JGR: Solid Earth

Supporting Information for

An object-based approach to differentiate pores and microfractures in petrographic analysis using explainable, supervised machine learning

Issac Sujay Anand John Jayachandran[1,2], Juan Carlos Laya[1], Holly Catherine Gibbs[3,4], Yemna Qaiser[2], Talha Khan[2], Mohammed Ishaq Mohammed Shoeb Ansari[5], Mohammed Yaqoob Ansari[5], Mohammed Malyah[2], Nayef Alyafei[2], Thomas Daniel Seers[2]

[1] Department of Geology & Geophysics, Texas A&M University, College Station, TX 77843, USA

[2] Department of Petroleum Engineering, Texas A&M University Qatar, Education City, Doha, Qatar

[3] Department of Biomedical Engineering, Texas A&M University, College Station, TX 77843, USA

[4] Microscopy and Imaging Center, Texas A&M University, College Station, TX 77843, USA

[5] Department of Electrical & Computer Engineering, Texas A&M University Qatar, Education City, Doha, Qatar

Contents of this file

Introduction

We expand on the methodological decisions made for our study. We also provide brief descriptions on the conceptual underpinnings of the supervised models used for our study. In addition, we include the results of the unsupervised clustering algorithms we tested (K-means and DBSCAN) as our study was limited to supervised models in scope. Univariate analytics on the simple shape features per secondary pore types is provided as well.

Text S1.

Pre-processing of the Images

Denoising

The kernel size of the non-local means filter was automatically estimated using the approach of Immerkaer (1996), with the smoothness factor maintained at a value of one for all images to limit over-smoothing of edges.

Sharpening

A standard unsharp mask radius of 1 with a mask value of 0.7 was applied for all images, using the built-in unsharp mask filter within Fiji. The pre-processed images after both denoising and sharpening are included in the dataset attached to this study.

Segmentation

To increase the connectivity of the microfractures, a more aggressive form of segmentation was pursued. The segmentation protocol implemented was performed in two phases. The first phase was manual thresholding of the blue epoxy impregnated pixels in the HSB (Hue, Saturation, Brightness) color space. The blue hue corresponding to the epoxy was delineated within 120 to 180, with the saturation unaltered. The lower threshold for brightness was decreased to accommodate all the blue epoxy, with values ranging from 30 to 255. The quality of the segmentation was evaluated visually in real time and parameters tuned accordingly. The second phase involved thresholding in the CIELAB color space. CIELAB color space is a device-independent method to objectively classify colors, where L stands for lightness, A for the continuum from red to green, and B for the continuum from blue to yellow (Mlynarczuk et al., 2013). Since the microfractures were filled with blue epoxy, the B channel was especially sensitive. The image was first converted from RGB to LAB color space using a Fiji built-in tool. The B channel was extracted and a simple contrast enhancement was needed to binarize the image. Both segmentation steps were performed independently. The post-processing pipeline was conducted on both the HSB and LAB binary masks in parallel, as shown in Fig. 2a.

Combining the Processed HSB and CIELAB Binary Masks

The post-processed binary masks from both color spaces were then added together using the Image Calculator tool in Fiji. The combination of both segmentations did not offer a significant boost in terms of pore connectivity as in both cases the segmentation results were similar, with the HSB binary mask offering visibly better results. Instead, the greatest effect was observed in microfractures as several individual microfractures which were disjointed from HSB thresholding displayed improved continuity in the composite image (Fig. S2). The microporous matrix zones and microporous grains were segmented as macropores as a byproduct of the aggressive segmentation. Additionally, the large quantities of these zones rendered manual masking impracticable. For this study, they

were approximated as pores, which is not entirely unreasonable for grain molds and small patches of blue haze in terms of shape. However, larger microporous patches are among the major artifacts present in the data. Moreover, the thin sections used herein were not purposed for digital image analysis and as such contain damage of different forms such as pen markings and microsampling scratches amongst others. However, due to their limited quantities these scene artifacts were removed using manual masking.

Feature Extraction

Labelling of the binary masks was performed in Python using the 'Connected components' function with 8-connect from the OpenCV library. The 'regionprops' function from the sci-kit image library was used to extract size and shape features of each object (Table 1). Two associated features metrics that require special mention are the major and minor axes of the best-fitting ellipse. These axes were fit using the normalized second central moments of the object, which is a region-based approach. Region-based approaches are generally more robust than contour-based approaches as the area of the object is less sensitive to noise (Mulchrone and Choudhury, 2004; Neal and Russ, 2012). The 'regionprops' features are mostly related to object size. This required supplementing with shape features engineered from these size metrics. Feature engineering was performed in the R programming language based on derivations laid out in Neal and Russ (2012) and Weger (2006). Engineered features were selected based on their popularity in the geological community (Anselmetti et al., 1998; Weger, 2006; Weger et al., 2009; Norbisrath et al., 2015; Abedini et al., 2018; Borazjani et al., 2016; Ghiasi-Freez et al., 2012; Mollajan et al., 2016; Sharifi et al., 2022; Wang et al., 2022).

Clustering Algorithms

The objective of clustering algorithms is to group similar datapoints into discrete clusters. Two independent clustering algorithms: k-means and DBSCAN, were utilized to check for the presence of natural clusters in the feature space, ideally corresponding to microfractures and pores. The clustering algorithms were applied directly on the data. Hierarchical clustering was ignored for this study on conceptual and practical grounds. Conceptually, the objects do not necessarily follow a hierarchy, so this form of clustering is not appropriate. From a practical perspective, hierarchical clustering is also computationally expensive for large datasets such as the one in this study.

K-means Clustering

K-means was chosen as it is one of the most widely used clustering algorithms (James et al., 2021). As K-means is distance-based, it uses distance mapping to measure the distances between each of the 'n' datapoints to each other within the feature space. It then attempts to minimize the total inter-cluster distance of all clusters (James et al., 2021). The number of clusters K = 2 was selected since this study is a binary classification problem. K-Means clustering was implemented using the 'kmeans' function from the 'stats' library with euclidean distance mapping. The resultant clustering was visualized via PCA, as the feature space exceeded three dimensions, with the boundary of the clusters defined via their convex hull (Fig. S4a). It can be observed in Fig. S4a that the k-means

algorithm essentially bisected the point cloud, which typically indicates a lack of natural clusters (Thrun, 2018; Thrun, 2021).

DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) was first proposed by Ester et al. (1996). The most significant advantages of this method compared to K-means are that it is not necessary to predefine the number of clusters and is significantly more robust to the presence of outliers (Schubert et al., 2017). DBSCAN attempts to classify the densest clusters with lower density collections of points potentially classed as outliers. This method contains two definable parameters: the cluster radius (epsilon), and the minimum number of points each cluster should contain to be considered a viable group (cluster density). Any point with the number of neighbors greater than the minimum points is considered a core point. Any point that does not have the threshold minimum points but is part of another core point neighborhood is designated a border point. If a point is neither core nor border, it is considered an outlier. For this study, the epsilon value ($\varepsilon = 0.4$) was determined by identifying the elbow of a 5-NN plot. The DBSCAN results in Fig. S4b show a single dense cluster with a few outliers, which supports the lack of natural clusters within the feature space.

