IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 December 2022, accepted 25 December 2022, date of publication 30 December 2022,
date of current version 12 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3233411

RESEARCH ARTICLE

Evaluation of Class Distribution and Class Combinations on Semantic Segmentation of 3D Point Clouds With PointNet

EIKE BARNEFSKE AND HARALD STERNBERG
HafenCity University Hamburg, Hydrography and Geodesy, 22335 Hamburg, Germany
Corresponding author: Eike Barnefske (eike.barnefske@hcu-hamburg.de)

ABSTRACT Point clouds are generated by light imaging, detection and ranging (LIDAR) scanners or depth imaging cameras, which capture the geometry from the scanned objects with high accuracy. Unfortunately, these systems are unable to identify the semantics of the objects. Semantic 3D point clouds are an important basis for modeling the real world in digital applications. Manual semantic segmentation is a labor and cost intensive task. Automation of semantic segmentation using machine learning and deep learning (DL) approaches is therefore an interesting subject of research. In particular, point-based network architectures, such as PointNet, lead to a beneficial semantic segmentation in individual applications. For the application of DL methods, a large number of hyperparameters (HPs) have to be determined and these HPs influence the training success. In our work, the investigated HPs are the class distribution and the class combination. By means of seven combinations of classes following a hierarchical scheme and four methods to adapt the class sizes, these HPs are investigated in a detailed and structured manner. The investigated settings show an increased semantic segmentation performance, by an increase of 31% in recall for the class Erroneous points or that all classes have a recall of higher than 50%. However, based on our results the correct setting of only these HPs does not lead to a simple, universal and practical semantic segmentation procedure.

INDEX TERMS 3D point clouds, data hyperparameter, hierarchical class combination, hyperparameter, PointNet, semantic classes, semantic segmentation, unbalanced data.

I. INTRODUCTION

Scenes of the real world are scanned with depth imaging cameras and light imaging, detection and ranging (LIDAR) scanners in a short time with high geometric resolution and accuracy [1]. The digitized scenes are mostly unsorted, unstructured and incomplete point clouds [2], [3], which form the basis of a geometric model. These kind of models are useful in a wide variety of applications such as, urban planning, tourism marketing, indoor navigation, robotic control, autonomous driving, building construction planning, building operation, heritage preservation, archaeological investigations, forestry and agriculture, or infrastructure maintenance [4], [5], [6], [7], [8]. The creation of these models is often done by hand, because humans are excellent at interpreting visualized 3D point clouds and identifying semantic objects within them. Automated modeling by an algorithm requires that each point carries semantic features that can be used to form discrete semantic objects in a scene. Extending the point cloud with semantic features is semantic segmentation. The automated semantic segmentation is often performed by Machine Learning (ML) and Deep Learning (DL) approaches, which are a current research topics [4], [7], [9], [10].

DL-methods for semantic segmentation of 2D images achieve very high accuracies, but cannot be simply applied to point clouds due to the above mentioned properties. Many approaches exist where the point cloud is first transformed into an order and structure [11], [12]. However, point-based methods such as PointNet [13] or RandLA-Net [14] omit this step and can perform a semantic segmentation directly from the original point cloud. In order to use these semantically

The associate editor coordinating the review of this manuscript and approving it for publication was Vlad Diaconita.

segmented point clouds to create a building information model (BIM), the semantic point segments must meet certain accuracy requirements that arise from the model specifications [15]. These accuracy requirements are defined in the Level of Accuracy (LoA) [16], Level of Detail (LoD) [17] or the Level of Development (LoDev) [18]. For a BIM of the level *LoA2* (15 mm to 500 mm) or *LoDev 200* (design planning) and higher, the point cloud segments often cannot fulfill the geometric or semantic requirements, so that an improvement of the semantic segmentation step is necessary. Considering the complexity of the point cloud datasets, the automatic semantic segmentation is a key processing step for an efficient modeling.

Increased accuracy of these semantic segmentation methods is possible with training data [19] and Hyperparameters (HPs) [20], in addition to the adaptation of the network architecture. HPs are selected before training and commonly prior knowledge is used for the selection. They control and influence the training progress [20]. In this work the influence of the HPs *Unbalanced class distributions* and different *Class combinations* are investigated using the established network architecture PointNet (Section IV). For this purpose, four data- or algorithm-based methods for harmonizing class sizes are applied and adapted. In addition, a hierarchical class definition for frequent classes in a BIM is developed and applied. Further central contributions of this work are:

- A review of HP determination methods, data augmentation methods, and hierarchical semantic segmentation methods (Section II).
- The creation of a new medium-sized dataset that is suitable for BIM applications (Section III).
- The systematic evaluation of data augmentation methods and of hierarchical class combinations (Section V and VI).

All findings are summarized in Section VII and an outlook on further investigations is given.

II. STATE OF THE ART

A ML model is influenced by a large number of HPs. One key challenge when working with complex ML methods is to fully capture these HPs and to define the optimal values for them, as investigated by [21], [22], [23], and [24]. In Fig. 1, the relevant HPs for semantic segmentations are grouped into six clusters. The top row represents general HPs that relate to network architecture, regulation, optimization, and initialization [25]. In the bottom row (area with the blue background), Data Hyperparameters (DHPs) are shown. The DHPs depend on the data characteristics and not on the chosen model.

DHPs can be distinguished according to semantic, structural, geometrical and spectral characteristics of the dataset. The semantic characteristics of point clouds are described by [26], [27], and [28]. In terms of structural characteristics, the definition of point neighborhood [29], data augmentation [30], and the unbalanced class distribution for training data in general (e.g., images) [31] are topics that have already been investigated in other studies. Generally, geometrical and spectral features of point clouds are often used as training data by manual augmentation of point feature spaces [32]. These hand drawn features are point normals, eigenvalues, density values or mixed features [33], [34], [35]. So far, only few studies on the unbalanced class distribution and hierarchic semantic segmentation in point clouds are published. An overview of them is presented in Sections II-B and II-C.

A. PointNet

DL-models for semantic segmentations of point clouds are usually distinguished by the input formats into which the point cloud is transformed. A categorization is presented in [36]. In their work, a categorization is made into discretization-based / structure-based (e.g., as voxel), projection-based (e.g., 2D-image), and point-based (e.g., raw points or graph) methods, which can further refined (Fig. 2). While initially discretization-based [37], [38], [39] and projection-based methods [40] were predominantly used, nowadays most of the (non-real-time) models are point-based [41]. Point-based methods use the unordered points themselves to perform semantic segmentation.

One of the most widely used point-based method is PointNet [13]. PointNet addresses the structural disadvantage of the point cloud format when processing them with DL-methods. This means that points do not have to be placed in a fixed order prior to processing. They can be arranged free in orientation and position in space.

The full functionality of PointNet is explained in the first published article from the developers [13] and in many reviews such as [43] and [44]. In the following, the central processing steps of PointNet are presented for a better understanding of our investigations. Furthermore, the limitations of PointNet will be outlined.

1) PROCESSING STEPS OF PointNet

Processing with PointNet can be divided into three main steps. In the first processing step, the features are transformed into a uniform n-dimensional space using an affine transformation (with the T-Net module of PointNet). The transformation parameters are learned by the network. This transformation ensures that all input blocks are nearly at the same position and almost have the same orientation. An example with a point cloud of a chair is given in Fig. 3. This transformation is repeated after the first extraction of depth features, so that the depth features are also aligned in the complex feature space (e.g., 64 dimensions) [13].

The second processing step is the extraction of depth features-based on the input features (e.g., 3D coordinates, point normals or color values) or previous depth features. This is done using different transformation layers or a multi-layer-perceptron [13]. In most implementations of PointNet, a 1D or a 2D convolutional layer is used. As shown in Fig. 4a, the rows of the tensor are equal to the number of block points and only one column is occupied. The features of the points are arranged in the depth layer of the tensor.

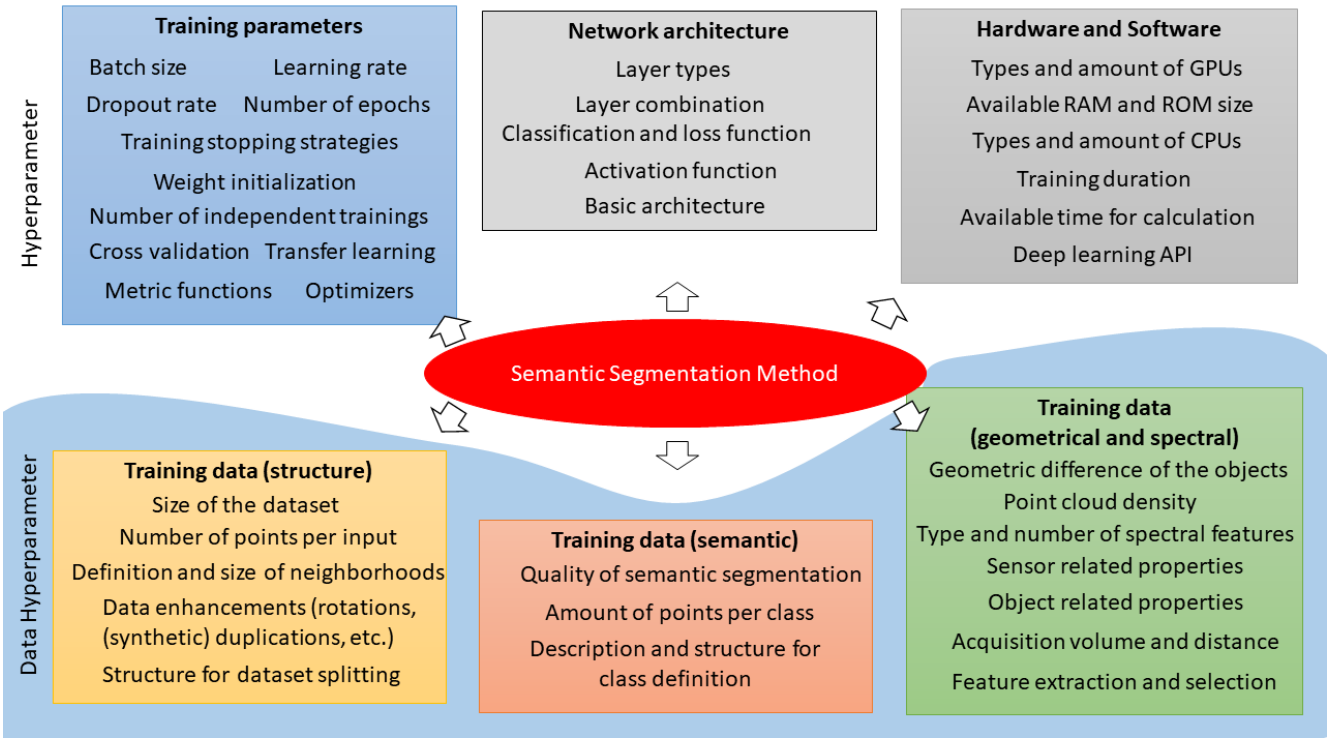

FIGURE 1. Influencing variables and parameters for the development of a DL-based semantic segmentation method. The parameters and influencing variables shown are a selection and might be adapted for other applications.

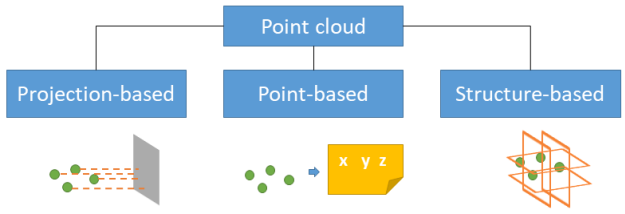

FIGURE 2. Preparation of point clouds for semantic segmentation with DL, by projection into image space, organization into a 3D structure, and usage of the raw point cloud. (Figure taken from [42] and adapted.)

Each convolutional filter contains only one value (which is fixed within the convolution), so the depth features are based only on the previous features of a point (Fig. 4a). Depending on the implementation, different numbers of convolutional layers and filters are used.

The third processing step is the aggregation of the features of the individual points into a global feature vector for the respective input block. This is done using the max-pooling function, in which only the largest value is kept for each feature (Fig. 4b). It results in a feature vector that can be used for classification of the point cloud block. For segmentation, this global feature vector is taken and appended to all individual point feature vectors. There is now a combination of inter-point and global features for each point, from which further depth features are generated. The depth features are used to classify each point (e.g., with a softmax function) [13].

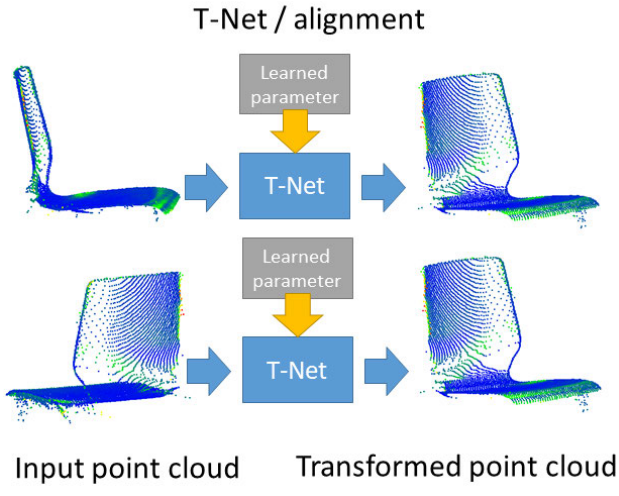

FIGURE 3. Intention of the T-Net module is that a point cloud is always aligned in a similar way by means of an affine transformation.

