

**PREDICTION OF MICROSTRUCTURALLY SMALL
FATIGUE-CRACK GROWTH USING DATA-DRIVEN
ANALYSIS AND MACHINE LEARNING**

by

Kyle David Pierson

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computing

School of Computing
The University of Utah
May 2019

Copyright © Kyle David Pierson 2019
All Rights Reserved

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of

Kyle David Pierson

has been approved by the following supervisory committee members:

<u>Ashley Spear</u> ,	Chair(s)	<u>14 August 2018</u> Date Approved
<u>P. Thomas Fletcher</u> ,	Member	<u>14 August 2018</u> Date Approved
<u>Srikumar Ramalingam</u> ,	Member	<u>14 August 2018</u> Date Approved

by Ross Whitaker , Chair/Dean of
the Department/College/School of Computing
and by David B. Kieda , Dean of The Graduate School.

ABSTRACT

Predicting the growth behavior of microstructurally small fatigue cracks is a practically relevant problem in the materials-science and structural-engineering communities, especially for applications where the useful life of a structural component is almost entirely governed by the evolution of cracks that are as small as the microstructural features of the material. However, understanding and predicting such complex behavior remains an open area of research. The challenge of discovering the underlying rules that govern crack propagation at small length scales and in three dimensions is well-suited to formulation as a machine learning problem. The complication then becomes defining what is meant by predicting crack growth, and choosing appropriate metrics for evaluation. Once the problem is formulated, further decisions include extracting the relevant features (or alternatively, selecting a meaningful representation from which the model can learn the relevant features), selecting appropriate response variables, and choosing a model for learning. This thesis explores various machine learning approaches to the aforementioned crack-growth problem. Data were previously acquired through cyclic loading of an aluminum alloy sample, which was imaged using scanning electron microscope imaging, X-ray computed tomography, and X-ray diffraction microscopy. Then, the data were reconstructed and digitized. In this work, an attempt is made to select features from this digitized dataset based on prior domain knowledge of crack growth. Support vector regression (SVR) is applied to the time series of discretized crack fronts, using the selected features for each point to predict its $\frac{da}{dN}$ value. Next, the experimental data are combined with simulation data to produce additional features. These features are correlated with the crack surface to determine which ones are most influential in the growth of the crack. These correlations help construct a representation of the data that can be used in a wider selection of machine learning models. Specific approaches that are explored include using random forests and convolutional neural networks (CNN) to predict the positions and growth rates of points on the final crack surface using the surrounding microstructural representation.

CONTENTS

ABSTRACT	iii
LIST OF FIGURES	vi
LIST OF TABLES	ix
NOTATION AND SYMBOLS	x
CHAPTERS	
1. INTRODUCTION	1
1.1 Machine learning models	2
1.1.1 Support vector machine	2
1.1.2 Support vector regression	4
1.1.3 Artificial neural networks	5
1.1.4 Convolutional neural networks	6
1.1.5 Ensemble methods	6
1.1.6 Decision trees	6
1.1.7 Random forests	8
1.2 Supplementary terms and procedures	8
1.2.1 Principal component analysis	8
1.2.2 Correlation	9
1.2.3 Cross-validation	9
1.2.4 Error metrics	10
1.3 Machine learning approach	10
1.4 References	11
2. CRACK FRONT PREDICTION UNDER CYCLIC LOADING	13
2.1 Data acquisition via experimentation	13
2.2 Feature extraction	14
2.3 Model training	16
2.4 Leave-one-front-out (LOFO)	17
2.5 Leave-one-wedge-out (LOWO)	17
2.6 Results	18
2.7 Conclusion	18
2.8 References	19
3. DATA-DRIVEN CORRELATION ANALYSIS BETWEEN OBSERVED 3D FATIGUE-CRACK PATH AND COMPUTED FIELDS FROM HIGH-FIDELITY, CRYSTAL-PLASTICITY, FINITE-ELEMENT SIMULATIONS	20
3.1 Introduction	21
3.2 Materials and methods	23

3.2.1	Experimental measurements and mesh generation from prior work	23
3.2.2	Numerical simulation of cyclic loading applied to uncracked specimen	24
3.2.3	Convergence of cyclic field variables	27
3.2.4	Correlation analysis	27
3.3	Results and discussion	30
3.4	Conclusions	39
3.5	Acknowledgements	40
3.6	References	40
4.	INFERENCE OF POSITION AND GROWTH RATE FOR ALL POINTS ON THE CRACK SURFACE	44
4.1	Data preprocessing	44
4.2	General strategy	46
4.3	Crack surface height inference	49
4.4	Crack-surface height inference based on a radial prediction strategy	50
4.5	Conclusion	53
4.6	References	54
TOPIC INDEX	56	

LIST OF FIGURES

2.1	Microstructure and crack fronts (a): All crack fronts superimposed on bottom half of microstructure. (b): Crack ID for all points on and between all crack fronts. (c): Crack ID for all points on all crack fronts	15
2.2	Projected crack-front positions based on average crack-growth rates applied over 10,000-cycle intervals. Left: visualization of the predicted crack fronts using a random split of training and testing data. Right: visualization of the predicted 6th crack front using a LOFO split of training and testing data.	18
3.1	Post mortem reconstructions from an Al-Mg-Si alloy fatigue specimen based on near-field HEDM. (a). Approximation of uncracked volume (b) and reconstructed fatigue-failure surface from X-ray CT (c). The reference coordinate system is shown on the uncracked microstructural volume. Adapted with permission from Refs. [24] and [26].	24
3.2	Concurrent multi-scale finite-element mesh and applied boundary conditions. The reference coordinate system is shown on the uncracked microstructural volume. Adapted with permission from Ref. [24].	24
3.3	Convergence of $\Delta\sigma_{zz}$ and $\Delta^2\sigma_{zz}$ (MPa) during cyclic loading.	27
3.4	Cyclic micromechanical Taylor factor computed for the uncracked polycrystal, shown at a particular slice through the volume. Superimposed is the trace of the actual crack surface from X-ray CT imaging. Also superimposed is the grid used for correlation analysis. The neighborhood of influence is defined by the distance L above and below the crack surface.	29
3.5	Colored data points within L vertical distance of the crack surface that meet the spatial-gradient threshold for inclusion in the correlation analysis of ϵ_{vm}	30
3.6	Data-extraction algorithm.	30
3.7	Grid data showing strain-tensor components computed at the peak load of the fifth loading cycle for an uncracked microstructure in a multi-scale finite-element simulation.	31
3.8	Grid data showing cyclic values of the strain-tensor components computed for an uncracked microstructure in a multi-scale finite-element simulation.	31
3.9	Grid data showing von-Mises and maximum-principal strains computed at the peak load of the fifth loading cycle (left) and the corresponding cyclic values (right) for an uncracked microstructure in a multi-scale finite-element simulation.	32
3.10	Grid data showing stress-tensor components (MPa) computed at the peak load of the fifth loading cycle for an uncracked microstructure in a multi-scale finite-element simulation.	33

3.11	Grid data showing cyclic values of the stress-tensor components (MPa) computed for an uncracked microstructure in a multi-scale finite-element simulation.	33
3.12	Grid data showing von-Mises and maximum-principal stresses (MPa) computed at the peak load of the fifth loading cycle (left) and the corresponding cyclic values (right) for an uncracked microstructure in a multi-scale finite-element simulation.	34
3.13	Grid data showing slip-based damage metrics computed at the peak load of the fifth loading cycle for an uncracked microstructure in a multi-scale finite-element simulation.	35
3.14	Grid data showing cyclic values of the slip-based damage metrics computed for an uncracked microstructure in a multi-scale finite-element simulation.	35
3.15	Correlation coefficients computed between the following metrics and distance to crack surface: (a) field variables, λ ; (b) cyclic change in field variables, $\Delta\lambda$; (c) spatial gradient of field variables, $\nabla(\lambda)$; (d) spatial gradient of cyclic field variables, $\nabla(\Delta\lambda)$	36
3.16	Correlation coefficients computed between the following metrics and distance to hypothetical crack surfaces located $125 \mu m$ below (left bars) and $125 \mu m$ above (right bars) the known crack surface: (a) field variables, λ ; (b) cyclic change in field variables, $\Delta\lambda$; (c) spatial gradient of field variables, $\nabla(\lambda)$; (d) spatial gradient of cyclic field variables, $\nabla(\Delta\lambda)$	37
3.17	Correlation coefficients computed between each variable and distance to z-normal planes located $200 \mu m$ above the bottom face (left bars) and $200 \mu m$ below the top face (right bars) of the microstructural domain.	38
4.1	The blue bars represent the fraction of variance explained by each of the first five principal components. The red is the cumulative amount of variance explained by the first five principal components, interpolated between the bars to form a smooth curve	46
4.2	Histogram of the distribution of z-offset values	48
4.3	Structure of the convolutional neural network used in this chapter	49
4.4	Actual and predicted crack surfaces (a) Actual crack surface (b) Hypothetical crack surface (c) CNN prediction, using PCA vectors and a $10 \times 10 \times 10 \mu m$ region (trained for 10 epochs) (d) CNN prediction, using PCA vectors and a $10 \times 10 \times 10 \mu m$ region (trained for 5 epochs) (e) CNN prediction, using Euler angles and a $10 \times 10 \times 10 \mu m$ region (trained for 10 epochs) (f) XGBoost prediction, using PCA vectors and a $10 \times 10 \times 10 \mu m$ region (g) XGBoost prediction, using PCA vectors and a $6 \times 6 \times 6 \mu m$ region (h) XGBoost prediction, using Euler angles and a $10 \times 10 \times 10 \mu m$ region	51
4.5	Visualization of the radial prediction strategy. On the left is shown the points for which the predicted z-values of a single point correspond. On the right is the desired effect of this prediction strategy.	52

4.6	Actual and predicted radial crack surfaces (a) Actual crack surface. (b) XGBoost prediction, using a radial propagation strategy (c) XGBoost prediction, using a wedge propagation strategy (d) SVR prediction, using a radial propagation strategy (e) SVR prediction, using a wedge propagation strategy Each of these predictions have been cropped to the shape of the measured crack surface inside of the final crack front.	53
-----	---	----

LIST OF TABLES

2.1	Feature descriptions	15
2.2	Label descriptions	16
2.3	Error metrics for all three data splitting strategies of the crack fronts.....	17
3.1	Calibrated material parameters.....	25
4.1	Crack surface height inference results for each chosen model.....	50
4.2	Crack surface height inference results for each chosen model, using a radial propagation strategy	52

NOTATION AND SYMBOLS

w	Weight vector
γ	Margin of a classifier
ζ	Slack variables
σ	Standard deviation, sigmoid function
σ	Stress tensor
ϵ	Error
ϵ	Strain tensor
M	Micromechanical Taylor factor
D	Damage metric
λ	Field variable of the microstructure
C	Cost
η	Learning rate
J	Loss function
ρ	Pearson moment-product correlation
B	Angle of crack growth
d_{GB}	Distance to grain boundary

CHAPTER 1

INTRODUCTION

Predicting how a crack propagates in three dimensions at the scale of the material microstructure is a challenging and open-ended problem. The abstract notion of crack propagation could be decomposed of any number of concrete variables: the rate at which a particular point at the crack front is growing at a particular time and in a particular direction, the position of the next crack front measured at the start of the next cyclic-loading interval, the positions that a point on the crack front is expected to occupy between loading intervals, etc. The intricacies of each of these variables are too complex to solve in the scope of this thesis. Instead, only a few are considered, which are thought to be meaningful and relevant. The goal, then, is to make accurate predictions of the chosen response variable using some knowledge about how the crack has previously grown, and the values of the material's properties that are thought to affect the response. This approach is a form of machine learning, a field of study that has gained popularity in a variety of disciplines.

Many attempts have been made to fuse machine learning with materials science. In Ref. [1], an overall strategy for incorporating data-driven modeling with materials science was presented. Aspects of this strategy included data storage, management, and sharing, as well as formatting data as process-structure-property (PSP) linkages. These linkages provided relationships between the process parameters, the material structure, and the material properties, with the goal of being able to predict structure from process and properties from structure. Kalidindi then built on his work in Ref. [2] by outlining how this strategy could improve materials innovation and manufacturing.

Several concrete problems have also been approached with machine learning in mind. In Ref. [3], the authors approached the problem of predicting the fatigue strength of steel based on the samples' time and temperature of the diffusion, carburizing, hardening, and tempering phases, among other composition and processing parameters. They

experimented with models ranging from polynomial regression to neural networks, and were able to achieve R^2 values of close to 0.98. Fatigue crack growth calculation was addressed specifically in Ref. [4], where specialized neural networks were used to predict the relationship between stress intensity factor range (ΔK) and crack growth rate (da/dN). In Ref. [5], a more refined technique called relevance vector machine (RVM) was applied to the problem of predicting the remaining useful life (RUL) of a safety-critical component. Recently, Spear et al. [6] provided an overview of the challenges and opportunities that exist for leveraging machine learning to address problems associated with fatigue of materials.

Numerous machine learning models exist for predicting a response based on prior experience. All of them operate under the same premise: given a known dataset, the model uses some subset of the data to train itself to provide predictions that are consistent with the training data. Typically, a single variable from the dataset is chosen as the value to be predicted, but multiple variables can also be chosen. Then, the rest of the data are used to test the accuracy of the model's predictions against the known values of the response variable in the test set. If the response variable is categorical, then the problem is known as a classification problem, and accuracy can be measured either by taking the number of times the prediction matched the ground truth as a fraction of the total number of test samples, or by using a more sophisticated metric such as F1 score. If the response variable is a continuous, real number, then the problem is known as a regression problem, and accuracy can be measured using a metric such as root mean squared error (RMSE) or R^2 .

1.1 Machine learning models

In an attempt to make this thesis reasonably self-contained, a brief survey of relevant machine learning algorithms is presented. The objective of this chapter is to familiarize the reader with the verbiage and notation used throughout the rest of this thesis.

1.1.1 Support vector machine

One of the most intuitive machine learning methods is the support vector machine (SVM) [7]. The concept is simple: given a binary classification problem (one in which the response variable takes one of two values), find the classifier that best separates the data

points into their respective classes. Here, the best classifier is the one that maximizes the margin between the separating hyperplane and the nearest data point of either class. Given a linear threshold unit (LTU) defined by $w^T x + b$, the margin is given by Equation 1.1 [8].

$$\gamma = \min_{x_i, y_i} \frac{y_i(w^T x_i + b)}{\|w\|} \quad (1.1)$$

where each (x_i, y_i) is a point-label pair from the training data. If a constraint is set such that $y_i(w^T x_i + b) \geq 1$, then maximizing this margin is the same as maximizing $\frac{1}{\|w\|}$, which is the same as minimizing $\|w\|^2$. This produces a classic linear programming problem, as shown in Equation 1.2 [8].

$$\begin{aligned} & \min_w \|w\|^2, \\ & \text{s.t. } \forall i, y_i w^T x_i \geq 1 \end{aligned} \quad (1.2)$$

This can be reformatted into a single optimization problem. First, some lenience needs to be allowed, since the problem is likely not linearly separable. This is done by the introduction of slack variables ζ_i , which relax the constraint that requires $y_i w^T x_i$ to be greater than 1 [7]. However, relaxing the constraints should come at a cost, and so a term is added to the optimization to ensure that not too much slack is allowed. The linear programming problem now takes the form of Equation 1.3.

$$\begin{aligned} & \min_w \|w\|^2 + C \sum_i \zeta_i, \\ & \text{s.t. } \forall i, y_i w^T x_i \geq 1 - \zeta_i, \\ & \quad \forall i, \zeta_i \geq 0 \end{aligned} \quad (1.3)$$

The constraints can then be merged into Equation 1.4. A constant of $\frac{1}{2}$ is inserted to simplify the future gradient calculation.

$$\min_w \frac{1}{2} \|w\|^2 + C \sum_i \max(0, 1 - y_i w^T x_i) \quad (1.4)$$

It turns out that this optimization problem has now taken the form of a regularized loss minimization problem. Here, the first term is the regularizer on the weights, ensuring that as many of the weights are zero or near-zero as possible. This is desirable because it results in a simpler solution, since small weighted coefficients reduce the number of variables in the equation, and Occam's Razor states that simpler explanations are always

more desirable than complex ones. The latter term is the loss function known as the hinge loss, which quantifies the error encountered when classifying a data point. The hinge loss penalizes points that fall inside the margin, and the extent to which these points are penalized is controlled by the parameter C . This equation is convex, and thus can be solved efficiently via stochastic gradient descent (SGD), which is shown in Equation 1.5 [9].

$$\begin{aligned} w &:= w - \eta \nabla J, \\ \nabla J &= \begin{cases} w & \text{if } \max(0, 1 - y_i w^T x_i) = 0 \\ w - Cy_i x_i & \text{otherwise} \end{cases} \end{aligned} \quad (1.5)$$

Here, J is the equation we are trying to minimize, and η is the learning rate. For each point in the training data, the weight vector w is updated using this rule. This can be repeated over multiple iterations of the data until SGD has converged (or nearly converged).

However, even allowing for lenience does not solve the problem of linear inseparability. The key to solving this problem is the observation that linearly inseparable data can become linearly separable when mapped to a higher dimension. Unfortunately, this can become computationally expensive, especially when the data already exist in a high-dimensional space. However, it is not necessary to explicitly increase the dimensionality of the data. Instead, kernel functions can replace the traditional dot product that is used when making a prediction on a new data point [10]. This is known as the kernel trick, and allows SVM to learn nonlinear classifiers.

1.1.2 Support vector regression

The concepts behind SVM can also extend to regression problems. This time, a penalty is assigned to points that lie *outside* the margin γ , rather than inside [11]. The slack variables are split into ζ and ζ^* , which account for error on either side of the classifier. The primal form of the optimization problem is given by Equation 1.6.

$$\begin{aligned} \min_w \frac{1}{2} \|w\|^2 + C \sum_i (\zeta_i + \zeta_i^*), \\ \text{s.t. } \forall i, \quad y_i - w^T x_i \leq \gamma + \zeta_i, \\ \forall i, \quad w^T x_i - y_i \leq \gamma + \zeta_i^*, \\ \forall i, \quad \zeta_i, \zeta_i^* \geq 0 \end{aligned} \quad (1.6)$$

Again, kernels are typically employed, allowing for SVR to learn nonlinear functions.

