Received: 16 March 2022 | Revised: 13 September 2022 | Accepted: 26 October 2022

DOI: 10.1002/pc.27144

RESEARCH ARTICLE

Polymer COMPOSITES WILEY

Determining damage initiation of carbon fiber reinforced polymer composites using machine learning

Alex Post[1,2] | **Shiyao Lin**[1] | **Anthony M. Waas**[1] | **Ilyas Ustun**[2]

[1]Department of Aerospace Engineering, University of Michigan, Ann Arbor, Michigan, USA

[2]School of Computing, College of Computing and Digital Media, DePaul University, Chicago, Illinois, USA

Correspondence
Anthony M. Waas, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI, USA.
Email: awaas@umich.edu

Abstract

Computational progressive failure analysis (PFA) is vital for the analysis of carbon fiber reinforced polymer (CFRP) composites. The damage initiation criterion is one of the essential components of a PFA code to determine the transition of a material's state from pristine or microscopically damaged to macroscopically damaged. In this paper, data-driven models are developed to determine the matrix damage initiation with the objective to save computation time. For 2D plane stress states, the computational cost for determining damage initiation can be dramatically reduced by implementing a binary search (BS) algorithm and predictive machine learning models. Machine learning models are evaluated against a regression problem of predicting damage crack angle as well as against a classification problem for predicting damage initiation outright. It is found that regression models perform much better for plane stress and 3D stress states when generating failure envelopes. In both cases, a neural network is able to produce a failure envelope that is over 99% accurate while reducing computational cost by over 90%.

KEYWORDS

composites, degradation, failure

1 | INTRODUCTION

Carbon fiber reinforced polymer composite (CFRP) materials have been widely used in the Aerospace industry. CFRP materials are superior in terms of strength/stiffness-to-weight ratios, fatigue resistance, and corrosion resistance. CFRP materials are composed of carbon fiber, and a polymer (resin or matrix). Due to their microscale architecture, the damage morphology of CFRP laminates is complex, usually consisting of fiber breaking, matrix cracking, and delamination. The complexity of the damage of CFRP materials makes experimental characterization and computational modeling challenging.

Numerous progressive failure analysis (PFA) codes for CFRP materials have been developed for decades to simulate and predict the damage of composites under external tensile/compressive/flexural/impact/fatigue loading. Typically, a PFA model is composed of three major parts to capture the three most important features of the damage of composites, including pre-peak nonlinearity, damage initiation, and post-peak degradation. The pre-peak nonlinearity is driven by microscale cracking in the matrix between fibers.[1] The nonlinearity has been modeled by semi-empirical expressions in.[2–4] A rigorous thermodynamically based theory was established to characterize the relationship between the nonlinearity and microscale

damage dissipation by Schapery.[5] This model has later been enhanced by the capability to model post-peak degradation as the enhanced Schapery theory (EST).[6] Damage initiation criteria determine the initiation of damage modes, including fiber tensile/compressive damage, matrix tensile/compressive/shear damage, and inter-laminar delamination. The most popular damage initiation criteria include the Hashin criteria,[7] Puck's criteria,[8] LaRC criteria,[4] and more recently the Christensen criteria.[9,10] The post-peak degradation laws control the softening of material properties due to cracking. Most of the degradation laws are established upon the crack band model[11] and the smeared crack model.[12] Mixed-mode post-peak degradation laws have been proposed to ensure simultaneous and progressive vanishing of stress components to improve numerical stability of PFA codes.[13–16]

Damage initiation criteria are of essential importance in determining the location, time, and types of composite damage initiation. The Hashin criteria are the most popular ones thanks to their simplicity and accuracy, especially in terms of fiber and matrix tensile damage. EST is established upon the Hashin criteria. However, the fracture planes in the Hashin criteria are prescribed, which might not be true for some stress states. For example, as demonstrated in Reference 4, a unidirectional composite laminate under transverse compression had a slanted matrix failure plane, which does not agree with the prescribed fracture planes of the Hashin criteria. A physically sound model was developed by Puck and Schürmann[8] by adapting the Mohr–Coulomb criteria to composite materials. In Puck's criteria, at a certain possible matrix crack plane, if matrix is under compressive normal stress, the matrix failure is essentially a shear fracture. Therefore, the matrix crack plane angle with the maximum damage potential needs to be searched for iteratively. Puck's criteria were later partially adopted by the LaRC criteria where the iterative matrix crack angle determination process was kept.[4] Cuntze developed damage initiation criteria based on stress invariants and the matrix crack angle searching was not required.[17] Several studies have compared the accuracy and efficiency of various damage initiation criteria. It was found that the criteria involving matrix crack angle determination are less efficient than that with prescribed fracture planes,[18,19] due to the complexity of the crack angle search process. The maturity of failure criteria were evaluated extensively and in detail in Reference 20. In the study, both the LaRC criteria (sometimes also referred to as the Pinho model) and Puck's criteria were ranked as the better theories. However, as described in Reference 18,19, these two models are less efficient due to the iterative search of the crack angle.

The use of machine learning to improve computational efficiency has been shown to be successful in the prediction of material properties. Hou et al. utilized machine learning techniques to predict the temperature difference and degree of cure difference during the curing process of composites.[21] Additionally, Özkan et al. used neural networks to accurately predict the mechanical properties of nanocomposite films.[22] The authors' previous work discusses the application of binary search (BS), Linear Regression, and Neural Networks to improve the computational efficiency of predicting damage initiation.[23] This previous work, however, considered only two possible models, analyzed only a numeric prediction, and used an inherently limited training data set. This paper will show that accuracy can be significantly increased by improving the training data set to better represent the physical reality, analyzing a combination of numeric and discrete predictions, and better formulating the problem statement to reflect the true objective of the damage initiation criteria.

The major objective of this paper is to improve the computational efficiency of PFA tools by accelerating the crack angle searching process of the damage initiation criteria based on the Mohr–Coulomb model. The crack angle searching and damage initiation determination process is referred to as the competing algorithm in this paper, explained in Section 2. The acceleration is achieved by using the BS algorithm and data-driven methods including linear regression (LR), light gradient boosting machine (LGBM), and deep neural network (DNN), which will be described in Section 2. Predicted results will be validated by experimental results from the "World-wide Failure Exercise" (WWFE)[24] in Section 3. Discussions, conclusions, and future directions will be provided in Section 4.

2 | METHODOLOGY

In this section, the methodology of the conventional matrix crack angle searching algorithm (the competing algorithm) will be outlined first. Then, the BS algorithm applicable to 2D plane stress states will be introduced. Data-driven approaches including the LR, DNN, and LGBM methods will be illustrated. The BS algorithm, LR model, DNN model, and LGBM model have been implemented to accelerate the determination of matrix crack angle.

2.1 | Matrix damage initiation criteria

This paper is mostly concerned with the damage initiation of matrix. Details about fiber damage initiation criteria can be found in 14. Matrix cracking of a $[0/90/0/90_9]_s$ laminate under center impact loading can be

visualized from a μCT slice cutting across the plate center as shown in Figure 1. The central bright lines indicate matrix cracking in the middle 18 plies of 90° laminae.[25] Figure 1 demonstrates the slanted matrix cracking. In order to capture these slanted cracks, physically sound matrix damage initiation criteria need to be applied.

Physically sound matrix damage initiation criteria, as that implemented in References 4,8, are mostly based on the modified Mohr–Coulomb model. Consider an unidirectional composite laminae with a 3D stress state, as illustrated in Figure 2A. Under this stress state, both tensile and compressive matrix damage may occur on a crack surface with an angle of θ. In order to determine the matrix damage initiation status, there are two key factors to be considered, which are the crack angle θ and normal stress components σ_{NN}. As shown in Figure 2B, σ_{NN}, τ_{NL}, and τ_{NT} are the stress components on the local crack surface coordinate system $N-L-T$. Therefore, following coordinate transformation, the stresses on the local coordinate system $N-L-T$ are computed following Equations 1–8.

$$\sigma_{LL} = \sigma_{11} \quad (1)$$

$$\tau_{NL} = \tau_{13}\sin(\theta) + \tau_{12}\cos(\theta) \quad (2)$$

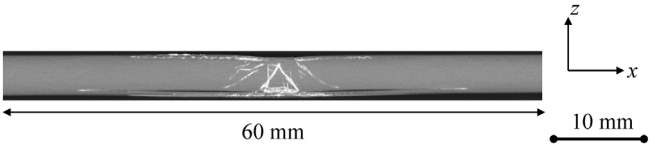

FIGURE 1 Edge-on μCT slice of an impacted $[0/90/0/90_9]_s$ laminate

The Competing Algorithm for Matrix Tensile/Compressive Crack Initiation

At inc i, $\boldsymbol{\sigma}^{i-1}$ is known. $\theta = 0{:}1{:}180$
do j=1:length(θ)
  calculate coordinate transformation matrix $\boldsymbol{T}(\boldsymbol{\theta}(j))$
  calculate transformed stress $\boldsymbol{\sigma}_{cr}^{i-1} = \boldsymbol{T}\boldsymbol{\sigma}^{i-1}\boldsymbol{T}^{T}$
  calculate $f_{mt} = \left(\frac{\sigma_{NN}}{S_{22}}\right)^2 + \left(\frac{\tau_{NL}}{S_{12}}\right)^2 + \left(\frac{\tau_{NT}}{S_{23}}\right)^2$
  calculate $f_{mc} = \left(\frac{\tau_{NL}}{S_{12}-\mu_L\sigma_{NN}}\right)^2 + \left(\frac{\tau_{NT}}{S_{23}-\mu_T\sigma_{NN}}\right)^2$
  if $f_{mt} \geq 1$ then
    tensile initiation
    $\theta_{cr} = \theta(j)$
    break
  end if
  if $f_{mc} \geq 1$ then
    compressive initiation
    $\theta_{cr} = \theta(j)$
    break
  endif
  if tensile/compressive initiated then
    break
  endif
end do

FIGURE 3 Pseudo code of the competing algorithm

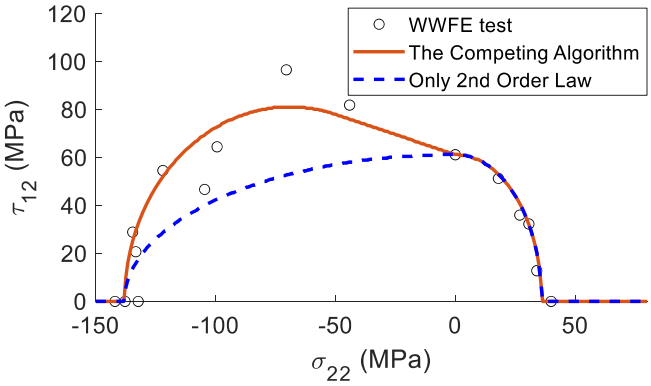

FIGURE 4 Failure envelopes for E-Glass/LY556 using solely second-order power law versus competing algorithm

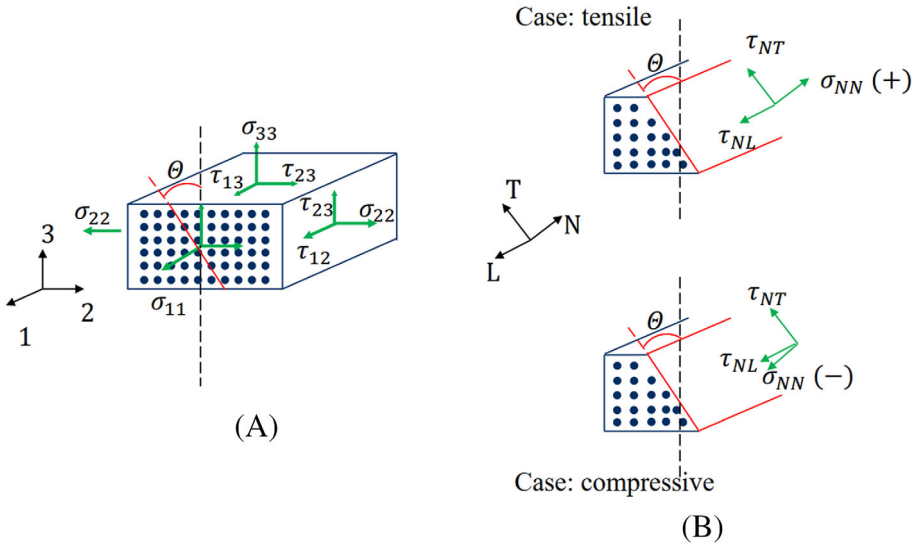

FIGURE 2 Illustration of the matrix cracking under a 3D stress state: (A) the stress state, and (B) matrix tensile and compressive crack surfaces

