessed for the available test data. Selection is based on the model with the least M_{MSE} and/or SE_{MSE} alternative to the highest Mean of Overall Accuracy (M_{OA},) among the *k* models. The goal is to find the bootstrap validation set corresponding to the model having

Algorithm: *k*CV-B	
Step	Input: point cloud. Output: a DL model.
1.	Define a DL (PointNet) network with its regular hyper-parameters
2.	Split the data into *k*-folds (see Fig. 1)
3.	for i = 1, …. , k do
4.	Train the DL model without the *i*th fold
5.	Draw *B* bootstrap validation data sets from the *i*th fold of same size with replacement
6.	for b = 1, … , B do
7.	Evaluate the model from Step 4 with the validation set *b*
8.	Store MSE_b
9.	Store the validation set *b* with the least MSE_b
10.	Calculate M_{MSE}, SE_{MSE} and $CI_{MSE_B}95\%$
11.	Find the best training set for which M_{MSE} and/or SE_{MSE}, are the least, and best validation set for which MSE_b is the least among the others corresponds to the best training set.
12.	Retrain the DL model using the best training set and the best validation set from Step 11.
13.	The final *k*CV-B based DL model derived from Step 12.

the least error. Next, the DL algorithm is retrained using the previously determined best training and best validation sets. Finally, the well-known F_1-score and OA metrics (Nurunnabi et al., 2021a, b) are used to evaluate the DL classification model. The pseudocode for the proposed algorithm *k*CV-B is defined in Algorithm *k*CV-B.

4. EXPERIMENTS, ANALYSIS AND EVALUATION

This section demonstrates the new algorithm (*k*CV-B) through two real world ALS data sets, and compares the outputs to those achieved with three existing methods: (1) train/test split, (2) bootstrap, and (3) *k*CV.

4.1 Experiments on the DALES data set

For the first experiment, the large-scale aerial LiDAR data sets DALES (Dayton Annotated LiDAR Earth Scan; Varney et al., 2020) is used. These data are of the city of Surrey in British Columbia, Canada and were acquired by a Riegl Q1560 dual channel LiDAR system from the flying height of 1,300m. They are arranged in 40 tiles, each of 500m×500m with a point density around 50/m^2. The data were labelled with 9 groups: ground, vegetation (Veg.), car, truck, power line (PL), fence, pole, building, and unclassified (uC). The data covered semi-urban and urban areas. Prior to usage, this data set was denoised by a robust statistical method as was proposed in Nurunnabi et al. (2015). We randomly select 5 parts of data sets of almost equal size from five different tiles for training and validation that have 6,070,267 points. Three more data sets are taken from 3 different tiles as the three hold out test sets.

Next, the PointNet algorithm was applied with its regular hyper-parameters for all the concerned data and applied to all four methods: train/test split, bootstrap, *k*CV and *k*CV-B. The input attributes included the tuple of point coordinates (x, y, z), return number, point height, scan angle and normalized x, y and z values (Nurunnabi et al., 2021b). A block size of 10m×10m having 2,048 points per block was selected. A batch size of 32 was selected, and MSE was used as the loss function instead of the cross entropy used in the original PointNet. The DL model is trained with 100 epochs. To perform train/test split and bootstrap, 80% of the points were randomly selected for training, and the remaining 20% were left for validation. The model was developed and evaluated excluding three test sets that were reserved for later testing and compare to the other methods. For *k*CV, 5 folds (k = 5) were used for training and validation, which are taken from 5 different tiles. For the new method, *k*CV-B, 100 (B) bootstrap samples were drawn from each of the k validation sets of same size. Train/test split, *k*CV, bootstrap and *k*CV-B were evaluated 1, 5 (k), 100, and 500 ($k\times B$) times, respectively with the corresponding validation sets. Hence, the proposed *k*CV-B are evaluated with the maximum number of validation data sets.

We calculate MSE values for every model w. r. t. the respective validation sets, and estimate M_{MSE}, SE_{MSE}, and $CI_{MSE_B}95\%$, these are available for the bootstrap, *k*CV and *k*CV-B. We find the fold of validation sets for which the values of M_{MSE} and/or SE_{MSE} are minimum. Next the best bootstrap validation set was established for which MSE_b is the least among the others corresponds to the best training set. The final *k*CV-B model was selected based on the best bootstrap validation set and the respective training set. The same process is then done with and without shuffle before splitting (folding) them to get training and validation sets. Table 1 presents the results obtained during both the model building process and the final model tested on three reserved (previously unused) data sets. Fig. 2 plots the line diagram for the MSE values for the bootstrap samples (with and without shuffle) corresponding to different validation sets (for folds: i, ii, iii, iv and v) for *k*CV-B.

