Materials Science & Engineering A 838 (2022) 142738

Contents lists available at ScienceDirect

Materials Science & Engineering A

journal homepage: www.elsevier.com/locate/msea

Understanding slip activity and void initiation in metals using machine learning-based microscopy analysis

Joseph Indeck [a], David Cereceda [b], Jason R. Mayeur [a], Kavan Hazeli [c,*]

[a] Mechanical and Aerospace Engineering Department, The University of Alabama in Huntsville, USA
[b] Department of Mechanical Engineering, Villanova University, USA
[c] Aerospace and Mechanical Engineering Department, The University of Arizona, USA

ARTICLE INFO

Keywords:
Machine learning
Mechanics
Fatigue
Crystal plasticity
Slip transmission

ABSTRACT

The use of machine learning techniques to supplement traditional data analysis in mechanics and materials research can improve the understanding of microstructure-property relationships. Identification of key microstructural features or correlation between deformation mechanisms and material response can be discerned that might otherwise have been overlooked. Motivated by the possibilities of gaining additional insight into the process of void nucleation in polycrystalline metals, several machine learning techniques are applied to the analysis of mesoscopic deformation mechanisms as determined by experimental characterization and modeling. Results from crystal plasticity modeling, experimental microstructural analysis, and theoretical models of slip transmission are combined to test a hypothesis regarding fatigue-induced void nucleation. Unsupervised spectral clustering was used with results from crystal plasticity simulations to characterize slip system activity for different crystallographic orientations. The slip system activity as determined by the clustering analysis was then fed into a K-nearest neighbor classifier to quantify the probability of slip transmission across different grain boundaries of interest and analyze grains containing fatigue-induced voids. An unique and unanticipated result from the unsupervised clustering analysis shows that including a group of partially-active slip systems was more appropriate than using the binary classification of active/non-active. Predicted slip activity behavior in a face-centered cubic material was shown to differ significantly from that of a body-centered cubic material due to non-Schmid effects. The outcome of the overall analysis was that grains containing fatigue-induced voids were more likely to be surrounded by grains with orientations that inhibited slip transmission according the Lee-Robertson-Birnbaum (LRB) criteria. Finally, it is demonstrated that smaller datasets using limited simulation results were equally effective at predicting a similar outcome when additional physical descriptors for the slip system activity are used.

1. Introduction

Machine Learning (ML) is a branch of artificial intelligence (AI) that uses a set of statistical, probabilistic, and optimization methods to automatically detect patterns in data [1,2]. Given its ability to uncover patterns, make predictions, and decisions, it has demonstrated outstanding capabilities in several fields such as speech and image recognition, spam detection, cancer prognosis, and personalized recommendations on streaming websites, among others [3–6].

In Mechanics of Materials, the application of ML techniques to understand the behavior of deformable solid materials and structures under internal and external actions has been the subject of much research and discussion in recent years. For example, the relationship between microstructure and twinning behavior, twin formation and twin propagation across grain boundaries, in Mg AZ31 was determined using decision trees [7]. Applying machine learning not only enabled the identification of the crystallographic attributes that influence twinning behavior, but also a ranking of the attributes based on their importance. In particular, twin formation was most influenced by the grain size and basal Schmid factor while twin propagation was most influenced by the grain boundary length and misorientation angle. In a similar manner, 25 different descriptors of steel were taken from an existing National Institute of Material Science fatigue dataset, and analyzed using machine learning to determine the most influential variables controlling

* Corresponding author.
E-mail address: hazeli@arizona.edu (K. Hazeli).

https://doi.org/10.1016/j.msea.2022.142738
Received 2 September 2021; Received in revised form 22 January 2022; Accepted 24 January 2022
Available online 4 February 2022

fatigue life [8]. The variables included steel compositions, manufacturing process parameters, and alloy properties. Assessment of all 25 parameters revealed tempering temperature, carburizing temperature, and through hardening temperatures as three of the most significant predictors of subsequent fatigue life. Fatigue crack growth rate of nickel-based superalloys was analyzed using 51 different input variables and concluded that log ΔK (where ΔK is stress intensity factor range) was more influential on the fatigue crack growth rate than ΔK as expressed in the seminal Paris law [9]. An important aspect of machine learning algorithms is the flexibility to accept any number of input variables as Singh et al. used 108 variables in their predictions of yield and tensile strengths in steel as a function of composition and rolling parameters [10]. The relative importance of individual parameters, such as initial plate thickness, percent reduction during rolling passes, delay time between rolling passes, weight percent of alloying elements, and carbon content can all be determined from the model.

Massive datasets are already becoming standard in some mechanics and materials research making the familiarization of oneself with these approaches unavoidable. A single tomography scan from x-ray microCT or diffraction CT can be > 100 GB, requiring advanced techniques for not only the data processing, but also data storage and handling [11]. Additionally, as the materials genome initiative [12,13], makes large amounts of research data available, it will be necessary to use data-driven analysis techniques to advance materials understanding because a person cannot practically synthesize and analyze data at this scale.

This work shows how to efficiently employ two data-driven approaches, spectral clustering and K-nearest neighbor classification (KNN), to better understand slip activity and fatigue-induced void formation. Each of these approaches provided insights that might have otherwise gone unnoticed. We hope that others may find inspiration for employing machine learning in their mechanics research through the discussion and illustration of these techniques applied to a relevant application. This paper demonstrates how the machine learning techniques were applied to electron backscatter diffraction data supported by single crystal plasticity modeling of slip system activity to draw conclusions about fatigue-induced void nucleation. The field of machine learning, especially that of deep learning, is incredibly diverse and the authors have not attempted to provide an exhaustive list of publications for machine learning. Rather the reader is pointed to a review article by Schmidhuber [14], as a starting point for an overview and history of deep learning.

2. Methodology and machine learning approaches

The different machine learning techniques used in this work are briefly introduced and discussed in this section. Each technique addresses a different aspect of the data analysis, specifically, grouping the data into different sets (clustering) and predicting what set an unknown query point will belong to (classification). There are numerous clustering and classification techniques available, and the conclusions reached using the methods adopted in this work can be reached using alternative methods as well. Python's scikit-learn toolbox was used for the clustering, classification, and cross-validation algorithms [15].

An overall workflow of the analysis methodology is shown in Fig. 1 and details the inputs and outputs of each analysis technique. The first step of the analysis was single crystal plasticity modeling of face-centered cubic (FCC) and body-centered cubic (BCC) single crystals to quantify slip details including Schmid factor, accumulated slip, and slip rate information. All of the information generated from the crystal plasticity simulations were used as feature inputs into the unsupervised spectral clustering algorithm in step two. The clustering analysis determines how each individual slip system should be grouped and creates a set of classification labels to be used in the subsequent analysis steps. Step three is a cross-validation analysis that optimizes the number of neighbors to use with the labels created in step two. This step uses different features than the crystal plasticity results analysis in order to be consistent with the type of data available from electron backscatter diffraction (EBSD) measurements that the classifier will operate on in the last step. The clustered dataset from step two, along with the optimized parameters from step three, produces the overall classified dataset of step four. This classified dataset is finally used with EBSD scan data

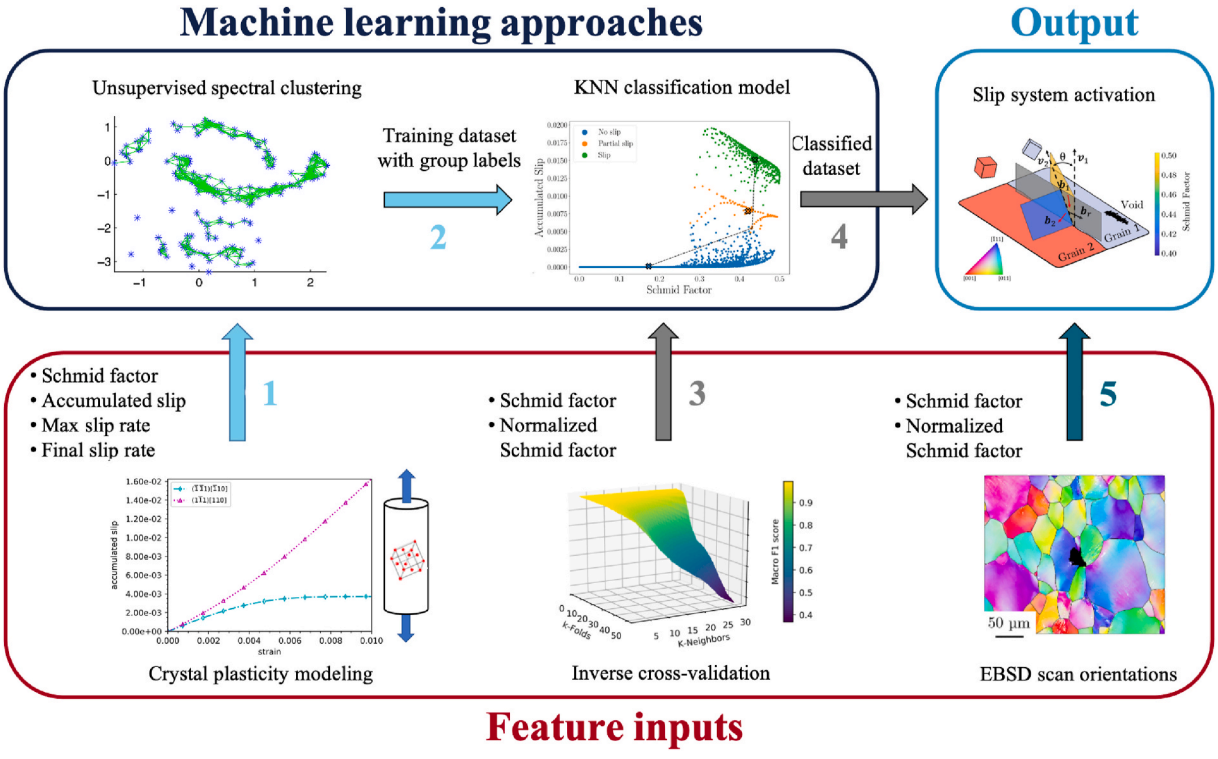


Fig. 1. Step-by-step workflow of the analysis processes showing the inputs and outputs at each step.

to determine slip system activity for any arbitrary grain orientation.

2.1. Spectral clustering

Clustering analysis is an approach used to find points that belong to the same group based on certain similarity metrics calculated between the points in the dataset [16–18]. Unsupervised clustering is an effective tool to classify points into an optimal number of different groups when there is no prior knowledge about the grouping. Different approaches are available for cluster analysis including: centroid-based clustering (e.g. K-means), density based clustering (e.g. density-based spatial clustering of applications with noise (DBSCAN)), probabilistic models (e.g. a Gaussian mixture model (GMM)), agglomerative clustering using different linkage criteria, and similarity-based clustering (e.g. spectral clustering) [18].

Spectral clustering is useful for grouping datasets with arbitrary shapes. Its name originates from spectral graph theory [19], and the use of spectral decomposition of the Laplacian matrix that is derived from an affinity (sometimes called adjacency) matrix. The affinity matrix is created based on links between data points and the strength of those links. Therefore, the distances between data points, and not the actual values of the data points, are used with the general goal to classify similar points into a group and dissimilar points into different groups. There are several methods used to group data points, including connecting points with their K-nearest neighbors, within a certain neighborhood size of ε, or all points with a fully connected graph and then using a kernel function to delineate between similar and dissimilar points.

In this work, a fully connected graph was used with a Gaussian kernel for calculating the affinity matrix, W, which describes the weights between each point:

$$W(x_i, x_j) = e^{\frac{-\left\|x_i - x_j\right\|^2}{2\sigma^2}} \tag{1}$$

where the numerator is the squared distance between the two points and σ is the neighborhood width. The degree matrix, D, is the diagonal matrix with entries equal to the sum of each row in the affinity matrix, i.e.:

$$D_{ii} = \sum_{j=1}^{n} W_{ij} \tag{2}$$

The unnormalized, L, and normalized Laplacian matrices, L_{sym}, respectively, are calculated as:

$$L = D - W \tag{3}$$

$$L_{\text{sym}} = D^{-1/2} L D^{-1/2} \tag{4}$$

Eigenvalues, λ, and eigenvectors, u, of L_{sym} can then be calculated and are used to determine which points are clustered together. One drawback of spectral clustering is that L and L_{sym} are $n \times n$ matrices, where n is the total number of data points. Thus, their calculation for large datasets requires a lot of memory and computation time. Additional discussions on the formulation of the affinity, degree, and Laplacian matrices, can be found in Refs. [18,20,21]. Finally, a measure of cluster validity, or scoring metric of the cluster performance, is needed to optimize the clustering parameters in the unsupervised case. Section 3.1.2 discusses the spectral clustering optimization process employed in this work.