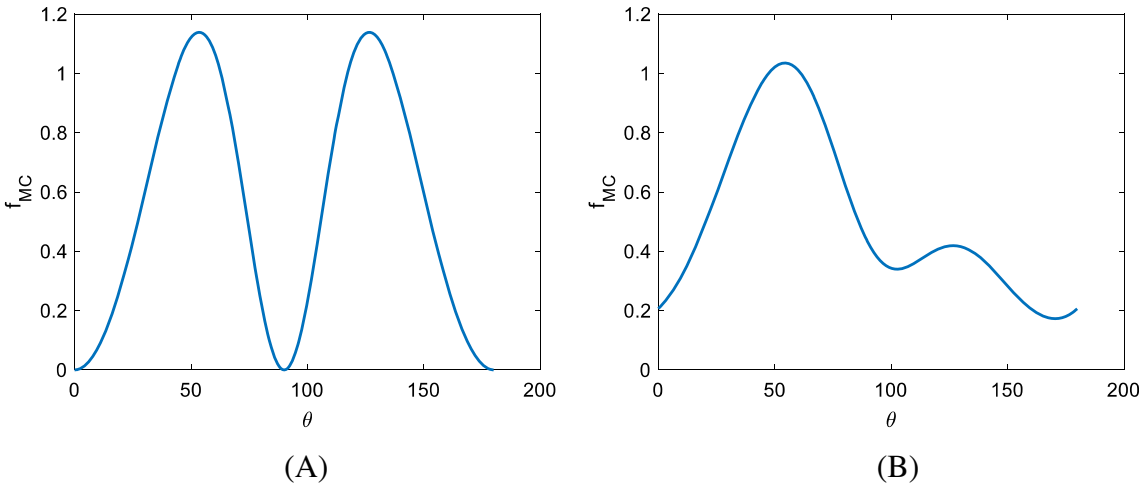

FIGURE 5 Examples of MC criterion distribution over θ in degrees for (A) plane stress and (B) 3D stress

TABLE 1 Descriptions of data-driven models

Model type	Models	Input	Target
Classification	1. LR 2. NN 3. LGBM	Stress components and material strengths	Damage initiation status
Regression	1. LR 2. NN 3. LGBM	Stress components and material strengths	Crack angle with the maximum damage potential

$$\tau_{LT} = \tau_{13}\cos(\theta) - \tau_{12}\sin(\theta) \tag{3}$$

$$\sigma_{NN} = \sigma_{22}\cos^2(\theta) + \sigma_{33}\sin^2(\theta) + 2\tau_{23}\sin(\theta)\cos(\theta) \tag{4}$$

$$\tau_{NT} = (\sigma_{33} - \sigma_{22})\sin(\theta)\cos(\theta) + \tau_{23}\left(\cos^2(\theta) - \sin^2(\theta)\right) \tag{5}$$

$$\sigma_{TT} = \sigma_{33}\cos^2(\theta) + \sigma_{22}\sin^2(\theta) - 2\tau_{23}\sin(\theta)\cos(\theta) \tag{6}$$

As seen in Figure 2B, if the normal stress σ_{NN} is tensile (positive), the matrix damage is tensile damage; otherwise, the matrix damage is compressive. Second-order damage initiation criterion in Equation (7) is applied for the tensile case.

$$\left(\frac{\sigma_{NN}}{S_{22}}\right)^2 + \left(\frac{\tau_{NL}}{S_{12}}\right)^2 + \left(\frac{\tau_{NT}}{S_{23}}\right)^2 \geq 1 \quad \sigma_{NN} \geq 0 \tag{7}$$

where S_{22}, S_{12}, S_{23} are the matrix transverse tensile, in-plane shear, and out-of-plane shear strengths. The second-order criterion in Equation (7) is usually referred to as the Hashin criteria.[15,26]

When the normal stress component is compressive (negative), the Mohr–Coulomb criterion is used to determine the matrix damage initiation, as in Equation (8).

$$\left(\frac{\tau_{NL}}{S_{12} - \mu_L\sigma_{NN}}\right)^2 + \left(\frac{\tau_{NT}}{S_{23} - \mu_T\sigma_{NN}}\right)^2 \geq 1 \quad \sigma_{NN} < 0 \tag{8}$$

where μ_L and μ_T are the internal coefficients of friction along the local L and T directions. Measurement and calculation of μ_L and μ_T can be found in.[4] The Mohr–Coulomb criteria in Equation (8) was first adopted by Puck and Schürmann in Reference 8 and frequently referred to as the Puck's criterion. However, in this paper, it is still referred to as the Mohr–Coulomb damage initiation criterion.

Stress components in Equations (7) and (8) are dependent on the crack angle θ, which is not known before the damage initiation criteria is satisfied. Therefore, the crack angle needs to be searched for to achieve the maximum value of Equation (7) or (8) until matrix damage initiates. The conventional crack angle searching process will be illustrated in Section 2.2.

ALGORITHM 1 **Training Data Generation**

```
s = n                                                  ▷ Desired size of data set
i = 0                                                  ▷ Number of rows generated
while i < n do
    S22^tensile = random(20, 150)
    S22^compressive = random(70, 450)
    S12 = random(30, 180)
    σ22 = random(-600, 200)
    σ33 = random(-600, 200)
    τ12 = random(-250, 250)
    τ13 = random(-250, 250)
    τ23 = random(-250, 250)
    θ, MC = computeFailureCriterionAndAngle()
    if MC < 1.3 then                                   ▷ 1.3 Threshold on MC Criterion values
        Append [yT, yC, sL, σ22, σ33, τ12, τ13, τ23, MC, θ]
        i += 1
    end if
end while
```

2.2 | The competing algorithm

The matrix crack angle searching process is implemented within the competing algorithm. Similar algorithms were adopted in References 2,4. The word "competing" means that the algorithm keeps evaluating the potentials of tensile and compressive matrix cracking, namely the values of Equations (7) and (8). Once Equation (7) or (8) gets satisfied, tensile or compressive matrix cracking initiates. The competing algorithm is illustrated in Figure 3. As shown in Figure 3, the matrix crack angle is scanned from 0 ° to 180 ° with an interval of 1 °. At each iteration, with the tentative crack angle θ, Equations (7) and (8) are evaluated. The looping process in Figure 3 is stopped immediately and the damage initiation status as well as the crack angle are returned after the satisfaction of Equation (7) or (8).

From the description, the competing algorithm is a brute-force algorithm whose complexity is $O(n)$ with respect to candidate failure angles, which means that if there are n iterations to be performed, it would take at most kn computations to find a crack angle where k is some constant.

The advantage of applying the competing algorithm to the damage initiation prediction is validated by Figure 4. Figure 4 presents experimental data from

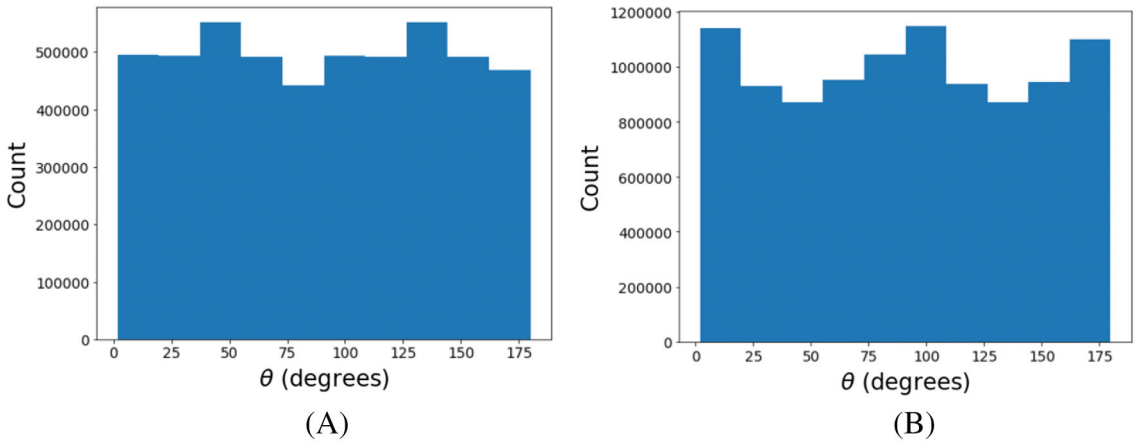

FIGURE 6 Examples of MC criterion distribution over θ (A) without 1.3 MC filter and (B) after applying the 1.3 filter

TABLE 2 Evaluation metrics for each trained model for plane stress crack angle prediction with respect to the competing algorithm baseline

Model	CAR^2	CA MAE	CA/FE R^2	CA/FE MAE	FP Acc	FP LL	FP/FE R^2	FP/FE MAE
BS	0.9878	0.0188	1.0000	2.5 e − 15	N/A	N/A	N/A	N/A
LR	0.7504	0.1330	0.9802	2.7486	0.8345	5.7151	−1.6956	37.8283
NN	0.9890	0.01187	0.9999	0.0062	0.9765	0.0554	0.9442	2.8855
LGBM	−0.7268	0.4550	−0.5631	16.9618	0.9127	0.2696	−4.0093	55.3705

Abbreviations: Acc, accuracy; CA, crack angle; FE, failure envelope; FE/CA, failure envelope using crack angle predictions; FP, failure predictions; FP/FE failure envelope using failure predictions; LL, Log Loss; MAE, mean absolute error; R^2, coefficient of determination.

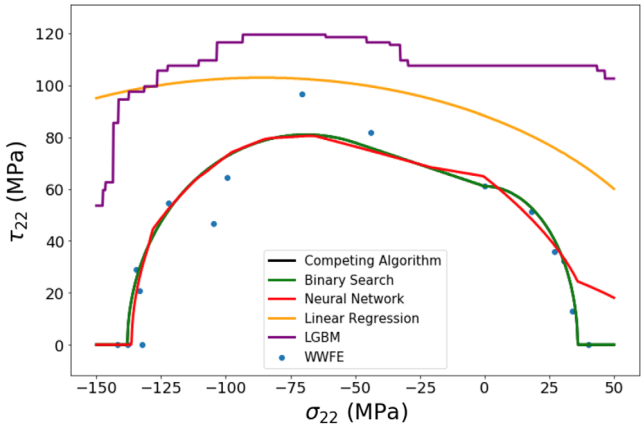

FIGURE 7 Failure envelope for E-Glass/LY556 produced from plane stress classification models

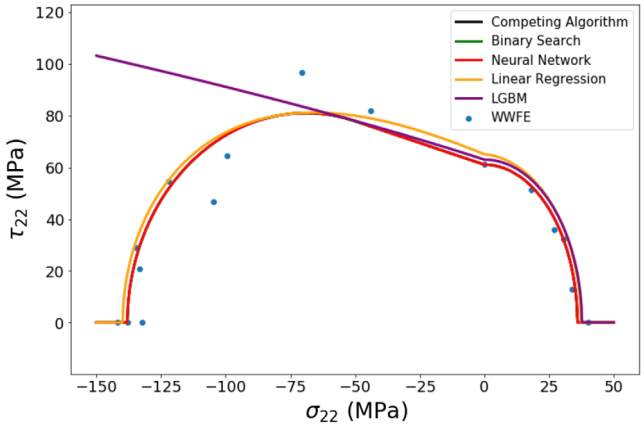

FIGURE 8 Failure envelope for E-Glass/LY556 produced from plane stress crack angle regression models

Reference 24 and failure envelopes predicted by using the competing algorithm and that using only second-order power law. It is clearly observed that when σ_{22} is compressive, the failure envelope obtained by the competing algorithm agrees much better with the test results.

2.3 | The BS algorithm

The BS algorithm is a common algorithm designed to search for objects within a sorted array of values. The algorithm works by iteratively splitting a given array in half until an objective value is found. The BS algorithm runs in $O(log_2(n))$ time, which is substantially faster than the Competing Algorithm. Considering the use case of finding the angle with the maximum damage potential, the problem space must be transformed into a sorted array to be able to take advantage of the BS algorithm.