2) LIMITATIONS AND ADVANCEMENTS OF PointNet

The central problem with PointNet is the selection of points for a block. This for instance is the case, if the area to be segmented semantically is very large, the point densities are in-homogeneous or different frequent classes are included. Regarding this challenge, different extensions, such as PointNet++ [45] or a systematic neighborhood searches,

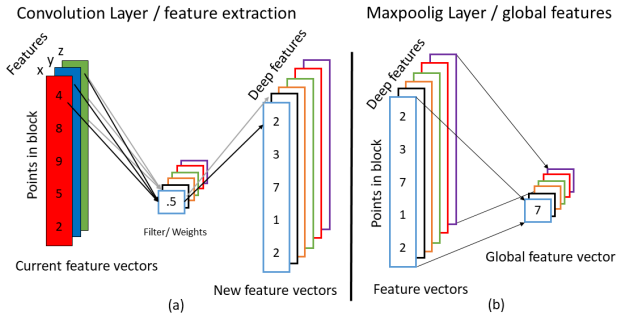

FIGURE 4. **Convolution (a) and max-pooling (b) functions for feature enhancement and aggregation with PointNet.**

such as by [46] have been developed. However, these developments also encounter limitations with extra-large and highly detailed datasets.

B. UNBALANCED CLASS DISTRIBUTION

One major concern for the semantic segmentation task solved with ML methods is, that different semantic classes in real-world data consist of different numbers of individual data objects [47]. For example, the background of an image is described by the majority of individual pixels and therefore it is learned more frequent by most ML algorithms. Often the algorithm learns only the background, because this way the highest accuracy is achieved for the whole dataset [31].

Basically, this problem exists for all ML methods, such as Support Vector Machine, k-Nearest-Neighbor (kNN), K-Mean Clustering, Convolutional Neural Network (CNN) and all kind of data types, such as data series, images, image databases or point clouds [48]. Various methods are developed to solve the class imbalance problem for certain data types. These methods can be clustered into four method groups (Fig. 5).

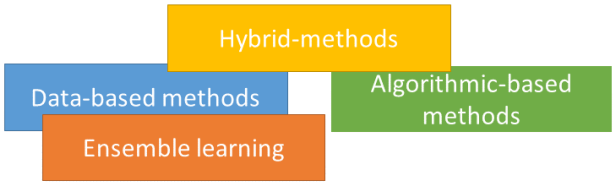

FIGURE 5. **Four method groups to address the problem of unbalanced class distribution. Arranged according to similarities of methods.**

The first method group, the data-based methods, encloses all methods, which actively change the number of the individual data objects (e.g., points or images). The dataset is filtered or augmented in such a way that the number of objects between different classes becomes equal or more similar.

Methods that reduce the number of data objects are referred as under-sampling (US) methods. These methods randomly [49] or systematically [50], [51], [52] select data objects per class to establish equality and ensures that only original (measured) data is used. The US methods have the central disadvantage that parts of the knowledge are not used and therefor learning is only performed on a subset of the information. The use of only a subset could lead to changes in the local neighborhood [31].

Unlike US, random [49] or systematic [48], [53], [54], [55] over-sampling (OS) methods enlarge the dataset and augment it with artificial or duplicated data. One popular method is the Synthetic Minority Oversampling Technique (SMOTE) by [56], where the neighborhood is considered to control the OS method. The datasets become large without gaining any new knowledge. In addition, typical data objects in the infrequent classes are emphasized strongly, therefore the trained model may not transfer well to new unknown datasets. Methods that minimize the adverse influence of both approaches use the original and OS datasets in different training phases [57] or combine the OS and US methods, each based on the epoch result [48].

The second group of methods are the **algorithm-based methods**. They modify the learning algorithm and aim for a stronger impact of classes with fewer data objects. That can be either done by adapting the loss function [58], [59], [60], modifying the network architecture [61], [62], [63], [64] or weighting the predictions [65], [66]. Learning can advantageously be done directly with raw data. But, if the class differences are very large, weighting can lead to a wrong relationship and a minor class may becomes too dominant.

The third group of methods are named as **hybrid methods**. They apply data- and algorithm-based methods together. The data are combined in a first phase at the level of ordinal features [47] or at the level of derived features to obtain highly differentiable features in the training data. The features can be created by grouping the initial features and deriving new features [67]. Other approaches create embedded features and adjust it in favor of the minor class [68] or taking into account possible high and low classification probabilities based on the feature distribution within the classes and its boundaries [69]. Hybrid methods are applied to CNN such that remaining differences due to the equalization of class sizes or optimization of the data are made by adjusting a loss function or using multiple loss functions.

The last method group is **ensemble learning**. These methods are applied to traditionally weak learning methods, such as Decision Trees or K-mean clustering. In ensemble learning, different classifiers or the same classifier are trained with different combinations of data or parameters. The results of all classifiers are evaluated to a combined result using hard or soft voting [70] (stacking and bagging). Boosting, as in *SMOTEBoost* [71], can be used as an alternative. Here, after each run, the classification parameters (e.g., selection of training data) are adjusted so that more attention is payed to hard-to-learn features.

C. HIERARCHICAL CLASS COMBINATION FOR SEMANTIC SEGMENTATION

The term *hierarchical semantic segmentation* is used in two definitions. The first definition is about the geometric size change of the segments. The segments can grow (segments

are merged) or shrink (segments are split) in the integrative and hierarchical segmentation process. The names and numbers of the classes always remain the same. In this approach, the semantics is added to the segments in a subsequent classification [72], [73]. Often, the point clouds are transformed into graphs, which are gradually refined or generate local features [74], [75], [76].

The second definition focuses on different classes at different stages of the segmentation. Here, the class definition is hierarchical and the semantic information changes by each level. This definition of hierarchical semantic segmentation is less described in literature, because for a semantic segmentation usually a fix set of semantic classes is defined in advanced and the process is done in one step. Frequent and infrequent classes are determined and segmented in the same step. In contrast, if the set of semantic classes is complex and/or oriented to a predefined hierarchical semantic schema, such as the CityGML [77], the Industry Foundation Classes (IFC) [18] or a non-institutional schema [78], [79], [80] another strategy can be applied. This performs semantic segmentation in several sub-steps. Semantic schemes usually have multiple aspects, such as geometry and semantic, and are organized into LoD [17]. The semantic LoD determines which class is determined in which level. Thereby, for each point only one class should be defined in one LoD. As shown in [81], this approach can help to better distinguish semantic classes with similar geometric features that appear in different LoDs. In addition, a combination of features from different LoD can help to increase the semantic accuracy for a semantic segmentation [81].

III. DATASET

Our dataset consists of more than 76 million individual points representing 27 rooms of the HafenCity University Hamburg main building (Figs. 6, 7 and 8). A subset of the point cloud was created for this work and contains the class Erroneous Points (subset A). This subset was extended by an existing dataset without the class Erroneous Points (subset B). Subset B was originally created for the Level 5 Indoor Navigation project [82] and was reorganized and improved for our experiments. The dataset is organized by rooms, which can be selected individually. The rooms are different in terms of furnishings, usage and shapes. Seminar rooms, lecture halls, offices, coffee kitchens, corridors and entrance halls are present in the dataset.

All rooms were surveyed using terrestrial laser scanners Z+F Imager 5010 or 5016. The survey was performed with a resolution of 6 mm at a distance of 10 m and the quality level ''normal'' [83]. Small and good observable rooms up to about 75 m^2 were surveyed from a single viewpoint. Larger or winding rooms were surveyed with multiple viewpoints so that all furnishings and building parts were captured completely. Small coverage gaps (e.g., on walls or on the floor due to obscuring furniture) are present in the data and accepted if the overall geometry of the semantic classes per room can be derived from the point cloud (Fig. 9).

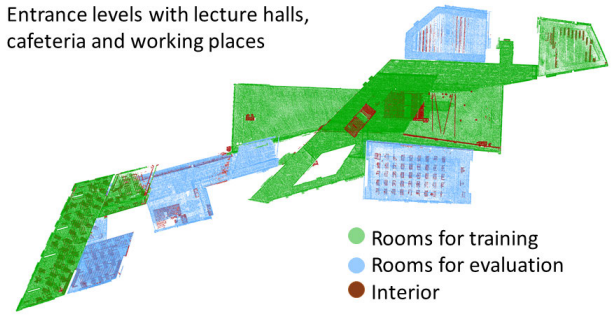

FIGURE 6. Point cloud dataset from the main building of HafenCity University Hamburg (entrance level).

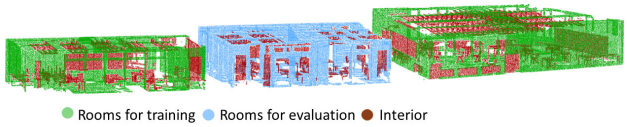

FIGURE 7. Point cloud dataset from the main building of HafenCity University Hamburg (office level).

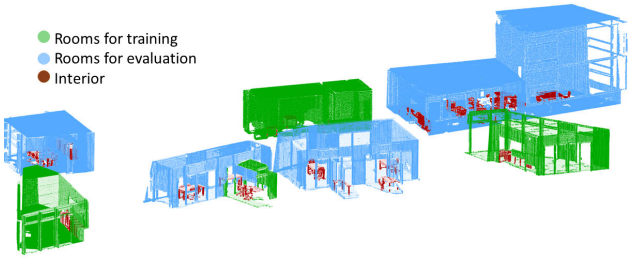

FIGURE 8. Point cloud dataset from the main building of HafenCity University Hamburg (lecture hall level).

The registration of the individual point clouds were carried out via discrete targets, which were measured automatically and manually in the scanned point clouds. Using the coordinates of a geodetic net measurement (via total station) the scanned point clouds are transferred into a global and uniform coordinate system (geo-referencing). The division by rooms was done in a manual segmentation procedure. For this purpose, the spaces are roughly selected in the entire point cloud and a partial point cloud is copied. The partial point clouds are processed so that only points of the respective room are included. This procedure leads to a more complete point cloud, because points of viewpoints in neighboring rooms are considered.

The second segmentation step is based on the semantic classes and was performed with CloudCompare [84] and Autocad Recap [85]. To achieve a high quality of the manually classified points, each point cloud was semantically segmented at least three times by different annotators. The annotators were previously trained in the task and received feedback on intermediate results. The individual segmentations of the same rooms were combined so that coarse individual errors are removed.

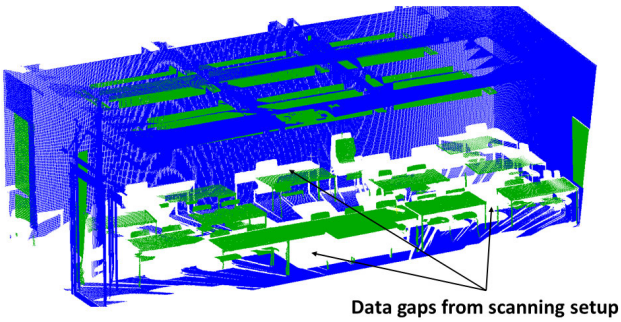

FIGURE 9. Gaps (white areas) in the point cloud caused by occlusions (e.g., furniture), and which are tolerated in the dataset.

IV. METHODOLOGY

The influence of certain DHPs, especially for point clouds are still little systematically studied. The semantic classes are usually defined according to the application, such as processing areal LIDAR point clouds for industrial use [2] or having a Scan2BIM application [4]. The semantic segmentation of all defined classes is usually achieved in one step. Reference [86] observed that the class definition and the class content have an influence on the semantic segmentation result. As an alternative to the application-oriented class definition, an algorithm-oriented class definition is also possible. This insight leads to a process in which the DHPs are set in favor to the algorithm by considering: The number of classes, the number of points per class, the presence or absence of erroneous points and the geometric difference of the objects in different classes. To investigate the DHP, an application and experimentation environment (AEE) was developed in which the common HPs and DHPs can be easily customized. The AEE offers different options for the point cloud augmentation using balancing (improvement) techniques.

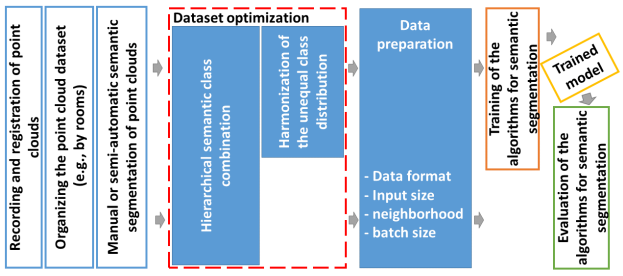

FIGURE 10. Processing steps for semantic segmentation of point clouds. The filled boxes are examined in detail.

The investigations follow the workflow shown in Fig.10. The data recording is followed by the registration, the organization of the sub-scans, and the manual semantic segmentation of the point clouds, so that these can be used as training and evaluation data. These steps are followed by a dataset optimization, which is the central focus of this work (Sections IV-B and IV-C). Next, the data is prepared for the processing step with the chosen automatic semantic segmentation method (Section IV-A) and the algorithm is trained. The efficiency of the training is evaluated with "unknown data" of the same dataset (e.g., other rooms).

