Fall 11-22-2021

# Data-driven damage initiation criteria for carbon fiber reinforced polymer composites

Alexander Richard Post
*DePaul University*, apost1@depaul.edu

# DATA-DRIVEN DAMAGE INITIATION CRITERIA FOR CARBON FIBER REINFORCED POLYMER COMPOSITES

BY

ALEXANDER RICHARD POST

A THESIS SUBMITTED TO THE SCHOOL OF COMPUTING, COLLEGE OF COMPUTING

AND DIGITAL MEDIA OF DEPAUL UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN DATA SCIENCE

DEPAUL UNIVERSITY
CHICAGO, IL
2021

DePaul University

College of Computing and Digital Media

# MS Thesis Verification

This thesis has been read and approved by the thesis committee below according to the requirements of the School of Computing graduate program and DePaul University.

Name: Alexander Richard Post

Title of dissertation: DATA-DRIVEN DAMAGE INITIATION CRITERIA FOR CARBON FIBER REINFORCED POLYMER COMPOSITES

Date of Dissertation Defense: 11/22/2021

**Dr. Ilyas Ustun**

Advisor*

**Dr. Jacob Furst**

1st Reader

**Dr. Rafaella Settimi-Woods**

2nd Reader

* A copy of this form has been signed, but may only be viewed after submission and approval of FERPA request letter.

# Abstract

**Computational progressive failure analysis (PFA) is vital for the design, verification, and validation of carbon fiber reinforced polymer (CFRP) composites. However, the computational cost of PFA is usually high due to the complexity of the model. 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 thesis, data-driven models are developed to determine the matrix damage initiation based on the Mohr-Coulomb model and Hashin model. 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. We have demonstrated the usage of BS and the training and evaluation of machine learning models as alternative methods to determine macroscale matrix damage initiation. With the data-driven methods, over 99% of the computational time has been saved, while the predictive accuracy stays 99.9% from the traditional approach. For a 3D stress state, 87% of the computational time is saved with the predictive accuracy of 94.7%.**


# I. Introduction

CFRP composites are being used widely in many industry sectors thanks to their high stiffness/strength-to-weight ratio, fatigue resistance, corrosion resistance, etc.. Due to the complexity of the composite materials and structures, the damage is usually of complex patterns and modes. Typical damage modes include fiber tensile/compressive breaking, matrix tensile/compressive/shear cracking, and delamination between laminae. PFA codes have been developed for decades to simulate and predict the damage of composites. For most of the cases, it is essential to predict the correct damage patterns. As one of the most important components of a PFA code, damage initiation criteria play an important role in determining the transition of the material from undamaged or microscopically damaged to macroscopically damaged. Therefore, it is important to establish physically sound, accurate, and robust damage initiation criteria.

Damage initiation criteria developed by Hashin have been demonstrated to be accurate in modeling fiber damage and matrix damage [1, 2]. The capability to predict matrix compressive damage initiation has been improved by Puck and Schurmann by adopting the Mohr-Coulomb model [3] and compared with the "World-wide Failure Exercise" (WWFE) results [4]. The model developed by Puck and Schurmann [3] is a combination of the Hashin model and the Mohr-Coulomb model, where the angle of the crack needs to be determined. The process of searching for the fracture plane angle is carried out iteratively as described in [5] and will be outlined in detail in Section 2.1.

High-fidelity PFA results have been obtained by computational codes established upon damage

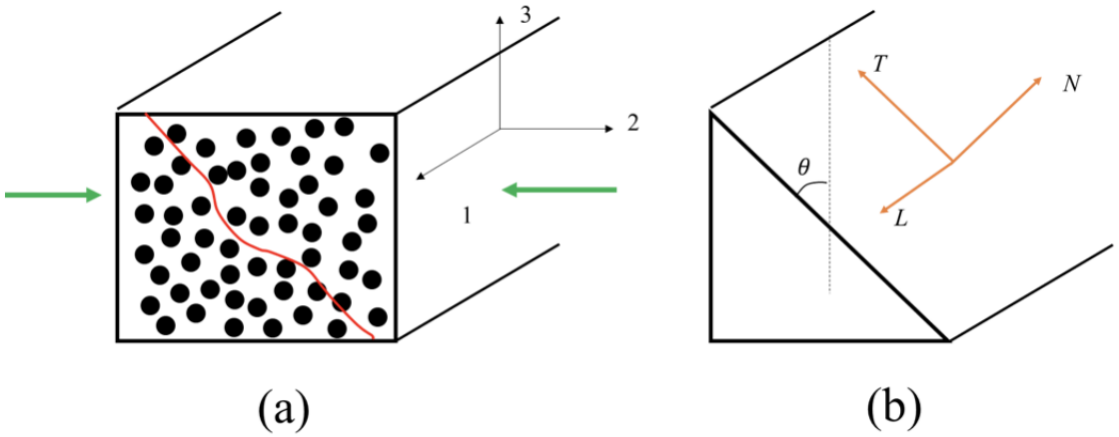


**Fig. 1 (a) A lamina under transverse compression in the 1-2-3 coordinate system, and (b) the failure plane in the L-N-T coordinate system.**

initiation criteria by Hashin and that based on the Mohr-Coulomb model [2, 4]. However, for the Mohr-Coulomb components of the criteria, computational time is a disadvantage since the algorithm is usually run iteratively to search for the crack angle. The major purpose of this article is to apply and evaluate several algorithms to predict the damage initiation and the crack angle accurately and efficiently. The first type of algorithm is similar to that implemented in [2], referred to as the competing algorithm. The second type is usually implemented to locate values in sorted arrays, named the Binary Search (BS) algorithm. The third type is based on machine learning models, specifically linear regression and neural networks. The three types of algorithms are described and compared in terms of accuracy and efficiency for matrix damage initiation with 2D plane stress states and 3D stress states.

## II. Theoretical Background of Mohr-Coulomb Failure Analysis

The Mohr-Coulomb (M-C) failure criterion is widely used for matrix compressive failure and to describe frictional sliding. It assumes that the matrix damage initiation and propagation occur at a certain combination of the greatest and least principal stresses. It has been observed that composite materials fail in transverse (2 and 3 directions) compression by shearing along the failure plane oriented at an angle $\theta$ with respect the loading direction, as shown in figure 1.

The M-C criterion is defined as in equation 1.

$$|\tau_{cr}| = S_i + |\sigma_N| \tan\phi \qquad \sigma_N < 0 \tag{1}$$

where:

$\tau_{cr}$ = shear stress on the crack plane

$\sigma_N$ = normal stress on the crack plane

$S_i$ = shear strength of the material

$\phi$ = angle of friction

The angle of friction ($\phi$) is a material parameter measured from transverse compressive tests, related to the internal coefficient of friction $\mu$ as shown in equation 2.

$$\phi = \tan^{-1} \mu \tag{2}$$

Figure 2 shows a geometric representation of the MC failure criterion where the purple circles are Mohr circles corresponding to each axis and the green line is the so called Coulomb Fracture Line. The relationship between the crack angle $\theta$ and the angle of friction $\phi$ is shown in equation 3.

$$2\theta = \phi + \frac{\pi}{2} \tag{3}$$

According to the MC failure criterion, fracture occurs when any circle is tangent to the fracture line [6]. The angle of the fracture plane for a unidirectional laminate loaded in transverse compression is easily obtained via experimental tests and can be considered a material property, so called $\alpha_0$ [6]. The typical $\alpha_0$ for a composite under transverse compression loading is 53° from experimental studies [3]. However, under combined loads and the uncertainty of material properties the angle of the fracture plane is unknown. For this reason, the MC failure criterion in three dimensions is measured in terms of the traction planes defined with respect to the fracture plane angle shown in equation 4.

$$\begin{aligned} \sigma_{nn} &= \sigma_{22} \cos^2\theta + \sigma_3 3 \sin^2\theta + 2\tau_{23} \sin\theta \cos\theta \\ \tau_{nt} &= (\sigma_{33} - \sigma_{22}) \sin\theta \cos\theta + \tau_{23}(\cos^2\theta - \sin^2\theta) \\ \tau_{lt} &= \tau_{13} \cos\theta - \tau_{12} \sin\theta \end{aligned} \tag{4}$$

These traction planes are used to define the M-C failure criterion as it is used in practice. Using the geometry shown in 2, there have been several attempts at defining an accurate model in terms of these traction planes. Puck and Schurmann first proposed equation 5 in the general case [3] [7].

$$f_{mc} = \left(\frac{\tau_{NL}}{S_{12} - \eta_L \sigma_{NN}}\right)^2 + \left(\frac{\tau_{NT}}{S_{23} - \eta_T \sigma_{NN}}\right)^2 = 1 \qquad \sigma_{NN} < 0 \tag{5}$$

Another proposed solution by Dávila et al. is shown in equation 6.