Figure 5 shows the distribution of the MC criterion over possible θ vales. The x-axis shows the range of possible failure angles and the y-axis shows the computed MC criterion value (f_{MC}) for that angle. The maximum f_{MC} in this distribution represents the angle at which the damage potential is the greatest and, subsequently, if that value is greater than one the material will fail at that angle. For all two dimensional plane stress states, it can be assumed that this distribution will be bi-modal and symmetric—more importantly, every local maximum

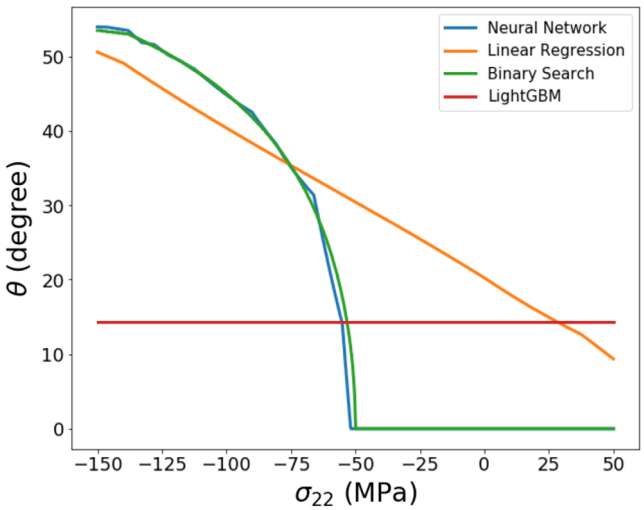

FIGURE 9 Crack angle predictions compared to the competing algorithm baseline

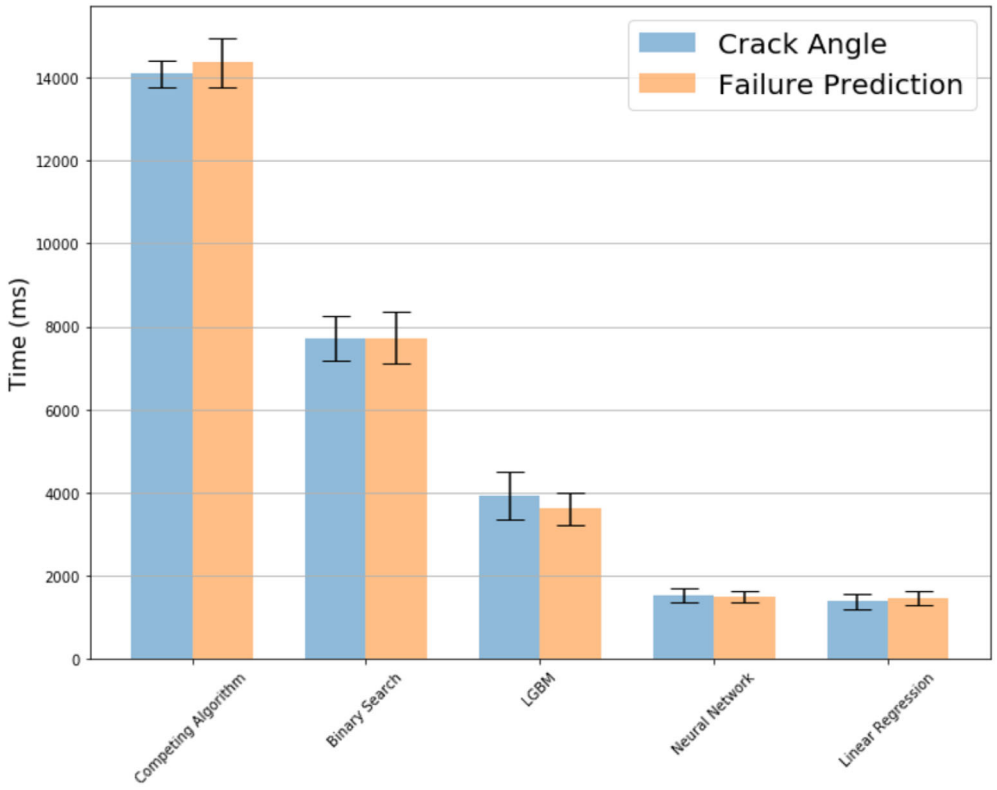

FIGURE 10 Performance comparison between different model types for plane stress state

value is equal to the global maximum value. This distribution is shown in Figure 5A. This property allows the BS algorithm to search for an element in the distribution for which the values on either side are smaller. The elements can also be lazily evaluated to save compute time, at each iteration only the angle in question and angle on each side need to be computed. The distribution of damage potential over θ for a 3D stress state is shown in Figure 5B. It is seen that the bi-modal and symmetric assumptions are no longer applicable. Therefore, the BS algorithm is not suitable for 3D stress states.

2.4 | Data-driven models

Data-driven models provide an opportunity to utilize machine learning techniques to reduce the computational complexity of the MC criterion computation. These machine learning techniques are particularly attractive because, while in general they vary among themselves in terms of runtime complexity, they will all run in $O(1)$ time with respect to the granularity of candidate failure angles.

Data-driven models can be broadly categorized into classification and regression models. In this paper, for each plane stress and 3D stress classification models are considered for predicting damage initiation outright, while regression models are considered for predicting the crack angle. The description of models can be found in Table 1.

For the regression models, the predicted crack angle is input back into Equations (1) to (8) to evaluate if the damage initiation criteria are satisfied, and the resulting failure envelopes. Each predictive model discussed in this paper was trained using a 40/30/30 split of the training data meaning that 40% of the available data was used for training, 30% for validation, and another 30% for testing. All metrics reported are with respect to the hold out test set.

2.4.1 | Metrics

A few different metrics are used to evaluate the performance of predictive models. For classification models, Accuracy and binary cross-entropy/log loss (LL) are used. Accuracy is simply the percentage of predictions, which were correct on the hold out test set. The equation for LL is shown in Equation (9). Accuracy is used because of its straightforwardness. LL provides some benefits over accuracy in that it provides a more continuous metric that penalizes prediction probabilities that are less certain of outcome.

$$\text{Loss} = -\frac{1}{n}\sum_{i=1}^{n} y_i \cdot \log \widehat{y}_i + (1-y_i) \cdot \log\,(1-\widehat{y}_i) \qquad (9)$$

where y_i is the observed value; $\bar{y}$ is the mean of the observed data; $\widehat{y}$ is the predicted value.

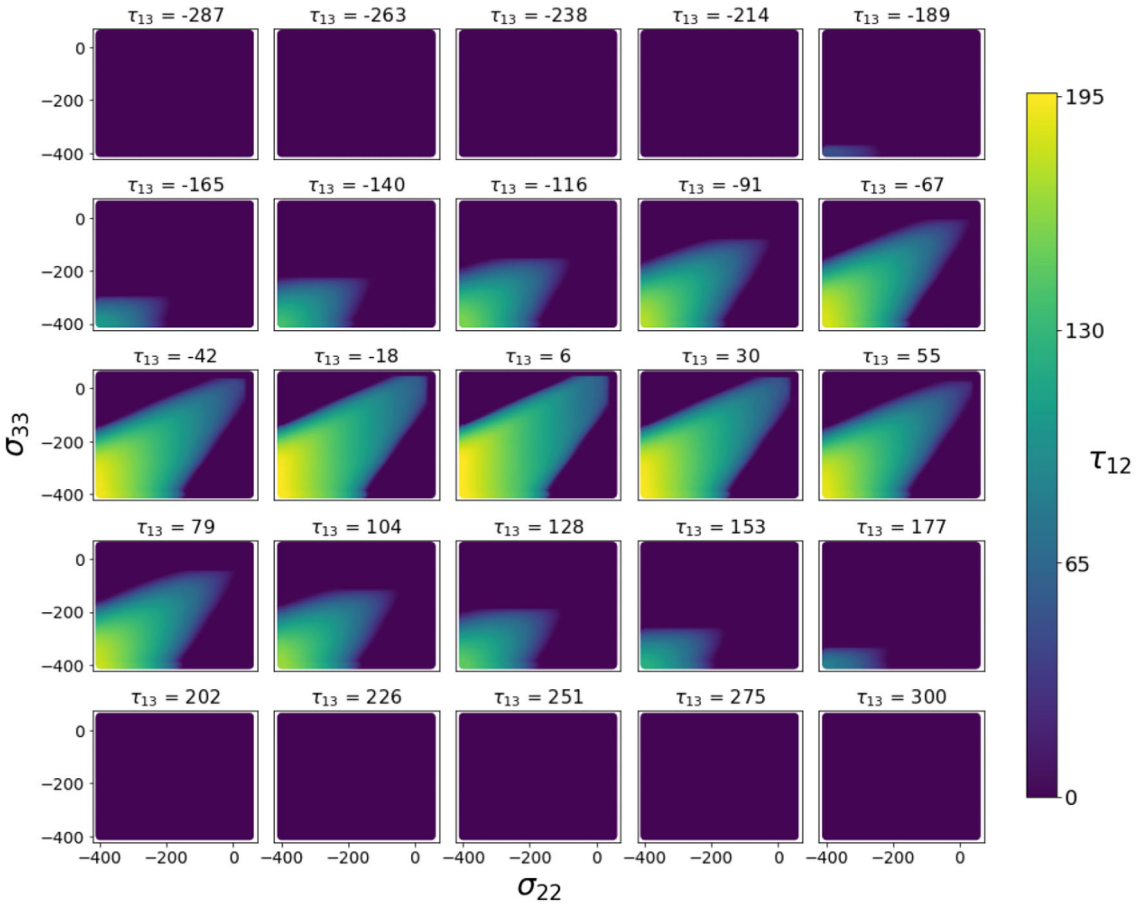

FIGURE 11 Failure envelope generated by the competing algorithm for E-Glass/LY556 across various τ_{13} values

For regression models, the metrics reported are R^2 and mean absolute error (MAE). R^2, also known as the coefficient of determination, can be described as the amount of variance in the test data captured by the model and is expressed as a percentage. The formula for R^2 is shown in Equation (10).

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{10}$$

MAE is defined by Equation (11).

$$\text{MAE} = \frac{\sum_{i=1}^{n} |y_i - \hat{y}_i|}{n} \tag{11}$$

2.4.2 | Training data generation

Training data generation is the most important part of the data driven process. The data must be accurate and large enough to provide sufficient data for the models to find patterns. Multiple iterations were performed to find the best set of training data. The available variables in training data generation are the raw stress values and material strength values used within the algorithm defined in Algorithm 1. Randomized data was used for each stress variable and material parameter to ensure sufficient distribution for each variable across the sample space. The damage initiation potential value was computed using the methodology described in Section 2.2.

During training data generation, it was found that the majority of randomly generated stress states resulted in high damage initiation potential values that would be impossible to achieve in practice. As a stress is applied to a material, the damage initiation potential value should increase progressively until it arrives at 1, at which point the material will have damage initiation. Strictly speaking, all stress states that result in a damage initiation potential value of greater than 1 could be discarded because stress cannot be applied instantaneously and a material would fail prior to reaching such a stress state. However, a threshold of 1.3 was applied to filter out unrealistic stress states to account for a margin of error. Applying this filter resulted in the discarding of approximately 86% of

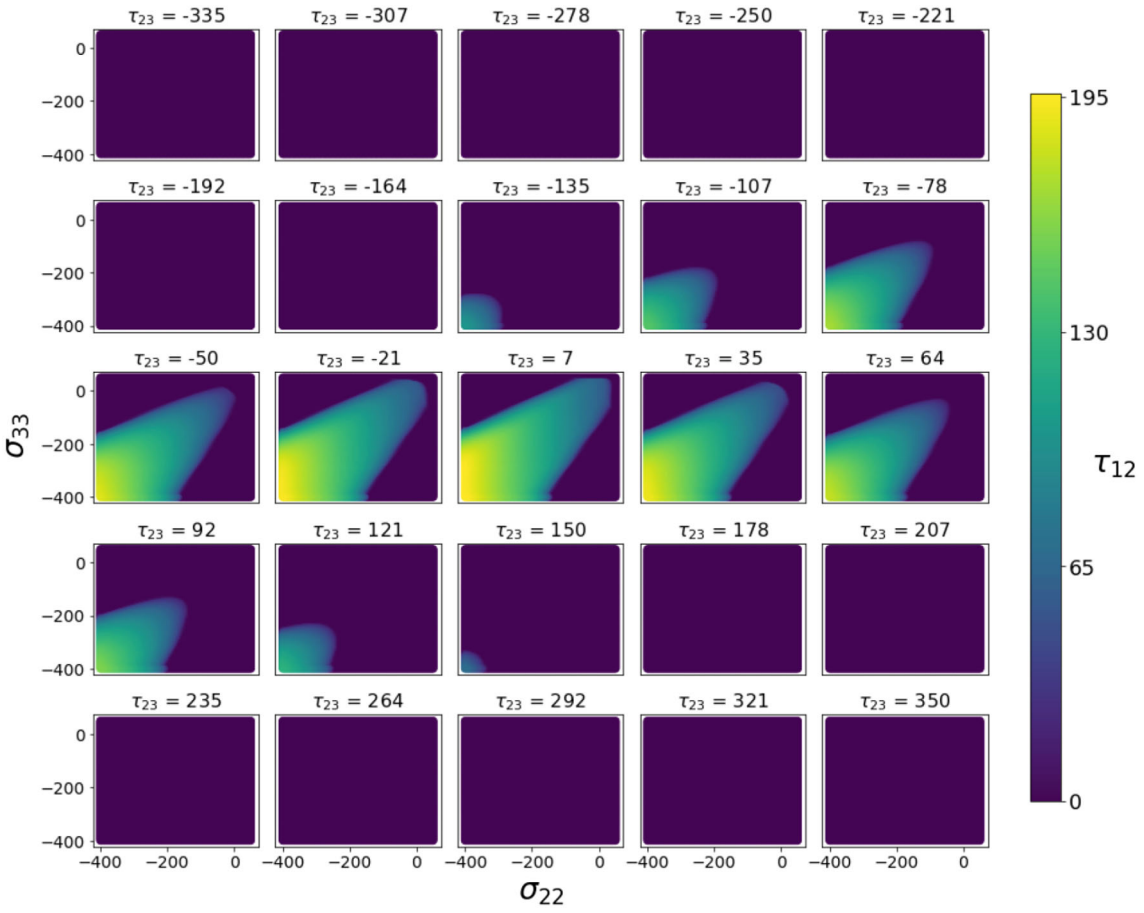

FIGURE 12 Failure envelope generated by the competing algorithm for E-Glass/LY556 across various τ_{23} values

TABLE 3 Evaluation metrics for each trained model for 3D stress crack angle prediction with respect to the competing algorithm baseline

Model	CA R^2	CA MAE	CA/FE R^2	CA/FE MAE	FP Acc	FP LL	FP/FE R^2	FP/FE MAE
LR	0.3034	0.6319	0.7482	7.0876	0.8009	6.8768	−0.1893	18.6581
NN	0.9271	0.0502	0.9922	0.7924	0.2936	0.0148	0.1421	15.2047
LGBM	0.7369	0.2996	0.7482	7.0876	0.2415	0.2606	0.7923	8.5473

Abbreviations: Acc, accuracy; CA, crack angle; FE, failure envelope; FE/CA, failure envelope using crack angle predictions; FP, failure predictions; FP/FE failure envelope using failure predictions; LL, Log Loss; MAE, mean absolute error; R^2, coefficient of determination.

generated stress states and produced a noticeable change in the distribution of damage initiation potential values, illustrated in Figure 6. The reason for such a high percentage of discarded stress states is that the ranges of stresses used to generate training data was intentionally set sufficiently wide, as shown in Algorithm 1.

