


Article

Generalized Extraction of Bolts, Mesh, and Rock in Tunnel Point Clouds: A Critical Comparison of Geometric Feature-Based Methods Using Random Forest and Neural Networks

Luke Weidner ^{1,2}  and Gabriel Walton ^{1,*}

¹ Department of Geology and Geological Engineering, Colorado School of Mines, 1400 Illinois St., Golden, CO 80401, USA; weidner@mines.edu

² BGC Engineering Inc., 600 12th St., Golden, CO 80401, USA

* Correspondence: gwalton@mines.edu

Abstract: Automatically identifying mine and tunnel infrastructure elements, such as rock bolts, from point cloud data improves deformation and quality control analyses and could ultimately contribute to improved safety on engineering projects. However, we hypothesize that existing methods are sensitive to small changes in object characteristics across datasets if trained insufficiently, and previous studies have only investigated single datasets. In this study, we present a cross-site training (generalization) investigation for a multi-class tunnel infrastructure classification task on terrestrial laser scanning data. In contrast to previous work, the novelty of this work is that the models are trained and tested across multiple datasets collected in different tunnels. We used two random forest (RF) implementations and one neural network (NN), as proposed in recent studies, on four datasets collected in different mines and tunnels in the US and Canada. We labeled points as belonging to one of four classes—rock, bolt, mesh, and other—and performed cross-site training experiments to evaluate accuracy differences between sites. In general, we found that the NN and RF models had similar performance to each other, and that same-site classification was generally successful, but cross-site performance was much lower and judged as not practically useful. Thus, our results indicate that standard geometric features are often insufficient for generalized classification of tunnel infrastructure, and these types of methods are most successful when applied to specific individual sites using interactive software for classification. Possible future research directions to improve generalized performance are discussed, including domain adaptation and deep learning methods.

Keywords: mine infrastructure; generalization; terrestrial lidar; semantic segmentation; tunnel infrastructure; rock bolt segmentation; wire mesh segmentation



Citation: Weidner, L.; Walton, G. Generalized Extraction of Bolts, Mesh, and Rock in Tunnel Point Clouds: A Critical Comparison of Geometric Feature-Based Methods Using Random Forest and Neural Networks. *Remote Sens.* **2024**, *16*, 4466. <https://doi.org/10.3390/rs16234466>

Academic Editors: Filiberto Chiabrando and Domenico Visintini

Received: 10 October 2024

Revised: 19 November 2024

Accepted: 26 November 2024

Published: 28 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Lidar point cloud data are an increasingly integral component of tunnel engineering practice, enabling inspection of site conditions at much higher accuracy and repeatability than was possible just ten or twenty years ago [1]. Fixed and mobile lidar data, in particular, are being used to extract and analyze concrete thickness [2] and cracking [3], segment relevant features of rail tunnels [4,5], identify changes through time [6,7], map rock structures [8], and characterize mine infrastructure such as rock bolts [9] and wire mesh [10]. There is also much interest in using mobile lidar in autonomous or semi-autonomous robots to improve safety [11,12].

Analysis of point clouds often requires segmentation to isolate the features of interest, which can be a manual, time-consuming task [13,14]. This has motivated the development of automated algorithms to filter and segment point clouds. The literature on point cloud segmentation algorithms has been well-established since the 1990s [15], with authors exploiting various geometric and spectral characteristics of individual points and point clusters to classify them [16]. However, early algorithms generally used small numbers

of features and/or rules [17]. In contrast, as data become easier to collect and higher-dimensional, more research has focused on using machine learning (ML) to quickly develop new problem-specific models with more features [18–20].

The two most popular ML approaches currently are (1) computing hand-crafted point/object features and applying traditional ML approaches such as random forest (RF) or knowledge-based rules to create a decision boundary (recent examples: [20–26]); and (2) various deep neural network (DNN) architectures that primarily use representation learning instead of manual feature engineering (e.g., PointNet [27] and KPConv [28]). The potential benefits of deep learning methods are improved accuracy on complex object types and the ability to control the input using natural language and pre-trained generalized models (such as the segment-anything model (SAM) applied to geospatial data [29–31]). However, two major downsides of deep learning are the need for large, annotated datasets, which are currently much less available for point clouds than for text and 2D data, and much higher computational costs for training [9,18,32]. For example, Ref. [32] compared PointNet++, KPConv, and SparseCNN and found that training alone took dozens to hundreds of hours on a GPU-equipped computer with high memory (512 GB RAM, NVIDIA RTX 2080 Ti with 11 GB RAM). In contrast, the efficiency in computing geometric features and, critically, the lower training data requirement makes them preferable for many practical applications [33,34]. However, new network architectures such as RandLA-Net are continually being developed to address this limitation [35]. A further limitation of deep learning is that the models are inherently less interpretable than traditional ML approaches, and there is resistance to their use in safety-critical applications [36]. It is, in part, due to these limitations that new applications of traditional ML continue to be explored in research. Similarly, in this study, we focus on traditional ML approaches because of their applicability in practical engineering scenarios with short timelines and limited resources.

Several recent papers have focused on the extraction of rock bolts using classical ML approaches; they are illustrative examples of the continued development of non-deep learning approaches for classification in underground excavations. Gallwey et al. [9] and Singh et al. [37] employ an approach for roof bolt identification by computing geometric features, training and applying RF and NN classification models, then using a clustering algorithm to separate bolt instances. Singh et al. [38] developed another similar approach using the CANUPO classifier [39] instead of RF or NN; however, RF and NN have shown generally higher accuracy than CANUPO [13]. Saydam [40] added a deep-learning-based classification step to this process to distinguish between true positive and false positive bolt instances. Li [41] recently created an entirely heuristic (non-ML) approach for roof bolt identification. While the ML techniques used by these sources are rarely new, they highlight that non-deep learning approaches continue to be attractive for engineering applications.

In point cloud ML research, it is increasingly common to perform generalization tests across completely separate datasets as a way to assess the real-world performance more accurately [18,42–44]. It is often found that generalized accuracy is lower than single-dataset accuracy, prompting the development of alternatives for domain adaptation to address this gap. However, to date, there have been no generalization tests performed for the task of identifying underground infrastructure such as rock bolts, which, in turn, may overestimate the real-world performance of the proposed methods. A further challenge in underground engineering contexts is that point cloud data are often collected by subcontractors and for a specific design purpose, without knowledge of how this may affect later attempts to extract certain attributes. As a result, most project-based point cloud analysis is an exercise in working with imperfect data.

With the above limitations in mind, the goals of this study are threefold: (1) To perform an independent evaluation of RF and NN classification methods from [9,37,38], focused on identifying rock bolts, wire mesh, and rock. These tests will use the same or similar geometric features as used in previous studies. (2) To perform novel testing of the classifiers across multiple different tunnels instead of just one, evaluating, for the first time, the impact of infrastructure object shape on classification accuracy and contributing to closing

a gap in the underground remote sensing literature. (3) To demonstrate the classification approach in an engineering project example. In doing this, we highlight which problem areas need additional investigation and suggest areas that could be deprioritized in future research studies.

2. Methods

The methodology of this study consists of collecting point cloud datasets from different sites, training multiple ML classifiers with different models, hyperparameters, and training site combinations, and analyzing the outputs for their ability to extract rock bolts and other objects. The overall workflow is illustrated in Figure 1, and the following sections describe each step in this process.

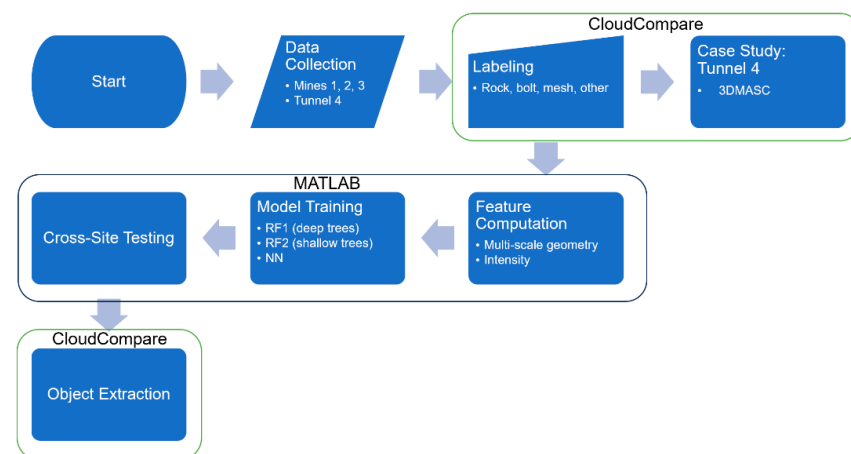


Figure 1. Workflow diagram of the research.

2.1. Datasets and Labeling

Three underground mine datasets and one unlined tunnel dataset were used for this research. Each mine wall surface was scanned using a tripod-mounted terrestrial laser scanner (TLS) and manually labeled into classes, as described below. TLS was used with the goal of maximizing point density on the tunnel walls and maximizing the coverage of fine details such as wire mesh. In each case, multiple scan positions were collected at different locations to maximize the coverage of features of the tunnel walls. Individual scan positions were merged using a combination of manual point matching and iterative closest point (ICP) alignment. The scanners used have a typical ranging error of approximately 2 mm or less at 25 m distance (one sigma) (see product specifications at [45–47]). Given that the diameter of steel wire mesh is of a similar dimension, and that a typical laser beam diameter is 2 mm, wire mesh is expected to be captured more poorly compared to other objects, which are generally much larger than the point precision and beam diameter.

- Mine 1: Approximately 10 m (in length) of supported mine tunnel face in Colorado, USA (Faro Focus TLS, Korntal-Münchingen, Germany; Figure 2). Average point spacing: 1 mm. Note that no wire mesh was present in the Mine 1 dataset.
- Mine 2: A 15 m long mine drift in Ontario, Canada (Leica HDS6000 TLS; Figure 3). This mine drift was surveyed at a point spacing of about 1 mm, which captures details of the wire mesh covering the tunnel’s walls and roof.
- Mine 3: A 15 m mine tunnel face in Colorado, USA (Faro Focus TLS; Figure 4). Average point spacing 3 mm. Mine 3 includes wire mesh, rock bolts, bare rock, and other infrastructure such as pipes.
- Tunnel 4: A 4 km long unlined water conveyance tunnel in eastern Canada (Leica ScanStation C10 TLS; Figure 5). Two sets of TLS scans were collected in two different years. The primary purpose of the data collection was to support geotechnical design investigations, but the data were also used to perform change detection over time and

to identify rockfalls. The depth of the tunnel below the ground surface is generally less than 200 m, and the tunnel is affected by brittle, kinematically controlled rock failures. Average point spacing is approximately 2 cm.

