

1 **A machine learning approach for the detection of supporting rock** 2 **bolts from laser scan data in an underground mine**

3 Jane Gallwey ^{a,*}, Matthew Eyre ^a, John Coggan ^a

4 ^a *Camborne School of Mines, University of Exeter, Penryn, United Kingdom*

5 ** Corresponding author: Jane Gallwey: j.gallwey@exeter.ac.uk, Tremough Campus, Penryn, United Kingdom, TR109AH*

6 **Keywords:** mining; rock bolting; machine learning; laser scanning; object detection

7 **Abstract**

8 Rock bolts are a crucial part of underground infrastructure support; however, current methods to locate
9 and record their positions are manual, time consuming and generally incomplete. This paper describes
10 an effective method to automatically locate supporting rock bolts from a 3D laser scanned point cloud.
11 The proposed method utilises a machine learning classifier combined with point descriptors based on
12 neighbourhood properties to classify all data points as either ‘bolt’ or ‘not-bolt’ before using the Density
13 Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm to divide the results into
14 candidate bolt objects. The centroids of these objects are then computed and output as simple
15 georeferenced 3D coordinates to be used by surveyors, mine managers and automated machines. Two
16 classifiers were tested, a random forest and a shallow neural network, with the neural network providing
17 the more accurate results. Alongside the different classifiers, different input feature types were also
18 examined, including the eigenvalue based geometric features popular in the remote sensing community
19 and the point histogram based features more common in the mobile robotics community. It was found
20 that a combination of both feature sets provided the strongest results. The obtained precision and recall
21 scores were 0.59 and 0.70 for the individual laser points and 0.93 and 0.86 for the bolt objects. This
22 demonstrates that the model is robust to noise and misclassifications, as the bolt is still detected even if
23 edge points are misclassified, provided that there are enough correct points to form a cluster. In some
24 cases, the model can detect bolts which are not visible to the human interpreter.

25 1 Introduction

26 Rock reinforcement is a crucial element of underground construction. When operating with any
27 underground excavation, an understanding of the rock mass characteristics as an engineering material is
28 critical in ensuring that risks from tunnel collapse are mitigated through the use of ground control
29 methods. Installation of rock bolts is the most widely used form of ground support (Li, 2017). The design
30 of such a system is site dependent and based on the mechanical behaviour of the rock mass, the in-situ
31 stress field and induced stress from the excavation (Hoek and Brown, 1982). In low stress conditions,
32 compression of the ground is needed to ensure loose blocks do not fall. This can be achieved either by
33 using spot bolting of discrete blocks or by a systematic bolting pattern. Spot bolting is carried out where
34 needed without following a set spacing, whereas systematic patterns are used to add a compression arch
35 to the rock mass, reducing the potential for unravelling. Schach et al (1979) shows that an increase in
36 bolt spacing leads to less interaction of neighbouring bolts, reducing the size of the compression zone
37 to a point at which the bolts no longer provide a wide coverage leading to potential fall of ground. To
38 ensure the required level of compressive cover is produced, it is important that correct installation of
39 bolt patterns is carried out. Reconciliation of installed bolts is therefore an important part of the ground
40 management process to ensure safe working underground.

41 Current methods of documenting rock bolt installation are usually hand sketch based and not
42 comprehensive (Öberg, 2013) due to the large volume of bolts that have to be recorded and the difficulty
43 and time-consuming nature of manually surveying such data, along with the associated human error for
44 this type of repetitive task. Another difficulty is that in many applications the entire surface is covered
45 with shotcrete after installation, rendering the exact locations of the rock bolts unknown or challenging
46 to discern (Öberg, 2013). Automatically detecting and recording the 3D coordinates of rock bolts either
47 retrospectively or at installation would allow for greater quality assurance and quality control, providing
48 a detailed record of exactly where rock bolts have been installed. These records also would be critical
49 in a fall of ground situation, where the exact bolting configuration that was installed prior to the incident
50 must be determined. Advancements in remote sensing techniques and machine learning algorithms
51 could allow this bolting pattern information to be obtained. However, currently the mining sector is not

52 fully utilising these new technologies despite being well placed to employ them due to a widespread
53 adoption of laser scanners and other high resolution surveying technologies both onboard vehicles and
54 as standalone survey technologies (Body, 2014).

55 To date, image based photogrammetric systems for automatically inspecting civil engineering tunnels
56 have been the primary research focus in this area. A review of these techniques is given in Attard et al.,
57 (2018) and successful implementations for crack detection by Huang et al. (2018) and moisture mark
58 detection by Zhao et al. (2020), demonstrating the power of remote sensing and machine learning for
59 underground infrastructure management. However, passive remote sensing methods such as
60 photogrammetry can be problematic underground, particularly in mines, due to challenges from uneven
61 illumination and dust (Gikas, 2012). Active systems such as laser scanning circumvent these issues, by
62 measuring using multiple high speed laser pulses emitted from the instrument itself (Eyre et al., 2016).
63 The data obtained from a laser scanner is in the form of a 3D point cloud which records the X, Y, Z
64 coordinates of the reflected point in 3D. Most scanners also record the intensity of the laser return and
65 some also use cameras to store an RGB colour value for each point. The primary issue with laser
66 scanners compared to cameras is the size of the data collected and the subsequent difficulty in efficiently
67 processing it. The raw output from the laser scanner is a large unordered set of 3D coordinates with no
68 semantic knowledge of the object they are surveying. This 3D point cloud data is currently used by
69 mines directly for surveying tasks such as change detection, geometric analysis and as-built to design
70 comparison (van der Merwe and Andersen, 2013). In order for this data to be utilised in a wider range
71 of applications such as automated machines, mine information databases and infrastructure monitoring
72 a level of semantic information needs to be added to the data, along with a reduction in the dataset size.
73 The only directly applicable prior work on this topic is by Martínez-Sánchez et al. (2016). In this paper
74 they built and trained an autoencoder based model to detect not only the rock bolts from laser scan data,
75 but also their orientations and the shotcrete thickness. Their work achieved a very high accuracy of 91%
76 showing that geometric neighbourhood based machine learning algorithms have great potential to solve
77 this engineering and monitoring problem. Laser scanners also have been used in tunnel inspection (Tan
78 et al., 2016, Xu et al., 2018) however, these studies have used the laser scan data to generate intensity

79 images rather than detecting objects from the 3D point cloud data. Soilán et al., (2019) give a full review
80 of the use of laser scanners for infrastructure monitoring; whilst there is minimal published work on
81 detecting discrete objects in an underground environment from laser scanned data, automatically
82 generating an understanding of a scene from point cloud data has been the topic of much research in
83 recent years. Most application oriented work in this field focuses on either identifying roadside objects
84 and road characteristics from surface mobile laser scan data (Yang et al., 2013, Lehtomäki et al., 2016,
85 Soilán et al., 2017 and Balado et al., 2018) or on ground cover classification from aerial LiDAR data
86 (Blomley et al., 2016), (Niemeyer et al., 2014) and (Rau et al., 2015). Properties of these types of surface
87 scenes, such as proliferation of regular vertical objects in streetscapes and a mostly fixed view angle in
88 aerial LiDAR can be leveraged to aid in detecting these types of objects, unlike in the underground
89 environment. Underground terrestrial and mobile laser scan data is complex as it is true 3D data, with
90 the possibility of multiple points sharing the same XY location but possessing different Z values.
91 Approaches used for identification of discrete objects on roads, such as Weinmann et al. (2017) for
92 trees and Lehtomäki et al. (2010) for poles can be considered the closest neighbours, and techniques
93 from these studies can be adapted to the problem of identifying underground features or objects with
94 regard to the particular properties of the underground environment.

95 The classical method for point cloud object detection is described in Weinmann et al. (2015a) and
96 involves three steps: neighbourhood selection, feature extraction and classification. Other methods that
97 do not follow this framework include directly classifying using Markov networks (Anguelov et al.,
98 2005), (Agrawal et al., 2009) and (Triebel et al., 2006), spectral hashing (Behley et al., 2010) and most
99 recently, approaches using deep learning. Whilst deep learning approaches have shown impressive
100 results (Maturana and Scherer, 2015, Qi et al., 2016, Riegler et al., 2017), the additional model
101 complexity, computational power, training time and the size of the training data required for successful
102 deployment make these methods less attractive for an efficient vehicle-based solution. For this
103 application the classical approach similar to Weinmann et al. (2017) was selected, allowing a low
104 computational burden which is more appropriate for time critical applications such as those deployed
105 on underground vehicles and equipment.

106 This paper will describe an automated approach for rock bolt identification from laser scan data using
107 machine learning. The method is based on the classical point cloud semantic segmentation technique
108 defined in Weinmann et al., (2015a), but implemented using a more extensive set of features from both
109 the robotics and remote sensing communities, alongside adaptations for the geometry of underground
110 environments. The machine learning element of the research compares the two popular classifiers, a
111 random forest and an artificial neural network. Following the classification, the bolt objects are extracted
112 via clustering and centroid generation. Section 1 outlines the task and examines related work, Section 2
113 describes the datasets used for model development, Section 3 details the methodology, Section 4
114 presents the results, Section 5 discusses the findings and Section 6 concludes the work.