For 2D plane stress states, the BS algorithm can be used to produce the required data for training data driven models. For 3D stress states, however, the assumption that the local maximum always equals the global maximum no longer holds, as illustrated by the counter example in Figure 5B. Since the BS algorithm can no longer be used, the competing algorithm was used to generate the training data for the data-driven models of the 3D stress states.

2.4.3 | The LR model

LR is a common model used for regression analyses. The model assumes a linear relationship between the dependent variable, y, and the set of regressors, X. The model is shown in Equation (12). There are many ways to determine the value of the coefficients but in this paper the

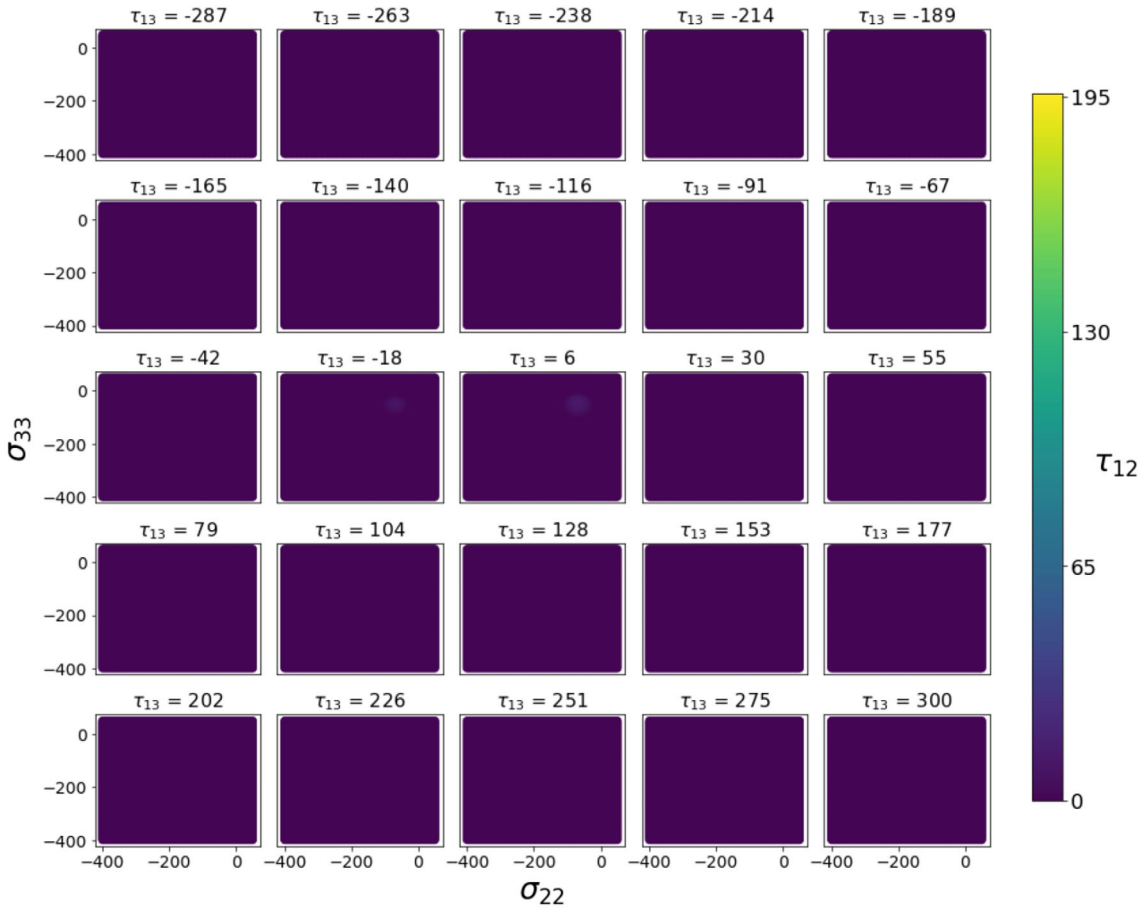

FIGURE 13 Failure envelope generated by the LR classification model for E-Glass/LY556 over τ_{13} values

coefficients β_i are fitted using Stochastic Gradient Descent.[27]

$$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_{n-1} X_{n-1} + \beta_n X_n \tag{12}$$

Logistic regression is another form of regression that can be used for classification. While linear regression attempts to find the best fit line, logistic regression attempts to find the best separating boundary that classifies each class correctly. The Logistic Regression equation is shown in equation 13.

$$f(x) = \frac{L}{1 + e^{-k(x - x_0)}} \tag{13}$$

where x_0, the x value of the Sigmoid's midpoint. L, the curve's maximum value. k, the logistic growth rate or steepness of the curve.

LR can have difficulty finding non-linear relationships between variables. By adding quadratic and interaction terms to the training data this can be somewhat mitigated, however, there are still models which may be better suited to this problem. Regardless, LR is perhaps the simplest predictive model available and serves as an important baseline by which to compare more complex models.

2.4.4 | The LGBM model

LGBM stands for Light Gradient Boosting Machine and is one of the most well known and best performing tree-based models. The technical details of implementation are discussed in.[28] Gradient boosted trees work by training models sequentially with each subsequent tree learning from the errors in the previous trees. One of the key differences between LGBM and other tree based gradient boosters is that LGBM grows trees vertically as opposed to horizontally. Additionally, LGBM has been shown to provide more accurate results in shorter time than other gradient boosters. LGBM is a highly configurable model and in this paper the configurations used in training are *objective*, *boosting_type*, *max_bins*, *num_leaves*, *learning_rate*, and *num_boost_rounds*.

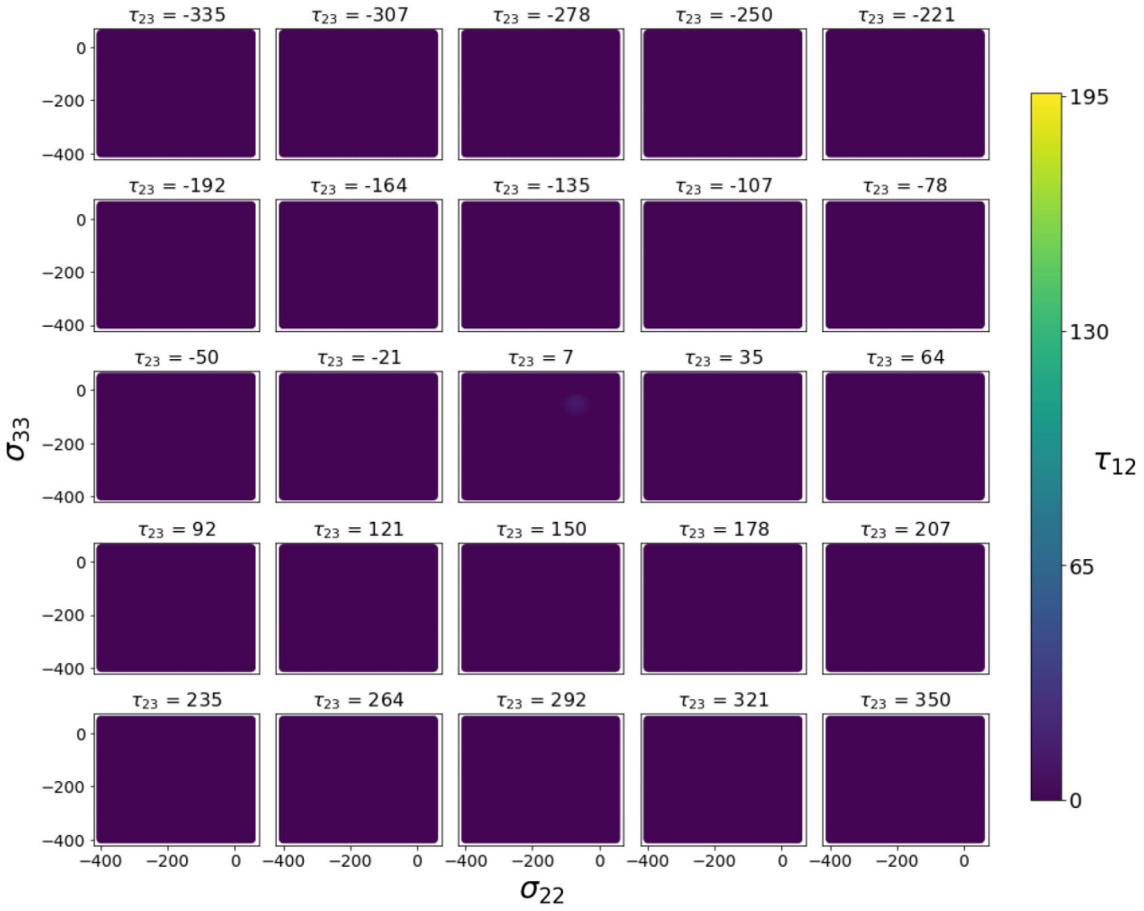

FIGURE 14 Failure envelope generated by the LR classification model for E-Glass/LY556 over τ_{23} values

objective denotes the type of prediction that should be produced. For regression problems, the objective is set to "regression," and for binary classification problems, the *objective* used is "binary." *boosting_type* indicates the type of boosting to use, for all models in this paper the *boosting_type* is Gradient Boosting Decision Tree, meaning traditional gradient boosting will be used.[28,29] *max_bins* indicates how many bins to use when discretizing continuous variables, this is an important configuration because it has a major impact on runtime. *num_leaves* configures the maximum number of leaves to allow in each iteration of tree building. *learning_rate* is the gradient descent learning rate.[29] Finally, *num_boost_rounds* is number of iterations to run boosting, this is comparable to epochs in neural networks. The loss metrics considered during training are consistent with reported loss metrics for all models in this paper, MAE is used for crack angle predictions and Binary Cross-Entropy is used for classification of failure.

The runtime complexity of LGBM models are O (#machine * #feature * #bin), meaning runtime is determined solely by the number of available machines, the number of features, and the number of bins.

2.4.5 | The DNN model

NN provides a superior method of capturing non-linear relationships compared to LR despite being conceptually similar. A NN is able to capture non-linear behavior because it typically consists of a multilayer perceptron with a non-linear activation function like Sigmoid (Equation 14) or rectified linear unit (ReLU; Equation 15).

$$S(x) = \frac{1}{1+e^{-x}} \tag{14}$$

$$R(x) = max(0, x) \tag{15}$$

Neural networks in this paper vary in architecture and activation functions but activation functions for all classification networks are Sigmoid and for the regression networks are ReLU. This is because the LL metric requires the Sigmoid function to work properly. NN has a prediction run time that scales with the number of nodes, which means that it does not depend on the granularity of candidate crack angles.

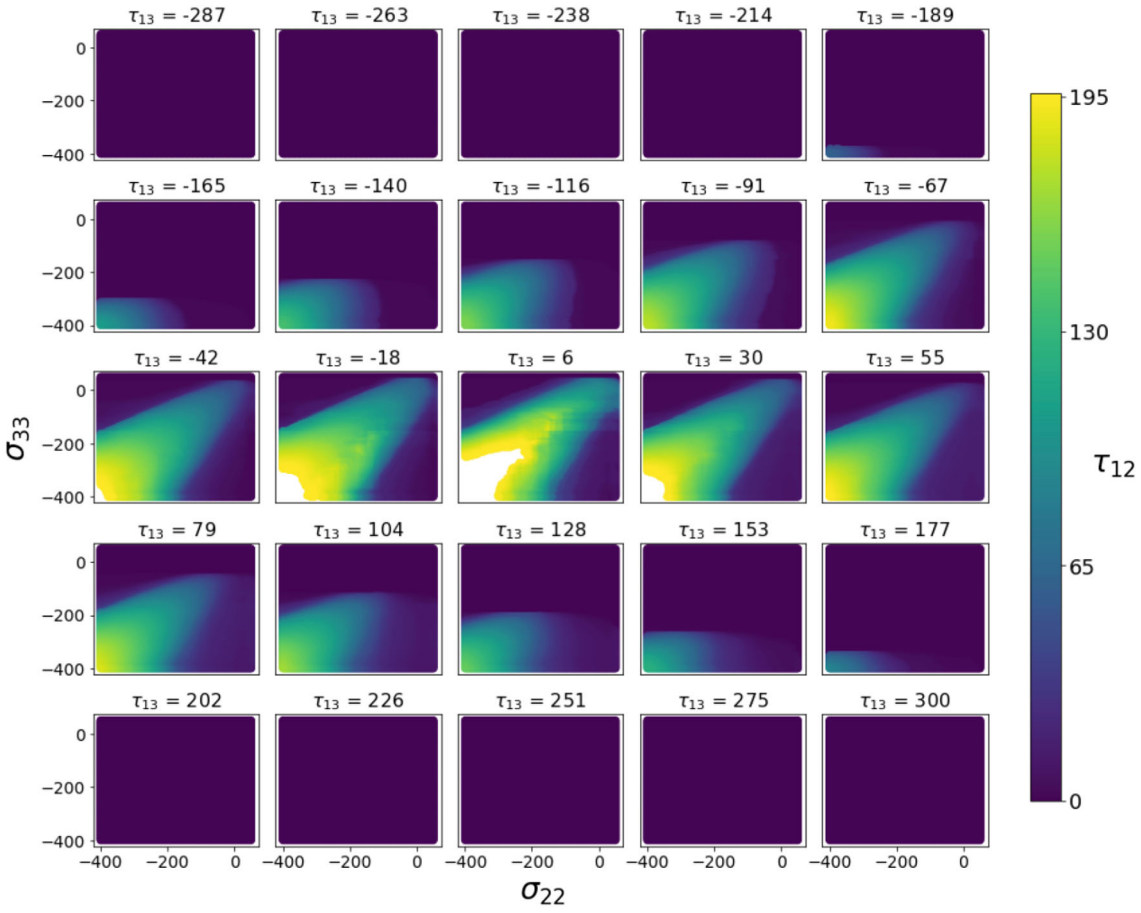

FIGURE 15 Failure envelope generated by the LGBM classification model for E-Glass/LY556 over τ_{13} values

3 | RESULTS

For both plane stress and 3D stress, three different predictive models were trained and evaluated: LR, LGBM, and Neural Networks. As stated above, each of these models were trained on both a classification problem to predict failure outright and on a regression problem to predict the crack angle. Additionally, for 2D stress, a BS algorithm is evaluated for the crack angle prediction.