A. APPLICATION AND EXPERIMENTATION ENVIRONMENT

Our work is based on the DL-architecture PointNet [13], for which optimal HPs were determined based on comprehensive preliminary investigations and literature research [41], [87]. PointNet is one of the established and foundational DL-architectures which makes our investigation results comparable with other studies. All parameters of the network architecture (except for the number of classes) remain as in the implementation of [88]. The AEE is developed that PointNet can be replaced by other point-based DL-architectures.

The main drawback of PointNet is that only a small number of points and only local features are used to assign a point to a class. Our approach to control the input of points is simple and is based on a random and uniform splitting of the point cloud into three equally sized sub-point clouds and the determination of a Local Neighborhood Box (LNB). The origin of coordinates of the entire point cloud is defined by the smallest values for the x- and y-coordinates. This origin of coordinates is used for the first sub-point cloud. For the following sub-point clouds, it is shifted in the x-y-plane by a fraction of the LNB edge length and additionally rotated by a fix angle (Fig. 11). For each of the shifted and rotated sub point clouds, the LNB are determined using the structure algorithms of *pyntcloud* library.

The local neighborhood is defined by a 1 x 1 m LNB whose height is the maximum possible room height of the dataset. By shifting and rotating, six different local neighborhoods are created for each original LNB. The rotated point cloud is an extension of the original point cloud. From each LNB a certain number of n randomly selected points is taken as network-input until all points have been fed into the network. If there are not n points left, the input is filled by random copied points from the LNB. In addition to the global normalized room coordinates (x_{glo}, y_{glo}, and z_{glo}) and the point normals (x_n, y_n, z_n), the local normalized coordinates of the LNBs (x_{loc}, y_{loc}, z_{loc}) are calculated. These nine geometric features are used as input features for all experiments.

B. METHODS FOR HARMONIZING THE UNBALANCED CLASS DISTRIBUTION

The semantic classes of point clouds from real objects differ by the number of points. Objects, such as walls and floors, take up more area (as well as points), compared to objects, such as doors and erroneous points. This is due to the fact that most surveying systems regularly scan surfaces with a fixed angular increment related to the sensor, which changes with distance. In addition, the measuring systems capture areas and not edges. Two backwards arise from the capture conditions for the training of semantic segmentation methods. First, a lot of information is collected which provides no or little new information for the separation of semantic objects. Second, there is often a lack of information about

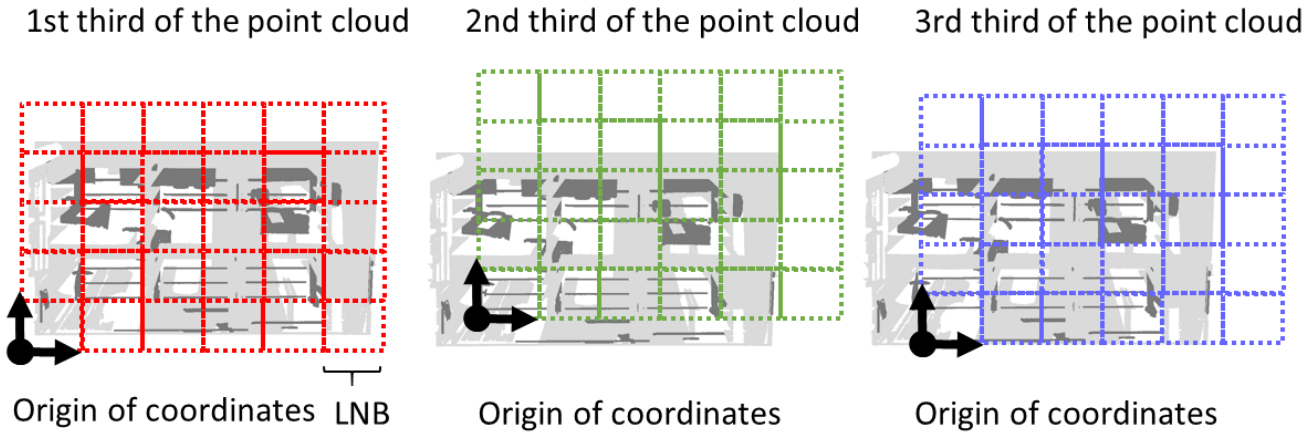

FIGURE 11. Describing the neighborhood for PointNet inputs using overlapping LNBs.

geometrically complex and variable objects, as well as of the class edge areas.

ML-based semantic segmentation methods learn a relationship between input features and semantic class over large amounts of data and try to determine an optimal separation over the majority of point features. If one class is dominant in the number of points, it can be observed that the best results are obtained by assigning almost all or all points to this class. In processing of medical images, this problem is well known by the fact that only a few pixels show the anomaly and most pixels show normal organs [89]. For our point clouds, the problem is transferable, because most points belong to frequent classes, such as wall, floor or ceiling. The underlying idea to solve this topic is to focus on the information that is important for the separation of the classes and to increase its importance. These is usually done by augmenting the points of the infrequent classes. The emphasis on the infrequent class(es) is investigated in the experimental studies of Section V-D by means of four techniques. These techniques are the SMOTE [56], the stack augmentation (SA) and two adaptions of the loss function.

1) SMOTE

In the applied implementation of SMOTE, the amount of all classes are expanded to the number of points of the largest class. Thereby, all classes consisted of the same number of points and are given homogeneously distributed into the model. Other variants of the SMOTE implementation, e.g. up to 50% of the size class or a combination with a US method could be examined alternatively. By using SMOTE, the expansion is controlled by the local neighborhood, so the later learning focus is placed on the areas of the point cloud that describe infrequent and usually more complex object classes. SMOTE uses the kNN algorithm to determine the k nearest neighbors of each point. The number of neighbors k is the factor by which the point cloud is augmented. If $k = 1$, then the point cloud is doubled. If the point cloud should be augmented to a certain number, then the multiplication number is $k + 1$. The unnecessary points have to be (randomly) deleted afterwords. For the calculation of the coordinates of the augmented points, the vector between the starting point and the nearest point is determined. The vector between this points is multiplied with a random value from 0 to 1 and added to the starting point. The coordinates of a new point are located in between both original points (Fig. 12). With SMOTE the density of the point cloud is artificially increased in the areas of the minority classes [56].

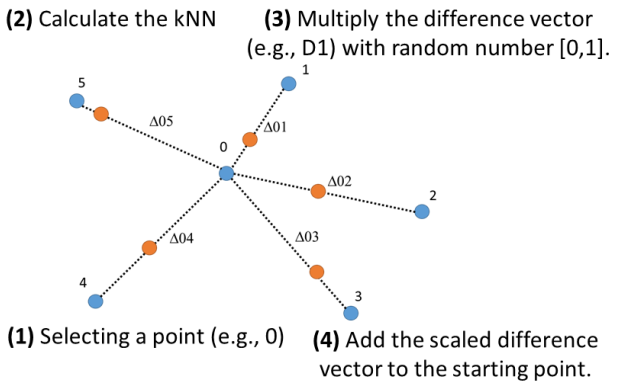

FIGURE 12. Calculations for data augmentation with the SMOTE method by [56]. In this example the point cloud is multiplied by $k = 5$ times.

2) STACK AUGMENTATION

SA is also a data-based augmentation method. However, the data is not augmented in a process ahead of DL-method and not to a fix amount of points. Instead the dataset is expanded during the creation of the training dataset. The advantage of the SA is a smaller increase of data, thus the augmentation is mainly applied to points of infrequent classes. The basic idea of the approach has been developed by [55]. They split the point cloud into chunks as network-input (similar to a voxel). Within a chunk the number of points was reduced to

a fixed amount of 4096 points. The content of the chunks is analyzed in regard to the number of points per class. Chunks with many points of infrequent classes are augmented more frequently than chunks with many points of frequent classes. The frequency of each chunk is determined by a nonlinear function. Using this data augmentation strategy, [55] are able to achieve an increase of about 10% for recall and precision for the outdoor laser scanner dataset *Semantic3D* [37].

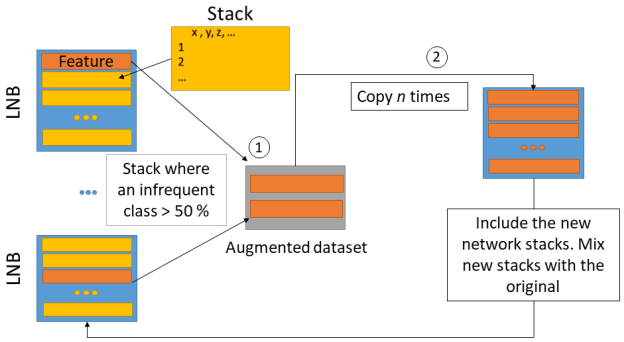

FIGURE 13. Process of stack augmentation for an optimization of the class distribution.

We adapt the method of [55] for our data processing and simplify the calculation for the augmentation factor. The augmentation and the analysis was performed on the basis of a stack with 1024 points, which is the input for the PointNet. Stacks are similar to chunks, however they do not have a fixed spatial dimension, since a stack consists of randomly selected points of an LNB (Fig. 13). The augmentation degree is determined by calculating the target proportion for each class (if all classes would be equal) and comparing it with the actual distribution. If the actual proportion of a class is smaller than the target proportion, then stacks in which this class is dominant are copied into an augmentation dataset (Fig. 13, step 1). The augmented dataset is duplicated after all stacks have been analyzed. The number of augmentations (n) is determined by the fact that the smallest class must have its target proportion (Fig. 13, step 2). For instance, the points of a point cloud should be classified into three semantic classes, so the target proportion is 33.3% to which the smallest class is augmented.

With this augmentation method, the focus should be directed to the infrequent objects in the point cloud but without losing information of large objects. Especially points in the edge zones, where small and large semantic objects meet, should be used more in the training. Stacks that contain a majority of infrequent classes should be augmented. This can be a stack, that consists only of points of the infrequent classes (Fig. 14b), but also stacks which contain few points of the frequent classes (Fig. 14a). Stacks with a majority of frequent classes are not augmented (Fig. 14c).

3) WEIGHTED LOSS FUNCTION

The third and forth methods for minimizing the unbalance class distribution are algorithm-based and addresses the loss

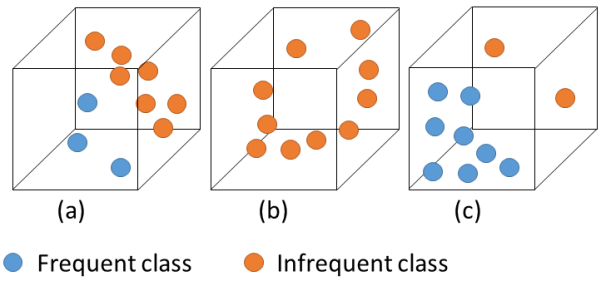

FIGURE 14. Geometric visualization of the stacks for input to a network. The black box represents the boundaries of a stack. (a) Majority of points is from the infrequent class. (b) Only points from the infrequent class are present. (c) Majority of points are from the frequent class. This stack will not be used for augmentation.

function that is used to calculate the classification error after each training pass. The loss function type used in this work is the Categorical-Cross-Entropy (CCE) loss function which is extended by two weighting options. The concept of loss calculation is shown in Fig. 15 and can be briefly described as follows.

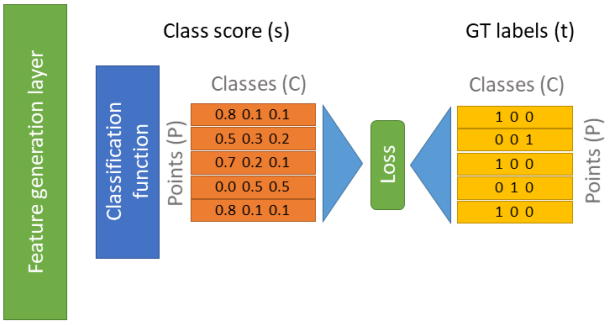

FIGURE 15. Process of feature extraction, classification and loss calculation at PointNet. Prediction class score (*s*) and ground truth label target (*t*) vector as one-hot encrypted matrix.