115 2 Datasets

116 A large amount of labelled data is required to train a machine learning classifier to detect objects. As
117 there are no available datasets of labelled laser scanned rock bolts, a dataset was collected and annotated
118 specifically for this study. The data was collected from a 250m section of underground workings in a
119 small tin and copper mine. This is a good training area, as the slaty nature of the country rock manifests
120 itself as a fair to poor quality rock mass, resulting in extensive spot bolting based on observations of
121 potential block fallouts. The area of interest was surveyed using a terrestrial laser scanning workflow.
122 The scanner was mounted on a static tripod to perform a scan, next the scanner was moved to a position
123 approximately 12m further down the tunnel and another scan was taken. This process was repeated for
124 25 scans. The individual scans were registered together to make a unified dataset in the point cloud
125 processing software Leica Cyclone. The hardware used was a Leica C10 laser scanner, as used in other
126 underground studies such as Ganić et al. (2011), Chen et al. (2018) and Long et al. (2018). This
127 instrument has a specified accuracy of $\pm 6\text{mm}$ per point (Long et al., 2018) and the scan resolution at
128 the chosen setting provides a point spacing of 5mm at 5m from the scanner. The scanner was set to
129 record only laser intensities not optical imagery values. This is due to the poor illumination in the mine
130 and the additional time required to take photographs with the inbuilt camera. The final dataset is
131 representative of real world underground scan data, containing laser noise, occlusions and many objects

132 that are neither tunnel nor bolt and it has not been manually cleaned and simplified for improved
133 machine learning results. A sample of the data is shown in Figure 1.



134
135 *Figure 1: A view of the underground data. Many challenging objects are present including pipes, brackets, ventilation bagging*
136 *and electrical boxes. The colour scheme is taken from the strength of the laser return.*

137 To generate the training data, the rock bolt points were manually separated from all other points and
138 given the class label 1 ‘bolt’. All other objects were labelled 0 ‘not-bolt’, including confusion objects
139 such as pipes, brackets and ventilation bagging, alongside the hanging wall, side wall and foot wall
140 surfaces. This dataset was then split into sections for training, cross-validation and testing, as shown in
141 Figure 2. The grey areas are unused and have been reserved for future algorithm testing.

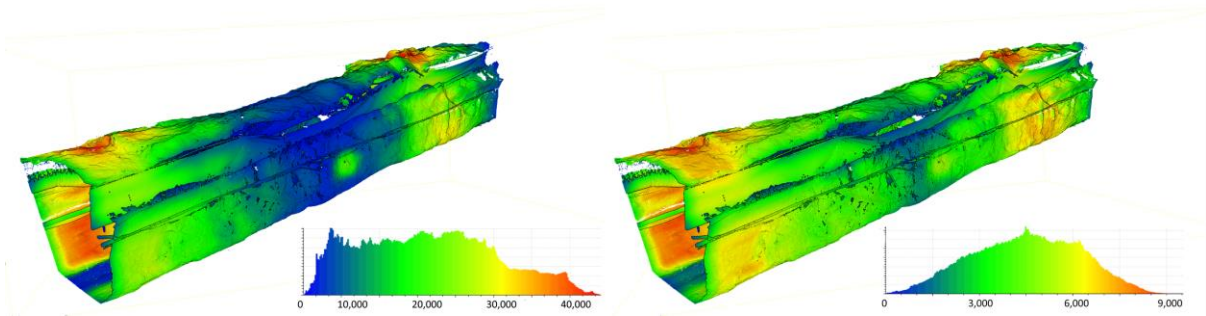


142
143 *Figure 2: Tunnel showing the areas for training (blue), cross-validation (green) and testing (red).*

144 2.1 Pre-processing

145 Before the point cloud dataset features can be generated a number of preprocessing steps are carried out,
146 using the open source software CloudCompare (Girardeau-Montaut, 2016). Firstly, the point clouds are
147 shifted from their real-world coordinates to a position near the origin to avoid potential precision loss
148 from processing very large numbers. Next, denoising is carried out using an algorithm which works
149 similarly to a low pass filter. This removes points which are further than a set factor of their neighbours
150 reprojection error onto a plane, where the plane itself is fitted to all points within a specified radius
151 (Girardeau-Montaut, 2016). The denoising settings used a radius of 10cm and a relative error factor of

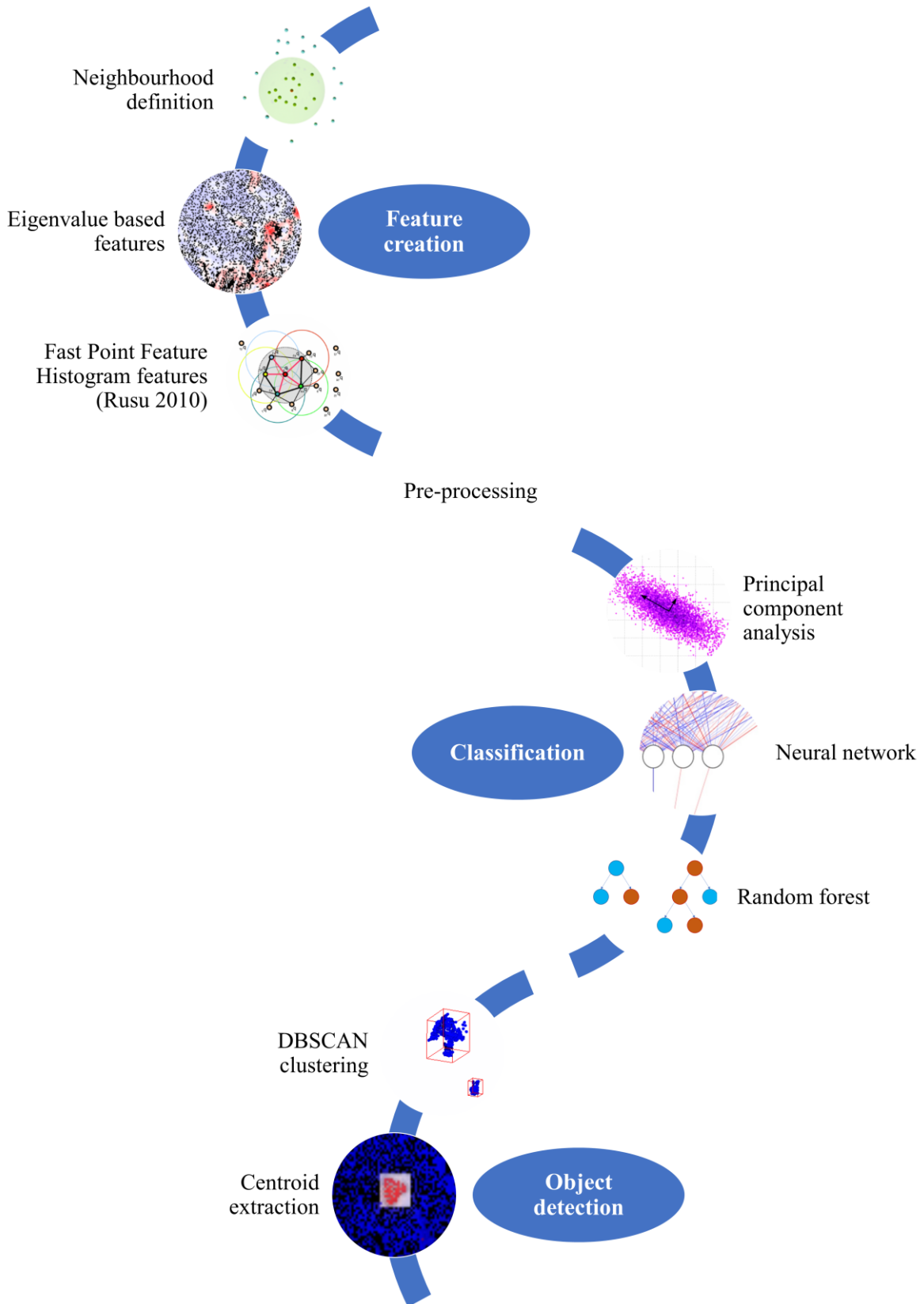
152 1. The final step in the base dataset creation is density reduction. Point clouds acquired from laser
153 scanners have a large variation in density due to many factors including an object's distance from the
154 scanner, the scan angle, overlap between neighbouring scans and occlusions. Whilst it is difficult to
155 create new points in areas of low density, it is straightforward to remove points in areas of high density
156 using resampling techniques. For this application the point cloud was spatially resampled to a density
157 of 1 point per cm maximum. Figure 3 shows the distribution of point densities before and after
158 resampling. The resampling algorithm also reduced the total number of points by ~40%. As shown in
159 Figure 3, the density range is now closer to a normal distribution, but still not constant across the point
160 cloud. This is because a constant density is undesirable for real world data, as there will always be areas
161 of low sampling due to occlusions, however, if the majority of the cloud is downsampled to match the
162 lowest density much of the useful detail can be lost. Following density reduction, the dataset contains
163 10 million labelled laser scanned points.



164
165 *Figure 3: A section of the training data showing the density before (left) and after (right) spatial resampling. The density is*
166 *measured as the number of points per square meter of tunnel surface. The graphs below each image show the range of data*
167 *densities, before resampling it ranges from 0-40,000 and after it ranges from 0-9,000.*

168 3 Methodology

169 The workflow for detecting bolts from the laser scanned point cloud dataset has three primary
170 components: feature descriptor creation, machine learning classification and object creation. An
171 overview of the processing workflow is shown in Figure 4.