Results in Table 1 (Columns 3-5) and plots a, b in Fig. 2 explore that in most cases; the shuffled data produce better results (lower MSE values) than the unshuffled data. With the shuffled data, *k*CV-B is able to achieved the overall minimum of M_{MSE} (0.02553) and SE_{MSE} (0.00007). This was achieved with fold v (*k*CV-B v) as the validation fold. When retrained *k*CV-B (final) was able to achieve an OA of 87.1% and 83.4% for the given data set with and without shuffle, respectively. In Fig. 2, Plot c shows that 95% of CI (red vertical lines) that holds the mean of MSE (cyan vertical line) values for kCV-B. Note that, bootstrap and train/test split approaches are typically applied to shuffled data. So, testing with those two approaches was not done for unshuffled data (Table 1). Final models were assessed on the three reserved test sets (Test 1, Test 2, and Test 3).

	Methods	M_{MSE}	SE_{MSE}	$CI_{MSE_B}95\%$	OA or M_{OA}	OA (Test 1)	OA (Test 2)	OA (Test 3)	M_{OA}
Without shuffle	*k*CV-B i	0.02800	0.00014	0.0277, 0.0283	83.9	81.8	83.1	83.0	—
	*k*CV-B ii	0.04933	0.00021	0.0489, 0.0498	68.5	81.5	84.5	81.8	—
	*k*CV-B iii	0.04693	0.00027	0.0464, 0.0475	69.7	70.7	75.2	68.5	—
	*k*CV-B iv	0.11242	0.00033	0.1118, 0.1130	36.5	82.6	81.9	81.7	—
	*k*CV-B v	0.04392	0.00023	0.0435, 0.0444	72.4	80.1	82.5	82.0	—
	kCV-B (final)	**0.03137**	—	—	**83.4**	**82.8**	**83.8**	**84.6**	**83.7**
	*k*CV	0.08247	0.07544		62.2	79.6	80.2	79.2	79.6
With shuffle	*k*CV-B i	0.03033	0.00009	0.0302, 0.0305	81.9	79.3	79.9	77.8	—
	*k*CV-B ii	0.02856	0.00008	0.0284, 0.0288	82.6	81.6	81.1	81.1	—
	*k*CV-B iii	0.02857	0.00008	0.0284, 0.0287	82.6	81.8	80.6	82.9	—
	*k*CV-B iv	0.02691	0.00009	0.0267, 0.0271	83.7	81.9	81.5	81.1	—
	*k*CV-B v	**0.02553**	**0.00007**	0.0254, 0.0256	84.8	81.9	81.5	83.9	—
	kCV-B (final)	**0.02230**	—	—	**87.1**	82.6	82.8	**85.4**	**83.6**
	*k*CV	0.02700	0.00105	—	83.8	81.3	81.4	**80.4**	**81.0**
	Bootstrap	0.02794	0.00009	0.0277, 0.0281	83.1	81.7	79.2	**80.8**	**80.6**
	Train/test split	0.03167	—	—	80.5	81.1	77.2	**79.8**	**79.4**

Table 1. Results of different methods for the validation data sets from different folds, and 3 test data sets. *k*CV-B (.) mentions *k*CV-B method when (.) is the fold used for validation.