Supervised ML

This section furnishes details on the conceptual underpinnings of each of the supervised ML models used in this study. Further details on each model can be found in Kuhn (2013) and James et al. (2021). The tested models can be broadly classified into two categories: linear and non-linear. Linear models generate linear decision boundaries in high-dimensional feature space, whereas non-linear models create non-linear decision boundaries, such as polynomial, radial, and more complex non-parametric curves.

Linear Models

Multiple Logistic regression (MLR) is designed for binary classification problems (Kuhn, 2013; James et al., 2021), where 'multiple' refers to the features used to train the model. Is based upon the concept of the logistic function, where the probability of classifying datapoints into the two classes resembles an S-shaped curve from 0 to 1. The logistic function is fit using maximum likelihood estimation. A major benefit of MLR is that it does not contain any tuning parameters, as the maximum likelihood estimate of the logistic function will provide the best possible model.

Linear Discriminant Analysis (LDA) and MLR only differ in their fitting procedure: whilst MLR uses maximum likelihood estimation for finding the best fitting model, LDA utilizes the Bayes' theorem. LDA assumes that the datapoints of each class belong to a Gaussian distribution, with all classes sharing a common covariance matrix. It is important to note that MLR does not require that the datapoints be drawn from multivariate Gaussian distributions and can potentially outperform LDA if the assumptions are unmet (James et al., 2021).

Non-Linear Models

Quadratic Discriminant Analysis (QDA) is similar to LDA in that it assumes the datapoints have been drawn from multivariate Gaussian distributions with the exception that each of the classes is considered to have its own covariance matrix. This difference results in a quadratic decision boundary. The greater flexibility in shape means that QDA has lower bias compared to LDA, although this typically comes at a cost of higher variance (James et al., 2021). Like MLR and LDA, QDA does not possess any tuning parameters.

Naive-Bayes, just like LDA, and QDA, is part of a family of models based on the Bayes' theorem. The 'Naive' refers to the classifier's assumption that each of the input features are uncorrelated to each other, which in this study and most other cases, is a flawed assumption. The Naive-Bayes implementation used in this study has three tunable parameters: namely the Laplace correction, distribution type, and the bandwidth adjustment.

K-nearest neighbors (KNN) is one of the simplest models commonly deployed for classification (Murphy, 2022). KNN classifies a datapoint as belonging to a certain class based on the classes of the datapoints closest to it. Thus, it does not depend on the underlying distributions of the classes, and is therefore non-parametric (James et al., 2021). The only tuning parameter for KNN is the number of neighbours (K) to each datapoint. The number of neighbors has to be odd to ensure a tiebreaker in the case of binary classification. Choosing the optimum K is non-trivial. If the number of neighbors chosen is too low, then there is a greater chance of. Conversely, if the number of neighbors is too high then decision boundaries are too general, potentially leading to underfitting.

Random Forest (RF) is an ensemble method that is based on aggregating the votes of several decision trees (Breiman, 2001; Kuhn, 2013). For each split of a decision tree, RF only allows a subset of the features to be selected. This restriction ensures that features that strongly influence the datapoints will not be preferred as several trees will not have the option to select it. Essentially, RF decorrelates the trees and therefore makes the results more reliable (Kuhn, 2013). The implementation of RF chosen only had one tunable parameter: the number of randomly selected predictors available for each tree split.

Support Vector Machines (SVM) were first proposed by (Cortes and Vapnik, 1995). This family of classifiers are the most complex models tested in this study. SVMs have two notable features: firstly, they are inherently binary classifiers, and secondly, they create linear hyperplanes (Murphy, 2022; Kuhn, 2013). SVM initiates by identifying datapoints of opposing classes proximal to one another. It then attempts to find the hyperplane that is equidistant from both sets of points, known as the maximum margin hyperplane. The opposing datapoints used to create this hyperplane are referred to as the support vectors. By this definition, SVMs are linear classifiers, but have been adapted

to create non-linear decision boundaries. Three SVM models are used in this study; linear SVM, SVM with radial basis function, and SVM with polynomial kernel. The linear SVM only has one tunable parameter: the cost. The radial SVM has two parameters: sigma and cost. The polynomial SVM has three parameters: the degree, scale, and cost.

Hyperparameter Optimization

For models which did not have any tunable hyperparameters, such as MLR, LDA, and QDA, the training was conducted using 10-fold cross-validation repeated 10 times with accuracy used as the metric. For models which contained tunable hyperparameters, a grid search technique was employed for each of the hyperparameters, with 10-fold cross-validation repeated 10 times applied to each set of hyperparameters. Hyperparameters for each model (if present), and the selected values are presented in Table S2. The hyperparameter optimization curves for each of the models are shown in Fig. S5.

Feature Importance using Shap Values

Shapley values were used to test the hypothesis regarding the importance of aspect ratio with respect to the other shape features in supervised ML models. Shapley values were first introduced by Lloyd Shapley in 1953 (Shapley, 1953) to fairly distribute winnings between players based on their contribution to the game. The two pillars of Shapley values are additivity, where the sum of the winnings of each player must equal the total winnings, and fairness, where the highest performers cannot receive a lower share than the lowest performers. A concise explanation of the mechanism is as follows; in a scenario containing 4 players, in order to identify the importance of Player 1, all possible subsets of the players are made with and without Player 1. In the subset containing Player 1, the amount Player 1 receives is calculated, and in the subset without Player 1, the other players share Player 1's winnings. The difference between the amounts of both subsets gives the marginal contribution of Player 1, and therefore the overall importance of Player 1. Shapley values were theoretically proven as the fairest possible manner to distribute winnings. Lundberg et al. (2017) appropriated this concept from cooperative game theory into artificial intelligence (AI) to impute the importance of input features within black-box models (a field now known as 'Explainable AI'). To differentiate from its usage in game theory, the authors coined the term Shap values. Some models such as MLR and Random Forest have built-in variable importance measures. For MLR it is the magnitude of the coefficient, whereas Random Forest computes variable importance from the mean decrease in Gini impurity at each split of the decision trees, as well as the mean decrease in overall out-of-bag accuracy. However, most models do not provide this information and are effectively black boxes. Shap values are advantageous in that they are model-agnostic and retroactive with respect to the model building process, offering an external check used to explain the feature contributions to the predictions. It is important to note that Shap values calculate the local importance of features, which is the importance of a particular feature to a specific subset of datapoints. An aggregation is performed to provide the global importance of each feature with regards to the entire dataset.

