
Spatial growth prediction for filiform corrosion: classification models with image-derived features

BRIAN E. RALSTON and DONALD E. BROWN

*Department of Systems and Information Engineering, College of Engineering and Applied Science
University of Virginia, Charlottesville, VA 22902, USA
E-mail: ber4n@virginia.edu or deb@cms.mail.virginia.edu*

The Government Accounting Office reports that the Department of Defense spends approximately \$20 Billion per year on prevention and repair of corrosion. The resources required to combat corrosion problems are desperately needed for equipment and personnel in today's high operations tempo environment. Despite its significance, modeling for the prediction of corrosion evolution has been largely ignored in the literature. The purpose of this research is to determine if features derived from images of corrosion growth can provide insight into the corrosion process. Feature data was derived from time sequenced images of filiform growth in Al2024-T3. Logistic regression, classification trees, generalized additive, and kernel density models were considered for predicting corrosion growth from day 1 to day 2 of exposure. A model for random growth was developed to provide a metric for evaluating the model performance. The logistic regression, classification tree, and generalized additive models all had similar performance in predicting corrosion growth for the training data. The models had similar performance on the test data with approximately 70% recall and precision. However, when compared to the random model, they offer no improvement in predictive performance. The features derived from the images of corrosion growth are insufficient to describe the corrosion growth process. Although these images are readily available to maintainers performing inspection of corrosion damage, smaller scale features should be considered to provide a more accurate prediction.

1. Introduction

Air Force Chief of Staff, General John Jumper recently reported that one of the biggest concerns facing the service in the coming years is the aging aircraft fleet [1]. In FY 2007, the average age of the U.S. Air Force aircraft will be over 25 years, and while the Air Force continues to stretch the life of each aircraft, the stress and strain on airframes builds [2]. A large percentage of commercial and military aircraft in operation today, have met or exceeded their original design life, which dramatically increases the required maintenance and repair costs [3]. As the acquisition of new equipment decreases and the age of current systems increases, corrosion is potentially the most significant driver of life-cycle cost [4]. Each year, the Air Force spends \$1-3 billion on repairs caused by corrosion or on programs to prevent the occurrence of corrosion. Corrosion prevention and maintenance is currently inadequate primarily because of the lack of understanding of the corrosion process, specifically the inability to predict the initiation and growth of corrosion in airframe components [4].

The effect of corrosion on aging aircraft is particularly applicable to the fleet of Air Force tankers. The KC-135 is the oldest aircraft in the Air Force inventory, with an average age of over 35 years. The Air Force maintains a fleet of over 500 KC-135's, and with no immediate plans for replacement, the aircraft could remain in service for another 25 years [3].

The extended life of the KC-135 and other Air Force aircraft translates into an increase in fatigue, wear, and corrosion. Corrosion decreases the overall strength of the airframe due to the loss in skin thickness, fatigue crack initiation, and increased crack growth rates [5]. The problems caused by corrosion have a direct impact on the maintenance costs and operational readiness of the fleet due to the increased time required for depot level repair [3]. In today's high operational tempo, the Air Force relies heavily on the tanker fleet to perform missions around the world. Each year, the Air Force spends between \$1-3 Billion on operations related to corrosion. The money spent on the prevention and repair of corrosion, is desperately needed for equipment and system upgrades. The situation highlights the need for accurate corrosion prediction models which can be used to develop a cost effective corrosion integrity program [5].

Currently, corrosion is not included in a damage tolerance estimate. When corrosion is found, the policy is to immediately repair [5]. The Air Force developed the structural damage management tool (SDMT) as a decision support system for corrosion prediction. Non-destructive inspection (NDI) methods, such as eddy current scans, produce an image of suspected material loss due to corrosion. These images are input into SDMT, where an operator selects a line of pixels through a region of suspected corrosion. SDMT produces a distribution of the amount of

material loss along the line of pixels and predicts how this distribution will change for a given basing strategy. The prediction is actually just a simple shift in the distribution determined by an assumed rate of future corrosion. The SDMT operator can use this prediction to determine the likelihood of reaching a given threshold of material loss. The mean value of the predicted distribution is then output to a fracture mechanics application to analyze crack growth. Despite the spatial nature of the data in the NDI images, all spatial information is discarded when material loss along a single line is considered. SDMT is only concerned with a line of pixels because of the stress loads that affect how the crack propagates. This methodology corresponds to an assumption that the corrosion rate, and thus the predicted amount of material loss, at a given pixel is independent of the corrosion that exists at other pixels.

