Ordinary Differential Equations with Machine Learning for Prediction of Smart Composite Fracture Toughness

RELEBOHILE GEORGE QHOBOSHEANE$^{1,2}$, MUTHU RAM PRABHU ELENCEZHIAN$^{1,2}$, VAMSEE VADLAMUDI$^{1,2}$, KENNETH REIFSNIDER$^{1,2}$ and RASSEL RAIHAN$^{1,2}$

Abstract:

This work in on the development of an ordinary differential equation (ODE) model coupled with statistical methods for the prediction of fracture toughness of a magnetostrictive, piezoelectric smart self-sensing Fiber Reinforced Polymer (FRP) composite. The smart composite with sensing properties encompasses Terfenol-D alloy nanoparticles and Single Walled Carbon NanoTubes (SWCNT). To explore various configurations the of nanoparticle constituents’ effect on fracture toughness within the FRP composite, the ODE model developed within a finite element analysis (FEA) environment is considered to attain fracture observations across the solution space. The acquired FEA data is then used to feed the machine-learning (ML) algorithms to obtain composite fracture toughness predictions. A comparison and development of artificial neural networks (ANN), decision trees and support vector machines (SVM) models for FRP smart self-sensing composite fracture toughness prediction is done. Qualitative results stating if the sample has fractured or not and quantitative data giving the fracture toughness and strain energy release rate for the smart self-sensing FRP composites is attained. A comparison of all predictions from the developed models for both fracture toughness is corroborated with literature data.

Keywords: Fracture toughness, Carbon Fiber Composites, Finite Element Analysis, Machine Learning, Artificial Neural Networks, Support Vector Machines, Decision Tress

$^1$Department of Mechanical and Aerospace Engineering, The University of Texas at Arlington, Arlington, TX 76019, USA
$^2$Institute for Predictive Performance and Methodologies, The University of Texas at Arlington Research Institute, Fort Worth, TX 76118, USA
INTRODUCTION

The development of new materials using numerical methods and machine learning techniques has become prevalent in recent studies [1] [2]. Smart material with structural health monitoring (SHM) capabilities require complex characterization to achieve optimal sensitivity which is simplified by numerical modeling methods. The complexity in volume fraction control and fabrication processes of composites embedded with nanomaterial for piezoelectric and magnetostrictive properties in this work is overcome by generating the system in finite element analysis. Deformation properties in smart composite materials are modelled in an ordinary differential model (ODE) to gather data to use in machine learning models. This work therefore applies crack propagation data from an ODE model into machine learning algorithms for further predictions of the smart self-sensing composite fracture toughness properties. This is summarized in Figure 1.

![Figure 1. Crack propagation through a composite model ode data generation feed to an ML model for prediction of the smart composite fracture toughness.](image)

The development of new material and understanding of failure types in composite systems using machine-learning techniques has been studied in works like [3] [4]. Aykut et al., conducted experimental work on Castamide surface properties during tooling. From experimental work, milling tooling parameters were changed and surface properties data were collected for all the variations. This dataset created was therefore used in developing an artificial neural network (ANN) model to predict the surface roughness of this material during machining [5]. Applications of machine learning techniques have also been seen in composite manufacturing studies. Seyhan et al explore the prediction of compressive strength in vacuum assisted resin transfer molding of glass fiber reinforce polymer composites. The aim was to use
ply layup to enable the prediction of compressive strength using a three input neuron ANN with one output and two hidden neurons. The data acquired from the ANN model was compared with multi-linear regression model data and ANN proved to be best-fit model for this work [6].

Characterization of crack propagation has shown to depend on various parameters in machine learning models development. An ANN network with both sigmoid, tangent hyperbolic nonlinear logistic and linear activation functions was applied in [7] to predict crack propagation direction in aluminum alloys. This work showed how ANN accuracy was dependent on the number of input variables such as crack size; crack offset distance, crack tip distance and crack inclination with loading axis. Methods used to understand a structure’s response to fatigue loading include strain life method, S-N methods and fracture mechanics methods. Zarrabi et al. focused on the fracture mechanics method when developing an ANN model for fatigue crack length growth prediction in structural systems. Since a structure going through fatigue loading can have a crack develop from the initial crack up to a point of critical crack where the structure fractures, a function that relates the crack length with load cycles was used to develop a prediction model for the time required for crack to grow [8]. More works [9] have focused on the development of ANN using this relation. Application of ANN using structure crack parameters dataset generated in finite element modeling ABAQUS environment was carried out in [10] to predict crack propagation in aluminum plates using lamb wave signals. The developed back propagation model in this work showed to better predict the required crack properties. In the current study, COMSOL™ Multiphysics environment was used to develop the ODE model and generate crack parameters dataset. The focus was on the development of an ANN, random forest model, support vector machines (SVM) and xgboost model from the same dataset.