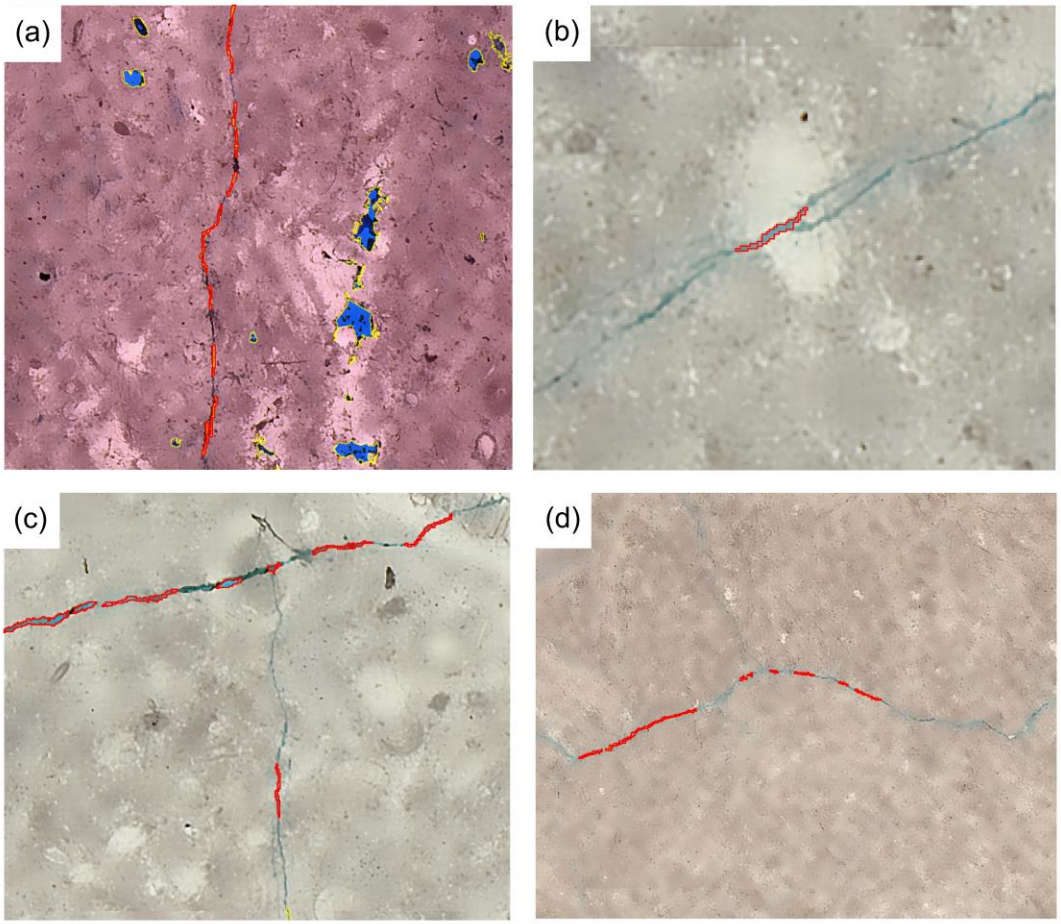

Figure S1. Fragmentation of microfractures from the thin section images. The red outlines indicate the segmented portions of the microfracture.

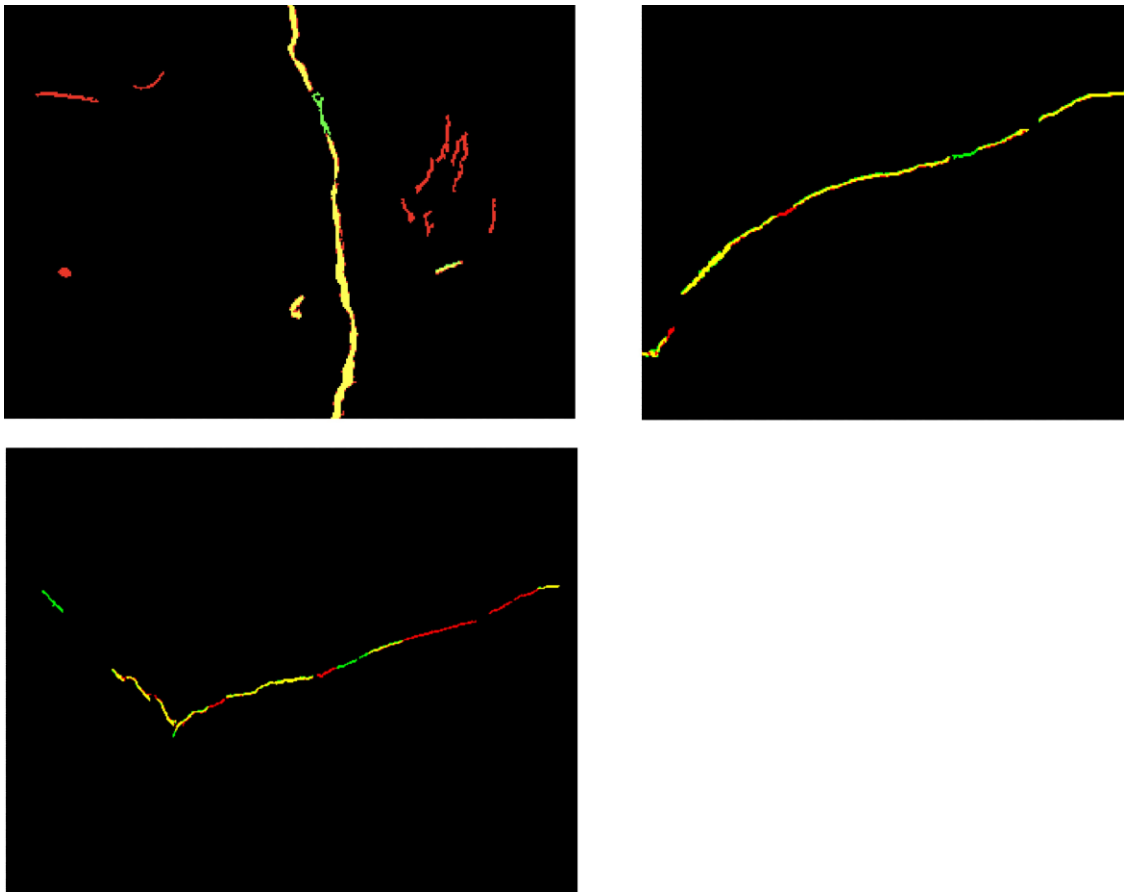

Figure S2. Composites of the HSB and LAB binary masks. Red signifies the HSB binary mask, green is the LAB binary mask, and yellow is the union of both masks. The HSB mask displays a stronger segmentation overall, but the LAB mask provides a notable boost in connectivity.