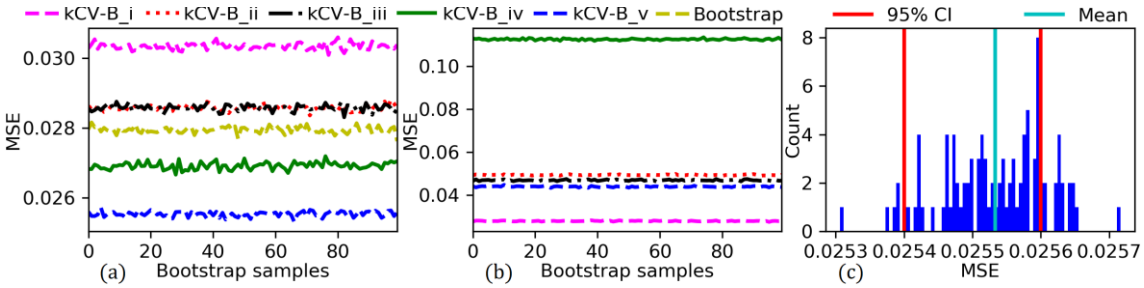

Figure 2. Exploration of the MSE values for the bootstrap samples from different folds and validation sets: (a) line diagrams for the shuffled data, (b) line diagrams for the unshuffled data, (c) histograms for the MSE values for the vth fold *k*CV-B v with 95% CI (red vertical line) and the mean (cyan vertical line).

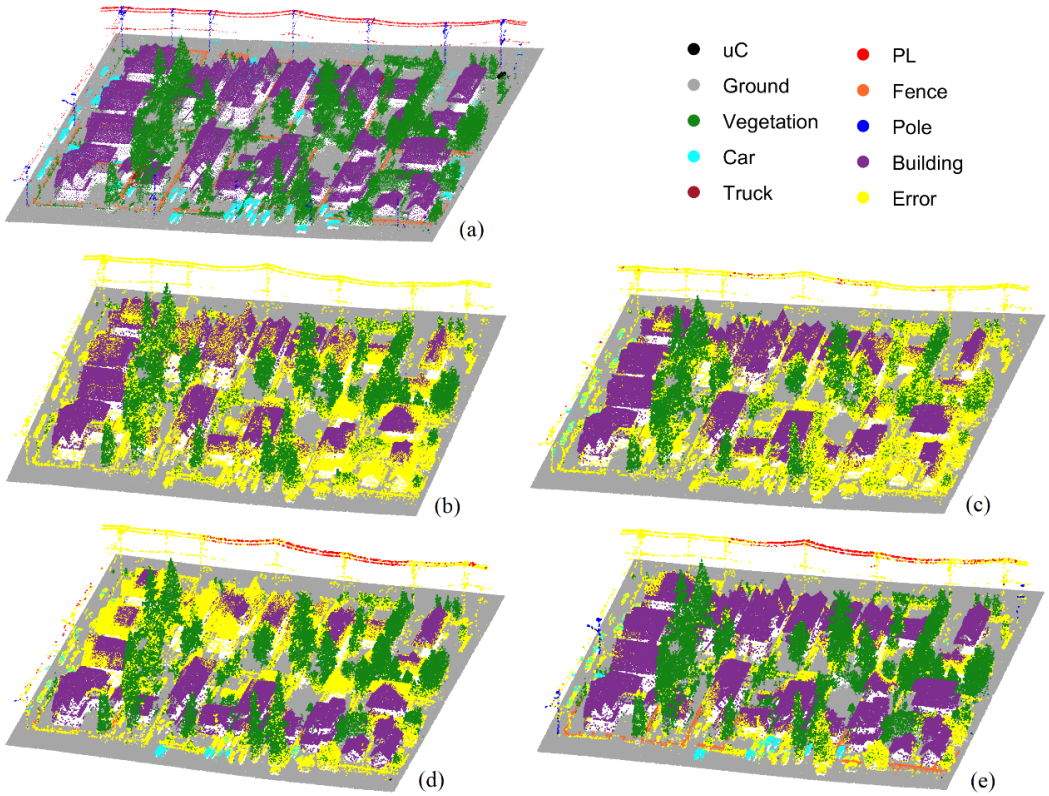

Figure 3. Classification results (misclassified points are in yellow) for the DALES Test 3 data set: (a) ground-truth, (b) train/test split, (c) bootstrap, (d) *k*CV, and (e) *k*CV-B. uC, Veg and PL define unclassified, vegetation and power line, respectively.

Class	Training points	Test points	Train/test split	Boot-strap	*k*CV	*k*CV-B
			F_1	F_1	F_1	F_1
uC	38,124	1,613	00.0	00.0	03.0	04.1
Ground	2,693,961	485,011	88.8	89.7	93.3	93.2
Veg	1,612,293	196,347	59.6	55.1	56.8	66.6
Car	121,440	18,819	04.5	15.1	22.2	32.8
Trucks	17,116	1,350	00.0	15.1	05.3	07.8
PL	17,042	3,866	00.0	11.5	53.2	51.5
Fence	29,824	13,040	00.0	00.0	06.1	11.4
Poles	6,489	2,211	00.0	00.0	16.6	18.4
Building	1,533,978	226,369	**72.9**	**77.2**	**68.2**	**83.6**
Mean F_1			25.1	29.3	36.1	41.0

Table 2. Classification results of DALES Test 3 data set.