2.2. K-nearest neighbor classification

The k-nearest neighbors (KNN) algorithm is a non-parametric method for classification and regression predictions that was first formulated in the early 1950s [22]. Its usage has significantly expanded since then, in part because it does not require information about the data distribution. A KNN classifier is created using known data called a training dataset. Any arbitrary query point can then be classified by analyzing a set number, K, of neighboring points in the training dataset. Point-to-point distances between the query point and each value in the training dataset are calculated using a specified distance metric. The Euclidean distance, d, between two points is frequently used. For Cartesian coordinates in n-dimensional space, d is given by:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{5}$$

where the two points have coordinates x_i and y_i, respectively. Classification of a query point is decided on by using either a majority voting or weighted distance method. Majority voting classifies a query point to the class with the highest estimated probability among the K-nearest neighbors. For example, if K = 5, and there are three occurrences of "Group A″ and two occurrences of "Group B″ among the five nearest neighbors, then the query point will be classified as "Group A". Issues may arise when there is an equal frequency of occurrence among groups. Such as three occurrences of both "Group A″ and "Group B″ if K = 6, or two occurrences of "Group A″ two occurrences of "Group B″ and one occurrence of "Group C″ for K = 5 when more than two classifications exist. A specified weighting function can be applied to each distance to avoid this situation. For example, an inverse weighting function can be used:

$$W_i = \frac{1}{d_i} \tag{6}$$

where d_i is the Euclidean distance, defined in Eq. (5), and W_i is the weighted distance. This function can be applied to each of the K-nearest neighbors and the query point will then be classified to the group with the largest weighted distance total. Thus, majority voting classification weights each neighbor equally, whereas closer neighbors have a stronger influence on the outcome in weighted distance classification.

In this work, we chose the inverse k-fold cross-validation method and the $F1$ score during hyperparameter optimization. Inverse k-fold cross-validation was also used to assess the performance of the model when classifying slip activity. The reader is referred to A for more details about the variety of k-fold cross-validation methods and performance metrics.

3. Results and discussion

In this section, the clustering and classification techniques described above are utilized to improve the current understanding about the interplay between slip transmission, slip irreversibilities, slip interaction, and void nucleation on fatigue damage initiation. Each subsection discusses how the techniques were applied and the resultant insights gained from the analysis, from a mechanics point of view. These insights were critical in the determination of slip activity differences between the two crystalline systems and their contribution to fatigue-induced void nucleation.

3.1. Slip activity grouping with spectral clustering

Single crystal plasticity simulations were performed to quantify the slip activity on each slip system as a function of orientation under a fixed global loading direction, and the accumulated slip histories were used to determine active slip systems. Specific grain orientations were selected through a discretization of the standard stereographic triangle. An inverse pole figure (IPF) key showing each of the 593 individual orientations considered is shown in Fig. 2. This is admittedly a simplified approach to estimate the slip system activity of grains within a polycrystal as it is well-established that the stress state of grains within a polycrystalline material subjected to remote uniaxial loading is not necessarily uniaxial due to the influence of grain-grain interactions. However, the point of this work is not to perform state-of-the-art 3D crystal plasticity simulations of realistic statistical volume elements.

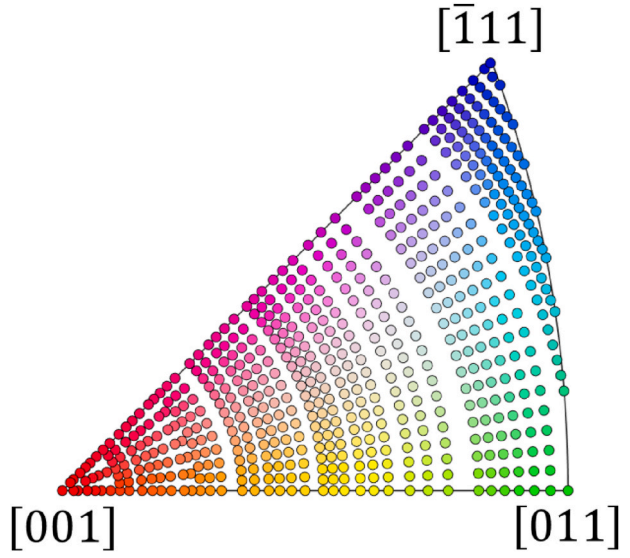


Fig. 2. Each dot on the IPF map represents a grain orientation whose response was simulated to determine accumulated slip histories for each slip system. In total 593 orientations were modeled.

Rather, the primary goal of these simulations is to generate necessary quantitative information on slip system activity under nominal loading conditions, beyond what can be obtained from basic Schmid factor analysis, from an efficient computational model that can be used to supplement EBSD data in an analysis of the likelihood of slip transmission across particular grain boundaries. State-of-the-art full-field 3D crystal plasticity simulations are computationally expensive and also involve approximations in converting 2D EBSD images into 3D polycrystalline ensembles. However, they unarguably offer a more accurate representation of the intergranular constraints that exist between neighboring grains. Taking an in-depth look at how slip activities obtained from full-field 3D polycrystalline simulations replacing the single crystal plasticity results used in the current analysis is a worthy endeavor, but it is postponed for future work.

3.1.1. Crystal plasticity modeling

Simulations for both FCC and BCC single crystals were performed for each orientation under uniaxial loading to 0.01 strain. The FCC simulations use the crystal plasticity constitutive model described in Ref. [23], which accounts for elastic anisotropy and uses thermally-activated slip kinetics with a dislocation density-based hardening law. The model parameters were taken from published literature values for pure aluminum [24], but with an appropriately scaled athermal slip resistance to account for the increased yield strength of 7075-T6 aluminum relative to the pure material. This is a reasonable modeling assumption given the quasi-static loading conditions and small applied strains e.g., initial yield behavior. The BCC simulations, which account for non-Schmid effects, were performed using the model and parameters given in Ref. [25].

Several model output variables were selected as input features for the dataset used for clustering: accumulated slip history, $\widehat{\gamma}^{\alpha} = \int |\dot{\gamma}^{\alpha}| dt$; normalized accumulated slip value, $\widehat{\gamma}^{\alpha}/\overline{\varepsilon}^{p}$; maximum slip rate, final slip rate, Schmid factor, and normalized Schmid factor. The normalized accumulated slip metric is computed by dividing the accumulated slip history on each system by the effective plastic strain history. The effective plastic strain is determined by:

$$\overline{\varepsilon}^{p} = \int \sqrt{\frac{2}{3}\dot{\boldsymbol{\varepsilon}}^{p} : \dot{\boldsymbol{\varepsilon}}^{p}}\, dt \tag{7}$$

where the plastic strain rate is defined as:

$$\dot{\boldsymbol{\varepsilon}}^{p} = \frac{1}{2}\sum_{\alpha} \dot{\gamma}^{\alpha}(\mathbf{s}^{\alpha} \otimes \mathbf{n}^{\alpha} + \mathbf{n}^{\alpha} \otimes \mathbf{s}^{\alpha}) \tag{8}$$

Lastly, the "total Schmid factor", which includes non-Schmid contributions for the BCC material is important to account for since slip behavior in BCC crystals is often influenced by non-glide stresses. Table 1 shows the features that are provided as inputs to the unsupervised clustering algorithm for both the FCC and BCC datasets. The Schmid (*SF*) and total Schmid (*TSF*) factor are calculated according to

$$SF^{\alpha} = \mathbf{P}_{s}^{\alpha} : (\mathbf{l} \otimes \mathbf{l}) \tag{9}$$

$$TSF^{\alpha} = \left(\mathbf{P}_{s}^{\alpha} + \mathbf{P}_{ns}^{\alpha}\right) : (\mathbf{l} \otimes \mathbf{l}) \tag{10}$$

where $\mathbf{l}$ is a unit vector parallel to the loading direction, and the Schmid and non-Schmid tensors are defined as

$$\mathbf{P}_{s}^{\alpha} = \mathbf{s}^{\alpha} \otimes \mathbf{n}^{\alpha} \tag{11}$$

$$\mathbf{P}_{ns}^{\alpha} = a_1 \mathbf{s}^{\alpha} \otimes \mathbf{n}_{ns}^{\alpha} + a_2(\mathbf{n}^{\alpha} \times \mathbf{s}^{\alpha}) \otimes \mathbf{n}^{\alpha} + a_3\left(\mathbf{n}_{ns}^{\alpha} \times \mathbf{s}^{\alpha}\right) \otimes \mathbf{n}_{ns}^{\alpha} \tag{12}$$

For a detailed description of the non-Schmid formulation, the reader is referred to the original work [25].

In the following, a brief rationale is given for choosing these simulation outputs. Slip systems with higher amounts of accumulated slip and those that contain a higher percentage of a grain's overall effective plastic strain are expected to have a higher probability of being grouped as active. The slip rates account for the intensity of slip activity during deformation and at the end of loading when the material has undergone a non-trivial amount of plastic deformation and will indicate whether or not the system is still actively accomodating slip. Finally, the Schmid factor, normalized Schmid factor, and total Schmid factor for the BCC crystal, capture the amount of driving force on the slip system. Slip systems were classified into different groups based on their overall slip activity employing these normalized features with the unsupervised clustering analysis, which is described in the next section.

3.1.2. Unsupervised clustering

An unsupervised clustering approach was implemented in order to divide the crystal plasticity simulation data into groups of slip systems that experience similar amounts of slip activity. A clustering technique was used since the maximum Schmid factor may not be the dominant predictor of sip system activity in all situations. For example, when considering slip transmission across a grain boundary in the study of fatigue-induced voids, the slip system with the maximum Schmid factor might not be the most likely to transmit dislocations because of other geometric factors involved with slip transmission. Therefore, a non-arbitrary method to distinguish between active and non-active slip systems was required. Unsupervised clustering algorithms are well suited to address this question.

Initially, feature scaling was applied to all input parameters. This preliminary step is critical as it addresses, for example, that the Schmid factor is independent of loading magnitude whereas slip and slip rate depend on the magnitude of loading. Then, both spectral clustering and density-based clustering methods were assessed. The comparison of

Table 1
Input features into the unsupervised clustering algorithm for both FCC and BCC crystals.

Input features	FCC	BCC
Accumulated slip	✓	✓
Normalized slip	✓	✓
Maximum slip rate	✓	✓
Final slip rate	✓	✓
Schmid factor	✓	✓
Normalized Schmid factor	✓	✓
Total Schmid factor		✓

these two approaches for characterizing slip system activity revealed several advantages of using spectral clustering in this work. First, spectral clustering was able to group data points that have any arbitrary shape making it a useful algorithm for unsupervised clustering when the distribution of the dataset is unknown. Thus, no prior information about the input variables are needed, such as the pattern or correlation between Schmid factor and accumulated slip. This section details the processes used to optimize input parameters for the spectral clustering, namely the number of clusters and the kernel coefficient. Furthermore, there was a large number of points that end up being classified as outliers in a density-based approach. Over 10% of the dataset were considered outlying points. While these points can be excluded from the dataset in other applications, doing so with this dataset would have inaccurately skewed the clustering results due to eliminating a substantial amount of the data. Therefore, using a density-based approach would have required making an assumption about a grouping assignment for the outliers.

The input parameter optimization began by finding the optimal number of clusters. Following the algorithm outlined by Ng et al. [26], a set of eigenvectors from L_{sym} in Eq. (4), can be grouped together in the matrix X, where each column in X corresponds to an eigenvector. The matrix Y is then calculated by normalizing each row of X to a unit length:

$$Y_{ij} = \frac{X_{ij}}{\left(\sum_j X_{ij}^2\right)^{1/2}} \tag{13}$$

The matrix Y can then be analyzed with a K-means clustering algorithm to determine the optimal number of clusters. This process creates a reduced dimension space of the largest eigenvectors. The squared distance between each point and its designated cluster centroid are summed to determine the number of clusters.