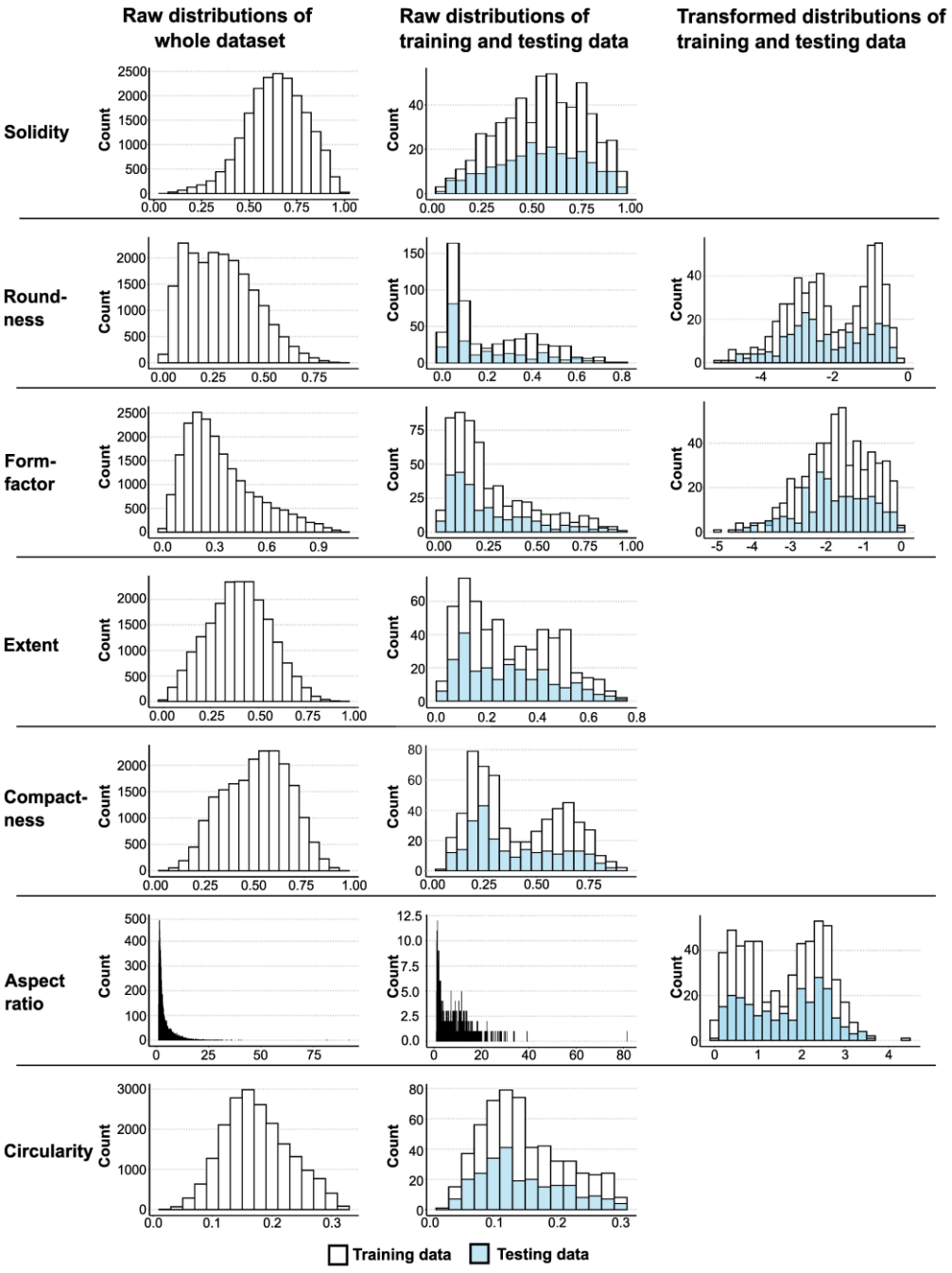

Figure S3. Univariate distributions of the shape features for the raw global dataset, the raw labelled dataset, and the transformed labelled dataset.

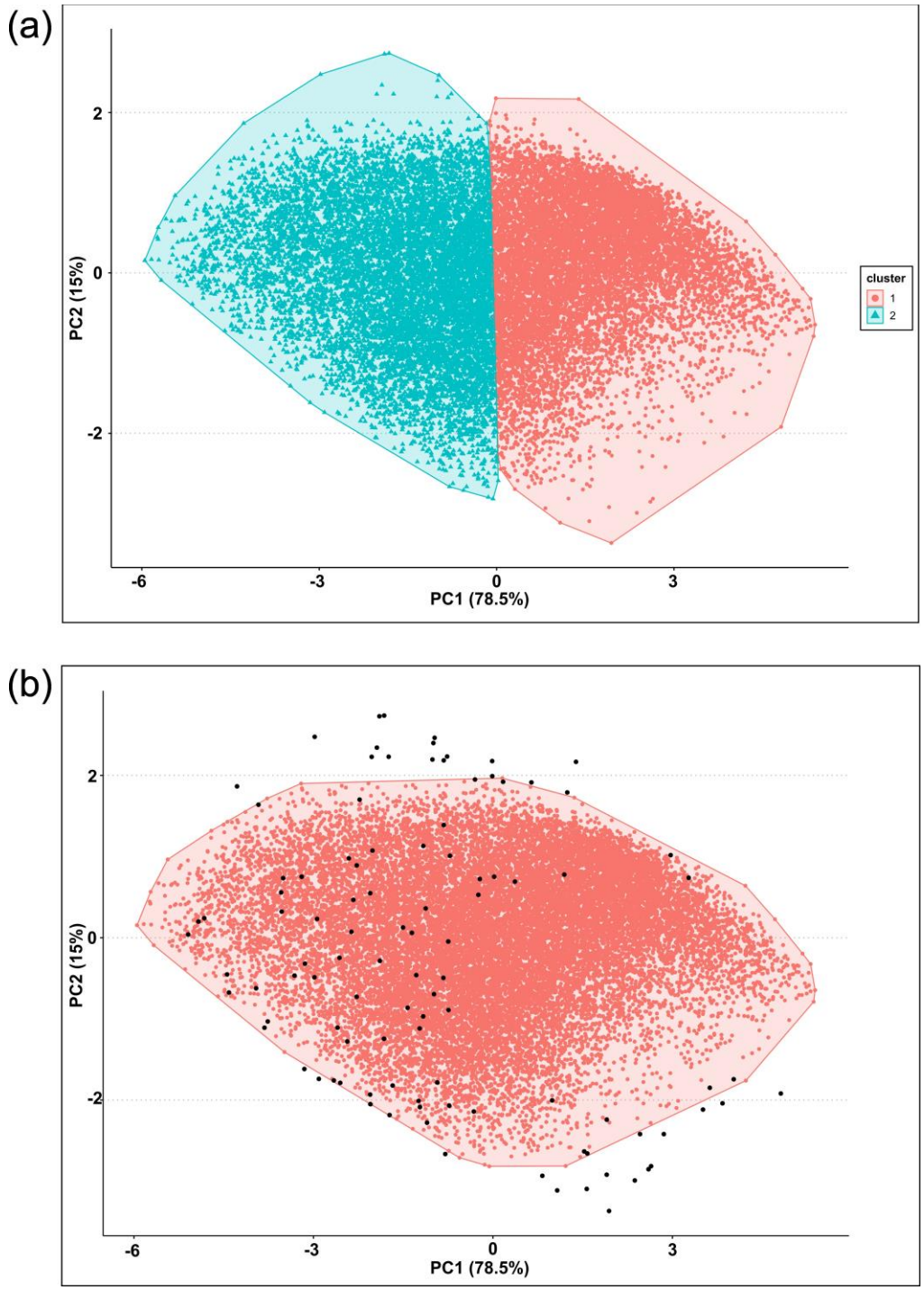

Figure S4. (a) Result of k-means on the global dataset. (b) Result of DBSCAN on the global dataset.