172

173 *Figure 4: Methodology diagram outlining the pipeline used for the task of identification of rock bolts from the laser scan data.*

174 3.1 Feature creation

175 Single laser scanned points are not adequate descriptors of the data they represent, as they contain only
176 3D cartesian coordinates and an intensity value. In isolation, this information is insufficient to describe
177 what type of object this point belongs to; therefore, the point cloud data must be encoded in a way that
178 allows a machine learning algorithm to differentiate between object types. This can be achieved by
179 describing each point in relation to the geometry of its neighbouring points, these descriptors are known
180 as features. The most popular features in the remote sensing community are based on the eigenvalues of
181 the point neighbourhood. Early work by Pauly et al. (2003) and Vandapel et al. (2004) introduced the
182 concept, which was extended by Jutzi and Gross (2009) and Weinmann et al. (2015b). The other
183 common features are proposed by Rusu (2010) and implemented by him in his Point Cloud Library
184 (PCL). This approach computes a fast point feature histogram (FPFH) based on the angular variations
185 between the normals of the points using a Darboux frame (Rusu et al., 2009).

186 For choosing a point neighbourhood, the dimensions of the object to be detected and the spacing of
187 points in the point cloud determine the optimum value. A typical mechanically anchored rock bolt
188 measures 16cm across the faceplate. Computing the number of neighbours per point over the resampled
189 point cloud using an 8cm radius found the mean number of neighbours to be close to 100, therefore, this
190 is a suitable neighbourhood size to adequately capture the geometry of a rock bolt. Once the
191 neighbourhoods have been defined, descriptive features can be constructed for each point using its
192 neighbours.

193 Two types of feature sets are calculated for each point in the cloud. The first are the ‘Geometric’ features,
194 described in Weinmann (2016). These include simple 2D and 3D properties of the neighbourhood
195 (density, vertical difference, minimum bounding box), eigenvalue based features which describe the
196 local shape properties of the neighbourhood and 2D accumulation map based features, an overview of
197 each individual feature is given in Table 1. These features were calculated using python code adapted
198 from the MATLAB script published by Weinmann et al. (2015a). The 2D accumulation map features
199 have the highest processing overhead and also are potentially less descriptive for an underground
200 scenario where the hanging wall and foot wall share the same XY coordinates, to investigate, the feature

201 sets were generated both with and without these features. The geometric feature set is powerful as it is
202 understandable and can be easily visualised, Figure 5 shows a small section of hanging wall with the
203 points coloured by the magnitude of different features. It can be seen that certain features are intuitively
204 better at differentiating between ‘bolts’ and ‘not-bolts’ for a human interpreter; however, some of the
205 less obvious features may be still be strong descriptors as they can help to separate between false
206 positives and true positives.

207 The second type of features used are the fast point feature histogram features (FPFH) proposed by Rusu
208 (2010). This type of feature representation uses the relationships between the points in the
209 neighbourhood and their normal vectors to describe the local geometry around the point. This is
210 calculated for each pair of points by defining a fixed Darboux coordinate frame at one point and using
211 it to compute the three angles which define the difference between the normal vectors. The complexity
212 is then reduced by not computing the same neighbourhood pairs for multiple points and instead using a
213 weighting scheme. Finally, the values are binned into a 33 bin histogram. Full derivation of the FPFH
214 is found in Rusu (2009). This step was implemented in C++ with the Point Cloud Library (Rusu and
215 Cousins, 2011).

216 As the intensity data adds further valuable information about the object, especially underground, two
217 additional features; the intensity of the point itself and the average intensity of the neighbourhood are
218 computed and added to the feature set. As all sets of features are computed individually for each point
219 using the same set K number of neighbours the geometric, FPFH and intensity features can be
220 concatenated, along with the X, Y, Z data for the point and the true class label. The result is a 65-
221 dimensional vector describing the local geometry in a way that can be statistically interpreted by the
222 machine learning classifiers in the next stage, shown in Table 1.

223

224

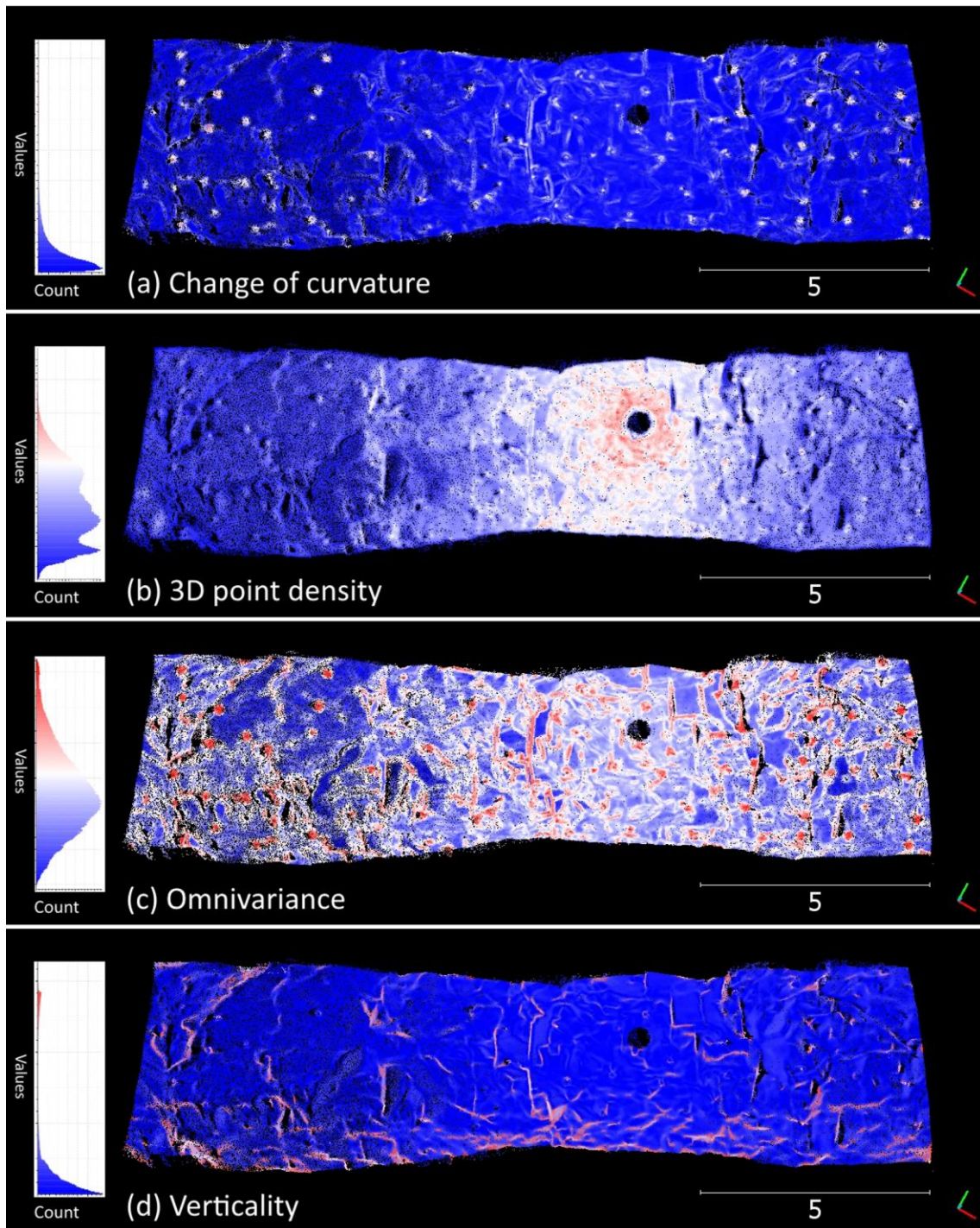
225

226

227 Table 1: Composition of the generated 65 dimensional vector including equations, where k = number of neighbours and λ_n =
 228 eigenvalue n . For classification, the point label column is removed and used as the supervised truth value and the X, Y, Z
 229 coordinates do not contribute to the classification. Eigenvalue equations shown are adapted from Weinmann (2016). For
 230 brevity, each FPFH value is not shown as they are numbered elements from the same histogram, see Rusu et al. (2009) for
 231 mathematical derivations of these features.

No	Name	Description	Equation
1	X	X coordinate of point	n/a
2	Y	Y coordinate of point	n/a
3	Z	Z coordinate of point	n/a
4	Label	Point label	n/a
5	Intensity	Reflectance intensity of point	n/a
6	Linearity	How much variance can be explained by only the largest eigenvalue	$(\lambda_1 - \lambda_2)/\lambda_1$
7	Planarity	How much variance can be explained by the two largest eigenvalues	$(\lambda_2 - \lambda_3)/\lambda_1$
8	Scattering	How much neighbourhood variance can be explained by the smallest eigenvalue	λ_3/λ_1
9	Omnivariance	Volumetric point distribution	$\sqrt[3]{(\lambda_1 \cdot \lambda_2 \cdot \lambda_3)}$
10	Anisotropy	Directional dependence	$(\lambda_1 - \lambda_3)/\lambda_1$
11	Eigenentropy	Order/disorder	$-\lambda_1 \ln(\lambda_1)$ $-\lambda_2 \ln(\lambda_2) - \lambda_3 \ln(\lambda_3)$
12	Sum EVs 3D	Sum of eigenvalues	$\lambda_1 + \lambda_2 + \lambda_3$
13	Curvature change	Local change in curvature	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
14	Z values	Absolute height of point	Z
15	KNN radius 3D	Size of the neighbourhood sphere	r_{knn-3D}
16	Density 3D	Points per m^3	$k + 1/(4/3 \cdot \pi \cdot r_{knn-3D}^3)$
17	Verticality	The difference from vertical of the Z component of the normal vector	$1 - n_z$
18	Change in Z	Maximum height difference	$Z_{max} - Z_{min}$
19	STD of Z	Standard deviation of heights	$\sigma_{Z,knn-3D}$
20	KNN radius 2D	Size of the neighbourhood circle	r_{knn-2D}
21	Density 2D	Points per m^2	$k + 1/\pi \cdot r_{knn-2D}^2$
22	Sum EVs 2D	Sum of eigenvalues from 2D structural tensor	$\lambda_{1-2D} + \lambda_{2-2D}$
23	EV ratio 2D	Ratio of the 2D eigenvalues	$\lambda_{2-2D}/\lambda_{1-2D}$
24	2D map	Frequency accumulation map	n/a
25	D_Z	Change in Z in accumulation map	n/a
26	Std_Z	Standard deviation of Z in accumulation map	n/a
27	EV _{3d-1}	First 3D eigenvalue	λ_1
28	EV _{3d-2}	Second 3D eigenvalue	λ_2
29	EV _{3d-3}	Third 3D eigenvalue	λ_3
30	EV _{2d-1}	First 2D eigenvalue	λ_{1-2D}
31	EV _{2d-2}	Second 2D eigenvalue	λ_{2-2D}
32	Mean_I	Mean intensity	$\sum \frac{(i_1+i_2...+i_k)}{k}$
33	FPFH ₁	FPFH value from bin number 1	n/a
-	---	---	---
65	FPFH ₃₃	FPFH value from bin number 33	n/a