Applying these steps to the datasets generated by the crystal plasticity simulations for both the FCC and BCC crystals, the optimum number of clusters was determined. Fig. 3a and b show the K-means cluster results of the first five eigenvectors of L_{sym} for the FCC and BCC dataset, respectively. In both datasets an inflection point, or "elbow", where the data value decreases linearly afterwards was seen at three clusters. This analysis is important because it shows that simply dividing the slip activity between active and non-active is not the most appropriate division of data. Therefore, the addition of a third cluster, to physically represent a partially-active slip system, was used. Another inflection point at six clusters was also observed in the BCC dataset. Still, we chose to keep three clusters for both FCC and BCC datasets. This decision is rooted in our goal of understanding slip activity through a physics informed approach of spectral clustering that leverages the information learned from the data with the fundamental laws describing the plastic behavior of FCC and BCC metals.

The next step in the spectral cluster optimization process was the determination of the neighborhood width, σ, in Eq. (1). Both the average silhouette score [27] and Davies-Bouldin (DB) index [28] were used to assess the cluster validity.

A dataset's silhouette score is based on the compactness of each cluster and the separation between clusters, and will range between −1 to 1. Compactness is defined as the mean distance between one data point to all other data points in the same cluster. Separation is defined as the mean distance between a data point and all other data points in the next closest cluster. Therefore, a silhouette score will be closer to one if every data point in each cluster is tightly grouped, and each cluster is well separated and far apart from all other clusters. A data point with a negative silhouette score indicates that it was incorrectly clustered and is closer (on average) to all the other data points of a different cluster than its own cluster. For its part, the DB index is the ratio of the intra-cluster dispersion to the inter-cluster distances. Therefore, a lower DB index implies small intra-cluster dispersion (i.e. a tightly compact cluster) and large inter-cluster distances (i.e. well separated clusters) and indicates better clustering of the data. Both metrics, the silhouette score and the DB index, operate on the concept that minimizing the intra-cluster distances while maximizing inter-cluster distances indicates better clustering. Metrics quantifying cluster quality are required for unsupervised clustering since no classification labels exist a priori to assess the performance of the clustering algorithm. These two metrics were calculated for a range of σ values and the results are shown in Fig. 4a and b for the FCC and BCC dataset, respectively. The σ value that optimized the quality of clustering was selected, along with three clusters, for use in the final spectral clustering algorithm.

3.1.3. Slip system distribution per grain

One aspect that can be explored after clustering is how each grain's slip systems are classified. Slip system breakdown of each grain shows how the slip activity is taking place inside the grains. A chart of the FCC dataset is shown in Fig. 5 and reveals that in 581 of the modeled cases a grain will have only one active slip system or two partially active slip systems. It was rare for a grain to accommodate both a partially-active and fully-active slip system, or no active slip system at all. There was still slip accumulation in those grains that were classified with zero active slip systems, however, the slip characteristics relative to all the other slip systems in other grain orientations was not enough to be considered active. Therefore, slip activity within an FCC grain can be generalized with two categories: (1) grains that have a single primary active slip system, or (2) grains that more evenly distribute the slip behavior between two partially-active slip systems.

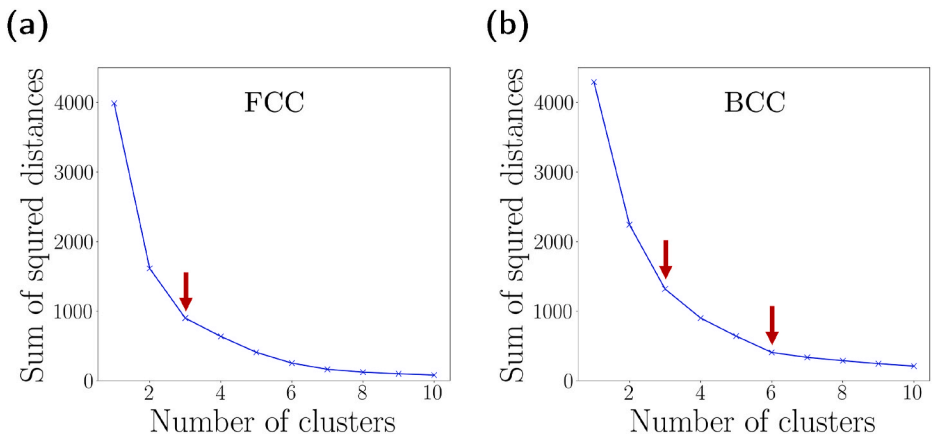


Fig. 3. Sum of the squared distance between each point and its cluster centroid for the (a) FCC and (b) BCC dataset. A neighborhood width value of 1 was used in both subfigures. An inflection point in each dataset at three clusters was observed. The three clusters will then represent non-active, partially-active, and active slip systems.

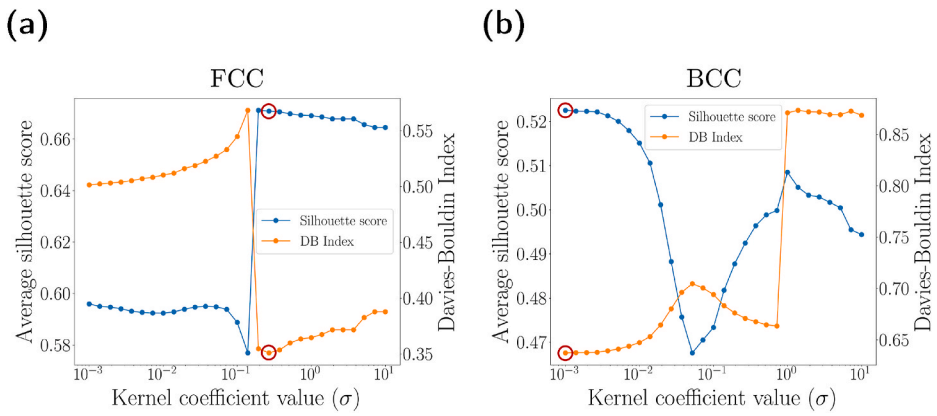


Fig. 4. Silhouette score and Davies-Bouldin index as a function of the kernel neighborhood width, σ, for fully connected spectral clustering of the (a) FCC and (b) BCC dataset. A high silhouette score and low DB index is an indication of better quality clustering. The selected value of σ which optimizes the clustering is circled.

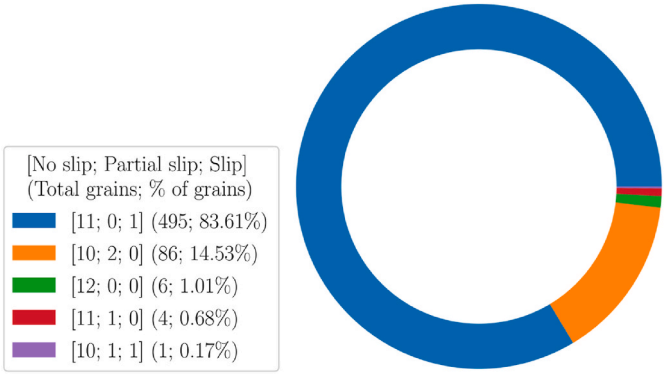


Fig. 5. Chart showing the breakdown of slip system classification for each FCC grain. It was found that 98% of the grains can be divided into only two types; grains that have one fully-active slip system and grains that have two partially-active slip systems. Values in the brackets represent the number of slip systems for each slip activity classification within a single grain, whereas values in the parentheses represent the total number and percentage of grains with that specific slip system classification distribution.

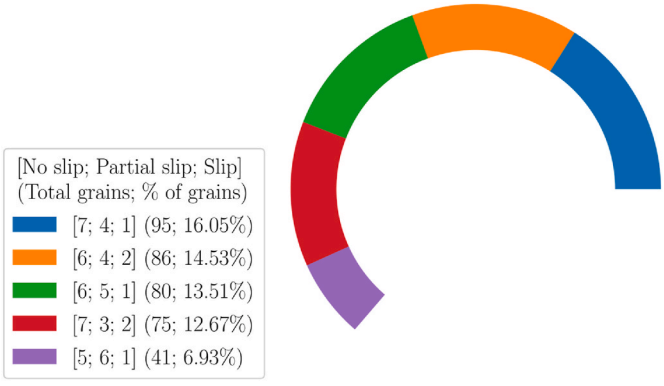


Fig. 6. Chart showing the breakdown of slip system classification for each BCC grain. It was found that multiple slip systems were activated in all of the grain orientations. This type of slip activity was considerably different compared to the FCC material. The remaining 17 slip activity combinations are omitted for clarity. Values in the brackets represent the number of slip systems for each slip activity classification within a single grain, whereas values in the parentheses represent the total number and percentage of grains with that specific slip system classification distribution.

On the other hand, the BCC dataset had a much greater variety in the different combinations of active slip systems. In total there were 22 different combinations and the top five most frequent classifications, accounting for 60% of the grains, are shown in Fig. 6. It was found that every grain orientation contained at least two partially-active or fully-active slip systems. The number of active systems in each grain is in contrast to the FCC grains where a majority of them contained a single active slip system. Distribution of active slip system combinations in Fig. 6 for the {110}< 111 > family of slip systems is expected to remain the same even if both the {112}< 111 > and {123}< 111 > family of slip systems were also accounted for in the modeling. The activation of particular {110}< 111 > slip systems did not change when modeling included the other slip system families [29]. The same {110}< 111 > slip systems are activated despite the fact that the {112}< 111 > and {123}< 111 > slip families have been shown to accommodate non-trivial amounts of shear strain during deformation of BCC metals [30]. In other words, modeling only the {110}< 111 > slip systems did not change which specific systems were activated during loading, it only changed the magnitude of accumulated slip because there were fewer slip systems available to accommodate the same magnitude of imposed strain.

3.1.4. Feature importance

Feature importance indicates how much influence a particular feature has in the fitting of a decision tree. A gain score can be calculated [31] at each split in the tree and attributed to the feature causing the split. Higher gain scores imply the associated feature is more important when making a prediction and deciding how to classify a data point. Gain scores for each feature are averaged over all branches of the tree and then normalized so their sum equals one. This importance metric can be calculated after the classification labels are created from the spectral clustering and shows which features are most important in determining the slip activity. Implementation of the Extreme Gradient Boosting (XGBoost) system [32] in Python was used to create the trees and calculate the average gain for each feature. The reader is referred to B for more details about the gain score calculation. The overall classification model was optimized by running a range of certain input parameters through a grid search to find the best resulting $F1_{weighted}$ score. Only 80% of the data was used in each iteration to avoid overfitting of the model.

Feature importance and a scatter plot of the two most important features for the FCC and BCC datasets are shown in Figs. 7 and 8, respectively. The Schmid Factor and total accumulated slip on a slip system had the largest amount of gain meaning they were the two most influential features when determining slip activity classification for both crystalline systems. It is shown in the scatter plot for the FCC crystals that these two variables do a good job separating out each slip activity group. Slip rates were also found to contribute substantially to the end

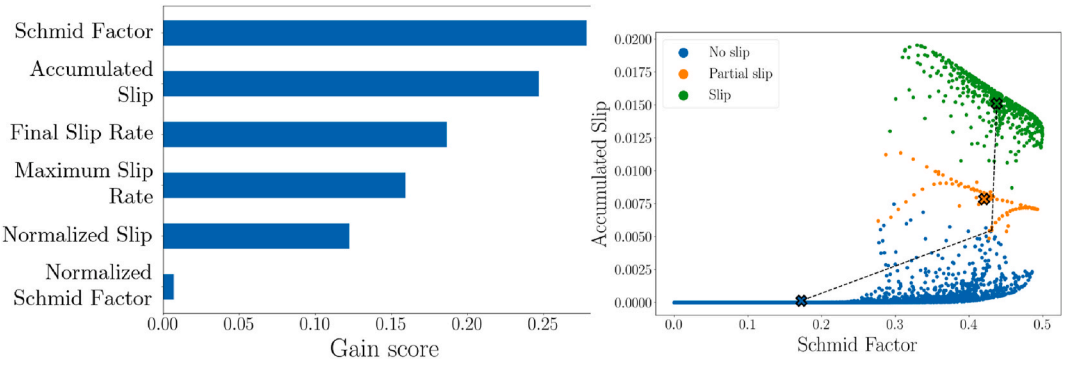


Fig. 7. (left) Gain score (i.e. feature importance) in deciding slip activity classification of the FCC dataset. (right) A scatter plot using the two most important features–the Schmid Factor and the overall accumulated slip on the slip system. These two features appear to separate each group effectively leading to their high importance. The markers denote each group's centroid and the lines show the distances from one partial slip point to the other group's centroids.