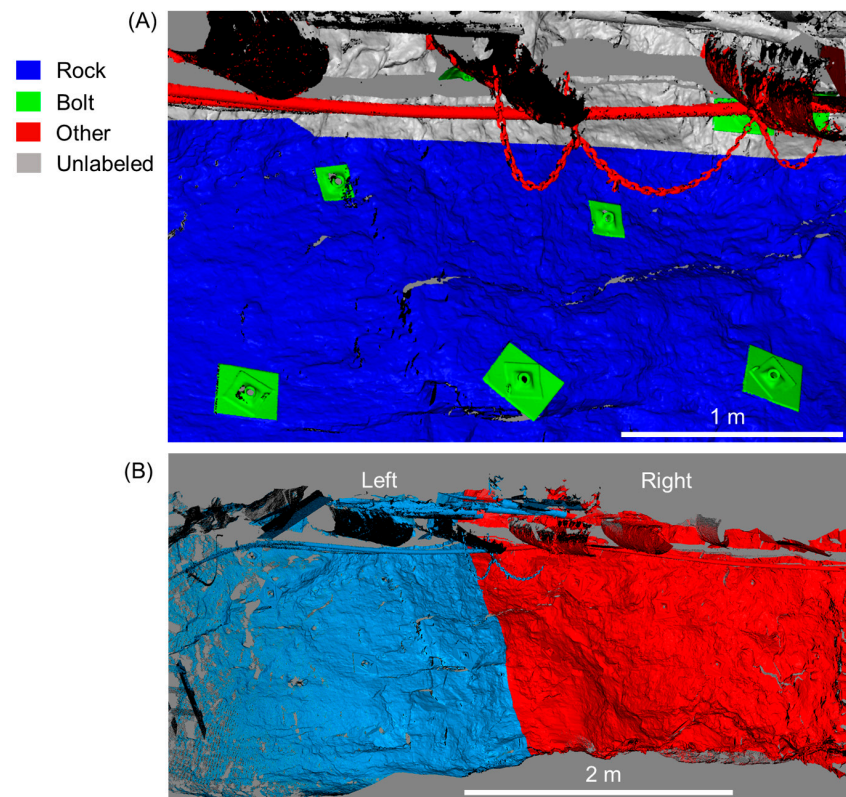


Figure 2. Illustration of the Mine 1 point cloud showing (A) point labels and (B) division of the dataset into left (blue) and right (red) portions for training and testing purposes.

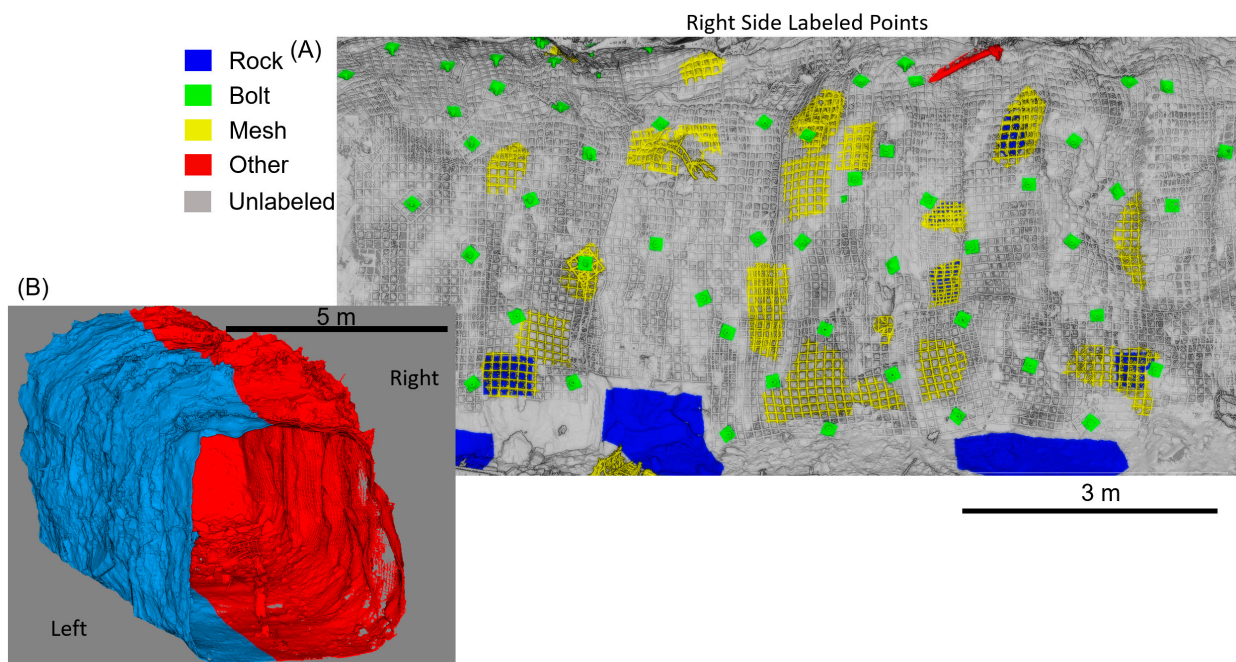


Figure 3. Illustration of the Mine 2 point cloud showing (A) point labels and (B) division of the dataset into left (blue) and right (red) portions for training and testing purposes.

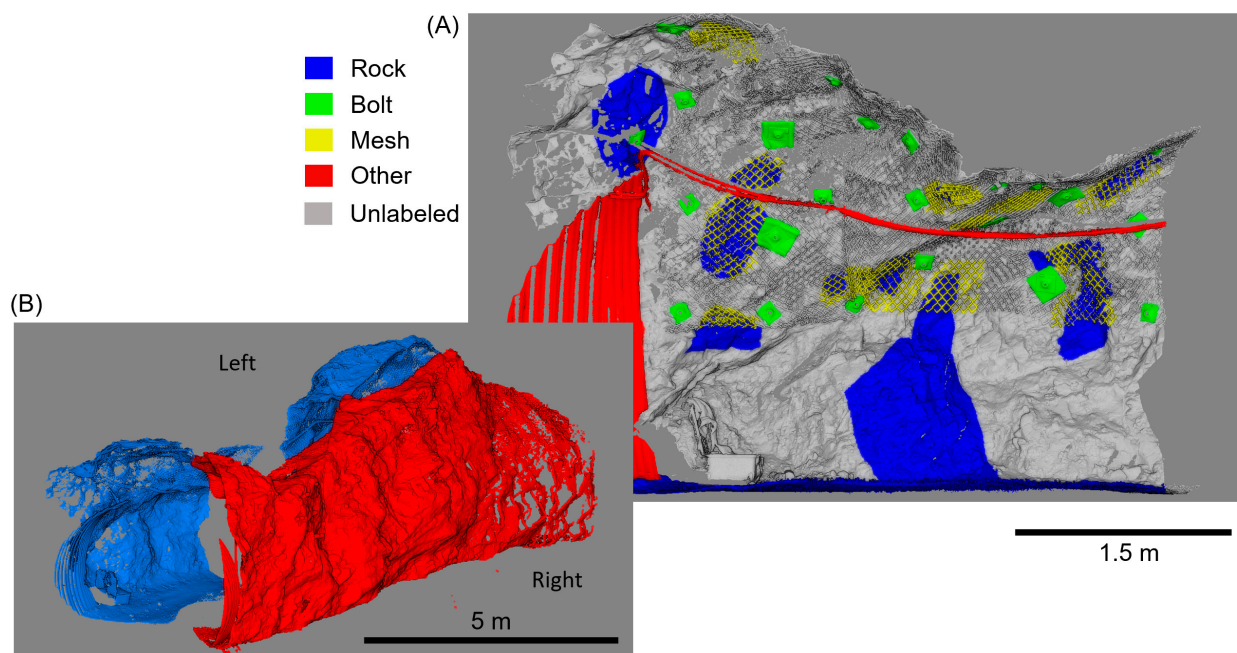


Figure 4. Illustration of the Mine 3 point cloud showing (A) point labels and (B) division of the dataset into left (blue) and right (red) portions for training and testing purposes.

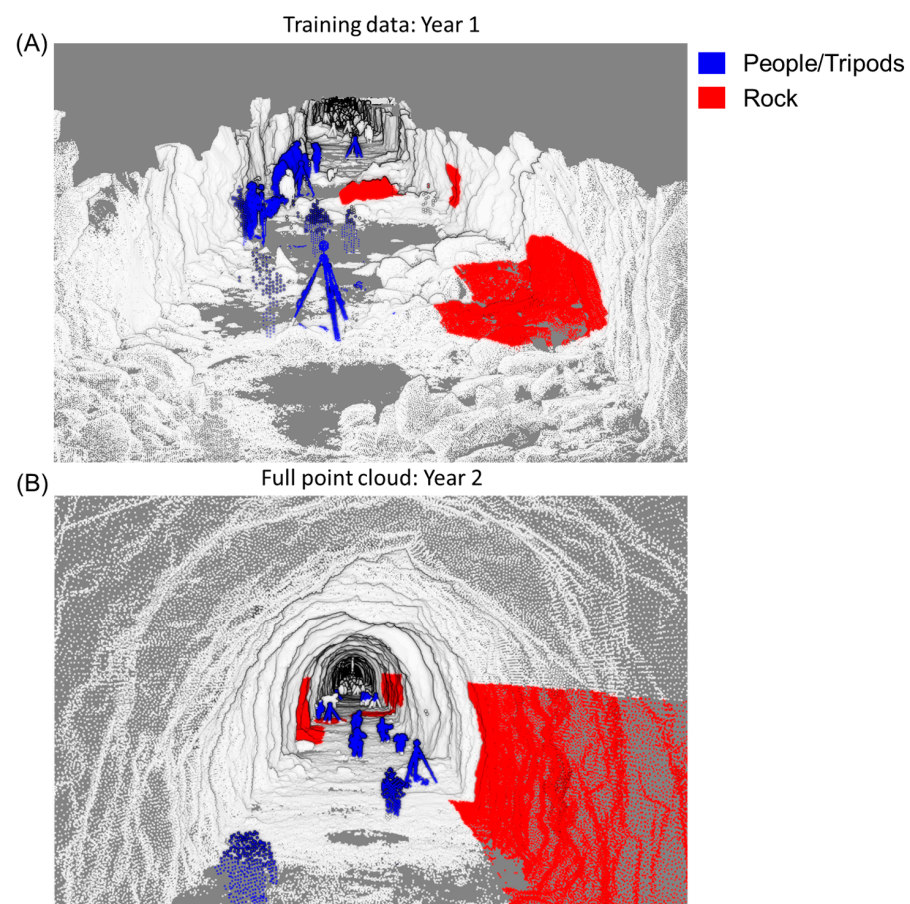


Figure 5. Illustration of the Tunnel 4 point clouds: (A) Year 1 with point labels (blue: people/tripods, red: rock, grey: unlabeled); (B) Year 2 with point labels. The top half of the Year 1 dataset was manually removed to make labeling of the tunnel floor easier.

As listed above, it is clear the Tunnel 4 dataset is relatively different from the three mine datasets. This is because of the different project scope, as well as limitations dictated by budget constraints and client requirements. The goal of scanning Tunnel 4 was to perform change detection to identify fall-of-ground locations and assess the stability conditions of the tunnel walls. As the tunnel is affected primarily by localized falls of ground, as opposed to stress-driven convergence, we do not explicitly consider or measure tunnel deformation in this study. Data collection was supported by tripod-mounted survey spheres to aid in the precise registration of each individual scan position and minimize accumulating errors. Consequently, however, each scan captured the sphere, tripod, and the human operator, and this was kept in the final point cloud dataset. Without the removal of these objects, change could not meaningfully be computed for the floor of the tunnel, but manual data cleaning would be impossible given budget constraints. Therefore, this dataset is an example practical application of a specialized/atypical machine learning classification problem. Also note that while bolts and mesh were present in the first 300 m of Tunnel 4 from the inlet, they are not generally resolvable from the point clouds due to the lower resolution of the data compared to Mines 1–3.

The point clouds were manually labeled into one of four classes: exposed rock; bolt; wire mesh; and “other infrastructure”, which included pipes, corrugated metal, and scrap metal (and people/tripods for Tunnel 4). While rock, bolts, and wire mesh classes were generally present in the mine datasets, the types of “other infrastructure” elements comprising the fourth class varied widely across sites (e.g., different diameters of pipes and ducts; the presence of chains or scrap piles only in one dataset). As such, we do not present or analyze results related to the fourth class in detail for the mine sites as they are not expected to generalize well.