232



233

234 *Figure 5: A section of hanging wall showing each point coloured by its feature value. In (a) the areas of high curvature change*
 235 *clearly correspond to rock bolt locations. In (b) the 3D density appears to be more related to the distance from the scanner*
 236 *than the bolt location, indicating that is probably not a particularly effective feature for locating rock bolts. The omnivariance*
 237 *feature shown in (c) is high for the bolts but also high for other areas of discontinuities, especially visible in the vertical lines*
 238 *near the centre of the image, whereas (d) shows the verticality feature which does not spot rock bolts but does have high*
 239 *values in the same areas of non-bolt discontinuities that were highlighted in (c). All scales are relative, and the colour scheme*
 240 *banding runs from blue (lowest) to red (highest) with white as the median value.*

241 3.2 Classification

242 Once the data has been transformed into meaningful features it can be classified into categories using a
243 variety of machine learning techniques. However, prior to classifier training several pre-processing steps
244 must be carried out to improve the machine interpretability of the data. For the problem of finding rock
245 bolts, the classifier is trained on large hand-labelled where less than 1% of the observed points are rock
246 bolts. If this data were directly used for training, even if the classifier always predicted ‘not-bolt’ it
247 would achieve 99% accuracy. Of the several possible methods of class rebalancing, the one chosen for
248 this study is down-sampling the majority class. Empirical testing on the cross-validation data found a
249 full downsampling (99% reduction) to match the minority class is not as effective as a less severe 80%
250 reduction of the majority class. After downsampling, each feature is standardised by removing the mean
251 and scaling to unit variance. The final classifier inputs are now a collection of m vectors of dimension
252 n where m corresponds to the number of laser scanned points and n is the number of features in the
253 feature set.

254 For learning the point representations, Weinmann et al. (2015a) tested many of the most popular types
255 of classifiers including instance based, rule based, probabilistic, max-margin, ensemble and a simple
256 neural network. They found that the ensemble method random forest performed best, which was the
257 method also chosen by Chehata et al. (2009), Niemeyer et al. (2014), Landrieu et al. (2017) and Hackel
258 et al. (2017). For our study, a preliminary test was carried out using multiple machine learning classifiers
259 including Random Forests (RF), Multi-Layer Perceptron (MLP), Support Vector Machines (SVM),
260 Quadratic Discriminant Analysis, Linear Discriminant Analysis and Naive Bayes. The Linear and
261 Quadratic Discriminant Analyses, along with the Naive Bayes proved unable to effectively classify the
262 bolt points and were not considered further. When comparing the remaining three classifiers, the
263 Random Forest produced higher accuracies on the minority bolt class than the Support Vector Machines;
264 these results agree with those found by Bassier et al. (2019), Kogut and Weistock (2019) and Weinmann
265 et al. (2015). However, the MLP outperformed both the SVM and the RF, this is in contrast to the results
266 observed by Bassier et al. (2019) and Weinmann et al. (2015). It is hypothesised that this difference may
267 be due to the larger number of hyperparameters required to produce a stable result from the MLP

268 classifier, as discussed by Nygren and Jasinski (2016). Based on this initial testing, the classifiers chosen
269 for this work were the Random Forest and the MLP. The Random Forest was chosen as it is one of the
270 highest performing classifiers in the literature and has been proven to be capable of achieving robust
271 high accuracy classifications for problems of this type. The MLP was chosen as it showed the best
272 performance in the initial tests and indicated strong generalisation potential when paired with
273 appropriate hyperparameters.

274 A random forest is a powerful machine learning algorithm based on a randomised forest of decision
275 trees (Breiman, 2001). It has a low number of hyperparameters to tune and is resilient to noise in the
276 data, making it an appropriate choice for remote sensing applications (Pal, 2005). An additional benefit
277 of the random forest classifier is the ability to output a feature importance ranking, allowing for the
278 relative contribution of individual features to the final prediction result to be observed (Strobl et al.,
279 2008). The second classifier, an MLP or artificial neural network, is a node-based architecture which
280 can approximate complex functions by learning weights for every node by a process known as
281 backpropagation (Hecht-Nielsen, 1992). Recent advances in processing power and vast dataset sizes
282 have led to deep learning networks many hundreds of layers deep performing increasingly complex
283 tasks (LeCun et al., 2015).

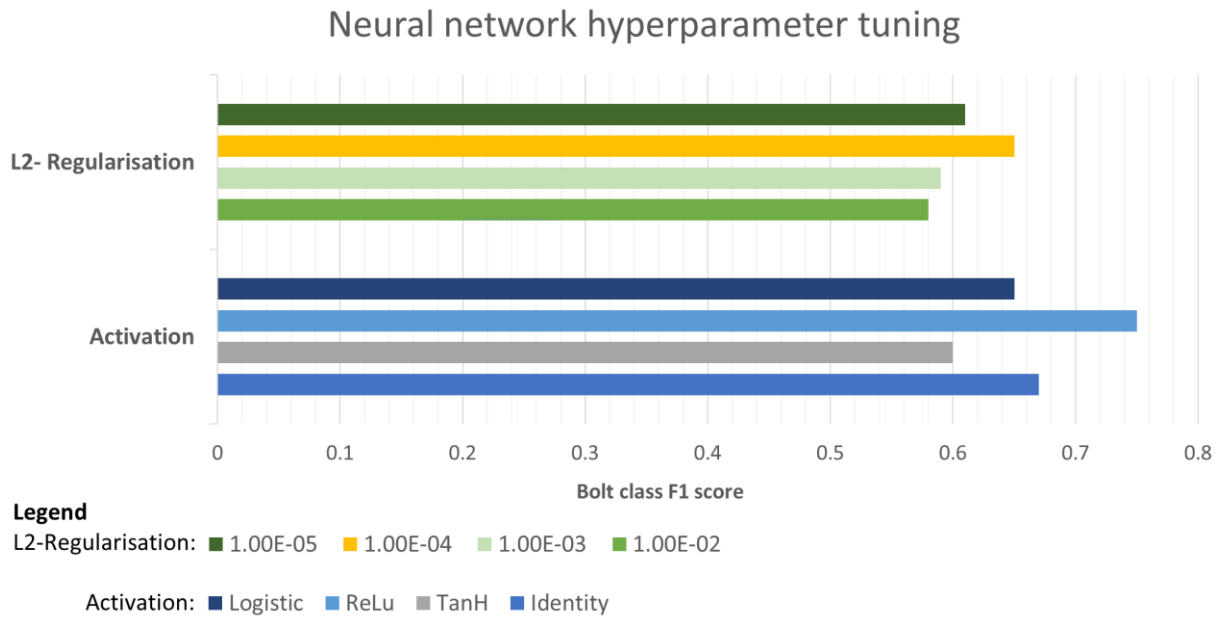
284 The structure chosen for the neural network used in this research is informed by the concept of effective
285 capacity. A deep learning algorithm's effective capacity is its ability to model complexity; good
286 performance is achieved when its effective capacity is appropriate for the complexity of the task and the
287 size of the available training data (Goodfellow et al., 2016). If it has more effective capacity than needed,
288 it will tend to overfit. The task of finding bolt points from multi-dimensional feature vectors requires a
289 relatively small effective capacity, as there are limited generalisation requirements. Combined with the
290 small bespoke training set, an appropriate starting point for the structure was defined as containing no
291 more than three hidden layers with no more than 40 nodes in each layer. Empirical testing was then
292 carried out using a variety of values within this parameter space; stable, effective performance was
293 obtained when the network contained two hidden layers with between 20-30 nodes in the first layer and

294 5-10 in the second layer. The final chosen structure contained 25 nodes in the first hidden layer and 5
295 nodes in the second hidden layer.

296 To decrease processing time, a Principal Component Analysis (PCA) dimensionality reduction (Wold
297 et al., 1987) is performed on the data prior to input, reducing the features from 65 to 40 whilst
298 maintaining 99.4% of the variance. These 40 features are then used as the input to the neural network
299 and are joined to every neuron in the first hidden layer by a weight, with the value of the neuron being
300 the weighted sum of all the features, transformed by the non-linear rectified linear unit (ReLU) function.
301 The second hidden layer has the same structure, with every neuron in each layer connected by weights,
302 and the final output is a binary ('bolt' or 'not-bolt') decision. The network learns by backpropagation
303 using the L-BFGS solver. Both classifications were carried out using the Scikit-learn libraries in Python
304 (Pedregosa et al., 2011).