1.1.3 Artificial neural networks

The neural network is by far the most recognizable of the models presented here, and the one that has gained the most traction in a wide variety of applications. The concept is based on the way neurons in the brain fire in biological neural networks. The structure is represented by a network of artificial neurons, which are connected to form a directed, acyclic graph. Each neuron takes a linear combination of its inputs with its own unique weights, passes the result through a nonlinear activation function, and sends the output to the neurons in the next layer of the network [12]. The input layer is the layer into which the raw input flows, and the output layer is the resulting prediction (which can be a single value or a vector of values). Any number of hidden layers can be added between the input and output layers. The architecture can be tweaked to fit the application, where the hyperparameters are the number of hidden layers, the number of neurons in each layer, the weight initialization strategy, the activation function, the regularizer, and the cost function. For traditional neural networks, a popular choice of activation function is the sigmoid, which is shown in Equation 1.7 for a given input z .

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (1.7)$$

Neural networks learn through an algorithm called backpropagation [13]. Essentially, whenever a mistake is made during training, the weight of each of the neurons is updated in a way that will make the final prediction more accurate on similar future inputs. A mistake is defined by the cost function, which will usually quantify the difference between the expected output and the actual output. An example of a cost function is the quadratic loss, which can be defined in Equation 1.8 for a single example x with label y and predicted label y' .

$$J = \frac{1}{2} ||y - y'||^2 \quad (1.8)$$

The updates can then be determined by taking the gradient of the cost function with respect to each of the weights. The explanation for how to do this is lengthy, and the reader is referred to [13] for more information. Once these gradients are known, then the weights can be updated during training using SGD.

1.1.4 Convolutional neural networks

The convolutional neural network (CNN) is a special type of neural network that addresses image-related problems [14]. The structure is similar to a traditional neural network, but the input is an entire image rather than a single data point. Instead of each neuron having a vector of weights to apply to its input, it has a small grid of values known as a filter. The filter is convolved with the input image to produce the output image, which is then passed on to the next layer. As before, an activation function is used to ensure that the network doesn't simply learn a linear function. The most common activation function for a CNN is the rectified linear unit (ReLU), which simply sets all negative values of the input to zero. Pooling layers are also included to downsample the image and reduce its dimensions. The filters, like weights, are learned through backpropagation.

1.1.5 Ensemble methods

Higher accuracy can often be achieved by using a collection, or ensemble, of classifiers. The final prediction is taken by using either the mode (classification) or the mean (regression) of all classifiers in the ensemble. The most popular implementation of this is the random forest, which are an ensemble of decision trees.

1.1.6 Decision trees

Decision trees are a family of classifiers based on the hierarchical data structure of a tree, i.e. a data structure described by a root node with some number of children, with each of those having some number of children, and so on [15]. In this case, the nodes (with the exception of leaf nodes) represent variables or features of the data, the branches represent all possible values (or ranges of values) those features can take, and the leaves are the class labels. Notice that this is much different from what the nodes represent in a neural network; a decision tree has no weights. Given a dataset, the tree is built by choosing features that best separate the examples into their respective classes. Separability is higher when certain values correspond with certain class labels, while other values correspond with the other class labels. Formally, this can be quantified using a heuristic such as entropy-based information gain. The information gain of a feature F with respect to a dataset D is defined in Equation 1.9.

$$Gain(D, F) = H(D) - \sum_{v \in Values(F)} \frac{|D_v|}{D} H(D_v) \quad (1.9)$$

where D_v represents the subset of examples having the value v for feature F . The formula for entropy is defined in Equation 1.10.

$$H(S) = \sum_{c \in C} -D_c \log(D_c) \quad (1.10)$$

where C is the set of all possible class labels, and D_c is the proportion of examples having the class label c .

Any time a branch results in a subset of data that is homogenous in a class label, a leaf node of that label is placed at the end of the branch. Using this strategy, the tree is built recursively in a greedy fashion [16]. First, the root is determined by the feature having the highest information gain. A branch for each value of this feature is added, extending from the root. Each child node is added by determining the next most informative feature, or by assigning a leaf node to the branch. This continues until all leaves are class labels, or until all features have been used. In the latter case, leaves must be assigned to all unresolved branches. This is typically done by taking the most common label of the subset of data that corresponds to each branch.

Prediction is done by taking the new data point and propagating it down the tree, following the path that defines the features of the data point. For example, given a data point x' , if the root of the tree is feature f , and the value of x' 's feature f is v , then the prediction would follow the branch of the root that corresponds to value v . If the child at the end of the branch is a leaf node, then the prediction is whatever class label is assigned to that leaf node. Otherwise, the node must define a feature, and the data point will then follow the next branch based on its value for that feature. This continues until a leaf node is reached, which by definition of the tree must be a class label. That label is the final prediction.

The benefit of using decision trees is that they can learn nonlinear functions relatively easily. However, due to the lack of weights, there is no function or formula produced as the resulting model. This can make it more difficult to infer meaning from the model with respect to the individual features.

1.1.7 Random forests

A random forest is simply an ensemble of decision trees. The collection is built using a strategy called bagging (short for bootstrap aggregating) [17]. Random samples of the data are drawn with replacement to form a subset. These data are used to build a single tree; however, when a new node is added, only a random subset of all available features are considered as candidate features. Other variations and optimizations exist, but the basic idea remains the same.

1.2 Supplementary terms and procedures

Statistics and linear algebra lie at the heart of machine learning, and there are many statistical techniques that can be used irrespective of the chosen model. A few of these model-agnostic procedures have been utilized and are defined here.

1.2.1 Principal component analysis

Principal component analysis (PCA) is a dimensionality-reduction strategy often used as a preprocessing procedure for many machine learning algorithms [18]. The idea is to find the components, which together can be thought of as an orthogonal set of basis vectors, along which the variance of the data are maximized. In a d -dimensional space, there are d principal components required to explain all of the variance in the data (assuming there are enough data points to sufficiently fill the space). These are typically ordered from greatest variance to least variance, where the first principal component describes the direction along which the data vary the most.

In most cases, almost all of the variance can be described by the first few principal components, sometimes called modes of variation. Thus, the dimensionality of the data can be reduced without the loss of too much information. This is done by projecting all data points onto the space formed by the first few principal components, which form an orthogonal basis. Given an n -dimensional data point x , and an $n \times m$ matrix A with the first m principal components as its columns (learned from the dataset), x can be projected onto the the space formed by A , resulting in a new m -dimensional data point x' . This is shown in Equation 1.11.

$$x' = A^T x, x \in R^n, x' \in R^m \quad (1.11)$$

In machine learning, PCA is a popular approach to feature transformation. It can take high-dimensional data of many (possibly correlated) variables, and reduce the dimensionality of the data down to a few linearly uncorrelated variables. This helps improve computation while also retaining the most important and most descriptive aspects of the data.

PCA is sensitive to the scaling of the variables, meaning it is important to normalize or standardize the data prior to performing PCA. One way to do this is by subtracting the mean and dividing by the standard deviation, ensuring that each feature has zero mean and unit variance.

1.2.2 Correlation

Correlation is a measure of the relationship between two variables. There are different methods of computing correlation, but the most common is the Pearson product-moment coefficient, which is calculated using Equation 1.12.

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad (1.12)$$

Correlation can be useful for examining the relationship between feature variables, as some models assume independence, and nonzero correlation implies dependence. It can also be used to get a sense of how well each of the individual feature variables measures the response variable. This is explored in Chapter 3, where a large-scale data-driven correlation analysis is performed to determine which features have a high correlation between their value at a given point and their proximity to the crack surface.

1.2.3 Cross-validation

Cross-validation is a technique that is used during training. The idea is to further split the training data into subsets, and then use some of those subsets as training data and some as validation data [19]. The validation data are not used to train the model, but rather to test the accuracy of the model. The difference between test data and validation data are that the results from validation data can safely be used to tweak the model, since it is technically still part of the original training data. For this reason, cross-validation is often used in hyperparameter selection.

A popular cross-validation technique is k -fold cross-validation. In this case, the training

data are split into k equally-sized subsets. Then, $k - 1$ of those subsets are used to train the model, and the last subset is used to test the model and produce an accuracy rating. Usually, the chosen model is trained and tested k times, with each of the k subsets used once as validation data. The mean of all k accuracy ratings can be used as an estimate for the true accuracy of the model. This can be done for multiple hyperparameter configurations, where the one with the best mean accuracy can be chosen as the best hyperparameter configuration.

1.2.4 Error metrics

Regression problems require a special metric for measuring performance, since classification accuracy cannot apply. There are many such metrics, and two are presented here. Any metric must take the form of a function that operates on the actual real-valued labels y , and the inferred labels \hat{y} . Each of these is a vector of size n , where n is the number of samples in the test set. The first metric is root-mean-squared-error (RMSE), which is calculated using Equation 1.13.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (1.13)$$

RMSE is a standard error measurement technique. It benefits from being in the same units as the predicted label. The problem of RMSE is that it can be difficult to interpret, and there is no definite standard on what values of RMSE equate to a good model. Another metric, R^2 , attempts to remedy these problems, and is calculated using Equation 1.14.

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (1.14)$$

Here, y_i is the actual value of a data point, \bar{y} is the mean of the actual data, and \hat{y}_i is the predicted value for the data point. The benefit of R^2 is that it more easily interpretable; values closer to “1” generally indicate a better fit. It also accounts for the natural variance of the data, and is essentially a measure of the proportion of total variance that is explained by the model.

1.3 Machine learning approach

The next few chapters attempt to apply these concepts to approximate the rules that govern 3D crack evolution. This is inspired by earlier work on training an artificial neural

network with data from high-fidelity fracture simulations to predict real-time residual strength [20]. In each case, the same general approach is followed.

1. Acquire data through experimentation and simulation.
2. Determine a quantifiable definition of crack growth, and use that as the response variable.
3. Discover and extract features that are relevant to predicting the response variable, either from previous experience or through systematic feature selection.
4. Train a model to learn a hypothesis for the response variable's governing equation using the observed features.
5. Ensure that the learned hypothesis can generalize well, with relatively low error under the chosen evaluation metric.

1.4 References

- [1] S. Kalidindi and M. D. Graef, "Materials data science: Current status and future outlook," *Annual Review of Materials Research*, vol. 45, no. 1, pp. 171 – 193, 2015.
- [2] S. R. Kalidindi, A. J. Medford, and D. L. McDowell, "Vision for data and informatics in the future materials innovation ecosystem," *JOM*, vol. 68, no. 8, pp. 2126–2137, 2016.
- [3] A. Agrawal, P. D. Deshpande, A. Cecen, G. P. Basavaraju, A. N. Choudhary, and S. R. Kalidindi, "Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters," *Integrating Materials and Manufacturing Innovation*, vol. 3, Mar 2014.
- [4] H. Wang, W. Zhang, F. Sun, and W. Zhang, "A comparison study of machine learning based algorithms for fatigue crack growth calculation," *Materials*, vol. 10, no. 5, p. 543, 2017.
- [5] E. Zio and F. D. Maio, "Fatigue crack growth estimation by relevance vector machine," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10681 – 10692, 2012.
- [6] A. Spear, S. Kalidindi, B. Meredig, A. Koutsos, and J.-B. le Graverend, "Data-driven materials investigations: The next frontier in understanding and predicting fatigue behavior," *JOM*, pp. 1–4, 2018.
- [7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, September 1995.
- [8] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT 92)*, pp. 144–152, July 1992.

- [9] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT2010*, pp. 177–186, 2010.
- [10] M. Hofmann, "Support vector machines - kernels and the kernel trick," *An elaboration for the Hauptseminar 'Reading club: Support vector machines'*, June 2006.
- [11] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1997.
- [12] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117.
- [13] D. Rumelhart, G. Hinton, and R. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, October 1986.
- [14] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," *International Workshop on Shape, Contour and Grouping in Computer Vision*, vol. 1681, pp. 319–345, Jan 1999.
- [15] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, March 1986.
- [16] B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali, "A comparative study of decision tree id3 and c4.5," *International journal of advanced computer science and applications*, pp. 13–19, 2014.
- [17] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, October 2001.
- [18] A. Tharwat, "Principal component analysis - a tutorial," *International Journal of Applied Pattern Recognition (IJAPR)*, vol. 3, no. 3, 2016.
- [19] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of database systems*, 2009.
- [20] A. Spear, A. Priest, M. Veilleux, A. Ingraffea, and J. Hochhalter, "Surrogate modeling of high-fidelity fracture simulations for real-time residual-strength predictions," *AIAA Journal*, vol. 49, no. 12, pp. 2770–2782, 2011.

CHAPTER 2

CRACK FRONT PREDICTION UNDER CYCLIC LOADING

The objective of this chapter is to provide an overview of the methods and procedures by which the data were acquired. It also outlines an initial machine learning approach taken with the data. After the data are acquired and converted to a usable format, features are selected and extracted from the data. These features are used in a simple SVR model to predict the $\frac{da}{dN}$ of each data point. The aim of the SVR is to see how effective a simple approach can be. It also provides insight into how complex the model needs to be, and to what degree the selected features are effective in predicting crack growth.

2.1 Data acquisition via experimentation

The dataset consisted of a finite-element mesh of (*a*) a sample of the microstructure and (*b*) the fracture surface, as well as a point cloud of the discretized crack fronts. These data were obtained through experimentation [1]. The propagation of a microstructurally small fatigue crack was observed in an aluminum alloy (Al-Mg-Si). The alloy originated as a cylindrical composite-overwrapped pressure vessel (COPV), from which a specimen was extracted. The location of the specimen was chosen based on electron backscatter diffraction data (EBSD), such that the crack would nucleate in an ideal position with minimal boundary effects. Electrical discharge machining (EDM) and milling were used to extract the specimen, create a notch, and polish the final gauge-region. The final specimen was 9.5×44.5 cm, with a maximum thickness of 1.75 cm and a minimum thickness of 0.4 cm at the midpoint of the notch.

Fatigue failure was induced via constant-amplitude cyclic loading in the z-direction, with a loading ratio of 0.5, a frequency of 10 Hz, and a maximum load of 1 kN [1]. Every 10,000 cycles, groups of either 4, 6, or 8 25-cycle load blocks were applied, which alternated between a loading ratio of 0.1 and 0.5. This would produce corresponding marker bands

on the crack surface, which would indicate the position of the crack front for each of the loading phases. Additionally, the specimen was removed from the fatigue test every 50,000 cycles to undergo scanning electron microscope (SEM) imaging, the data from which were used to complement and register the marker bands.

After the test had completed and the specimen had fractured completely, wire EDM was used to cut a 1 mm wide strip from both the top and bottom portions near the fracture surface [1]. A 3D reconstruction of this sample was achieved using X-ray computed tomography and near-field high-energy X-ray diffraction microscopy, the latter giving information about the crystallographic grain geometry and orientation. The crack fronts were approximated as best as possible using the marker bands and SEM data, along with splines to ensure continuity.

2.2 Feature extraction

Once the positions of the crack fronts had been determined, each crack front was digitized and sampled at a constant rate. This resulted in a discretized point-representation of each crack front, with fronts farther from the nucleation point containing more points (due to their increased lengths). This is shown in Figure 2.1. These points could then be converted into a dataframe, a data structure compatible with most machine learning algorithms. Each row of the dataframe would correspond to a point x on a crack front, and the columns would indicate the features and labels. The column values for all points were assessed by composing functions that would calculate the necessary fields for a given point x , and then mapping these functions over all points. The functions would typically query the surrounding raw microstructure data to compute the desired feature. For example, to compute the distance to grain boundary d_{GB} , a breadth-first search was performed starting at x and proceeding outward through the voxelized microstructure, until a grain ID was encountered that differed from the one to which x belonged. In the end, 7 features were used to fully describe a single point x . Table 2.1 shows the 7 features that were used in the machine learning model. All distances are in microns (μm).

The goal of the machine learning model was to learn a function of these inputs that best approximated the true unknown function that determined the response variable. In our case, there were two response variables of interest, β and $\frac{da}{dN}$, which could also be called

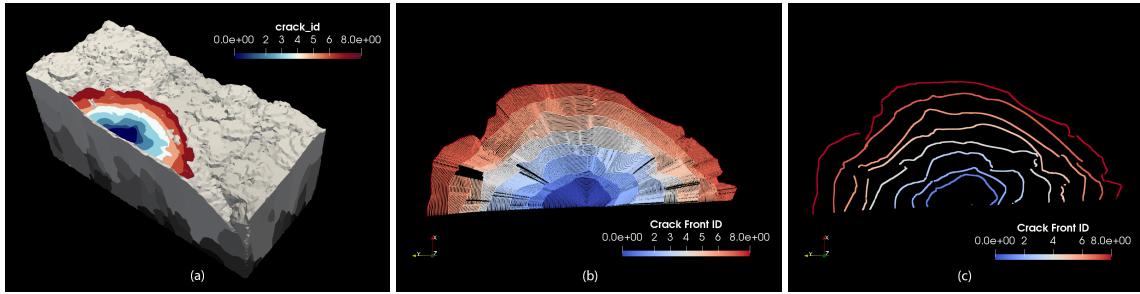


Figure 2.1: Microstructure and crack fronts (a): All crack fronts superimposed on bottom half of microstructure. (b): Crack ID for all points on and between all crack fronts. (c): Crack ID for all points on all crack fronts

Table 2.1: Feature descriptions

Feature	Description
Curvature distance	The difference between <ol style="list-style-type: none"> the distance from the nucleation point to x and the average distance from the nucleation point to the two points $5 \mu\text{m}$ away on either side of this point.
Crack normal to nearest grain boundary angle	The angle between the vector normal to the crack front at x and the vector to the nearest grain boundary.
Distance to average distance	The distance between x and the average distance from the nucleation point for the crack front to which x belongs.
Distance to grain boundary	Distance from x to the nearest grain boundary.
Misorientation	The difference between the Euler angles that describe the orientation of the grain in which x resides and those of the nearest grain.
Previous $\frac{da}{dN}$	The $\frac{da}{dN}$ of the closest corresponding point on the previous crack front. The closest corresponding point is the one that is closest to the ray extending from the nucleation point to x .
Signed Δ Taylor factor	The difference in Taylor factor between the grain in which x resides and that of the nearest grain.
Taylor factor	A number that describes the relationship between the orientation of the grain in which x resides and the direction that the grain is being pulled/stressed.

the labels. These labels are described in Table 2.2. These values were also calculated for each point x and included in the dataframe.