As illustrated in Figures 2–5, not every point in the clouds is assigned a label; instead, a representative subset of regions is labeled across the tunnel wall. Labeling was completed by the lead author, an engineering researcher experienced in point cloud manipulation. In our experience, the labeling process is sufficient for ML training if it results in good spatial coverage of the dataset for all classes and emphasized accuracy over speed. As such, it is not necessary to label every point. In some cases, poorly defined classes could result in less precise classification results because objects may be labeled inconsistently in training data [13]. For this study, however, we find that the four classes are clearly and unambiguously distinct from each other. Previous research [48] has shown that occlusions and missing data could negatively affect classification results. In this study, the TLS datasets were collected from multiple vantage points to avoid the presence of occlusions in the center of the data, and regions along the edges of the datasets with poorer coverage were not included in the training or testing data. This represents a typical scanning scenario as could be expected in practical applications, where multiple scans are collected, or a mobile system is used, to capture data from multiple vantage points and minimize occlusions.

For each mine site, the point cloud is also separated into approximately equal left and right regions for training and testing purposes. This splitting was performed generally along the tunnel axis for convenience, as labeling a single face at a time in CloudCompare [49] is easier for the operator compared to labeling the entire tunnel diameter. For Tunnel 4, labeled data from the Year 1 dataset were used as training data, and data from the Year 2 dataset were used as testing data. Labeled data from an approximately 400 m tunnel was used as training data for Tunnel 4, subjectively deemed as just enough data to adequately capture the variation in the two classes (Figure 5). For labeling training data, the tunnel roof was manually segmented out of the Year 1 dataset to aid in visualization of the floor.

2.2. Point Cloud Classification

This paper implements and compares previously developed methodologies for point cloud classification and generalization.

First, we compute a series of standard multi-scale geometric features based on principal component analysis (PCA) of a series of spherical point neighborhoods. The geometric features are mainly computed as various combinations and ratios of the normalized eigenvalues computed during the PCA. Geometric features are sometimes referred to as “dimensionality” features because they describe the degree to which a local point neighborhood can be considered as “1-dimensional”, “2-dimensional”, and “3-dimensional” [39]. “Multi-scale” refers to the fact that the features are computed for multiple different spherical search radii around each analysis point, creating a representation of the local structure of the point cloud at different scales. Similar works on rock bolt classification use varying methods for defining the point neighborhood, but all are based on roughly how large rock bolts are in the datasets. Gallwey [9] used a single neighborhood defined by 100 nearest neighbors, while Saydam [40] used a single 10 cm radius. Others [37,38] use multiple radii between 1 cm and 15 cm. Because these features are widely used, there are a number of high-quality references which describe the feature calculation, and they are not described in detail here (see [9,19,39,44] for definitions of geometric features; our feature computation code is available at [50]). Similarly, previous works have extensively evaluated how different combinations of features affect performance [9,33,51,52]. In our experiments, we employed a core set of features that are common to most or all recent studies.

Table 1 summarizes the point features and scales calculated (70 geometric, 20 intensity). Each feature is computed at five different search radii (scales) between 0.2 m and 0.02 m. This range was chosen to be larger than the point density (on the low end) and larger than the average edge length of a bolt faceplate (~0.15 m). In addition to geometric features, we include intensity as recorded by the TLS as a feature. We use four intensity-based features as described by [14] for the classification of different lithology types.

Table 1. Summary of point features.

Feature Type	Feature Name	Search Radii (m)
Geometric Features [19]	Slope (mean, std. dev., skew, kurtosis)	0.2, 0.1, 0.07, 0.04, 0.02
	PCA3	
	PCA2	
	PCA1	
	Anisotropy	
	Sphericity	
	Linearity	
	Planarity	
	Omnivariance	
	Eigentropy	
	Curvature	
Intensity [14]	Mean Intensity	
	Mean—Point	
	Point—Min	
	Max—Point	

Second, we trained three different classifier configurations, based on recent studies using NN and RF to classify rock bolts (Table 2). These can be simply described as (1) a shallow neural network with one hidden layer; (2) random forest with 200 trees and each tree can grow large; and (3) random forest with 500 trees and shallow individual trees. Hyperparameters for each of these classifiers were selected to be the same as in recent studies by Gallwey et al. [9] and Singh et al. [37]. These two studies arrived at notably different optimal hyperparameters for their RF classifiers despite similar classification goals and datasets, and our goal was to compare the two. Gallwey et al. [9] used a randomized hyperparameter search to identify optimal values for their RF, while Singh et al. [37] mentions that RF hyperparameters were tuned but did not specify how the final values were selected. Singh et al. [37] apparently used default hyperparameters (except number of hidden layers) to train their NN classifier using MATLAB, and we used the

same parameters (tan-sigmoid activation, scaled conjugate gradient training, cross-entropy loss, default early stopping criteria). Singh et al. [37] selected the number of hidden layers by a brute force search with numbers of neurons ranging from 1 to 200.

Table 2. Description of tested classifier configurations.

Classifier Name	Type	Hyperparameters
NN [38]	Neural network	1 hidden layer, 20 neurons
RF1 [9]	Random forest	200 trees, min leaf size 1
RF2 [38]	Random forest	500 trees, max 2 decision splits, min leaf size 5

2.3. Training and Testing Procedures

To train each classifier, a graphical software tool developed in MATLAB by the authors was used (Terpukto, see Ref. [50]). Features were computed for all point clouds using a stratified sample of 20,000 points from each of the four classes (total number of points approximately 80,000). This ensures that each class is equally represented in the model. Features were computed once using all full density point clouds, and a second set was computed with each raw point cloud subsampled to have a point spacing of at least 5 mm to reduce variations in density between different mine sites. Classifiers were also trained with and without intensity features. Each RF model took approximately two minutes to run training, while NN models took approximately five minutes. Training was conducted on a desktop computer with 64 GB RAM and an Intel i7-5960X CPU. No GPU resources were used for training. Specifically for the Tunnel 4 case study, all processing was completed using a standard Dell Latitude business laptop with 32 GB RAM and an Intel Core Ultra 7 165U CPU.

Classifiers were tested in both “same-site” and “generalized” (or “cross-site”) configurations. “Same-site” testing means that the classifier was trained on the left side of the dataset, and test accuracy was evaluated on the right side. Complete spatial separation of the training and testing data is necessary to ensure that accuracy is not overestimated due to overlapping neighborhoods of training and testing points [44]. “Generalized” testing means that the classifier is trained on both the left and right sections of one site and tested on the left and right sections of another site.

Considering two densities, with and without intensity features, three classifiers, and all possible generalization combinations, a total of 108 (2·2·3·9) individual classifiers were trained. For each test result, the overall accuracy and F scores for each of the four classes were recorded.

Initially, we performed an additional leave one feature out (LOFO) feature importance analysis using a random forest with 200 trees and no intensity features. For each training–testing case considered, including generalized combinations, we trained 70 RF1 models (as defined in Table 2), each with a different feature left out, for a total of 630 additional models trained. The purpose of this analysis was to identify if certain features and/or scales were consistently less helpful in the generalized case than in the same-site case, indicating that the feature was highly site specific. However, inspection of these results did not identify consistent differences between same-site and cross-site features, which may be due, in part, to the high correlation of features between different scales [53,54]. Further, individual geometric features, such as curvature or omnivariance, neither consistently increased nor decreased performance across different generalization cases. As such, we kept the full feature set for use in the ML workflow, consistent with previous works. The results table of this analysis are included in the Supplementary Materials for this paper.

2.4. Post-Processing

In previous studies focused on bolt extraction [9,37], additional steps are included to separate individual bolt instances, using clustering algorithms like DBSCAN [55]. The focus of this paper is on the point-wise classification step, but we also illustrate preliminary

instance segmentation results on the bolt class. To this end, after point-wise classification is complete, the bolt class is exported, and the Connected Components algorithm in CloudCompare is used to separate individual bolts. The Connected Components algorithm requires two parameters: grid step size and minimum number of points. Because this algorithm is quick to run, we tested several combinations of these two parameters as needed using trial and error to arrive at the best result for each dataset. Grid step size was generally approximately 1 cm, and the minimum number of points was between 200 and 500 depending on the dataset.

3. Results

3.1. Influence of Point Density Variation and Noise

Here, we briefly address the results of subsampling the data. Note that the data table used to generate all results figures is included in the Supplementary Materials of this paper. Figure 6 shows the differences in performance between uniformly subsampled and full-density data for all combinations of training/testing and with/without intensity features. While the difference for individual tests could be quite large, i.e., up to a 0.2 to 0.4 difference in each accuracy metric, the mean difference was generally close to zero, indicating, on average, a 0.01 to 0.03 higher accuracy for the full-density data compared to the subsampled data. This is consistent with [44], who concluded that higher density was correlated with higher accuracy. Having similar density between training and testing is important [48], but the effect is likely insignificant for the amounts of density variation present in our datasets. For the results in subsequent sections, only results with full density are shown for clarity.

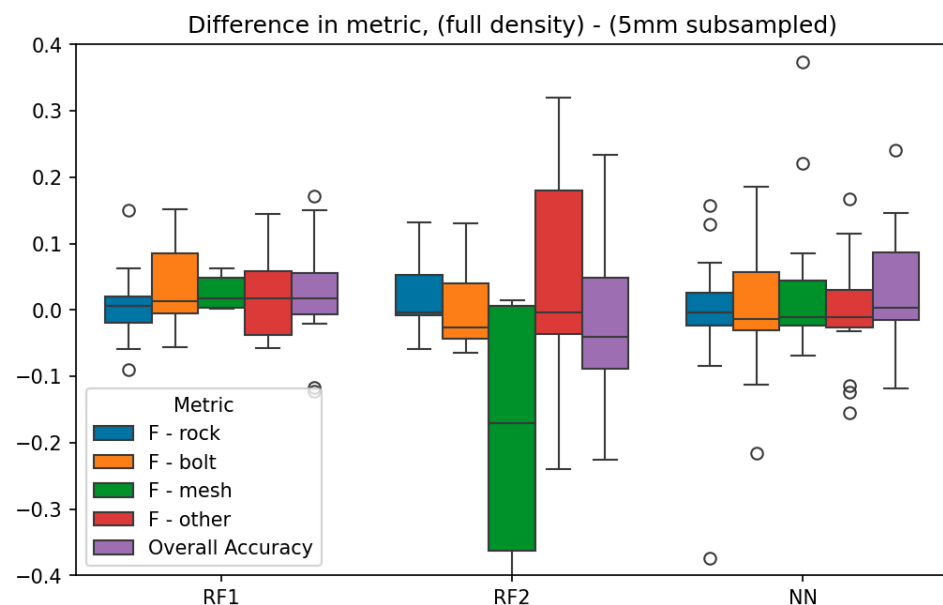


Figure 6. Box plots showing the difference in accuracy metrics (F score and overall accuracy) between full-density data and data subsampled to 5 mm. Results are grouped by metric type (box color) and ML model type (RF = random forest; NN = neural network).

We also investigated differences between datasets in terms of noise derived from scanner ranging error and alignment error. For a 50 cm by 50 cm patch of bare, planar rock in each of the mine datasets, we computed surface roughness with a radius of 5 cm using CloudCompare. For all three datasets, the mean roughness was between 2 mm and 3 mm, which was interpreted as indicating that each dataset is similar in terms of ranging noise and residual alignment error. Note that this simplified analysis does not account for possible real differences in roughness caused by rock joint characteristics, but due to

the low overall roughness values computed, we can assume that variations in rock joint roughness are small.