The raw training data given to the network is unbalanced, and the depth features are computed based on the original data. Applying a classification function (e.g., softmax), a one-hot-encode class vector for each point is determined based on the features. The class vector consists of the same number of elements as possible target classes (C) exists. In the case of the softmax function, the vector is normalized such that the vector sum is always one and the values of the vector express the probability for each class. In the predictions of the DL-method, commonly the maximum value of each point vector is determined and the one-hot encryption is decrypted. Also, the class vector is used to determine the loss during the training. For PointNet the CCE function from (1) is

commonly used.

$$CCE = -\sum_{c=1}^{C} t_c * \ln(s_c) \tag{1}$$

The CCE function is used to calculate the loss of a classification by summation of all multiplications between the logarithmized elements of the class vector (s_c) and the corresponding elements of the target vector (t_c) from ground truth (GT) label. Thereby, mean loss of each input stack and for the entire point cloud is determined. The mean loss does not distinguish whether the classes of the points are difficult or easy to learn or if the points are frequent or infrequent. Classes that occur infrequently and have a high loss are included in the mean value to a less extent than classes that occur frequently. The algorithm learns frequently occurring classes better. To minimize this disadvantage of the infrequent classes, the CCE can be improved by a weight vector (w), as described in (2).

$$WCCE = -\sum_{c=1}^{C} t_c * w_c * \ln(s_c) \tag{2}$$

The target vector (t_c) is multiplied with the weight vector (w_c), allowing the loss of the infrequent classes being emphasized in the mean loss. This loss function is called weighted CCE (WCCE) loss function and is shown in (2).

$$w_c = 1 - \frac{P_c}{P} \tag{3}$$

The calculation of these weights is usually done by the class distribution [90]. The weights in our experiments are calculated and tested using two independent experiments. In the first experiment, the proportion of a class is determined by calculating the ratio of the amount of points of one class (P_c) and calculating the amount of all points (P). The ratio of P_c and P boost the frequent classes, so it must be subtracted from 1 to emphasize the infrequent classes (3). This method reduces the loss, which can lead to a too early termination of the training phase. To minimize this reduction of the loss, the weights can be calculated according to (4).

$$w_c = \frac{1}{C} - \frac{P_c}{P} + 1 \tag{4}$$

The minor or superior proportion of each class is calculated by (4). Minor or superior proportion result from the difference to a class distribution having classes of the identical size. This leads to the fact that frequent classes get weak weights and infrequent classes get strong weights without changing the total amount of the loss.

The two WCCE functions are developed on the basis of [91] code and have been integrated as an option into the AEE. A major advantage of this method is that the weighting is only effective during training and (theoretically) the algorithm does not have to be trained again on the original data. The feature extraction and the classification itself are only indirectly influenced by learnable weights.

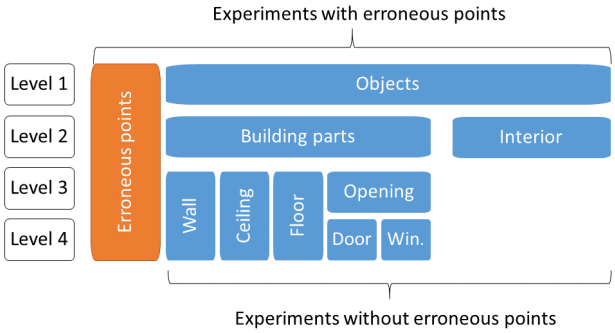

FIGURE 16. Semantic model for the examinations. A distinction is made between the main classes of Building parts, Interior and Erroneous Points. Sub-classes are considered separately starting from level 2. Each level can and cannot include erroneous points.

C. HIERARCHICAL SEMANTIC CLASS COMBINATION

The class definition specifies the semantic classes in which a point should be subdivided. The size of the individual semantic classes is indirectly given by this class definition. In applications where weak ML methods, such as Random Forrest, are used, hierarchical class definitions are used to increase the efficiency [92], [93]. A hierarchical class definition consists of several levels. General classes are defined in the top layer, which are subdivided further and further until the target classes for an application are reached. For instance in the top layer, building parts and interior can be distinguished, which can be further distinguished into classes, such as Wall, Floor, Ceiling or Window. Using a hierarchical class definition can be beneficial for the semantic segmentation because fewer distinctions in one step need to be made and the imbalance of the classes are minimized by a optimal definition. The study of [81] on PointNet++ shows that combining feature vectors from different hierarchical layers of the class definition results in a better discrimination for some classes. In their research unmanned aerial vehicle LIDAR data is analyzed and [81] state that many different semantic classes are geometrically similar. If these geometric classes are already separated by previous levels, confusion between these classes is eliminated.

Based on the work of [81] and our theoretical considerations, we developed a class definition for indoor applications, which is summarized in Fig. 16. The full hierarchical class definition is shown in Tables 13 and 14 in the appendix. By developing this, trade-offs were made between semantic reasonableness, the different geometric shapes of objects in a class, and class sizes. The goal is to form classes that are usable for a possible semantic application, that are geometrically different, and are as similar in distribution as possible. In particular, the equal class distribution is often in contradiction to other goals. These goals can possibly be achieved by combining the infrequent and geometrically similar classes Door and Windows into Opening.

In addition to the classes for real objects, the class Erroneous Points is formed as an extra semantic class for a subset of the dataset. This semantic class includes the points that are

caused by the measurement system, the measurement setup or unfavorable object properties (e.g., highly reflective). The works from [26] and [94] state that this class can have a measurable influence on the semantic segmentation results. Usually the class Erroneous points is determined with a correctness and precision of less than 20%. Even if the erroneous points are difficult to determine, such a semantic class can theoretically contribute to an improvement of the other classes [42]. In the following experiments, semantic segmentation is performed with and without this class. It should be noted that in the case of semantic segmentation without the class Erroneous Points, these points were removed from the point cloud using parameter-based filters and manually segmentation.

The semantic class definition is structured in such a way that only a subset of the points is segmented semantically in the lower levels. Thus, it is assumed that the previous level already has sufficient segmentation accuracy. In our experiments, the manually created semantic point clouds are used, so that a consideration of the maximum possible segmentation is performed. For the investigations, different splits resp. semantic generalization degrees are used in the 3rd and 4th level and all investigations were carried out for all augmentation methods of Section IV-B. The classes are split into seven combinations, the classes for each combination are shown in Table 1.

TABLE 1. Used class combinations including sub-classes.

Combination	Classes
1	Erroneous Points, Objects
2-1	Erroneous Points, Building parts, Interior
2-2	Building parts, Interior
3-1	Erroneous Points, Floor, Ceiling, Window, Door, Wall
3-2	Floor, Ceiling, Window, Door, Wall
3-3	Floor, Ceiling, Opening, Wall
3-4	Erroneous Points, Floor, Ceiling, Opening, Wall

V. EXPERIMENT SETUP

In this work, the AEE is used for analyzing the influence of the DHPs. The four data and algorithm-based augmentation methods for minimizing the influence of class size differences, as described in Section IV, are investigated in detail. In addition, the influence of a step-wise semantic class definitions is determined.

A. RESEARCH FIELD AND QUESTIONS

The semantic segmentation of point clouds makes point clouds interpretable for machines. It is one of the key steps for the automated high-accurate digitization of the real world, as performed by surveyors. From a surveyor's perspective, the data, the data quality and the DHPs are of high interest for the evaluation of different point clouds in terms of reliability, efficiency and accuracy. For the development of an automatic processing method, it is necessary to estimate the influence of the individual DHPs. The DHPs, class combination and class distribution are examined in detail in the following, in order to determine these influencing variables for the following developments or to neglect them, if they do not show a significant influence. Our investigations clarify which improvement for the semantic and geometrical accuracy can be achieved with a data or algorithm-based data augmentation method. Furthermore, we investigate if classes can be learned better by a step-wise segmentation of the point cloud.

B. HARDWARE, SOFTWARE AND HYPERPARAMETER

Training for all experiments was performed on a single workstation. The parameters of the hardware used for the computations are summarized in Table 2. The AEE was developed entirely in Python and uses Tensorflow and Keras as DL-frameworks (Table 3). Programming was preformed for a single GPU. An adaptation for a multi-GPU system is given.

TABLE 2. Hardware used for our AEE development and in the experiments.

CPU	GPU	GPU RAM	RAM
AMD Ryzen Threadripper 2970WX	GeForce RTX 2080 Ti	11 GB	64 GB

TABLE 3. Software and software versions used for our AEE development and in the experiments.

DL-Framework	Program Language	GPU Accelerator
Tensorflow 2.3.0	Python 3.8	CUDA 10.1

All experiments on one class combination with the different data augmentation methods are performed as one block of experiments. In order to compare the methods, the semantic segmentation with the original point clouds are performed for each level. The duration of the training varied from 49 to 287 minutes due to the size of the given subset and the data expansion methods. As an example, run times for all basic types of class sets are shown in Fig. 15. Longest run times were observed for the SMOTE and the SA method, since the number of points increases for both methods. A reduction of the run time was observed for the WCCEa method for the predominant cases, which can be explained by the general reduction of the loss (Fig. 17).

The initial set of common HPs were determined based on the work of [1], [2], [8], and [13] and optimized empirically. The optimized HPs are summarized in Table 4. To reduce the learning time, early-stopping was introduced. The training is stopped after 25 epochs in which the metric *eval-loss* does not decrease by more than 0.01. To optimize loss, the common *Adam* optimizer [95] is used with a learning rate that is reduced while training progresses. This should help to increase the learning efficiency. Batch size, number of epochs, points per stack and stack size were selected identically for all experiments.

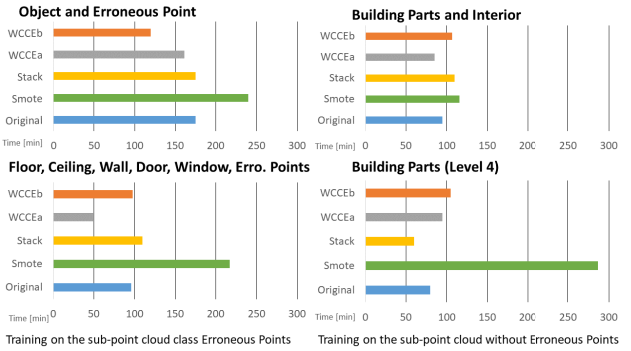

FIGURE 17. Selection of training run times for the class combinations and the different data augmentation methods.

TABLE 4. Selected HPs for all experiments.

Stack size	Batch size	Epochs	Early stopping
1024 points	16	1000	after 20 Epochs
No. of features	**Lear. rate**	**Stack dim.**	**Indep. trainings**
9	0.001 to 0.00025	1 x 1 x 6 m	9

C. EVALUATION PARAMETER

The evaluation of the semantic segmentation is carried out with the three evaluation parameters recall (RP), precision (PP) and standard deviation of the false positive (SDFP) points. The parameters *True Positive* (TP), *False Negative* (FN) and *False Positive* (FP) points are determined by comparison with the GT labels. The parameter n_C in (7) (geometric accuracy) stands for number of points for the current class. x_i is a predicted point for this class and x_{GT} is the closest point to x_i from the set of GT points for this class (reference point cloud). The main evaluation parameters are determined using (5) to (7). These scores were determined at the room level and were averaged for the analysis.

$$RP = \frac{TP}{TP + FN} \tag{5}$$

$$PP = \frac{TP}{TP + FP} \tag{6}$$

$$SDFP = \sqrt{\frac{1}{n_C}\sum_{i=1}^{n_C}(x_i - x_{GT})^2} \tag{7}$$

The SDFP points describes how precise the geometry of an object class is and can be seen as a supplementary parameter to the semantic PP. If SDFP points is greater than an application-related threshold (e.g., 100 mm), then gross segmentation errors are present. In most cases the segments of this object class cannot be used for the target application. The geometry of the segments is strongly changed (enlarged). If SDFP is smaller than the threshold, this parameter can be used to examine whether the segments are suitable for a particular LoD representation. This evaluation parameter can vary between different classes in a dataset.

Class equality (CE) for class combinations is introduced as an additional parameter and is determined using (8). Values close to 1 indicate an unequal class size and values close to 0 indicate an equal class size. The parameter CT_c is the proportion in case of an equal distribution of points per class and the parameter CA_c is the actual proportion of points per class.

$$CE = \sum_{c=1}^{P_c} \|(CT_c - CA_c)\| \tag{8}$$

A detailed class definition is available for each class combination. Further parameters concerning the point cloud quality were not considered for the analysis of these experiments. The manual semantic segmentation is reviewed for major errors.

D. PROCEDURE OF THE EXPERIMENTS

The experiment can be divided into two phases as shown in Fig. 18. In phase 1 of the experiment, 35 different experiments consisting of data augmentation methods and class combinations are studied. All experiments were initialized with random learnable parameters (weights). The weights of the network are randomly but they were used identically for all experiments, so only the influence of the training process is shown by different segmentation performances.

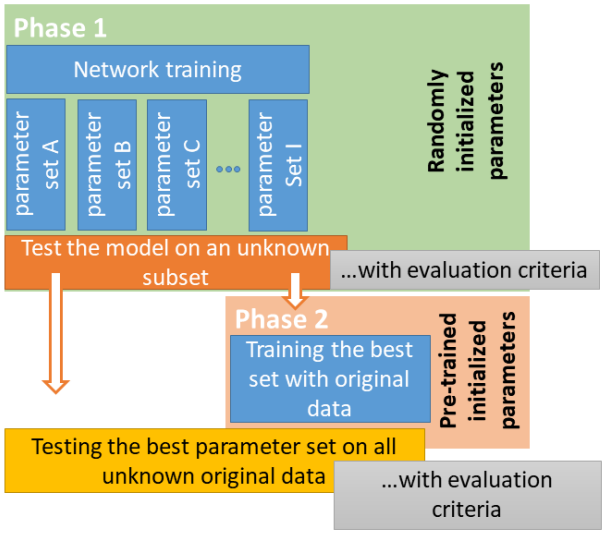

FIGURE 18. Evaluation and Transfer Learning strategy.