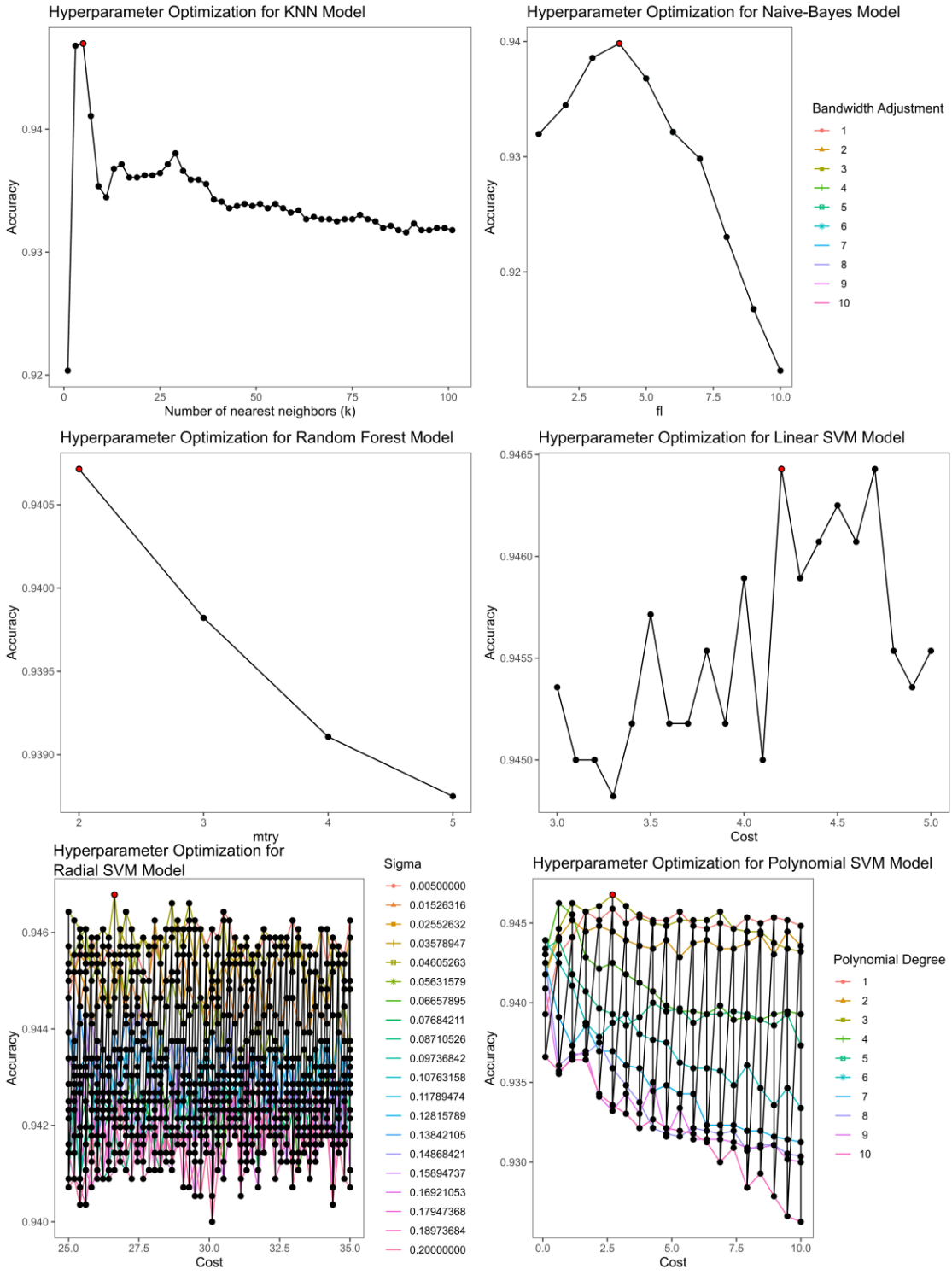

Figure S5. Hyperparameter optimization visualizations for the supervised ML models. MLR, LDA, and QDA did not contain any tunable hyperparameters and hence not included.