3.2. Classification Model Types and Generalization

In terms of classifier performance, we did not find consistent statistically significant differences between NN and RF model results (Figure 7). In addition to one-way ANOVA, as shown in Figure 7, we performed paired *t*-tests for all performance metrics between NN and RF1. Interestingly, bolt F score was significantly higher for NN than for RF1 ($p < 0.001$), but all other metrics were not significantly different between the two models, including overall accuracy. Thus, on average, NN accuracies were slightly higher than both RF1 and RF2 but slightly less stable (a few results with very low accuracy, see wire mesh F scores in Figure 7). A RF model with deep trees (RF1) tends to outperform a model with a larger number of shallow trees (RF2); the overall accuracy of RF1 was significantly higher than RF2 in paired *t*-tests ($p < 0.05$).

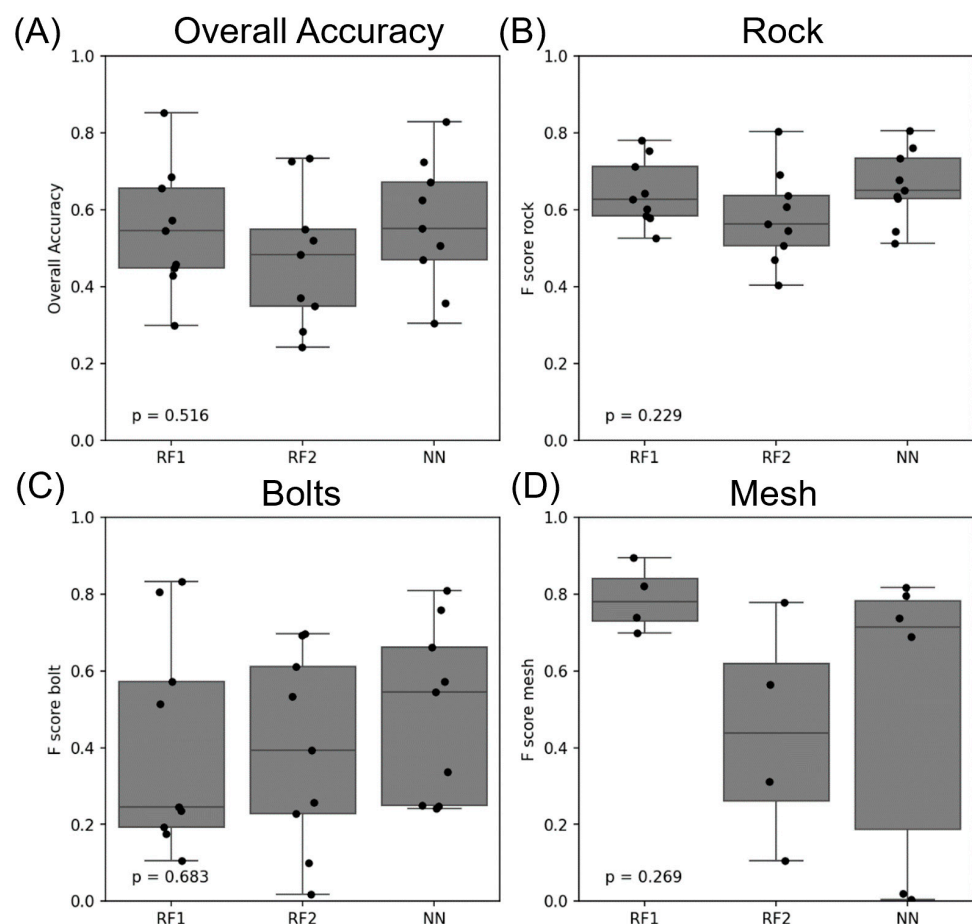


Figure 7. Accuracy metrics for different ML model types (full-density data and without the use of intensity features): (A) overall accuracy; (B) rock F score; (C) bolt F score; (D) mesh F score. *p*-values were computed using one-way ANOVA (one way F test) with the ML model as the grouping variable.

In terms of generalization, which refers to training the classifier on one dataset (e.g., Mine 1) and evaluating it on another dataset (e.g., Mine 2), the same-site configurations were uniformly much higher accuracy than the generalized ones (Figures 8 and 9). This can be explained by two main factors. First, the nature of the intensity data varied between the datasets: the reflectivity of objects and surfaces appeared to vary for different mines, even when using the same scanner type. Figure 9C shows that including intensity data as a feature improved the results, on average, for same-site training but reduced accuracy, on average, for cross-site (generalization) tests. Second, even when only geometric features

are used, generalization accuracy is often much lower than same-site accuracy due to differences in the shapes and the arrangement of objects in the scene. For example, the Mine 3 and Mine 2 datasets both have bolts underlain by one or more layers of wire mesh (examples shown in Figure 10), and their generalization accuracy values (without intensity) are 10–25% lower than the same-site accuracy. In contrast, the Mine 1 dataset has no wire mesh, meaning that the geometric characteristics and their contrast between bolts and other objects is very different than at the other two mines. Consequently, F scores for cross-site training/testing using Mine 1 data are considerably lower than all other tests, especially for the bolt class (Figure 9B).

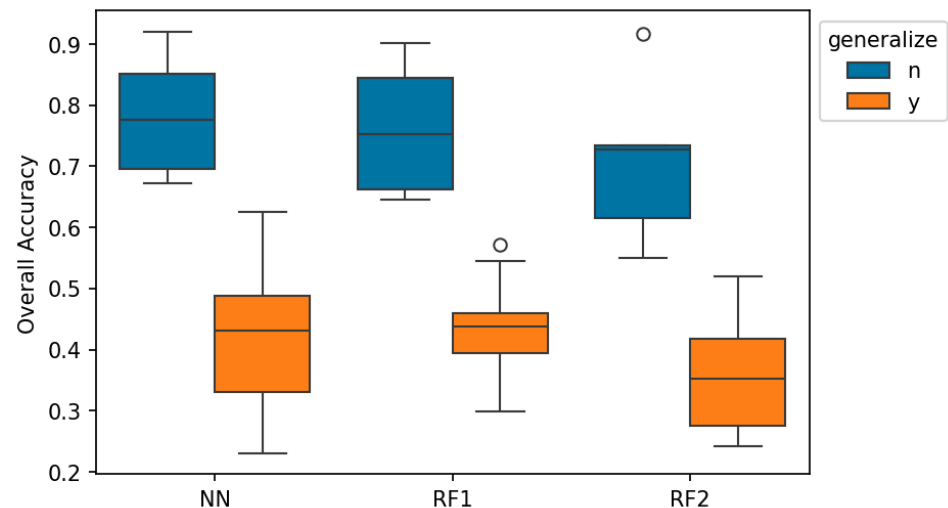


Figure 8. Boxplots showing aggregation of overall accuracy results (full-density data) by ML model type and whether they were trained same-site (generalize: n) or cross-site (generalize: y).

3.3. Visual Interpretation and Bolt Identification

Visual inspection of classification results aids in the assessment of the demonstrated approach's strengths and weaknesses in generalization. For reference, Table 3 summarizes key F scores in the same site and cross site conditions for the three mine sites. Figure 11 shows classification and bolt identification results for the Mine 3 dataset, using intensity features and training on the same dataset. The F scores for this configuration for rock, bolts, and mesh, are all between 0.8 and 0.9 (from Figure 9). In the highlighted region, all 19 bolts are correctly identified (object recall: 1), and four extraneous false clusters remain, which could be readily filtered out based on having a small number of points.

Table 3. Point-wise NN F score of the bolt class in the mine datasets (with intensity features) compared with the maximum F score of any NN classifier obtained from training on another site.

Test Dataset	F Score—Same Site	Max F Score—Cross Site
Mine 1	0.88	0.27
Mine 2	0.77	0.66
Mine 3	0.90	0.57
Gallwey [9]	0.64	—

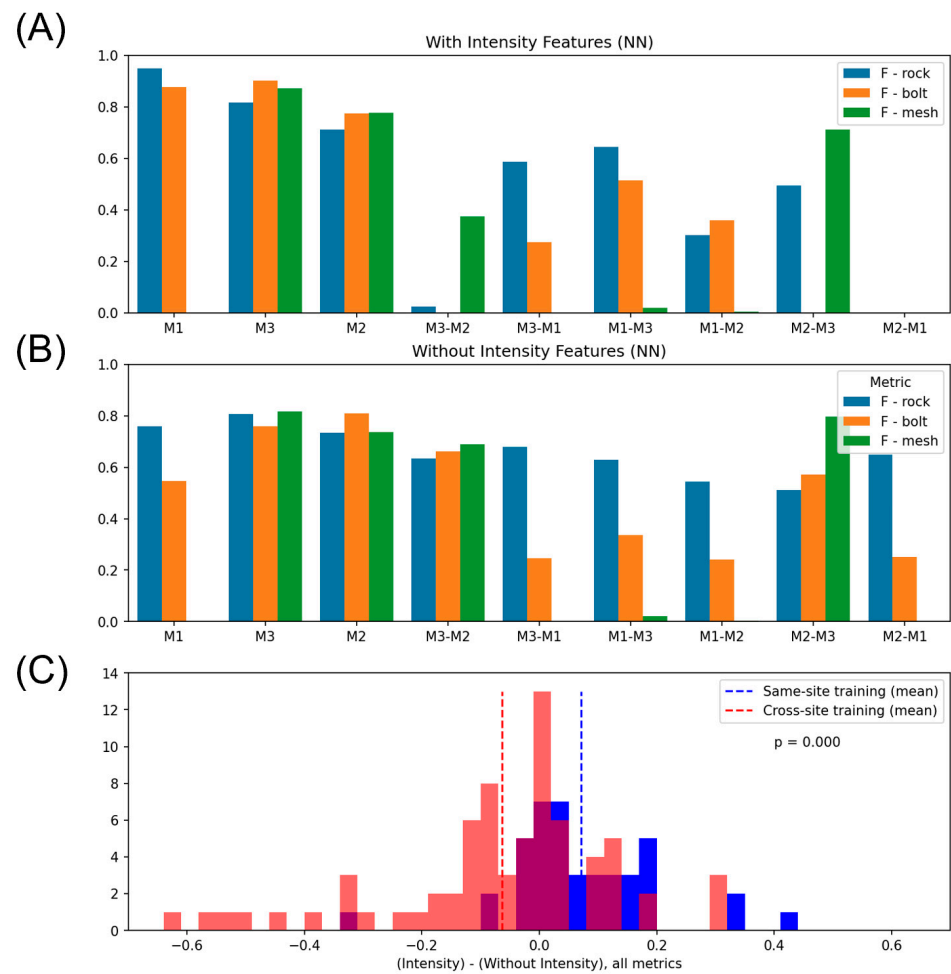


Figure 9. Plots of F scores for the NN (neural network) model, different materials, and different training/testing configurations (for example, “M3-M2” indicates training on the Mine 3 dataset and testing on the Mine 2 dataset): (A) results including intensity features; (B) results with geometric features only; (C) histogram of differences between corresponding pairs of results (any metric) with and without intensity features (positive = metric with intensity features was higher than without intensity features).