ORDINARY DIFFERENTIAL MODEL DEVELOPMENT

The ODE model was developed in the COMSOL Multiphysics environment. A two-dimensional model was designed for composite specimen with length of 125 mm and height of 5 mm. The model was setup for mode I fracture toughness test, therefore the designed specimens were within the mode I ASTM D5528 [11] specimen’s aspect ratio. All samples in this work were designed to have a pre-generated crack of 20 mm on the loaded edge which had both prescribed displacement of zero for the top edge and bottom edge loaded. The other end of the specimen was given zero prescribed displacement in y direction. These boundary conditions are reflected in Figure 2.
A mapped mesh distribution is also shown in Figure 2 with 27 elements at the initial crack area, 160 elements throughout the interfacial region of the specimen and 2 elements throughout the width of the specimen. Contact pairs were defined at the interfacial region of the specimen and penalty method used for the formulation of contact behavior. The penalty method creates a stiff spring between the two contact pairs known as the penalty factor which is active when the pairs overlap. The cohesive zone model used a displacement-based damage to model the mode I behavior of the composite specimens. Since the aim of this model was to develop a dataset to be used in ML models, the tensile strength at the interfacial region was randomized using a normal distribution function. This was so that the model can generate random crack propagation behavior for every computation. A stationary direct solver was used and stress distribution during crack propagation determined as shown in Figure 3.
The Von Misses stress distributions showed maximum stress concentration at the crack tip for all specimens modelled. Material properties for each specimens shown in Table I were change to match glass fiber reinforced polymer composites (GFRP), GFRP with single walled carbon nanotubes (SWCNTs), GFRP with Terfenol-D nanoparticles and GFRP with both SWCNTs and Terfenol-D nanoparticles for each model run to populate the dataset.

<table>
<thead>
<tr>
<th>Material</th>
<th>Modulus of Elasticity (GPa)</th>
<th>Poisson’s ratio</th>
<th>Density (g/cc)</th>
<th>Relative Permeability</th>
</tr>
</thead>
<tbody>
<tr>
<td>GFRP</td>
<td>46.5</td>
<td>0.2</td>
<td>2.44</td>
<td>1.4</td>
</tr>
<tr>
<td>SWCNT</td>
<td>34.65</td>
<td>0.311</td>
<td>1.9</td>
<td>100.3</td>
</tr>
<tr>
<td>Terfenol-D</td>
<td>50-90</td>
<td>0.5</td>
<td>9.25</td>
<td>2-10</td>
</tr>
</tbody>
</table>

These material types were included in the dataset as a feature with each type defined by 1, 2, 3 and 4 respectively. The dataset also included force, displacement, strain energy release rate and crack condition as dataset features.

**DATASET DESCRIPTION**

The developed dataset had features related to the properties of the composite, unstable crack propagation or no crack propagation. This is the crack-developed from the tip of the initial pre-crack that was designed in the ODE COMSOL Multiphysics model. The displacement noted in the dataset is the crack opening when the lower edge of the composite specimen was loaded. This displacement was divided by the force to calculate the compliance. The cube root of the compliance plot against the change in crack length was then used to evaluate the correction factor used in strain energy release calculations, which was another feature. The strain energy release rate represents the energy required for the crack to propagate through the specimen. The specimen type was included as another feature in the dataset that was labeled from 1 to 4 as described above. The condition of the specimen in the dataset represented the specimen has cracked propagation and no crack propagation noted by 1 or 0 respectively. Therefore, the input variables in this work were the displacement, force, strain energy release rate, specimen type and output was the specimen condition. This dataset was used for all models developed in this work.

**CRACK PREDICTION MODELS**

Four different types of machine learning algorithms were used in this work. First was the artificial neural network model, then decision tree algorithm for classification, random forest algorithm was also used, support vector machine and extreme gradient boost (XGBoost) model.
Prediction accuracy for all models was compared. All models in this work were developed using python language [12] [13].

**Artificial Neural Network**

The first step was to import the dataset into Google Colab environment and visualize the contribution of each feature to the composite crack initiation. Back Propagation Artificial Neural Network (BPANN) was following in this work. The algorithm training followed error backpropagation (BP) and comprised of multiple hidden layers, connection weights, one input layer and an output layer as shown in Figure 4 BPANN structure.