2.3 Model training

The resulting dataframe was loaded into a script written in the R programming language [2]. R was chosen for its rich library of built-in statistical analysis functions, as well as its CARET package [3], which includes an impressive collection of easy-to-use machine learning algorithms. The first attempted technique was SVR, using radial basis functions (RBF) as the kernel. In every case, the data were preprocessed through centering (removing the mean) and scaling (dividing by the standard deviation). The best values of the hyperparameters σ and C were determined using cross-validation over a grid search, i.e., for each $\sigma = 0.1, 0.2, 0.4$ and for each $C = 128, 256, 512$, the best combination was chosen as the one that minimized the cross-validation error metric. This metric was one of either $RMSE$ or R^2 , and both are given in the results. As a first pass, 80% of the data were used as training data (on which cross-validation was performed) and 20% was used as testing data. The data points were split randomly into one of these two sets, and the training data were used to build the model. For simplicity, only $\frac{da}{dN}$ was used as the response variable, meaning that the predictor would output a single real-valued number. After the model had been trained on all the training data using the hyperparameter combination, it was used to predict $\frac{da}{dN}$ for all points in the test set. These predicted values were compared against the known expected values to calculate the error. Surprisingly, this initial approach yielded an R^2 value of higher than 0.98, as shown in Table 2.3, which appeared to be quite good. There was, however, a problem with the way in which the data were split. Since the data

Table 2.2: Label descriptions

Label	Description
β	An angle describing the change in z for x . This is calculated by extending a ray from the nucleation point through x , stepping backward along this ray $10\mu m$ to a point a , and stepping forward along this ray $10\mu m$ to a point b . β is the angle between the line segment xa and the line segment xb .
$\frac{da}{dN}$	The distance between x and its closest corresponding point on the next crack front. The closest corresponding point is the one that is closest to the ray extending from the nucleation point through x .

Table 2.3: Error metrics for all three data splitting strategies of the crack fronts

Random split		Leave-one-front-out		Leave-one-wedge-out	
R^2	RMSE	R^2	RMSE	R^2	RMSE
0.984	0.000165	-2.877	0.00193	-2.052	0.00159

points were divided randomly, the test set happened to include many points that were spatially very close to points from the training set. Thus, the features of these test points highly resembled those of training points, along with similar values of $\frac{da}{dN}$. These values were much easier to predict, since the model had already seen comparable data points. The positive result was that this showed the ability of the model to learn complex functions, being able to mimic the expressive power of something like a 1-nearest neighbor classifier. The negative result was that this splitting approach is not useful for any sort of real-world scenario. A new approach was needed that better emulated the problem to be solved.

2.4 Leave-one-front-out (LOFO)

The ideal capability of the model would be to accurately predict future $\frac{da}{dN}$ given some knowledge about how the crack has grown previously, in addition to information about the microstructure. It would make sense to use this objective as inspiration by training on the earlier crack fronts and testing on the later crack fronts. The most intuitive strategy would be to use the points from the first $n - 1$ crack fronts as training data, and see how well the resulting model predicts the points on the n^{th} crack front. This meant essentially treating the data as a time-series of crack fronts. To get the most out of the single dataset, 9 different models were trained, each one leaving out one of the crack fronts during training. The final test accuracies of all 9 models were averaged together to achieve the mean accuracy.

2.5 Leave-one-wedge-out (LOWO)

In addition to splitting the data points by the crack front to which they belonged, a different strategy was used that split the data into wedges. Since the crack began at a single nucleation point and propagated outward radially, its shape was roughly semicircular. The justification for splitting this shape into wedges was that it provided the model with some points from all crack fronts, giving more context to the expected predictions for $\frac{da}{dN}$. It also could have some real-world applications, such as handling missing values of

measurements for areas with high uncertainty. The data points were split into 5 wedges of equal angular degree, and 5 models were trained to give a mean test accuracy.

2.6 Results

The resulting values for both error metrics are given in Table 2.3. Also included is Figure 2.2, which includes plots of the crack fronts points, with positions determined using only the actual and predicted values of $\frac{da}{dN}$, i.e. flattening the crack fronts for visualization.

2.7 Conclusion

It is clear that there is not enough predictive power in these 7 features alone to give accurate predictions of the crack growth rate, especially for such a small amount of data. Although growth-rate data are confined to these 9 crack fronts, it would be ideal to leverage more of the available data from this single experiment. Furthermore, feature selection, which was based on prior domain knowledge, could be improved. A model that could learn the relevant features given a relatively raw representation of the data might perform better than one that uses hand-picked features. The next two chapters focus on discovering this representation and using it effectively in an appropriate machine learning model.

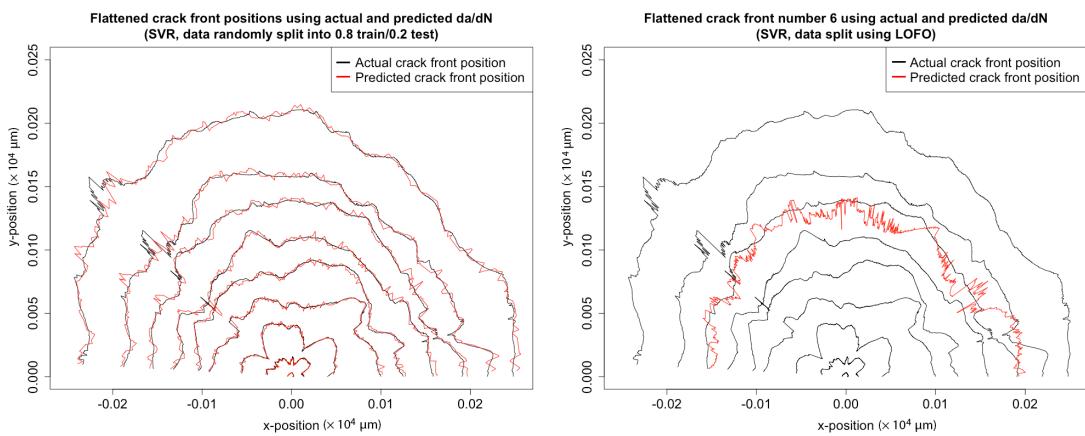


Figure 2.2: Projected crack-front positions based on average crack-growth rates applied over 10,000-cycle intervals. **Left:** visualization of the predicted crack fronts using a random split of training and testing data. **Right:** visualization of the predicted 6th crack front using a LOFO split of training and testing data.

2.8 References

- [1] A. Spear, S. F. Li, J. Lind, R. Suter, and A. Ingraffea, "Three-dimensional characterization of microstructurally small fatigue-crack evolution using quantitative fractography combined with post-mortem x-ray tomography and high-energy x-ray diffraction microscopy," *Acta Materialia*, vol. 76, pp. 413–424, 2014.
- [2] R. Statistical Package, "R: A language and environment for statistical computing," *Vienna, Austria: R Foundation for Statistical Computing*, 2009.
- [3] M. Kuhn, "Building predictive models in r using the CARET package," *Journal of Statistical Software*, vol. 28, no. 5, 2008.

CHAPTER 3

DATA-DRIVEN CORRELATION ANALYSIS BETWEEN OBSERVED 3D FATIGUE-CRACK PATH AND COMPUTED FIELDS FROM HIGH-FIDELITY, CRYSTAL-PLASTICITY, FINITE-ELEMENT SIMULATIONS

As described in the previous chapter, a better representation of the data is needed in order to facilitate a more exhaustive investigation of the features related to crack growth. The inspiration for this representation is the way in which images are typically depicted, namely by a 2-dimensional grid of pixels, where each pixel contains a vector of integers corresponding to an element of a color space. The motivation for using images as an inspiration is based on the ability of convolutional neural networks (CNNs) to learn relevant features given such a representation. CNNs can easily extend to 3-dimensions, so all spatial information can be retained by simply converting the 3-dimensional microstructure into a 3-dimensional grid with $1\mu m$ spacing. All that remains is to choose which values will fill the vector at each grid point. This chapter includes content from a paper published in the Journal of the Minerals, Metals, and Materials Society (JOM) in 2018 [1]. It explores using correlation between the values of various micromechanical field variables and distance to crack surface as a method of choosing the values for these vectors. The supplementary materials referenced in this chapter are also included.¹

A systematic correlation analysis is performed between simulated micromechanical fields in an uncracked polycrystal and the known path of an eventual fatigue-crack surface based on experimental observation. A concurrent multi-scale finite-element simulation of

¹Reprinted by permission from [Springer Nature]: [The Minerals, Metals, and Materials Society] [JOM] [1] (Data-Driven Correlation Analysis Between Observed 3D Fatigue-Crack Path and Computed Fields from High-Fidelity, Crystal-Plasticity, Finite-Element Simulations, K. Pierson, A. Spear, J. Hochhalter), © (2018)

cyclic loading is performed using a high-fidelity representation of grain structure obtained from near-field high-energy X-ray diffraction microscopy measurements. An algorithm is developed to parameterize and systematically correlate the 3D micromechanical fields from simulation with the 3D fatigue-failure surface from experiment. As a comparison, correlation coefficients are also computed between the micromechanical fields and hypothetical, alternative surfaces. Correlation of the fields with hypothetical surfaces is found to be consistently weaker than correlation with the known crack surface, suggesting that the micromechanical fields of the cyclically loaded, uncracked microstructure might provide some degree of predictiveness for microstructurally small fatigue-crack path, although the extent of such predictiveness remains to be tested. In general, gradients of the field variables exhibit stronger correlations with crack path than the field variables, themselves. Results from the data-driven approach implemented here can be leveraged in future model development for the prediction of fatigue-failure surfaces (for example, to facilitate univariate feature selection required by convolution-based models).

3.1 Introduction

Microstructural features play a governing role in the initiation and early stages of fatigue-crack growth. Variation in those features leads directly to variation in the paths and growth rates of microstructurally small cracks and, consequently, to scatter among fatigue lifetimes of structural components. Modeling this variability is critical given that most of the service life of fatigue-critical components can be consumed by initiation and growth of microstructure-sensitive cracks. Yet, these early stages of fatigue-crack evolution are difficult to model because of the complex dependence on a broad scope of microstructural features and the tendency to exceed propagation rates of long cracks with equivalent nominal stress intensity factors [2]. The reader is directed to Ref. [3] for an encompassing review of the metallographic aspects of microstructural heterogeneities and their role in fatigue cracking. Similarly, a review of the micromechanical descriptions of the effect of microstructural heterogeneities is given in Ref. [4].

Empirically based fatigue-life models were developed to link variability in fatigue life to microstructural features that were directly observable and quantifiable. Early examples of these approaches, which are overviewed in Refs. [5, 6, 7], based fatigue models on

microstructural characteristics such as inclusion size, shape, and location [8, 9, 10]. While empirical approaches have provided foundational knowledge regarding microstructural effects on fatigue performance, the resulting correlations and applicability of the developed models are valid only within the domain of the measured data and experimental parameters (e.g., boundary conditions, cyclic-load ratio, etc.). The formative works of Wei and Harlow [11, 12] clearly illustrate the need to use experimentation to discover and formulate hypotheses regarding the micromechanics at hand, not to fit empirical parameters.

Over the past two decades, there has been a shift toward computationally modeling microstructural features to investigate their impact on fatigue-crack initiation and early propagation. Such efforts typically use crystal-plasticity formulations to incorporate elastic and plastic anisotropy and either statistically representative or directly replicated microstructural domains to capture heterogeneities. For example, Bozek et al. [13] simulated the effect of cyclic loading on cracking of second-phase particles. Subsequently, Hochhalter et al. [14, 15] used fatigue indicator parameters to predict which cracks would extend beyond those cracked particles. Twin boundary crack initiation sites, and their dependence on local microstructure, were presented by Yeratapally et al. [16]. Fatigue indicator parameters were used by Musinski et al. [17] and Castelluccio et al. [18] to develop models for the subsequent propagation of cracks across a polycrystal. An encompassing study of the fatigue indicator parameters proposed to date is provided by Rovinelli et al. [19]. These studies represent a small sampling of high-fidelity, microstructure-sensitive fatigue modeling, and a more complete review can be found in Refs. [20, 21, 22, 23].

Advancements in these micromechanical modeling methods are being closely coupled with volumetric interrogation methods. This coupled approach provides a capability whereby parameters that cannot be measured directly can be computed in a simulation that serves to replicate the particular microstructure of the specimen. Spear et al. [24] used near-field high-energy X-ray diffraction (HEDM) to measure grain morphology in a sample of Al 6061-T6. These data were then used to generate a finite-element model, which replicated the as-measured grain structure and crack morphology. Rovinelli et al. [19] used diffraction contrast and phase contrast tomography to measure microstructure morphology and evolved crack faces in a near- β Ti alloy. Those data were used to generate a 3D FFT simulation with crystal plasticity. Bayesian networks were then used to analyze correlation

between the proposed short-crack driving forces and experimental observations.

As highlighted by the aforementioned studies, integrating X-ray-based measurement methods with high-fidelity simulation tools is providing a promising new approach to developing models for short-crack propagation [25]. New focus is on the efficient processing of these data (which can be quite large and complex) to extract useful information using data-driven approaches. In light of this, the objective of this paper is to describe a systematic, data-driven, correlation analysis between computed micromechanical fields of an uncracked, cyclically loaded 3D polycrystal and the known path of a 3D fatigue-failure surface based on direct observation from prior experimental measurements.

3.2 Materials and methods

3.2.1 Experimental measurements and mesh generation from prior work

The data used in this work are derived from experimental measurements [26] of 3D fatigue crack propagation within a grain-mapped volume of an Al-Mg-Si alloy. In that work, a naturally nucleated fatigue crack was propagated to failure under cyclic loading. The material volume above and below the failure surface was characterized post mortem using X-ray computed tomography and near-field HEDM. The former provided a high-resolution reconstruction of the failure surface, and the latter provided 3D grain maps adjacent to the failure surfaces, which can be seen in Figure 3.1. Of the entire measured crack surface, 31.8% was found to be approximately normal (within 20°) to the global loading direction. Additionally, 41% and 59% of the entire crack surface was deemed to be intergranular and transgranular, respectively, with the transgranular crack surface occurring along a wide variety of crystallographic planes.

The two halves of post mortem data were then virtually merged to generate a conformal finite-element mesh that represents explicitly and with high fidelity the internal grain boundaries and incremental crack surfaces measured from experiment, which is detailed in Ref. [24]. In this work, the conformal finite-element mesh representing the uncracked microstructure is used to obtain computed micromechanical fields, which are correlated with the reconstructed failure surface. Figure 3.1 summarizes the prior relevant work.

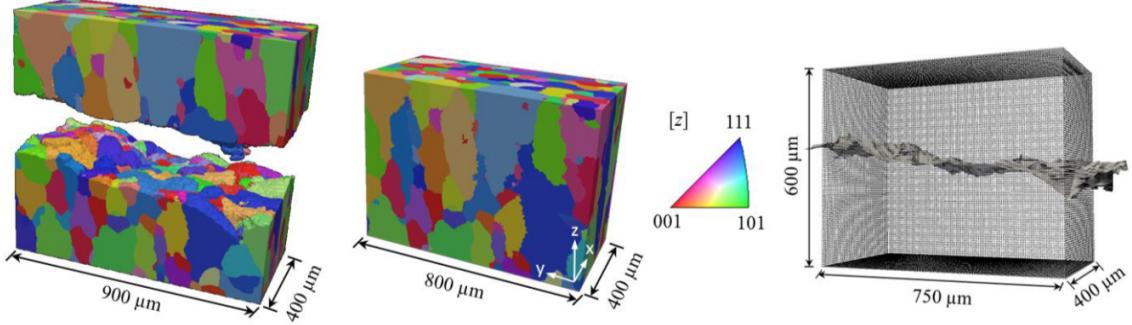


Figure 3.1: Post mortem reconstructions from an Al-Mg-Si alloy fatigue specimen based on near-field HEDM. (a). Approximation of uncracked volume (b) and reconstructed fatigue-failure surface from X-ray CT (c). The reference coordinate system is shown on the uncracked microstructural volume. Adapted with permission from Refs. [24] and [26].

3.2.2 Numerical simulation of cyclic loading applied to uncracked specimen

A concurrent multi-scale finite-element model is used to simulate cyclically applied displacement on the fatigue specimen tested in previous work [24]. The previously generated mesh from Ref. [24] consists of a local, polycrystalline region representing the uncracked microstructure and a global region representing the geometry of the fatigue specimen, shown in Figure 3.2. A mesh convergence study, detailed in Ref. [27], was carried out to ensure that both the global force-displacement response and local stresses, strains, and accumulated slip along an arbitrary query path through the polycrystalline domain are sufficiently converged. The converged, multi-scale mesh comprises 11.86M quadratic tetrahedral elements.

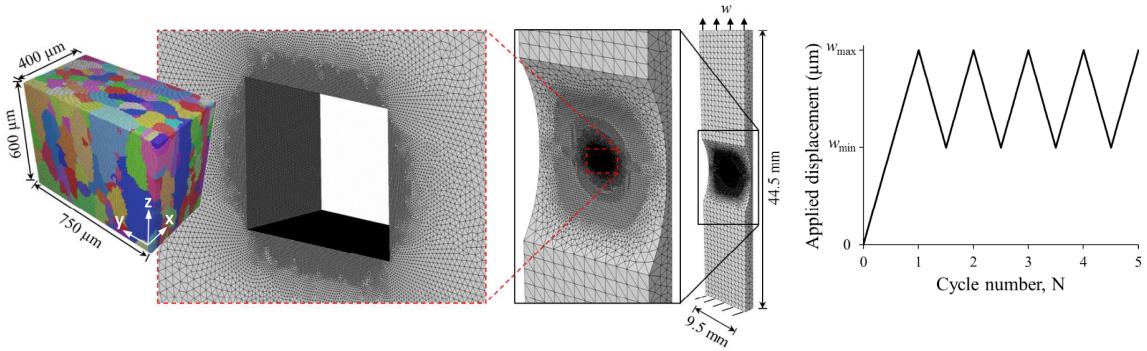


Figure 3.2: Concurrent multi-scale finite-element mesh and applied boundary conditions. The reference coordinate system is shown on the uncracked microstructural volume. Adapted with permission from Ref. [24].