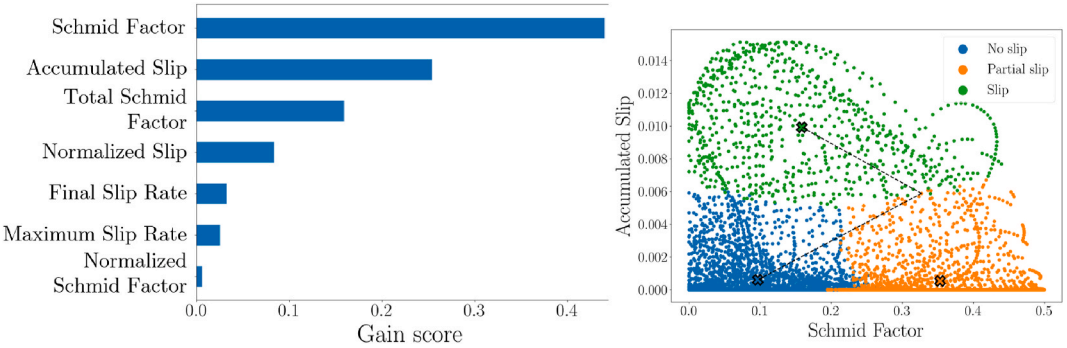


Fig. 8. (left) Gain score (i.e. feature importance) in deciding slip activity classification of the BCC dataset. (right) A scatter plot using the two most important features–the Schmid Factor and the overall accumulated slip on the slip system. There are many slip systems that have intuitively contradictory results; those with high accumulated slip but low Schmid factors, and those with low accumulated slip values but high Schmid factors. These sets of points illustrate the influence of the non-Schmid effects, accounted for in the third most important feature, the total Schmid factor. The markers denote each group's centroid and the lines show the distances from one partial slip point to the other group's centroids.

classification of each FCC slip system.

However, the twinning/anti-twinning non-Schmid effects in the BCC crystals (accounted for in the total Schmid factor feature) were found to be important in the classification and accounted for almost 20% of the gain. The influence of the twinning/anti-twinning non-Schmid effects is evident in the BCC scatter plot of the two most important features, accumulated slip and the Schmid factor. There are many slip systems that have relatively large amounts of accumulated slip with very low Schmid factors, in addition to many slip systems that have high Schmid factors and no accumulated slip. The fact that these seemingly incompatible extremes occur with regularity highlights the role non-Schmid effects play in the slip activity of BCC crystals, an observation that agrees with previous works in the field [33,34].

3.2. Slip transmission analysis with KNN

Data from the crystal plasticity simulations and clustering analysis was subsequently used in the calculation of parameters to estimate the likelihood of dislocation slip transmission between two grains. The K-nearest neighbors algorithm (described in Section 2.2) was used with the Schmid factor data to determine how the slip systems in any arbitrarily oriented grain as determined through EBSD should be classified: non-active, partially-active, and active. Input features for the KNN analysis, including the cross-validation and subsequent classification, only include the Schmid factor and normalized Schmid factor. A reduced set of input features, relative to what was used for spectral clustering, were used with KNN to be consistent with the information available from an EBSD scan. Information obtained from an EBSD scan includes the Euler angles, which determine grain orientation, and the Schmid factors of each slip system. Accumulated slip data cannot be readily obtained from an EBSD scan making those input features unavailable for classification. Our rationale for this criterion was twofold: (1) it was desirable to cluster and create classification labels with the additional information from the crystal plasticity modeling to obtain better clustering metrics and more accurate labels, and (2) it was desirable to train and cross-validate the KNN model with only information obtained from an EBSD scan to optimize the classification performance for that case.

3.2.1. Number of required crystal plasticity simulations

It is worthwhile to estimate how many simulations are required to obtain a consistently high success rate when using the nearest neighbor classification to predict slip system's activity. Crystal plasticity and molecular dynamic simulations provide valuable information that might be impossible to obtain experimentally. These results supplement experimental research leading to new insights and deeper understanding of materials behavior. However, the simulations can be computationally expensive to perform and require a lot of time to obtain results. Therefore, it is impractical to always require a large number of simulations covering all possible scenarios. Machine learning offers the ability to dramatically reduce the number of required simulations, yet still have meaningful results to inform analysis decisions. An important goal of this study is to validate the hypothesis that large datasets are not always needed to utilize and take advantage of machine learning algorithms. This section outlines an analysis showing that a limited number of simulations informing the KNN algorithm was still capable of producing accurate predictions. It might not be appreciated that even small

datasets–with only a few observations–can still be effective in creating a predictive machine learning model.

Section 2.2 described the process of repeated k-fold and inverse cross-validation in testing the efficacy of a model's performance. Using the inverse k-fold cross-validation data allocation allowed an assessment of how well the model could classify slip activity as a function of the number of crystal plasticity simulations used to create the training dataset. As specified in Fig. A18, the k-fold range was set from 2 to 50 meaning the classifier models were created using only 50%–2% of the data which was equivalent to running 297 to 12 crystal plasticity simulations, respectively.

The overall weighted $F1$ scores, receiver operator characteristics (ROC), and area under the curve (AUC) as a function of the percentage of training data and number of neighbors are shown in Figs. 9 and 10 for the FCC and BCC dataset, respectively. Both ROC and AUC metrics are used to check the performance of a machine learning model's ability to correctly classify unknown data points. An AUC value of 1 indicates the model is perfect at predicting data points between the different groups, whereas an AUC value of 0.5 indicates the model has no ability to discern between the different groups. First, it was found that the FCC dataset, shown in Fig. 9, had consistently high $F1$ scores even when data from only 25 simulations was used to train the classifier. The plane drawn at a weighted $F1$ score of 0.95 in the top row of Fig. 9 shows that only classifiers trained with less than 5% of the simulation data had $F1$ scores below that value. In addition, there was only a 2% decrease in the weighted $F1$ score when training the classifier with only 5% of the simulation data instead of 50%. However, the ROC and AUC values (bottom row) Fig. 9 show that the classifier does not perform as well at correctly determining partially-active slip systems. Furthermore, its ability to predict these types of slip systems degrades, as would be expected, with less amounts of training data. These observations demonstrate that only using a single scoring metric to judge the quality of a machine learning model can sometimes be misleading and result in over-confidence in that model.

Notably, it was found that the overall ability of the classifier in predicting the type of slip activity in the BCC dataset, Fig. 10, was worse than the FCC classifier. The data was downsampled and some non-active and partially-active data points were removed since there was approximately four times fewer fully-active data points. Nevertheless, the AUC score of 0.59 in predicting active slip systems is barely above that of a random classifier whose AUC equals 0.50, even with balanced classes. The weighted $F1$ score decreases by 7% when training the classifier with 5% of the total simulation data instead of 50%. However, because the AUC remained somewhat consistent, the classifier's ability did not become significantly worse with the lower amount of training data.

Overall, the classifier performed well with less training data for the FCC dataset, but worse for the BCC dataset. The results are what would be expected–namely that creating a model on too little data returns a lower quality classifier. Although, it was observed that decreasing the amount of training data from 50% to 5% did not significantly degrade the classifier to the point of uselessness. Therefore, similar performance can be obtained using a limited number of crystal plasticity simulations saving computation time, which highlights the ability of machine learning models to still work well with modest-sized datasets. Ultimately, a value of K = 11 was chosen for the FCC model and K = 19 for the BCC model based on standard cross-validation results.

3.2.2. *Partially-active slip system weighting*

A method to account for partially-active slip systems in the calculation of slip transmission parameters was also needed. The contribution

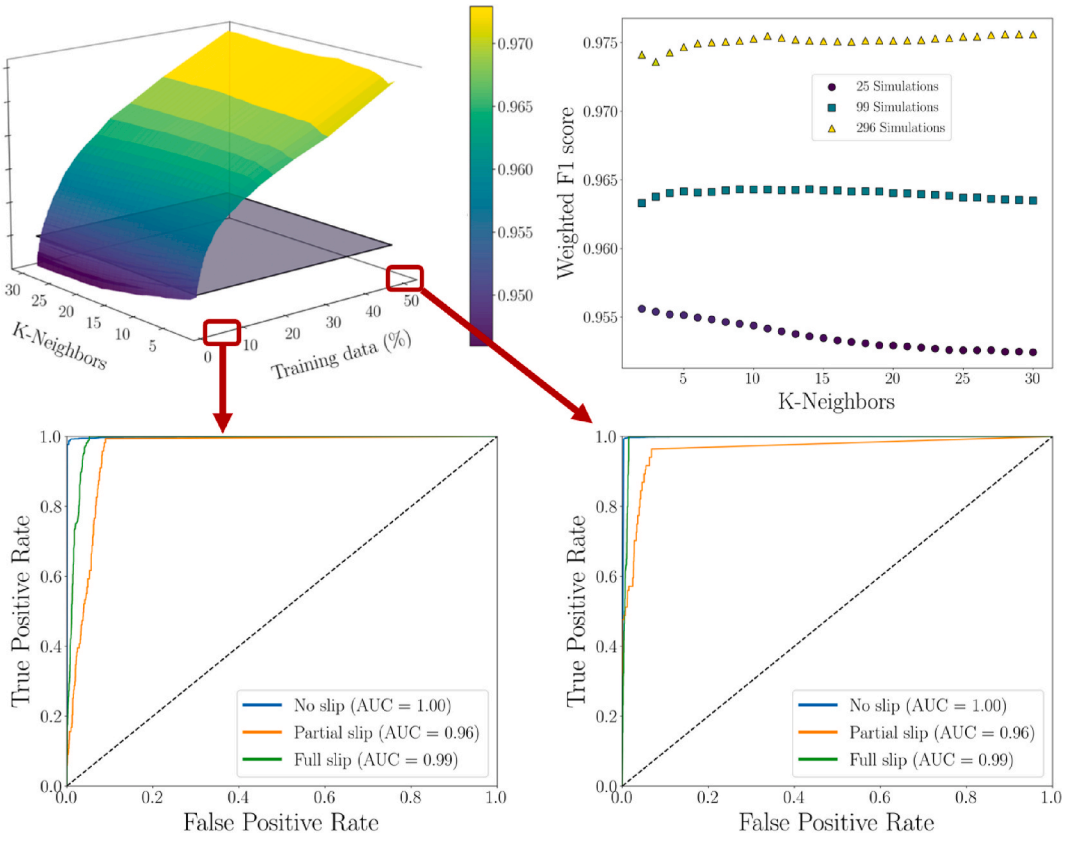


Fig. 9. Inverse cross-validation results for the FCC dataset. (top row) Weighted $F1$ scores using different percentages of the total simulation data to train the classifier. As expected, overall performance of the classifier decreases when less data is used to train it. A plane is drawn at a weighted $F1$ score of 0.95. (bottom row) Receiver operating characteristic curve for (left) 99 and (right) 296 simulations. Using lower amounts of training resulted in a decreased ability to predict partially-active slip systems, however the weighted $F1$ score decreases by only 2%.