305 During model training, suitable values for hyperparameters of the classifiers were determined using a
306 dual strategy. Firstly, a randomised search of the probable value space was carried out, using the Scikit-
307 learn model selection tool 'RandomisedSearchCV' (Pedregosa et al., 2011). Taking the results of this
308 search, empirical testing was then carried out above and below the best random search values to
309 determine the exact hyperparameters choice. This hyperparameter tuning was carried out on the cross-
310 validation section of the dataset via two-fold cross-validation. For the random forest, it was found that
311 only the 'number of estimators' hyperparameter affected the results to any appreciable degree.
312 Therefore, to ease repeatability, the random forest hyperparameters were all kept at the Scikit-learn
313 default values except for the 'number of estimators' hyperparameter which was changed to 200.

314 The neural network hyperparameters examined included the solver, the activation function and the L2
315 regularisation term. There was no appreciable difference in accuracy observed from using different
316 solvers, however, the LBFGS converged faster and required fewer additional hyperparameters. Figure
317 6 shows the results from the empirical testing of the L2 regularisation term and activation function,
318 showing that the best accuracies are obtained with an L2-regularisation term of $1e-4$ and the ReLU
319 activation function.



320

321 *Figure 6: Results from the neural network manual hyperparameter tuning.*

322 Object creation

323 The type of machine learning used in this research acts on the features derived for each individual point
 324 in the cloud. Because there is no spatial connectivity, they suffer from noise due to isolated misclassified
 325 points. In the processing pipeline, after the point wise classification, the resulting point cloud is split
 326 using the predicted values and the points that have been labelled as ‘not-bolt’ are now discarded, greatly
 327 reducing the dataset size. The remaining cloud now contains all the correctly predicted bolt points and
 328 the falsely predicted non-bolt points. From visual examination of this remaining cloud, it can be seen
 329 that the point cluster separation is good, with adequate empty space visible between the clusters of
 330 predicted points.

331 Cloud segmentation was carried out using DBSCAN (Density-Based Spatial Clustering of Applications
 332 with Noise). This algorithm finds core samples and generates clusters from high density areas adjacent
 333 to them, allowing for clusters of any shape (Ester et al., 1996, Schubert et al., 2017). The maximum
 334 distance between neighbourhood samples parameter (ϵ) was set to 5cm and the minimum cluster size
 335 was set to 10 points. The ϵ value was chosen based on the heuristic proposed by Ester et al. (1996) of a
 336 suitable value being approximately the distance to the 4th nearest neighbour, in this case 5cm for the
 337 1cm resampled point cloud. The minimum cluster size was set to 10 points; as the ground truth bolt
 338 clusters contained between 20-400 points a number set at 50% of the sparsest bolt cluster was a suitable

339 choice of parameter. The Euclidian distance metric was used as the inputs are coordinates in 3D space
340 and the K-D tree algorithm was used to compute the neighbours as the data dimensionality is low.

341 The final processing step is to calculate the centroid of each cluster to use as the predicted bolt location.
342 The final step is to export these cluster centroids as a X, Y, Z file of only a few kilobytes that can be
343 easily shared with machines and surveyors. This clustering greatly reduces the algorithm's sensitivity
344 to misclassifications in the individual points. Provided at least 10 points from a bolt have been classified
345 correctly the bolt will be detected, reducing missed detections.

346 4 Results

347 The performance of the proposed methodology was assessed on both the raw point prediction accuracy
348 and also on the number of bolts correctly detected. The results were evaluated using the measures of
349 precision and recall, along with the F1 score. These metrics were chosen as others such as the overall
350 accuracy are inadequate in cases such as this, where large class imbalances are present in the data. The
351 precision is defined as the measure of what proportion of the positive predictions are correct; it is the
352 number of true positives divided by the number of all predicted positives (true positives and false
353 positives). The recall is a measure of what proportion of actual positives were correctly identified; it is
354 defined as the number of true positives divided by the number of actual positives (true positives and
355 false negatives). The F1 score is the harmonic mean of the precision and recall.

356 The first experiment tested which set of point feature descriptors provided the most accurate results. It
357 compared the full geometric feature set proposed by Weinmann (2016) consisting of 26 features, a
358 reduced version of this feature set with the accumulation map features removed (23 features), the FPFH
359 features (33 bin histogram), the combined feature sets (59 features) and finally the combined features
360 plus the intensity features (61 features). Table 2 shows the results of the feature set comparison on both
361 classifiers, with the F1 score used as the performance metric. For this test the PCA reduction was not
362 carried out on the neural network dataset to more clearly isolate the effect feature sets have on the results.
363 The random forest classifier also outputted the feature importance rankings, shown in Figure 8 and

364 discussed in Section 5. As the combined features with intensity achieved the highest accuracy, this was
 365 the feature set used for the final model which was applied to the unseen test data.

366 *Table 2: F1 scores for differing feature sets. The reduced geometric features refer to the set with the 3 highest computation*
 367 *time features (accumulation maps) removed.*

Feature set	Geometric features full	Geometric features reduced	FPFH features	Combined features	Combined features and intensity
<i>No. features</i>	26	23	33	59	61
Neural network	0.42	0.41	0.51	0.63	0.64
Random forest	0.49	0.43	0.37	0.56	0.58

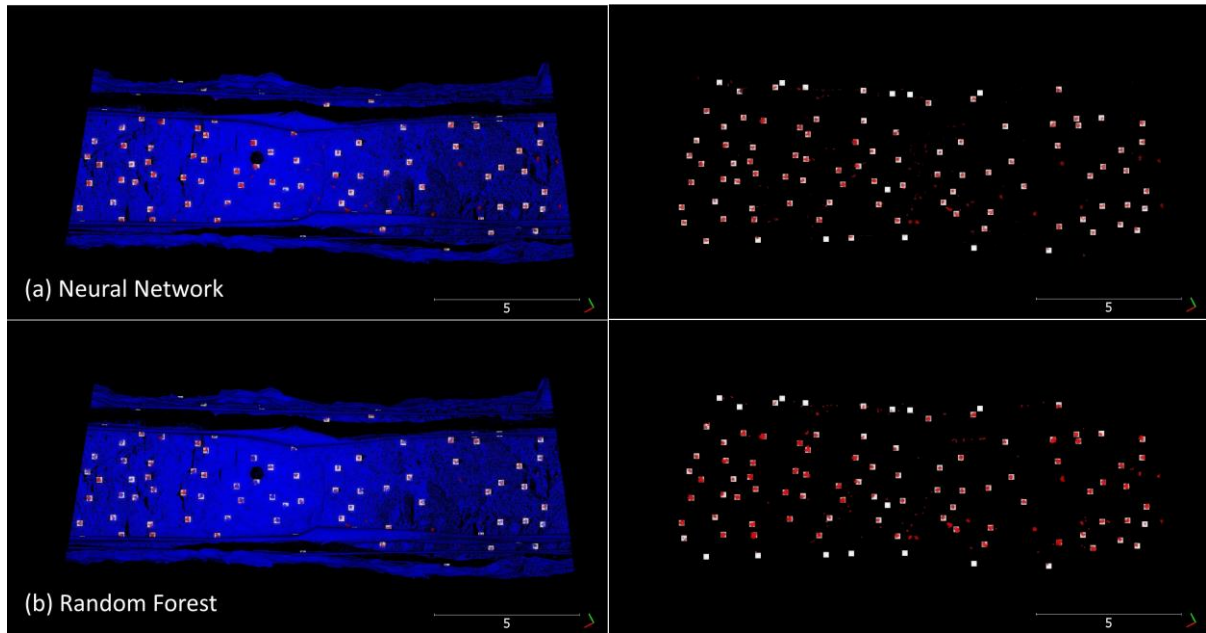
368

369 Once the feature set choice was finalised, the per point prediction results were examined against the
 370 human generated ones for the test data, totalling almost 1.5 million point predictions. These results are
 371 given in Table 3. Figure 7 gives a graphical view of the point prediction results. In this figure the footwall
 372 has been removed and the viewing angle is directly vertical towards the hanging wall. The predicted
 373 bolt points are shown in red, the predicted not-bolt points in blue and the overlaid white squares show
 374 the true bolt locations. The left images show the whole point cloud and the right images show just the
 375 predicted bolt class points after the removal of the not-bolt predictions. In the right-hand images,
 376 anywhere that the red points do not have a corresponding white square overlay indicates incorrect
 377 objects classified as bolts, and any white squares without red points indicate missed bolts.