$$f_{mc} = \left(\frac{\langle |\tau_T| + \mu_T \sigma_n \rangle}{S_T}\right)^2 + \left(\frac{\langle |\tau_L| + \mu_L \sigma_n \rangle}{S_L}\right)^2 = 1 \tag{6}$$

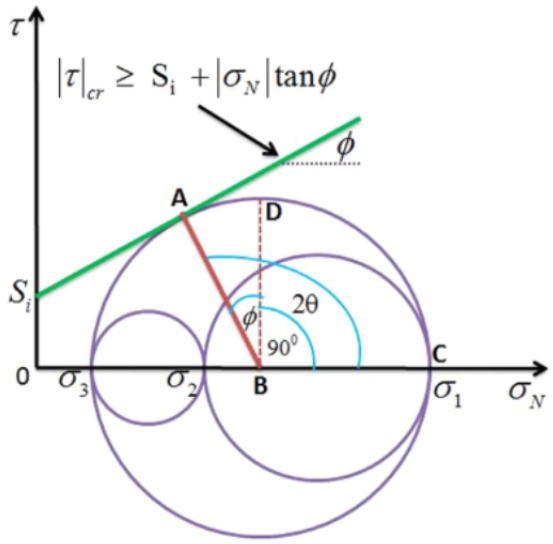


**Fig. 2 The M-C criterion illustrated with the Mohr circle [8]**

where the operator $\langle . \rangle$ is the Mc-Cauley bracket defined by $\langle x \rangle = max\{0, x\}, x \in \mathbb{R}$ and $S_L$ is the longitudinal shear strength. The difference between these two equations being that equation 5 considers that the compression stress increases the effective strength of the material, while equation 6 considers that the compression stress reduces the effective shear stress [7]. Additionally, Puck and Schurmann propose equation 7 as their final model due to a better fit to their experimental data.

$$f_{mc} = \frac{(\tau_T)^2}{S_T^2 - 2\mu_T S_T \sigma_n} + \frac{(\tau_L)^2}{S_L^2 - 2\mu_L S_L \sigma_n} = 1 \tag{7}$$

While all of these formulations provide certain benefits, each one is defined in terms of the traction planes, meaning that the fracture angle must be determined. Dávila and Camanho propose an algorithm that systematically checks possible angles and computes the failure criterion for each one, deciding the correct fracture angle is the one which maximizes the failure index (equation 5, 6, or 7) [6]. Noting that in the plane stress case, the correct failure angles never exceed $\alpha_0$, they propose limiting the domain of potential angles to be $0 < \alpha < \alpha_0$. To validate this assumption, distributions of failure angles were created by running this algorithm without the domain limitation. From figure 3, it is seen that all the possible fracture plane angles are smaller than 53°. In addition, it should be pointed out that the angle of 53° Assuming is dependent on the material properties of composites. Therefore, assuming $\alpha_0$ being equal to 53° is a valid assumption for plane stress states.

Additionally, Puck and Schurmann note that an analytical solution for finding the failure angle exists for plane stress [3]. They also note, however, that a closed form solution does not exist for a 3D stress state. The domain limitation described by Dávila and Camanho also is not applicable for 3D stress states. Using the algorithm they described [6] without imposing any limitation on the

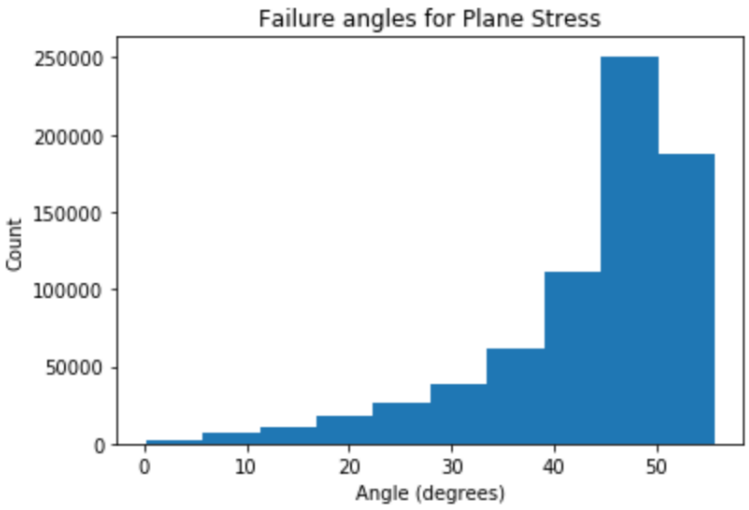


**Fig. 3 Distribution of failure angles for plane stress**

domain, a failure angles were collected for a wide variety of 3D stress states and the distribution is shown in figure 4. It is clear that in a 3D stress state the failure angle can range from 0° to 180° and there is no way to reduce the number of iterations required by limiting the domain to be less than $\alpha_0$.

## III. Fundamental Mechanics of Composite Materials and Finite Element Analysis

Composite materials are materials that are made of two or more materials on a macroscopic scale. CFRPs consist of two distinct materials - a fiber and a matrix. The matrix is used to bind fibers together and is usually considerably weaker than the fibers themselves, this can lead to some difficulties in modeling the behavior of these composites. Most common engineering materials are both homogeneous and isotropic, meaning that the material has uniform properties throughout and the properties are the same in every direction. Composite materials, however, are usually both heterogeneous and nonisotropic, and so there are slightly more complex methods needed to effectively model their behavior [9].

The two fundamental concepts in all materials are those of stress and strain. These two measures allow investigation into the behavior of materials under different conditions. Stress is a measure of the force put on an object over the area of that object, while strain is a measure of the change in length over the original length of the object. Additionally, stress and strain can be further broken down to normal and shear components. The normal component is acting perpendicular to the surface of the axis while a shear component acts parallel to the surface.

In addition to stress and strain, there are two important constants that are commonly used in modeling composite materials: Young's Modulus ($E$) and Poisson's Ratio ($\nu$). Young's Modulus


**Fig. 4 Distribution of failure angles for full stress state**

quantifies the relationship between tensile stress and axial strain in the linear elastic region of a material. The formula for computing Young's Modulus is in equation 8. The higher the modulus, the more stress is needed to create the same amount of strain. For example, a perfectly rigid body would have an infinite Young's modulus while a very soft material such as a liquid would have zero Young's Modulus. There is also the concept of a shear modulus that is used in conjunction with these two values. the shear modulus is a material property defined as the ratio between shear stress and shear strain.

$$E = \frac{\sigma}{\epsilon} \tag{8}$$

where:

$$\sigma = \text{force per unit area}$$

$$\epsilon = \text{proportional deformation}$$

When a material is subjected to tension or compression along one axis, strain is often also observed along the perpendicular axes. For example, when a rubber band is stretched, it not only becomes longer along the axis of tension, but it also becomes thinner. The rate of this change along the perpendicular axis is called Poisson's ratio. Poisson's Ratio measures the deformation of a material in directions perpendicular to loading and is defined as the negative ratio of transverse strain to axial strain as shown in equation 9.

$$\nu = -\frac{d\epsilon_{trans}}{d\epsilon_{axial}} \tag{9}$$

**A. Computing Stress and Strain**

Strain is easy to compute by simply measuring displacements. Normal strain is shown in equation 10 and shear strain is shown in equation 11.

$$\epsilon_{xx} = \frac{\partial u}{\partial x}, \epsilon_{yy} = \frac{\partial v}{\partial y}, \epsilon_{zz} = \frac{\partial w}{\partial z} \tag{10}$$

$$\gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}, \gamma_{zx} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}, \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \tag{11}$$

where:

$u$ = displacement in the x direction

$v$ = displacement in the y direction

$w$ = displacement in the z direction

Strain is often denoted in matrix form, illustrated in equation 12.

$$[\epsilon] = \begin{bmatrix} \epsilon_{xx} & \gamma_{xy} & \gamma_{zx} \\ \gamma_{xy} & \epsilon_{yy} & \gamma_{yz} \\ \gamma_{xz} & \gamma_{yz} & \epsilon_{zz} \end{bmatrix} \tag{12}$$

This matrix is symmetrical because the moments are zero - there is no rotation occurring. Due to the symmetry, only 6 unique strain values exist and are often expressed as a 6 by 1 matrix, shown in equation 13

$$[\epsilon] = \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \epsilon_{zz} \\ \gamma_{yz} \\ \gamma_{zx} \\ \gamma_{xy} \end{bmatrix} \tag{13}$$

Writing the strains in this way makes it convenient to compute the corresponding stresses by using a single stiffness matrix, or constitutive matrix, $C$ composed of material specific constants relating stresses and strains. The generalized equation for relating stresses and strains is in equation 14.

$$\sigma_i = C_{ij}\epsilon_j \qquad i, j = 1, ..., 6 \tag{14}$$

where:

$$\sigma_i = \text{stress components}$$
$$C_{ij} = \text{stiffness matrix}$$
$$\epsilon_j = \text{strain components}$$

Constructing the stiffness matrix is all that must be done to relate stresses and strains together and the values of each element vary depending on the type of material. For this paper, we assume a transversely isotropic material, meaning that at every point there is a one plane in which the mechanical properties are equal in all directions. For a CFRP, that plane would be the plane which is parallel to the fiber direction. For this case, the stress strain relationship in matrix form can be seen in equation 15 as well as the equations for the individual constants in the stiffness matrix in equation 16.