3.1 | Matrix damage initiation with 2D plane stress states

3.1.1 | Prediction evaluation

The models evaluated for plane stress are the same models discussed in Section 2: BS (as a replacement for the competing algorithm), LR, LGBM, and NN. BS was able to produce exact results for both the crack angle and the failure envelope. The R^2 and MAE values for the BS algorithm are not 100% because in some cases the BS actually produces more accurate results than the competing algorithm. This is because the competing algorithm breaks out of its loop the first time it finds an angle that satisfies the failure criterion while BS is sure to find the angle with the maximum damage potential. In extreme compressive stress those two values are not necessarily the same, though those stress states are likely impossible to achieve without the material first failing at a lower stress state. The BS should be viewed as a benchmark by which machine learning methods should be evaluated. This is because the BS algorithm produces exact results—if machine learning methods sacrifice any accuracy, they should at least provide an increase in efficiency over the best exact method.

The LR model was built from the Scikit-Learn python library[30] using the SGDRegressor and SGDClassifier classes, which uses stochastic gradient descent[27] to optimize the regression coefficients. The LGBM model was built using the LGBM python package[28] with traditional gradient boosting decision trees. Additional parameters for LGBM were set for max bins to be 500, the learning rate was set to 0.001, and the number of boosting rounds was set to 2000. Finally the NN was trained with two hidden layers of

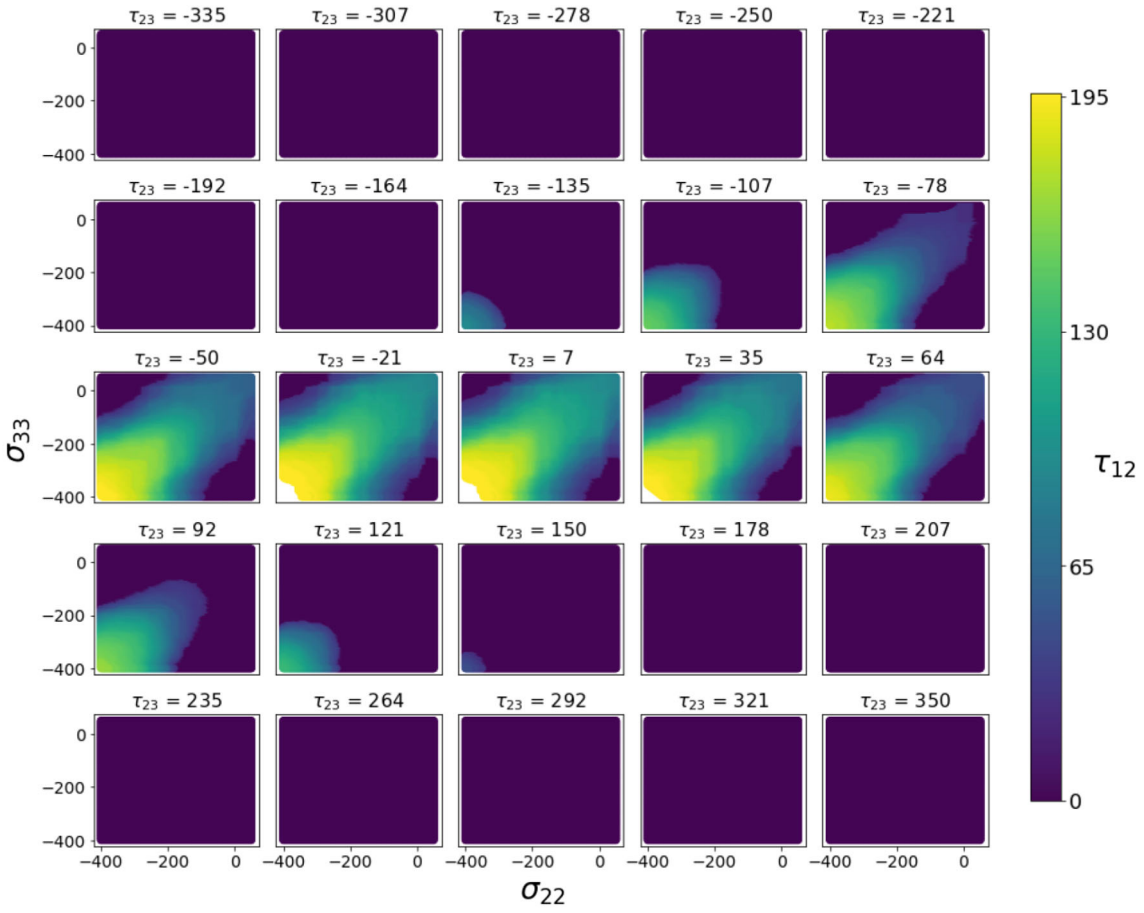

FIGURE 16 Failure envelope generated by the LGBM classification model for E-Glass/LY556 over τ_{23} values

16 nodes each. Each layer used a ReLU activation function. The NN model converged within just 30 epochs. The models were each evaluated on their performance on the hold out test set as well as on the failure envelope created by the predicted crack angles. Prediction results for the crack angle are displayed in Table 2. The results are plotted in Figures 7, 8, and 9.

Figure 7 shows the failure envelope predicted by the competing algorithm, the BS algorithm, and the classification models. The dots are from experimental data reported in Reference 24. The failure envelope obtained by the competing algorithm and the BS algorithm can be regarded as benchmark analytical results. Data-driven models were trained using data generated by the BS algorithm. From Figure 7, it seems that the LR and LGBM classification models cannot effectively capture the failure envelope, while the NN model has significantly improved accuracy. The black curve and green curve, corresponding to the competing algorithm result and BS algorithm results, are identical, hence only the green curve can be seen in Figure 7.

Figure 8 contains results obtained by the data-driven regression models. It should be reiterated here that the regression models output the crack angle with the maximum damage initiation potential. Then, the crack angle is substituted into Equations (1)–(8) to evaluate the damage initiation status. According to Figure 8, all the regression models except for the LGBM model are accurate, in fact the similarity of predictions between models makes it hard to differentiate the plotted results. The red, green, and black curves are on top of each other. The NN model has the highest accuracy that the predicted failure envelope is on top of the competing algorithm and BS algorithm results.

More details of the crack angle prediction is investigated and presented in Figure 9. For each data point on the predicted failure envelopes in Figure 8, there is a corresponding crack angle. The predicted crack angles varying σ_{22} are plotted in Figure 9. It is noticed that the crack angle prediction by the NN model agrees very well with the BS algorithm. However, the LR and LGBM models are not able to obtain correct crack angles, with the LGBM model performing the worst.

The results provided show that clearly for plane stress it is better to approach the problem as a regression problem and to predict the crack angle. It is

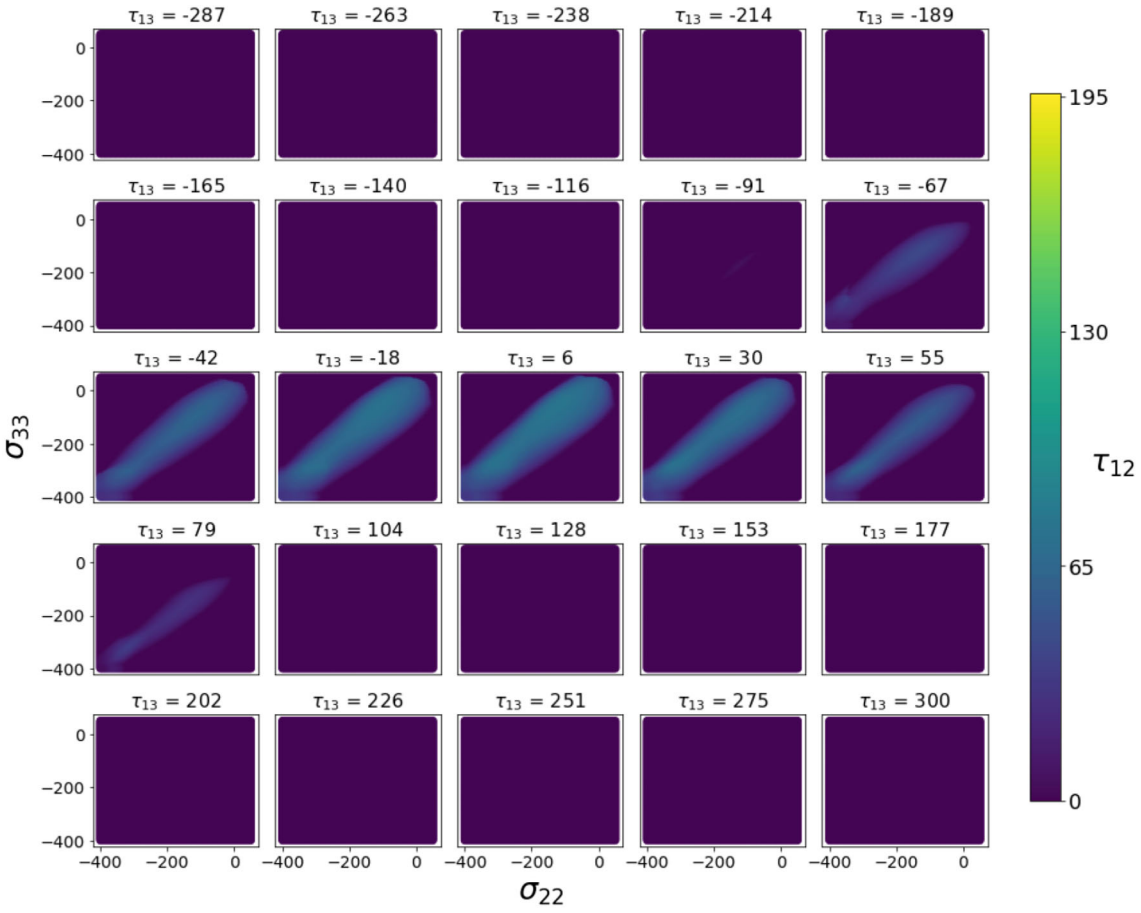

FIGURE 17 Failure envelope generated by the NN classification model for E-Glass/LY556 over τ_{13} values

difficult for each of these models to accurately predict damage initiation as a classification problem for plane stress. This is likely due to the fact that there are fewer explanatory variables available. In the training data for plane stress there only exist eight explanatory variables making it difficult to make discrete predictions. Even moderately accurate models on the crack angle translate very well to the prediction of the failure envelope. For example, LR had only 75% accuracy when predicting the crack angle, however those same predicted crack angles produced a failure envelope that was over 98% accurate. In contrast, the LR model has far worse accuracy for the predicted failure envelope with damage initiation predictions despite a higher model accuracy at 83.5%.

The LGBM model performs reasonably well with respect to the test data but from the plot in Figure 8 it is clear that it is not the right tool for plane stress damage initiation predictions. The nature of tree-based methods leads to discrete predictions and, in turn, the jagged failure envelope prediction. For more complex models with additional explanatory variables this becomes less noticeable, but it does not seem to perform well enough to be used for plane stress.

Figure 9 shows that the LR model greatly oversimplifies the crack angle predictions while the NN fits nearly perfectly the shape of the crack angles. Despite this difference, the failure envelope prediction by each model are 98% and 99.99%, respectively. The sensitivity of the damage initiation to the crack angle is dependent upon the stress state and material strengths. Results as shown in Figure 9 may indicate that the damage initiation state is not closely related to the damage angle prediction in the current combination of material strengths and stresses. However, this is not universally true and requires more systematic examination in the future.