378 *Table 3: Results from the point-wise classifiers in the test dataset.*

Neural network	<i>Predict not-bolt</i>	<i>Predict bolt</i>	Precision	0.59
<i>Not-bolt</i>	1471791	6586	Recall	0.70
<i>Bolt</i>	4071	9370	F1 score	0.64
Random forest	<i>Predict not-bolt</i>	<i>Predict bolt</i>	Precision	0.72
<i>Not-bolt</i>	1475540	2837	Recall	0.38
<i>Bolt</i>	6809	6632	F1 score	0.58

379



380

381 *Figure 7: Graphical view of the point cloud classification. The red points are those that the classifier predicts are bolts, the blue*
 382 *points are the classifier predicted as not bolts and the white boxes indicate the actual bolt locations. The left-hand images*
 383 *show the entire classified cloud, whilst the right-hand images show just the points predicted to be bolts.*

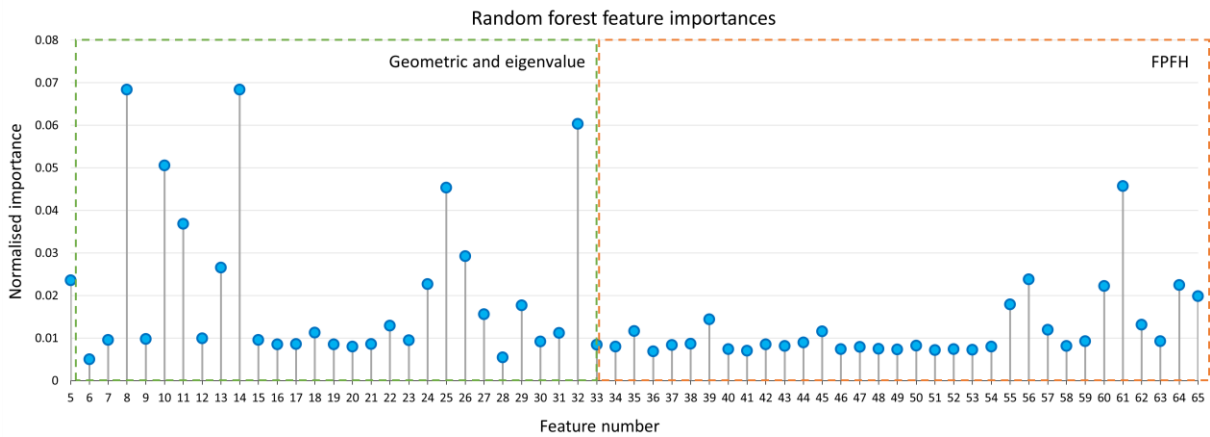
384 As can be seen in Table 3, the results, whilst overall positive still contain many misclassified points. To
 385 investigate whether the DBSCAN clustering can extract individual bolt object locations to a greater
 386 degree of accuracy, the extracted centroids were overlaid with the 101 true bolt centroids and the number
 387 of true positives, false positives and false negatives were counted. For this test, the bolt was classed as
 388 detected if the human generated and machine generated centroids were within the bolt faceplate radius
 389 distance 8cm of each other. These results are given in Table 4.

390 *Table 4: Results of bolt detection algorithm.*

Neural network	<i>Predict Not Bolt</i>	<i>Predict Bolt</i>		
<i>Not Bolt</i>	n/a	6	Precision	0.94
<i>Bolt</i>	13	88	Recall	0.87
			F1	0.90
Random forest	<i>Predict Not Bolt</i>	<i>Predict Bolt</i>		
<i>Not Bolt</i>	n/a	3	Precision	0.95
<i>Bolt</i>	46	55	Recall	0.54
			F1	0.69

391 **5 Discussion**

392 The feature set test shows that the combined 61 feature set is more effective than either the geometric
393 or FPFH based features applied separately. Using only the geometric feature set, the random forest
394 outperforms the neural network; this agrees with the results obtained by Weinmann et al. (2015a) using
395 the same feature types. Using FPFHs the random forest scores relatively poorly, though the combination
396 does still improve on the score recorded from just the geometric feature set. These results infer that the
397 addition of the FPFH features does contribute to the overall accuracy of the random forest, but that they
398 are less important than the geometric features. To examine the feature contributions further, the feature
399 importances were calculated using the Gini importance method. This technique measures how much the
400 Gini impurity is reduced when using a particular feature, averaged across all trees in the forest (Géron,
401 2017). The feature importances are then normalised so that the sum of all importances equals one. Figure
402 8 graphs the feature importances across the classification vector, this shows that the more important
403 features are primarily from the geometric set, though several from the FPFH set also score highly. The
404 highest ranked features (above 0.05) are scattering, absolute height, mean intensity and anisotropy.

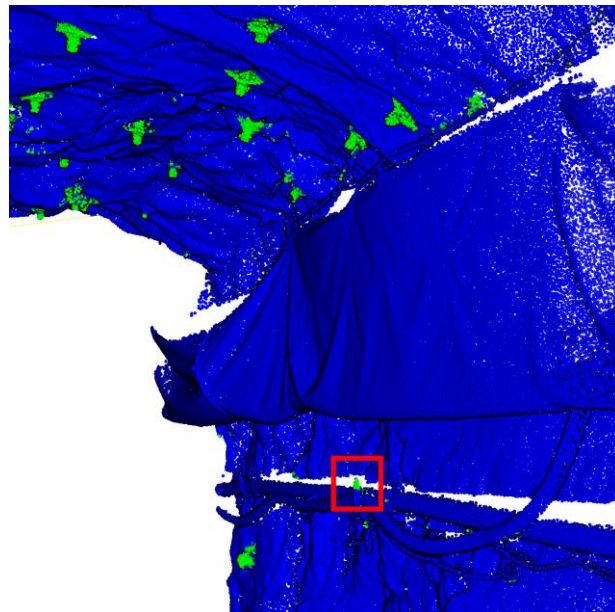


405
406 *Figure 8: Graph showing the individual feature importances for the random forest classifier. The green box indicates the*
407 *geometric and eigenvalue based features and the amber box indicates the FPFH features. For details on feature numbers see*
408 *Table 1.*

409 The neural network classifier cannot output a feature importance ranking; however, from examining the
410 results it appears that the neural network is utilising more of the FPFH set features, as this was the
411 highest non-combined score for all classifier and feature set combinations. The intensity features

412 provided an improvement of 0.02 to both classifiers' scores; these intensity features are some of the
413 simplest to compute and are therefore a strong addition to the feature sets.

414 The point-wise results are positive despite some misclassifications. This is due to the challenging dataset
415 and the many confusion objects. Most importantly, they contain enough positively identified points to
416 enable the DBSCAN algorithm to detect the actual bolt objects. Primarily the incorrectly identified bolt
417 points (false positives) occurred as isolated points, allowing them to be easily removed by the clustering
418 operation. Only rarely, as in the instance of pipe mounting steelwork which closely resembles a bolt,
419 did the algorithm misclassify enough points in close proximity to create a false positive cluster, as seen
420 in Figure 9, where the cluster inside the red box is large enough to make it through the DBSCAN stage.
421 The isolated incorrect points visible on the hanging wall in Figure 9 will all be removed by the DBSCAN
422 process.



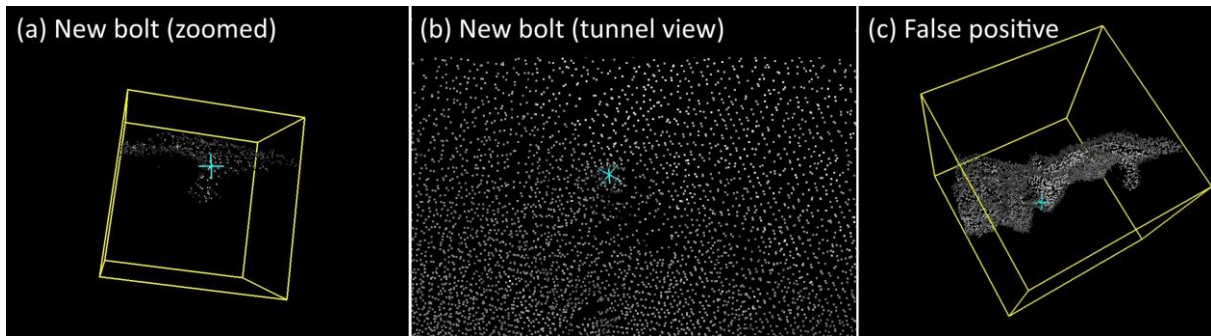
423
424 *Figure 9: Instance of misclassified cluster of points by the random forest classifier. Blue points are predicted not-bolt, green*
425 *points are predicted bolt and the red box indicates a piece of pipe mounting bracket incorrectly classified as a bolt.*

426 At the object extraction stage, as both classifiers had only a few false positives, these were manually
427 checked in the original highest resolution scan data to determine if there was in fact a bolt present at that
428 location which had been missed at the labelling stage. From this examination, it appeared that the neural
429 network correctly identified 5 bolts which were badly scanned and highly obscured that had not been

430 picked up at the dataset creation stage. To verify this result, the test area of the tunnel was physically
431 inspected to determine the ground truth. All bolts visible from the scanner positions were verified,
432 confirming the 96 human detected bolts and the 5 bolts missed by the human labelling exercise. Bolts
433 entirely hidden from the scanner position were not included as these were not detectable from the scan
434 data by a human or an algorithm.

435 This brought the number of true bolts in the test data up to 101. This demonstrates the value of machine
436 learning technologies for automated quality assurance and quality control as in these difficult cases the
437 neural network surpassed the human inspector. The testing dataset was then used to estimate the level
438 of label noise present in the training dataset. The test dataset label noise was ~5% at the cluster/object
439 level (5 missed out of 101 total) and ~3% at the individual point level (471 missed out of 13,912 total).
440 The figures are expected to be far lower for the training dataset as the mislabelled points are all in the
441 ‘not bolt’ class, which has been randomly resampled to contain only 20% of its points. Neural networks
442 and random forests have been shown to be highly robust to label noise below 10% (Folleco et al. 2009,
443 Pelletier et al. 2017), therefore, the small number of mislabelled points in the training dataset is not
444 expected to have has a meaningful impact on the classifier training. Comparing the human result to the
445 neural network, the human is still superior with a precision of 1 and a recall of 0.95; however, in a real
446 world inspection case, the human takes much longer to identify the bolts, suffers from fatigue and still
447 cannot detect every bolt. Figure 10 (a) and (b) show an example a bolt missed by the human operator
448 but found by the neural network and Figure 10 (c) shows an actual incorrect detection by the neural
449 network.