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \tau_{23} \\ \tau_{13} \\ \tau_{12} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{13} & 0 & 0 & 0 \\ C_{13} & C_{13} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & (C_{11} - C_{12})/2 \end{bmatrix} \begin{bmatrix} \epsilon_{11} \\ \epsilon_{22} \\ \epsilon_{33} \\ \gamma_{23} \\ \gamma_{13} \\ \gamma_{12} \end{bmatrix} \tag{15}$$

where:

$$\begin{aligned} C_{11} &= \frac{1 - \nu_{23}\nu_{32}}{E_2 E_3 \Delta} \\ C_{22} &= \frac{1 - \nu_{13}\nu_{31}}{E_1 E_3 \Delta} \\ C_{33} &= \frac{1 - \nu_{12}\nu_{21}}{E_1 E_2 \Delta} \\ C_{12} &= \frac{\nu_{12} + \nu_{32}\nu_{13}}{E_1 E_3 \Delta} \\ C_{13} &= \frac{\nu_{13} + \nu_{12}\nu_{23}}{E_1 E_2 \Delta} \\ C_{44} &= G_{23} \\ \Delta &= \frac{1 - \nu_{12}\nu_{21} - \nu_{23}\nu_{32} - \nu_{31}\nu_{13} - 2\nu_{21}\nu_{32}\nu_{13}}{E_1 E_2 E_3} \end{aligned} \tag{16}$$

and

$$G_{23} = \text{shear modulus in the 2-3 plane}$$

Using these equations it is possible to compute all stress values for a material by measuring only

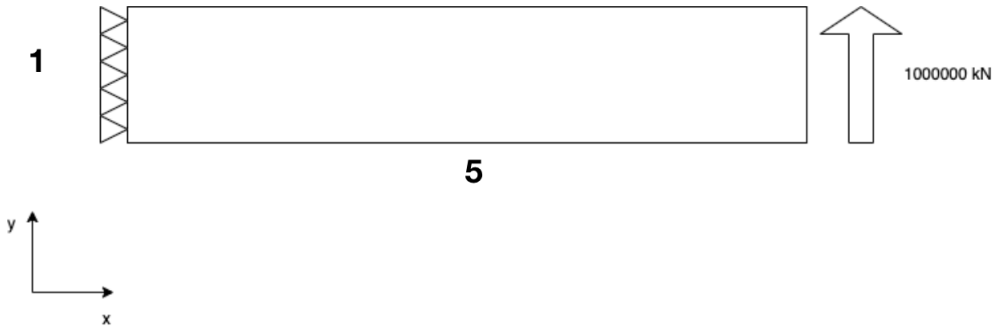


**Fig. 5 FEM example initial conditions**

the strain values. This is used extensively in modeling materials and, more specifically, in the use of finite element analysis to model materials.

**B. Finite Element Analysis**

The most basic idea behind the Finite Element Method (FEM) is the separation of a given domain into a set of simple subdomains - *finite elements*. Any shape that provides for the computation of the solution can be considered a finite element. Further, finite elements are comprised of selected points for which the solution will be approximated, these points are called *nodes* [10]. The collection of finite elements established to represent the object in question is known as a *mesh*. FEM problems concerning material loading can be generalized into the following steps:

1) Describe the material ($C$ Matrix, Young's Modulus, Poisson's Ratio)
2) Construct Mesh
3) Compute Stiffness Matrix
4) Set boundary conditions
5) Solve for global displacements: a = F\K
6) For each element, evaluate stresses and strains

*1. Plane Stress FEM Example*

As an example, consider a homogeneous and isometric rectangular plane 5 meters long and 1 meter wide. The material of this plane has a $E = 10e7$ MPa and $\nu = 0.3$ and the desired load is to be applied along the right edge while the left edge is held fixed. These initial conditions are illustrated in figure 5.

Since the first step of the FEM process is provided in the problem definition, the next step is constructing the finite element mesh. Constructing a mesh that is 20 elements long and 10 elements wide results in the mesh shown in figure 6.

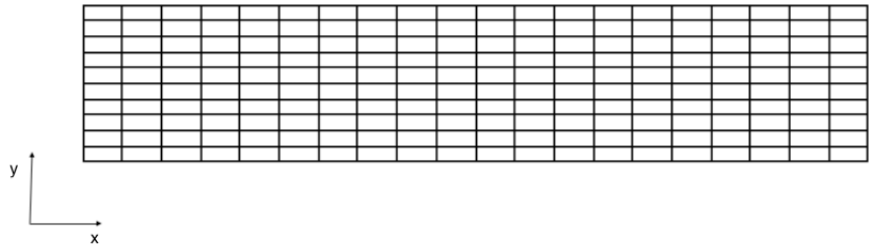


**Fig. 6 Finite Element Mesh**

Next, the $C$ matrix is determined using Young's Modulus and Poisson's Ratio. The $C$ matrix is only 3x3 in this example because plane stress considers only two dimensions, meaning the only stress values are $\sigma_{xx}$, $\sigma_{yy}$, and $\tau_{xy}$. Assuming the material is linear and elastic, the $C$ matrix used to relate stresses and strains is defined as shown in equation 17 [11].

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{bmatrix} = \begin{bmatrix} \frac{E}{1-\nu^2} & \frac{\nu E}{1-\nu^2} & 0 \\ \frac{\nu E}{1-\nu^2} & \frac{E}{1-\nu^2} & 0 \\ 0 & 0 & G = \frac{E}{2(1+\nu)} \end{bmatrix} \begin{bmatrix} \epsilon_{xx} \\ \epsilon_{yy} \\ \gamma_{xy} \end{bmatrix} \tag{17}$$

where:

$E$ = Young's Modulus (Modulus of Elasticity)

$\nu$ = Poisson's Coefficient

Using this defined $C$ matrix, a square stiffness matrix is constructed with dimensions corresponding to the number of nodes in the mesh. Boundary conditions are set based on the initial problem description. In this case, the boundary conditions are that the y axis is held fixed on the left edge of the object while force is applied to each node on the right edge. For each node, displacements are computed based on the constructed stiffness matrix and the amount of load applied. The resulting displacements relative to the x axis are shown in figure 7.

The displacements along the x axis shown in figure 7 show positive displacement along the bottom edge (tension) and negative displacement along the top edge (compression). Strain can be computed from the displacements and finally the stress can be computed from the strain by solving equation 17. The resulting stresses from this example are shown in figure 8.

Today, models such as those proposed by Hashin [1] and Puck [3] are applied through the use of the Finite Element Method (FEM). Since the number of nodes is proportional to the accuracy of the model, FEM models can easily amass millions of finite elements to reach a desired level of accuracy. Running a simulation over time for an FEM model requires solving equations for each individual element at each time step. This can be extremely computationally intensive and so ensuring efficient

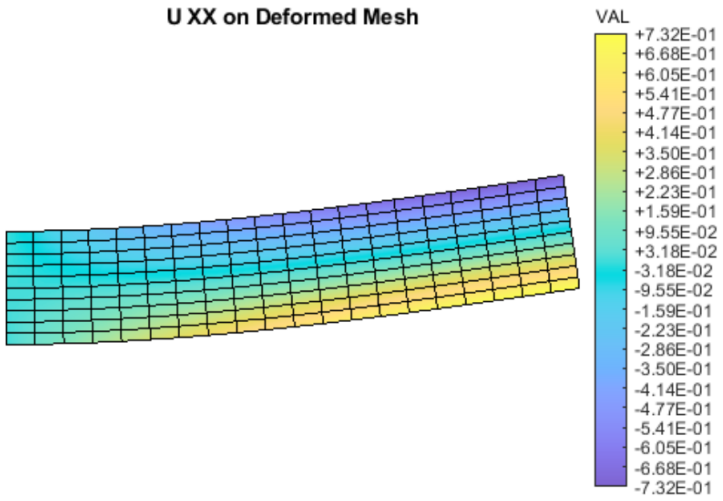


**Fig. 7 Plane Stress Displacements**

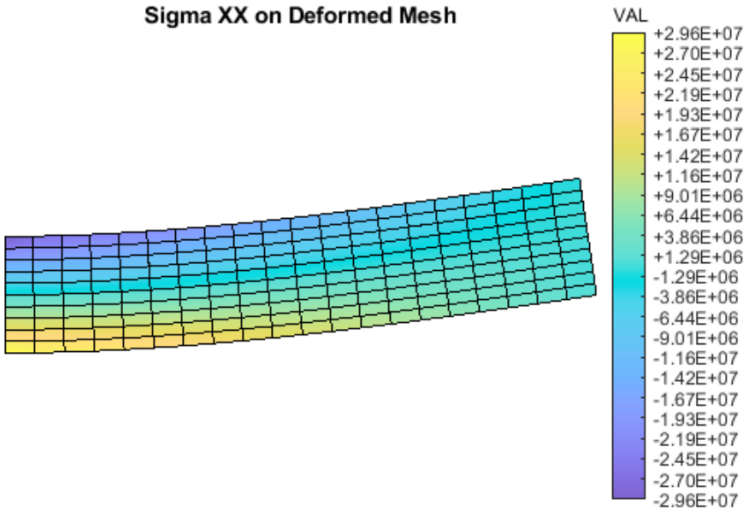


**Fig. 8 Stresses for plane stress example**

computations for each element is critical to creating an FEM model which can be run in a reasonable time frame.