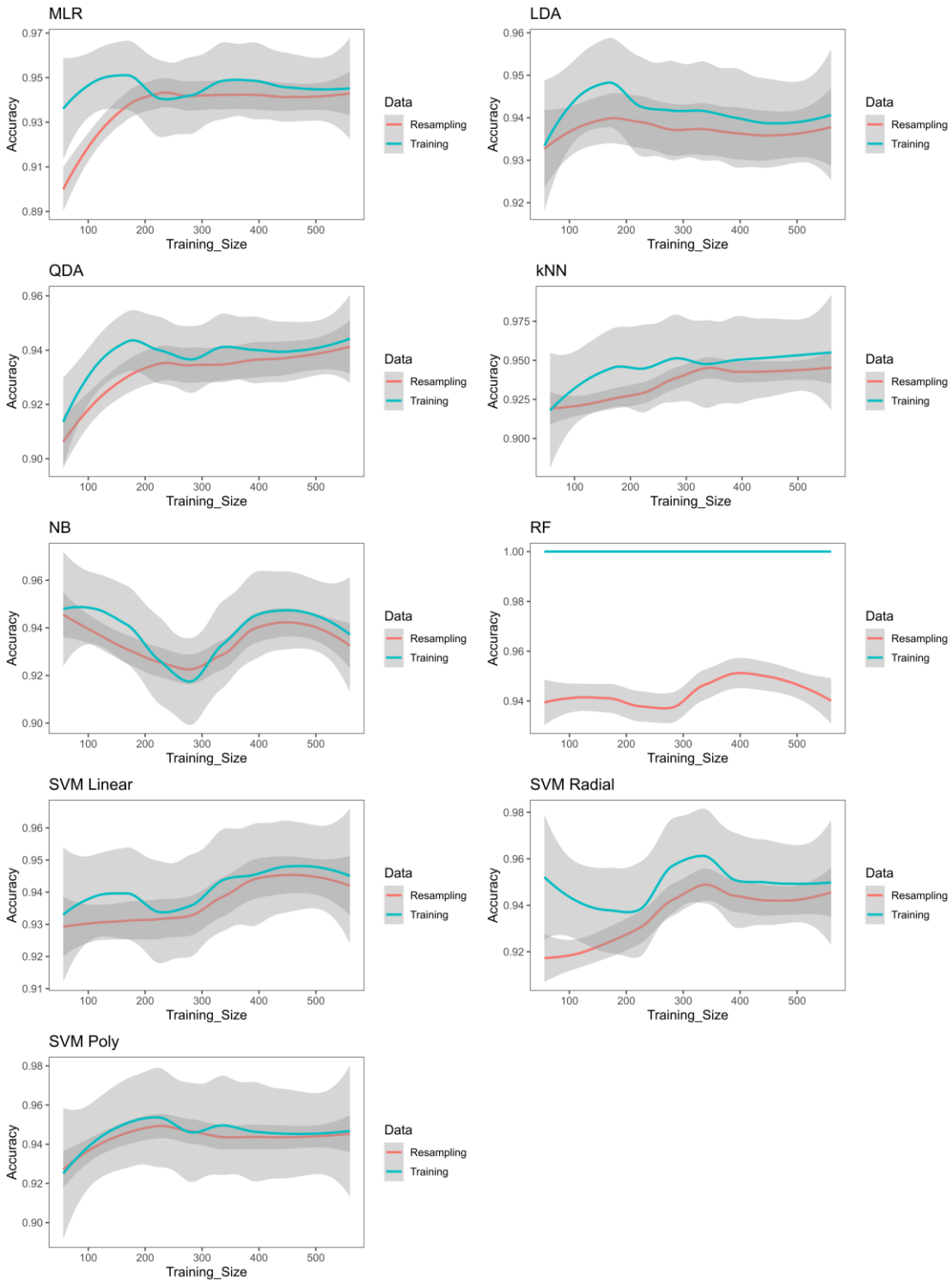

Figure S6. Learning curves for all the supervised models. Random forest was the only model which showed overfitting as the training accuracy was constantly 100% with the resampling accuracy significantly lower.

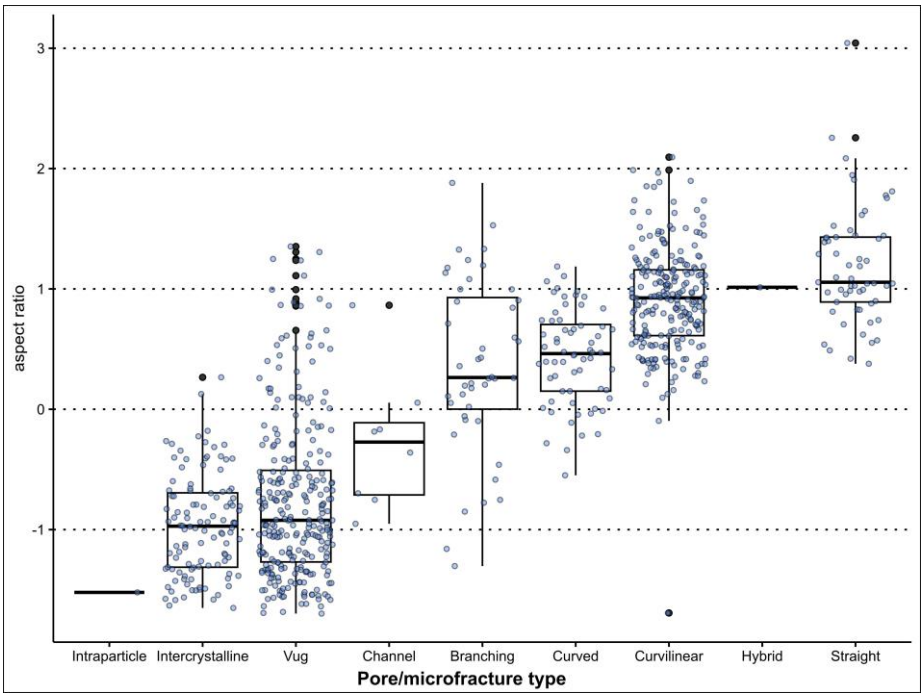

Figure S7a. Boxplot of aspect ratio ranges for the secondary labels of pore and microfracture types.

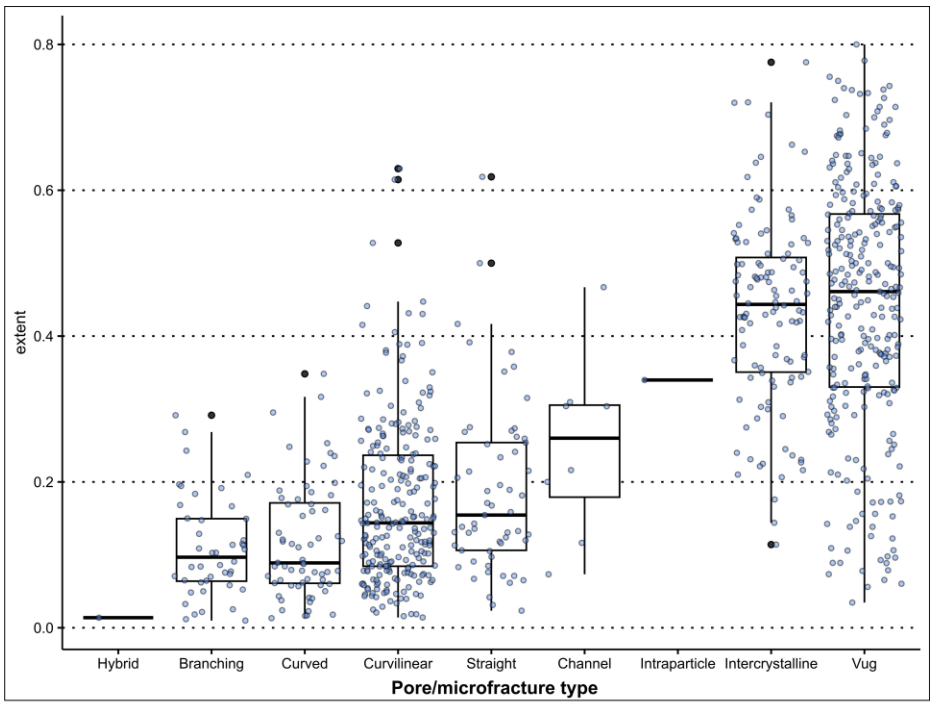

Figure S7b. Boxplot of extent ranges for the secondary labels of pore and microfracture types.

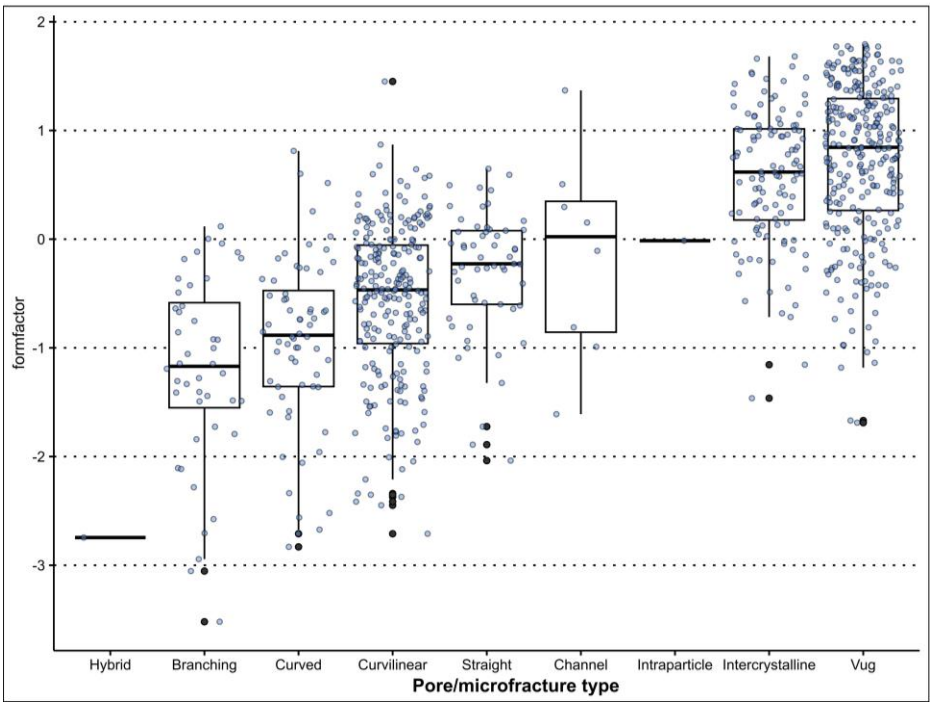

Figure S7c. Boxplot of formfactor ranges for the secondary labels of pore and microfracture types.