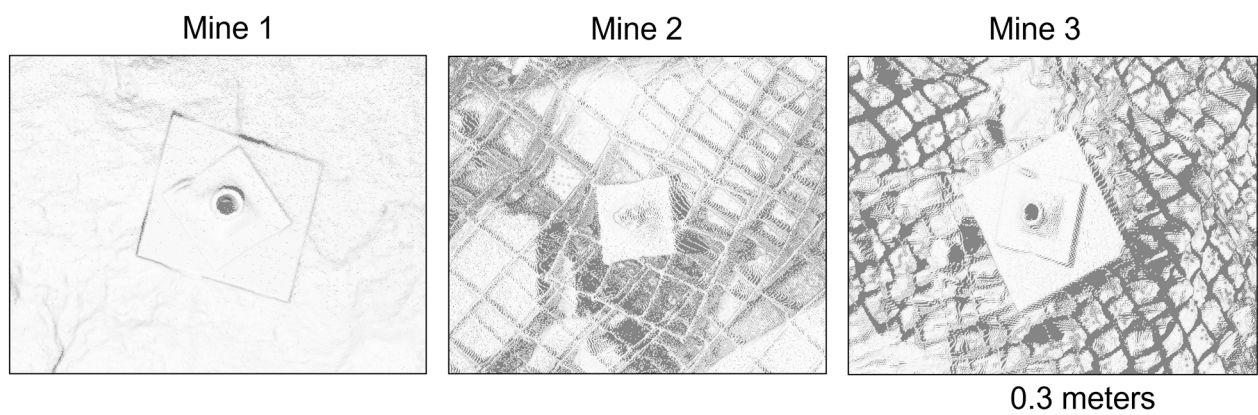
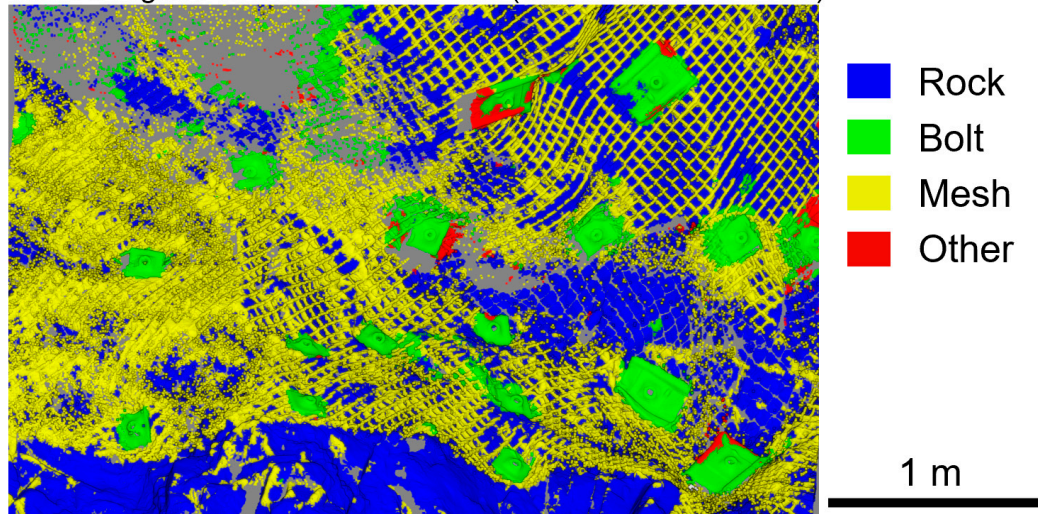


Figure 10. Examples of typical bolt shapes and wire mesh conditions present in the three mine datasets.

(A) Mine 3 Right: Point Label Predictions (Trained on Mine 3 Left)



(B) Bolts Extracted

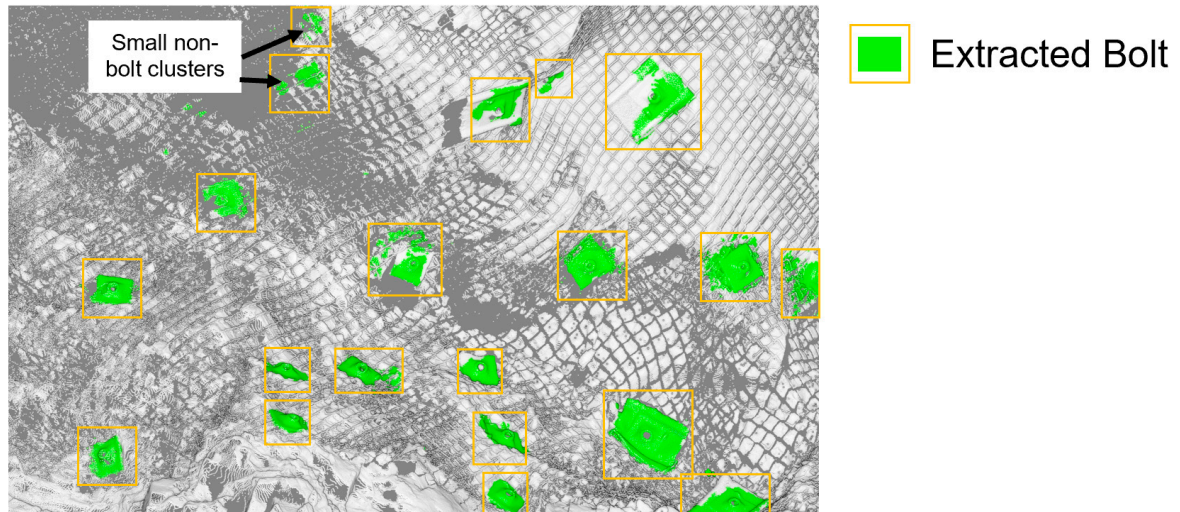


Figure 11. (A) Output classification results for a subset of the Mine 3 (right side) dataset using intensity features. The classifier was trained using the Mine 3 (left side) dataset. (B) Extracted bolts (yellow boxes) after applying the Connected Components algorithm.

Figure 12 shows classification results for this same area of the Mine 3 dataset, except attempting to apply a geometric-only classifier trained on the Mine 2 dataset. In this case, very few bolts are correctly identified and there are many false positives. Rock and bolt F scores are between 0.5 and 0.6 for this example, while the F score for mesh was 0.82.

Same-site classification results for Mine 2 are shown in Figure 13. The Mine 2 mine wall is the most geometrically complex and has the highest roughness of the three datasets, with multiple layers of mesh covering most of the wall, and many bolts are partially or mostly covered by mesh. Further, bolts do not have a consistent shape, with some having the bolt protrude and others with a central depression instead (Figure 10). To highlight bolt and non-bolt classification specifically, the rock, mesh, and infrastructure classes are merged as “non-bolt” for this example. The point-wise F score for bolts in this example is 0.83 (Figure 9B). A total of 14 bolts are correctly identified and 4 are missed (object recall: 0.78).

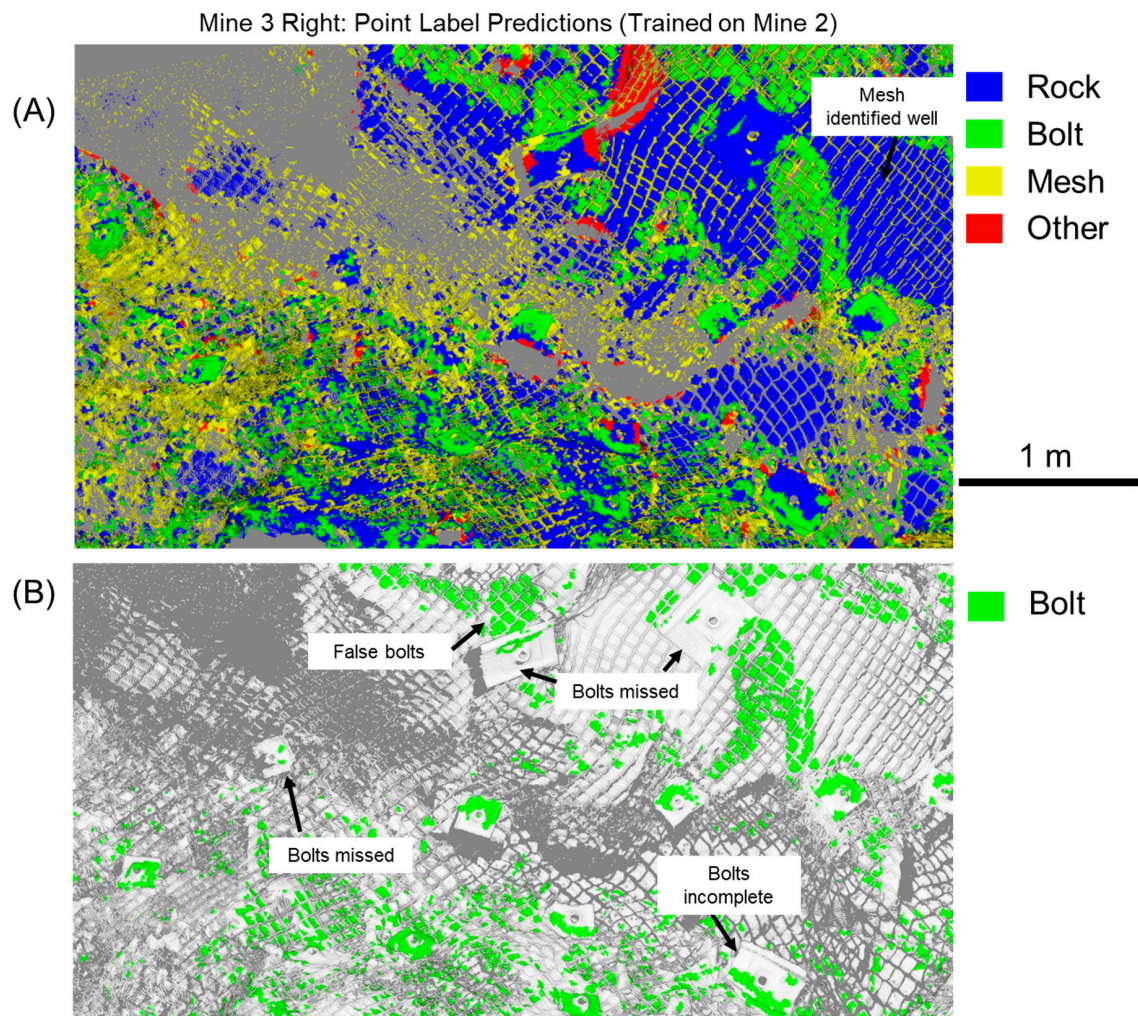


Figure 12. (A) Output classification results for a subset of the Mine 3 dataset using a classifier trained on Mine 2 (no intensity features). (B) Extracted bolt points, highlighting incorrect instances because of the poor initial classification.

3.4. Rock and Mesh Identification Results

Overall, F scores for same-site identification of rock and mesh materials were generally good and in similar ranges as have been found in previous studies. For example, in our study, rock and mesh F scores were observed to be typically between 0.70 and 0.90 in same-site tests (Figure 9). Correspondingly, for rock [44], a median generalized F score of 0.69 and [51] were found to have an average same-site F score of up to 0.83. For mesh, Ref. [10] computed an F score of 0.88 for a same-site test. This comparison suggests that same-site test performance for rock and mesh is relatively consistent at various sites, whereas in the generalized case, performance may be much lower.

3.5. Tunnel 4 Results

As noted in Section 2, the Tunnel 4 datasets were collected as part of a client-funded project with a specific change detection mandate. Given these constraints, we used the 3DMASC plugin in CloudCompare [33], which implements the same ML methodology as used above, to filter only points corresponding to rock and remove the points corresponding to tripods and human operators in as efficient a manner as possible. The 3DMASC tool implements a classical ML workflow in an open-source graphical user interface, allowing the user to input manually labeled training data and specify the features to be calculated, which can include color, intensity, or any other scalar field, in addition to most commonly

used geometric features. The tool trains an RF model with several hundred deep trees by default, but the number and depth of trees can be modified. A classifier was trained using 3DMASC with only geometric features (intensity values were not present in the available data) and default RF hyperparameters (200 decision trees, max depth 70).