A crystal, elasto-viscoplastic, constitutive model based on the implementation by Matous and Maniatty [28] is applied to the polycrystalline domain, and a J2-plasticity model is applied to the global domain. Both models are calibrated to ensure that the nominal (averaged) stress-strain behavior matches experimental data for the same material and that the simulated macroscopic-strain fields in the notch region match those from digital image correlation measurements, which are described elsewhere [27]. The crystal, elasto-viscoplastic, constitutive model is capable of predicting inhomogeneous deformation and stress fields that arise at the mesoscale as a result of interactions among discrete grains. In the model, plastic deformation is manifested by slip evolution on 12 octahedral slip systems ($\{111\} < 110 >$). All elements within the polycrystalline domain are assigned the same material properties; however, each element is assigned a crystal orientation based on the grain to which that element belongs. The crystal orientations are derived directly from the near-field HEDM measurements described above. The calibrated parameters for both constitutive models are provided in Table 3.1. Properties for the J2-plasticity model include the elastic modulus (E), Poisson's ratio (ν), yield strength (σ_y), and hardening modulus (h'). Properties for the crystal-plasticity model include a reference slip rate ($\dot{\gamma}_0$), a hardening rate-sensitivity parameter (m), a hardening-rate parameter (G_0), and initial hardness (g_0).

Boundary conditions are applied to replicate constraints and loading applied in the actual experiment. Namely, the grip ends of the specimen are constrained from displacing in the x and y directions. The lower grip end is further constrained from displacing in the z direction. The upper grip end is subjected to vertical displacement, w , which cycles between $w_{max} = 65 \mu\text{m}$ and $w_{min} = 38 \mu\text{m}$. The values for applied displacement are selected to reproduce the applied loading from experiment, detailed in Ref. [26].

Numerical simulation is performed using the parallelized finite-element code ScIFEN [29]. In total, five load cycles are simulated. The following list summarizes all local variables that are recorded for the entire polycrystalline domain at the beginning and end of each simulated load cycle.

Table 3.1: Calibrated material parameters.

J2-plasticity model				Crystal plasticity model			
$E(\text{MPa})$	ν	$\sigma_y(\text{MPa})$	$h'(\text{MPa})$	$\dot{\gamma}_0(\text{s}^{-1})$	m	$G_0(\text{MPa})$	$g_0(\text{MPa})$
70326	0.33	206.5	1200	0.05	0.0049	150.0	95.5

D_1	= Maximum value of accumulated slip among the 12 octahedral slip systems
D_2	= Maximum value of total accumulated slip over each slip plane
D_3	= Accumulated slip summed over all slip systems
D_4	= Maximum value of energy dissipated on a given slip plane during plastic deformation
D_5	= Modified Fatemi-Socie parameter
$\bar{\epsilon}$	= Symmetric strain tensor composed of ϵ_{xx} , ϵ_{yy} , ϵ_{zz} , ϵ_{xy} , ϵ_{xz} , and ϵ_{yz}
ϵ_1	= Principal eigenvalue of the strain tensor
ϵ_{vm}	= Von Mises strain
$\bar{\sigma}$	= Symmetric stress tensor composed of σ_{xx} , σ_{yy} , σ_{zz} , σ_{xy} , σ_{xz} , and σ_{yz}
σ_1	= Principal eigenvalue of the stress tensor
σ_{vm}	= Von Mises stress
M^{micro}	= Micromechanical Taylor factor [30]

The variables $D_{1\ldots 5}$ represent slip-based damage metrics described by Hochhalter et al. [14] and implemented within the ScIFEN framework. Additionally, micromechanical Taylor factor, M^{micro} , is computed throughout the polycrystalline domain based on the work of Raabe et al. [30], as shown in Equation 3.1.

$$M^{micro} = \frac{D_3}{\epsilon_{vm}}. \quad (3.1)$$

In Equation 3.1, D_3 is the summation over all N_s slip systems of the slip accumulated on each slip system, α , throughout cyclic loading:

$$D_3 = \sum_{\alpha=0}^{N_s} \int_0^t |\dot{\gamma}^\alpha| dt, \quad (3.2)$$

where $\dot{\gamma}^\alpha$ is the slip rate on a given slip system. The term ϵ_{vm} represents the local von Mises equivalent strain, which is computed as:

$$\epsilon_{vm} = \sqrt{\frac{2}{3} \bar{\epsilon} : \bar{\epsilon}}. \quad (3.3)$$

The variables in the above list, along with the cyclic changes in those variables, are included in the systematic correlation analysis.

3.2.3 Convergence of cyclic field variables

The convergence of field variables is assessed by considering the change in each variable throughout the entire polycrystalline domain as a function of cycle count. For any given variable, λ , its cyclic value is computed at each point in the model based on the change in that variable between minimum to maximum displacement over a given loading cycle. The change in each cyclic value is also computed between successive loading cycles. In other words, at each point in the model, and for all variables in a given cycle, N :

$$\Delta\lambda_N = \lambda_{w_{max},N} - \lambda_{w_{min},N} \quad (3.4)$$

$$\Delta^2\lambda_N = \Delta\lambda_N - \Delta\lambda_{N-1}. \quad (3.5)$$

Figure 3.3 illustrates the convergence of $\Delta\sigma_{zz}$ and a visualization of $\Delta^2\sigma_{zz}$ approaching zero (similar convergence is verified for all variables). Convergence of the cyclic field variables implies that the results taken from the fifth loading cycle sufficiently represent the state of the polycrystalline domain to perform a meaningful correlation study.

3.2.4 Correlation analysis

The finite-element results are first converted to a format amenable to performing the correlation analysis. Results associated with the fifth loading cycle are subsampled from the unstructured finite-element mesh onto a $383 \times 750 \times 600 \mu\text{m}^3$ grid with $1 \mu\text{m}$ spacing between points. This is done for all field variables, producing a scalar-valued grid for each variable, λ , corresponding to peak load, and for each cyclic value, $\Delta\lambda$.

Additionally, the spatial gradients of every λ and $\Delta\lambda$ variable are calculated and included in the correlation analysis. Here, the gradients of λ and $\Delta\lambda$ at each point in the model are computed based on finite differences in the y and z directions (reminiscent of

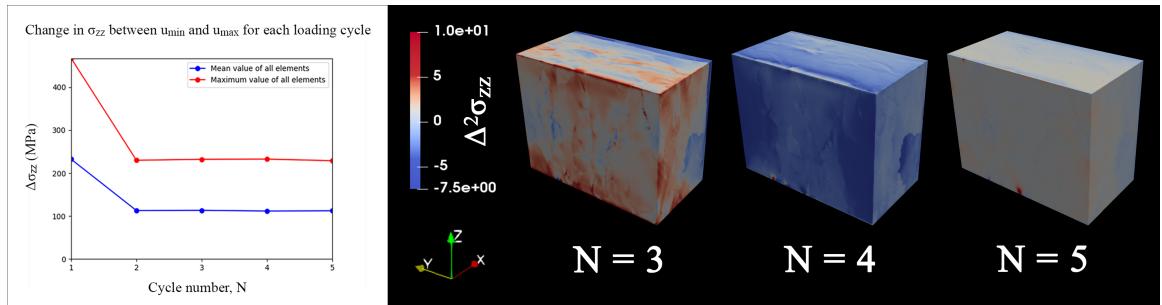


Figure 3.3: Convergence of $\Delta\sigma_{zz}$ and $\Delta^2\sigma_{zz}$ (MPa) during cyclic loading.

2D image slices through the volume) using $h = 3 \mu\text{m}$ spacing², after which the L_2 norm is taken to produce a scalar value. In the following equations, subscripts indicate the grid coordinates of a given point.

$$\frac{d\lambda_{x,y,z}}{dy} = \frac{\lambda_{x,y+h,z} - \lambda_{x,y-h,z}}{2h}, \quad \frac{d\lambda_{x,y,z}}{dz} = \frac{\lambda_{x,y,z+h} - \lambda_{x,y,z-h}}{2h} \quad (3.6)$$

The next step is to represent the 3D crack surface, which was previously reconstructed from post mortem X-ray CT data and aligned with the uncracked microstructural domain, as a 2D grid of elevation values. This is accomplished by initializing a $383 \times 750 \mu\text{m}^2$ grid with $1 \mu\text{m}$ spacing between points, and then assigning to each point an interpolated value of the corresponding z coordinate of the crack surface, resulting in a height-map. The interpolation method is an inverse bilinear interpolation. For a given point in the $x - y$ grid plane, the corresponding height-map value of the crack surface is used to identify neighboring points in the 3D scalar-valued grids.

In total, there are 88 scalar-valued grids to consider in the correlation analysis: one for each field variable, λ , listed in Section 3.2.2; one for each cyclic variable, $\Delta\lambda$; and one for the gradient values of both λ and $\Delta\lambda$. Each grid consists of $383 \times 750 \times 600 = 1.724 \times 10^9$ data points derived from high-fidelity numerical simulation of the uncracked microstructure. There are an additional $383 \times 750 = 2.87 \times 10^5$ data points derived from the experimentally observed fatigue-failure surface.

Using this as input data, the goal of the algorithm implemented here is to determine – with minimal prior assumptions – which micromechanical field variables are correlated with fatigue-crack path. The method chosen here is to compute the correlation between the value of a given variable at a particular point in the microstructure and its vertical distance to the crack surface. Only a local neighborhood around the crack surface is considered, i.e., a region into which the crack could plausibly have grown from any given configuration. The value of L was systematically varied, and ultimately a value of $L = 25 \mu\text{m}$ (approximately 25% of the average grain diameter) above and below the crack surface is selected to optimize the correlation results. This value is used in all correlation analyses described throughout this paper. The definition of L is shown in Figure 3.4 for a slice and subsection

²Values of $1 \mu\text{m}$ and $3 \mu\text{m}$ were considered for h , with the latter being equal to half the size of the discretization for the quadratic finite-element mesh. Ultimately, $h = 3 \mu\text{m}$ was found to provide stronger correlations.

of the grid defined for ΔM^{micro} . The grid points are shown in black with a spacing of $1 \mu\text{m}$, and the finite-element results obtained using an unstructured mesh are shown in the background. The trace of the actual crack surface is superimposed for reference.

Of this subset of points within L of the surface, only regions where the gradient of a given field variable is sufficiently high are considered. This filter is implemented due to the propensity of some variables to exhibit near-zero change within the neighborhood, L . In such cases, the correlation of that variable with distance to crack surface does not add value to the analysis. In order for a gradient of a variable to be considered sufficiently high, it must be at least $t\%$ of its value at the same point. Here, t is determined independently for each variable by finding the value of t that maximizes the correlation of that variable, while still retaining at least 10% of the total crack surface (see Figure 3.5). For example, $t = 8\%$ for ϵ_{zz} , while $t = 11\%$ for ΔD_3 . This implementation guarantees that regions contributing to the correlation analysis are not overly sparse, yet contain λ -values that are highly informative.

This method can be easily parallelized, as computations are performed on each point of the crack surface independently of all other points. Since computations must be performed for each of the $88 \times 383 \times 750 \times 50 = 1.264 \times 10^9$ points, this parallelization is a necessity. The data are loaded into shared memory accessible by all processes, after which the code

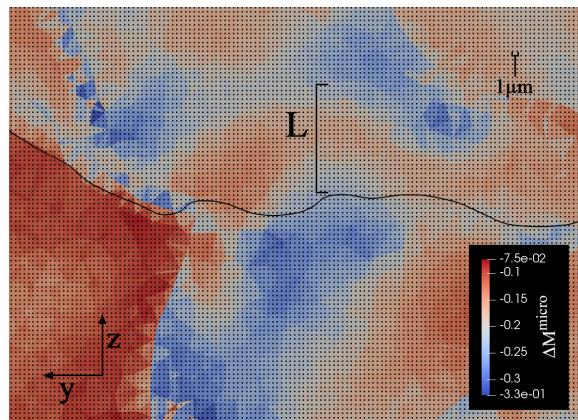


Figure 3.4: Cyclic micromechanical Taylor factor computed for the uncracked polycrystal, shown at a particular slice through the volume. Superimposed is the trace of the actual crack surface from X-ray CT imaging. Also superimposed is the grid used for correlation analysis. The neighborhood of influence is defined by the distance L above and below the crack surface.

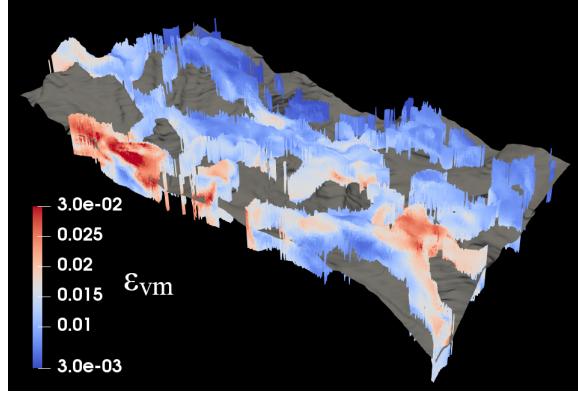


Figure 3.5: Colored data points within L vertical distance of the crack surface that meet the spatial-gradient threshold for inclusion in the correlation analysis of ϵ_{vm} .

is run on multiple cores. The entire algorithm is described in Figure 3.6.

The results from the algorithm consist of data frames, where each row is an observed point from the microstructure and each column is the value of a field variable, or distance to crack surface in the case of the last column. These data frames are then imported into R [31], which provides robust libraries for correlation analyses and visualizations. Pearson correlation coefficients are computed for each column with respect to the final column, and then visualized for comparison.

3.3 Results and discussion

The computed fields for all 22 metrics and their respective cyclic values are visualized for the uncracked microstructure in Figs. 3.7, 3.8, 3.9, 3.10, 3.11, 3.12, 3.13, and 3.14. The computed correlation coefficients are shown in Figure 3.15. A negative correlation indicates that as the distance to crack surface decreases, the variable of interest increases,

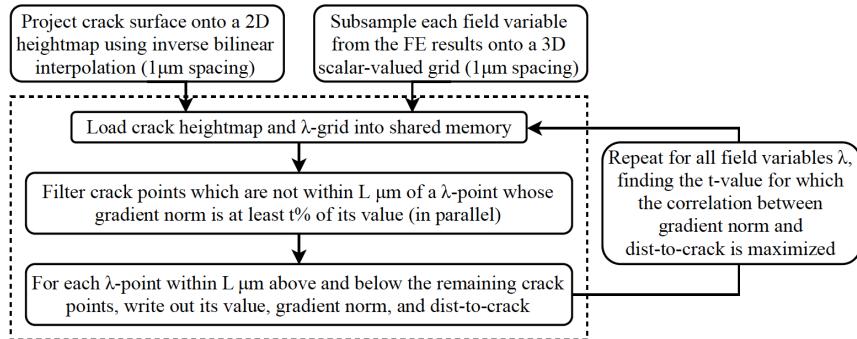


Figure 3.6: Data-extraction algorithm.

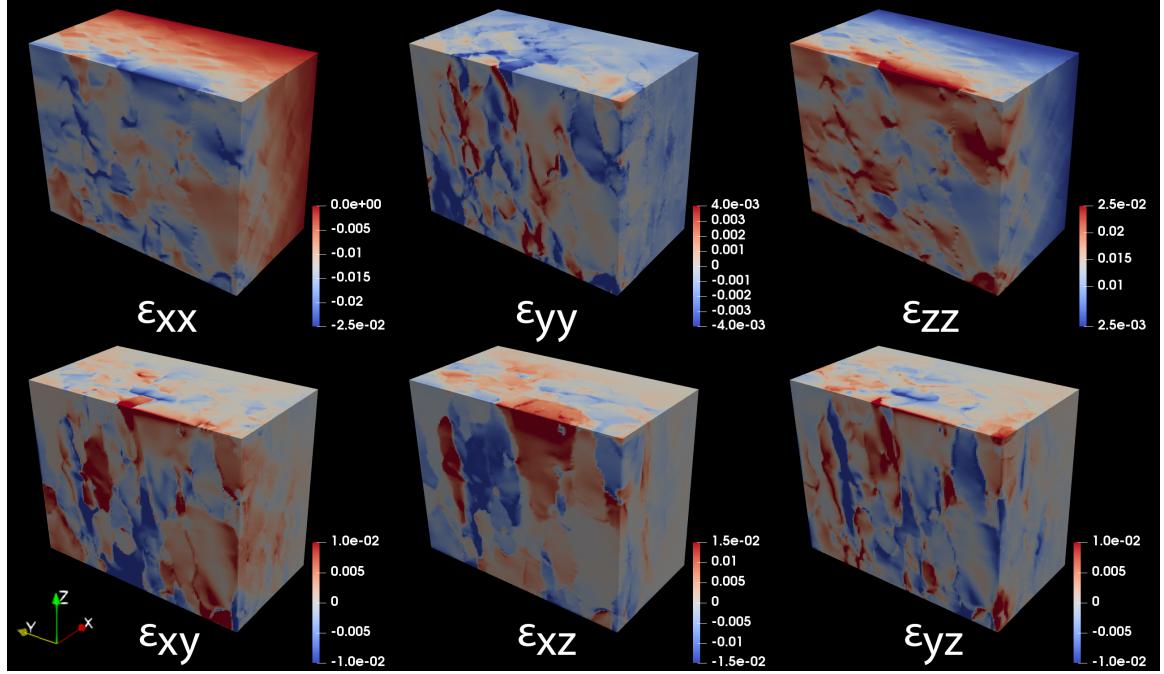


Figure 3.7: Grid data showing strain-tensor components computed at the peak load of the fifth loading cycle for an uncracked microstructure in a multi-scale finite-element simulation.