The *k*CV-B (final) achieved the highest OA in all cases, and overall, M_{OA} was 83.6% versus 81% for *k*CV, 80.6% for bootstrap and 79.4 % for train/test split. Results clearly show that *k*CV-B achieves better generalization power than the existing methods.

Table 2 shows the per class classification performance for the Test 3 data set in terms of an F_1-score (F_1), which is a combination of precision and recall. For most of the classes *k*CV-B performed better than the others. For example, in the category of building, *k*CV-B identified points at an F_1-score of 83.6%, whereas, *k*CV, bootstrap and train/test split achieved only 68.2%, 77.2% and 72.9%, respectively. Table 2 also presents the number of points per class. In classes with significantly lower numbers of points (e.g., PL and poles), the two non-*k*CV approaches performed very poorly. Critically, the two *k*CV-based approaches were much less sensitive to this well-known problem of imbalanced data (Nurunnabi et al., 2021b). Fig. 3 visualizes this clearly for the powerlines. Not only does this point to a better

robustness in this class of solutions but may be also indicative of the need for less training data. While not the focus of this paper, the topic of data size selection has been clearly established as an open question for point cloud data (Majgaonkar et al. 2021).

4.2 Experiments on the AHN data set

In the second experiment, aerial LiDAR data from *Actueel Hoogtebestand Nederland* version 3 (AHN3) were used. These data cover the entirety of The Netherlands and managed into 500m × 500m tiles. The point density is less than half of the previous data set at around 20pt/m^2. Most tiles were pre-labelled with 5 classes: ground, vegetation, building, water, and bridges. For this study, they were relabelled into only three classes: ground, building and unclassified (uC includes vegetation, vehicles and others). This was done to reduce the imbalance in number of points per class. Common ways to reduce the effects of imbalanced data include use of oversampling and adding noise (Xie et al., 2019). Point distributions for AHN data set are in Table 4.

For the initial training and validation data, five tiles were selected to cover landscape variations of urban and semi-urban areas consisting of different objects (e.g., big and small buildings, vegetation and vehicles). Data were also selected from one additional tile and held in reserve. The five tiles used for training and validation contained 5,472,556 points in total. The reserve tile for test contained 3,276,800 points.

We perform the PointNet algorithm for the same data splitting procedures with network inputs: point coordinates (x, y, z), intensity, return number, point height, and normalized x, y and z values. The hyper-parameters were used as described for the first experiment. In this experiment, we investigate our objectives with shuffled data. We see that results of M_{MSE} (0.02596) and SE_{MSE} (0.00015) are the lowest for kCV-B ii. That means, for the fold-ii, bootstrap produces better samples for validation sets that produce corresponding OA of 94.9%, which is better than any other of the rest of the four folds. We search for the best validation set among the 100 bootstrap validation samples of fold-ii that produces kCV-B of OA = 95.4%, whereas train/test split, bootstrap and kCV produce OA of 95.3%, 94.7%, 93.9%, respectively. Now, we use the final kCV-B based model for the test data set that achieves OA of 91.3%. Interesting finding is that although train/test split produces competitive results for the existing validation set with OA of 95.3%, but for the test data it gets only OA of 81.6% which is because of its low generalization capability, as it evaluates the developed model against only one validation set. Classification results of the AHN test data set for all four methods are plotted in Fig. 4.

Methods	M_{MSE}	SE_{MSE}	$CI_{MSE_B}95\%$	OA/M_{OA}	OA (Test)
kCV-B i	0.02699	0.00017	0.0267, 0.0274	94.7	89.3
kCV-B ii	**0.02596**	**0.00015**	0.0256, 0.0262	**94.9**	90.1
kCV-B iii	0.02619	0.00019	0.0258, 0.0265	94.8	72.9
kCV-B iv	0.03100	0.00015	0.0307, 0.0313	93.9	91.7
kCV-B v	0.04616	0.00023	0.0458, 0.0465	90.9	60.3
***k*CV-B**	0.02417	—	—	**95.4**	**91.3**
kCV	0.03093	0.00626	0.0257, 0.0416	**93.9**	89.2
Bootstrap	0.02713	0.00016	0.0268, 0.0274	**94.7**	83.5
Train/test split	0.02424	—	—	**95.3**	**81.6**

Table 3. Results of different methods for the validation data set from different folds and one test data set. kCV-B (.) mentions kCV-B method when (.)th fold is used for validation.