An analysis of the results is performed according to the scoring scheme of Fig. 19. The best weight set is used for a detailed investigation and the Transfer Learning (TL) in the second phase. In the TL phase, on the basis of the best weight set, nine new trainings are executed per combination without using a data augmentation method. The data augmentation methods SMOTE and SA change the point clouds and new unwanted patterns may appear. These patterns can adversely affect the generalization, so they should be avoided if possible. The aim of phase 2 is to describe the influence of TL.

The evaluation scheme defines that at least half of the points in each class are correctly identified and that there are

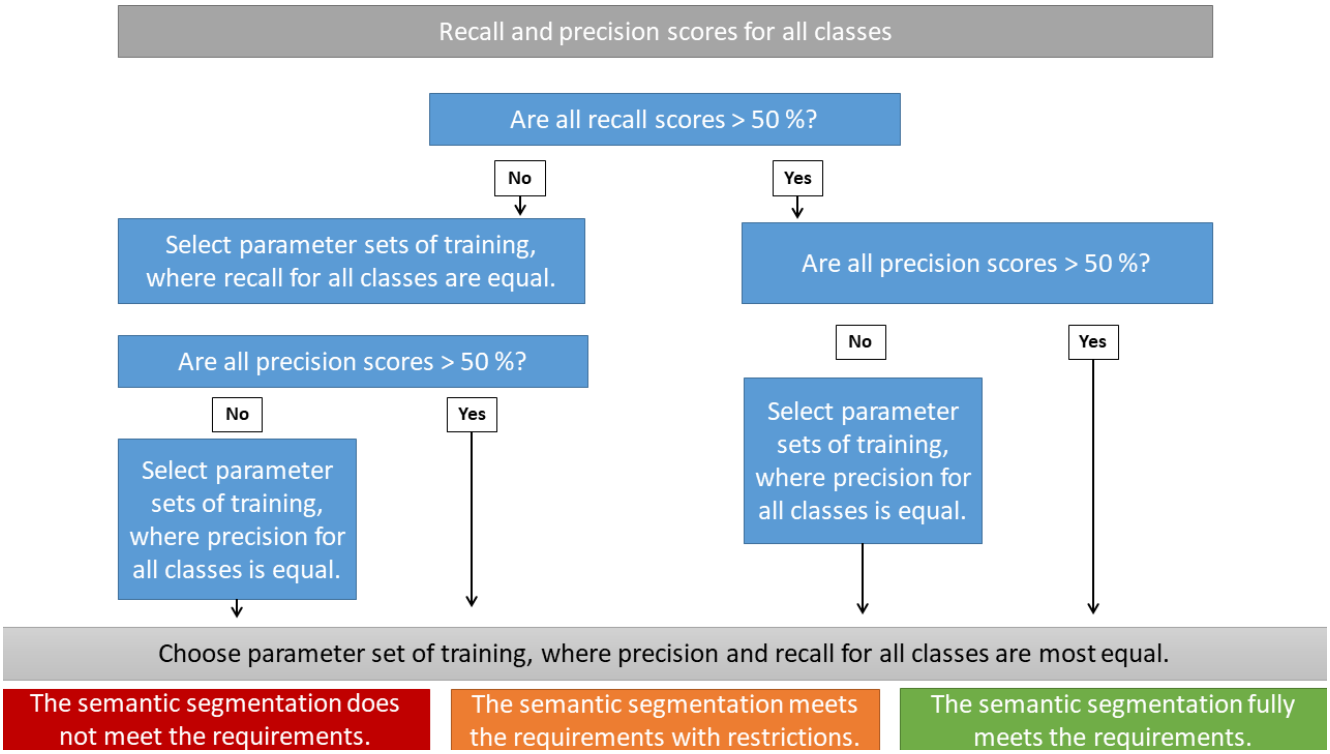

FIGURE 19. Selection scheme for Transfer Learning and semantic segmentation evaluation.

not more false points than true points in the class (''Requirement fulfilled''). This requirement seems logical from a human perspective, assuming that something is learned, if it is done more frequently correct than incorrect. However, in semantic segmentation with DL, this requirement is rarely met, and for a large number of tasks it does not have to be met. In order to evaluate which weight set provides the best performance, two additional levels have to be defined. The levels: ''Requirements meet with restriction'' and ''Requirements not fulfilled''. The requirements are meet with restrictions, if either RP or PP is less than 50% for one class. The requirements are not fulfilled, if both RP and PP is less than 50% for one class. In these cases, the best weight set is the one with the highest detectability and PP of the weakest classes. This evaluation scheme is primarily used to determine the best weight set for the subsequent comparison of the combinations.

VI. RESULTS AND DISCUSSION

Training results are stored as weight sets and can be loaded for evaluation with the full dataset. The evaluation of the weight sets from phase 1 and 2 is performed with the full test dataset or subset B, as described in Section III. From nine weight sets per combination, the weight set that preforms best to the evaluation scheme of Fig. 19 is selected and is analyzed in more detail below. In addition to the four data augmentation methods (Section IV-B), a not adapted version of the network architecture (Base method) is trained for each class combination.

The influence of the investigated HPs is expressed by the evaluation metrics RP, PP and SDFP points for each class (Section V-C) and in the form of a class average. These evaluation metrics allow a detailed evaluation for the creation of a BIM, based on a semantically segmented point cloud. The RP shows how complete a class is detected and the PP shows how many points of other classes are erroneously assigned to the considered class. For the creation of a structural model (walls, ceilings and floors), it is important that the segments of the relevant classes are as semantically precise as possible (high PP) and the predicted segments are geometrically identical to the GT segments (low SDFP of points). A complete assignment of all points can often be considered as less meaningful for this application. However, a high RP for the class Erroneous Points is very important, since all erroneous points should be removed from the data.

A. CLASS COMBINATION 1

The first examined class combination (**combination 1**) consists of the two classes Erroneous Points and Objects. According to the test procedure (Fig. 18), three (intermediate) results are available for each class combination and each augmentation method. For the SMOTE method of combination 1 (shown in Fig. 20a - d), the following evaluations are based on the best-retrained (Fig. 20c) and the TL (Fig. 20d) semantic-segmented point clouds. The Base method and the SA method do not meet the requirements. All other methods meet the requirements with restrictions from the evaluation scheme.

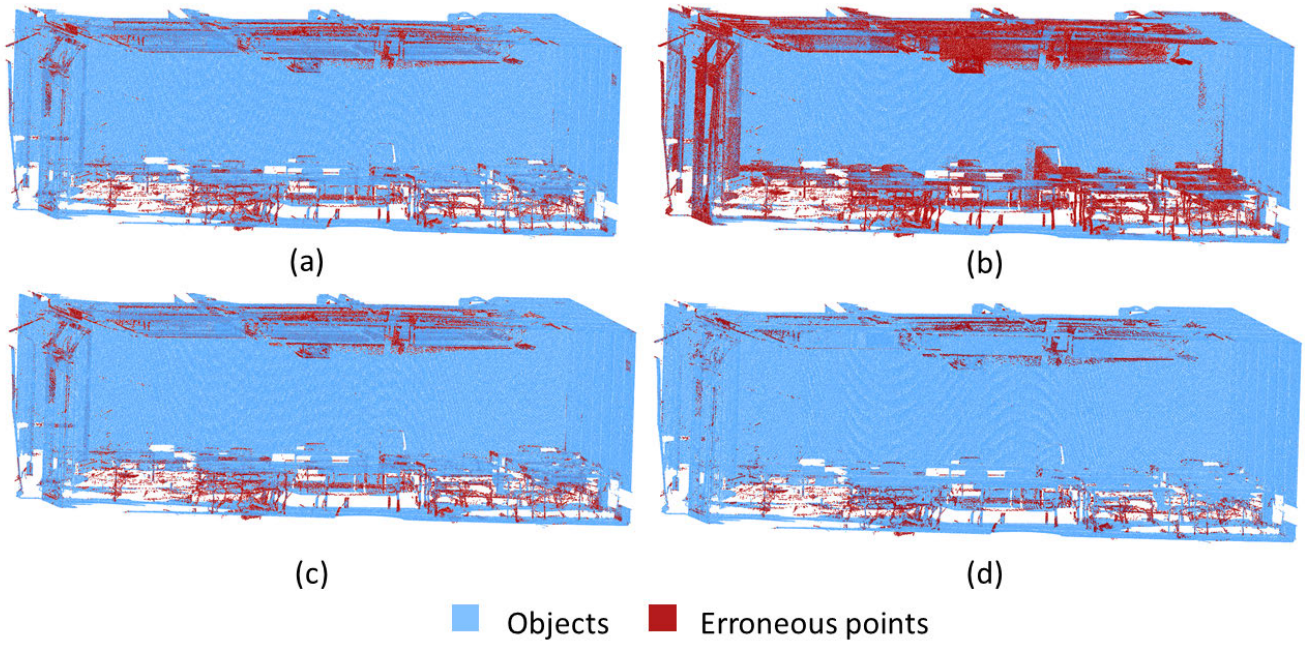

FIGURE 20. Sample point cloud of combination 1 with applied SMOTE method. (a) GT Point cloud. (b) Semantically segmented point cloud with random training. (c) Segmented point cloud by retrained best random version. (d) Semantically segmented point cloud with applied TL.

TABLE 5. Semantic accuracy of the class combination 1 (subset A). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.96	0.00	0.96	0.96	0.96
Precision					
Err. Points	47%	↓ 23%	43%	↓ 25%	39%
Objects	96%	98%	96%	99%	97%
Class average	71%	↓ 61%	↑ 96%	62%	68%
Recall					
Err. Points	44%	↑ 75%	42%	↑ 84%	↑ 59%
Objects	96%	↓ 82%	96%	↓ 82%	93%
Class average	70%	78%	69%	↑ 83%	76%

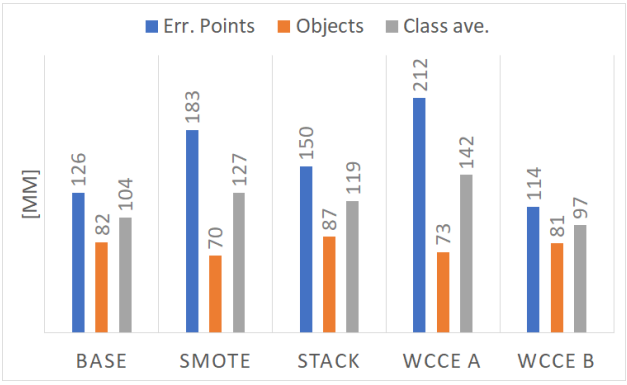

FIGURE 21. Geometric accuracy of the class combination 1 (subset A). The geometric accuracy is expressed by the SDFP points.

The CE rate is high for all methods, with 0.96, except of SMOTE. The present class combination is unfavorable for DL-based semantic segmentation.

In Table 5 it can be seen, that the methods SMOTE, WCCEa and WCCEb increase the recognizability of the small class. The erroneous points are better recognized, which leads to a more precise class Object in this binary-class case. Less precise is the class Erroneous Points for these methods and more points of the class Object are recognized as erroneous points. For the class Erroneous Points the PP decreases by 23% (Table 5). The SDFP points improve for the methods SMOTE and WCCEa by approximately 10 mm (Fig. 21). The SDFP points for the class Object is smaller than 100 mm, so that erroneous points change the object geometry at most by this amount. A model of captured structure can be created with this uncertainty. Such a model can be used for indoor pedestrian navigation or creating a rough spatial map [96].

Applying TL in phase 2, the semantic and geometric accuracy of all methods are equal to the Base method. The TL does not provide any advantage in this case. The methods SMOTE, WCCEa and WCCEb without TL improve the separation of the classes. This can be seen for SMOTE by comparing Fig. 20a with Figs. 20c and 20d.

B. CLASS COMBINATIONS 2-1 AND 2-2

The second class combination consists of the classes Interior and Building Parts, with (combination 2-1) and without (combination 2-2) the class Erroneous Points.

For **combination 2-1** (Table 6), the Base method and the SA method do not meet the requirements. All other methods meet the requirements with restrictions. These findings

TABLE 6. Semantic accuracy of the class combination 2-1 (subset A). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.76	0.00	0.54	0.76	0.76
Precision					
Interior	84%	89%	89%	88%	80%
Err. Points	44%	↓ 29%	46%	40%	↓ 34%
Build. Parts	94%	93%	93%	93%	95%
Class average	74%	70%	76%	73%	70%
Recall					
Interior	85%	↓ 65%	81%	78%	84%
Err. Points	42%	↑ 73%	48%	51%	50%
Build. Parts	94%	90%	97%	95%	88%
Class average	73%	76%	75%	75%	74%

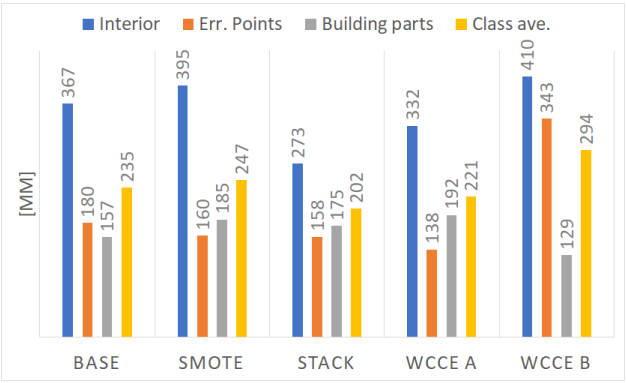

FIGURE 22. Geometric accuracy of the class combination 2-1 (subset A). The geometric accuracy is expressed by the SDFP points.