In this paper, we focus on the information available in images of corrosion growth. Brenc and Brown [6] have taken the first step in improving the ability of operators to manage corrosion maintenance by developing a regression tree model to classify corrosion from NDI images. This removes the subjectivity of corrosion identification. Our goal is to determine if the features which can be derived from these images can be used to predict the spatial growth of corrosion. An accurate prediction of future corrosion based solely on the current picture of material loss would be invaluable to aircraft maintainers making corrosion repair decisions in the field. In Section 2 we describe the process of data collection from the time sequenced images of corrosion growth. In Section 3 we discuss the various models considered and the training methodology for fitting the model parameters. The models are then evaluated in Section 4 against an assumption of random growth. We provide conclusions and insights in Section 5.

2. Image Processing

The data used in this research is processed from time-sequenced images of filiform corrosion growth in aluminum. Filiform corrosion occurs under the surface of coatings such as a paint primer. Corrosion filaments propagate in a worm-like fashion delaminating the protective coating. Three 2" x 2" samples of Al2024-T3 were polished to 1200 grit and coated with epoxy polyamide. The coating was then scribed across the center of the sample and exposed to concentrated HCl for 30 seconds to initiate filiform growth. The samples were kept at 40°C and 80% relative humidity.

Data was derived from images of filiform growth after 24 and 48 hours of exposure. The images are 3.1 mega pixel JPEG files with 8-bit color depth. The images were cropped to capture only the areas around the scribe where filiform corrosion occurs. Two of the samples were used for training data, and sample 3 was kept as a test set. Fig. 2 shows the time sequenced images for sample 3. The top two are the initial cropped images, while the bottom images show the result of registration and segmentation.

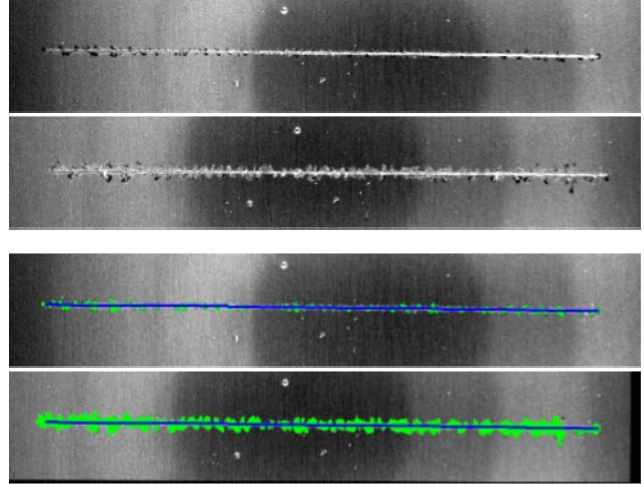


Fig. 1. Time sequenced images of sample 3. The top 2 images show the initial image at $t=1$ and $t=2$. The bottom 2 images show the result of image registration and segmentation. The scribe is denoted by blue pixels, and corrosion is shown in green.

2.1. Image Registration and Segmentation

Our approach requires feature values for each pixel in time-sequenced images of corrosion growth. For non-corroded pixels, we use the pixel's feature values at time $t=1$ to predict the possibility of corrosion at $t=2$. This approach requires a one-to-one mapping of pixels to ensure consistency from $t=1$ to $t=2$.

Image registration was performed using Matlab to first select control points in each image. The coating contains numerous small inclusions which are visible in the images as small high intensity objects. These objects provide consistent control points for image registration. 8-10 pairs of control points were used to register images for each sample. Once the control points were selected, a linear conformal transformation was used to describe the change in the image from $t=1$ to $t=2$. The linear conformal transformation is appropriate because it can account for distortion through combinations of translation, rotation, and scaling of the image. Using this transformation, the images at $t=2$ were registered to the corresponding image at $t=1$.

The value of the response variable at each pixel (row, column) location, $Y_{r,c}$, is determined by the classification of the pixel in the image at $t=2$. Therefore, $Y_{r,c} = 1$ if the pixel at location (r,c) becomes corroded and $Y_{r,c} = 0$ otherwise. To determine this classification, the image had to be segmented to identify pixels belonging to corrosion objects from non-corroded pixels. In addition, the scribe had to be identified so only those pixels that are candidates for corrosion will be considered by the model.

Image segmentation seeks to partition the pixels in the image as belonging to various image objects (background, corrosion, or scribe). There are many appropriate techniques for segmenting an image. Two of the most popular methods are thresholding and edge detection [7]. Thresholding seeks

to classify pixels based on their intensity value, whereas edge detection classifies based on the change in intensity value in comparison to neighboring pixels. Thresholding and edge detection functions in Matlab were used to identify the scribe and corrosion objects in the images. Because only three samples were used, after the thresholding and edge detection were complete, each image was classified by hand to further ensure accuracy of the initial segmentation.

2.2. Derived Features

After the images were processed, the feature values for each observation had to be derived. The most obvious features are the distance from each pixel to the scribe, and the distance to the nearest initiation point. The initiation locations are classified as the center corroded pixel along the scribe of each corrosion object. In general, we would expect pixels closer to the scribe and/or to an initiation point to be more likely to corrode than those that are farther away. Fig. 2 illustrates the calculation of the distance to the scribe ($a=6$) and the distance to the closest initiation point ($b=11.8$) for a given pixel in the test data set.