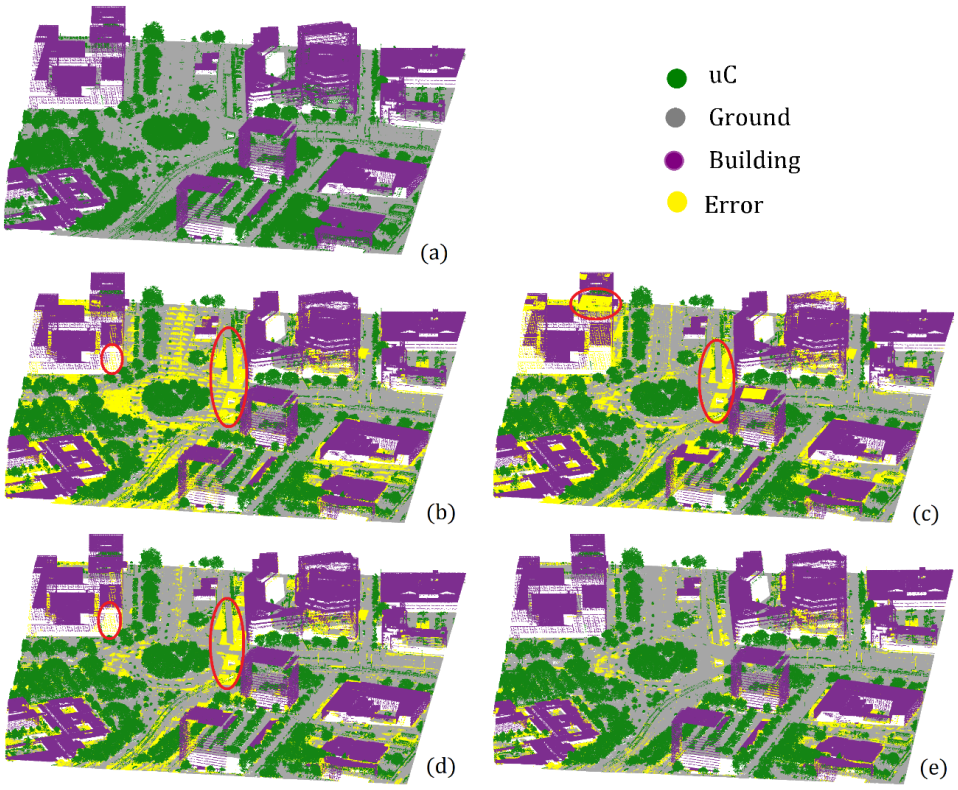

Figure 4. Classification results (misclassified points in yellow) for the AHN test data set: (a) ground-truth, (b) train/test split, (c) bootstrap, (d) kCV, and (e) kCV-B. Many building and ground points are misclassified in red ellipses.

Fig. 4, plots (e) and (b) portray the best and the worst classification results that are produced by *k*CV-B and train/test split methods respectively. Many building and ground points in the red ellipses were misclassified for train/test split, bootstrap and *k*CV based methods. Plot (e), *k*CV-B wrongly classified some of the points, but these are significantly less in numbers than the others. One-point worth noting that is clearly visible in Fig. 4 is that many of the areas in which *k*CV-B continued to struggle with the classification are in the areas of low vegetation, especially when that vegetation is close to a building. This problem was initially identified by Aljumaily et al. (2015) in the application of ML techniques to point clouds.

Class	Training points	Test points	Train/test split	Boot-strap	*k*CV	*k*CV-B
			F_1	F_1	F_1	F_1
uC	1,878,838	693,778	85.2	80.3	84.3	89.2
Ground	2,152,235	1,680,188	81.9	85.5	91.1	93.2
Building	1,441,483	902,834	78.9	82.9	**89.9**	**89.2**
Mean F_1			81.9	82.9	88.5	90.5
OA			**81.6**	83.5	89.2	**91.1**

Table 4. Classification results of AHN test data set.