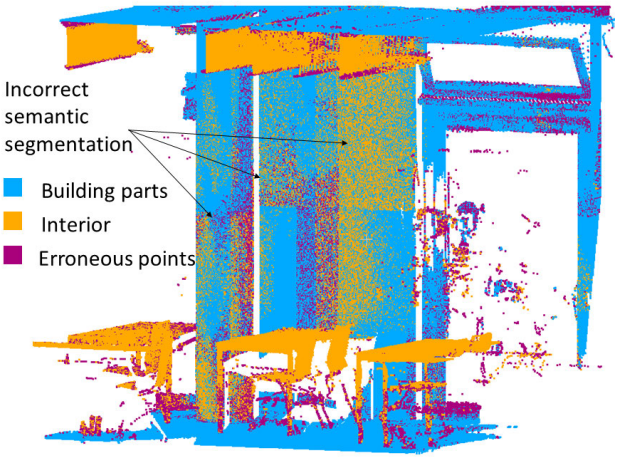

FIGURE 23. Example of major semantic segmentation errors in class combination 2-1. The parts of the wall (class Building Parts) become a segment of the class Interior.

are similar to combination 1. The CE rate for the Base, WCCEa and WCCEb methods is high with 0.76. The SMOTE method has the optimal class distribution and the SA method improves the rate to an moderate score of 0.54. The proportion of the smallest class (Erroneous Points) remains at 4% as in combination 1.

The larger classes Interior and Building Parts show a PP value higher than 80% and RP of higher than 65% (Tabel 6). The RP for these classes is decreased by the augmentation methods in favor of an increase of the erroneous points by up to 31% (e.g., for SMOTE). The RP for erroneous points of the method SA increases by 6% in comparison to the Base method at the lowest. However, the SA method is the only method with a PP higher than the Base method for all object classes and the average SDFP points is lower with 202 mm (Fig. 22). Therefore, this method achieves the highest accuracy (Table 6 and Fig. 22). The PP of the erroneous points for all other augmentation methods decrease by a maximum of 15% (e.g., for SMOTE) compared to the Base method. In comparison, the object classes have a high PP of more than 80%.

The geometric accuracy varies with a SDFP points from 129 mm to 410 mm. These SDFP points are very high and indicate major errors in the segmentation as shown in Fig. 23. Interior objects located within a range of about 200 mm from the wall cannot be reliably recognized. In the further course of the investigation, it is shown that the ceiling and floor are better separable from the furniture. A separation of wall and interior is only possible with this very high inaccuracy. The lowest SDFP points for the building parts can be obtained with the WCCEb method. For an overview model of a building, the point cloud of the class building parts can be used. This point cloud can also be used as the basis for a fast manual or parametric algorithm-based further processing.

The TL of the augmentation methods with the Base method leads overall to a small improvement of the semantic accuracy for the object class, but disfavors the class Erroneous Points by a decrease in RP.

The investigated methods lead to an improvement in the detectability of the class Erroneous Points. The recognition and PP of the object classes are not improved by the augmentation methods. The semantic PP of these classes are high as shown in Table 6. The geometric accuracy is low by LoA1 (according to the schema of [16]) and as shown in Fig. 22.

For **combination 2-2** (subset B), all augmentation methods meet the requirements with restrictions. The CE rate for the Base, WCCEa and WCCEb methods is moderate with 0.46. The SMOTE and the SA method have an optimal class distribution (Table 7). No erroneous points are included in this dataset.

RP and PP of the Base method and the augmentation methods are at the same accuracy level. For the class Building Parts, the RP of the methods SMOTE, SA and WCCEa is reduced by up to 3% in comparison to the Base method. The recognizability of the class Interior is increased by up to 11% for these methods. In this context, the SA method is the only augmentation method with an improvement in both parameters, RP and PP (Table 7). The PP of the class Building Parts is very high with as values of 96% and 97% for all methods. However, the PP of the Interior is very low with a value of approximately 30% for all methods (Table 7). Points of the

TABLE 7. Semantic accuracy of the class combination 2-2 (subset B). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.46	0.00	0.00	0.46	0.46
Precision					
Interior	33%	33%	36%	32%	35%
Build. Parts	96%	96%	96%	97%	96%
Class average	65%	65%	66%	65%	66%
Recall					
Interior	63%	68%	66%	↑ 74%	60%
Build. Parts	91%	89%	89%	87%	93%
Class average	77%	78%	78%	81%	76%

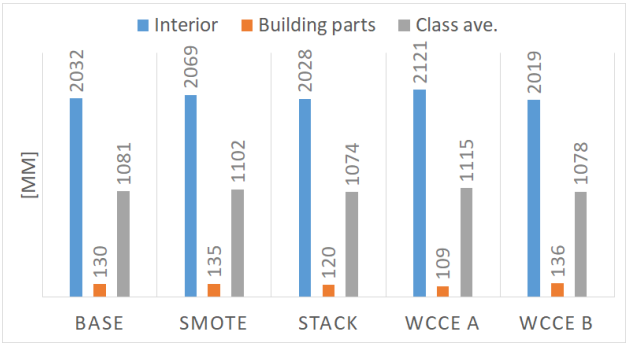

FIGURE 24. Geometric accuracy of the class combination 2-2 (subset B). The geometric accuracy is expressed by the SDFP points.

class Building Parts are sorted to a greater extent into the class Interior. This can also be seen in the SDFP points for the Interior, which are larger than 2000 mm (Fig. 24).

Subset B contains more different spaces with larger dimensions, so that larger SDFP points are also possible, as shown in Fig. 24. Furthermore, it can be observed that the SDFP points does not increase with larger rooms in subset B. Compared to subset A with erroneous points, this parameter even decreases.

The data augmentation methods do not lead to any increase in semantic and geometric accuracy for this class combination. It can be observed that the PP of the infrequent classes is not increased. The applied methods only increase the recognizability of the infrequent classes, but the discrimination is not increased. The reason for this is a lack of generalizability of the Base method for the used dataset. Rooms in the dataset differ strongly in terms of completeness, object surfaces, object geometries and sizes. An examination of the individual rooms shows that rooms with a 20 m x 20 m floor space, in which the scanner is positioned in the center, are best semantically segmented. In these rooms, there are usually only tables and chairs. Here, the RP and PP are higher than 88% for all methods. In rooms with rare objects, such as shelves, the semantic accuracy is usually less than 50%. The conditions in the different rooms influence segmentation quality strong.

The **combination 2-1** and **2-2** can not be compared, because they consist of different rooms. For a comparison

TABLE 8. Semantic accuracy of the class combination 2-2 (subset A).

	Base	SMOTE	SA	WCCEa	WCCEb
Precision					
Interior	93%	93%	92%	92%	92%
Build. Parts	95%	94%	94%	95%	95%
Class average	94%	94%	93%	93%	94%
Recall					
Interior	90%	88%	88%	90%	90%
Build. Parts	96%	97%	96%	96%	96%
Class average	93%	92%	92%	93%	93%

the subset A without erroneous points is therefore used. The results for this subset are shown in Table 8. The comparison of these data with (Table 6) and without (Table 8) the class Erroneous Points shows that the absence of this class leads to an increase of up to 23% in the semantic accuracy of the object classes. An improvement through the data augmentation methods cannot be identified in the presented investigation.

C. CLASS COMBINATIONS 3-1, 3-2, 3-3 AND 3-4

The class Building Parts is further subdivided to distinguish individual building parts (application level). The choice of classes is based on those frequently used in as-built models or in BIM applications [97], such as defined in the IFC standard [98]. The subdivision is carried out for two levels in order to examine if combining the infrequent classes door and window leads to a better semantic segmentation.

In the combination 3-1 all building parts (floor, ceiling, window, door and wall), as well as the erroneous points are included. Combination 3-2 is identical to combination 3-1 without the class Erroneous Points. In combination 3-4 the infrequent classes Window and Door are combined as Opening (level 3), all other classes remain unchanged as in combination 3-1. The combination 3-3 is identical to combination 3-4 without the class Erroneous Points. All class combinations are shown in Table 1.

For **combination 3-1**, non of the methods meet the requirements (Table 9). The CE rate for the Base, WCCEa and WCCEb methods reaches a high value of 0.84. The SMOTE method shows the optimal class distribution and the SA method improves the rate to a moderate score of 0.52.

The SMOTE method is not suitable for class combination 3-1, because the semantic accuracy is reduced compared to the Base method. The average RP is 32% and the average PP is 39%. The Base method and all other methods have a RP of more than 58% for the frequent classes. For the infrequent class Door ($<$ 1% of the dataset) the RP varies between 20% to 30%. This class cannot be learned by the methods as shown in Table 10.

The PP of the classes Floor and Ceiling is 99% for the data augmentation methods. In contrast, the PP of the class Window is very low ($<$ 45%) for all methods, because many points, especially of the class Wall and Door are assigned due to the large geometrical similarity of this class (Table 9). This low semantic accuracy correlates with a low geometric

TABLE 9. Semantic accuracy of the class combination 3-1 (subset A). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.84	0.00	0.52	0.84	0.84
Precision					
Floor	99%	↓ 70%	99%	98%	99%
Ceiling	98%	↓ 44%	99%	99%	99%
Err. Points	75%	↓ 21%	75%	68%	↓ 63%
Window	39%	↓ 5%	42%	39%	↓ 29%
Door	53%	↓ 11%	58%	54%	52%
Wall	90%	83%	90%	88%	94%
Class average	76%	↓ 39%	77%	74%	73%
Recall					
Floor	99%	↓ 10%	99%	99%	99%
Ceiling	99%	90%	99%	98%	99%
Err. Points	72%	↓ 60%	76%	73%	72%
Window	68%	↓ 3%	58%	62%	77%
Door	26%	↓ 3%	20%	30%	24%
Wall	79%	↓ 24%	85%	77%	↓ 61%
Class average	77%	↓ 32%	73%	73%	72%

TABLE 10. Semantic accuracy of the class combination 3-4 (subset A). The geometric accuracy is expressed by the SDFP points.The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.72	0.00	0.26	0.72	0.72
Precision					
Floor	99%	99%	99%	98%	99%
Ceiling	98%	99%	99%	98%	97%
Err. Points	62%	71%	↓ 42%	67%	↑77%
Wall	88%	90%	91%	90%	89%
Opening	35%	↑ 51%	44%	30%	36%
Class average	76%	82%	75%	77%	80%
Recall					
Floor	95%	99%	99%	97%	99%
Ceiling	98%	96%	98%	95%	99%
Err. Points	71%	77%	↓ 60%	66%	71%
Wall	77%	↑ 90%	↑ 88%	73%	76%
Opening	54%	51%	↓ 41%	↑ 64%	63%
Class average	79%	83%	77%	79%	82%

accuracy for this class. The SDFP points is larger than 6000 mm, so that this semantic class occupies nearly the whole room.

Since the classes Window and Door are not learned by the methods, they are combined in an intermediate step to the class Opening. The idea behind the summary is, that this class could be subdivided in the case of a good semantic segmentation in a following step, without negatively affecting the class Wall.

For **combination 3-4** the Base, WCCEa and WCCEb methods meet the requirements with restrictions. The method SA does not meet the requirements and the SOMTE class meets the requirements. The CE rate for the Base, WCCEa and WCCEb methods is high with a value of 0.72. The SMOTE method shows the optimal class distribution and the SA method improves the CE rate to a sufficient score of 0.26. Due to the rough description of the distribution for these combination, it can be seen that an increase of the semantic accuracy is achieved (Table 10). The recognizability of the

TABLE 11. Semantic accuracy of the class combination 3-2 (subset B). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.72	0.00	0.38	0.72	0.72
Precision					
Floor	97%	95%	96%	97%	98%
Ceiling	98%	96%	95%	97%	97%
Window	20%	16%	18%	21%	20%
Door	33%	25%	↓ 21%	23%	28%
Wall	76%	↑ 86%	↑ 86%	↑ 87%	↑ 87%
Class average	65%	64%	63%	65%	58%
Recall					
Floor	99%	97%	↓ 77%	94%	↓ 77%
Ceiling	72%	↑ 89%	↑ 95%	↑ 91%	81%
Window	46%	↓ 35%	↑ 68%	43%	51%
Door	4%	↑ 46%	↑ 30%	↑ 42%	↑ 37%
Wall	59%	↓ 45%	↓ 39%	↓ 40%	↓ 42%
Class average	56%	62%	62%	62%	58%

class Opening is low compared to the other classes. The PP of this class is for almost any method below 50%, and the SDFP points is higher than 3600 mm. Nevertheless, a good semantic segmentation can be performed with the SMOTE method. But it does not work for the combination 3-1.

For combinations 3-1 and 3-4, the TL phase leads to results comparable to the Base method.

For **combination 3-2**, no method meet the requirements. The CE rate for the Base, WCCEa and WCCEb methods is with 0.72 high. The SMOTE method shows the optimal class distribution and the SA method improves the rate to a moderate score of 0.38 (Table 11).