## IV. Literature Review

Machine learning has been used in a variety of different applications with respect to composite materials. Supervised learning is used most commonly to predict properties of known materials given the vast amount of training data available from simulations. For example, machine learning methods can be used to predict the stress field of a composite material under fracture [12]. This type of supervised learning is known as the forward modeling problem and is based on known physics-based models as the training data is derived from simulations built using these models [12]. Another type of problem for which there has been some success in using machine learning up to this point is known as the inverse design problem. The inverse design problem involves designing new materials and often has to rely heavily on domain expertise and brute force searches through design spaces [12]. Brute force searches using optimization techniques such as gradient optimization, genetic algorithms, and simulated annealing have been proposed but all of these suffer from limitations when applied to large complex systems [13].

Linear regression is the simplest and most common machine learning technique available and has been used in a number of different materials applications. Due to its simplicity and explainability, Linear Regression is often considered in addition to more complex models like Artificial Neural Networks (ANN) as somewhat of a control group. Tiryaki et al. demonstrate a linear regression model applied to predict the compressive strength of heat treated wood [12, 14]. The authors used three input variables: Wood Species, Temperature, and Exposure Time to predict the compressive strength of heat treated wood based on a training data set created through experimentation containing 48 samples. Using Multiple Linear Regression (MLR) an $R^2$ value of 0.83 was obtained, indicating that 83% of the variance in the data was able to be explained by the model. The authors also trained an Artificial Neural Network model on the same data and found an $R^2$ value of over 99% [14]. While the authors claim the ANN model should be preferred to the MLR model given the significant improvement in accuracy, MLR provides greater explainability and given the small sample size could potentially generalize better.

Gu et al. propose machine learning as a solution to a 2-D composite design problem in which the design space contains $6.67x10^{240}$ combinations. This many possible combinations makes a brute force solution impossible and so an alternate solution using machine learning inspired by Google's DeepMind AlphaGo program. The authors use a two step process where the first step predicts toughness and strength of a composite material and the second step searches for high performance designs [13]. Once again, a linear model was considered along with a Convolutional

Neural Network (CNN). CNNs are typically used for image processing and are used for predicting high performance designs by treating each potential element like a pixel in an image. A FEM model is used to generate the ground truth data for this problem. The CNN outperforms the linear model by roughly 2% when predicting strength and toughness, producing an accuracy of roughly 98% compared to accuracies of 95% and 96% for strength and toughness, respectively, from the linear model. Though the CNN outperforms in this respect, it does not actually rank designs any better despite than the linear model despite the increased model complexity in the CNN. The authors conclude that high accuracy predictions can be achieved even with small training data and few training loops [13]. This work shows that even simple models like linear regression can be used to make complex design optimizations.

Another work by Yang et al. explores creating a deep learning model to capture the nonlinear mapping between the three-dimensional material microstructure and its macroscale stiffness in materials. Their objective was to predict macroscale elastic properties of a composite given its microstructure information. Given the shape of the microstructure information, the authors trained a CNN on simulated data to achieve this result. While the authors note room for improvement in the tuning and training of the CNN, they found a substantial improvement in accuracy of up to 54% over traditional physics based models [15].

Davidson and Waas demonstrate the usage of surrogate models in Monte Carlo simulations to analyze defects and their impact on the performance of composite materials. They used training data generated from high fidelity FEM models to train a Support Vector Classifier (SVC) to predict the failure mode of a composite (kink or split). The combination of Monte Carlo simulations and the SVC model were able to provide a successful framework for accurately predicting the compressive strength and failure mode of a defective structure. More importantly, the Monte Carlo simulation was able to converge within only 300 sampling points, representing a significant improvement over the FEM process [16].

One major limiting factor of using machine learning models for solving problems in composites is the generation of training data. Experimental training data is time consuming and expensive to produce, making it difficult to create training datasets of sufficient size. Fortunately, the FEM process makes it easy to produce vast amounts of simulation data which has opened the door to producing more useful machine learning models. Each of the works discussed above utilize simulated data to train their models with the exception of Tiryaki et al. who used only 48 experimental samples. With this data limitation removed by simulation, there is clearly a wide breadth of problems in composite materials that can be solved using machine learning an artificial intelligence. Not only has it been demonstrated that predictive models can outperform traditional physics based models [15] but they can also reduce the computational complexity of generating those predictions [16].

While there is no literature on using machine learning to predict damage initiation, there is quite

a bit of literature on the prediction of failure using other modeling techniques. Many of these works are discussed in section II and were the result of the "World Wide Failure Exercise" (WWFE) [17]. This exercise brought together researchers from around the world to determine a universal definition of what it means to have failure of a composite material and produced several new models to more accurately predict that failure. One key aspect of the WWFE was the curation of experimental test cases by which new models can be evaluated. These data points represented a critical benchmark by which models can be compared and are used in this thesis as reference points.

# V. Damage Initiation Criteria and Algorithms

## A. The Damage Initiation Criteria for Matrix Macroscale Damage

In this paper, the damage initiation criteria discussed are mainly concerned with that for determining the macroscale damage initiation of matrix. For the tensile damage initiation, the Hashin criterion is used. For the compressive damage initiation, a criterion based on the Mohr-Coulomb model is used. The related computations of the damage initiation criteria are performed for stress components on a slanted crack surface whose normal is marked as N, as shown in Figure 1. In Figure 1, the coordinate system 1-2-3 is the lamina coordinate system and N-L-T is the local coordinate system on the fracture surface. $\phi$ is the crack angle in figure 1.

### *1. Matrix Tensile Damage Initiation – The Hashin Criterion*

The Hashin criterion has proved to be capable of capturing matrix tensile cracking onset accurately. The stress-based Hashin criterion for matrix tensile damage initiation on a certain crack plane is

$$\left(\frac{\sigma_{NN}}{S_{22}}\right)^2 + \left(\frac{\tau_{NL}}{S_{12}}\right)^2 + \left(\frac{\tau_{NT}}{S_{23}}\right)^2 = 1 \qquad \sigma_{NN} \geq 0 \tag{18}$$

where $S_{22}$, $S_{12}$, $S_{23}$ are the matrix strength values in the corresponding directions.

### *2. Matrix Compressive Damage Initiation – The Mohr-Coulomb Criterion*

The Mohr-Coulomb criterion implemented in the matrix compressive cracking are sometimes also referred to as the Puck and Schurmann criterion [3]. The criterion is

$$\left(\frac{\tau_{NL}}{S_{12} - \eta_L \sigma_{NN}}\right)^2 + \left(\frac{\tau_{NT}}{S_{23} - \eta_T \sigma_{NN}}\right)^2 = 1 \qquad \sigma_{NN} < 0 \tag{19}$$

where, $\eta_L$ and $\eta_T$ are the internal coefficients of friction along the local L and T axes, which can be measured by matrix compressive tests.