Revisiting Figure 8, another interesting observation is that all the algorithms produce results relatively similar with $\sigma_{22} \geq -67\,\text{MPa}$. With a lower stress value (higher compressive stress), the LGBM deviates from the other results. Referring to the damage angle–stress curves in Figure 9, it is shown that the ground-truth damage angle predicted by the BS algorithm starts to grow rapidly to large values after σ_{22} decreases less than $-60\,\text{MPa}$. Therefore, the reason for the LGBM result deviating from other results in Figure 8 may be that LGBM lacks the

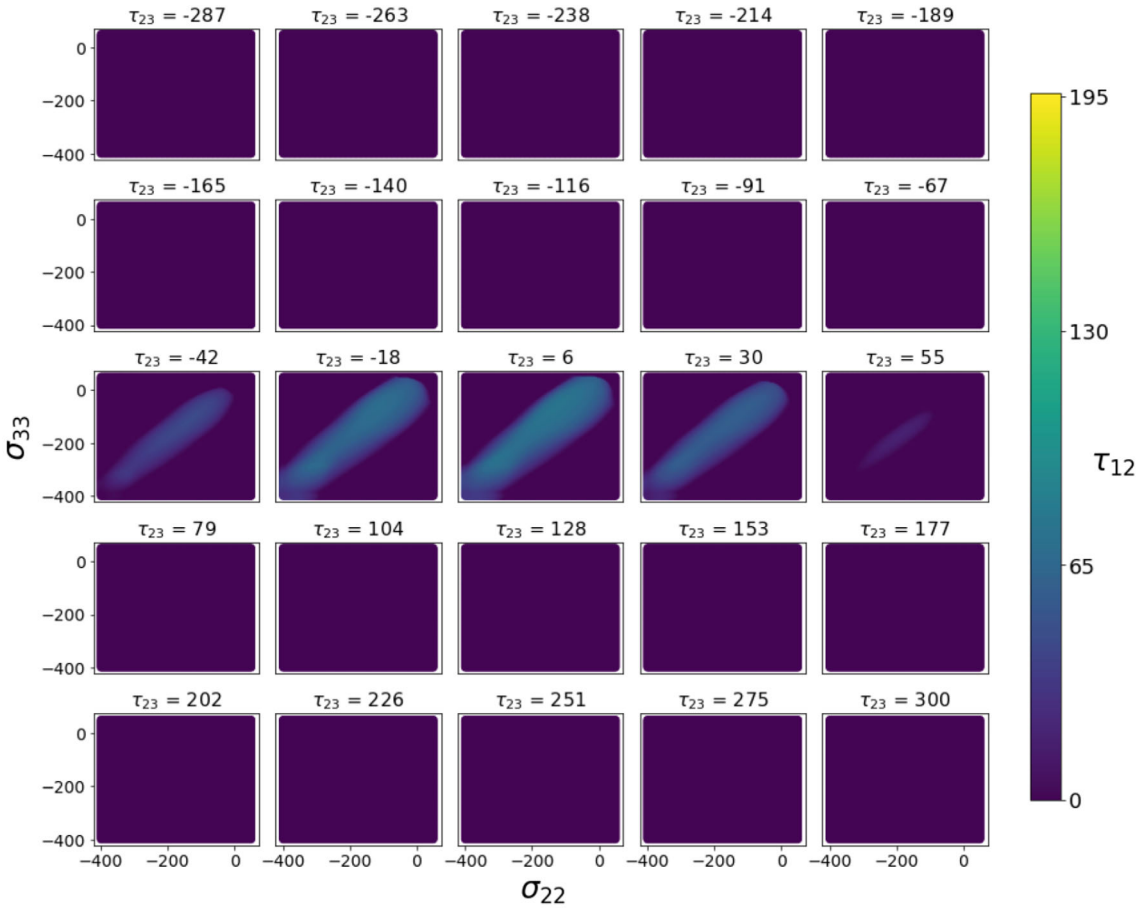

FIGURE 18 Failure envelope generated by the NN classification model for E-Glass/LY556 over τ_{23} values

capability to capture complex damage initiation scenarios with a large crack angle.

3.1.2 | Performance evaluation

LGBM, NN, and LR all run in $O(1)$ time with respect to the number of candidate failure angles, but they do vary in runtime depending on the complexity of the models themselves, so it is still important to evaluate performance in terms of actual run time. This evaluation was performed on a MacBook Pro with a 2.6 GHz 6-Core Intel Core i7 processor and all computations were done sequentially. Performance evaluation was performed by generating 10,000 random samples with randomized input data within the ranges used for training and having each model produce a result for each sample. The time reported is the average total time it took to score the full set of 10,000 samples over 20 iterations. The error bars represent the 95% confidence interval. The performance results are displayed in Figure 10. As shown in the figure, the BS algorithm immediately provides a significant improvement over the run time of the competing algorithm. This serves as an important benchmark because it shows the improvement that can be gained without using any machine learning techniques. LGBM clearly does not perform efficiently enough to be considered as an alternative over the BS, however, the NN and LR models certainly could provide a significant efficiency improvement. Additionally, LR and NN both performed very well when predicting failure envelopes using crack angle predictions, meaning that the increase in efficiency does not necessarily indicate a decrease in accuracy.

It is always important to maximize the accuracy of predictive models, but for predicting damage initiation criteria, it is equally important to make the models as efficient as possible. The NN trained for plane stress was extremely simple with only two hidden layers of 16 nodes each. This allows the NN runtime to be comparable to the LR runtime. Had the NN architecture been more complex, the accuracy could have been marginally better but runtime would have decreased quickly. By using the smaller architecture, failure envelope accuracy still achieved 99.99% while retaining efficient performance.

These results show that machine learning can be used to very accurately predict crack angle and failure envelopes

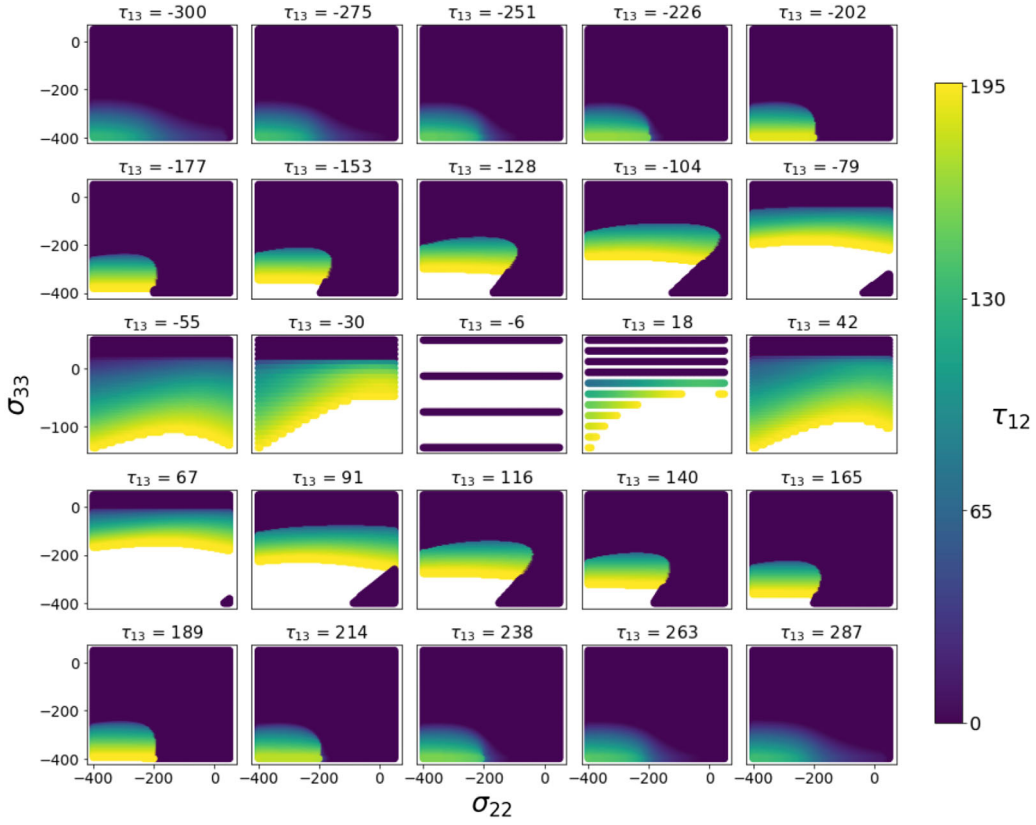

FIGURE 19 Failure envelope generated by the LR regression model for E-Glass/LY556 over τ_{13} values

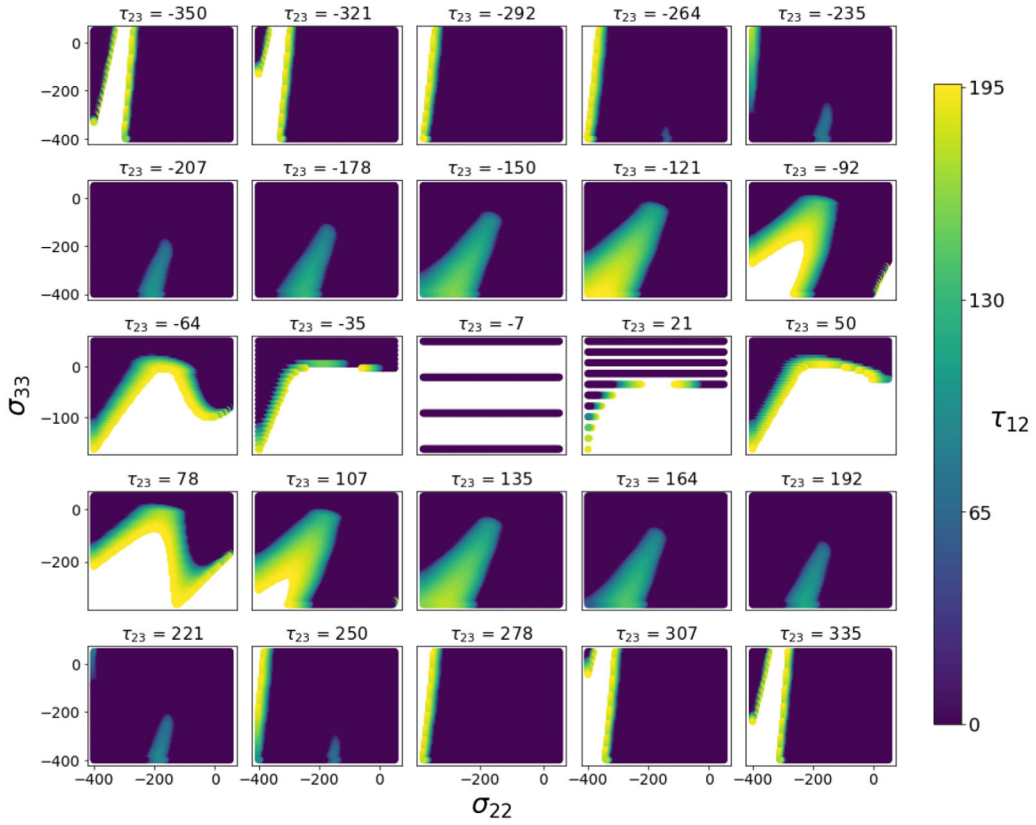

FIGURE 20 Failure envelope generated by the LR regression model for E-Glass/LY556 over τ_{23} values

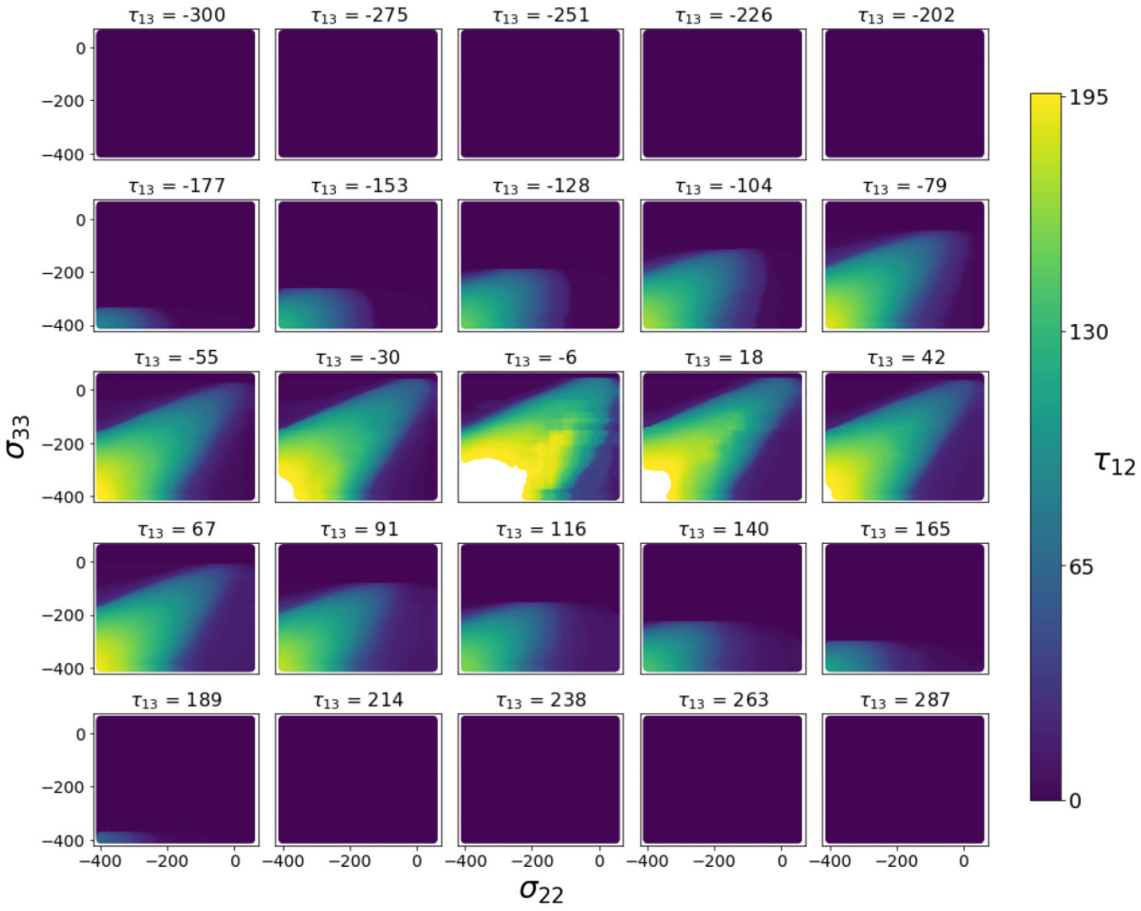

FIGURE 21 Failure envelope generated by the LGBM regression model for E-Glass/LY556 over τ_{13} values

for plane stress use cases. Specifically, predicting the crack angle and subsequently using that prediction in the damage initiation criteria formulas is the best approach for predicting failure envelopes for composite materials in plane stress. BS is an option that can compute the crack angle and failure envelopes with exact precision in a fraction of the time of the competing algorithm without requiring machine learning. However, neural networks can provide even faster execution time while sacrificing less that 0.01% in accuracy with respect to the failure envelope.