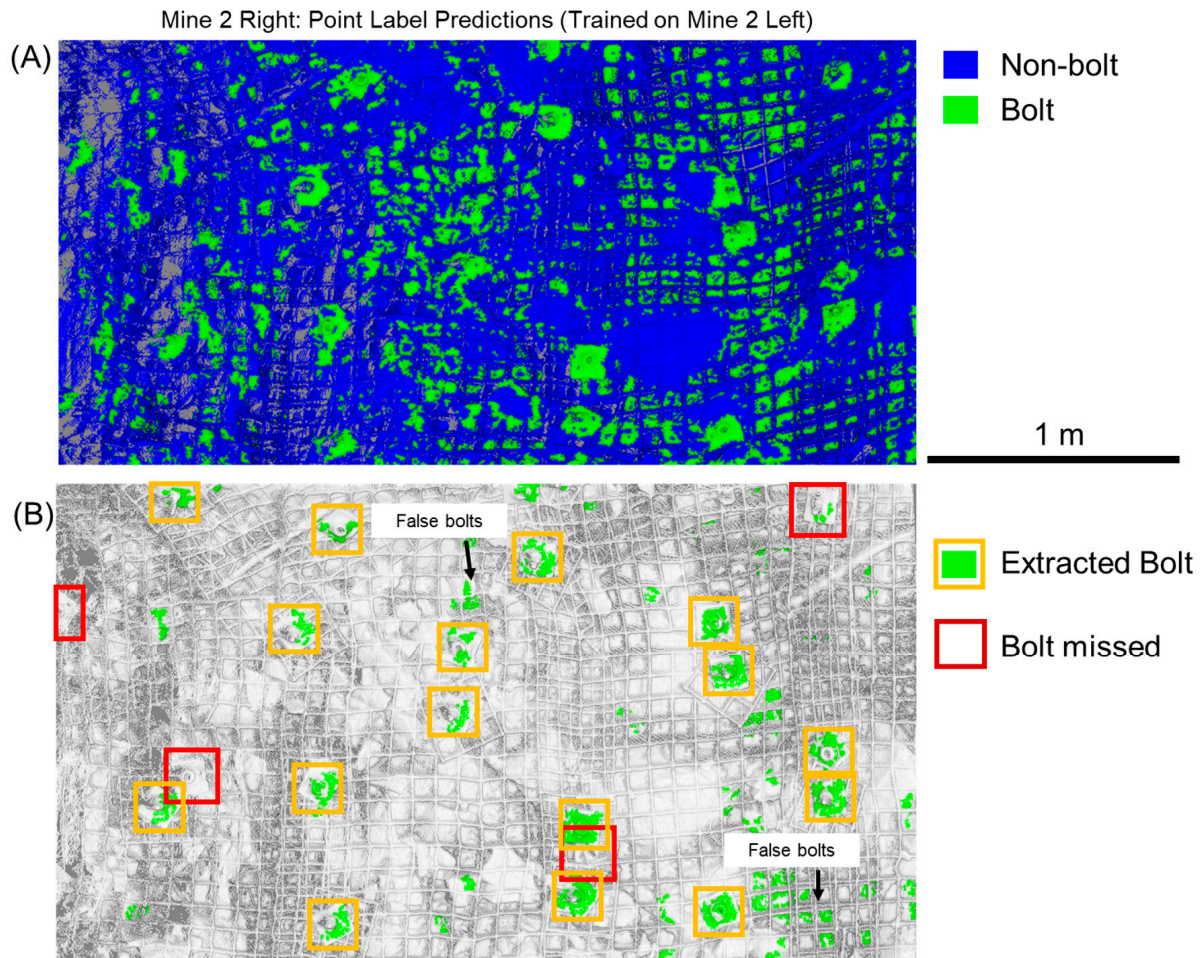


Figure 13. (A) Point-wise classification results for same-site classification of Mine 2, considering only the bolt class and no intensity features. (B) Extracted bolts using the Connected Components algorithm (yellow boxes), with missed bolts shown with red boxes.

The trained classifier was applied to both the dataset from Year 1 and from Year 2 (Figure 14); based on qualitative evaluation of the similarity in point density and shapes of objects between the two datasets, it was deemed likely that the classifier would be capable of generalizing to another scan of the same location. Inspection of the results showed that this assessment was correct. The overall accuracy of the classification assessed on Year 2 was 0.98, with a rock F score of 0.99 and a “people/tripods” F score of 0.71. Further analyzing the people/tripods results, we see that the precision was 0.58 and the recall was 0.93, indicating that a significant number of actual rock points were incorrectly classified as people/tripods. These false positives tended to be located at larger boulders on the floor and the edges of data gaps, such as puddles of water or a circle of missing data below the scanner itself. As these errors were generally randomly distributed and isolated points, the error rate was deemed acceptable for the purpose of pre-change-detection filtering for detecting rockfalls.

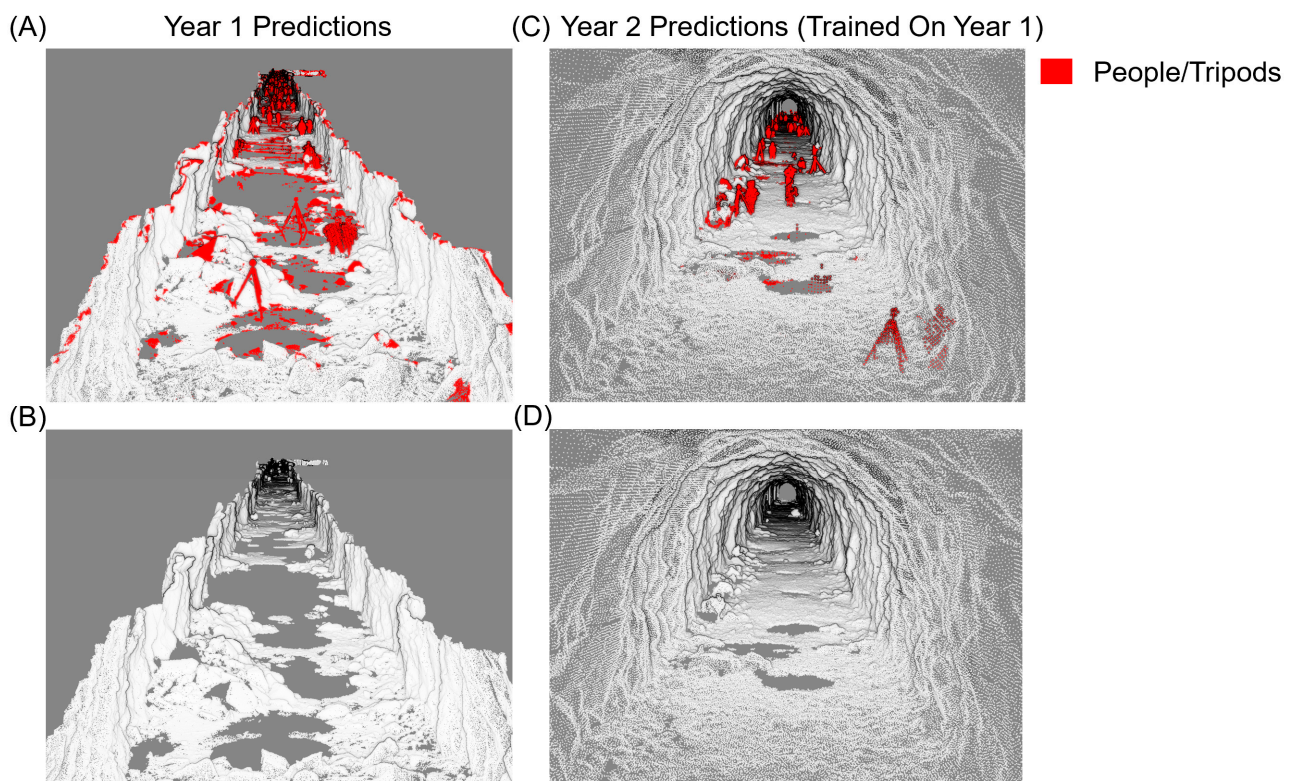


Figure 14. Classification outputs using the Year 1 training data shown in Figure 5: (A,B) show classification results for the dataset from Year 1 with people/tripods (red) (A) and showing only rock (grey points) (B); (C,D) show classification results applied to the Year 2 dataset using the same classifier trained on Year 1 with and without people/tripods, respectively.

As this was a client-funded project, only basic, readily available computing resources were used, which consisted of a standard business laptop. Despite this, the entire classification process took less than approximately 1 h (data labeling: 15 min; feature computation: 15 min; training: 10 min; testing: 5 min; post-processing: 15 min), which emphasizes the practical usefulness of the 3DMASC tool. The project moved forward with the filtered datasets, and changes were computed for the entire tunnel circumference.

4. Discussion

4.1. ML Model Types

Several recent studies have demonstrated that underground infrastructure, particularly bolts, can be extracted using established ML classification methods. In comparing RF and NN ML model configurations from [9,37], we found that across the three mine datasets tested and multiple training–testing combinations, differences between ML models were relatively minor. As noted in Section 3.2, NN performed significantly better than RF with deep trees, but for rock bolts only, and an RF with deep trees outperformed an RF with shallow trees in all cases. Overall, we suggest that both RF and NN model types are adequate for the purpose of geometric feature-based classification, especially given the generalization considerations discussed below.

The standard ML workflow using either RF or NN is generally successful when training and applying it within a single dataset, with F scores generally above 0.8 for the four datasets quantitatively analyzed here. However, even in this simplest case, there is considerable variation across datasets; the Mine 2 mine wall presented a serious challenge to the algorithm due to the wall’s overall complexity and the relative unhelpfulness of the intensity values in distinguishing materials. When intensity values were not used, trained models had more difficulty distinguishing the edges of the bolt plates from the surrounding

rock or mesh, indicating that geometric dimensionality features alone are not sufficient to fully describe bolts as they exist in our datasets.

4.2. Generalization

What previous studies have not accounted for is variation in characteristics between different datasets, which may significantly alter classification results. We have shown that such differences have a negative effect on accuracy in many cases, and these accuracy changes are typically much larger than differences caused by modifying the classification algorithm. This does not necessarily suggest that previous results are not useful or correct; it simply highlights that the research community needs to be realistic in terms of the current capabilities and limitations of these methods. In broader computer vision domains, standard benchmark datasets are typically used to reduce the influence of dataset variation on published results (e.g., Semantic3d.net [56]), but this may not be practical in developing for the variety of small-scale classification problems continually being tested in engineering. Two recent works have proposed tunnel benchmark datasets [57,58], indicating that there is increasing interest in tunnel classification and the potential for increased collaboration. This collaborative approach could accelerate the development of robust benchmarks and enhance the overall quality and comparability of research in this field.

We tested homogenizing density between datasets but found that this did not have a consistent influence on accuracy, despite previous work suggesting that density variation could influence generalization [44]. We hypothesize that since the three mine datasets all have relatively high starting density (point spacing < 1 cm), there may be a threshold effect whereby at sufficiently high density, relatively small changes in density do not significantly alter the geometric details of objects. Similarly, we found that noise levels were similar between datasets and thus cannot be directly linked to variations model performance.

In limited cases, such as the included case study (Tunnel 4), transfer of a trained algorithm to another site also produced reasonable results (e.g., mesh extraction from for M3 using the model trained on M2 was relatively good), but this is contingent on very high similarity between the training and evaluation data. This likely means that the most practical alternative is for the algorithm to be retrained nearly every time a new dataset or project is encountered [44]. As a result, we suggest two avenues for addressing these challenges:

1. To facilitate easy adaptation and retraining of models, tools and software should be developed which emphasize calculation speed, include options for active learning and domain adaptation [48,59–64], and create a streamlined workflow from raw data to the final classification output with immediate feedback. In general, we suggest that RF is an appropriate algorithm to use for most practical scenarios, and the open source 3DMASC tool [33] is a meaningful step in this direction. To improve on this tool, the following may be useful future developments: adversarial domain adaptation techniques, in which a model learns to minimize the differences between the source and target domain, have shown promising performance [63]; further, active learning aims to reduce manual effort by selecting and presenting the most informative data points for human labeling [65].
2. There are still opportunities to improve the classification skill of object types for which standard geometric features are not effective in real-world scenarios. For the example of bolts, previous studies have presented example datasets where geometric features are strongly discriminative; we present some cases where they are effective and others where they are not as effective. Object-based classification and the semantic grouping of points with similar morphological features could be useful as classification of entire homogenous objects may result in higher accuracy and sharper, more meaningful boundaries than point-based classification (e.g., [23,66]). Alternative feature extraction methods, such as those using supervoxel segmentation, should be explored to improve features for generalization [67,68]. This is also where deep learning-based approaches

may be worth further investigation for the detailed segmentation of object boundaries (e.g., [40]).