![BPANN Structure](image)

Each layer consists of its individual nodes which are represented by l, m and n for input, hidden and output layers respectively in this work. In this model, the input propagates through the hidden nodes to the output passing through a sigmoid activation function given by [12],

\[
f(z) = \frac{1}{(1 + e^{-z})}
\]

(1)

Where z is related to the previous layer input \(Z_1, Z_2, \ldots, Z_l\) for each input node. The results from each hidden layer using the data inputs and estimated weights \((W_{ij})\) was calculated using,

\[
X_j = f \left( \sum_{l=1}^{i} W_{ij}Z_l - T_j \right)
\]

(2)

Where T is the threshold for the hidden layer and \(j = 1, 2, \ldots, p\). This function is therefore used to calculated the output of the output layer as,

\[
Y_k = \sum_{j=1}^{l} X_jW_{jk} - S_k
\]

(3)
Where $S$ represents the threshold of the output layer. The prediction error was then calculated by deduction $Y_k$ from the actual output $y_k$ for each output node. If the output is not the desired one, the error was propagated back and the algorithm weights were modified to allow for correction of the error.

**Random Forest Algorithm**

Another algorithm applied for the developed dataset was random forest (RF) to predict crack generation and required strain energy release for the composites in this work. This algorithm initial development was by L. Breiman [13] as a way to implement stochastic discrimination to classification. It was further developed by different researchers [14] [15] to the algorithm applied in this work. This is a supervised learning algorithm for classification and regression. The data set was divided into $N$ sub-datasets in creating decision trees using a bootstrap resampling method. Each training sub-dataset was of the same size as the original training dataset. Column subsampling was followed in this work for more effect on the model. The optimal number of samples to incorporate to grow each tree was confirmed after achieving the best number of tree and variables. For each bootstrap training sub-dataset, a classification regression tree that’s not pruned was created to develop a forest of $N$ decision trees. A random selection of $i$ less or equal to $j$ of all features determined the branching process for each tree. Training of each independent decision tree occurred in parallel for better efficiency. The results are based on a voting process of the output for each decision tree. In this work, goodness of fit and predictive performance was characterized by the mean squared error (MSE) and $R^2$ given by [13],

\[
MSE = \frac{1}{n} \sum_{i}^{n} (y_i - \hat{y}_i)^2 
\]

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

Where the predicted output and response are given as $\hat{y}_i$ and $y_i$ respectively and the mean of the response variable is $\bar{y}_i$. In this algorithm of model ensemble test accuracy is improved while reducing the cost associated.

**Support Vector Machine**

In this section, support vector machine (SVM) was applied to find an optimal line that represents the decision boundary defining the hyperplane. In this algorithm, an optimal solution that maximizes the distance between the hyperplane and the difficult points close to the decision boundary was determined. This decision function was fully specified by a subset of training
samples, the support vectors. The hyperplane was parameterized by a vector w and a constant b as [16],

$$w \cdot x + b = 0$$ \hspace{1cm} (6)

where the hyperplane that can correctly classify the data was,

$$f(x) = \text{sign}(w \cdot x - b)$$ \hspace{1cm} (7)

The distance from each point to the hyperplane is computed following the total margin formulations that are dependent on the weights (w). Minimizing the weight vector is a nonlinear optimization task and that follows,

$$\vec{\omega} = \sum_{i=0}^{N} \lambda_i y_i \tilde{x}_i$$ \hspace{1cm} (8)

And this therefore maximizes the margin between the data points and the hyperplane. Therefore, this model drew a decision boundary by looking at the extreme cases of the dataset. Figure 5 shows how a hyperplane divides data classes and how margins are related to the support vectors.

**Figure 5. Illustrates classification with support vectors.**

**XG Boost Algorithm**

In extreme gradient Boost (XGBoost) algorithm [17], each tree boosts attributes that led to misclassifications of a previous tree. This method was selected in this work because of its ability to fix drawbacks of individual tree models as compared to other methods and also its capability of accurate predictions with imbalanced data. It entails regularized boosting, which prevents over fitting and can handle missing values automatically. An ensemble method of independent trees combined together to form a random forest was developed for the dataset in this work. Boosting therefore combined weak learners sequentially so that each new decision tree corrected the errors.
of the previous one. A single decision tree was first fit into the data and its performance evaluated using a cross entropy loss function given by,

$$\text{loss}(p, q) = -\sum_x p(x) \log q(x)$$  \hspace{1cm} (9)$$

where the label and the prediction are represented by p and q respectively. The loss is high when the label and prediction agree and the loss is zero when they are in perfect agreement. Addition of another tree lowers the loss and this is also dependent on the learning rate. During training in this model, cross validation was conducted to analyze the accuracy. In XGBoost, weights were adjusted at each step of learning.