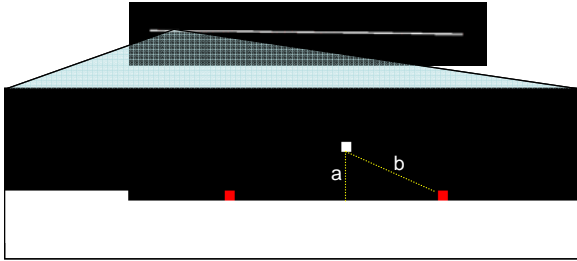


Fig. 2. Calculation of location-based features. The number of features is limited when no corrosion is present. The only features available are the distance to the scribe, and the distance (or angle) to nearest initiation point.

Once corrosion occurs, the number of possible derived features increases. Features were considered for use based on assumptions about the growth patterns of filiform corrosion. Fig. 3 explains the majority of the geometry-based features which can capture the directional growth and interdependence of corroded pixels. Other features were also considered to determine if the size, shape, and orientations of the nearest corroded object influenced the growth of corrosion. A total of 19 features were derived for use as predictor variables for filiform growth from $t=1$ to $t=2$.

The initial analysis of the data provided some useful insights into the process. Fig. 4 compares the distribution of feature values for corroded and non-corroded pixels at $t=2$ for 3 of the feature variables. The majority of corrosion at $t=2$ occurs within a distance of 35 from the scribe. Roughly half of the corrosion occurs where small gaps exist between current corrosion, but the other half occurs where no gap exists. The angle features are more difficult to analyze because pixels that are farther away do not corrode and have a tendency to be at an angle close to $\pi/2$. Corroded pixels are also more prevalent at $\pi/2$, but are fairly uniform in the tails.

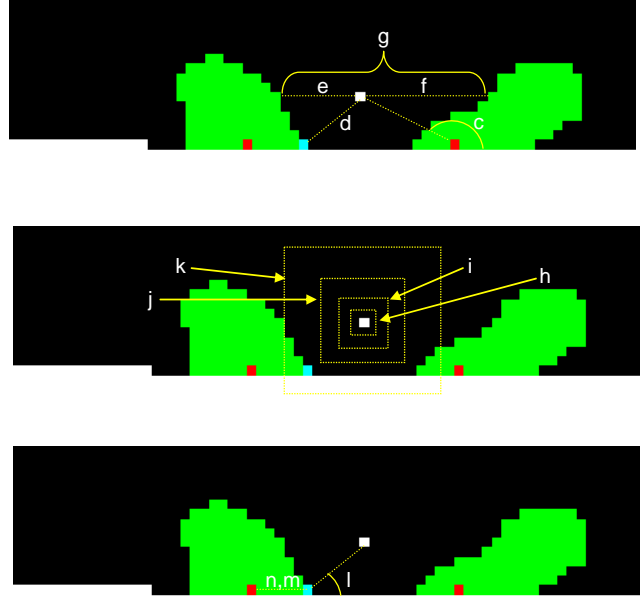


Fig. 3. Calculation of geometric-based features. Angle to the nearest initiation point ($c=2.68$), distance to the nearest corroded pixel ($d=7.81$), distance left to a corroded pixel ($e=9$), distance right to a corroded pixel ($f=14$), and the resulting gap distance ($g=23$). Number of corroded pixels within a 3×3 region ($h=0$), 5×5 region ($i=0$), 9×9 region ($j=0$), and 17×17 region ($k=13$). Angle from the nearest corroded point to the pixel ($l=0.69$), angle from the initiation point to the nearest corroded pixel ($m=0$), and the distance from the initiation point to the nearest corroded pixel ($n=6$).

Unfortunately, an obvious linear discriminant in one or more of the variable is not evident from the initial analysis of the feature data.

3. Predictive Models

The features discussed in the previous section result in a vector of predictor variables $X_{r,c}^{(t)} = \{x1_{r,c}^{(t)}, x2_{r,c}^{(t)}, \dots, x19_{r,c}^{(t)}\}$ at pixel location (r,c) and at time t . The general form of the predictive model becomes:

$$\hat{Y}_{r,c}^{(t+1)} = f(X_{r,c}^{(t)}) \quad (1)$$

where the predicted binary response for pixel location (r,c) at time $(t+1)$ is a function of the feature values at that location at time t . The binary response is 1 if the pixel becomes corroded and 0 otherwise.

The overall goal is to determine the best set of features and the appropriate functional form to accurately describe the corrosion growth process. There are many supervised learning methods available for modeling the occurrence of corrosion as a function of the feature values. An important consideration in model selection, however, is the large data set for this application. Because the data is derived from images where each pixel represents an observation, the training set has 19 variables and over 400,000 data points.