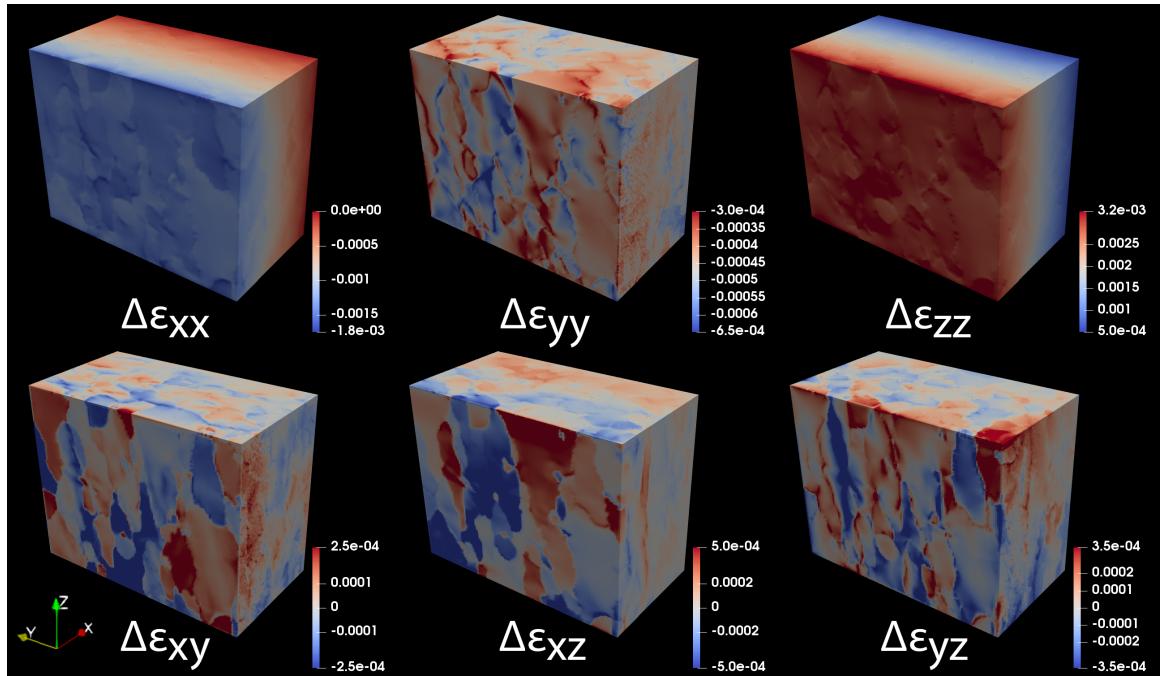


Figure 3.8: Grid data showing cyclic values of the strain-tensor components computed for an uncracked microstructure in a multi-scale finite-element simulation.

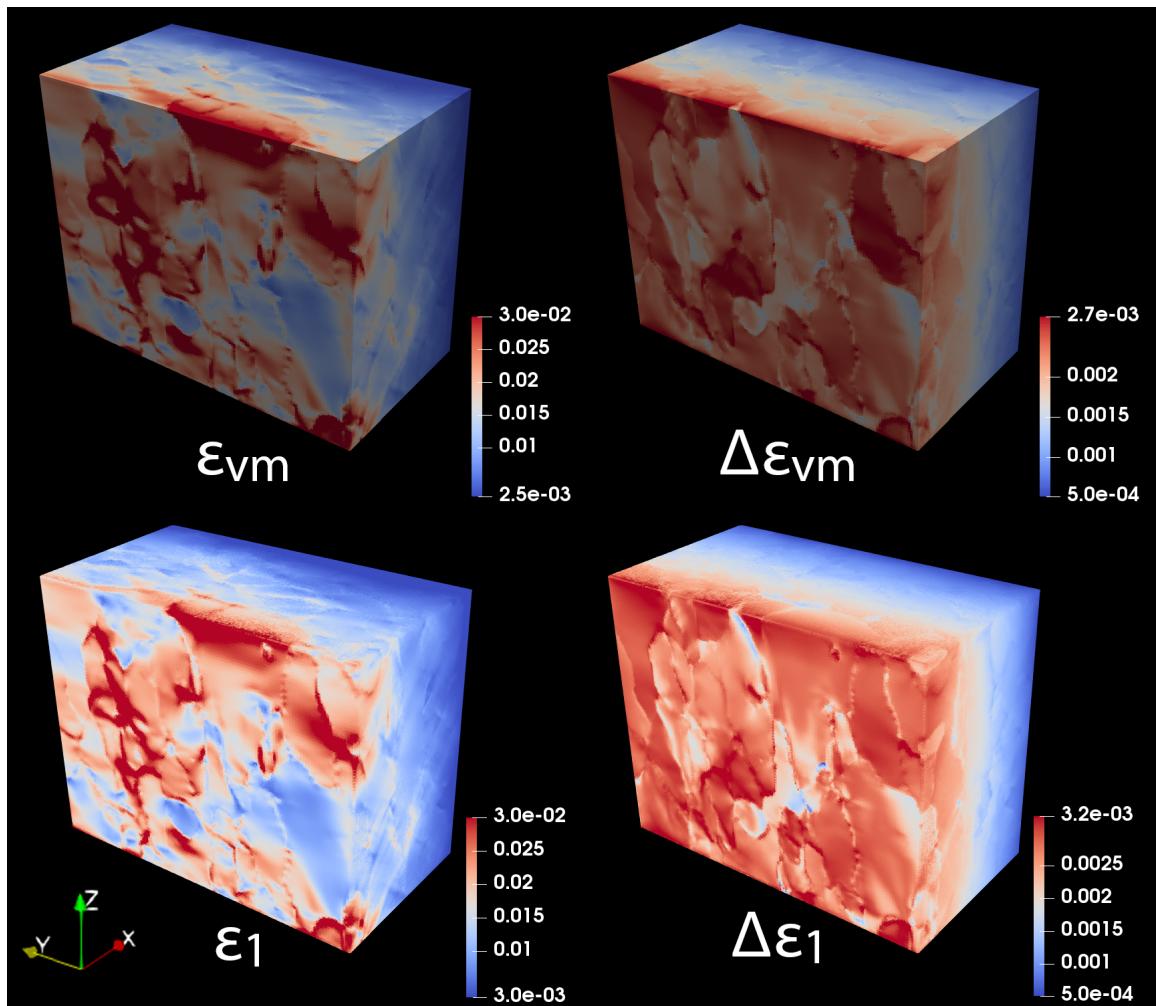


Figure 3.9: Grid data showing von-Mises and maximum-principal strains computed at the peak load of the fifth loading cycle (left) and the corresponding cyclic values (right) for an uncracked microstructure in a multi-scale finite-element simulation.

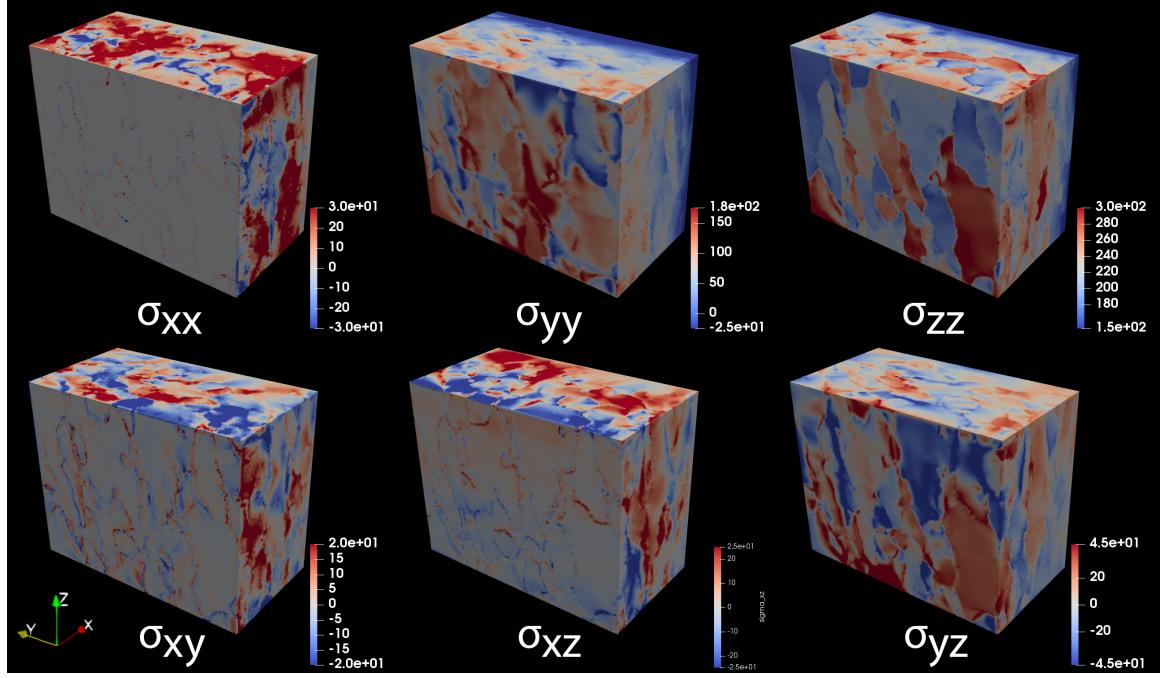


Figure 3.10: Grid data showing stress-tensor components (MPa) computed at the peak load of the fifth loading cycle for an uncracked microstructure in a multi-scale finite-element simulation.

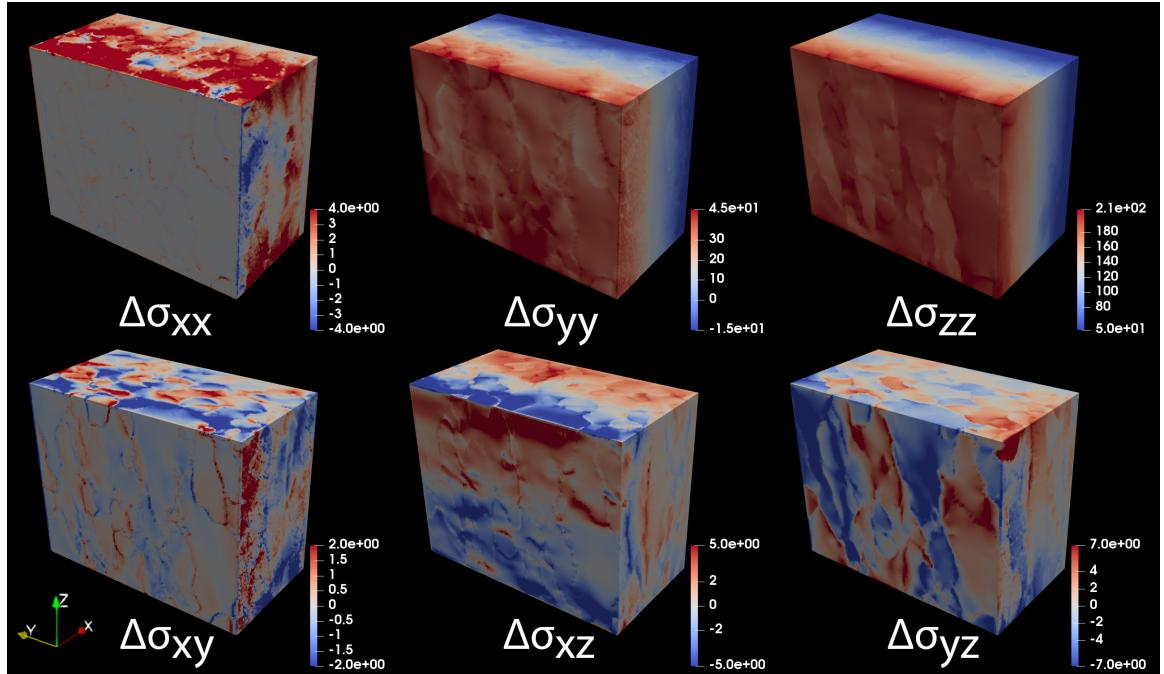


Figure 3.11: Grid data showing cyclic values of the stress-tensor components (MPa) computed for an uncracked microstructure in a multi-scale finite-element simulation.

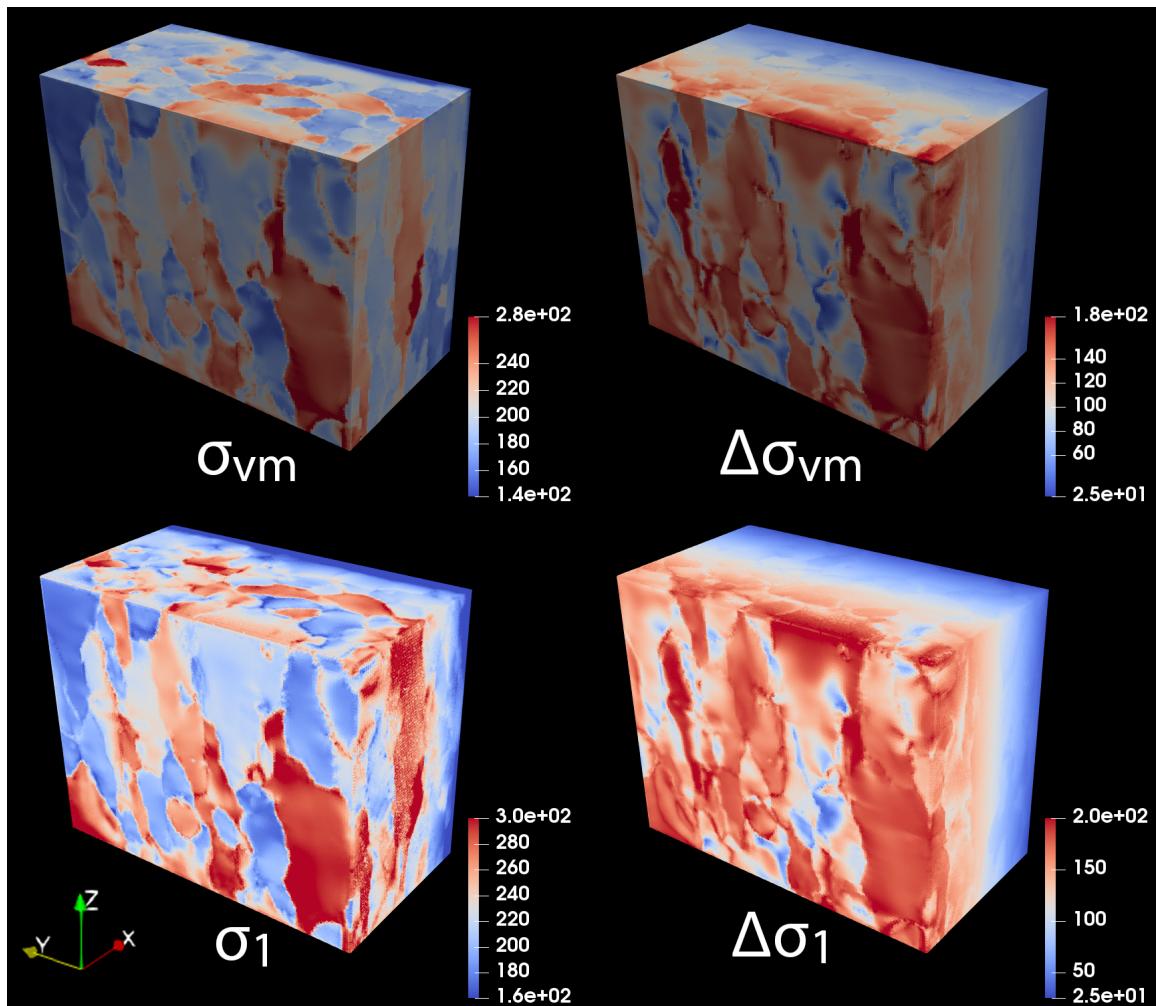


Figure 3.12: Grid data showing von-Mises and maximum-principal stresses (MPa) computed at the peak load of the fifth loading cycle (left) and the corresponding cyclic values (right) for an uncracked microstructure in a multi-scale finite-element simulation.

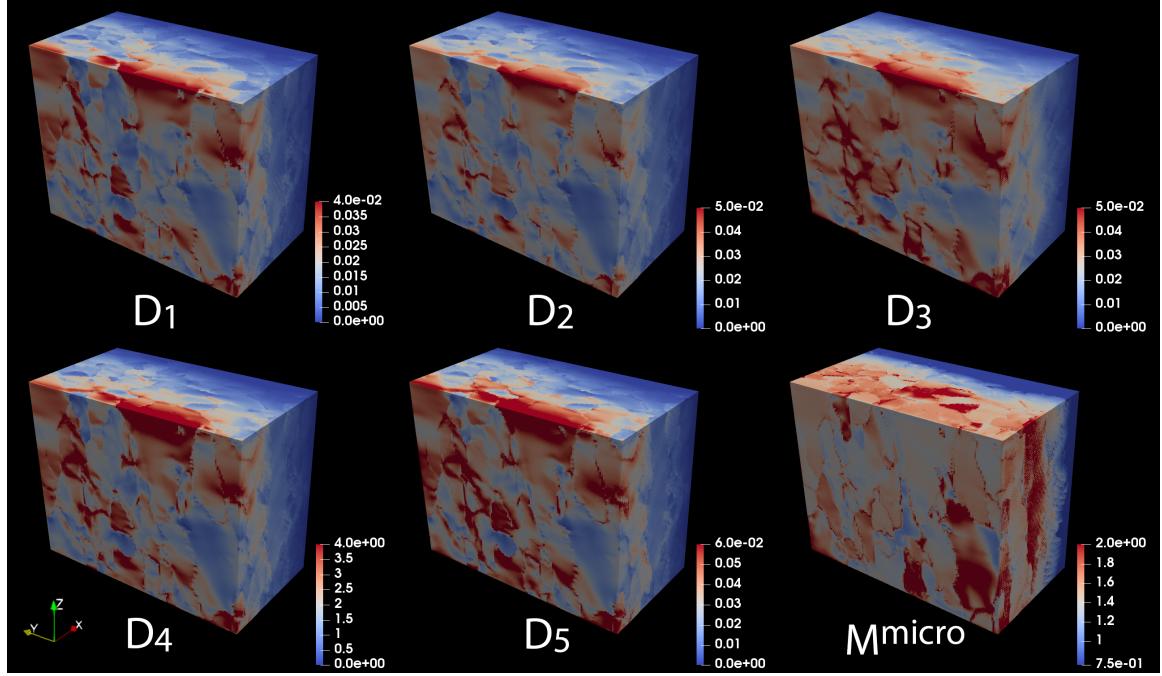


Figure 3.13: Grid data showing slip-based damage metrics computed at the peak load of the fifth loading cycle for an uncracked microstructure in a multi-scale finite-element simulation.