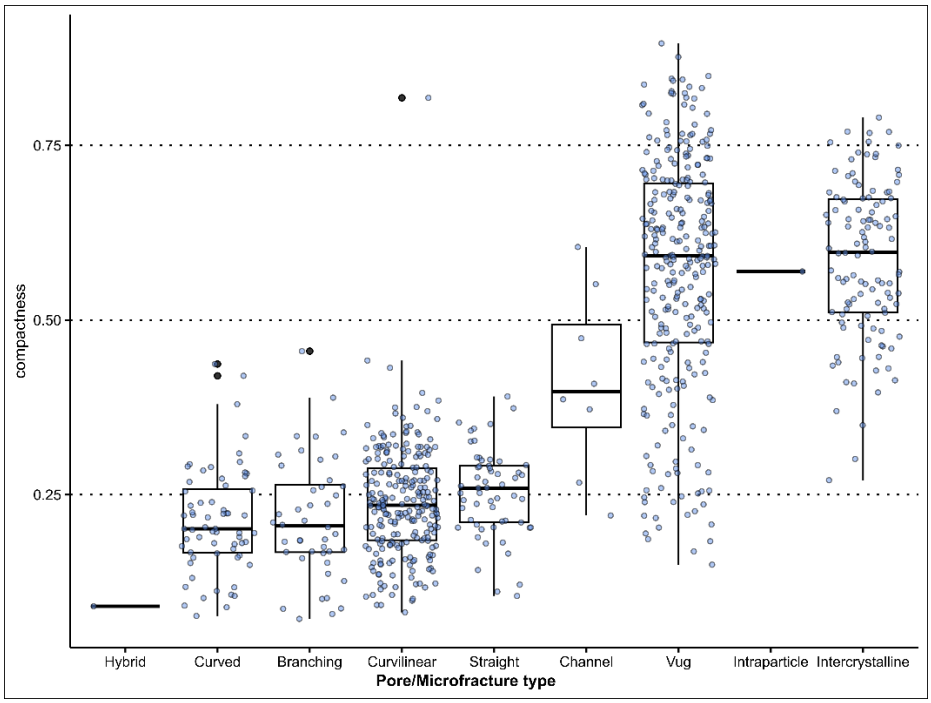

Figure S7d. Boxplot of compactness ranges for the secondary labels of pore and microfracture types.

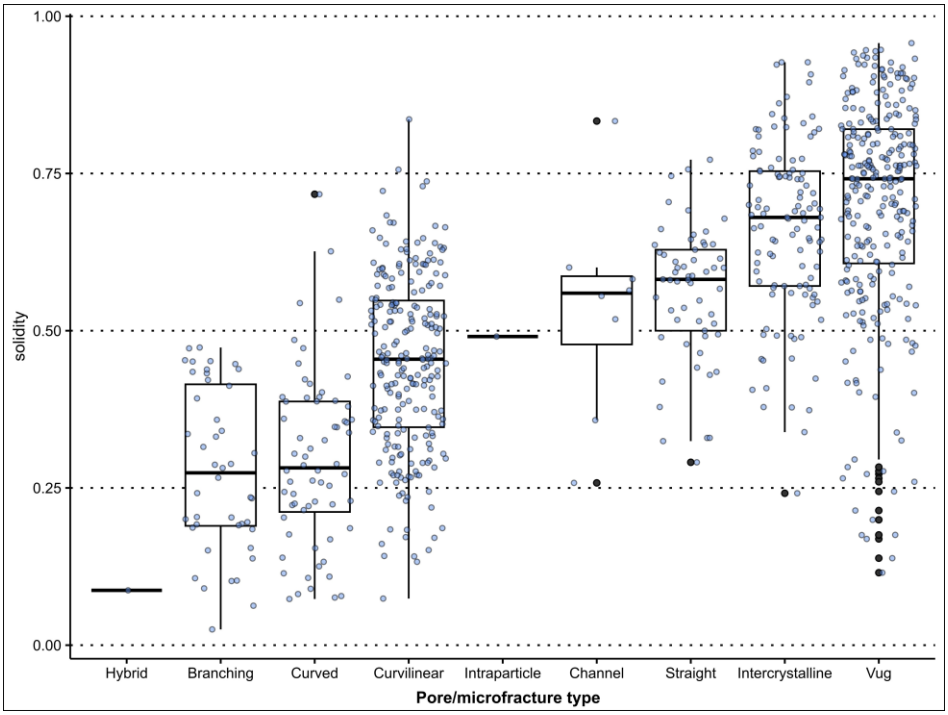

Figure S7e. Boxplot of solidity ranges for the secondary labels of pore and microfracture types.