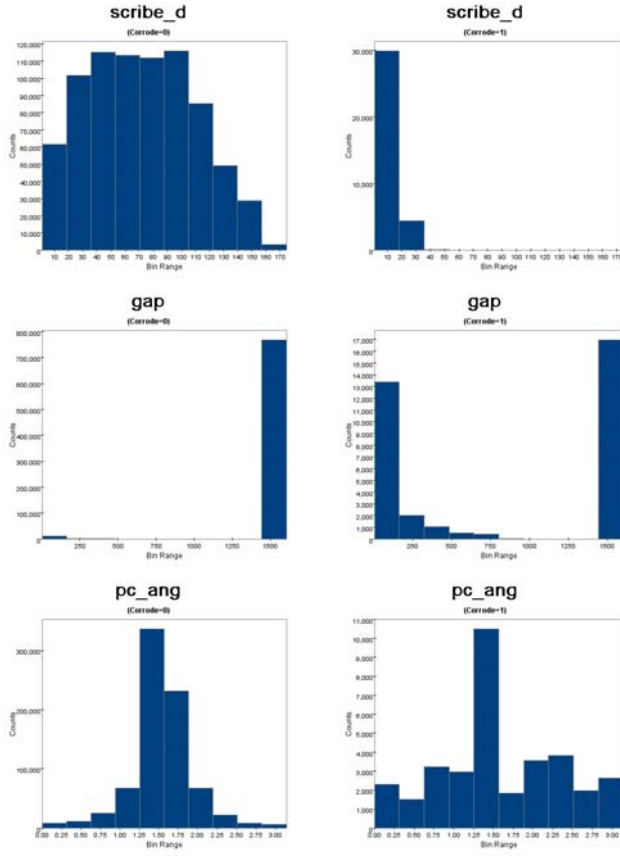


Fig. 4. Histograms of feature data by class. The initial analysis of data shows some differences in corroded and non-corroded pixels, but an obvious discriminant is not apparent. Data for the distance to the scribe (*scribe_d*), the left and right gap distance (*gap*), and the angle from the pixel to the nearest corroded point (*pc_ang*) is shown here.

The models considered for this application range from simple linear logistic regression to more complex kernel density classifiers. There are many examples in the literature covering the use of these methods in various applications from inspection maintenance to ecology [8], [9], [10], [11]. Logistic regression, classification trees, and generalized additive models were implemented using the Insightful Miner software which runs with SPlus. Kernel density estimation was implemented using Matlab. Each model was fit using the training data from samples 1 and 2.

3.1. Logistic Regression

Logistic regression seeks to model the posterior probabilities of classes as linear functions of the predictor variables. Typically, logistic regression is used to understand the relationship between the predictor variables and the response. Although logistic regression models the

response as a linear function of the predictors, nonlinearities can be accounted for using interaction terms between the variables. The general form of the logistic regression model is

$$\log\left[\frac{\Pr(\hat{Y}_{r,c}^{(t+1)} = 1)}{1 - \Pr(\hat{Y}_{r,c}^{(t+1)} = 1)}\right] = X_{r,c}^{(t)}\beta. \quad (2)$$

All of the features described in the previous section were included in the model as well as all 2-way and selected 3-way interaction terms. The coefficients for the predictor variables are estimated using maximum likelihood. A Wald statistic was calculated for each of the predictor variables to determine the significance of including each term. The Wald statistic is the square of the t-statistic and generally provides a better measure for how the variable contributes to the model as a whole. Although problems of bias with the Wald statistic have been published [12], the coefficients for this application are small and the availability of the statistic make its use acceptable. The Wald statistic suggested very little reduction in predictor variables. Some of the most significant variables included the distance from initiation point, distance to nearest corrosion point, angle to nearest corrosion point, the number of pixels within a 17x17 region, and the angle between the nearest corrosion point and the initiation point.

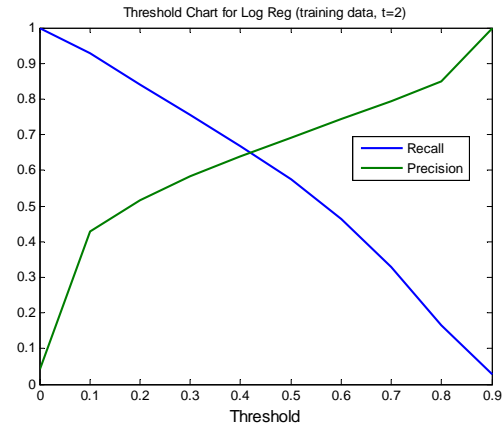


Fig. 5. Threshold chart for logistic regression. A threshold value is determined from the training data in order to classify pixels using a logistic regression of features. A threshold of approximately 0.4 is suggested in order to maximize both recall and precision.