450 The false negatives from the neural network also were examined, and it was found that 10 out of the 11
451 missed detections were low bolts on the sidewall. From this, we can infer that that the Z values and the
452 relatively few sidewall bolts compared to roof bolts in the training data are influencing the model’s
453 decision making. Future work will investigate whether this result can be improved by adding more
454 sidewall training data examples through augmentation or by removing the features related to absolute Z
455 value. Future work would also consider reimplementing the pipeline in C++ using the PCL libraries for
456 increased speed for a production mining environment.



457

458 *Figure 10: Examination of false negatives and false positives. (a) shows the false negative bolt zoomed and extracted to a*
 459 *specific angle and (b) shows how the false negative appears to a human in the full tunnel dataset. (c) shows a sharp*
 460 *discontinuity (false positive) that has been mistaken for a bolt by the classifier.*

461 6 Conclusion

462 This paper describes a methodology to automatically detect supporting rock bolts from laser scan data.
 463 After the scans have been extracted from the instrument, the workflow is implemented entirely with
 464 open source software. Our methodology is customised to the underground environment and improves
 465 upon previously published surface applications by utilising a larger feature set and robust clustering to
 466 address the challenges from noise, confusion objects and multiple Z values present in a typical excavated
 467 tunnel. The neural network classifier produced the strongest point-wise classification results, allowing
 468 the DBSCAN algorithm to successfully locate the candidate bolt objects. Further work will focus on
 469 extending this approach to other mining datasets gathered with different types of 3D laser scanners
 470 including low cost mobile solutions. The bolt location output files could be used for multiple
 471 applications including verifying bolting patterns that have been installed to specification, recording spot
 472 bolting locations for geotechnical reference and for linking the onboard hole information recorded by
 473 bolting machines to a real world coordinate. These applications offer mining companies valuable
 474 opportunities to embrace new technologies for improved productivity and safety in a digitally connected
 475 world.

476 **Funding:** This research was carried out as part of a PhD studentship funded by the University of Exeter.

477 **J. Gallwey:** Methodology, Writing-Original draft preparation, Investigation, Software **M. Eyre:**
478 Conceptualization, Writing- Reviewing and Editing, Supervision. **J. Coggan:** Supervision, Project
479 Administration, Writing- Reviewing and Editing.

480 References

- 481 Agrawal, A., Nakazawa, A., Takemura, H., 2009. MMM-classification of 3D Range Data, in: IEEE
482 International Conference on Robotics and Automation. Kobe, pp. 2003–2008
- 483 Anguelov, D., Taskarf, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A., 2005.
484 Discriminative learning of Markov random fields for segmentation of 3D scan data. *Comput. Vis.*
485 *Pattern Recognition*, 2005. CVPR 2005. IEEE Comput. Soc. Conf. 2, 169–176 vol. 2.
486 <https://doi.org/10.1109/CVPR.2005.133>
- 487 Attard, L., Debono, C.J., Valentino, G., Di Castro, M., 2018. Tunnel inspection using photogrammetric
488 techniques and image processing: A review. *ISPRS J. Photogramm. Remote Sens.* 144, 180–188.
489 <https://doi.org/10.1016/J.ISPRSJPRS.2018.07.010>
- 490 Balado, J., Díaz-Vilariño, L., Arias, P., González-Jorge, H., 2018. Automatic classification of urban
491 ground elements from mobile laser scanning data. *Autom. Constr.* 86, 226–239.
492 <https://doi.org/https://doi.org/10.1016/j.autcon.2017.09.004>
- 493 Bassier, M., Van Genechten, B., Vergauwen, M., 2019. Classification of sensor independent point cloud
494 data of building objects using random forests. *J. Build. Eng.* 21, 468–477.
495 <https://doi.org/10.1016/j.jobe.2018.04.027>
- 496 Behley, J., Kersting, K., Schulz, D., Steinhage, V., Cremers, A.B., 2010. Learning to hash logistic
497 regression for fast 3D scan point classification. *IEEE/RSJ 2010 Int. Conf. Intell. Robot. Syst. IROS*
498 *2010 - Conf. Proc.* 5960–5965. <https://doi.org/10.1109/IROS.2010.5650093>
- 499 Blomley, R., Jutzi, B., Weinmann, M., 2016. Classification of Airborne Laser Scanning Data Using
500 Geometric Multi-Scale Features and Different Neighbourhood Types. *ISPRS Ann. Photogramm.*
501 *Remote Sens. Spat. Inf. Sci.* III–3, 169–176. <https://doi.org/10.5194/isprsannals-III-3-169-2016>

502 Body, D., 2014. The Changing Face of Surveying within the Mining Industry [WWW Document].
503 URL <https://www.bentley.com/-/media/A633A2E0F9EB48D190DE94E3F1529CFF.ashx>
504 (accessed 10.03.20)

505 Breiman, L., 2001. Random Forests. *Mach. Learn.* 45, 5–32. <https://doi.org/10.1023/A:1010933404324>

506 Chehata, N., Guo, L., Mallet, C., 2009. Airborne LIDAR feature selection for urban classification using
507 random forests. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* 38.

508 Chen, S., Walske, M.L., Davies, I.J., 2018. Rapid mapping and analysing rock mass discontinuities with
509 3D terrestrial laser scanning in the underground excavation. *Int. J. Rock Mech. Min. Sci.* 110, 28–
510 35. <https://doi.org/https://doi.org/10.1016/j.ijrmmms.2018.07.012>

511 Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters
512 a Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, in:
513 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining,
514 KDD'96. AAAI Press, pp. 226–231

515 Eyre, M., Wetherelt, A., Coggan, J., 2016. Evaluation of automated underground mapping solutions for
516 mining and civil engineering applications. *J. Appl. Remote Sens.* 10, 1–18.
517 <https://doi.org/10.1117/1.JRS.10.046011>

518 Folleco, A.A., Khoshgoftaar, T.M., Van Hulse, J., Napolitano, A., 2009. Identifying learners robust to
519 low quality data. *Inform.* 33, 245–259.

520 Ganić, A., Milutinović, A., Tokalić, R., Ognjanović, S., 2011. Measuring methods for cross sections of
521 underground mine chambers. *Podzemn. Rad.* 0

522 Géron, A., 2017. Hands-On Machine Learning with Scikit-Learn & Tensorflow, 1st ed. O'Reilly,
523 Sebastopol.

524 Gikas, V., 2012. Three-dimensional laser scanning for geometry documentation and construction
525 management of highway tunnels during excavation. *Sensors (Switzerland)* 12, 11249–11270.
526 <https://doi.org/10.3390/s120811249>

527 Girardeau-Montaut, D., 2016. Cloud compare documentation [WWW Document]. URL
528 http://www.cloudcompare.org/doc/wiki/index.php?title=Main_Page (accessed 11.02.20)

529 Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Cambridge MA.

530 Hackel, T., Wegner, J.D., Schindler, K., 2017. Joint classification and contour extraction of large 3D
531 point clouds. ISPRS J. Photogramm. Remote Sens. 130, 231–245.
532 <https://doi.org/10.1016/j.isprsjprs.2017.05.012>

533 Hecht-Nielsen, R., 1992. III.3 - Theory of the Backpropagation Neural Network**Based on “nonindent”
534 by Robert Hecht-Nielsen, which appeared in Proceedings of the International Joint Conference on
535 Neural Networks 1, 593–611, June 1989. © 1989 IEEE., in: Wechsler, H. (Ed.), Neural Networks
536 for Perception. Academic Press, pp. 65–93. <https://doi.org/https://doi.org/10.1016/B978-0-12-741252-8.50010-8>
537

538 Hoek, E., Brown, E., 1980. Underground Excavations in Rock. Institution of Mining and Metallurgy,
539 London ISBN 0900488549

540 Huang, H., Li, Q., Zhang, D., 2018. Deep learning based image recognition for crack and leakage defects
541 of metro shield tunnel. Tunn. Undergr. Sp. Technol. 77, 166–176.
542 <https://doi.org/10.1016/j.tust.2018.04.002>

543 Jutzi, B., Gross, H., 2009. Nearest neighbour classification on laser point clouds to gain object structures
544 from buildings. ISPRS Hann. Work. 2009 High-Resolution Earth Imaging Geospatial Inf.
545 XXXVIII, 6

546 Kogut, T., Weistock, M., 2019. Classifying airborne bathymetry data using the Random Forest
547 algorithm. Remote Sens. Lett. 10, 874–882. <https://doi.org/10.1080/2150704X.2019.1629710>

548 Landrieu, L., Raguét, H., Vallet, B., Mallet, C., Weinmann, M., 2017. A structured regularization
549 framework for spatially smoothing semantic labelings of 3D point clouds. ISPRS J. Photogramm.
550 Remote Sens. 132, 102–118. <https://doi.org/10.1016/j.isprsjprs.2017.08.010>

551 LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

552 <https://doi.org/10.1038/nature14539>

553 Lehtomäki, M., Jaakkola, A., Hyypä, J., Kukko, A., Kaartinen, H., 2010. Detection of vertical pole-
554 like objects in a road environment using vehicle-based laser scanning data. *Remote Sens.* 2, 641–
555 664. <https://doi.org/10.3390/rs2030641>