3.2 | Matrix damage initiation with 3D stress states

As mentioned in Section 2.3, the added complexity of the 3D stress state means that BS could no longer be used to generate a baseline. Additionally, it is more complicated to generate metrics on failure envelopes due to the higher dimensionality of the data. LR, NN, and LGBM models were trained as regression models predicting crack angle and classification models predicting failure, similarly to the plane stress use case. R^2 and MAE values for regression models were evaluated on a hold out test set as were the Accuracy and LL values for the classification models. In the interest of simplicity, R^2 and MAE values on the resulting failure envelopes for all models were computed using the predicted τ_{12} values. This was done because this is how those same metrics were reported for the plane stress use case.

The NN model for 3D stress states required a much larger architecture than for plane stress to achieve good accuracy. The NN model was the same architecture for crack angle and failure predictions with five hidden layers of 128 nodes each. For crack angle predictions, each layer had a ReLU activation function and the loss metric was MAE. For failure classification, each layer had a Sigmoid activation function and the loss metric was Log Loss. The LR model remained an SGDRegressor for crack angle predictions and SGDClassifier for failure classifications.[30] Finally the LGBM max bins were set to be 500 and the number of boosting rounds were set to 2000, identical to the plane stress model, however, the learning rate was adjusted for 3D stress. The learning rate for crack angle predictions was set to 0.01 and for failure classification it was set to 0.005. These adjustments were made to help the model converge faster on the more

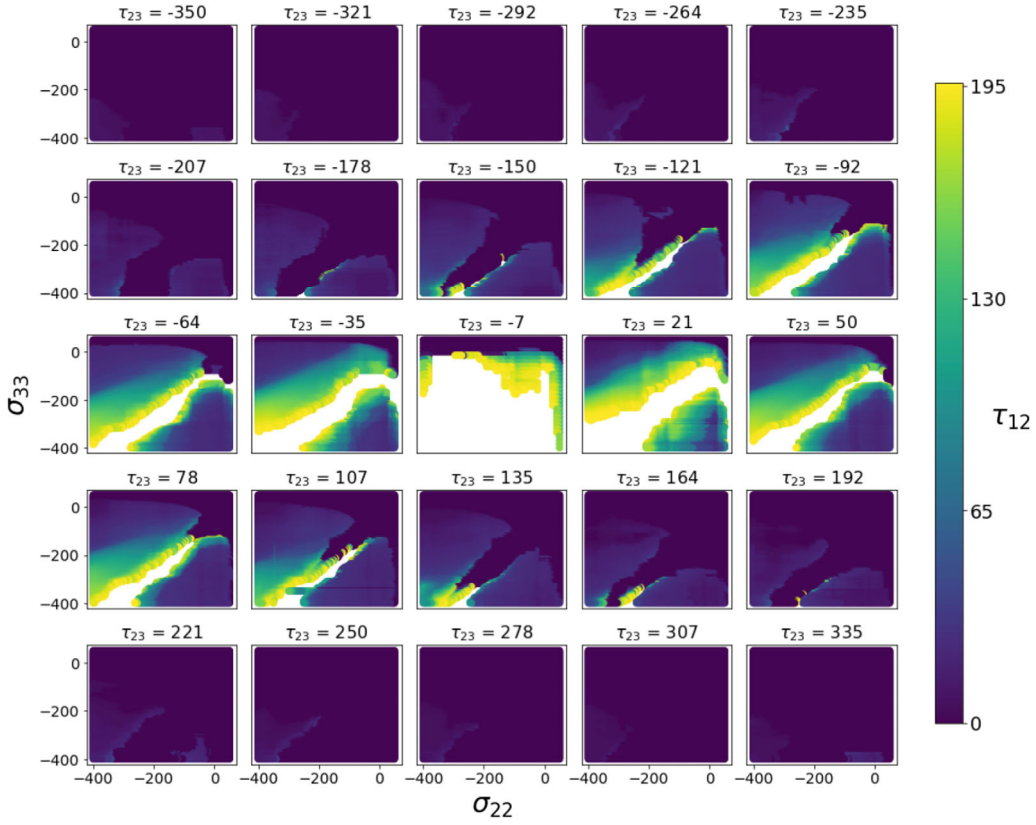

FIGURE 22 Failure envelope generated by the LGBM regression model for E-Glass/LY556 over τ_{23} values

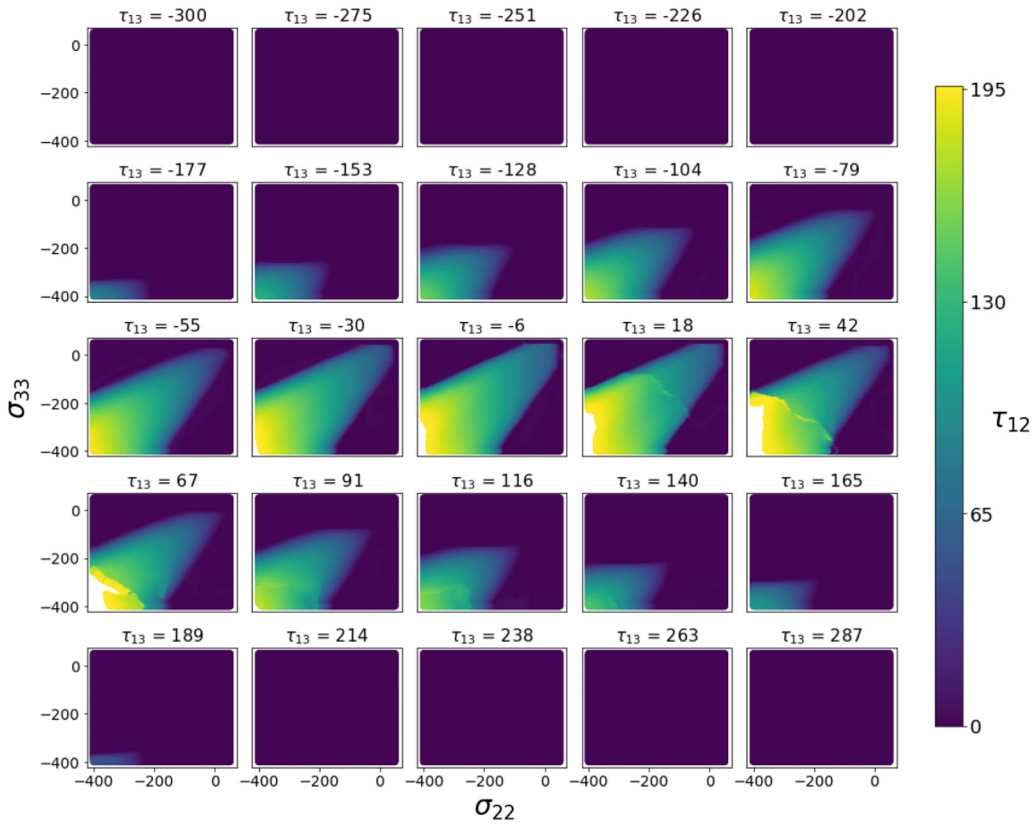

FIGURE 23 Failure envelope generated by the NN regression model for E-Glass/LY556 over τ_{13} values

complex use case at the possible expense of accuracy, however it was observed that smaller learning rates would require a large number of iterations and would not provide substantial efficiency improvement to computing the damage initiation potential value.

3.2.1 | Prediction evaluation

Similarly to plane stress, regression models were trained to predict the crack angle as well as classification models to predict damage initiation. The baseline failure envelopes obtained by the competing algorithm can be visualized in Figures 11 and 12. In these figures, the *x*-axis shows σ_{22}, the *y*-axis shows σ_{33}, and the color represents the τ_{12} values. Each individual plot shows those values for a given value of τ_{13} or τ_{23} in Figures 11 and 12, respectively. In some cases, the predicted failure envelopes contain areas for which there is no failure predicted. In these cases, the plots appear to have blank spaces that are not filled in with τ_{12} values. This indicates that for the provided range of τ_{12} values, there did not exist a value where failure occurred. The evaluation results are shown in Table 3. The failure envelopes with 3D stress states predicted by the data-driven classification and regression models are presented in Figures 13–24.

3.2.2 | Failure envelopes predicted by the classification models

Failure envelopes predicted by the LR classification model are shown in Figures 13 and 14. Comparing to the baseline solution in Figures 11 and 12, it is seen that the LR classification model fails to capture the failure envelope even remotely.

The LGBM classification model results are shown in Figures 15 and 16. The predicted failure envelopes seem quite similar to that in Figures 11 and 12. According to Table 3, the model performs quite poorly on the hold out test set with just 24% Accuracy value but does reasonably well overall on the failure envelope with an R^2 of 0.79.

The NN classification model results are shown in Figures 17 and 18. According to the figures, the NN classification model manages to visually capture the shape of the

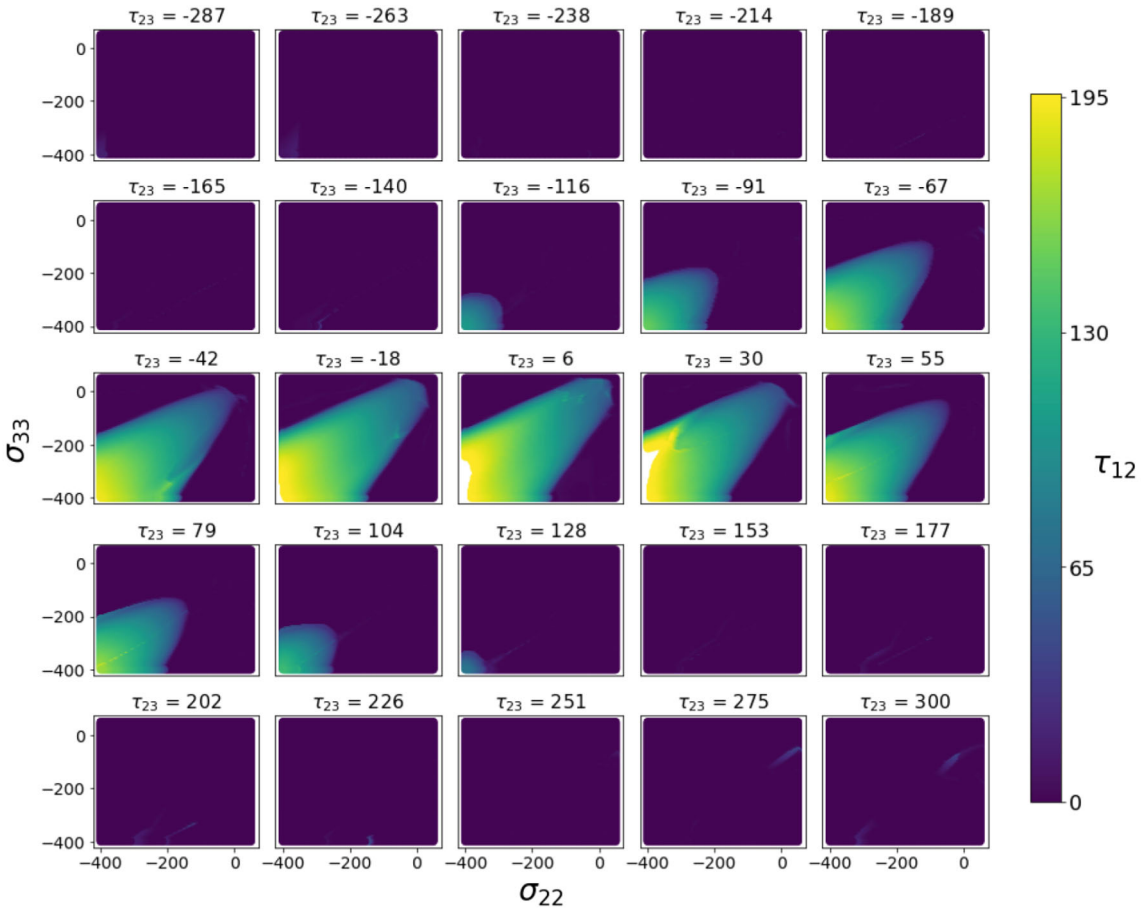

FIGURE 24 Failure envelope generated by the NN regression model for E-Glass/LY556 over τ_{23} values

failure envelope but struggles to capture the appropriate magnitude of τ_{12} values. This is also evidenced by the evaluation metrics with only a 0.14 R^2 value for the failure envelope prediction, as shown in Table 3.