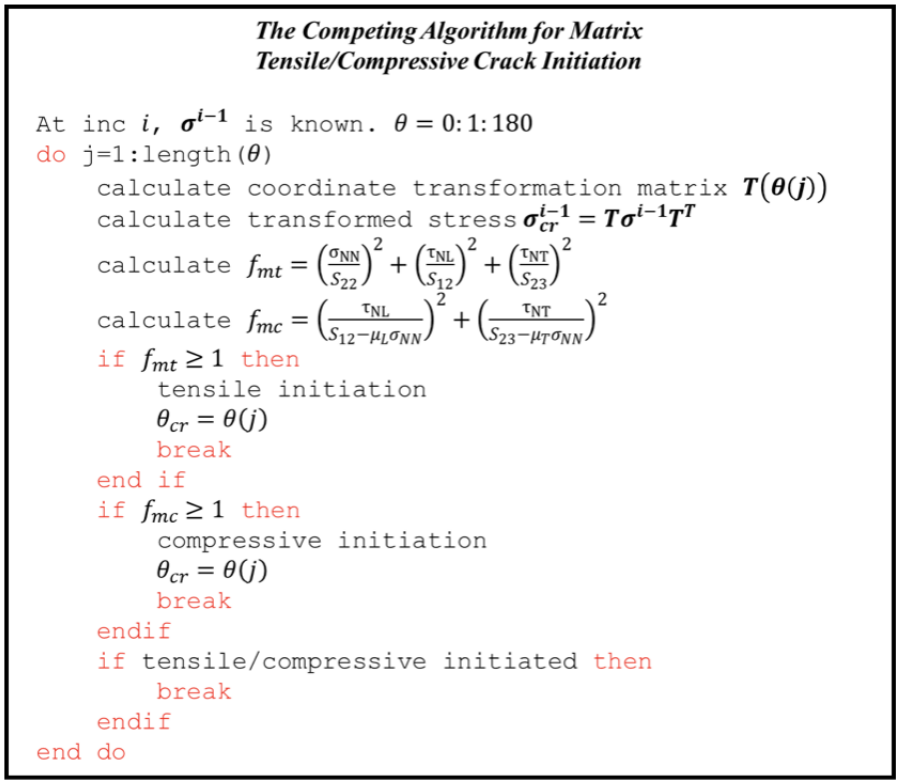


**Fig. 9 The competing algorithm to determine matrix tensile/compressive cracking and the crack plane.**

### B. Algorithms for Determining the Crack Angle

*1. The Big-O Notation for Evaluating the Computational Efficiency*

In this paper, we are concerned with evaluating algorithms not only by accuracy but also by computational efficiency. Effectively comparing the efficiency of algorithms using run times is misleading at best and at worst can provide the wrong results. The reason for this is twofold. First, different machines and different languages can produce vastly different results for the same problem, and even using the same environment and language, there is variability in runtime between multiple attempts. Second, algorithms often scale at different rates for different problems, so simply stating the run time for sample problems does not necessarily represent the algorithm performance in the general sense.

To avoid these issues, we compare algorithm efficiency using Big-O notation as a method of evaluating algorithm performance. Big-O notation specifically measures the "worst case" performance of an algorithm in terms of n, the size of the input data. Big-O notation is extensively used in computer science to measure the complexity and efficiency of algorithms rigorously. More details can be found in [18].

*2. The Competing Algorithm*

With a generic stress state $[\sigma_{11}, \sigma_{22}, \sigma_{33}, \tau_{23}, \tau_{13}, \tau_{12}]^T$, the crack plane needs to be determined based on if the stress state satisfies Equation 18 or Equation 19 first. A competing algorithm is proposed here. At each time increment, the crack plane angle $\theta$ is rotated from 0° to 180° with a certain interval, such as 1°. At each angle, Equations 18 and 19 are evaluated based on the previously calculated $[\sigma_{11}, \sigma_{22}, \sigma_{33}, \tau_{23}, \tau_{13}, \tau_{12}]^T$. Once the tensile/compressive damage initiation criterion is met, the crack plane is fixed and the post-peak degradation is performed on the crack plane. The flow chart of the competing algorithm to determine the crack plane and damage initiation is as shown in Figure 9, where the coordinate transformation matrix $T(\theta)$ is,

$$T(\theta) = \begin{bmatrix} cos(L,1) & cos(L,2) & cos(L,3) \\ cos(N,1) & cos(N,2) & cos(N,3) \\ cos(T,1) & cos(T,2) & cos(T,3) \end{bmatrix} \tag{20}$$

where, the axes 1-2-3 and L-N-T are as shown in Figure 1. The competing algorithm is a brute-force algorithm whose complexity of the algorithm is $O(n)$, which means that if there are n iterations to be performed in Figure 9, it would take n computations at most to find a solution for the crack angle.

*3. The Binary Search (BS) Algorithm*

BS is a search method used in computer science that can be used to locate values in a sorted array. The algorithm iteratively splits the given array in half, each time checking if the value at the center index is the target value, as demonstrated in Figure 10. Figure 10 describes the process of searching for the solution equal to 7 using the BS algorithm. A binary comparison is performed in each iteration to determine the next tentative solution until the final solution is reached. Since the array gets progressively smaller by half each iteration, this algorithm runs in $O(\log_2(n))$ time, where $n$ is the number of values in the array. It also guarantees that the target value is found if it exists in the array.

In this paper, BS is used to search for the angle which has the highest possibility of cracking based on the Hashin and Mohr-Coulomb model. In order to perform the BS algorithm, the problem as described in Figure 9 is firstly transformed into a sorted array with values $f_{MC}$ which is defined by Equation 21 and in Figure 9.

$$f_{MC}(\theta) = \left(\frac{\tau_{NL}}{S_{12} - \eta_L \sigma_{NN}}\right)^2 + \left(\frac{\tau_{NT}}{S_{23} - \eta_T \sigma_{NN}}\right)^2 \tag{21}$$

To improve efficiency, $f_{MC}$ is computed lazily, meaning that it is only computed at the $\theta$ values

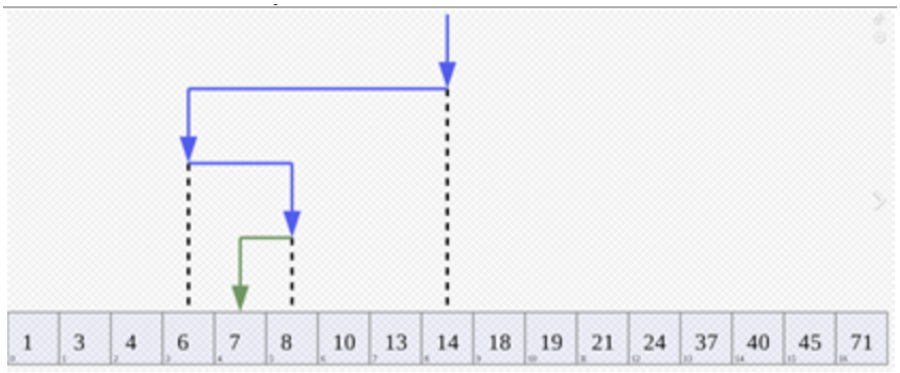


**Fig. 10 Illustration of the BS algorithm.**

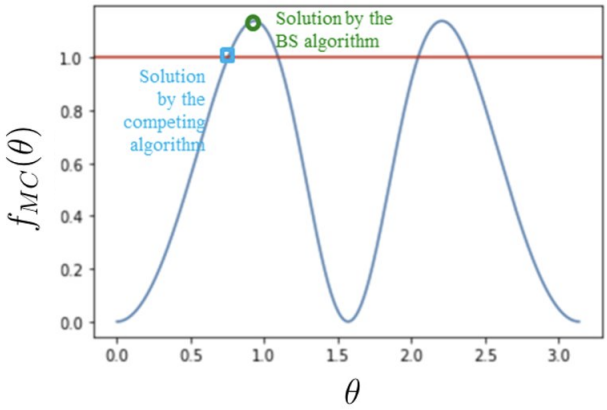


**Fig. 11 The distribution of $f_{MC}$ by $\theta$**

required for comparison in the BS algorithm. By plotting the variation of $f_{MC}$ as a function of $\theta$ under a plane stress state in Figure 11, we can see that the competing algorithm finds the solution $\theta$ where $f_{MC}(\theta)$ first achieves a value of 1 (compressive matrix damage initiation), while the BS algorithm finds the crack angle with the highest possible $f_{MC}(\theta)$ value, where it is most likely to have compressive matrix damage. Therefore, the BS algorithm is more efficient and physically accurate than the competing algorithm. For efficiency purposes, transformations are only performed at indexes required by the algorithm.

The application of the BS algorithm finds the maximum $\theta$ value specifically by finding a maximum. Therefore, the algorithm requires two assumptions be met to ensure that the value found i the global maximum:

1) The distribution of $f_{MC}(\theta)$ is bimodal on the domain $0 \leq \theta \leq \pi$
2) The distribution of $f_{MC}(\theta)$ is symmetrical about $\theta = \frac{\pi}{2}$

If these assumptions are not met then there is no guarantee that the BS algorithm will select the correct value of $\theta$ as it might find a local maximum that is not equal to the global maximum.

The complexity of the BS algorithm is $O(\log_2(n))$. Compared to the competing algorithm, this is a significant improvement in the computational efficiency. For example, in order to search for the crack angle $\theta$ among 1000 candidate angles ranging from 0 to $\pi$, in the worst case, the competing algorithm will have to perform 1000 computations while the BS algorithm will perform 10 calculations.

*4. Data-Driven Algorithms*

The data-driven algorithms applied in this paper are used to search for the crack angle. By predicting the crack angle $\theta$ with the highest potential of cracking, $\theta$ will be substituted back into Equations 18 and 19 to determine if the matrix tensile/compressive damage has initiated.

**Training Data**

Given that the crack angle is defined for all inputs, the machine learning performed in this paper is supervised learning. Training data generation involves simply computing failure criterion values and failure angles for a wide breadth of potential input values. This process can take a substantial amount of time. However, the amount of time required for training data generation is a one-time cost that does not impact the run time of predictive models.