The logistic regression produces a probability of corrosion value for each pixel observation. A threshold value must be selected in order to classify the pixel as corroded or not. Low threshold values result in a higher false positive rate and high thresholds produce high false negative rates. Because we are primarily concerned with identifying corrosion, recall and precision are used as metrics. Recall represents the percentage of truly corroded pixels that are accurately classified. Precision represents the percentage of predicted corroded pixels that are accurately classified. Fig. 5 shows how the recall and precision values change as the threshold is

increased. This chart suggests an optimal threshold of approximately 40% to maximize recall and precision simultaneously. Therefore, the training data suggests that the logistic regression model will result in approximately 65% for recall and precision.

3.2. Classification and Regression Trees

Like logistic regression, classification trees are a simplistic yet powerful method for performing feature selection. Classification trees are grown by recursively partitioning the feature space, and then assigning a classification to each. The feature and split point for each partition are chosen based on a predetermined figure of merit, such as the Gini index or measure of entropy. Both the Gini index and entropy measure the impurity at each node in the tree. The measures estimate the probabilities for each class using the proportions at each node. In general, the Gini index results in large splits, while the entropy metric favors more balanced splits [14].

One of the major drawbacks of classification trees is the inherent high variance in the results. Typically, small changes in the input data can result in dramatic changes to the resulting tree [14]. One method of variance reduction in trees is called bagging. Bagging uses a combination, or ensemble, of trees and averages the results to produce more stable predictions.

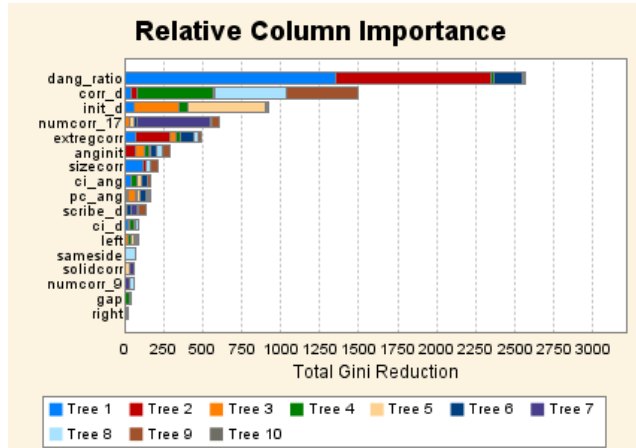


Fig. 6. Variable importance from classification tree. Each variable's contribution in reducing the value of the Gini index is shown for an ensemble of 10 classification trees.

For this study, ensemble trees were constructed using the training data. Both the Gini index and entropy metrics were considered, but the Gini index provided better accuracy results for the training data, therefore, the entropy metric was not used further. The amount of impurity that is removed by splitting on a given feature provides a metric for the significance of that feature in predicting the response classification. Fig. 6 shows the relative importance of the features used in the ensemble trees. The derived feature

dang_ratio, as well as the distance to nearest corrosion (corr_d) and distance to nearest initiation point (init_d) provided the largest reduction in Gini index score. These results are somewhat similar to the logistic regression coefficients, although the order of significance is slightly changed. The classification tree achieved 61% and 72% in recall and precision for the training data.

3.3. Generalized Additive Models

Typically, the relationship between predictor and response variables is nonlinear. Without prior knowledge of nonlinearities and transforming the predictor variables, the logistic regression model cannot capture this relationship. Generalized additive models (GAM) provide a more straightforward method for characterizing nonlinear regression effects [14]. Fig. 7 shows a sample of the nonlinearities that exist in the training data.

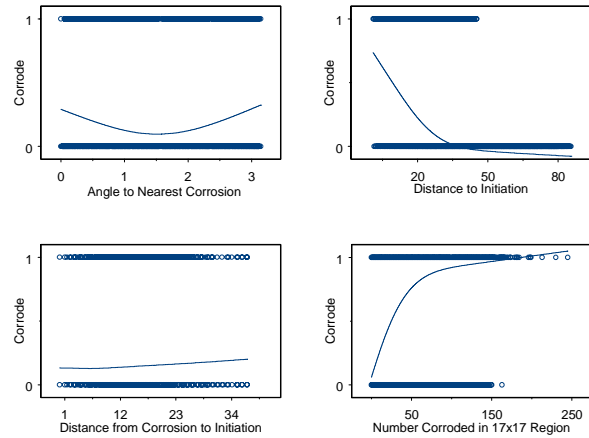


Fig. 7. Spline estimates for predictor variables. The plots show the relationship between the 4 most significant predictor variables (using the Wald statistic) and the response. The smoothing spline function is used to estimate the non linear relationships.