3.2.3 | Failure envelopes predicted by the regression models

The LR regression model predicting crack angle only has a 0.30 R^2 but a 0.74 R^2 for the corresponding failure envelope. While 74% is moderately accurate for a failure envelope prediction, the LR model fails to capture the shape of the data in some important cases, particularly when there is high curvature to the failure envelope, as shown in Figures 19 and 20.

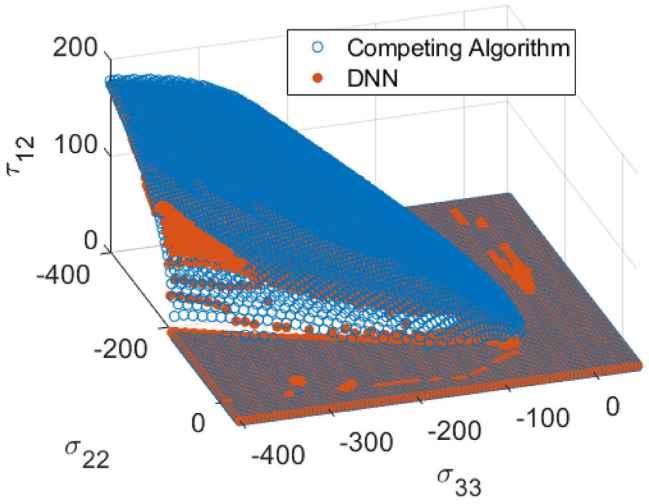

FIGURE 25 3D visualization of failure envelopes predicted by the competing algorithm and the NN model

The LGBM model behaves slightly differently, with an R^2 value of 0.73 on the crack angle and a comparable R^2 of 0.74 on the resulting failure envelopes as shown in Figures 21 and 22. While the R^2 value is reasonably high, the model clearly does not perform well in all cases. Figure 22 shows that for τ_{23} values with high curvature the model does not remotely match the baseline.

Finally, the NN model performs well on crack angle prediction with a 0.92 R^2 but performs much better on the resulting failure envelope generation with an R^2 of 0.99, explaining over 99% of the variance in the failure envelope. Figures 23 and 24 show the fit is very good with only minor deviations from the competing algorithm baseline. This is consistent with the plane stress models in that the crack angle prediction needs only to be moderately accurate in order to produce near perfect failure envelopes.

A 3D visualization of a failure envelopes predicted with the competing algorithm and the NN model are shown in Figure 25. As seen, the NN predicts the damage initiation very accurately, with the capability to capture highly curved parts of the failure envelope.

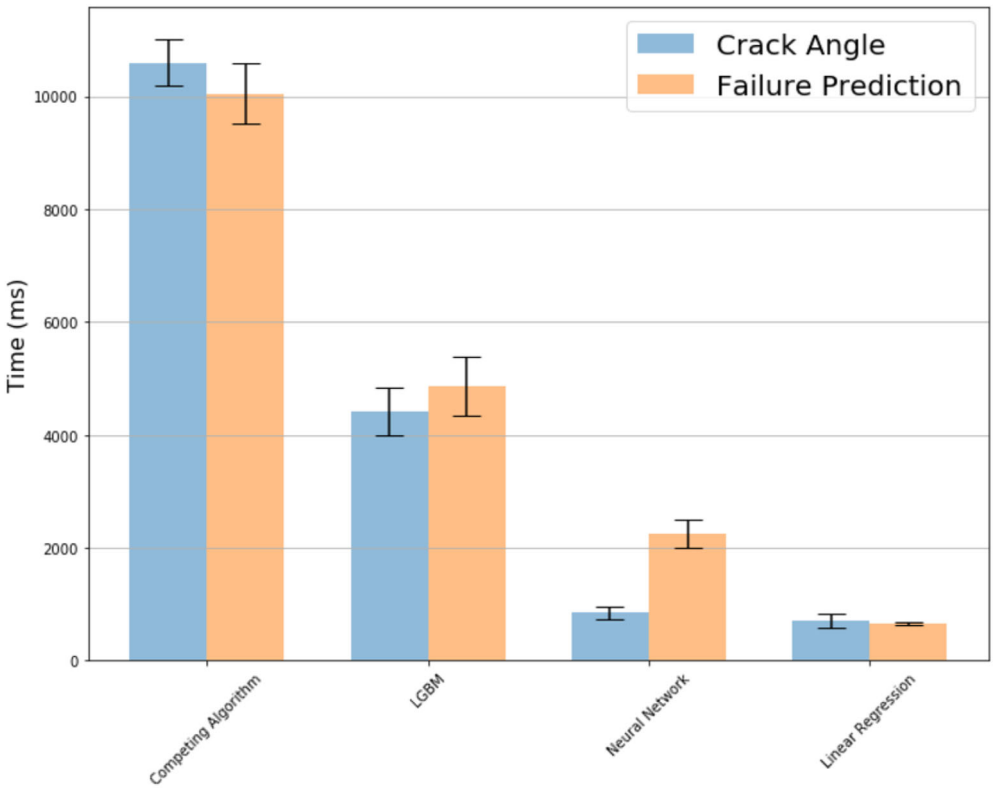

FIGURE 26 Performance comparison between different model types for 3D stress state

3.2.4 | Performance evaluation

Performance evaluation was performed in the same manner as for the plane stress models. They were run on a MacBook Pro with a 2.6 GHz 6-Core Intel Core i7 processor and the time measure is the time required to determine whether or not damage would occur in 10,000 randomly sampled stress states. The results can be visualized in Figure 26 The results are similar to the plane stress results with all predictive models outperforming the competing algorithm in terms of efficiency. LGBM is the slowest of the predictive models for both regression and classification. This is likely because LGBM was optimized to improve training speed but not necessarily inference speed.[28] LR is the fastest model for both regression and classification, which is expected because it is the simplest model considered. Neural networks performed comparably to the LR model for crack angle predictions but performed much slower for classification. This is because of the use of the Sigmoid activation function. Computing the Sigmoid activation function is significantly slower than computing the ReLU activation function.

4 | DISCUSSIONS AND CONCLUSIONS

This paper has discussed the use of predictive models to determine damage initiation criteria in composite materials for the purpose of reducing computational cost. For plane stress scenarios, greater efficiency can be achieved most simply through utilizing a BS algorithm to search for crack angles with the greatest damage initiation potential. This is desirable because using a BS algorithm provides exact results using the same mathematical models as the competing algorithm. However, using machine learning the computational cost can be reduced even further.

Machine learning models require large, representative training data to produce useful results. Training data is easy to generate using existing simulation methods but must be carefully constructed to contain realistic samples. Randomizing input data is important to achieve sufficient distribution of input features, however, is prone to generating unrealistic samples. Without filtering of randomly generated samples to ensure realistic stress states, the training data generation described in this paper would produce a dataset with over 85% unrealistic stress states—that is, stress states which could never be achieved in practice. If machine learning models are trained on these data, they will not be optimized for real world problems.

Additionally, the problem use case should be well defined and models should be evaluated on that desired use case. For example, in this paper accuracy was defined as the ability to predict the failure envelope of a test composite material. Machine learning models were trained to predict that failure envelope directly though a classification task, however those models were not the best suited to predict the failure envelope most accurately. It was far more effective to utilize a hybrid approach of predicting the failure angle and feeding that prediction back into the mathematical models to produce the failure criterion. In fact, it was found that models that were even moderately effective in predicting failure angle could produce highly accurate failure envelopes through this method. The LR model for plane stress had only a 0.75 R^2 score on crack angle but predicted the failure envelope with 98% accuracy. Similarly for 3D stress, the most accurate model was the NN which predicted the failure envelope with 99% accuracy yet predicted crack angle with just 91% accuracy.

With respect to the computational cost savings, LR and neural networks provide over 5 times speed up from BS and 11 times improvement over the competing algorithm for plane stress while sacrificing little to no accuracy in failure envelope predictions. For 3D stress, the NN was able to provide approximately a 12 times speed improvement over the competing algorithm while retaining 99% accuracy in failure envelope prediction. It is worth noting that these performance measurements were taken by measuring the time to make 10,000 predictions. In a finite element model, there could be millions of predictions required, and because predictive models run in $O(1)$ compared to $O(n)$ for the competing algorithm, the speed difference between the methods would become more exaggerated as the number of iterations increase.

In conclusion, machine learning models provide an opportunity to greatly decrease the computational cost for damage initiation criteria computations while sacrificing little in terms of accuracy. In the future, the NN model will be trained using a combination of computational data and experimental data with variable fidelity.[31,32] In addition, the trained NN model will be implemented into the EST model for better computation efficiency.

ORCID

Anthony M. Waas https://orcid.org/0000-0002-4916-2102

REFERENCES

[1] W. H. Ng, A. G. Salvi, A. M. Waas, *Compos. Sci. Technol.* **2010**, *70*, 1126.
[2] W. Tan, B. G. Falzon, L. N. Chiu, M. Price, *Compos. A: Appl. Sci. Manuf.* **2015**, *71*, 212.
[3] Y. Shi, T. Swait, C. Soutis, *Compos. Struct.* **2012**, *94*, 2902.
[4] S. Pinho, L. Iannucci, P. Robinson, *Compos. A: Appl. Sci. Manuf.* **2006**, *37*, 63.

[5] M. Lamborn, R. A. Schapery, *J. Compos. Mater.* **1993**, *27*, 352.
[6] E. J. Pineda, A. M. Waas, *Int. J. Fract.* **2013**, *182*, 93.
[7] Z. Hashin, *J. Appl. Mech.* **1980**, *47*, 329.
[8] A. Puck, H. Schürmann, *Compos. Sci. Technol.* **2002**, *62*, 1633.
[9] R. M. Christensen, *J. Appl. Mech.* **2013**, *81*, 1.
[10] R. M. Christensen, *Compos. Sci. Technol.* **2019**, *182*, 107718.
[11] Z. P. Bažant, B. H. Oh, *Mater. Struct.* **1983**, *16*, 155.
[12] J. G. Rots, P. Nauta, G. Kuster, J. Blaauwendraad, *Heron* **1985**, *30*(1), 1985.
[13] N. Nguyen, A. M. Waas, *Compos. Struct.* **2016**, *156*, 253.
[14] M. H. Nguyen, A. M. Waas, *Compos. Part C: Open Access.* **2020**, *3*, 100073.
[15] A. P. Joseph, P. Davidson, A. M. Waas, *Compos. Struct.* **2018**, *203*, 523.
[16] P. P. Camanho, C. G. Davila, M. De Moura, *J. Compos. Mater.* **2003**, *37*, 1415.
[17] R. Cuntze, *Compos. Sci. Technol.* **2004**, *64*, 487.
[18] O. Völkerink, E. Petersen, J. Koord, C. Hühne, *Comput. Struct.* **2020**, *236*, 106280.
[19] M. Rezasefat, A. Gonzalez-Jimenez, M. Giglio, A. Manes, *Compos. Struct.* **2021**, *278*, 114654.
[20] A.-S. Kaddour, M. J. Hinton, *J. Compos. Mater.* **2013**, *47*, 925.
[21] J. Hou, B. You, J. Xu, T. Wang, *Polym. Compos.* **2022**.
[22] M. Özkan, A. Karakoç, M. Borghei, J. Wiklund, O. J. Rojas, J. Paltakari, *Polym. Compos.* **2019**, *40*, 4013.
[23] A. Post, S. Lin, I. Ustun, A. M. Waas, AIAA SCITECH 2022 Forum, 0104, 2022.
[24] P. Soden, M. Hinton, A. Kaddour, *Compos. Sci. Technol.* **2002**, *62*, 1489.
[25] S. Lin, A. M. Waas, *J. Appl. Mech.* **2021**, *88*, 081001.
[26] S. Lin, S. I. Thorsson, A. M. Waas, *Compos. Struct.* **2020**, *251*, 112530.
[27] S. Ruder, *ArXiv Preprint ArXiv:1609.04747* **2016**.
[28] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, *Adv. Neural Inf. Proces. Syst.* **2017**, *30*, 3146.
[29] J. H. Friedman, *Ann. Stat.* **2001**, *29*, 1189.
[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, *J. Mach. Learn. Res.* **2011**, *12*, 2825.
[31] K. Tian, X. Ma, Z. Li, S. Lin, B. Wang, A. M. Waas, *Int. J. Solids Struct.* **2020**, *193*, 1.
[32] K. Tian, Z. Li, J. Zhang, L. Huang, B. Wang, *Compos. Struct.* **2021**, *273*, 114285.

How to cite this article: A. Post, S. Lin, A. M. Waas, I. Ustun, *Polym. Compos.* **2023**, *44*(2), 932. https://doi.org/10.1002/pc.27144