The input parameters of the 2D plane stress models include stress components $\sigma_{22}$, $\tau_{12}$, matrix tensile strength $S_{22}$, matrix in-plane shear strength $S_{12}$, and matrix out-of-plane shear strength $S_{23}$. Through feature engineering, three additional "engineered" terms are also used as inputs, including $\sigma_{22}^2$, $\tau_{12}^2$, and $\left(\frac{\tau_{12}}{\sigma_{22}}\right)^2$ to provide additional information to be used by the predictive models. The addition of the quadratic parameters enables capturing the quadratic behavior instead of being bounded to simple linear behavior with the raw values. The addition of the ratio term is to reflect the interaction between stress components. Neural networks created without these engineered terms required larger architectures to produce the same results and so the interaction effects were used to avoid the need for extra nodes and hidden layers.

The input parameters for the 3D full stress models are similar to that of the 2D models with additional dimensions included. The included parameters include matrix strength values, raw values of normal stress and shear stress, quadratic terms for each stress value, and interaction terms for each combination of stress values. The interaction terms for the 3D full stress states are created by multiplication of stress values to avoid problems when dividing by zero. The resulting feature vector is defined as $[S_{22}, S_{12}, S_{23}, \sigma_{22}, \sigma_{33}, \tau_{12}, \tau_{13}, \tau_{23}, \sigma_{22}^2, \sigma_{33}^2, \tau_{12}^2, \tau_{13}^2, \tau_{23}^2, \sigma_{22} * \sigma_{33}, \sigma_{22} * \tau_{12}, \sigma_{22} * \tau_{13}, \sigma_{22} * \tau_{23}, \sigma_{33} * \tau_{12}, \sigma_{33} * \tau_{13}, \sigma_{33} * \tau_{23}, \tau_{12} * \tau_{13}, \tau_{12} * \tau_{23}, \tau_{13} * \tau_{23}]$. The interaction effects are produced for the same reason as in the plane stress use case. The output of the predictive models is the crack angle $\theta$.

The BS algorithm is used for generating the training data for 2D plane stress predictive models

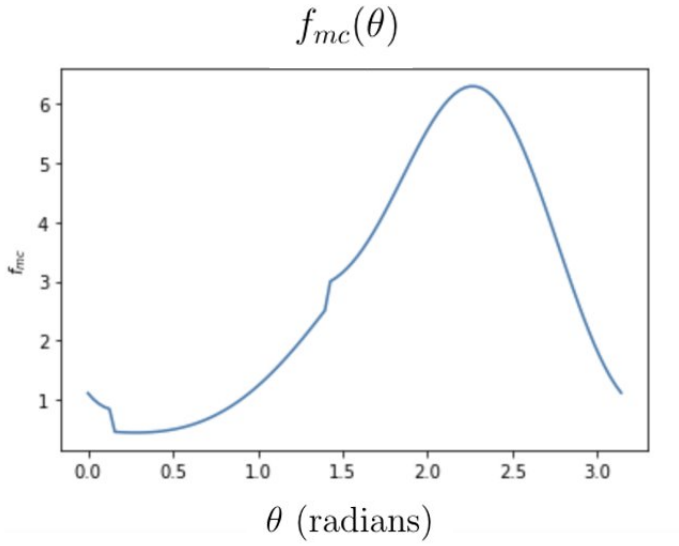


**Fig. 12 Crack Angle Distribution for one 3D stress state**

thanks to its efficiency and accuracy. Approximately 1.8 million data points were used for training. The Competing Algorithm must be used to generate training data for 3D stress states because the BS algorithm cannot guarantee the correct solution. The BS algorithm requires that the distribution of failure angles is bimodal and symmetric and that assumption does not hold in 3D stress states. One such counter-example is shown in figure 12. Approximately 3.8 million data points were used for training 3D models. For all models 80% of the data were used for training and 20% were held out for testing.

**Feature Selection**

Feature selection is an important part of predictive modeling. Refining the set of input features can reduce the complexity of a model to combat overfitting as well as improve the runtime efficiency of the model. For 2D plane stress there are only 8 features available and so no feature selection was performed. The training data for 3D full stress states, however, contained 23 features and so feature selection using Permutation Importance [19] was performed to reduce the number of input variables.

Permutation Importance, also known as Mean Decrease Accuracy (MDA), is a method of determining the "importance" of a given feature by measuring the decrease in score for a model when that feature is removed. For each available feature, that feature is replaced with random noise and the model is re-scored. The difference between the initial accuracy and the accuracy without the feature provides a score for that feature. Ultimately, features can be ranked by score and features with the lowest scores can be confidently removed from the training data. In this paper, the eli5 python package was used to generate Permutation Importance scores to aid in feature selection.

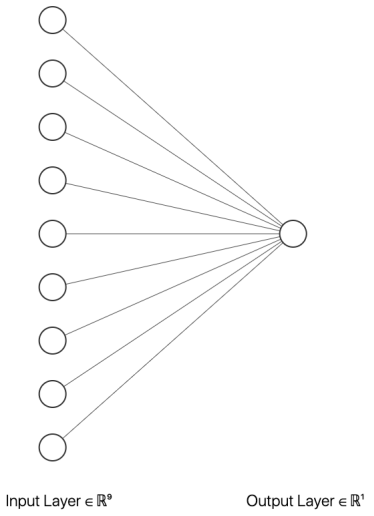


**Fig. 13 Illustration of a Linear Regression model**

**Linear Regression (LR)**

LR is one of the most common predictive models used in the field of machine learning. The model produces a linear combination of weights for each input feature to produce a prediction. To determine the weights, a stochastic gradient descent algorithm with an adaptive learning rate and an elastic net regularization term. One limitation of LR is the assumption that the relationship between the input and output variables is linear. We have attempted to mitigate this limitation by including quadratic and interaction terms. However, there are still models which may be better suited to this problem. The runtime of linear regression prediction is $O(1)$, meaning that the run time is constant with respect to the granularity of the crack angles. The structure of the LR is as illustrated in Figure 13. As seen, the input layer has 8 parameters, and the output layer has one parameter – the crack angle $\theta$.

**Neural Network (NN)**

NN provides a superior method of capturing non-linear relationships compared to linear regression. A NN typically consists of a multilayer perceptron with a non-linear activation function like Sigmoid or rectified linear unit (ReLU). The NN we used for this paper had two hidden layers with four nodes, each using a ReLU activation function. The architecture is illustrated in Figure 14. NN has a prediction run time that scales with the number of nodes, which means that it does not depend on the granularity of crack angles in our problem. Therefore, neural networks have a run time of $O(1)$, or constant time.

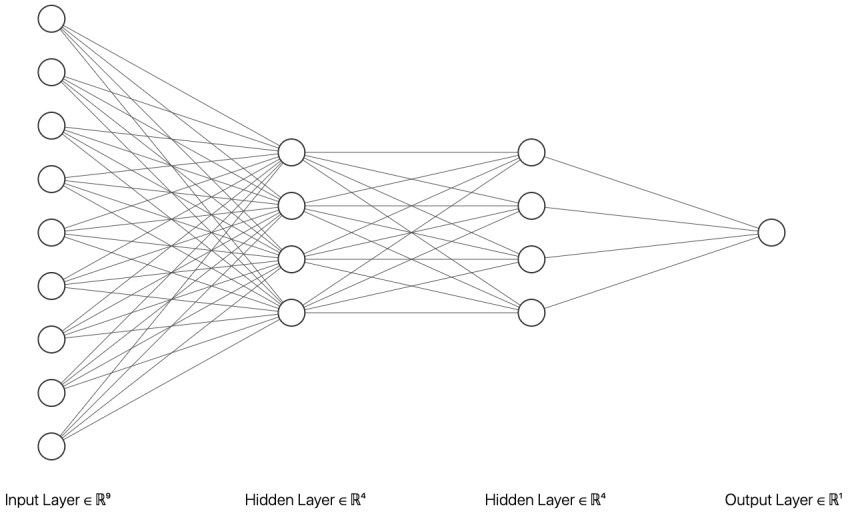


**Fig. 14 Illustration of the trained Neural Network model**

# VI. 2D Results

## A. Failure Envelope

The failure envelope is a curve that illustrates the combinations of $\sigma_{22}$ and $\tau_{12}$ leading to matrix damage initiation. The $R^2$ coefficient of determination has been used to compare failure envelopes generated by the algorithms described in Section V. The value of 1 indicates a perfect fit. The failure envelopes of plane stress states of the material E-Glass/LY556 predicted using the competing algorithm, and BS algorithm are compared in Figure 15 alongside the corresponding experimental data points from the WWFE [17]. These data are not used in training and can be considered a hold out testing set and the $R^2$ values discussed are produced from this hold out set. The WWFE data points are meant to be used as reference points and so evaluation metrics are not discussed with respect to these data. The purpose of this thesis is to investigate results with respect to the competing algorithm which is the current best option. It is seen that the failure envelope obtained by the two algorithms are identical, with an $R^2$ value being 1. Since the fit is 100% for the BS algorithm, we use the BS results as a baseline since the data generation is substantially faster.