Like logistic regression, the GAM seeks to predict the value of the response variable as an additive function of the predictors. However, a smooth function is fit to each predictor variable which results in the more general form

$$\log\left[\frac{\Pr(\hat{Y}_{r,c}^{(t+1)} = 1)}{1 - \Pr(\hat{Y}_{r,c}^{(t+1)} = 1)}\right] = f_1(x1_{r,c}^{(t)}) + \dots + f_{19}(x19_{r,c}^{(t)}) \quad (3)$$

All of the main effects were included in the GAM. A cubic spline with 3 degrees of freedom was used as the functional form for each predictor. The results with the training data once again showed that all of the predictor variables were significant at the $\alpha=0.01$ level. The results for GAM in terms of recall and precision were 71% and 66%.

3.4. Kernel Density Estimates

Another unsupervised learning procedure which can be used for classification is kernel density estimation (KDE). KDE seeks to estimate the probability density, $f_X(x)$, for some random sample x_1, \dots, x_N by averaging a known density function across the data observations. The averaging of the kernels produces a smooth approximation to the true density.

The major drawback to KDE is its reliance on the entire training data set. The KDE model is the data set, and fitting of the model is done at evaluation [14]. Because of computational visualization considerations, we chose to reduce the dimension of the data set. Principal component analysis (PCA) provides an accepted method for shrinking the number of features and can simplify KDE [9], [13]. In addition, it ensures that regardless of any correlation in the original feature variables, the new principal component features will be uncorrelated. The first 3 principal components were considered for this data. Cumulatively, they explain 63% of the variability in the data.

The kernel density estimates were fit to the 3 principal components using Gaussian kernels. The most significant factor in KDE is the selection of the bandwidth, h . We used a basic normal scale estimate [13] which is a function of the sample standard deviation, $\hat{\sigma}$, and the sample size, N . For a Gaussian kernel, the bandwidth is:

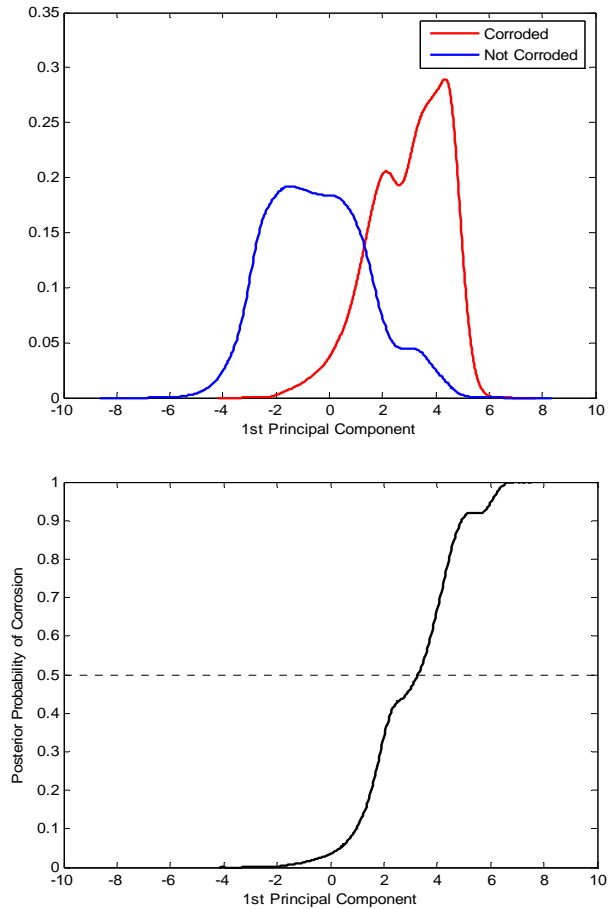
$$h = 1.06\hat{\sigma}N^{-1/5}. \quad (4)$$

KDE was used to estimate the population class density for pixels being corroded and not corroded. Fig. 8 shows the class densities in one dimension. However, to determine the posterior probability of a pixel corroding, we need to adjust for the prior probability of corrosion. In this case the sample proportion of corroded pixels was about 15%. The posterior probabilities can be calculated using Bayes' theorem. The one-dimensional posterior probability shown in Fig. 8 suggests that observations with 1st principal component values > 3 should be classified as corroded.

We calculated the posterior probabilities from the 3-dimensional kernel density estimates. Pixels in the training data were classified as corroded when the posterior probability > 0.5 . These results were then compared to the true class labels. The results for KDE were significantly lower than the other methods considered. The recall and precision values were 43% and 76%.

4. Model Evaluation

Based on the performance with the training data set, the logistic regression, classification tree, and GAM were considered for model choices. All of these models had similar recall and precision results and were relatively efficient given the size of the data. Each fitted model was used to predict the growth of corrosion on the test data sample. Because of the low incidence of corrosion, we are



primarily concerned with accurate prediction of corroded pixels.