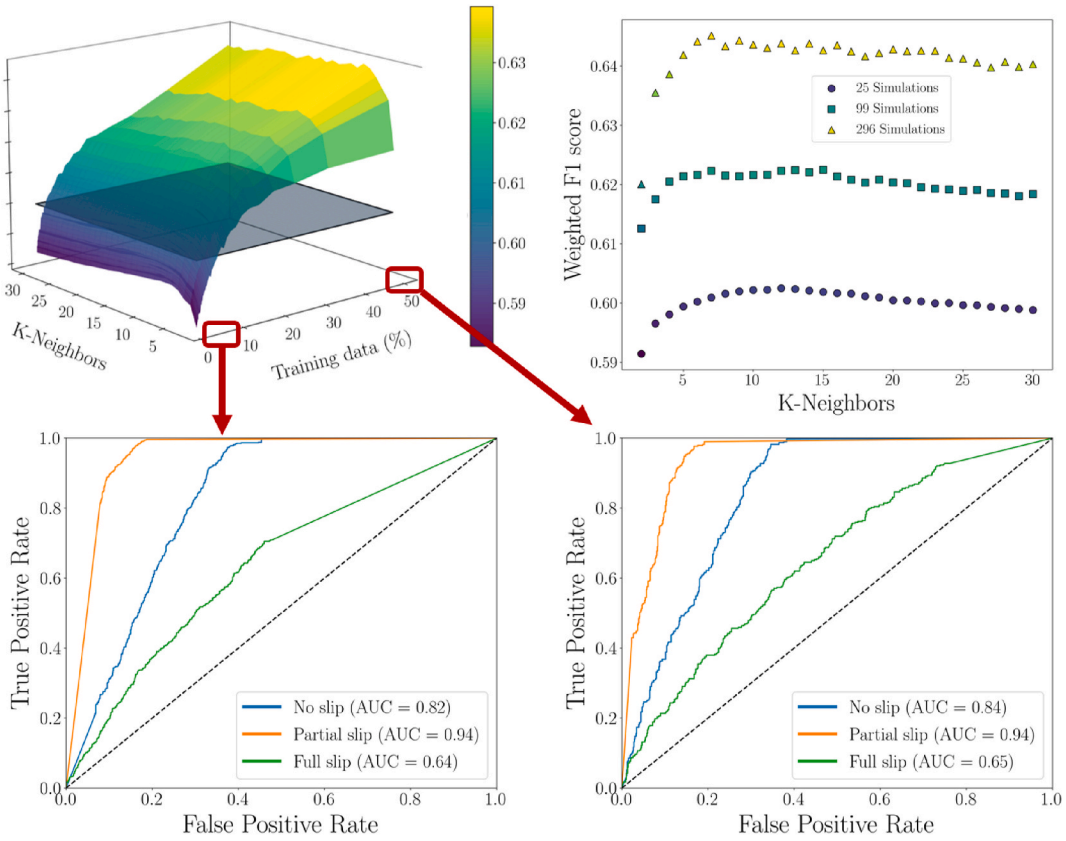


Fig. 10. Inverse cross-validation results for the BCC dataset. (top row) Weighted $F1$ scores using different percentages of the total simulation data to train the classifier. As expected, overall performance of the classifier decreases when less data is used to train it. A plane is drawn at a weighted $F1$ score of 0.60. (bottom row) Receiver operating characteristic curve for (left) 99 and (right) 296 simulations. There is not a significant change in the AUC using less data, even though the weighted $F1$ score decreases 7%.

to potential slip transmission should not be as great as active slip systems, but it also should not be negligible. Therefore, a contribution score between 0 and 1 was calculated using the Euclidean distance of each partially-active slip system to the centroids of the other two groups. Examples of the distance between a single partially-active slip system and the slip and no-slip centroids, are shown in the right hand side of Figs. 7 and 8. Each distance was weighted by the feature importance to give more weight to those features that have a greater influence in the slip activity classification. First, the data for each feature was normalized by the maximum value so each feature had the same range between 0 and 1. The weighted distance, $\overline{d}$, was calculated by:

$$\overline{d} = \sqrt{\sum_{i=1}^{n} (p_i - c_i)^2 * G_i} \tag{14}$$

where p_i is the point coordinate, c_i is the centroid coordinate, G_i is the importance, or gain, of each feature and n is the total number of features. The contribution of each partially-active slip system is then calculated as:

$$\frac{\overline{d}_{non-active}}{\overline{d}_{non-active} + \overline{d}_{active}} \tag{15}$$

where $\overline{d}_{non-active}$ is the distance to the centroid of all the non-active points and $\overline{d}_{active}$ is the distance to the centroid of all the active points.

3.2.3. Determination of slip transmission parameters

Recently we reported that fatigue-induced voids introduced during cyclic loading below the macroscopic yield stress contributed to subsequent strength degradation in both pure α-iron and 7075-T6 aluminum alloy [35,36]. Additionally, we found that crack initiation in a polycrystalline titanium-aluminum alloy did not occur in regions with the largest localized strain, but instead at locations with significant barriers to slip transfer through a boundary [37]. Therefore, EBSD scans were collected around voids found in the fatigued material to investigate dislocation slip transmission across those grain boundaries. The details about EBSD scan set up and parameters are reported in Ref. [36]. The potential for slip transmission across a grain boundary can be evaluated using three parameters outlined by Lee, Robertson, and Birnbaum (LRB) [38,39]: the Schmid factor, the residual Burgers vector, and the angle, θ, formed as the angle of intersection between the incoming and outgoing slip planes in the grain boundary plane. These three parameters account for the driving force on each slip system in addition to the alignment of the two slip systems across a grain boundary. Following this definition dislocation slip transmission across a grain boundary is more likely to occur when the Schmid factor is maximized, the residual Burgers vector is minimized, and the angle θ is minimized.

Herein we calculated Schmid factors for each slip system from grain orientations obtained from EBSD. Equation (16) was used to calculate the residual Burgers vector, where $\mathbf{b_1}$ and $\mathbf{b_2}$ are the slip directions in grains one and two, respectively.

$$|\mathbf{b_r}| = |\mathbf{b_1} - \mathbf{b_2}| \tag{16}$$

Lastly, θ is calculated using Eq. (17) where $\mathbf{v_1}$ and $\mathbf{v_2}$ represent the vectors for the lines of intersection.

$$\cos\theta = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{|\mathbf{v_1}| \cdot |\mathbf{v_2}|} \tag{17}$$

It was assumed that the grain boundary plane was normal to the two-dimensional EBSD scan. Thus, the vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ were calculated using Eq. (18) where $\mathbf{n_i}$ and $\mathbf{n_{gb}}$ are the normal vectors to the slip plane

and grain boundary plane, respectively.

$$\mathbf{v_i} = \mathbf{n_i} \times \mathbf{n_{gb}} \tag{18}$$

To highlight the slip transmission parameters Fig. 11 presents an illustration of two grains, one containing a void, from an EBSD scan. The direction and plane of the slip system with the maximum Schmid factor and its intersection with the grain boundary plane is drawn for each grain. It can be seen that both $\mathbf{b_r}$ and θ go to zero as the slip systems become perfectly aligned, indicating a greater likelihood of dislocation slip transmission along those two slip systems across the grain boundary.

3.2.4. Fatigue-induced void analysis: α-iron

Orientations of grains with voids and their neighboring grains were obtained through EBSD. Schmid factors for each of the grain's slip systems were calculated from the orientation and global loading direction information. Slip systems selected for the slip transmission analysis were determined by whether or not the specific system was deemed active or partially-active based on the crystal plasticity modeling, spectral clustering, and KNN classification. Classification predictions with KNN were done using the Schmid factor and normalized Schmid factor inputs because these were the only two variables that could be calculated from the EBSD grain orientations.

An example EBSD scan of the α-iron at approximately 80% of the fatigue life is shown in Fig. 12a. It was observed that at this, and later, portions of the fatigue life additional voids were nucleated as a result of the fatigue loading [35]. The residual Burgers vector and θ parameter for slip transmission were calculated for combinations of partially and fully-active slip systems between each neighboring grain pair. The ten grains that were most likely to prohibit dislocation slip transfer based on having the highest combination of residual Burgers vector, θ, and contribution (as defined in Eq. (15)) were highlighted on the EBSD scans. Data at 80% and 95% of the fatigue life are shown in Fig. 13. It was found that in all six EBSD scans (three at 80% and three at 95% of the fatigue life) at least one grain adjacent to where the void formed had one of the highest combinations of slip transmission parameters. This finding implies that the restriction of dislocation slip across a grain boundary contributed to the void formation during fatigue loading. Void formation as a result of the lack of slip transmission could arises from (1) stress concentrations at the grain boundaries and the absence of other deformation mechanisms for the energy to relax, or (2) during reverse loading dislocation motion will occur in the opposite direction, but the glide plane is not necessarily perfectly reversed [40]. Thus, these dislocations cause an effective back stress on the dislocations that moved during the first half loading cycle and not all of these dislocations move back to their original location remaining pinned at the obstacle. While the inhibition of slip transfer appears to be related to void nucleation, it is not the only factor as evidenced by the multitude of grains with high slip transmission parameters and no adjacent void.

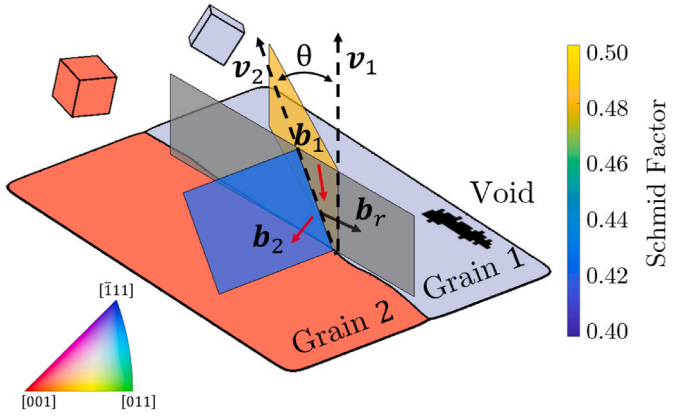


Fig. 11. Schematic showing the parameters used to assess likelihood of slip transmission across a grain boundary. The residual Burgers vector, $\mathbf{b_r}$, and the angle θ are used to determine the possibility of slip transmission across a grain boundary for a set of slip systems. Each grain's mean crystal orientation is denoted by the inverse pole figure (IPF) map.

3.2.5. Fatigue-induced void analysis: 7075-T6 aluminum alloy

Slip transmission parameters between one grain, which contained a void, and its neighbors were compared between fatigued states in the 7075-T6 aluminum alloy material. Voids in as-received material and material fatigued with zero mean stress to 75% of the fatigue life were both considered process-induced voids based on SEM image processing that showed similar percentages of overall voids in the material [36]. Voids from material fatigued with a tensile mean stress of 194 MPa to 75% of its fatigue life were considered fatigue-induced. An example EBSD scan is presented in Fig. 12b. Distribution of the slip transmission parameters is shown in Fig. 14 and reveals an approximately 40◦ shift in the peak of the θ parameter between the two void types. An increased value of θ implies that slip transmission was less likely to occur in those grain combinations suggesting that voids were nucleated during the fatigue due to accumulation of strain energy when dislocations were piling up at a grain boundary. These fatigue-induced voids were responsible for a subsequent 7% degradation of strength [36]. The overall distribution of slip transmission parameters differs from those originally reported by Indeck et al. [36] due to the implementation of spectral clustering as a non-subjective criterion to define slip activation.

4. Summary

Several different machine learning techniques were reviewed, with a goal of combining experiments, microscopy, and modeling to gain insight into specific problems concerning both mechanics and materials science. Spectral clustering was used to analyze and classify slip activity data predicted by crystal plasticity simulations as a function of orientation, which resulted in the classification groups of active, partially-active, and inactive. These groups were then used in conjunction with K-nearest neighbor classification to predict slip system activity within grains of interest in EBSD datasets. The likelihood of slip transmission was then calculated for grains containing fatigue-induced voids to see if low slip transmissivity correlated with void nucleation.

Two areas of the work are summarized in order to highlight the main observations and findings. First, are general considerations and thoughts on the machine learning techniques:

- Machine learning algorithms, especially shallow learning algorithms, can be readily implemented during data analysis to provide novel insights into mechanical behavior of materials. The machine learning techniques offered a more objective analysis approach in certain situations even though human input and knowledge about the data was still required.
- Unsupervised spectral clustering was used to determine how many groups the dataset should be divided into and what group a data point belongs to. Input parameters to the algorithm can be methodically chosen based on domain expertise to optimize the clustering output. Upon optimization, it was observed that including an additional partially-active slip system classification was more representative of the data groupings. The effect of a partially-active slip system was accounted for with the use of an appropriate weighting function.
- A nearest neighbor approach was a computationally cheap and effective way to classify unknown query points given an initial dataset. Satisfactory classifications were made even with a limited number of initial observations. Demonstration that even a small number of outputs from high-fidelity simulations, in conjunction with machine learning, can augment experimental results to provide additional physical insights is an important aspect of this work.