For combination 3-2, the majority of the points of the classes Door and Window are not assigned to the correct classes. In addition, the geometrically similar class Wall is less recognized compared to combinations 3-1 and 3-4. The PP of Door and Window is low with a maximum of 33% over all methods (Table 11). The geometric accuracy of the two classes has a high SDFP points. For the class Window, the SDFP points is larger than 4300 mm and for the class Door it is larger than 3600 mm. Based on these evaluation parameters, it can be stated that the class distribution has no influence in this case. A semantic segmentation with the class combination 2-2 leads to a high semantic and geometric accuracy only for the classes Floor and Ceiling. Also, the combination of the classes Door and Window to Opening in an intermediate step is tested in combination 3-3, too.

For **combination 3-3**, the methods Base, SA and WCCEb meet the requirements. The SMOTE and the WCCEa methods meet the requirements with restrictions. The CE rate for the Base, WCCEa and WCCEb methods is with value of 0.40 moderate. The SMOTE method shows the optimal class distribution and the SA method improves the rate to an score of 0.08. The class distribution becomes favorable after the consolidation (Table 12).

The combination 3-3 leads to an increase in the semantic segmentation accuracy of all classes. The RP of the class Opening is higher than 50% for all methods. The class Wall,

TABLE 12. Semantic accuracy of the class combination 3-3 (subset B). The symbols ↑ and ↓ indicate a change of more and less than 10%, resp., compared to the base method.

	Base	SMOTE	SA	WCCEa	WCCEb
Class equality	0.40	0.00	0.08	0.40	0.40
Precision					
Floor	98%	95%	94%	97%	97%
Ceiling	96%	95%	94%	96%	95%
Wall	89%	81%	84%	86%	88%
Opening	29%	30%	31%	32%	31%
Class average	78%	75%	76%	78%	78%
Recall					
Floor	68%	↑ 99%	↑ 99%	↑ 92%	↑ 96%
Ceiling	83%	↓ 69%	↑ 95%	85%	79%
Wall	34%	↑ 51%	↑ 45%	↑ 52%	↑ 46%
Opening	83%	↓ 59%	↓ 65%	↓ 67%	73%
Class average	67%	69%	76%	74%	74%

with which the class Opening is often confused, is correctly recognized only by SMOTE and WCCEa of the point majority. Based on the low PP of 29% to 32%, the confusion with the class Opening is confirmed (Table 12). Even with this combination, the neighboring classes Wall and Opening cannot be accurately separated. Larger variations for different rooms are observed here, but there are no rooms that can be segmented semantically very accurately. An influence of the class Erroneous Points is not observed.

A TL with the Base method results in an increase of 1% to 3% for RP and PP for all methods and classes.

D. SUMMERY AND OVERALL FINDINGS

The results show for the investigated settings, class definition and class combination, that two examined DHPs have only a minor influence on the semantic and geometric accuracy of semantic segmentation. The applied augmentation methods lead to an improved recognition of the infrequent classes. In the classification step, points are more often assigned to an infrequent class. This leads to a reduction in PP of infrequent classes. The SDFP points remains unchanged or even decreases due to the used augmentation methods. The PP of the segmentation improves stronger for frequent classes.

The number of classes itself has no influence on the semantic segmentation performance. Instead, the geometric similarity and the distance of the objects are important for distinguishing classes. The classes Floor and Ceiling can be well distinguished because of the large geometric distance (no shared boundary), whereas the classes Window and Wall are difficult to distinguish. When defining a class, the geometric distinguishability of the objects must be taken into account. This must be valid for the entire dataset, since rooms, for example, vary strong in size, shape and furnishing.

Using a class combination without erroneous points leads to an increase in PP and RP for the classes that already have a higher PP in a semantic segmentation with the class Erroneous Points. Classes that have a lower semantic PP in the semantic segmentation with erroneous points are recognized worse without this class and have a lower PP.

Applying an additional TL phase, where the previous result serves as a starting point for training with the Base method, does not lead to an increase in accuracy. For the SMOTE and SA methods, it result in less frequent detection of the infrequent classes and a similar performance as with the Base method.

VII. CONCLUSION AND OUTLOOK

The performance of DL-methods in semantic segmentation is influenced among other factors by HPs. In this work, the DHP, class combinations and methods to minimize the unbalanced classes have been studied. For the investigation, an AEE has been developed in which the established PointNet architecture has been implemented.

The class combinations were organized in a hierarchic order, so that a semantic segmentation is performed only for a particular part of the point cloud, for combinations in level 3 and 4. Infrequent classes were combined and semantically segmented afterwards. This resulted in higher semantic and geometric accuracy for the class Building Parts and its frequent sub classes. The class Erroneous Points leads to a slightly higher semantic accuracy for infrequent classes.

The use of two data-based augmentation methods and two algorithm-based methods only achieved a small increase in semantic recognition. The applied methods usually increase the RP, so that the infrequent classes are recognized more often and the more frequent classes become more precise. This is advantageous for the combinations in level 1 and 2, because only the more frequent classes are needed for a building modeling.

The primary goal of this work is to increase RP and PP to over 50% for all classes using the augmentation methods. This goal was only achieved for the combination 3-4 with the SMOTE method. An increase in RP to a value higher than 50% is achieved with the SMOTE method additional four times, whereas the WCCEa method fulfills it for five of the seven combinations. This increase of the RP is achieved four times with the WCCEb method. The SA method results in an increase in RP and PP, but less than 50% in most cases. With the Base method, a RP of all classes higher than 50% was achieved twice. The primary goal was partly achieved.

In the course of the investigation, it was discovered that the geometric similarity of classes must be considered when forming the class combinations. Also, the choice of LNB has a large impact on the segmentation performance. Based on our observations, the choice of the local neighborhood and the differences between the individual rooms in the dataset are highly influential. The focus of further investigations should be on these DHPs. The influence of data augmentation methods is measurable, but currently of little relevance according to our sample BIM application. In terms of augmentation methods, we plan to examine the impact of US methods as well as a combination of US methods, OS methods and weighted loss functions.

TABLE 13. Semantic class definitions for the classes of two top levels.

Class name	Sub-classes	Description
Object	Building Parts, Interior	Points of the class Objects describe a true object. They describe a surface with a small variation of a few millimeters per surface (< 10 mm).
Erroneous Points		Points of the class Erroneous Points are individual points, that appear in obscured places, that represent tails on edges and (measurement) noise around smooth surfaces (> 10 mm).
Building Parts	Door, Ceiling, Floor, Wall, Window, Opening	Points of the class Building Parts include all points that belong to the building structure. Not including: switches, lamps or boards.
Interior		Interior objects are all objects that have been brought into the building or installed in the building after the shell has been completed. Examples are switches, vents, furniture, decoration, people or measuring equipment.

TABLE 14. Semantic class definition for classes of the super-class Building Parts.

Class name	Sub-classes	Description
Wall		The class Wall is the vertical shell of a room. It can be hidden by furnishing objects. The class Wall includes baseboards. Frames of windows and doors are the horizontal boundaries. In the vertical direction, the wall is delimited by intersections with the ceiling and the floor. Window frames are not part of the wall. Free-standing columns are part of the wall.
Floor		The class Floor is defined by the lowest points that span a horizontal plane. This plane can be hidden by furnishings. Its extension is bordered by the vertical walls. Points count as a part of a plane if they do not deviate from the plane by more than 5 mm.
Ceiling		The class Ceiling is defined by the top points that span a horizontal plane. This plane can be hidden by lamps or other interior objects. It is bounded by the wall in the vertical direction. Points count as a plane if they do not deviate from the plane by more than 5 mm.
Window		Points in the class Window describe the window frames. Points in the glass areas are considered to be disturbances (erroneous points). No distinction is made between windows that can be opened and those that cannot be opened. Window sills belong to the class Window.
Door		Points of the class Door can belong to the door leaf or the door frame. The viewing windows next to the door leaf belong to the class Door.
Opening	Door, Window	Combination of classes Window and Door.

APPENDIX
CLASS DEFINITION

The class definition for the two upper levels (Fig. 14) are shown in Table 13. The class definitions for the super-class Building Parts are summarized in Table 14. This class definition is developed for a semantic segmentation as a basis for creating a BIM model of a public building.

ACKNOWLEDGMENT

Many thanks to the annotators: Clemens Semmelroth, Stefanie Stand, and Olga Konkova. Special thanks to Annette Scheider, Lena Barnefske, and Christopher Klocke for proofreading.

REFERENCES

[1] J. Zhao, X. Zhang, and Y. Wang, "Indoor 3D point clouds semantic segmentation bases on modified pointnet network," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 369–373, Aug. 2020.

[2] M. Soilán, R. Lindenbergh, B. Riveiro, and A. Sánchez-Rodríguez, "Pointnet for the automatic classification of aerial point clouds," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 445–452, May 2019.

[3] S. De Geyter, J. Vermandere, H. De Winter, M. Bassier, and M. Vergauwen, "Point cloud validation: On the impact of laser scanning technologies on the semantic segmentation for BIM modeling and evaluation," *Remote Sens.*, vol. 14, no. 3, p. 582, Jan. 2022.

[4] F. Noichl, A. Braun, and A. Borrmann, "'BIM-to-scan' for scan-to-BIM: Generating realistic synthetic ground truth point clouds based on industrial 3D models," in *Proc. Eur. Conf. Comput. Construct.*, Jul. 2021, pp. 164–172.

[5] S. Chen, J. Fang, Q. Zhang, W. Liu, and X. Wang, "Hierarchical aggregation for 3D instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 15447–15456.

[6] F. Poux and R. Billen, "Voxel-based 3D point cloud semantic segmentation: Unsupervised geometric and relationship featuring vs deep learning methods," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 5, p. 213, May 2019.

[7] Y. Xie, J. Tian, and X. X. Zhu, "Linking points with labels in 3D: A review of point cloud semantic segmentation," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 38–59, Dec. 2020.

[8] C. Morbidoni, R. Pierdicca, R. Quattrini, and E. Frontoni, "Graph CNN with radius distance for semantic segmentation of historical buildings TLS point clouds," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vols. XLIV-4/W1-2020, pp. 95–102, Sep. 2020.

[9] L. Winiwarter and G. Mandlburger, "Classification of 3D point clouds using deep neural networks," in *Proc. Dreiländertagung der DGPF, der OVG und der SGPF in Vienna*, vol. 28. Österreich-Publikationen der DGPF, 2019, pp. 663–674.

[10] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao, "Are we hungry for 3D LiDAR data for semantic segmentation? A survey of datasets and methods," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6063–6081, Jul. 2022.

[11] H. Riemenschneider, A. Bodis-Szomoru, J. Weissenberg, and L. V. Gool, "Learning where to classify in multi-view semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2014, pp. 516–532.

[12] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[14] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11105–11114.