Fig. 8. One-dimensional kernel density estimates. The population class densities for corroded and non corroded pixels show possible discriminant areas. The posterior probability of corrosion shows how classification can be done using KDE.

4.1. Random Comparison

Because of the lack of a suitable baseline model, an informed random model was developed as an additional measure of comparison. The random model uses the corrosion rate calculated from the training data to determine the number of pixels that will corrode in the test data. The rate of corrosion for the training data was 6.23% which results in 31,127 predicted corroded pixels for the test data. The random model defines eligible pixels as those that neighbor a currently corroded pixel. The model then randomly selects a single eligible pixel to corrode. Then the set of eligible pixels is updated and the process continues until the desired number of corroded pixels is reached. The result is predicted growth that is uniformly mounded around the current corrosion.

Multiple iterations of the random model were run to create a credible interval of random corrosion. A probability field for corrosion growth can be deduced using the number of times a given pixel is randomly selected to corrode divided by the total number of iterations. Fig. 9 shows the results of 20 iterations of the random model on the left end of the test sample. Greater blue intensity represents pixels that are corroded in more iterations (i.e. more likely to corrode according to the random model).

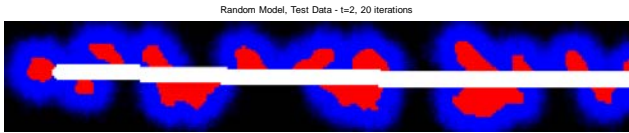


Fig. 9. Random growth simulation results. The figure shows the estimated random growth density for a portion of sample 3 (test data). The higher intensity blue pixels represent a greater likelihood for corrosion based on 20 iterations of the random model.

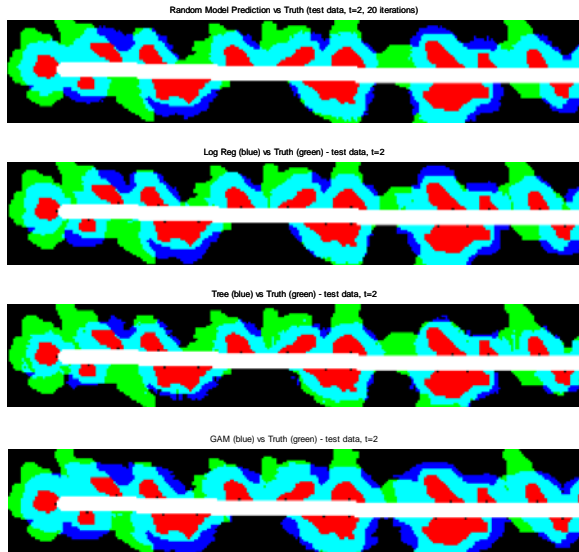


Fig. 10. Predicted model results vs. actual growth. Red denotes previous growth, light blue shows correctly classified corrosion, green and dark blue denote type I and II errors.

4.2. Predictive Results

The fitted models were evaluated based on their ability to predict corrosion growth on the sample 3 test data. Fig. 10 shows a portion of the sample with a comparison of predicted and true corrosion growth. Table 1 provides the recall and precision classification results for the test data. The F-measure provides a method for combining the recall and precision values that is more sensitive to differences than a simple average. The F-measure is calculated as

$$2(\text{Recall})(\text{Precision})/(\text{Recall} + \text{Precision}). \quad (5)$$

As expected, the random model predicts uniform growth around the current corrosion, but is unable to capture areas where the filament “fingers” in a particular direction. The other models, however, also failed to capture areas of directed growth. The logistic regression, tree, and GAM models appear to capture separation between corrosion objects better than the random model which tends to force the objects to merge together. However, the predicted image shows that the model is not a very accurate representation of the true corrosion process.

The recall and precision results quantify the results shown in the predictive images. The random model captures the least amount of true corrosion, but has the fewest type II errors. The GAM model has the highest recall, but results in the most type II errors. The GAM model had fairly high recall and precision values and in terms of the F-measure, was the best predictor.

Table 1. Model prediction results

Model	Recall	Precision	F-Measure
Random	60.6%	75.7%	66.8%
Logistic	63.8%	72.9%	68.0%
Tree	69.6%	71.8%	70.7%
GAM	71.7%	70.7%	71.2%

4.3. Percentile Score

Another metric for model comparison is the percentile score, which allows us to statistically compare the predictive performance of the models. The percentile score has been used in similar applications to compare density estimates for event predictions [15]. Because all models produce a probability of corrosion for each pixel, we can compare the probability estimate for corroded pixels to evaluate the ability of the model to capture growth. The percentile score for a given corroded pixel is the percentage of pixels with a lower estimated probability value. We can compare the average percentile score for corroded pixels for the various models to evaluate how well the model performs in assessing corrosion growth. More accurate models will produce a higher value for percentile score at an actual corroded pixel. If s_i^g is the location of the i^{th} pixel for a corrosion image containing N pixels, and d_g is the density estimate for an arbitrary location s , then the percentile score at location s is defined as

$$p_s = (100 / N) \sum_{i=1}^N 1(d_s \geq d_{s_i^g}). \quad (6)$$

The percentile score approaches the density \leq to the predicted estimate for the actual location of corrosion initiation in the test data. Higher percentile scores indicate improved predictive ability.