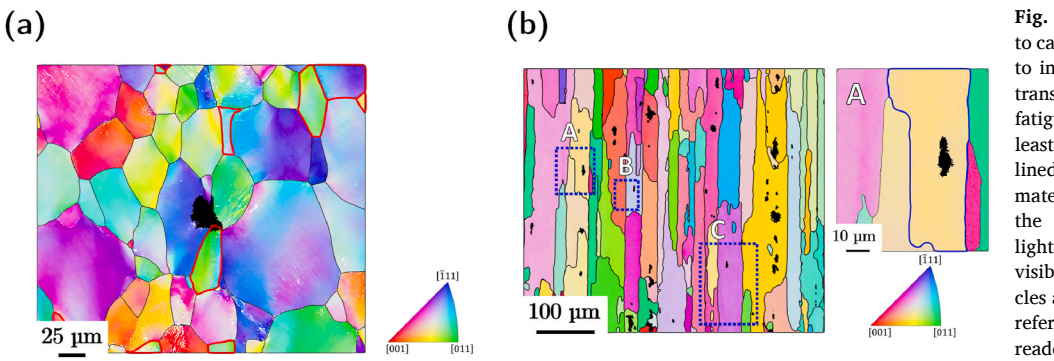


Fig. 12. Grain orientation from EBSD scans to calculate the residual Burgers vector and θ to investigate the relative likelihood of slip transmission. (a) α-iron material with a fatigue-induced void. The ten grains with the least likelihood of slip transmission are outlined in red. (b) 7075-T6 aluminum alloy material. An inset of region A is shown on the right with the grain boundary highlighted. Some of the larger black areas visible in the EBSD scan are inclusion particles and not voids. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

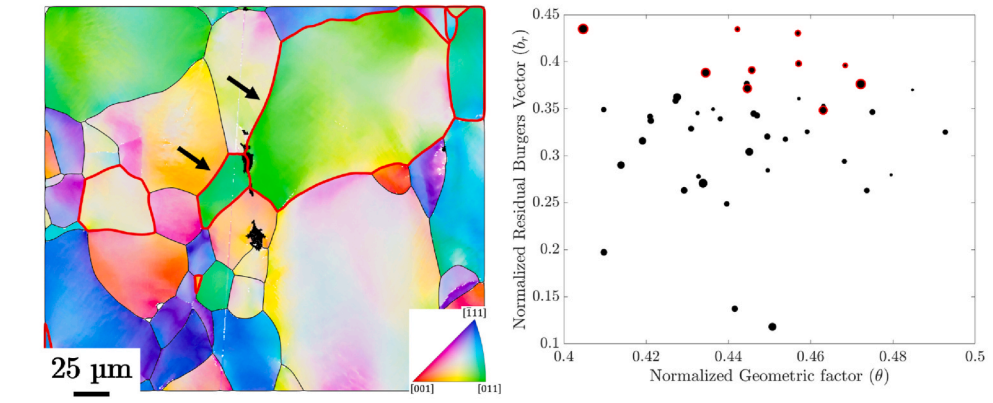
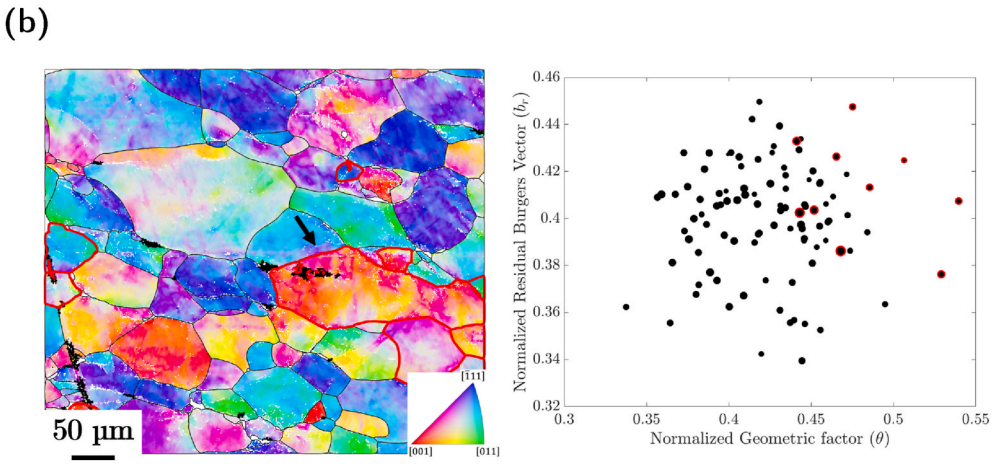


Fig. 13. Electron backscatter diffraction scan and slip transmission scatter plot for α-iron at (a) 80% and (b) 95% of the fatigue life. The ten grains with the greatest probability to prohibit dislocation slip across its grain boundary are outlined in red. Outlined grains adjacent to fatigue-induced voids are pointed out with arrows for clarity. The scatter plots show the slip transmission parameters for each grain. Again, the top grains are outlined in red and the size of the marker correlates to its contribution. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)



Second, are the material specific observations gleaned from the application of the machine learning techniques:

- Slip activity classification showed the extensive differences in slip behavior between FCC and BCC metals, due to complexities associated with non-Schmid effects in the latter. The FCC material was relatively simple in the sense that an increase in Schmid factor correlated to an increase in accumulated slip that ultimately determined its level of activity. Furthermore, each FCC grain had either one dominant fully-active slip system or two partially-active slip systems to accommodate the deformation. In contrast, the BCC material had a multitude of slip systems that had high accumulated slip values with low Schmid factors and vice versa. In addition, each grain had multiple combinations of partially and fully-active slip systems.
- In both pure α-iron and 7075-T6 aluminum alloy, fatigue-induced voids developed around grains with higher residual Burgers vectors and θ values that are less likely to promote slip transmission. The increase in voids/porosity was responsible for a decrease in quasi-static strength in both materials. While an increase in the slip transmission parameters by itself is not a sufficient condition to determine void nucleation, it does contribute to the understanding of microstructural mechanisms responsible for fatigue damage.

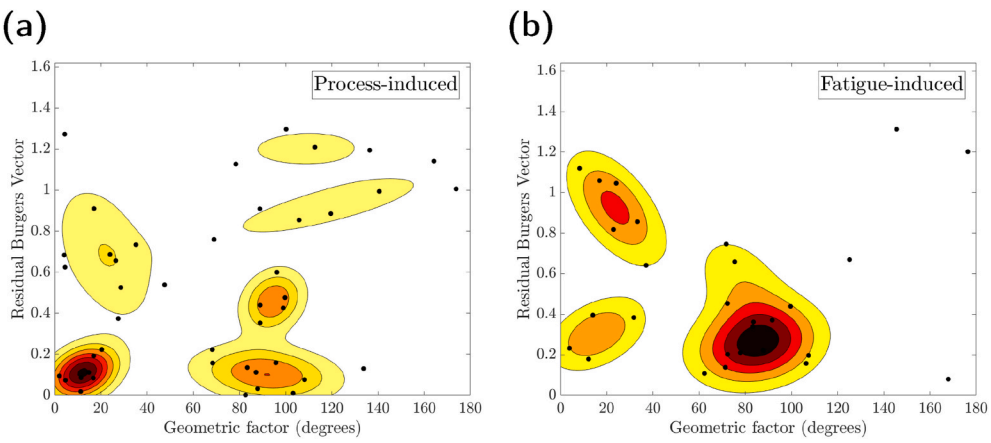


Fig. 14. Distribution density of the slip transmission parameters $\mathbf{b_r}$ and θ for (a) process-induced voids and (b) fatigue-induced voids. Process-induced voids were analyzed from both as-received material and material fatigued 75% of the fatigue life under with no mean stress (i.e. fully-reversible cyclic loading). Fatigue induced voids were those analyzed from material fatigued to 75% of the fatigue life under a 194 MPa tensile mean stress. An increase in the angle θ where the peak occurs can be seen between the two conditions.

While the machine learning techniques covered in this work are only a small fraction of available methods, they can be applied to a wide range of datasets. Furthermore, terms such as dimensionality reduction, clustering, classification, and regression are enough to lead other engineers and scientists to learn more about these, and other, techniques for their specific applications.

Data availability

The raw data required to reproduce these findings are available to download from the link to be provided by contacting the corresponding author: hazeli@arizona.edu.

Author contribution statement

Joseph Indeck: performing experiments, data analysis, machine learning, plotting, and manuscript writing. **David Cereceda:** machine learning, data interpretation, and section writing. **Jason Mayeur:** crystal plasticity, data interpretation, and section writing. **Kavan Hazeli:** providing academic support to J.I, design experiments, and section writing, methodology, data interpretation, writing, and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by Dynetics (contract number: PO AL014663) and the United States Army Space and Missile Defense Command under contract number: W9113M-18-C-0004. The authors would like to specially thank Mark Fisher (Dynetics), Shawn Finnegan (Dynetics), and James M. White (Army/SMDC) for their support during the presented investigation.

Appendix A. Details on k-fold cross-validation methods

Cross-validation is a technique that uses only the training dataset to optimize input parameters into a model (such as K in KNN). It works by removing a subset of data from the training dataset, then creating the classifier, and finally testing the predictive capability on the subset that was removed. For example, if a training dataset has n number of total data points and p data points were removed, the classifier would be created using the $n - p$ dataset, and finally tested on the removed p points. Scoring metrics for the classifier's performance are calculated by comparing the predicted classification of the p points to the actual classification as a function of K. The K value that results in the best performance score is chosen as the input parameter when classifying new data points. Two common performance metrics include the accuracy and $F1$ score. Accuracy is defined as the number of correct classifications divided by the total number of classifications. However, accuracy scoring for a dataset that has a large imbalance between two classes can be misleading [41]. A more appropriate metric in these situations is the $F1$ score. In a classification problem there are four outcomes: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Defined in Eq. (A.1), the $F1$ score is the harmonic mean of the two terms, precision and recall.

$$F1 = \frac{2(\text{Precision}\cdot\text{Recall})}{\text{Precision} + \text{Recall}} \tag{A.1}$$

where

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

There are three different types of $F1$ scores that can be calculated for a multi-classification problem, i.e. when instances must be classified in more than two groups. The first is the micro-$F1$ score, which is calculated from the total precision and recall values not accounting for different classification groups. The second type is the macro-$F1$ score, which is the average of each group's individual $F1$ score. For instance, with three groups: A, B, and C:

$$F1_{macro} = \frac{F1_A + F1_B + F1_C}{3}$$

The macro-$F1$ score weights each group the same and does not account for any class imbalance. The last type of $F1$ score is the weighted-$F1$ score which accounts for the number of observations in each group. For instance, if n is the total number of observations then $F1_{weighted}$ is calculated by:

$$F1_{weighted} = \frac{(F1_A \cdot n_A) + (F1_B \cdot n_B) + (F1_C \cdot n_C)}{n}$$

An extension of cross-validation is a method called k-fold cross-validation that breaks the original training dataset into k sets of approximately equal size. Each set then takes a turn being the tested dataset. Fig. A15 depicts the k-fold cross-validation process where an initial training dataset is divided into five different parts. Five different models are trained using the four parts colored in blue and tested using the one part colored in orange. A performance score is calculated for each of the five models and then averaged to end up with an overall score for the model. A value of k = 5 or k = 10 is commonly selected for k-fold cross-validation [42–44].

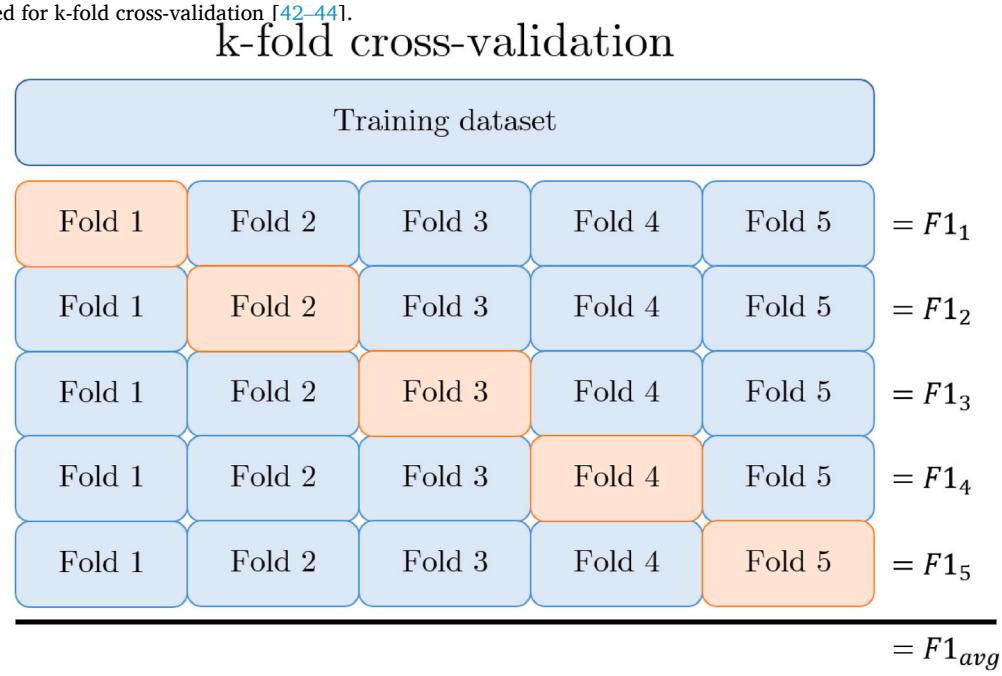


Fig. A.15. Depiction of data division in k-fold cross-validation for k = 5. The blue colored folds represent the new training dataset used to create the model and the orange colored folds represent the withheld test points. A performance metric can be calculated for each model and averaged to produce a total model score.