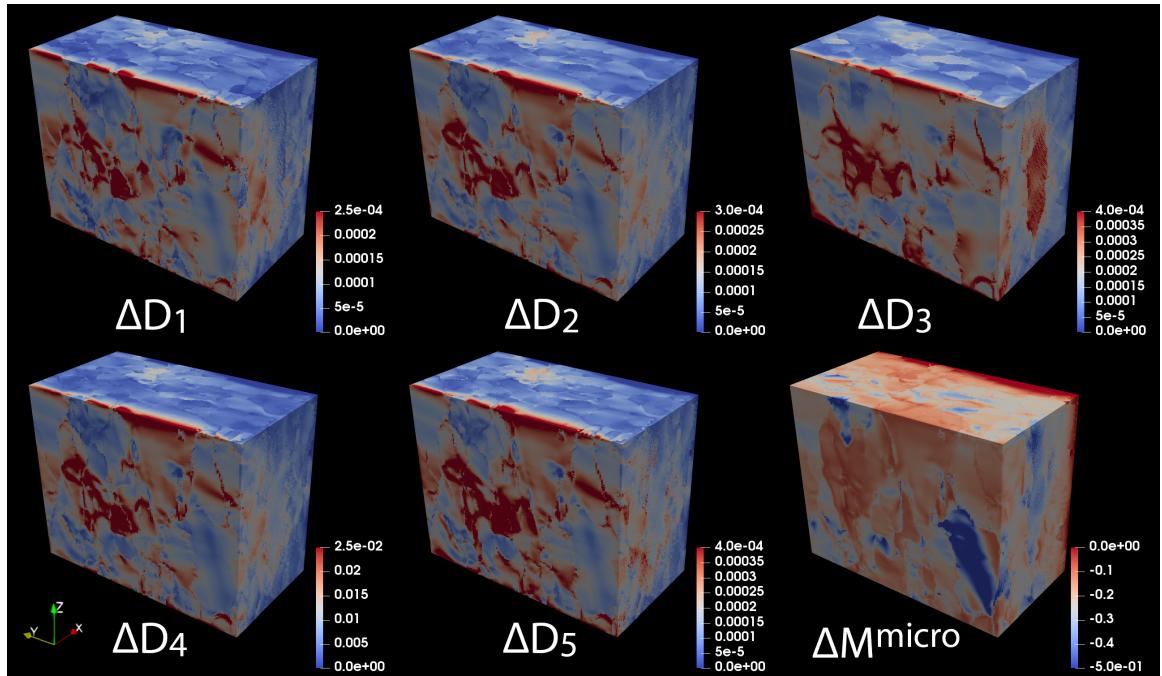


Figure 3.14: Grid data showing cyclic values of the slip-based damage metrics computed for an uncracked microstructure in a multi-scale finite-element simulation.

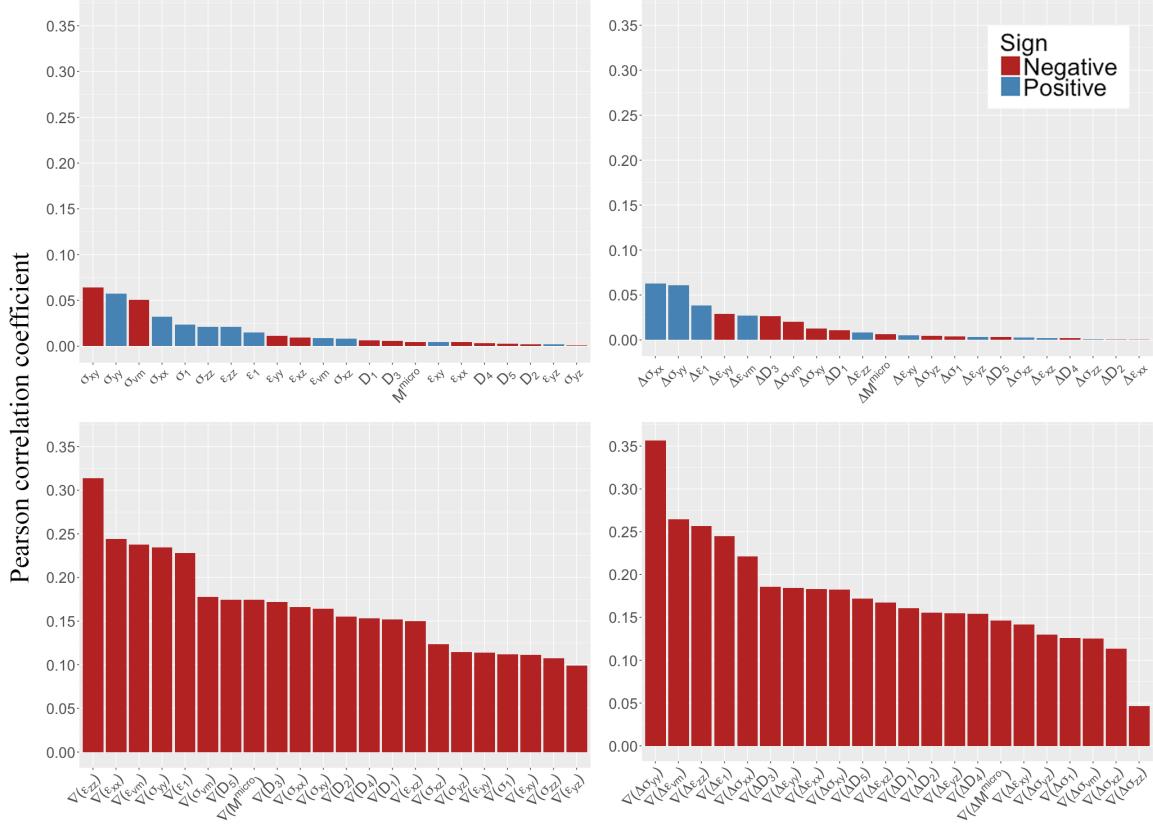


Figure 3.15: Correlation coefficients computed between the following metrics and distance to crack surface: (a) field variables, λ ; (b) cyclic change in field variables, $\Delta\lambda$; (c) spatial gradient of field variables, $\nabla(\lambda)$; (d) spatial gradient of cyclic field variables, $\nabla(\Delta\lambda)$.

and vice versa.

The correlation coefficients shown in Figure 3.15 are not, at first look, overwhelmingly high. The challenge in assessing the statistical significance of these correlations is that the independent and identically distributed assumption required by a typical *t-test* has been violated. A nonparametric measure, such as bootstrapping [32] or using Spearman rank, could potentially help resolve this problem. An alternative approach to determine whether the correlation values in Figure 3.15 are meaningful (but not necessarily statistically significant) is to apply the same correlation algorithm to alternative surfaces that could serve as hypothetical crack paths through the microstructure. Four alternative paths are considered: two z-normal planes positioned 200 μm above and below the lower and upper faces, respectively, of the microstructural domain, and two instances of the measured crack surface offset by 125 μm above and below the known crack path. Figure 3.16

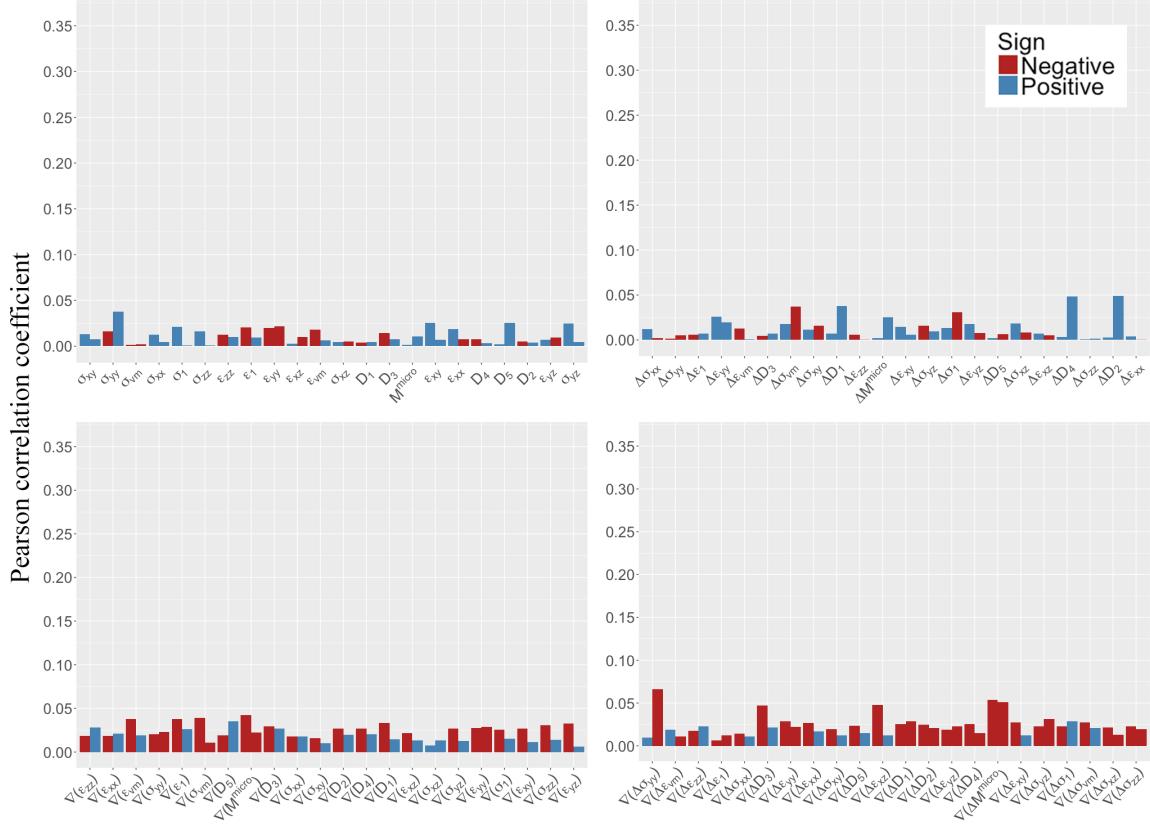


Figure 3.16: Correlation coefficients computed between the following metrics and distance to hypothetical crack surfaces located 125 μm below (left bars) and 125 μm above (right bars) the known crack surface: (a) field variables, λ ; (b) cyclic change in field variables, $\Delta\lambda$; (c) spatial gradient of field variables, $\nabla(\lambda)$; (d) spatial gradient of cyclic field variables, $\nabla(\Delta\lambda)$.

shows the results for the latter two cases. Results for the two z-normal planes are provided in Figure 3.17. In all four cases, the correlations are consistently weaker than those for the actual crack surface, providing some evidence that micromechanical fields from a cyclically loaded, uncracked microstructure tend to correlate with the actual path of the 3D fatigue crack.

As shown in Figure 3.15, the crack path is generally shown to be more strongly correlated with the spatial-gradient values than with the field variables at peak load or with the cyclic field variables, suggesting that the eventual fatigue crack sought paths of high gradients in stress and/or strain space. Considering only the gradient-based parameters, D_3 and D_5 exhibit the strongest correlation with crack path among all slip-based damage metrics, although the difference is relatively marginal. This indicates that the

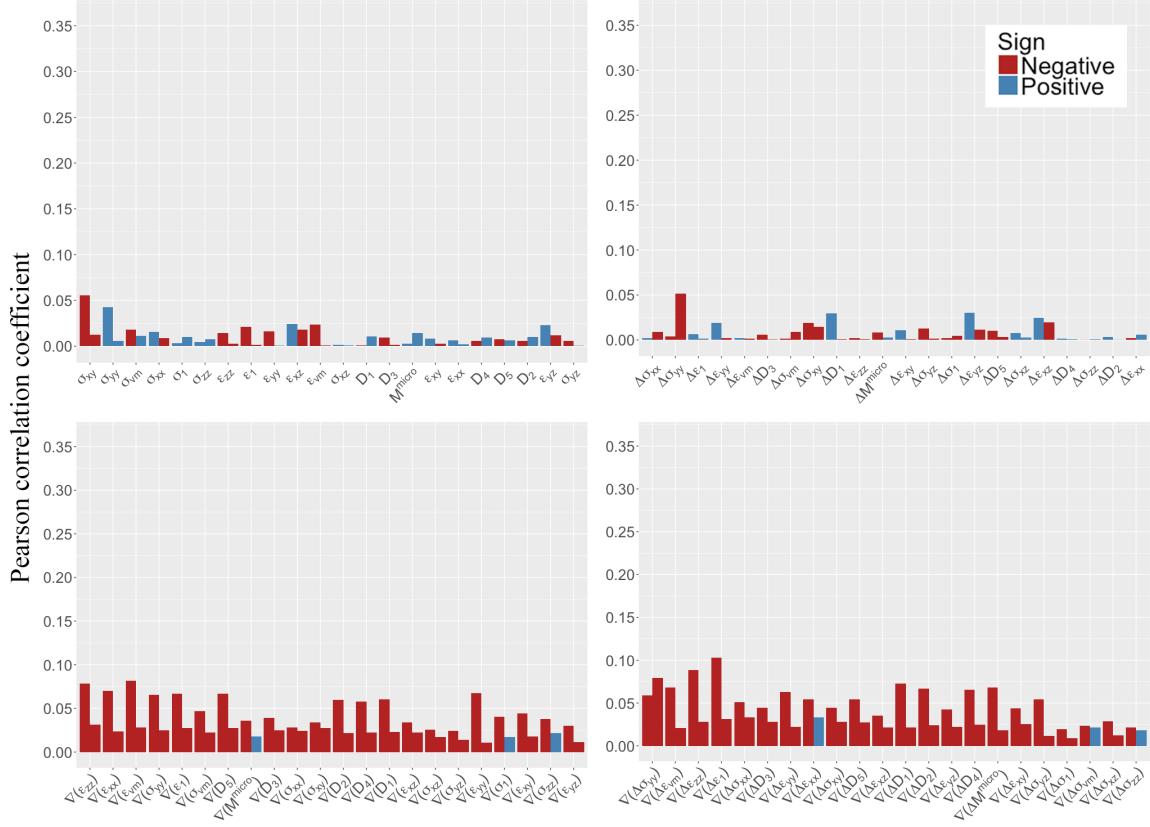


Figure 3.17: Correlation coefficients computed between each variable and distance to z-normal planes located $200 \mu\text{m}$ above the bottom face (left bars) and $200 \mu\text{m}$ below the top face (right bars) of the microstructural domain.

combined effect of slip activity on multiple slip systems (D_3) as well as the combined effect of crystallographic slip and tensile stress on a slip plane (D_5) play a more significant role in predicting crack path than just the maximum value of slip on a single slip system or slip plane. While it seems reasonable for some of the metrics to have a relatively strong correlation with crack path (e.g., $\nabla(\epsilon_{zz})$, $\nabla(\Delta\epsilon_{vm})$, and $\nabla(\Delta\epsilon_1)$), there are other correlations that are not immediately intuitive (e.g., $\nabla(\epsilon_{xx})$ and $\nabla(\Delta\sigma_{yy})$). Clearly, the factors affecting crack growth are highly complex, and we cannot rely on treating all field variables as independent mechanisms. As such, it is likely that there exists some complex combination of variables that serves to accommodate, promote, or hinder crack propagation, which corroborates previous conclusions in the literature (e.g. [21, 15, 19]). It will require further investigation using, for example, machine learning to understand how interaction of the variables leads to such apparent correlations.

Since the discontinuity of the crack is not modeled in this work (as was done in previous work [24]), the micromechanical fields computed here do not account for stress redistribution due to the formation of new traction-free surface area; nor do they account for plastic zones or stress concentrations in the vicinity of a crack front. However, the objective is to identify what, if any, correlations exist between micromechanical fields in an uncracked microstructure and the 3D path of an eventual fatigue-crack surface. The implications of relatively strong correlations could be significant, in that the crack path might be approximated prior to (or without having to) incorporate geometrically explicit crack representations. It is expected that such correlations would be relatively strong within a limited spatial domain surrounding the crack-nucleation site. Future work could investigate the size of this domain, beyond which the correlations are expected to diminish.

The analysis from this work could provide insight into the extraction of relevant features for predictive machine-learning models. In models where univariate feature selection is required, such as convolution-based networks that infer useful information based on a grid of scalars, choosing the correct variables to use as a representation is critical. With new insight from the correlation analysis presented here, ongoing studies by the authors focus on the use of machine learning to identify critical combinations of, and relationships among, the most correlated variables with the evolution of fatigue-crack surfaces.

3.4 Conclusions

This work presents a systematic, data-driven approach to parameterize and correlate local micromechanical fields computed for an uncracked, cyclically loaded specimen with the known 3D fatigue-crack path observed from experiment. Specifically, local micromechanical field variables in the vicinity of an eventual crack surface are correlated with distance to that eventual surface. The intent is to identify whether the response of the uncracked microstructure subjected to realistic far-field loading might provide any predictive power in identifying the path of a fatigue-failure surface, which could have implications for future modeling efforts. In this work, a total of 88 micromechanical parameters, and 1.264×10^9 data points, are considered in the analysis. Thus, the data used here are considered to be large and rich in nature, albeit for just a single specimen. The parameters include field variables and slip-based damage metrics computed at peak load, as well

as their corresponding cyclic values. Also considered are the gradients of all previously mentioned parameters. The micromechanical parameters, taken at discrete points, are then correlated with the distance from a given point to the known crack surface.