To compare the percentile score, we used a paired t-test to test for the hypothesis that the mean percentile scores at corroded pixel locations were equal. Equal means imply that the models have the same predictive power. A positive

difference would infer that the given model performed better than the random model. The 95% confidence interval for the mean difference between models is provided in Table 2. The random model has a higher mean percentile score for corroded pixels than the other models considered. Therefore, the models offer no improvement in predictive ability.

Table 2. Paired t-test of percentile score difference

Comparison	95% Conf. Int.		Significance
	Lower	Upper	
Log - Random	-2.32	-1.98	<0.01
Tree - Random	-2.29	-1.94	<0.01
GAM - Random	-2.23	-1.89	<0.01

5. Insights and future work

Despite the relatively high values for recall and precision, none of the proposed classification models adequately captured the growth process of filiform corrosion. Although the features used in the models were statistically significant, they are insufficient to describe the underlying process. If the features derived from the images produced significant improvement in predictive ability over the random model, then the filiform growth would be completely dependence driven. The shape and location features could be easily extracted from digital images and maintainers could use them to determine future growth.

Other features need to be considered in order to improve the performance and allow the model to predict the non-isotropic filament growth. There is ongoing research which suggests that the intermetallic microstructure can influence the growth of filiform corrosion. Data for the intermetallics can be captured and included in the models to explore their relationship to growth.

If the intermetallic features do not improve the performance of the models, then the analysis would need to progress to an even finer scale. Apart from the intermetallic microstructure, the other primary contributor would be the features inherent to the coating. Once the scale progresses past the microstructure, however, the ability to utilize these features at an operational level to predict corrosion growth in the field is lost.

References

1. Air Force Research Laboratory (AFRL). [2004] "Air Force Corrosion Prevention and Control Office," [Technology Horizons](#), February 2004.
2. Garamone, Jim. [2001] "As Equipment Ages, Readiness Suffers," American Forces Information Service. (Defense Link News). July 2001.
3. The National Academy Press, Aging of U.S. Air Force Aircraft [1977]. www.nap.edu/books/0309059356/html/index.html.
4. CC Technologies Laboratories Inc. [2002] Corrosion Costs and Preventative Strategies in the United States (A report for the Department of Transportation). <http://www.corrosioncost.com/pdf/techbrief.pdf>.
5. CC Technologies Laboratories Inc. [1998] Mathematical Model to Predict Fatigue Crack Initiation in Corroded Lap Joints, Draft. Dublin Ohio.
6. Brence, John R., and Donald E. Brown [2002] "Data Mining Corrosion from Eddy Current Non-Destructive Tests", *Computers & Industrial Engineering*, Vol. 43, pp. 821-840.
7. Pal, N.R. and S.K. Pal [1993], "A review of image segmentation techniques", *Pattern Recognition*. Vol 26, pp 1277-1294.
8. Yan, Jihong, Muhammed Koc, and Jay Lee [2004], "A prognostic algorithm for machine performance assessment and its application", *Production, Planning, and Control*, Vol. 15, No. 8, pp.796-801.
9. Cooley, Craig, and Steven MacEachern [1998], "Classification via kernel product estimators", *Biometrika*, Vol. 85, No. 4, pp. 823-833.
10. Bell, John F. [1996] "Application of classification trees to the habitat preference of upland birds", *Journal of Applied Statistics*, Vol. 23, No. 2-3, pp. 349-359.
11. Guisan, A., T. C. Edwards, Jr., and T. Hastie [2002], "Generalized linear and generalized additive models in studies of species distributions: setting the scene", *Ecological Modeling*, Vol. 157, pp. 89-100.
12. Menard, Scott [2001] [Applied Logistic Regression Analysis](#), 2nd Edition. Thousand Oaks, CA: Sage Publications. Series: Quantitative Applications in the Social Sciences, No. 106.
13. Scott, David W. [1992] [Multivariate Density Estimation](#). New York, NY: John Wiley & Sons, Inc.
14. Hastie, Trevor, R. Tibshirani, J. Friedman [2001] [The Elements of Statistical Learning: Data Mining, Inference, and Prediction](#). New York, NY: Springer.
15. Liu, Hua and Donald.E. Brown [2003] "Criminal incident prediction using a point-pattern based density model", *Int. Journal of Forecasting*, Vol. 19, No. 4, pp. 603-622.