556 Lehtomäki, M., Jaakkola, A., Hyypä, J., Lampinen, J., Kaartinen, H., Kukko, A., Puttonen, E., Hyypä,
557 H., 2016. Object Classification and Recognition From Mobile Laser Scanning Point Clouds in a
558 Road Environment. *IEEE Trans. Geosci. Remote Sens.* 54, 1226–1239.
559 <https://doi.org/10.1109/TGRS.2015.2476502>

560 Li, C.C., 2017. Principles of rockbolting design. *J. Rock Mech. Geotech. Eng.* 9, 396–414.
561 <https://doi.org/10.1016/j.jrmge.2017.04.002>

562 Long, N.Q., Buczek, M.M., Hien, L.P., Szlapińska, S.A., Nam, B.X., Nghia, N.V., Cuong, C.X., 2018.
563 Accuracy assessment of mine walls' surface models derived from terrestrial laser scanning. *Int. J.*
564 *Coal Sci. Technol.* 5, 328–338. <https://doi.org/10.1007/s40789-018-0218-1>

565 Martínez-Sánchez, J., Puente, I., González-Jorge, H., Riveiro, B., Arias, P., 2016. Automatic thickness
566 and volume estimation of sprayed concrete on anchored retaining walls from terrestrial LIDAR
567 data. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.* 41, 521–526.
568 <https://doi.org/10.5194/isprsarchives-XLI-B5-521-2016>

569 Maturana, D., Scherer, S., 2015. VoxNet: A 3D convolutional neural network for real-time object
570 recognition. *IEEE/RSJ Int. Conf. Intell. Robot. Syst.* 922–928.
571 <https://doi.org/10.1109/IROS.2015.7353481>

572 Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building
573 object detection in urban areas. *ISPRS J. Photogramm. Remote Sens.* 87, 152–165.
574 <https://doi.org/10.1016/j.isprsjprs.2013.11.001>

575 Nygren, P., Jasinski, M., 2016. A Comparative Study of Segmentation and Classification Methods for
576 3D Point Clouds. University of Gothenburg.

577 Öberg, F., 2013. Method and system for monitoring and documenting installation of rock reinforcement
578 bolt. US 8,606,542 B2

579 Pal, M., 2005. Random forest classifier for remote sensing classification. *Int. J. Remote Sens.* 26, 217–
580 222. <https://doi.org/10.1080/01431160412331269698>

581 Pauly, M., Keiser, R., Gross, M., 2003. Multi-scale Feature Extraction on Point-Sampled Surfaces.
582 *Comput. Graph. Forum* 22, 281–289. <https://doi.org/10.1111/1467-8659.00675>

583 Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M.,
584 Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M.,
585 Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*
586 12, 2825–2830

587 Pelletier, C., Valero, S., Inglada, J., Champion, N., Sicre, C.M., Dedieu, G., 2017. Effect of training
588 class label noise on classification performances for land cover mapping with satellite image time
589 series. *Remote Sens.* 9. <https://doi.org/10.3390/rs9020173>

590 Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2016. PointNet: Deep Learning on Point Sets for 3D Classification
591 and Segmentation, in: 4th International Conference on 3D Vision, 3DV 2016. pp. 601–610.
592 <https://doi.org/10.1109/3DV.2016.68>

593 Rau, J.Y., Jhan, J.P., Hsu, Y.C., 2015. Analysis of oblique aerial images for land cover and point cloud
594 classification in an Urban environment. *IEEE Trans. Geosci. Remote Sens.* 53, 1304–1319.
595 <https://doi.org/10.1109/TGRS.2014.2337658>

596 Riegler, G., Ulusoy, A.O., Geiger, A., 2017. OctNet: Learning deep 3D representations at high
597 resolutions. *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017 2017-Janua*,
598 6620–6629. <https://doi.org/10.1109/CVPR.2017.701>

599 Rusu, R.B., 2010. Semantic 3D Object Maps for Everyday Manipulation in Human Living
600 Environments. *Künstliche Intelligenz* 24, 345–348. <https://doi.org/10.1007/s13218-010-0059-6>

601 Rusu, R.B., Blodow, N., Beetz, M., 2009. Fast Point Feature Histograms (FPFH) for 3D registration.

602 2009 IEEE Int. Conf. Robot. Autom. 3212–3217. <https://doi.org/10.1109/ROBOT.2009.5152473>

603 Rusu, R.B., Cousins, S., 2011. 3D is here: Point Cloud Library (PCL), in: 2011 IEEE International
604 Conference on Robotics and Automation. pp. 1–4. <https://doi.org/10.1109/ICRA.2011.5980567>

605 Schach, R., Garshol, K., Heltzen, A.M., 1979. Rock Bolting: A Practical Handbook, Pergamon Press,
606 Oxford, ISBN 0080225039

607 Schubert, E., Sander, J., Ester, M., Kriegel, H.P., Xu, X., 2017. DBSCAN Revisited, Revisited: Why
608 and How You Should (Still) Use DBSCAN. *ACM Trans. Database Syst.* 42.
609 <https://doi.org/10.1145/3068335>

610 Soilán, M., Riveiro, B., Martínez-Sánchez, J., Arias, P., 2017. Segmentation and classification of road
611 markings using MLS data. *ISPRS J. Photogramm. Remote Sens.* 123, 94–103.
612 <https://doi.org/10.1016/j.isprsjprs.2016.11.011>

613 Soilán, M., Sánchez-Rodríguez, A., Del Río-Barral, P., Perez-Collazo, C., Arias, P., Riveiro, B., 2019.
614 Review of laser scanning technologies and their applications for road and railway infrastructure
615 monitoring. *Infrastructures* 4. <https://doi.org/10.3390/infrastructures4040058>

616 Strobl, C., Boulesteix, A.-L., Kneib, T., Augustin, T., Zeileis, A., 2008. Conditional variable importance
617 for random forests. *BMC Bioinformatics* 9, 307. <https://doi.org/10.1186/1471-2105-9-307>

618 Tan, K., Cheng, X., Ju, Q., Wu, S., 2016. Correction of Mobile TLS Intensity Data for Water Leakage
619 Spots Detection in Metro Tunnels. *IEEE Geosci. Remote Sens. Lett.* 13, 1711–1715.
620 <https://doi.org/10.1109/LGRS.2016.2605158>

621 Triebel, R., Kersting, K., Burgard, W., 2006. Robust 3D scan point classification using associative
622 Markov networks. *Proc. - IEEE Int. Conf. Robot. Autom.* 2006, 2603–2608.
623 <https://doi.org/10.1109/ROBOT.2006.1642094>

624 van der Merwe, J.W., Andersen, D.C., 2013. Applications and benefits of 3D laser scanning for the
625 mining industry . *J. South. African Inst. Min. Metall.*

626 Vandapel, N., Huber, D., Kapuria, A., Hebert, M., 2004. Natural terrain classification using 3-d lidar

627 data. IEEE Int. Conf. Robot. Autom. 2004. Proceedings. ICRA '04. 2004 5117–5122.
628 <https://doi.org/10.1109/ROBOT.2004.1302529>

629 Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015a. Semantic point cloud interpretation based on
630 optimal neighborhoods, relevant features and efficient classifiers. ISPRS J. Photogramm. Remote
631 Sens. 105, 286–304. <https://doi.org/10.1016/j.isprsjprs.2015.01.016>

632 Weinmann, M., Urban, S., Hinz, S., Jutzi, B., Mallet, C., 2015b. Distinctive 2D and 3D features for
633 automated large-scale scene analysis in urban areas. Comput. Graph. 49, 47–57.
634 <https://doi.org/10.1016/j.cag.2015.01.006>

635 Weinmann, M., 2016. Reconstruction and analysis of 3D scenes: From irregularly distributed 3D points
636 to object classes. Reconstr. Anal. 3D Scenes From Irregularly Distrib. 3D Points to Object Classes
637 1–233. <https://doi.org/10.1007/978-3-319-29246-5>

638 Weinmann, Martin, Weinmann, Michael, Mallet, C., Brédif, M., 2017. A classification-segmentation
639 framework for the detection of individual trees in dense MMS point cloud data acquired in urban
640 areas. Remote Sens. 9. <https://doi.org/10.3390/rs903277>

641 Wold, S., Esbensen, K., Geladi, P., 1987. Principal component analysis. Chemom. Intell. Lab. Syst. 2,
642 37–52. [https://doi.org/https://doi.org/10.1016/0169-7439\(87\)80084-9](https://doi.org/https://doi.org/10.1016/0169-7439(87)80084-9)

643 Xu, T., Xu, L., Li, X., Yao, J., 2018. Detection of Water Leakage in Underground Tunnels Using
644 Corrected Intensity Data and 3D Point Cloud of Terrestrial Laser Scanning. IEEE Access 6,
645 32471–32480. <https://doi.org/10.1109/ACCESS.2018.2842797>

646 Yang, B., Fang, L., Li, J., 2013. Semi-automated extraction and delineation of 3D roads of street scene
647 from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 79, 80–93.
648 <https://doi.org/10.1016/j.isprsjprs.2013.01.016>

649 Zhao, S., Zhang, D.M., Huang, H.W., 2020. Deep learning–based image instance segmentation for
650 moisture marks of shield tunnel lining. Tunn. Undergr. Sp. Technol.
651 <https://doi.org/10.1016/j.tust.2019.103156>