**RESULTS**

This section details the prediction accuracies of each model and unpacks which model proved to have the best predictions. Four models were used in this work, artificial neural network, random forest algorithm, support vector machine and extreme gradient boosting model. Data split for each model was maintained as 60% training, 20% validation and 20% testing. The statistical relation of each feature in the dataset is shown in Figure 6 of python seaborn plot of each attribute.

![Python seaborn plot for each feature within the dataset.](image)

Figure 6. Python seaborn plot for each feature within the dataset.
The correlation of each feature shown in seaborn plot reflects a variational relation between displacement force and strain. These three features contribute to the generation of fracture in the composite specimen. It was observed how the features from the dataset contributes to the crack in ANN, random forest, SVM and XGBoost. First the ANN model training loss and validation loss of number of epochs is shown in Figure 7 a) and the training accuracy and validation accuracy over the number of epochs is also shown on Figure 7 b).

![Figure 7. Artificial Neural network model a) training loss vs validation loss and b) training accuracy vs prediction accuracy.](image)

This results reflect the loss of the model using the training data set as shown in Figure 7 a) and the separated validation loss is also shown. During training and validation, the results showed a similar trend therefore proving that the data was not over fitted. Training accuracy data also showed similar trend with validation data and proving a high accuracy of 98%. Random forest algorithm was also applied to the same dataset as detailed in the methodology section. Evaluation of the model accuracy was done following the classification metrics, precision recall which is a measure of relevant instances retrieved and the f1 score which also measures the precision of the model with highest precision being 1.0 and lowest being 0. An accuracy close to that of the random forest algorithm was achieved in the SVM model with an overall F1-score of 70 % and accuracy of 84%. Comparison of all the models in this work, XGBoost algorithm showed to have the highest accuracy and F1-score. Data for all algorithm’s evaluation metrics is therefore shown in Table II.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Accuracy</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>0.96</td>
<td>0.98</td>
<td>0.97</td>
<td>0.94</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.92</td>
<td>0.98</td>
<td>0.96</td>
<td>0.90</td>
</tr>
<tr>
<td>SVM</td>
<td>0.98</td>
<td>0.98</td>
<td>0.97</td>
<td>0.96</td>
</tr>
<tr>
<td>XGBoost</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
<td>0.99</td>
</tr>
</tbody>
</table>
The data showed that the gradient boosting method had a better performance than all other models in prediction the fracture of the composite specimens in this work. This model showed to have a relatively low root mean square error as shown in Figure 8 for both testing and training.

![Figure 8. Extreme gradient boosting algorithm root mean square error.](image)

For all models in this work, the conducted comparison study proved the capability of XGBoost model to accurately predict the given imbalanced data and yield the closest predictions than all the other models in this work. Determination of features of importance in the given data set was also done as shown in Figure 9.

![Figure 9. Ranking of all features for features of importance.](image)
Features of importance identification based on the gradient boosting model illustrated the importance of displacement, force and strain energy on the initiation of crack in the composite specimens. This is due to the dependance of crack propagation on displacement and strain energy required. Further studies on fracture properties of the smart composite in this work will be conducted following the extreme gradient boosting algorithm.

CONCLUSION

An ordinary differential equation (ODE) model coupled with statistical methods for the prediction of fracture toughness of a magnetostrictive, piezoelectric smart self-sensing fiber reinforced polymer (FRP) composite was developed in this work. The smart composite with sensing properties was embedded with Terfenol-D alloy nanoparticles and single walled carbon nanotubes (SWCNT). To explore various configurations the of nanoparticle constituents’ effect on fracture toughness within the FRP composite, the ODE model developed within a finite element analysis (FEA) environment is considered to attain fracture observations across the solution space. The acquired FEA data was then used to feed the machine-learning (ML) algorithms to obtain composite fracture toughness predictions. A comparison and development of artificial neural networks (ANN), random forest algorithm, support vector machines (SVM) and XGBoost algorithm ML models for FRP smart self-sensing composite fracture toughness prediction is done. Qualitative results stating if the sample has fractured or not and quantitative data giving the fracture toughness and strain energy release rate for the smart self-sensing FRP composites is attained. The data showed that the gradient boosting method had a better performance than all other models in prediction the fracture of the composite specimens in this work.

REFERENCES