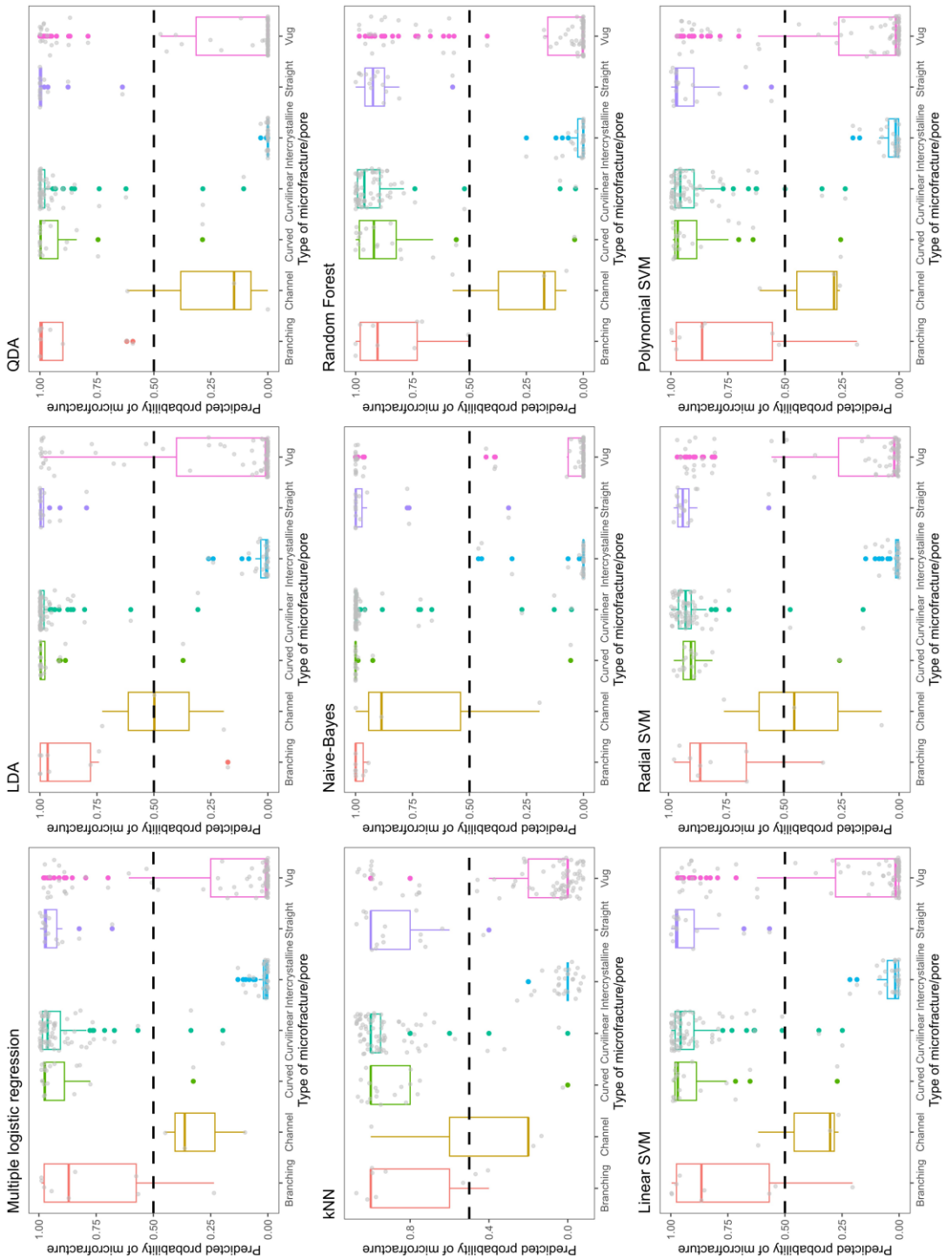

Figure S8. Probability of microfracture prediction per secondary label for each supervised model. The dashed line represents the 50% decision threshold, any objects above 50% are classified as microfractures and any object below the threshold are classified as pores.

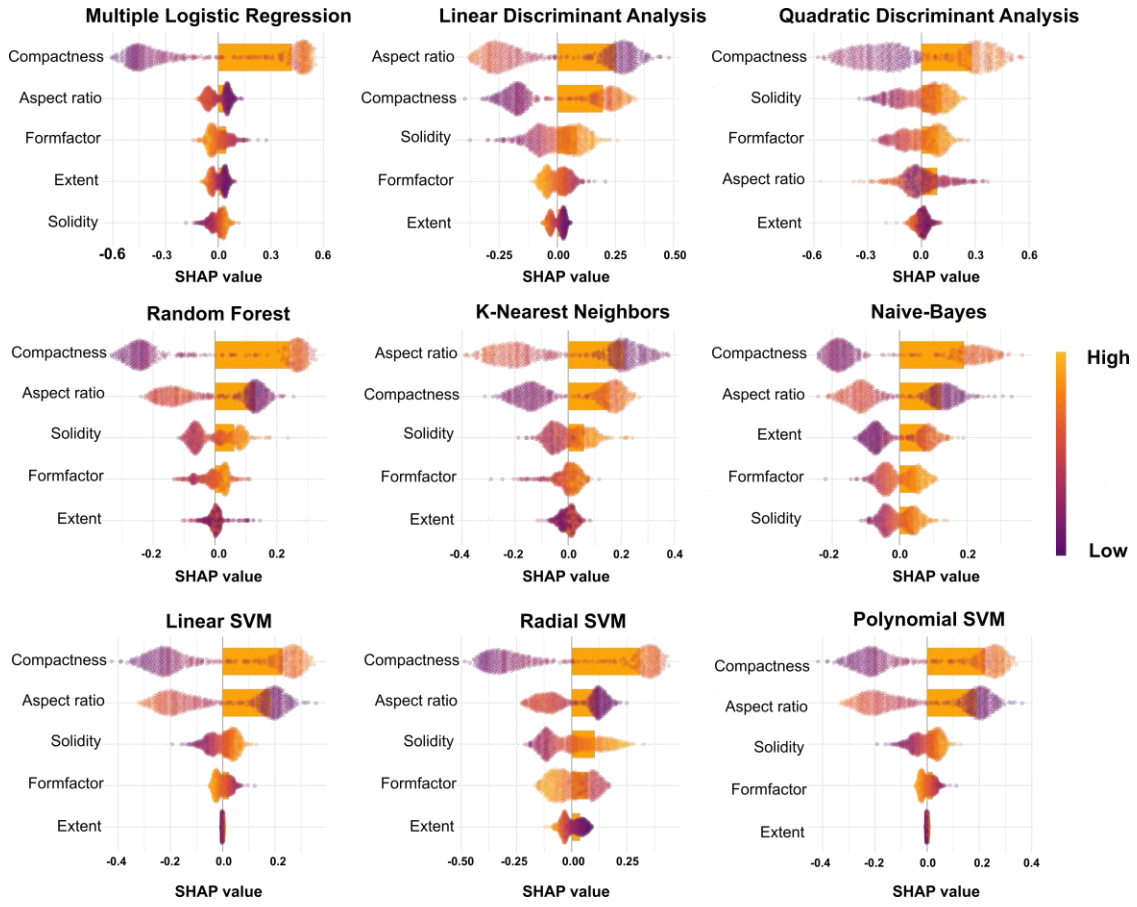

Figure S9. Probability of microfracture prediction per secondary label for each supervised model. The dashed line represents the 50% decision threshold, any objects above 50% are classified as microfractures and any object below the threshold are classified as pores.

Table S1. Studies on automated pore typing using AI.

Table S2. List of hyperparameters for each supervised ML model and the final values chosen.

Model	**Number of hyperparameters**	**Hyperparameters**	**Final values**
MLR	0	-	-
LDA	0	-	-
QDA	0	-	-
kNN	1	Number of neighbours (k)	k = 3
Naive-Bayes	3	Laplace (fL), Kernel, bandwidth adjust (BA)	fL = 2, Kernel = True, BA = 2
Random Forest	1	Number of randomly selected variables at each split (mtry)	mtry = 2
Linear SVM	1	Cost (C)	C = 0.3
Radial SVM	2	Cost (C) and Sigma	C = 22.63, Sigma = 0.04
Polynomial SVM	3	Cost (C), degree of polynomial, and scale	C = 0.974 , degree = 4, scale = 0.1

Data Set S1. Image data for the study.