In general, the gradients of the micromechanical field variables appear to exhibit a stronger correlation with crack path than the field variables, themselves. This supports the claim that fatigue cracks generally seek paths of high gradients of stress, strain, or both. The variables, treated independently, are not sufficient to fully describe evolution of fatigue-crack surfaces. However, the systematic correlation analysis from this work provides insight into the extraction of relevant features for future development and testing of predictive models.

3.5 Acknowledgements

This material is based on research sponsored by the Air Force Office of Scientific Research Young Investigator Program, under agreement number FA9550-15-1-0172. The authors gratefully acknowledge Drs. Ravi Chona, Ben Smarslok, and Brian Gockel of the Air Force Research Laboratory for their support of the work. The support and resources from the Center for High Performance Computing at the University of Utah are gratefully acknowledged.

3.6 References

- [1] K. Pierson, J. Hochhalter, and A. Spear, "Data-driven correlation analysis between observed 3d fatigue-crack path and computed fields from high-fidelity, crystal-plasticity, finite-element simulations," *JOM*, 2018.
- [2] R. Ritchie and J. Lankford, *Small fatigue cracks*. The Metallurgical Society INC., Warrendale, PA, Jan 1986.
- [3] S. V. Narayana Murty, S. K. Manwatkar, and P. R. Narayanan, "A review on the role of microstructural heterogeneities in the failure of aerospace components: A metallographer's perspective," *Materials Performance and Characterization*, vol. 6, no. 3, pp. 311–333, 2017.
- [4] H. Mughrabi, "Microstructural mechanisms of cyclic deformation, fatigue crack initiation and early crack growth," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 373, no. 2038, 2015.
- [5] Y. Murakami and M. Endo, "Effects of defects, inclusions and inhomogeneities on fatigue strength," *International Journal of Fatigue*, vol. 16, pp. 163–182, Apr 1994.

- [6] A. Fatemi and L. Yang, "Cumulative fatigue damage and life prediction theories: A survey of the state of the art for homogeneous materials," *International Journal of Fatigue*, vol. 20, no. 1, pp. 9–34, 1998.
- [7] K. Hussain, "Short fatigue crack behaviour and analytical models: A review," *Engineering Fracture Mechanics*, vol. 58, no. 4, pp. 327–354, 1997.
- [8] P. J. Laz and B. M. Hillberry, "Fatigue life prediction from inclusion initiated cracks," *International Journal of Fatigue*, vol. 20, no. 4, pp. 263–270, 1998.
- [9] J. R. Brockenbrough, A. J. Hinkle, P. E. Magnusen, and R. J. Bucci, "Microstructurally based model of fatigue and growth," in *Proceedings, FAA/NASA International Symposium on Advanced Structural Integrity Methods for Airframe Durability and Damage Tolerance* (C. Harris, ed.), vol. NASA Conference Publication 3274, Part I, pp. 71–84, Sep 1994.
- [10] M. A. Przystupa, R. J. Bucci, P. E. Magnusen, and A. J. Hinkle, "Microstructure based fatigue life predictions for thick plate 7050-T7451 airframe alloys," *International Journal of Fatigue*, vol. 19, no. 93, pp. 285–288, 1997.
- [11] R. P. Wei and D. G. Harlow, "Mechanistically based probability modelling, life prediction and reliability assessment," *Modelling and Simulation in Materials Science*, vol. 13, no. 1, pp. R33–R51, 2005.
- [12] D. G. Harlow and R. P. Wei, "Probability modeling and material microstructure applied to corrosion and fatigue of aluminum and steel alloys," *Engineering Fracture Mechanics*, vol. 76, pp. 695–708, 2009.
- [13] J. E. Bozek, J. D. Hochhalter, M. G. Veilleux, M. Liu, G. Heber, S. D. Sintay, A. D. Rollett, D. J. Littlewood, A. M. Maniatty, H. Weiland, R. J. Christ Jr, J. Payne, G. Welsh, D. G. Harlow, P. A. Wawrzynek, and A. R. Ingraffea, "A geometric approach to modeling microstructurally small fatigue crack formation: I. probabilistic simulation of constituent particle cracking in AA 7075-T651," *Modelling Simulation in Materials Science*, vol. 16, no. 6, p. 065007 (28pp), 2008.
- [14] J. Hochhalter, D. Littlewood, R. Christ Jr, M. Veilleux, J. Bozek, A. Ingraffea, and A. Maniatty, "A geometric approach to modeling microstructurally small fatigue crack formation: II. Physically based modeling of microstructure-dependent slip localization and actuation of the crack nucleation mechanism in AA 7075-T651," *Modelling and Simulation in Materials Science and Engineering*, vol. 18, no. 4, p. 045004, 2010.
- [15] J. D. Hochhalter, D. J. Littlewood, M. G. Veilleux, J. E. Bozek, A. M. Maniatty, A. D. Rollett, and A. R. Ingraffea, "A geometric approach to modeling microstructurally small fatigue crack formation: III. Development of a semi-empirical model for nucleation," *Modelling and Simulation in Materials Science and Engineering*, vol. 19, no. 3, p. 035008, 2011.
- [16] S. R. Yeratapally, M. G. Glavicic, M. Hardy, and M. D. Sangid, "Microstructure based fatigue life prediction framework for polycrystalline nickel-base superalloys with emphasis on the role played by twin boundaries in crack initiation," *Acta Materialia*, vol. 107, pp. 152–167, 2016.

- [17] W. D. Musinski and D. L. McDowell, "Simulating the effect of grain boundaries on microstructurally small fatigue crack growth from a focused ion beam notch through a three-dimensional array of grains," *Acta Materialia*, vol. 112, pp. 20–39, 2016.
- [18] G. M. Castelluccio and D. L. McDowell, "Microstructure-sensitive small fatigue crack growth assessment: Effect of strain ratio, multiaxial strain state, and geometric discontinuities," *International Journal of Fatigue*, vol. 82, pp. 521–529, 2016.
- [19] A. Rovinelli, Y. Guilhem, H. Proudhon, R. A. Lebensohn, W. Ludwig, and M. D. Sangid, "Assessing reliability of fatigue indicator parameters for small crack growth via a probabilistic framework," *Modelling and Simulation in Materials Science and Engineering*, vol. 25, no. 4, p. 045010, 2017.
- [20] M. D. Sangid, "The physics of fatigue crack initiation," *International Journal of Fatigue*, vol. 57, pp. 58–72, 2013. Fatigue and Microstructure: A special issue on recent advances.
- [21] D. McDowell and F. Dunne, "Microstructure-sensitive computational modeling of fatigue crack formation," *International Journal of Fatigue*, vol. 32, no. 9, pp. 1521–1542, 2010. Emerging Frontiers in Fatigue.
- [22] G. M. Castelluccio, W. D. Musinski, and D. L. McDowell, "Recent developments in assessing microstructure-sensitive early stage fatigue of polycrystals," *Current Opinion in Solid State and Materials Science*, vol. 18, no. 4, pp. 180–187, 2014. Slip Localization and Transfer in Deformation and Fatigue of Polycrystals.
- [23] H.-J. Christ, C.-P. Fritzen, and P. Köster, "Micromechanical modeling of short fatigue cracks," *Current Opinion in Solid State and Materials Science*, vol. 18, no. 4, pp. 205–211, 2014. Slip Localization and Transfer in Deformation and Fatigue of Polycrystals.
- [24] A. Spear, J. Hochhalter, A. Cerrone, S. F. Li, J. Lind, R. Suter, and A. Ingraffea, "A method to generate conformal finite-element meshes from 3d measurements of microstructurally small fatigue-crack propagation," *Fatigue and Fracture of Engineering Materials and Structures*, vol. 39, no. 6, pp. 737–751, 2016.
- [25] T. J. Turner, P. A. Shade, J. V. Bernier, S. F. Li, J. C. Schuren, P. Kenesei, R. M. Suter, and J. Almer, "Crystal plasticity model validation using combined high-energy diffraction microscopy data for a Ti-7Al specimen," *Metallurgical and Materials Transactions A*, vol. 48, pp. 627–647, Feb 2017.
- [26] A. Spear, S. F. Li, J. Lind, R. Suter, and A. Ingraffea, "Three-dimensional characterization of microstructurally small fatigue-crack evolution using quantitative fractography combined with post-mortem x-ray tomography and high-energy x-ray diffraction microscopy," *Acta Materialia*, vol. 76, pp. 413–424, 2014.
- [27] A. Spear, *Numerical and Experimental Studies of Three-Dimensional Crack Evolution in Aluminum Alloys: Macroscale To Microscale*. PhD thesis, Cornell University, 2014.
- [28] K. Matouš and A. Maniatty, "Finite element formulation for modelling large deformations in elasto-viscoplastic polycrystals," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 14, pp. 2313–2333, 2004.

- [29] J. E. Warner, G. F. Bomarito, G. Heber, and J. D. Hochhalter, "Scalable implementation of finite elements by NASA - Implicit (SciFEi)," *NASA/TM-2016-219180*, 2016.
- [30] D. Raabe, M. Sachtleber, Z. Zhao, F. Roters, and S. Zaefferer, "Micromechanical and macromechanical effects in grain scale polycrystal plasticity experimentation and simulation," *Acta Materialia*, vol. 49, no. 17, pp. 3433–3441, 2001.
- [31] R. Statistical Package, "R: A language and environment for statistical computing," *Vienna, Austria: R Foundation for Statistical Computing*, 2009.
- [32] P. I. Good, *Permutation, parametric and bootstrap tests of hypotheses*. Springer, 2010.

CHAPTER 4

INFERENCE OF POSITION AND GROWTH RATE FOR ALL POINTS ON THE CRACK SURFACE

The objective of this chapter is to apply the results of Chapter 3 to produce a more sophisticated machine learning pipeline than that presented in Chapter 2. Now that the relevance of the features has been determined in a data-driven fashion, the more relevant features can confidently be used for prediction. First, the dimensionality of the relevant features is reduced to make the computation more feasible. Then, appropriate machine learning models are chosen that can utilize the spatial-relations of the features belonging to each point in the microstructure. Finally, prediction strategies are presented that mirror the actual growth behavior of a microstructurally small fatigue-crack.

4.1 Data preprocessing

Using the results from the previous chapter, a new representation of the data can be found, one that is both highly informative as well as amenable to spatial-relation based-learning algorithms. To achieve this representation, the existing data are transformed into a new domain that has a low dimensionality for each point in the 3D microstructure. As an analogy, consider how an image is represented in a low dimensional color space (such as RGB) and contains simply a 3-vector at each of the pixels on a 2D plane. This allows for feasible convolutions and other operations during training. In this work, the dimensionality of the points in the microstructure is reduced from 88 (as in Chapter 3) to 3, while still retaining the high-value information for predicting crack propagation.

The results from the previous chapter provided insight into which micromechanical features were relevant and meaningful in terms of correlation with crack growth. These insights help determine the best way to restructure the data into a more manageable format with minimal loss of information. This restructuring is imperative to the feasibility of

efficient model training; using all 88 features of all 1.7235×10^8 points would render the computation intractable for the given hardware constraints. It is recognized that many of the features are related to each other and thus should not be treated as independent variables in the context of the machine-learning algorithm.

The first step to reducing the dimensionality is to recognize that, in terms of correlation with crack path, the spatial-gradients of the micromechanical features (Figures 3.15c, 3.15d in Chapter 3) are more strongly correlated than the micromechanical features, themselves (Figure 3.15a, 3.15b in Chapter 3). Furthermore, of the 44 spatial-gradient features, the 22 associated with cyclic change between loading and unloading (Figure 3.15d from Chapter 3) have slightly greater values of Pearson correlation coefficients. Thus, the feature set considered for input to the machine-learning models is down-selected to the 22 $\nabla(\Delta\lambda)$ features identified in Chapter 3.

Next, principal component analysis (PCA) is applied to this set of 22 features to further reduce the dimensionality of the feature set. As described in Section 1.2.1, PCA provides the orthogonal basis vectors in decreasing order of the amount of variance explained along their directions. Prior to running PCA, the features are normalized by subtracting the mean and dividing by the standard deviation, such that the units are of the same order and magnitude. Results from the PCA show that the first three modes happen to explain 96% of the total variance (Figure 4.1), and those three are selected as the basis for the new data. In other words, the 22 features at each point in the microstructure are projected to this basis, resulting in a 3-vector based on the micromechanical fields at each point.

In addition to the features based on micromechanical fields, an additional feature is considered based purely on the geometrical configuration of the microstructure. Experimental data (both from this work and reported throughout the literature) suggest that microstructurally small fatigue-cracks behave much differently near grain boundaries than within a single grain. It is postulated that the fatigue-crack investigated in this work could have been influenced by certain field variables when near a grain boundary, and others when far from a grain boundary. To account for such spatially dependent correlation, an additional feature called d_{GB} , which is the distance from a given point to the nearest grain boundary, is added to each PCA 3-vector to produce a 4-vector. This 4-vector is called a descriptor, and is used as the “raw” representation from which the fracture surface is

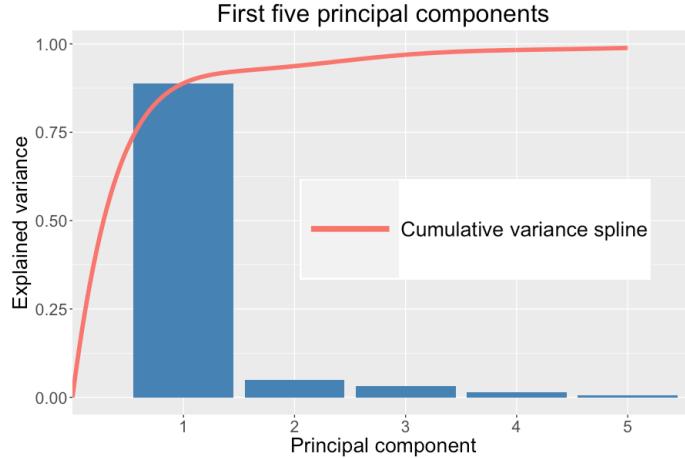


Figure 4.1: The blue bars represent the fraction of variance explained by each of the first five principal components. The red is the cumulative amount of variance explained by the first five principal components, interpolated between the bars to form a smooth curve

inferred.

4.2 General strategy

Once the data are preprocessed, the objective is to train a machine-learning model to infer the crack surface as a function of the descriptor described above. As a first attempt, a simple prediction/propagation strategy is implemented. This strategy is model-agnostic, and will be described without mentioning the specific models chosen. The entire crack surface described in the previous chapter is split into two equal halves, the front half (the half containing the nucleation point of the crack) and the back half. The front half of the surface is used to train the machine-learning model, and the back half is predicted using the trained model. The predicted surface is validated by comparing to the known crack surface on the back half of the sample. The prediction for a single data point is defined as follows:

1. For a given point on the current crack surface, take all descriptors within the $\Delta x \times \Delta y \times \Delta z \mu m$ region centered at this point. Since the microstructure is discretized onto a grid with $1 \mu m$ spacing, this region will contain $\Delta x \times \Delta y \times \Delta z$ descriptors.
2. Flatten these descriptors into a feature vector of length $4 \times \Delta x \times \Delta y \times \Delta z$.
3. Given a feature vector, use the trained model to predict the relative z-offset to the

next point in the x-direction (assumed as the approximate growth direction of the crack).

4. Use a radial basis function (RBF) smoothing spline to ensure that rows of predicted points form a smooth surface. The purpose of the spline is to bring the values of independently predicted adjacent points closer together. This spline is used in addition to the predicted values, and is part of the entire machine learning pipeline. It can be thought of as a type of prior, ensuring that adjacent independently-predicted points on the crack surface do not diverge in a physically unrealistic manner.

Here, the response variable is the z-offset, which represents the vertical distance by which the crack deviates from its current position during propagation. Specifically, the z-offset for a particular point on the crack surface is the signed vertical distance between itself and the next point in the positive x-direction. Since the crack surface is assumed to emulate an injective function $f(x, y)$, the problem of predicting the shape of the crack surface is reduced down to predicting the z-value for a given point (x, y) . If the z-offset is predicted for the point $\{x, y\}$ then the z-value for the point $\{x + 1, y\}$ can be calculated by adding the predicted z-offset to the z-value of the point $\{x, y\}$.

The features used for prediction are the descriptors for the points belonging to a region surrounding the point for which the z-offset is being predicted. The size of this region is a hyperparameter that is modified during model comparison, the largest attempted value being $10\mu\text{m} \times 10\mu\text{m} \times 10\mu\text{m}$ (such a region produces $10 \times 10 \times 10 \times 4 = 4000$ features for a given point).

Because the prediction of a particular point requires the known value of the prior crack surface point, the prediction methodology implemented here closely emulates the actual sequence of crack evolution in a physical sample. To better understand this, consider that the model has trained on all points in the crack surface from $0 \dots n$, where n is the halfway point in the x-direction. At the beginning of the testing phase, the model uses the feature vectors associated with the points at n to predict the z-positions of the points at $n + 1$. Then, the feature vectors associated with the predicted z-positions for the points at $n + 1$ are used to predict the z-positions of the points at $n + 2$. This continues until the entire crack surface had been predicted.

While errors can potentially propagate throughout the testing phase due to dependence on previous predictions, this prediction strategy closely mirrors the actual real-world use case for predicting crack growth. Specifically, the model needs to be able to perform well under the assumption that large portions of the crack surface are unknown, where the most common case is when the unknown portions are in the path of future crack growth.

Both R^2 and root mean squared error (RMSE) are used as error metrics and are calculated by using the difference in actual and predicted z-values. The difficulty in measuring model accuracy for this strategy is that predicted points closer to the training points generally have lower errors, while predicted points farther from the training points had higher errors. While using the R^2 and RMSE for all testing points helps combat this problem, there is still a need for a sense of relative success of the model. Therefore, much like in the last chapter, a hypothetical crack surface is created against which predicted crack surfaces are compared. This hypothetical crack surface simply uses the z-values of the crack surface points where the training data ended, and extends those z-values forward in the x-direction. This is the most sensible approach, since the distribution of relative z-offsets is a normal distribution with mean zero, as shown in Figure 4.2.