4.3. Deep Learning Methods

As mentioned in the introduction, we do not consider the growing number of deep learning methods for point cloud classification in our study for tunnel infrastructure identification. Classical machine learning approaches have many practical advantages, including training time (typically a few minutes compared to hours for deep learning), lighter computation, and interpretability [33]. However, complex object shapes likely extend beyond the information simple geometric indices are capable of representing. Saydam et al. [40] highlighted that geometric features are typically able to identify bolt *locations* with reasonable precision, but PointNet was better able to identify the precise bolt *boundary* and distinguish between candidates that the coarser filtering step could not. This is concordant with our results, in which high performance was typically associated with a clear link between specific features and simple geometric objects (e.g., intensity differences between metal and rock, linear strands of wire mesh), whereas bolts with a subdued profile on the rock face were detected much more poorly. This limitation needs to be acknowledged when applying geometric features to increasingly complex and varied object types.

4.4. Limitations

The classification accuracy of the proposed methods may be significantly influenced by the quality of the data cloud, which can vary due to numerous factors. The current study assumes a high-quality data cloud collected by a terrestrial laser scanner, but poorer quality data clouds collected by alternative methods, such as with a mobile laser scanner or autonomous vehicles, could result in poorer differentiation between object types and worse generalization performance. A qualitative evaluation of data quality and its potential impacts on object detection should be performed before every project.

Additionally, the methods discussed require the manual labeling of data rather than relying on unsupervised techniques, which can be more labor-intensive and time-consuming. This process necessitates a certain level of knowledge and training in machine learning to ensure accurate and effective implementation.

5. Conclusions

Our study focused on the independent testing and validation of tunnel object extraction from terrestrial laser scanning (TLS) using geometric features and classical machine learning (ML) techniques. The classification model comparison highlighted that neural network (NN) and random forest (RF) models exhibited statistically comparable overall accuracies, but NN models demonstrated slightly higher accuracy on average, particularly in bolt identification.

Regarding generalization, our findings indicated significantly higher accuracy for same-site evaluations compared to cross-site evaluations. This discrepancy can be attributed to variations in intensity data and geometric characteristics between different mine sites. Visual interpretation of the results further emphasized the strengths and limitations of the classification approach, showcasing moderately to highly successful identification in same-site scenarios and challenges in cross-site evaluations.

A practical case study involving a water conveyance tunnel demonstrated the adaptability of the classification method to real-world applications. Despite challenges posed by the presence of non-rock elements in the TLS datasets (tripods and human operators), the developed workflow successfully identified rock within the tunnel and enhanced the usefulness of the datasets in subsequent engineering analyses. The case study emphasizes the need for tools that facilitate the rapid evaluation and retraining of models based on limited data, as real-world projects are often budget constrained.

Considering our findings, we propose two key recommendations for addressing challenges in underground object extraction from lidar datasets: the development of tools

and software emphasizing calculation speed, active learning, and a streamlined user-feedback-driven workflow; and exploration of deep learning-based approaches (and other alternative point cloud descriptors) to enhance classification skills for object types where standard geometric features may be less effective in real-world scenarios, such as bolts. Specifically, the following deep learning topics should be explored:

1. Hybrid methods: Combine classical geometric feature-based techniques with deep learning to balance speed and accuracy [69]. This approach leverages the strengths of both methods, ensuring that the system can quickly process data while maintaining high classification accuracy. By integrating these techniques, we can address the limitations of each method when used independently, such as is demonstrated in [40].
2. Efficient architectures: Develop lightweight deep learning models like MobileNets for resource-constrained environments [70]. These models are designed to perform well on devices with limited computational power, such as mobile devices or embedded systems. By focusing on efficiency, we can ensure that the models are practical for real-world applications where resources may be limited.
3. Transfer learning: Use pre-trained models and fine-tune them on specific tunnel point cloud data to improve performance [71]. Transfer learning allows us to leverage existing models that have been trained on large datasets, reducing the time and resources needed to develop new models. Fine-tuning these models on underground infrastructure can enhance their accuracy and effectiveness in this use case.
4. Real-time processing: Optimize algorithms for parallel processing on GPUs [72]. This involves modifying algorithms to take advantage of the parallel processing capabilities of modern GPUs, significantly speeding up computation times. Real-time processing is crucial for applications that require immediate feedback or rapid data analysis, such as in-field assessments or dynamic monitoring systems.

Other techniques, such as data augmentation and domain adaptation, may also be considered to further improve generalization performance.

Future work on this problem should consider the inclusion of multiple benchmark datasets used by other authors (e.g., [57,58]) and the addition of other more diverse mines and tunnel datasets for training via increased collaboration amongst researchers. We suggest that newly developed classification algorithms should always be compared with a standard classification approach, such as those used in this study, and across multiple datasets to effectively demonstrate improvement.

Supplementary Materials: The following supporting information can be downloaded at <https://www.mdpi.com/article/10.3390/rs16234466/s1>, Table S1. Accuracy metrics for all tests and for the leave-one-feature-out analysis.

Author Contributions: L.W.: conceptualization, methodology, software, writing—original draft, investigation; G.W.: conceptualization, methodology, supervision, project administration, funding acquisition, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: The research conducted for this study was funded by the National Institute for Occupational Safety and Health (NIOSH) under Grant Number 75D30119C05412.

Data Availability Statement: Feature data computed for point clouds and used for classifier training are available upon reasonable request. Point clouds are unfortunately not available due to confidentiality agreements with clients.

Conflicts of Interest: The authors declare that they have no conflicts of interest. The private client for Tunnel 4 had no role in the design of the study; in the analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Kumar Singh, S.; Pratap Banerjee, B.; Raval, S. A review of laser scanning for geological and geotechnical applications in underground mining. *Int. J. Min. Sci. Technol.* **2023**, *33*, 133–154. [\[CrossRef\]](#)
2. Martínez-Sánchez, J.; Puente, I.; González-Jorge, H.; Riveiro, B.; Arias, P. Automatic Thickness and Volume Estimation of Sprayed Concrete on Anchored Retaining Walls from Terrestrial Lidar Data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B5*, 521–526. [\[CrossRef\]](#)
3. Zhou, H.; Gao, B.; Wu, W. Automatic Crack Detection and Quantification for Tunnel Lining Surface from 3D Terrestrial LiDAR Data. *J. Eng. Res.* **2023**, *11*, 239–257. [\[CrossRef\]](#)
4. Soilán, M.; Nóvoa, A.; Sánchez-Rodríguez, A.; Riveiro, B.; Arias, P. Semantic Segmentation of Point Clouds with Pointnet and Kpconv Architectures Applied to Railway Tunnels. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *V-2-2020*, 281–288. [\[CrossRef\]](#)
5. Sánchez-Rodríguez, A.; Riveiro, B.; Soilán, M.; González-deSantos, L.M. Automated detection and decomposition of railway tunnels from Mobile Laser Scanning Datasets. *Autom. Constr.* **2018**, *96*, 171–179. [\[CrossRef\]](#)
6. Fahle, L.; Petruska, A.J.; Walton, G.; Brune, J.F.; Holley, E.A. Development and Testing of Octree-Based Intra-Voxel Statistical Inference to Enable Real-Time Geotechnical Monitoring of Large-Scale Underground Spaces with Mobile Laser Scanning Data. *Remote Sens.* **2023**, *15*, 1764. [\[CrossRef\]](#)
7. Walton, G.; Diederichs, M.S.; Weinhardt, K.; Delaloye, D.; Lato, M.J.; Punkkinen, A. Change detection in drill and blast tunnels from point cloud data. *Int. J. Rock Mech. Min. Sci.* **2018**, *105*, 172–181. [\[CrossRef\]](#)
8. Singh, S.K.; Banerjee, B.P.; Lato, M.J.; Sammut, C.; Raval, S. Automated rock mass discontinuity set characterisation using amplitude and phase decomposition of point cloud data. *Int. J. Rock Mech. Min. Sci.* **2022**, *152*, 105072. [\[CrossRef\]](#)
9. Gallwey, J.; Eyre, M.; Coggan, J. A machine learning approach for the detection of supporting rock bolts from laser scan data in an underground mine. *Tunn. Undergr. Space Technol.* **2021**, *107*, 103656. [\[CrossRef\]](#)
10. Mills, G.; Fotopoulos, G. Rock Surface Classification in a Mine Drift Using Multiscale Geometric Features. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1322–1326. [\[CrossRef\]](#)
11. Tatsch, C.; Breda, J.A.; Covell, D.; Tulu, I.B.; Gu, Y. Rhino: An Autonomous Robot for Mapping Underground Mine Environments. In Proceedings of the 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Seattle, WA, USA, 28–30 June 2023; pp. 1166–1173. [\[CrossRef\]](#)
12. Papachristos, C.; Khattak, S.; Mascarich, F.; Alexis, K. Autonomous Navigation and Mapping in Underground Mines Using Aerial Robots. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019; pp. 1–8.
13. Weidner, L.; Walton, G.; Kromer, R. Classification methods for point clouds in rock slope monitoring: A novel machine learning approach and comparative analysis. *Eng. Geol.* **2019**, *263*, 105326. [\[CrossRef\]](#)
14. Walton, G.; Mills, G.; Fotopoulos, G.; Radovanovic, R.; Standcliffe, R.P.W. An approach for automated lithological classification of point clouds. *Geosphere* **2016**, *12*, 1833–1841. [\[CrossRef\]](#)
15. Axelsson, P. Processing of laser scanner data—Algorithms and applications. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 138–147. [\[CrossRef\]](#)
16. Vosselman, G. Point cloud segmentation for urban scene classification. *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *XL-7/W2*, 257–262. [\[CrossRef\]](#)
17. Yan, W.Y.; Shaker, A.; El-Ashmawy, N. Urban land cover classification using airborne LiDAR data: A review. *Remote Sens. Environ.* **2015**, *158*, 295–310. [\[CrossRef\]](#)
18. Becker, C.; Häni, N.; Rosinskaya, E.; d’Angelo, E.; Strecha, C. Classification of Aerial Photogrammetric 3D Point Clouds. *arXiv* **2017**, arXiv:1705.08374. [\[CrossRef\]](#)
19. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [\[CrossRef\]](#)
20. Beni, T.; Nava, L.; Gigli, G.; Frodella, W.; Catani, F.; Casagli, N.; Gallego, J.I.; Margottini, C.; Spizzichino, D. Classification of rock slope cavernous weathering on UAV photogrammetric point clouds: The example of Hegra (UNESCO World Heritage Site, Kingdom of Saudi Arabia). *Eng. Geol.* **2023**, *325*, 107286. [\[CrossRef\]](#)
21. Xie, Y.; Tian, J.; Zhu, X.X. A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 38–59. [\[CrossRef\]](#)
22. Thomas, H.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Gall, Y.L. Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018. [\[CrossRef\]](#)
23. Farmakis, I.; Bonneau, D.; Hutchinson, D.J.; Vlachopoulos, N. Targeted Rock Slope Assessment Using Voxels and Object-Oriented Classification. *Remote Sens.* **2021**, *13*, 1354. [\[CrossRef\]](#)
24. Bonneau, D.A.; DiFrancesco, P.-M.; Hutchinson, D.J. A method for vegetation extraction in mountainous terrain for rockfall simulation. *Remote Sens. Environ.* **2020**, *251*, 112098. [\[CrossRef\]](#)
25. Carbonell-Rivera, J.P.; Estornell, J.; Ruiz, L.Á.; Crespo-Peremarch, P.; Almonacid-Caballer, J.; Torralba, J. Class3Dp: A supervised classifier of vegetation species from point clouds. *Environ. Model. Softw.* **2024**, *171*, 105859. [\[CrossRef\]](#)
26. Atik, M.E.; Duran, Z.; Seker, D.Z. Explainable Artificial Intelligence for Machine Learning-Based Photogrammetric Point Cloud Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 5834–5846. [\[CrossRef\]](#)

27. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
28. Thomas, H.; Qi, C.R.; Deschaud, J.-E.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6410–6419.
29. Wu, Q.; Osco, L.P. samgeo: A Python package for segmenting geospatial data with the Segment Anything Model (SAM). *J. Open Source Softw.* **2023**, *8*, 5663. [\[CrossRef\]](#)
30. Yarroudh, A. LiDAR Automatic Unsupervised Segmentation using Segment-Anything Model (SAM) from Meta AI. 2023. Available online: <https://github.com/Yarroudh/segment-lidar> (accessed on 23 December 2023).
31. Farmakis, I.; DiFrancesco, P.-M.; Hutchinson, D.J.; Vlachopoulos, N. Rockfall detection using LiDAR and deep learning. *Eng. Geol.* **2022**, *309*, 106836. [\[CrossRef\]](#)
32. Li, N.; Kahler, O.; Pfeifer, N. A Comparison of Deep Learning Methods for Airborne Lidar Point Clouds Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 6467–6486. [\[CrossRef\]](#)
33. Letard, M.; Lague, D.; Le Guennec, A.; Lefèvre, S.; Feldmann, B.; Leroy, P.; Girardeau-Montaut, D.; Corpetti, T. 3DMASC: Accessible, explainable 3D point clouds classification. Application to BI-spectral TOPO-bathymetric lidar data. *ISPRS J. Photogramm. Remote Sens.* **2024**, *207*, 175–197. [\[CrossRef\]](#)
34. Hackel, T.; Wegner, J.D.; Schindler, K. Fast Semantic Segmentation of 3d Point Clouds with Strongly Varying Density. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *III-3*, 177–184. [\[CrossRef\]](#)
35. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigon, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. *arXiv* **2020**, arXiv:1911.11236. [\[CrossRef\]](#)
36. Rudin, C. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.* **2019**, *1*, 206–215. [\[CrossRef\]](#)
37. Singh, S.K.; Raval, S.; Banerjee, B. Roof bolt identification in underground coal mines from 3D point cloud data using local point descriptors and artificial neural network. *Int. J. Remote Sens.* **2021**, *42*, 367–377. [\[CrossRef\]](#)
38. Singh, S.K.; Raval, S.; Banerjee, B. A robust approach to identify roof bolts in 3D point cloud data captured from a mobile laser scanner. *Int. J. Min. Sci. Technol.* **2021**, *31*, 303–312. [\[CrossRef\]](#)
39. Brodu, N.; Lague, D. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: Applications in geomorphology. *ISPRS J. Photogramm. Remote Sens.* **2012**, *68*, 121–134. [\[CrossRef\]](#)
40. Saydam, S.; Liu, B.; Li, B.; Zhang, W.; Singh, S.K.; Raval, S. A Coarse-to-Fine Approach for Rock Bolt Detection From 3D Point Clouds. *IEEE Access* **2021**, *9*, 148873–148883. [\[CrossRef\]](#)
41. Li, S.; Yue, D.; Zheng, D.; Cai, D.; Hu, C. A Geometric-Feature-Based Method for Automatic Extraction of Anchor Rod Points from Dense Point Cloud. *Sensors* **2022**, *22*, 9289. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Xie, Y.; Schindler, K.; Tian, J.; Zhu, X.X. Exploring cross-city semantic segmentation of ALS point clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *XLIII-B2-2021*, 247–254. [\[CrossRef\]](#)
43. Dai, M.; Xing, S.; Xu, Q.; Li, P.; Pan, J.; Zhang, G.; Wang, H. Multiprototype Relational Network for Few-Shot ALS Point Cloud Semantic Segmentation by Transferring Knowledge From Photogrammetric Point Clouds. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5702017. [\[CrossRef\]](#)
44. Weidner, L.; Walton, G. The influence of training data variability on a supervised machine learning classifier for Structure from Motion (SfM) point clouds of rock slopes. *Eng. Geol.* **2021**, *294*, 106344. [\[CrossRef\]](#)
45. Leica Geosystems Leica ScanStation C10 Product Specifications. Available online: https://www.universityofgalway.ie/media/publicsub-sites/engineering/files/Leica_ScanStation_C10_DS.pdf (accessed on 3 October 2024).
46. Leica Geosystems Leica HDS6000 Product Specification. Available online: https://www.geotech.sk/OLD/teo4_Leica%20HDS6000_EN.pdf (accessed on 3 October 2024).
47. FARO Technical Specification Sheet for the Focus Laser Scanner. FARO® Knowledge Base. Available online: https://knowledge.faro.com/Hardware/Focus/Focus/Technical_Specification_Sheet_for_the_Focus_Laser_Scanner (accessed on 3 October 2024).
48. Weidner, L.; Walton, G.; Kromer, R. Generalization considerations and solutions for point cloud hillslope classifiers. *Geomorphology* **2020**, *354*, 107039. [\[CrossRef\]](#)
49. Girardeau-Montaut, D. CloudCompare, Version 2.12. Available online: <https://cloudcompare.org> (accessed on 30 January 2021).
50. Weidner, L. Terpunkto. MATLAB. 21 January 2022. Available online: <https://github.com/lmweidner/terpunkto> (accessed on 23 December 2023).
51. Weidner, L.; Walton, G.; Krajnovich, A. Classifying rock slope materials in photogrammetric point clouds using robust color and geometric features. *ISPRS J. Photogramm. Remote Sens.* **2021**, *176*, 15–29. [\[CrossRef\]](#)
52. Weinmann, M.; Jutzi, B.; Mallet, C.; Weinmann, M. Geometric Features and Their Relevance for 3d Point Cloud Classification. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *IV-1/W1*, 157–164. [\[CrossRef\]](#)
53. Gregorutti, B.; Michel, B.; Saint-Pierre, P. Correlation and variable importance in random forests. *Stat. Comput.* **2017**, *27*, 659–678. [\[CrossRef\]](#)
54. Darst, B.F.; Malecki, K.C.; Engelman, C.D. Using recursive feature elimination in random forest to account for correlated variables in high dimensional data. *BMC Genet.* **2018**, *19*, 65. [\[CrossRef\]](#) [\[PubMed\]](#)

55. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, OR, USA, 2–4 August 1996; AAAI Press: Washington, DC, USA, 1996; pp. 226–231.
56. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. Semantic3d.net: A New Large-Scale Point Cloud Classification Benchmark. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *IV-1/W1*, 91–98. [\[CrossRef\]](#)
57. Lin, W.; Sheil, B.; Zhang, P.; Zhou, B.; Wang, C.; Xie, X. Seg2Tunnel: A hierarchical point cloud dataset and benchmarks for segmentation of segmental tunnel linings. *Tunn. Undergr. Space Technol.* **2024**, *147*, 105735. [\[CrossRef\]](#)
58. Cui, H.; Li, J.; Mao, Q.; Hu, Q.; Dong, C.; Tao, Y. STSD: A large-scale benchmark for semantic segmentation of subway tunnel point cloud. *Tunn. Undergr. Space Technol.* **2024**, *150*, 105829. [\[CrossRef\]](#)
59. Crawford, M.M.; Tuia, D.; Yang, H.L. Active Learning: Any Value for Classification of Remotely Sensed Data? *Proc. IEEE* **2013**, *101*, 593–608. [\[CrossRef\]](#)
60. Nguyen, V.-L.; Shaker, M.H.; Hüllermeier, E. How to measure uncertainty in uncertainty sampling for active learning. *Mach. Learn.* **2022**, *111*, 89–122. [\[CrossRef\]](#)
61. Matskevych, A.; Wolny, A.; Pape, C.; Kreshuk, A. From Shallow to Deep: Exploiting Feature-Based Classifiers for Domain Adaptation in Semantic Segmentation. *Front. Comput. Sci.* **2022**, *4*, 805166. [\[CrossRef\]](#)
62. Segev, N.; Harel, M.; Mannor, S.; Crammer, K.; El-Yaniv, R. Learn on Source, Refine on Target: A Model Transfer Learning Framework with Random Forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1811–1824. [\[CrossRef\]](#)
63. HassanPour Zonoozi, M.; Seydi, V. A Survey on Adversarial Domain Adaptation. *Neural Process. Lett.* **2023**, *55*, 2429–2469. [\[CrossRef\]](#)
64. Tan, C.; Guan, J. A Feature Space Alignment Learning Algorithm. In *PRICAI 2012: Trends in Artificial Intelligence*; Anthony, P., Ishizuka, M., Lukose, D., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 795–800. [\[CrossRef\]](#)
65. Ye, S.; Yin, Z.; Fu, Y.; Lin, H.; Pan, Z. A multi-granularity semisupervised active learning for point cloud semantic segmentation. *Neural Comput. Appl.* **2023**, *35*, 15629–15645. [\[CrossRef\]](#)
66. Karantanellis, E.; Marinos, V.; Vassilakis, E.; Christaras, B. Object-Based Analysis Using Unmanned Aerial Vehicles (UAVs) for Site-Specific Landslide Assessment. *Remote Sens.* **2020**, *12*, 1711. [\[CrossRef\]](#)
67. Farmakis, I.; Bonneau, D.; Hutchinson, D.J.; Vlachopoulos, N. Supervoxel-Based Multi-Scale Point Cloud Segmentation Using Fnea for Object-Oriented Rock Slope Classification Using Tls. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *XLIII-B2-2020*, 1049–1056. [\[CrossRef\]](#)
68. Liao, L.; Tang, S.; Liao, J.; Li, X.; Wang, W.; Li, Y.; Guo, R. A Supervoxel-Based Random Forest Method for Robust and Effective Airborne LiDAR Point Cloud Classification. *Remote Sens.* **2022**, *14*, 1516. [\[CrossRef\]](#)
69. Özdemir, E.; Remondino, F.; Golkar, A. Aerial Point Cloud Classification with Deep Learning and Machine Learning Algorithms. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-4-W18*, 843–849. [\[CrossRef\]](#)
70. Zhang, Y.-M.; Cheng, C.-Y.; Lin, C.-L.; Lee, C.-C.; Fan, K.-C. Develop a Lightweight Convolutional Neural Network to Recognize Palms Using 3D Point Clouds. *Information* **2023**, *14*, 381. [\[CrossRef\]](#)
71. Imad, M.; Doukhi, O.; Lee, D.-J. Transfer Learning Based Semantic Segmentation for 3D Object Detection from Point Cloud. *Sensors* **2021**, *21*, 3964. [\[CrossRef\]](#)
72. Jeon, W.; Ko, G.; Lee, J.; Lee, H.; Ha, D.; Ro, W.W. Chapter Six—Deep learning with GPUs. In *Advances in Computers*; Kim, S., Deka, G.C., Eds.; Hardware Accelerator Systems for Artificial Intelligence and Machine Learning; Elsevier: Amsterdam, The Netherlands, 2021; Volume 122, pp. 167–215. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.