Table 4 contains per class classification performance for the test data set in terms of F_1-score, Mean F_1 (MF_1) and OA. For all the classes *k*CV-B achieves better OA than the others. The one exception was for buildings, where *k*CV achieved 89.9% versus 89.2% for *k*CV-B. In all other instances and in overall F_1-scores *k*CV-B outperformed the other methods.

5. DISCUSSIONS

Using the PointNet architecture the proposed algorithm *k*CV-B was applied for per point classification (semantic segmentation) through two aerial LiDAR point clouds of significant differences in average point density (20 and 50 pts/m^2). Both experiments showed that the new algorithm, *k*CV-B classified better than the existing methods and was either less sensitive to imbalance classes of training data or potentially needed less data to achieve reasonable results. When the results of Experiment 1 are plotted solely as a function of the number of training points (Fig. 5), potentially further insight is gained as to the minimum number of training points that may be needed for the various approaches and for the different classes of object. As no consideration is given here to average point density per object or per square meter of object or as to its three-dimensional versus two-dimensional nature, further generalization would only be speculative, but this fully establishes further research needs in making ML and DL approaches more rational in terms of parameter selection including the size of training data sets. In case of Experiment 1, for the training data, results were significantly better, when we did shuffle before splitting the available data to have training and validation data sets. Although, it was not clearly supportive for the test data sets, it is reasonable that data shuffling can produce better generalization power because appearing the validation set(s) is(are) not limited to any specific part of the data, it can consider points from every part of training sets. If we do partition (fold) based on different spatial regions, it is practical that we may miss certain types of objects and classes that are not present equally in every region. For example, usually distribution of

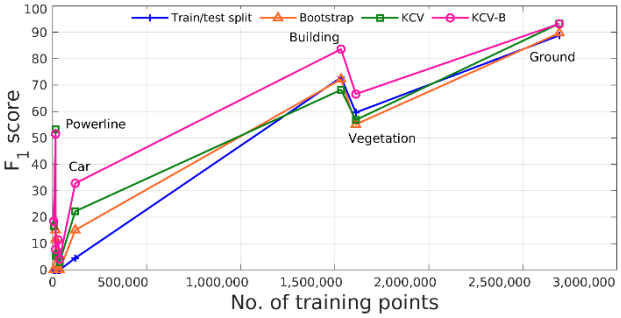

Figure 5. Experiment 1; F_1-score results plotted solely as a function of available training data points.

vegetation and buildings are different based on landscape and location (e.g., urban or rural area). However, *k*CV-B unshuffled can run faster than the shuffled *k*CV-B.

6. CONCLUSIONS

Supervised learning on point clouds, especially DL, is known to need vast amounts of labelled data, which is often not feasible. Data insufficiency is influenced by the problem of overfitting. As such, this paper investigates the potential use of a bootstrap resampling algorithm for new data creation for efficiently generating validation sets to enhance the generalization power of a DL algorithm for point cloud classification. The proposed bootstrap coupling with *k*CV was demonstrated to improve model quality. The new algorithm, *k*CV-B needs to optimize the values of *k* and *B*. The user can fix the values depending on their data and study. Using large values of *k* and *B* improve the generality and performance of a model, but there is a trade-off between generalization, accuracy and time to compete the process. Reasonably, *k*CV-B takes more time than the existing methods, but researcher who needs more accuracy and has available high-powered computing facilities would be benefited incorporating bootstrap with CV for more data generation that can produce higher generalization power for the test and future data. Further studies will investigate more on different bootstrap approaches that can be faster, more efficient and robust for new data generation and effective for large-scale data sets.

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. Additional assistance was provided through the National Science Foundation award 1940145.

REFERENCES

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

Aljumaily, H., Laefer, D., Cuadra, D., 2015. Big-data approach for three-dimensional building extraction from aerial laser scanning. *J. Comp. Civil Eng., ASCE*, https://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000524.

Basiri, S., Ollila, E., Koivunen, V., 2017. Enhanced bootstrap method for statistical inference in the ICA model. *Signal Process*, 138, 53–62.

Becker, C., Rosinskaya, E., Häni, N., d'Angelo, E., Strecha, C., 2018. Classification of aerial photogrammetric 3D point clouds. *Photogramm Eng Remote Sensing*, 84(5), 287–295.