To provide further comparison, a simpler set of features is also investigated. Instead of using the PCA 3-vector described above, the descriptor for comparison contains simply the Euler angles representing the grain orientation at each point. This is a worthwhile exercise, in that Euler angles are intrinsic properties of the material whose acquisition requires no

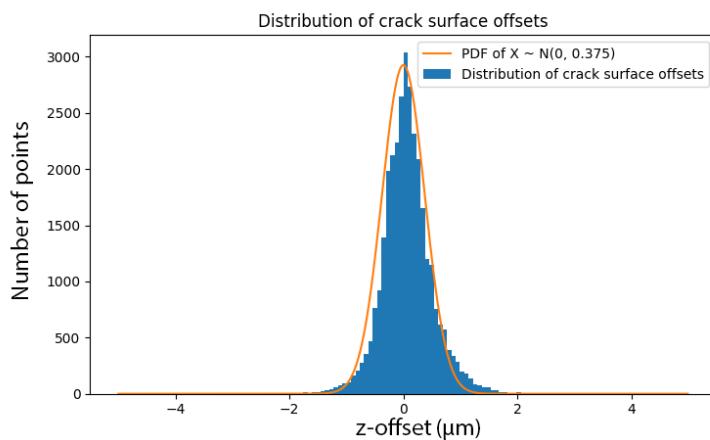


Figure 4.2: Histogram of the distribution of z-offset values

simulation, as described in Chapter 3. The rationale behind using the Euler angles as features is to assess how well the model can predict crack growth without load-dependent features such as stress and strain. As before, d_{GB} is appended to each 3-vector to produce a 4-vector.

4.3 Crack surface height inference

In this work, two machine-learning models are considered for the prediction of the response variable. The first chosen model is a convolutional neural network (CNN), in accordance with the assumption that spatial-relation based algorithms might perform well on this representation of the data. The CNN is implemented by the Keras library for Python [1]. The structure of the CNN is shown in Figure 4.3.

The CNN was trained for either 5 or 10 epochs. Beyond 10 epochs, the model did not improve much, and was at risk for overfitting. The second chosen model is the XGBoost library [2], which uses a random forest based model. This model contains many default parameters, but the most significant are likely the number of trees (100) and maximum tree depth (6).

Table 4.1 compares the results of each model. Figure 4.4 shows the resulting surface for a selection of the rows from the table. The front half of the crack surface is constant in all images, since it was treated as “known” and used as the training data. The back half is different across all images, and represents the area predicted by the model.

The base performance achieved by simply “extending” the crack surface is shown in the last row of Table 4.1. Notice that most of the trained models surpass this performance, which is promising. As hypothesized, the CNN seems to be the most accurate in its predictions, as indicated by the first row in the table. This reaffirms the notion that models

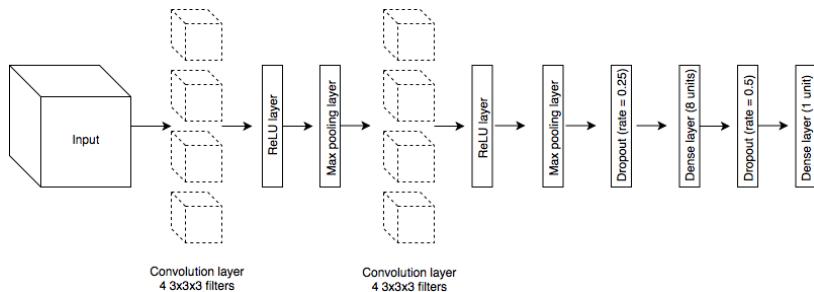


Figure 4.3: Structure of the convolutional neural network used in this chapter

Table 4.1: Crack surface height inference results for each chosen model

Model	Features	XY-distance	Z-distance	Epochs	R^2	RMSE
CNN	PCA vectors	10 μm	10 μm	10	0.891	10.156 μm
CNN	PCA vectors	10 μm	10 μm	5	0.789	14.131 μm
CNN	Euler angles	10 μm	10 μm	10	0.844	12.156 μm
XGBoost	PCA vectors	10 μm	10 μm	N/A	0.888	10.314 μm
XGBoost	PCA vectors	6 μm	6 μm	N/A	0.875	10.873 μm
XGBoost	Euler angles	10 μm	10 μm	N/A	0.669	17.690 μm
Hypothetical crack surface					0.788	14.168 μm

which encode spatial relationships in the microstructure are capable of achieving greater success. When using PCA vectors as features, both models attain higher accuracy than when using Euler angles. However, this disparity is smaller in the case of the CNN, indicating again that Euler angles are much more meaningful when spatially associated with grains. This makes sense when considering that the CNN may be learning filters based on the locations of the grain boundaries, which are implicitly encoded in the Euler angles. Another interesting observation is that decreasing the size of the region from which descriptors are sampled for a given point from $6 \times 6 \times 6$ to $10 \times 10 \times 10$ does not have a huge impact on performance. This shows that increasing the area gives diminishing returns, which is a useful observation to anyone concerned about computation.

The images show that all models make relatively safe predictions, i.e. that the models are reluctant to predict high z-offset values, which results in smoother surfaces. This is one area where the chosen prediction strategy is lacking. Prediction of anomalies is highly unlikely under this particular strategy.

4.4 Crack-surface height inference based on a radial prediction strategy

Another prediction strategy is implemented that more closely emulates the actual growth behavior of the observed crack. Rather than training on the front half and propagating towards the back half of the sample in the x-direction, the method presented here trains on a semicircular area emanating outward from the nucleation point of the crack. The prediction then attempts to expand this area radially outward. This is done by moving point-wise along a semi-circular “known” region of the crack surface (which began as simply the training region) and predicting the z-offset for three of the adjacent points on

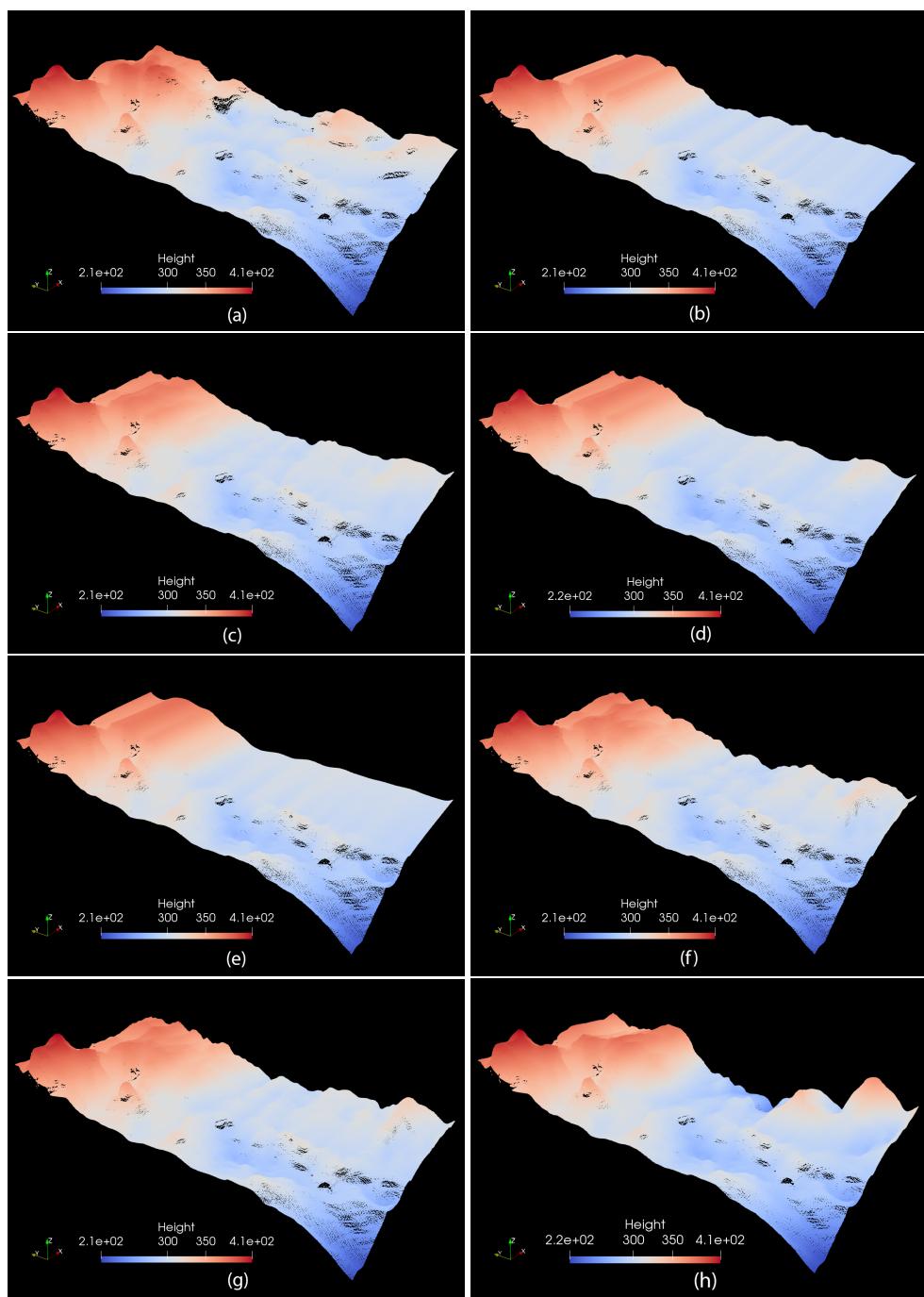


Figure 4.4: Actual and predicted crack surfaces (a) Actual crack surface (b) Hypothetical crack surface (c) CNN prediction, using PCA vectors and a $10 \times 10 \times 10 \mu\text{m}$ region (trained for 10 epochs) (d) CNN prediction, using PCA vectors and a $10 \times 10 \times 10 \mu\text{m}$ region (trained for 5 epochs) (e) CNN prediction, using Euler angles and a $10 \times 10 \times 10 \mu\text{m}$ region (trained for 10 epochs) (f) XGBoost prediction, using PCA vectors and a $10 \times 10 \times 10 \mu\text{m}$ region (g) XGBoost prediction, using PCA vectors and a $6 \times 6 \times 6 \mu\text{m}$ region (h) XGBoost prediction, using Euler angles and a $10 \times 10 \times 10 \mu\text{m}$ region

the xy-plane, as shown in Figure 4.5. If any of the z-values for these points are already known, the predicted value is ignored. Otherwise, the predicted value is used to produce a z-coordinate for the point. If the predictions of multiple known points overlap, then the mean of all predictions is taken. Once a prediction for the z-coordinate is produced, the point is added to the known region. This continues until the z-coordinates of all points within the area of the final crack front have been predicted.

The benefit of this strategy is that it more closely resembles the actual shape of the experimentally observed crack fronts. An additional benefit of this approach is that fatigue-crack growth rate, $\frac{da}{dN}$, was previously parameterized in the same manner (viz., along radial paths from the nucleation site). Consequently, $\frac{da}{dN}$ could be considered in future work as the prediction metric to complement the prediction of crack-surface height.

The wedge prediction strategy from Chapter 2 is also implemented here, as it has merit in inferring portions of the crack surface where measurements are produced with high levels of uncertainty. In this case, the area containing the crack fronts is split into 6 wedges of equal angular degree, and the 3 wedges that contain the highest percentage of uncertain data are used as the testing portions. A selection of the results for models predicting z-offset are given in Table 4.2, with images given in Figure 4.6. For comparison, SVR is used as one of the models, since it had been used in a similar fashion in Chapter 2.

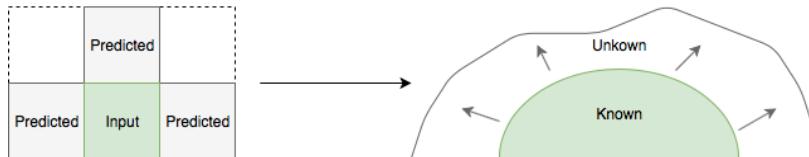


Figure 4.5: Visualization of the radial prediction strategy. On the left is shown the points for which the predicted z-values of a single point correspond. On the right is the desired effect of this prediction strategy.

Table 4.2: Crack surface height inference results for each chosen model, using a radial propagation strategy

Model	Features	XY-distance	Z-distance	Split	R ²	RMSE
XGBoost	PCA vectors	10 μm	10 μm	Radial	0.784	9.344 μm
XGBoost	PCA vectors	10 μm	10 μm	Wedge	0.579	12.216 μm
SVR	PCA vectors	10 μm	10 μm	Radial	0.653	11.049 μm
SVR	PCA vectors	10 μm	10 μm	Wedge	0.572	12.312 μm

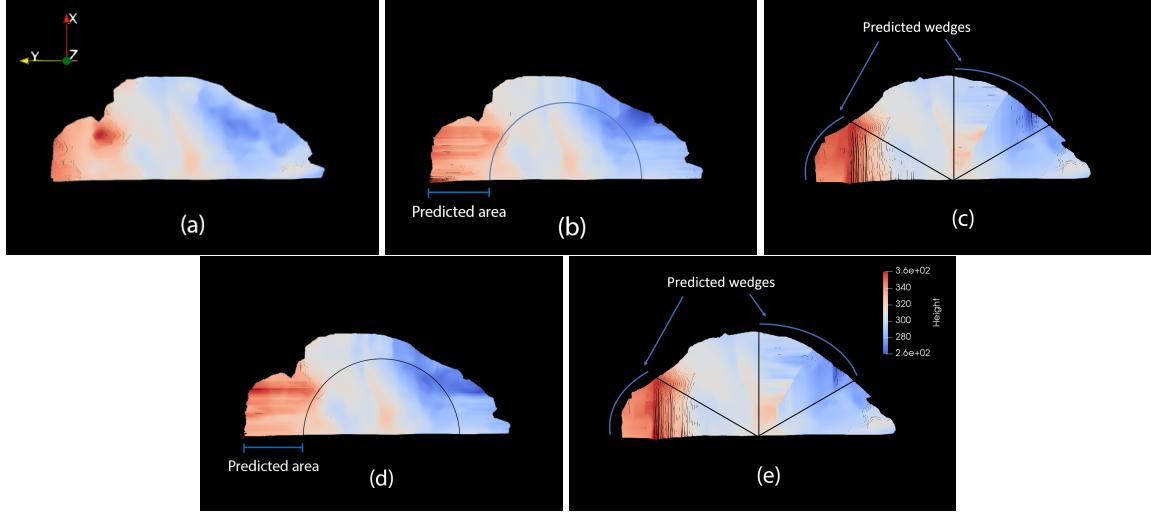


Figure 4.6: Actual and predicted radial crack surfaces (a) Actual crack surface. (b) XGBoost prediction, using a radial propagation strategy (c) XGBoost prediction, using a wedge propagation strategy (d) SVR prediction, using a radial propagation strategy (e) SVR prediction, using a wedge propagation strategy Each of these predictions have been cropped to the shape of the measured crack surface inside of the final crack front.

Interestingly, the wedge strategy performs poorly in both test cases. This is likely because the prediction strategy does not follow the actual direction of crack growth, and so predicted z-values are dependent of points that may not have chronologically preceded the predicted point. The XGBoost library outperforms SVR in both cases, suggesting that the actual crack growth function is highly nonlinear.

4.5 Conclusion

The results from this chapter are much more promising than those of Chapter 2. Nearly all of the z-offset predictions are better using a trained model vs. using the hypothetical crack surface. This shows that the trained model is more successful than not when making predictions about which direction that crack will grow. In some cases, the improvement is quite significant, reaching up to a 12.69% improvement in R^2 and a 27.20% improvement in RMSE the optimal case (row 1 of Table 4.1). Not only does this mean that the machine-learned models are capable of making relatively accurate short-term predictions, but also that they can minimize and even correct the propagation of errors in long-term predictions.

While the z-offset predictions seem meaningful, their usefulness remains unknown. Without more data, it is difficult to know how the number of training points can affect

the results, as well as how long the prediction strategy can remain accurate before small errors begin to cause large deviations. It is also difficult to determine how well this strategy generalizes to other cracks, i.e. if the training data from one crack can be used to predict the results of a completely different crack. Additionally, without the associated $\frac{da}{dN}$ information, it is impossible to know when the crack will grow to the various predicted locations. Still, knowing approximately where the crack will grow can be useful for determining when it might start to compromise the integrity of the material.

One of the drawbacks of using sophisticated machine learning methods is lack of interpretability. There are ways of analyzing certain aspects of the model to get an idea of what influences its decisions. For a CNN, this could mean examining the learned convolutional filters, whereas for a random forest, this could mean looking at which features are used as the root node in the majority of trees. However, when the CNN has multiple layers and thousands of filters, this can be challenging. Even in less complicated models, if PCA representations are used for features, any interpretation of how the original features relate to the response variable becomes much more challenging.

Predicting $\frac{da}{dN}$ remains a difficult problem. This is partly due to the few number of data points, as well as the limited number of unique crack fronts. Additionally, the extracted features are more prone to being relevant to crack point location, since the correlations were calculated with respect to distance to crack surface. Thus, the success of the z-offset predictions is actually quite promising for the potential success of $\frac{da}{dN}$ predictions. Under a different feature selection strategy, the pipeline presented in this chapter might achieve comparable success with $\frac{da}{dN}$ predictions.

Future work might explore these feature selection strategies, as well as incorporate more data. Having multiple instances of fatigue-cracks would be very useful in training models that generalize well. Beyond this, a challenging remaining problem is the prediction of anomalies in the crack surface. A data split that isolated these anomalies to see how well they could be predicted would be an interesting and enlightening endeavor.

4.6 References

- [1] F. Chollet *et al.*, “Keras,” 2015.
- [2] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of*

the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,
pp. 785–794, ACM, 2016.

TOPIC INDEX

A

Artificial neural network 5
Convolutional 6, 49

C

Cross-Validation 9
Leave-one-front-out 17
Leave-one-wedge-out 17

D

Decision tree 6

E

Electrical discharge machining 13
Electron backscatter diffraction 13
Error metrics 10, 46
 R^2 10, 46
Root mean squared error 10, 46

N

Near-field high-energy X-ray diffraction microscopy 13

P

Pearson moment-product correlation 9
Principal component analysis 8, 45

R

Random forest 8, 49

S

Scanning electron microscope 13
Support vector machine 2, 49
Regression 4

X

X-ray computed tomography 13