The failure envelopes generated from the linear regression model and the neural network are shown in Figures 16 and 17, respectively. The linear regression model produced an $R^2$ of 0.899, and the neural network produced an $R^2$ of 0.999 with respect to the binary search results. It is seen that the NN model outperforms the LR model, especially with large compressive $\sigma_{22}$ values.

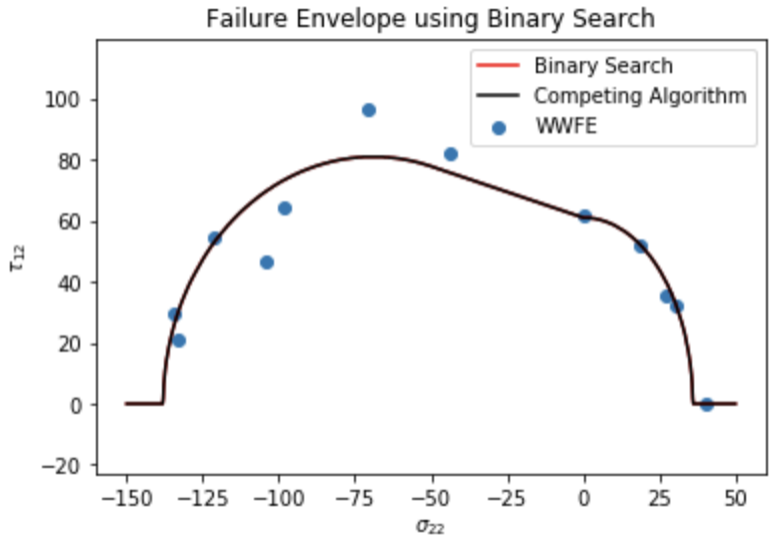


**Fig. 15 Failure envelopes predicted by the competing algorithm and the BS algorithm.**

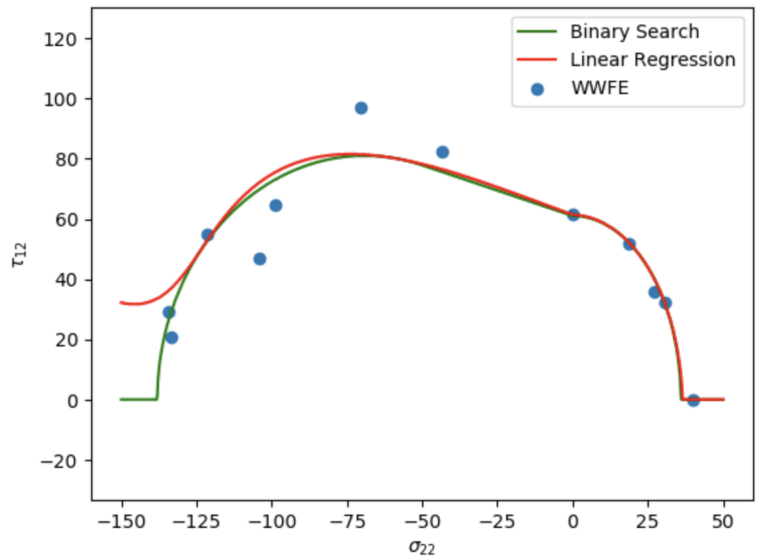


**Fig. 16 Failure envelopes predicted by the BS algorithm and the LR model.**

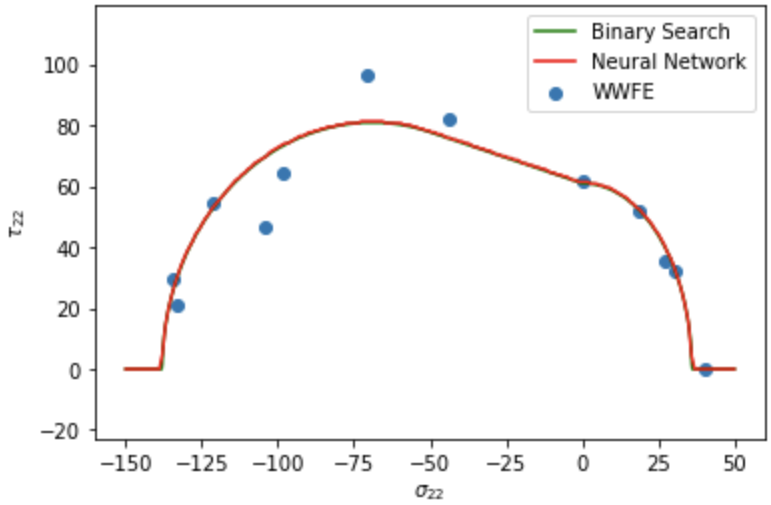


**Fig. 17 Failure envelopes predicted by the BS algorithm and the NN model.**

### B. Crack Angle

Another predicted value of concern is the crack angle at matrix damage initiation. Again, the $R^2$ value is used for measuring the accuracy of the algorithms. The crack angles predicted at damage initiation by the competing algorithm and the BS algorithm are plotted as a function of $\sigma_{22}$ in Figure 18. The curves agree well with each other. The curve obtained by the BS algorithm is smoother than that obtained with the competing algorithm. The $R^2$ value is 0.989, which is an indication that the BS is more accurate than the competing algorithm. This is expected as the competing algorithm does not find the crack angle with the maximum damage criterion value (as illustrated in Figure 11), and therefore produces a curve that is more jagged and slightly less accurate. Due to improved accuracy and efficiency using the BS algorithm, we use the BS results as the baseline when evaluating predictive model results.

Figure 19 shows the BS and LR results with an $R^2$ value of 0.856, and it is clear that the LR does not capture the shape of the crack angles sufficiently. On the other hand, the NN results illustrated in Figure 20 show an $R^2$ of 0.999 and clearly captures the shape of the damage angles.

### C. Evaluation of the Performance

To quantify the performance algorithms, the computational time of the applied algorithms for a case with 100 values of $\sigma_{22}$, 100 values of $\tau_{12}$, and 1000 possible crack angles are compared. Figure 21 shows the performance times for each algorithm on a log scale. It is obvious that the BS and predictive data-driven models are all substantially more efficient than the competing algorithm. Figure 22 shows the same information presented relative to the competing algorithm. The value performance multiplier, which quantifies the speed improvement, is used to evaluate the efficiency