Bishop,C. M., 2006. *Pattern Recognition and Machine Learning.* Springer, New York, USA.

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

Boulch, A., 2020. ConvPoint: Continuous convolutions for point cloud processing. *Comput Graph*, 88, 24–34.

Daszykowski, M., Walczak, B., Massart, D. L., 2002. Representative subset selection. *Anal Chim Acta*., 468, 91–103.

Davison, A., Hinkley, D., 1997. *Bootstrap Methods and their Application.* Cambridge Univ. Press, Cambridge.

Efron, B., Tibshirani, R., 1993. *An Introduction to the Bootstrap.* Chapman and Hall, New York.

Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT press.

Harrington, P. D., 2017. Multiple versus single set validation of multivariate models to avoid mistakes. *Crit Rev Anal Chem*., 48, 33–46.

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni N., Markham, A., 2020. Randla-Net: Efficient semantic segmentation of large-scale point clouds. *IEEE CVPR*, 11108–11117.

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*, Springer.

Kingma, D. P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. ImageNet classification with deep convolutional neural networks. *Adv Neural Inf Process Syst*, 1097–1105.

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

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

Majgaonkar, O., Panchal, K., Laefer, D. F., Stanley, M., Zaki, Y., 2021. Assessing LiDAR training data quantities for classification models. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci.*, Vol. 46, 101-106.

Montavon, G., Samek, W., Müller, K-R., 2018. Methods for interpreting and understanding deep neural networks. *Digit. Signal Process.*, 73, 1–15.

Nurunnabi, A., Belton, D., West, G., 2012. Robust segmentation for multiple planar surface extraction in laser scanning 3D point cloud data. *IEEE ICPR*, 1367–1370.

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., Sadahiro, Y., Lindenbergh, R., Belton, D., 2019. Robust cylinder fitting in laser scanning point cloud data. *Measurement*, Vol. 138, 632–651.

Nurunnabi, A., Teferle, F. N., Li, J., Lindenbergh, R. C., Hunegnaw, A., 2021a. An efficient deep learning approach for ground point filtering in aerial laser scanning point clouds. *Int. Arch. Photogramm.Remote Sens. Spat. Inf. Sci*, XLIII-B1,31–38.

Nurunnabi, A., Teferle, F. N., Li, J., Lindenbergh, R. C., 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., 2022. Resampling methods for a reliable validation set in deep learning-based point cloud classification. *Int. Arch. of the Photogramm. Remote Sens. and Spat. Info. Sci.*, Vol. XLIII-B2-2022, 617–624.

Puzyn, T., Mostrag-Szlichtyng, A., Gajewicz, A., Skrzyński, M., Worth, A. P., 2011. Investigating the influence of data splitting on the predictive ability of QSAR/QSPR models. *Struct Chem*., 22, 795–804.

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

Raschka, S., 2020. Model evaluation, model selection, and algorithm selection in machine learning. arXiv:1811.12808v3

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, 108272.

Taylor, B. J., 2005. *Methods and Procedures for the Verification and Validation of Artificial Neural Networks*. Springer-Verlag, New York, Inc., Secaucus, NJ, USA.

Thomas, J. D., Efron, B. 1996. Bootstrap confidence intervals. *Stat Sci*., 11(3), 189–212.

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.

Tuia, D., Persello, C., Bruzzone, L., 2016. Domain adaptation for the classification of remote sensing data: An overview of recent advances. *IEEE Geosci. Remote Sens. Mag.*, 4(2), 41–57.

Varney, N., Asari, V. K., Graehling, Q., 2020. DALES: a large-scale aerial LiDAR data set for semantic segmentation *IEEE CVPR Workshops*, 186 –187.

Wainer, J., Cawley, G., 2021. Nested cross-validation when selecting classifiers is overzealous for most practical applications. *Expert Syst. Appl*., 182, 115222.

Weidner, L., Walton, G., 2021. The influence of training data variability on a supervised machine learning classifier for Structure from Motion (SfM) point clouds of rock slopes. *Eng Geol.,* 294(106344), 1–16.

Xie, W., Liang, G., Dong, Z., Tan, B., Zhang, B., 2019. An improved oversampling algorithm based on the samples selection strategy for classifying imbalanced data. *Math. Prob. Eng.* https://doi.org/10.1155/2019/3526539.