[15] T. Bender, M. Härtig, E. Jaspers, M. Krämer, M. May, M. Schlundt, and N. Turianskyj, "Building information modeling," in *CAFM-Handbuch*. Wiesbaden, Germany: Springer, 2018, pp. 295–324
[16] *Engineering Survey*, Standard DIN18710, Sep. 2010.
[17] M.-O. Löwner and G. Gröger, "Das neue lod konzept für citygml 3.0," in *GeoForum MV*, vol. 13. Rostock-Warnemünde, Germany, 2017, pp. 23–30.
[18] (Jun. 24, 2021). BuildingSMART. *Industry Foundation Classes 4.0.2.1*. [Online]. Available: https://standards.buildingsmart.org
[19] E. S. Malinverni, R. Pierdicca, M. Paolanti, M. Martini, C. Morbidoni, F. Matrone, and A. Lingua, "Deep learning for semantic segmentation of 3D point cloud," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 735–742, Aug. 2019.
[20] D. Passos and P. Mishra, "A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks," *Chemometric Intell. Lab. Syst.*, vol. 223, Apr. 2022, Art. no. 104520. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169743922000314
[21] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," 2015, *arXiv:1502.02127*.
[22] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1—Learning rate, batch size, momentum, and weight decay," 2018, *arXiv:1803.09820*.
[23] C. Cooney, A. Korik, R. Folli, and D. Coyle, "Evaluation of hyperparameter optimization in machine and deep learning methods for decoding imagined speech EEG," *Sensors*, vol. 20, no. 16, p. 4629, Aug. 2020.
[24] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," Mar. 2020, *arXiv:2003.05689*.
[25] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine Learning*. Cham, Switzerland: Springer, 2019, pp. 3–33.
[26] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9296–9306.
[27] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu, "Shape2Motion: Joint analysis of motion parts and attributes from 3D shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8868–8876.
[28] W. Zimmer, A. Rangesh, and M. Trivedi, "3D BAT: A semi-automatic, web-based 3D annotation toolbox for full-surround, multi-modal data streams," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 1816–1821.
[29] M. Weinmann, B. Jutzi, C. Mallet, and M. Weinmann, "Geometric features and their relevance for 3D point cloud classification," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 157–164, May 2017.
[30] M. Negassi, D. Wagner, and A. Reiterer, "Smart(sampling)augment: Optimal and efficient data augmentation for semantic segmentation," *Algorithms*, vol. 15, no. 5, p. 165, May 2022.
[31] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, Mar. 2019.
[32] M. Weinmann, B. Jutzi, S. Hinz, and C. Mallet, "Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers," *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 286–304, Jul. 2015.
[33] M. Weinmann, B. Jutzi, and C. Mallet, "Semantic 3D scene interpretation: A framework combining optimal neighborhood size selection with relevant features," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 2, pp. 181–188, Aug. 2014.
[34] T. Hackel, "Large-scale machine learning for point cloud processing," Ph.D. dissertation, Inst. Geodesy Photogramm., ETH Zürich, Zürich, Switzerland, 2018.
[35] N. Shukla, *Machine Learning With TensorFlow*. Shelter Island, NY, USA: Manning, Publ., 2018.
[36] E. Camuffo, D. Mari, and S. Milani, "Recent advancements in learning algorithms for point clouds: An updated overview," *Sensors*, vol. 22, no. 4, p. 1357, Feb. 2022.
[37] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D.Net: A new large-scale point cloud classification benchmark," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 4, pp. 91–98, May 2017.
[38] S. Wirges, T. Fischer, C. Stiller, and J. B. Frias, "Object detection and classification in occupancy grid maps using deep convolutional networks," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3530–3535.
[39] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner, "ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans," in *Proc. CVPR*, vol. 1, Jun. 2018, p. 2.
[40] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7652–7660.
[41] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors*, vol. 19, no. 19, p. 4188, Sep. 2019.
[42] E. Barnefske and S. Harald, "Automatisch semantisch-segmentierte punktwolken - möglichkeiten und herausforderungen," in *DVW-Seminar MST von (A)nwendungen bis (Z)ukunftstechnologien*, vol. 103. Augsburg, Germany: Wißner-Verlag, 2022, pp. 173–184.
[43] D. Koguciuk, Ł. Chechliński, and T. El-Gaaly, "3D object recognition with ensemble learning—A study of point cloud-based deep learning models," in *Advances in Visual Computing*. Cham, Switzerland: Springer, 2019, pp. 100–114.
[44] S. A. Bello, S. Yu, C. Wang, J. M. Adam, and J. Li, "Review: Deep learning on 3D point clouds," *Remote Sens.*, vol. 12, no. 11, p. 1729, May 2020.
[45] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.
[46] F. Engelmann, T. Kontogianni, A. Hermans, and B. Leibe, "Exploring spatial context for 3D semantic segmentation of point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 716–724.
[47] H.-I. Lin and M. C. Nguyen, "Boosting minority class prediction on imbalanced point cloud data," *Appl. Sci.*, vol. 10, no. 3, p. 973, Feb. 2020.
[48] Z. Jiang, T. Pan, C. Zhang, and J. Yang, "A new oversampling method based on the classification contribution degree," *Symmetry*, vol. 13, no. 2, p. 194, Jan. 2021.
[49] J. V. Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 935–942.
[50] I. Mani and I. Zhang, "KNN approach to unbalanced data distributions: A case study involving information extraction," in *Proc. Workshop Learn. Imbalanced Datasets (ICML)*, vol. 126, 2003, pp. 1–7.
[51] M. Kubat and S. Matwin, "Addressing the curse of imbalanced training sets: One-sided selection," in *Proc. 14th Int. Conf. Mach. Learn.*, 1997, pp. 179–186.
[52] R. Barandela, R. M. Valdovinos, J. S. Sánchez, and F. J. Ferri, "The imbalanced training sample problem: Under or over sampling?" in *Structural, Syntactic, and Statistical Pattern Recognition* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2004, pp. 806–814.
[53] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM SIGKDD Explor. Newslett.*, vol. 6, no. 1, pp. 40–49, Jun. 2004.
[54] P. Hensman and D. Masko, "The impact of imbalanced training data for convolutional neural networks," KTH, School Comput. Sci. Commun., Stockholm, Sweden, May 2015. [Online]. Available: https://www.kth.se/social/files/588617ebf2765401cfcc478c/PHensmanDMasko_dkand15.pdf
[55] D. Griffiths and J. Boehm, "Weighted point cloud augmentation for neural network training data class-imbalance," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 981–987, Jun. 2019.
[56] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.
[57] H. Lee, M. Park, and J. Kim, "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3713–3717.
[58] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, and P. J. Kennedy, "Training deep neural networks on imbalanced data sets," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 4368–4374.
[59] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.
[60] H. Wang, Z. Cui, Y. Chen, M. Avidan, A. B. Abdallah, and A. Kronzer, "Predicting hospital readmission via cost-sensitive deep learning," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1968–1978, Dec. 2018.
[61] C. Zhang, K. C. Tan, and R. Ren, "Training cost-sensitive deep belief networks on imbalance data problems," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 4362–4367.
[62] Y. Zhang, L. Shuai, Y. Ren, and H. Chen, "Image classification with category centers in class imbalance situation," in *Proc. 33rd Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, May 2018, pp. 359–363.

[63] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, and R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3573–3587, Aug. 2018.
[64] W. Ding, D.-Y. Huang, Z. Chen, X. Yu, and W. Lin, "Facial action recognition using very deep networks for highly imbalanced class distribution," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Dec. 2017, pp. 1368–1372.
[65] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Netw.*, vol. 106, pp. 249–259, Oct. 2018.
[66] J. Morel, A. Bac, and T. Kanai, "Segmentation of unbalanced and inhomogeneous point clouds and its application to 3D scanned trees," *Vis. Comput.*, vol. 36, nos. 10–12, pp. 2419–2431, Sep. 2020.
[67] C. Huang, Y. Li, C. C. Loy, and X. Tang, "Learning deep representation for imbalanced classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5375–5384.
[68] S. Ando and C. Y. Huang, "Deep over-sampling framework for classifying imbalanced data," in *Machine Learning and Knowledge Discovery in Databases*. Cham, Switzerland: Springer, 2017, pp. 770–785.
[69] Q. Dong, S. Gong, and X. Zhu, "Imbalanced deep learning by minority class incremental rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1367–1381, Jun. 2019.
[70] T.-Y. Liu, "EasyEnsemble and feature selection for imbalance data sets," in *Proc. Int. Joint Conf. Bioinf., Syst. Biol. Intell. Comput.*, 2009, pp. 517–520.
[71] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "SMOTEBoost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases*. Berlin, Germany: Springer, 2003, pp. 107–119.
[72] O. Hassaan, A. Shamail, Z. Butt, and M. Taj, "Point cloud segmentation using hierarchical tree for architectural models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1582–1586.
[73] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung, "SceneNN: A scene meshes dataset with aNNotations," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 92–101.
[74] L. Jiang, H. Zhao, S. Liu, X. Shen, C.-W. Fu, and J. Jia, "Hierarchical point-edge interaction network for point cloud semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 10432–10440.
[75] Y. Li and G. Baciu, "HSGAN: Hierarchical graph learning for point cloud generation," *IEEE Trans. Image Process.*, vol. 30, pp. 4540–4554, 2021.
[76] T. Jiang, J. Sun, S. Liu, X. Zhang, Q. Wu, and Y. Wang, "Hierarchical semantic segmentation of urban scene point clouds via group proposal and graph attention network," *Int. J. Appl. Earth Observ. Geoinformation*, vol. 105, Dec. 2021, Art. no. 102626.
[77] T. H. Kolbe, T. Kutzner, C. S. Smyth, C. Nagel, C. Roensdorf, and C. Heazel, "OGC city geographymarkup language (CityGML) part 1: Conceptual modelstandard," Pen Geospatial Consortium, Arlington, VA, USA, Tech. Rep. 20-010, 2021.
[78] Y. Verdie, F. Lafarge, and P. Alliez, "LOD generation for urban scenes," *ACM Trans. Graph.*, vol. 34, no. 3, pp. 1–14, May 2015.
[79] L. Tang, L. Li, S. Ying, and Y. Lei, "A full level-of-detail specification for 3D building models combining indoor and outdoor scenes," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 11, p. 419, Oct. 2018.
[80] H. Ledoux, K. A. Ohori, K. Kumar, B. Dukai, A. Labetski, and S. Vitalis, "CityJSON: A compact and easy-to-use encoding of the CityGML data model," *Open Geospatial Data, Softw. Standards*, vol. 4, no. 1, pp. 127–140, Jun. 2019.
[81] X. Li, C. Li, Z. Tong, A. Lim, J. Yuan, Y. Wu, J. Tang, and R. Huang, "Campus3D: A photogrammetry point cloud benchmark forhierarchical understanding of outdoor scene," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 238–246.
[82] V. Stojanovic et al., "A conceptual digital twin for 5G indoor navigation," in *Proc. 11th Int. Conf. Mobile Services, Resour., Users (MOBILITY)*, Apr. 2021, pp. 5–14.
[83] *Reaching New Levels, z+f Imager5016, User Manual, v2.1*, Zoller+Fröhlich-GmbH, Wangen im Allgäu, Germany, 2019.
[84] CloudCompare. *3D Point Cloud and Mesh Processing Software Open-Source Project*. Accessed: Jun. 24, 2021. [Online]. Available: http://www.cloudcompare.org/
[85] Autodesk-Recap. *Youtube Channel*. Accessed: Jun. 24, 2022. [Online]. Available: https://www.youtube.com/user/autodeskrecap/
[86] E. Barnefske and H. Sternberg, "Evaluating the quality of semantic segmented 3D point clouds," *Remote Sens.*, vol. 14, no. 3, p. 446, Jan. 2022.
[87] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 12, pp. 4338–4364, Dec. 2021.
[88] S. Rakshit and S. Paul. (Oct. 2020). *Point Cloud Segmentation with PointNet*. [Online]. Available: https://github.com/soumik12345/point-cloud-segmentation
[89] Z. Li, K. Kamnitsas, and B. Glocker, "Analyzing overfitting under class imbalance in neural networks for image segmentation," *IEEE Trans. Med. Imag.*, vol. 40, no. 3, pp. 1065–1077, Mar. 2021.
[90] M. Abdou, M. Elkhateeb, I. Sobh, and A. Elsallab, "End-to-end 3D-PointCloud semantic segmentation for autonomous driving," 2019, *arXiv:1906.10964*.
[91] M. Barel. (Dec. 2019). *Multi-Class Weighted Loss for Semantic Image Segmentation in Keras/Tensorflow*. Accessed: Sep. 23, 2022. [Online]. Available: https://stackoverflow.com/questions/59520807/multi-class-weighted-loss-for-semantic-image-segmentation-in-keras-tensorflow
[92] E. Grilli, D. Dininno, G. Petrucci, and F. Remondino, "FROM 2D TO 3D supervised segmentation and classification for cultural heritage applications," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 399–406, May 2018.
[93] S. Teruggi, E. Grilli, M. Russo, F. Fassi, and F. Remondino, "A hierarchical machine learning approach for multi-level and multi-resolution 3D point cloud classification," *Remote Sens.*, vol. 12, no. 16, p. 2598, Aug. 2020.
[94] E. Barnefske and H. Sternberg, "Klassifizierung von fehlerhaft gemessenen punkten in 3D-punktwolken mit convnet," in *Ingenieurvermessung 20. Beiträge zum 19*. T. Wunderlich, Ed. Berlin, Germany: Herbert Wichmann Verlag, Mar. 2020, pp. 127–139.
[95] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–23.
[96] C. Schuldt, H. Shoushtari, N. Hellweg, and H. Sternberg, "L5IN: Overview of an indoor navigation pilot project," *Remote Sens.*, vol. 13, no. 4, p. 624, Feb. 2021.
[97] J. Blankenbach, *Ingenieurgeodäsie*. Berlin, Germany: Springer, 2017, pp. 23–53.
[98] BIM-Forum. *Level of Development Specification Part1 & Commentary*. Accessed: Dec. 8, 2020. [Online]. Available: https://bimforum.org/lod/

EIKE BARNEFSKE received the bachelor's and master's degrees in geomatics from HafenCity University Hamburg, and the M.Sc. degree, in 2016. He is currently pursuing the Ph.D. degree in hydrography and geodesy. From 2016 to 2017, he worked as a Research Assistant for engineering geodesy and geodetic metrology. Since 2017, he has been a Research Assistant. His research interests include analysis of mass data, such as laser scanning data, and the development of multi-sensor-systems.

HARALD STERNBERG received the degree in surveying from the University of the Federal Armed Forces Germany, Munich, in 1986, and the Dr.-Ing. degree from the University of the Bundeswehr Germany, in 1999. He worked in the administrative chain as an Artillery Officer and as a Research Assistant with the University of the Bundeswehr Germany. In 2001, he became a Professor of engineering geodesy with the Hamburg University of Applied Sciences, and from 2009 to 2017, he was a Professor of engineering geodesy and geodetic metrology with HafenCity University Hamburg, where he was also the Vice-President for studies and teaching, from 2009 to 2022. In 2017, he took over the professorship for hydrography and geodesy. His research interests include mobile mapping systems on different carriers (cars, ships, and indoor cars), the use of low-cost sensors for positioning, indoor positioning, including with 5G, monitoring of structures, autonomous underwater vehicles, automatic analysis of underwater images, interpretation of backscatter data, and analysis of mass data using artificial intelligence.