A further extension is repeated k-fold cross-validation (sometimes called Monte Carlo cross-validation) where the random assignment of data into each fold is repeated over multiple iterations. The k-fold cross-validation process depicted in Fig. A15 is repeated over a specified number of iterations, where each iteration performs a different random data assignment into each fold as shown in Fig. A16. An average of the model's performance metric (in this case the $F1$ score) can be calculated as previously described with respect to k-fold cross-validation. Increasing the number of iterations will decrease uncertainty in the estimated performance, but can substantially increase the computational requirement. The application of k-fold cross-validation and subsequent KNN classification to predict the likelihood of slip transmission across a grain boundary is demonstrated in Section 3.2.

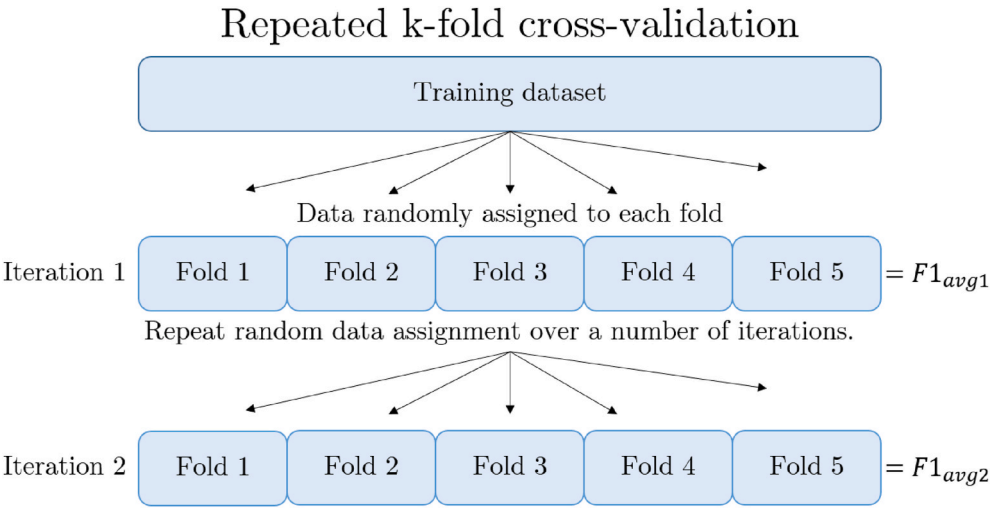


Fig. A.16. Portrayal of repeated k-fold cross-validation. Data from the original dataset is randomly assigned to the folds over multiple iterations.

A variation of the k-fold cross-validation process, referred to herein as inverse k-fold cross-validation, was used to test the model's performance when only a small portion of data was used during model creation. Fig. A17 shows how data is divided between the training and test datasets for inverse k-fold cross-validation. A single fold is used to create the model for inverse k-fold cross-validation and then k − 1 folds are tested to compute

the $F1$ scores. Contrast this to k-fold cross-validation where k − 1 folds are used to create the model and only one fold is tested.

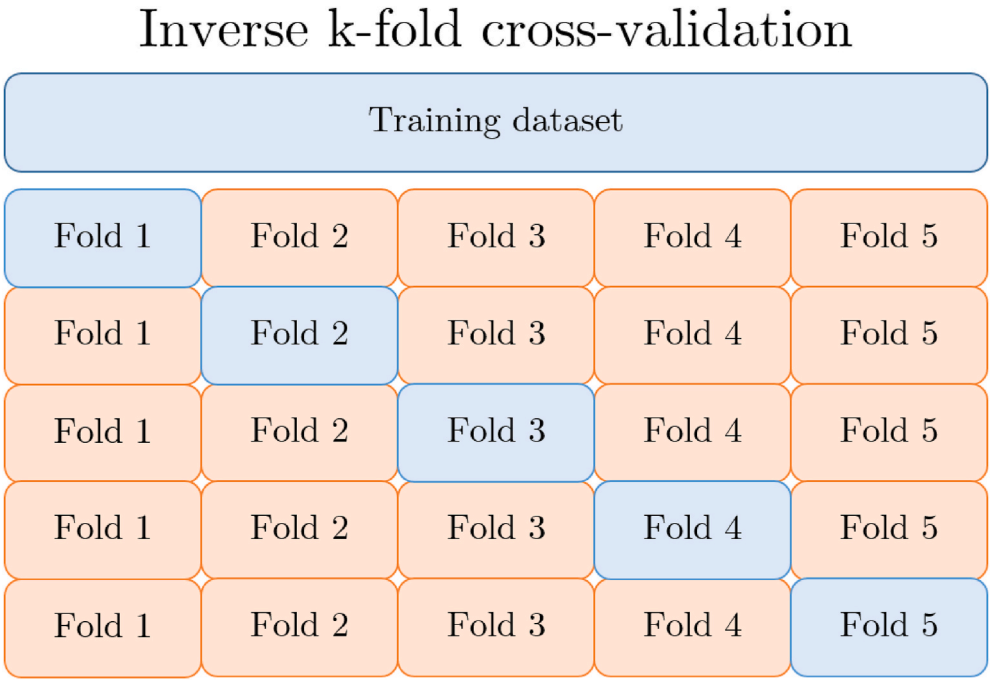


Fig. A.17. Depiction of data allocation for inverse k-fold cross-validation. Like Fig. A15, blue colored folds represent the new training dataset used to create the model and orange colored folds represent the withheld test points. Only a small portion of the original dataset is used to train the classifier model with inverse k-fold cross-validation.

A flowchart of the repeated inverse cross-validation process implemented in this work is sketched in Fig. A18. First, the overall training dataset is randomly divided into the specified number of k-folds. Next, a different classifier model is created with each single fold being used as the training dataset (see Fig. A17). Then, the $F1$ metrics are calculated using K-neighbors with the remaining data folds being used as query points. These steps, beginning with the random data assignment, are repeated for a number of iterations. Finally, the $F1$ scores were averaged resulting in a performance score for a number of k-folds as a function of K-neighbors. In this analysis, ten iterations were run for K = 1 to 30 nearest neighbors, using k = 2–50 folds.

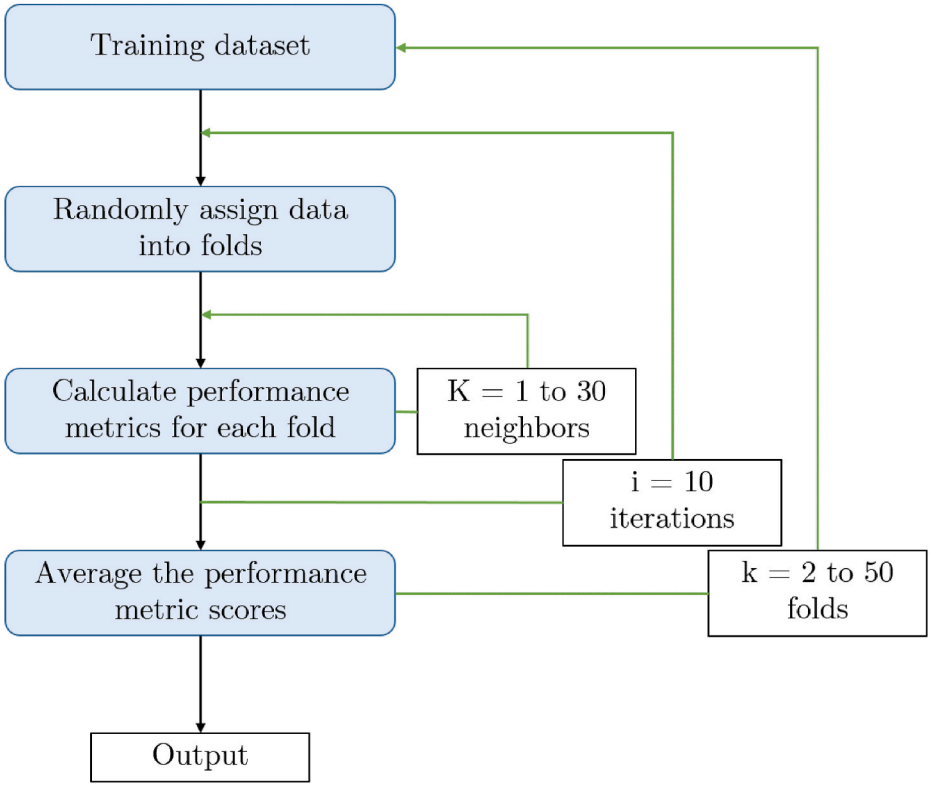


Fig. A.18. Flowchart of the inverse k-fold cross-validation process. The output is an $F1$ score for each number of k-folds, as a function of K-neighbors, averaged over a number of iterations.

Appendix B. Details on the calculation of gain scores

Gain quantifies how much a decision tree classifier improves when splitting the data into branches. Some incorrectly classified elements along a branch will subsequently be correctly classified with each new split. The amount of improvement "gained" at each decision node can be attributed to the feature used for the split. That amount of gain for each feature, at all decision nodes is summed and ordered to generate the feature importance.

The purpose of a supervised machine learning model is to make a prediction, $\hat{y}$, given a set of inputs, x_{ij}. One model example is a linear fit where the prediction is the sum of weighted input features:

$$\hat{y}_i = \Sigma_j \theta_j x_{ij}$$

where x_{ij} is the input data containing multiple features, and θ_j are the parameters (or coefficients) calculated by fitting the model to the input data. An objective function quantifies how well the model actually fits the input data. Objective functions consist of two parts, the training loss and regularization term:

$$\text{obj}(\theta) = L(\theta) + \Omega(\theta)$$

where L is the training loss and Ω is the regularization term. Training loss is a measure of how well the predictions are with respect to the input data. A common training loss function is the mean squared error and is given as:

$$L(\theta) = \sum_i (y_i - \hat{y}_i)^2$$

The regularization term regulates the complexity of the model by penalizing more complex models to prevent overfitting. Some techniques include L_1 regularization or LASSO (Least Absolute Shrinkage and Selection Operator) and L_2 regularization or Ridge regression. More details about these, and other, techniques can be found in Refs. [44–46].

Extreme Gradient Boosting (XGBoost) [31] is a decision tree ensemble model that sums the prediction of multiple decision trees together to improve the overall predictive capability of the model. The objective function for this type of model can be written as:

$$\text{obj}(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k)$$

where n is the number of data points, K is the number of total trees, and f_k is a function in the space of all possible trees. The prediction, $\hat{y}_i$, being a sum of multiple decision trees can be written as:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F}$$

where $\mathcal{F}$ is the set of all possible trees. Optimizing an objective function that consists of the structure and leaf scores of multiple decision trees is unmanageable to do all at once, instead leading to an additive training technique. Consider the same objective function as the decision tree ensemble model:

$$\text{obj} = \sum_i^n l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{i=1}^t \Omega(f_i)$$

where at each step t a new tree is added to the prediction value $\hat{y}_i^{(t)}$. Therefore, the continually updated prediction can be expanded as:

$$\begin{aligned}
\hat{y}_i^{(0)} &= 0 \\
\hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\
&\vdots \\
\hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)
\end{aligned}$$

If the mean squared error is used for the loss function, then at each step, t, the objective function becomes:

$$\begin{aligned}
\text{obj}^{(t)} &= \sum_{i=1}^n l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + C \\
&= \sum_{i=1}^n \left[y_i - \left(\hat{y}_i^{(t-1)} + f_t(x_i)\right)\right]^2 + \Omega(f_t) + C \\
&= \sum_{i=1}^n \left[2\left(\hat{y}_i^{(t-1)} - y_i\right) f_t(x_i) + f_t(x_i)^2\right] + \Omega(f_t) + C
\end{aligned}$$

where C is a constant. A more generalized form of the objective function uses a Taylor series, up to the second order term, to express the loss function:

$$\text{obj}^{(t)} = \sum_{i=1}^n \left[l\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) + C$$

where:

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right)$$
$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l\left(y_i, \hat{y}_i^{(t-1)}\right)$$

Eliminating the constants means the objective function can be written as:

$$\text{obj}^{(t)} = \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t)$$

The second part of the objective function, the regularization term, is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2$$

where T is the total number of leaves and w is the vector of scores from the decision tree leaves. An approximation of the objective function then becomes:

$$\text{obj}^{(t)} \approx \sum_{i=1}^{n}\left[g_i w_{q(x_i)} + \frac{1}{2}h_i w^2_{q(x_i)}\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2$$

where q is a function that assigns data points to their corresponding leaf. This expression of the objective function can be compressed into:

$$\text{obj}^{(t)} = \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}\left(H_j + \lambda\right)w_j^2\right] + \gamma T$$

by defining:

$$G_j = \Sigma_{i\in I_j} g_i$$
$$H_j = \Sigma_{i\in I_j} h_i$$

where:

$$I_j = \{i | q(x_i) = j\}$$

and is the set of data point indices assigned to the j^{th} leaf. The best obtainable objective function which quantifies how well a decision tree works is:

$$\text{obj} = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T$$

Consider a decision tree with only a single split and two leaves to illustrate the concept and calculation of gain. When a tree node (or leaf) is split in two branches the improvement it gains is defined as:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma \tag{B.1}$$

where the first term describes the score on the new left leaf, the second term the score on the new right leaf, the third term the score on the original leaf, and the fourth term the regularization (or penalty) of the split. The parameters λ and γ are inputs into the model and control the model's complexity to prevent overfitting.

The regularization constant, λ, determines the amount of penalty imposed by the regularization term in the objective function where $\lambda \geq 0$. If $\lambda = 0$ no penalty is imposed.

The term γ is the minimum loss required to create a new split in the decision tree. If the gain score of the new split (the first three terms in Eq. (B.1)) is not greater than γ, then there is nothing gained from adding the split and including it makes the model worse. Removing branches and leaves of a decision tree that do not meet a certain criterion, in this case a minimum gain, is one pruning technique.

References

[1] K.P. Murphy, Machine Learning: a Probabilistic Perspective, MIT press, 2012.
[2] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, vol. 1, MIT press Cambridge, 2016.
[3] K. Kourou, T.P. Exarchos, K.P. Exarchos, M.V. Karamouzis, D.I. Fotiadis, Machine learning applications in cancer prognosis and prediction, Comput. Struct. Biotechnol. J. 13 (2015) 8–17.
[4] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, et al., Deep speech 2: end-to-end speech recognition in English and Mandarin, in: International Conference on Machine Learning, PMLR, 2016, pp. 173–182.
[5] I.H. Witten, E. Frank, Data mining: practical machine learning tools and techniques with java implementations, Acm Sigmod Record 31 (1) (2002) 76–77.
[6] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, https://doi.org/10.1038/nature14539.
[7] A.D. Orme, I. Chelladurai, T.M. Rampton, D.T. Fullwood, A. Khosravani, M. P. Miles, et al., Insights into twinning in Mg AZ31: a combined EBSD and machine learning study, Comput. Mater. Sci. 124 (2016) 353–363, https://doi.org/10.1016/j.commatsci.2016.08.011.
[8] A. Agrawal, P.D. Deshpande, A. Cecen, G.P. Basavarsu, A.N. Choudhary, S. R. Kalidindi, Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters, Integrating Mater. Manuf. Innov. 3 (1) (2014) 90–108, https://doi.org/10.1186/2193-9772-3-83.
[9] H. Fujii, D.J.C. Mackay, H.K.D.H. Bhadeshia, Bayesian neural network analysis of fatigue crack growth rate in nickel base superalloys, ISIJ Int. 36 (11) (1996) 1373–1382, https://doi.org/10.2355/isijinternational.36.1373.
[10] S.B. Singh, H.K.D.H. Bhadeshia, D.J.C. MacKay, H. Carey, I. Martin, Neural network analysis of steel plate processing, Ironmak. Steelmak. 25 (1998) 355–365.
[11] R.C. Atwood, A.J. Bodey, S.W.T. Price, M. Basham, M. Drakopoulos, A high-throughput system for high-quality tomographic reconstruction of large datasets at

Diamond Light Source, Phil. Trans. Math. Phys. Eng. Sci. 373 (2043) (2015) 20140398, https://doi.org/10.1098/rsta.2014.0398.
[12] J.J. de Pablo, B. Jones, C.L. Kovacs, V. Ozolins, A.P. Ramirez, The Materials Genome Initiative, the interplay of experiment, theory and computation, Curr. Opin. Solid State Mater. Sci. 18 (2) (2014) 99–117, https://doi.org/10.1016/j.cossms.2014.02.003.
[13] J.J. de Pablo, N.E. Jackson, M.A. Webb, L.Q. Chen, J.E. Moore, D. Morgan, et al., New frontiers for the materials genome initiative, npj Comput. Mater. 5 (1) (2019), https://doi.org/10.1038/s41524-019-0173-4.
[14] J. Schmidhuber, Deep learning in neural networks: an overview, Neural Network. 61 (2015) 85–117, https://doi.org/10.1016/j.neunet.2014.09.003.
[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (85) (2011) 2825–2830. URL, http://jmlr.org/papers/v12/pedregosa11a.html.
[16] L. Kaufman, P.J. Rousseeuw (Eds.), Finding Groups in Data, John Wiley & Sons, Inc., 1990, https://doi.org/10.1002/9780470316801.
[17] B.S. Everitt, S. Landau, M. Leese, D. Stahl, Cluster Analysis, John Wiley & Sons, 2011, ISBN 0470749911. URL, https://www.ebook.de/de/product/12064950/brian_s_everitt_sabine_landau_morven_leese_daniel_stahl_cluster_analysis.html.
[18] C. Hennig, M. Meila, F. Murtagh, R. Rocci (Eds.), Handbook of Cluster Analysis, Chapman and Hall/CRC, 2015, https://doi.org/10.1201/b19706.
[19] F. Chung, Spectral graph theory, in: Providence, R.I: Published for the Conference Board of the Mathematical Sciences by the, American Mathematical Society, 1997, ISBN 9780821803158.
[20] C. Aggarwal, C. Reddy, Data Clustering: Algorithms and Applications, CRC PR INC, 2013, ISBN 1466558210. URL, https://www.ebook.de/de/product/20369720/data_clustering_algorithms_and_applications.html.
[21] U. von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416, https://doi.org/10.1007/s11222-007-9033-z.
[22] B.W. Silverman, M.C. Jones, E. Fix, J.L. Hodges, An important contribution to nonparametric discriminant analysis and density estimation: commentary on Fix and Hodges (1951), Int. Stat. Rev./Rev. Int. Stat. 57 (3) (1951) 233, https://doi.org/10.2307/1403796, 1989.
[23] C.A. Bronkhorst, J.R. Mayeur, V. Livescu, R. Pokharel, D.W. Brown, G.T. Gray, Structural representation of additively manufactured 316L austenitic stainless steel, Int. J. Plast. 118 (2019) 70–86, https://doi.org/10.1016/j.ijplas.2019.01.012.
[24] A.S. Khan, J. Liu, J.W. Yoon, R. Nambori, Strain rate effect of high purity aluminum single crystals: experiments and simulations, Int. J. Plast. 67 (2015) 39–52, https://doi.org/10.1016/j.ijplas.2014.10.002.
[25] A. Patra, T. Zhu, D.L. McDowell, Constitutive equations for modeling non-Schmid effects in single crystal bcc-Fe at low and ambient temperatures, Int. J. Plast. 59 (2014) 1–14, https://doi.org/10.1016/j.ijplas.2014.03.016.
[26] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, in: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, 2001, pp. 849–856.
[27] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65, https://doi.org/10.1016/0377-0427(87)90125-7.
[28] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. (2) (1979) 224–227, https://doi.org/10.1109/tpami.1979.4766909. PAMI-1.
[29] A.C. Lewis, S.M. Qidwai, A.B. Geltmacher, Slip systems and initiation of plasticity in a body-centered-cubic titanium alloy, Metall. Mater. Trans. 41 (10) (2010) 2522–2531, https://doi.org/10.1007/s11661-010-0284-5.
[30] D. Raabe, Contribution of {123} $\lessbin 111 \gt $ slip systems to deformation of b.c.c. metals, Phys. Status Solidi 149 (2) (1995) 575–581, https://doi.org/10.1002/pssa.2211490208.
[31] Introduction to boosted trees, URL, https://xgboost.readthedocs.io/en/latest/tutorials/model.html, 2020. (Accessed 14 April 2021).
[32] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 785–794, https://doi.org/10.1145/2939672.2939785.
[33] S. Narayanan, D.L. McDowell, T. Zhu, Crystal plasticity model for BCC iron atomistically informed by kinetics of correlated kinkpair nucleation on screw dislocation, J. Mech. Phys. Solid. 65 (2014) 54–68, https://doi.org/10.1016/j.jmps.2014.01.004.
[34] D. Cereceda, M. Diehl, F. Roters, D. Raabe, J.M. Perlado, J. Marian, Unraveling the temperature dependence of the yield strength in single-crystal tungsten using atomistically-informed crystal plasticity calculations, Int. J. Plast. 78 (2016) 242–265, https://doi.org/10.1016/j.ijplas.2015.09.002.
[35] J. Indeck, J. Cuadra, C. Williams, K. Hazeli, Accumulation and evolution of elastically induced defects under cyclic loading: quantification and subsequent properties, Int. J. Fatig. 127 (2019) 522–536, https://doi.org/10.1016/j.ijfatigue.2019.05.025.
[36] J. Indeck, G. Demeneghi, J. Mayeur, C. Williams, K. Hazeli, Influence of reversible and non-reversible fatigue on the microstructure and mechanical property evolution of 7075-T6 aluminum alloy, Int. J. Fatig. 145 (2021) 106094, https://doi.org/10.1016/j.ijfatigue.2020.106094.
[37] T.R. Bieler, P. Eisenlohr, F. Roters, D. Kumar, D.E. Mason, M.A. Crimp, et al., The role of heterogeneous deformation on damage nucleation at grain boundaries in single phase metals, Int. J. Plast. 25 (9) (2009) 1655–1683, https://doi.org/10.1016/j.ijplas.2008.09.002.
[38] T.C. Lee, I.M. Robertson, H.K. Birnbaum, Anomalous slip in an FCC system, Ultramicroscopy 29 (1–4) (1989) 212–216, https://doi.org/10.1016/0304-3991(89)90248-9.
[39] T.C. Lee, I.M. Robertson, H.K. Birnbaum, An in Situ transmission electron microscope deformation study of the slip transfer mechanisms in metals, Metall. Trans. A 21 (9) (1990) 2437–2447, https://doi.org/10.1007/bf02646988.
[40] U. Essmann, Irreversibility of cyclic slip in persistent slip bands of fatigued pure fcc metals, Philos. Mag. A 45 (1) (1982) 171–190.
[41] S. Cohen, The evolution of machine learning: past, present, and future, in: Artificial Intelligence and Deep Learning in Pathology, Elsevier, 2021, pp. 1–12, https://doi.org/10.1016/b978-0-323-67538-3.00001-4.
[42] A.M. Molinaro, R. Simon, R.M. Pfeiffer, Prediction error estimation: a comparison of resampling methods, Bioinformatics 21 (15) (2005) 3301–3307, https://doi.org/10.1093/bioinformatics/bti499.
[43] G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer, New York, 2013, https://doi.org/10.1007/978-1-4614-7138-7.
[44] M. Kuhn, K. Johnson, Applied Predictive Modeling, Springer New York, 2018, ISBN 1461468485. URL, https://www.ebook.de/de/product/20211095/kjell_johnson_max_kuhn_applied_predictive_modeling.html.
[45] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed., SPRINGER NATURE, 2009, ISBN 0387848576. URL, https://www.ebook.de/de/product/8023140/trevor_hastie_robert_tibshirani_jerome_friedman_the_elements_of_statistical_learning_data_mining_inference_and_prediction_second_edition.html.
[46] A. Muller, S. Guido, Introduction to Machine Learning with Python, O'Reilly UK Ltd., 2016, ISBN 1449369413. URL, https://www.ebook.de/de/product/23308778/sarah_guido_introduction_to_machine_learning_with_python.html.