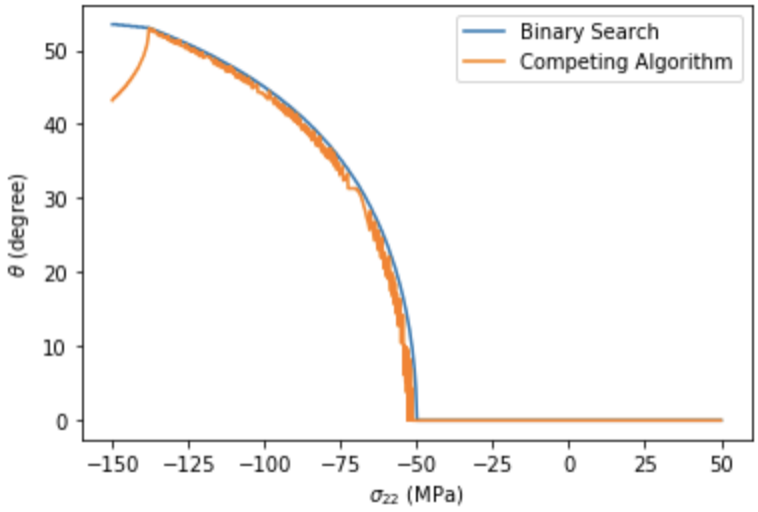


**Fig. 18 Crack angles at matrix damage initiation predicted by the BS algorithm and the competing algorithm.**

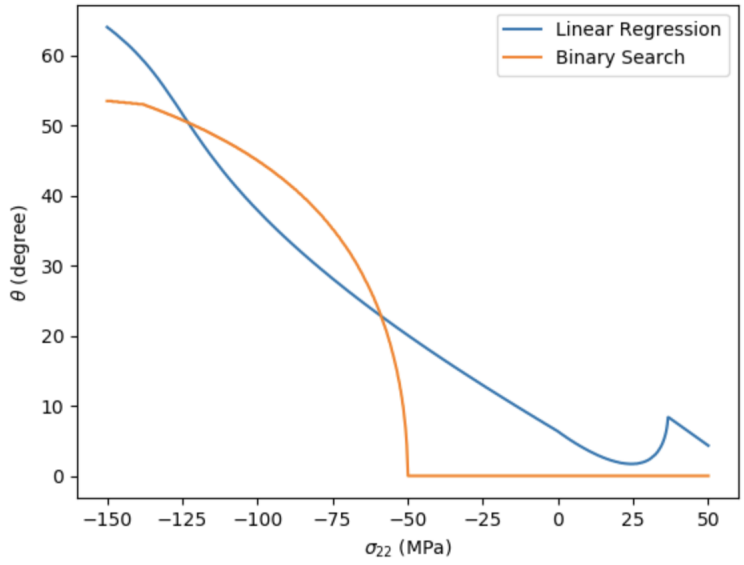


**Fig. 19 Crack angles at matrix damage initiation predicted by the BS algorithm and the LR model.**

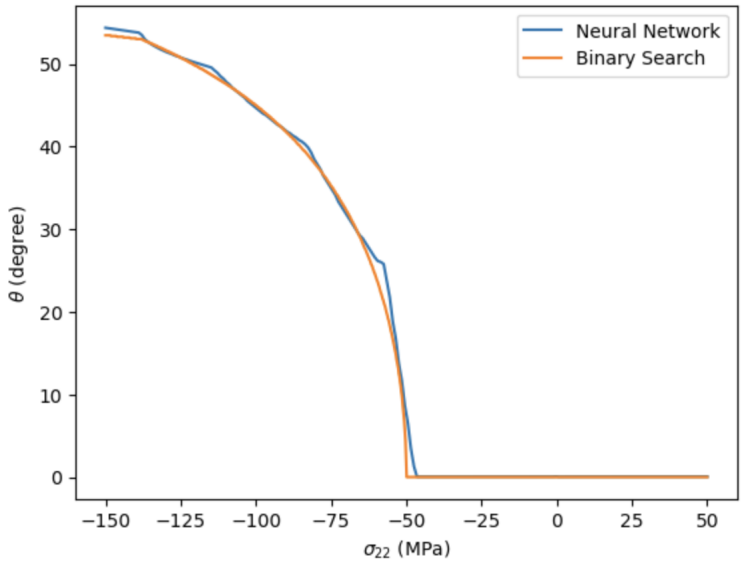


**Fig. 20 Crack angles at matrix damage initiation predicted by the BS algorithm and the NN model.**

of the algorithms. It is seen that the NN is the most efficient and runs approximately 131 times faster than the competing algorithm for this sample use case.

## VII. 3D Stress State Results

### A. Training the Neural Network

The full stress state requires a much more complex neural network than the plane stress state. Many different architectures were attempted but the most successful architecture had five hidden layers with 128 nodes each using a ReLU activation function. Using this configuration, feature selection was performed using permutation importance as the method for determining feature importance. Feature selection was performed by first categorizing each feature into the following buckets:

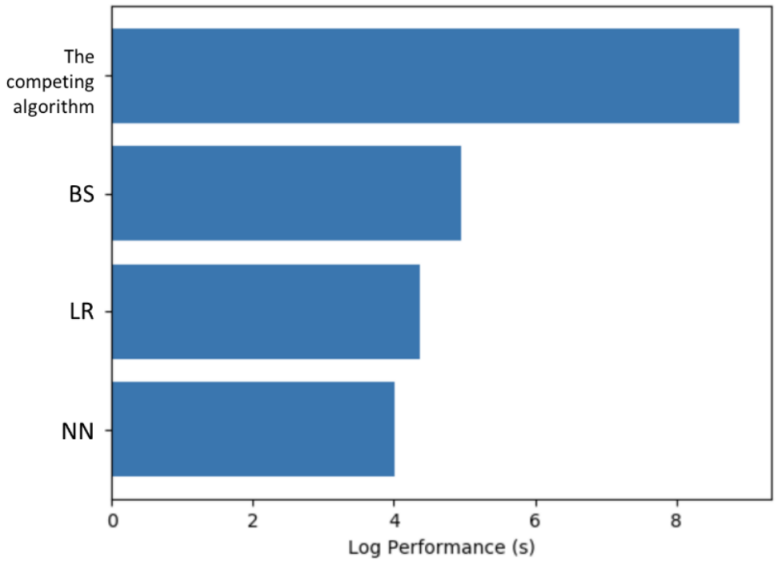


**Fig. 21 The comparison of performance in terms of the computational time.**

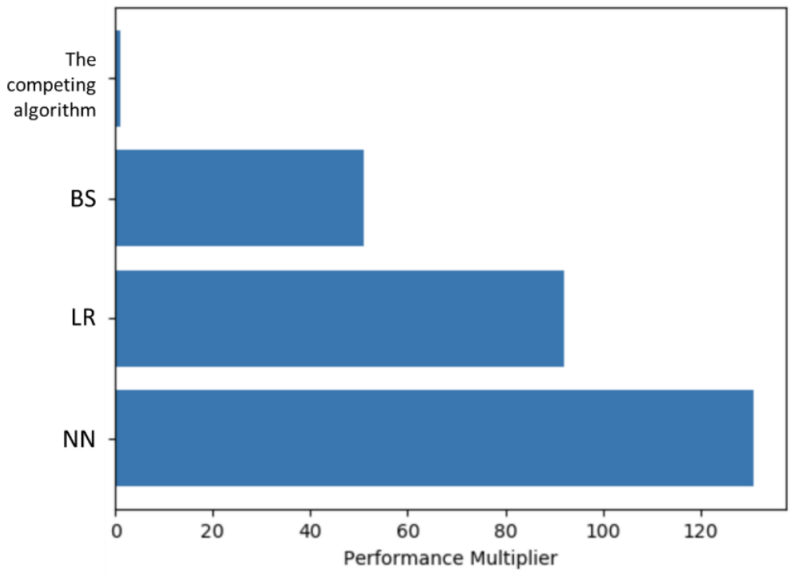


**Fig. 22 The comparison of performance in terms of the performance multiplier.**

| Material Properties | Raw Stress Values | Stress Magnitude | Interaction Effects |
|---|---|---|---|
| $Y_T$ | $\sigma_{22}$ | $\sigma_{22}^2$ | $\sigma_{22} * \sigma_{33}$ |
| $Y_C$ | $\sigma_{33}$ | $\sigma_{33}^2$ | $\sigma_{22} * \tau_{12}$ |
| $S_L$ | $\tau_{13}$ | $\tau_{13}^2$ | $\sigma_{22} * \tau_{13}$ |
| | $\tau_{23}$ | $\tau_{23}^2$ | $\sigma_{22} * \tau_{23}$ |
| | $\tau_{12}$ | $\tau_{12}^2$ | $\sigma_{33} * \tau_{12}$ |
| | | | $\sigma_{33} * \tau_{13}$ |
| | | | $\sigma_{33} * \tau_{23}$ |
| | | | $\tau_{12} * \tau_{13}$ |
| | | | $\tau_{12} * \tau_{23}$ |
| | | | $\tau_{13} * \tau_{23}$ |

The purpose of categorizing the input variables is to improve the explainability of the model and better represent the underlying physical properties these variables are measuring. It would not make sense to remove $\sigma_{22}$ from the model while including $\sigma_{33}$ because while they are two distinct measurements, they are measuring the same type of stress. The neural network was initially trained with all available features in the training data and the permutation importance values are shown in figure 23. The raw stress values seem to cluster near the bottom of the importance rankings, and so the model was retrained without including the raw stress values. The permutation importance values from the refined model are shown in figure 24. The refined model produces substantially fewer features with a permutation importance score of less than -0.2 and the $R^2$ value for the crack angle changed from 94.9% to 94.6% when those features were removed. This is an indication that the features removed were not contributing to the results substantially. Additionally, while the crack angle $R^2$ value changed negligibly, the failure envelope appeared much smoother with the refined features which is likely a result of less overfitting due to the use of fewer inputs.

**B. Crack Angle**

Visualizing the crack angle predictions is difficult due to the number of dimensions in a full stress state. For this reason, $R^2$ is used as the method for evaluation. The $R^2$ value is generated by comparing the predicted crack angle with the crack angle generated from the baseline. The neural network that was created produced an $R^2$ value of 94.7%, which can be interpreted as saying that 94.7% of the variance in crack angles is captured by the model.

| Weight | Feature |
|---|---|
| -0.0221 ± 0.0000 | tau12*tau23 |
| -0.0233 ± 0.0000 | tau13*sig22 |
| -0.0236 ± 0.0000 | tau12*tau13 |
| -0.0246 ± 0.0000 | tau23*sig22 |
| -0.0257 ± 0.0000 | tau23*tau13 |
| -0.0287 ± 0.0000 | sl |
| -0.0339 ± 0.0000 | tau13*sig33 |
| -0.0407 ± 0.0000 | tau23*sig33 |
| -0.0411 ± 0.0000 | tau12*sig22 |
| -0.0476 ± 0.0000 | tau12*sig33 |
| -0.0488 ± 0.0000 | sig33^2 |
| -0.0606 ± 0.0000 | sig22^2 |
| -0.0624 ± 0.0000 | yc |
| -0.0742 ± 0.0000 | tau12^2 |
| -0.1105 ± 0.0000 | tau12 |
| -0.1109 ± 0.0000 | yt |
| -0.1487 ± 0.0000 | sig22*sig33 |
| -0.1706 ± 0.0000 | tau13 |
| -0.2051 ± 0.0000 | sig33 |
| -0.2116 ± 0.0000 | sig22 |
| -0.2155 ± 0.0000 | tau13^2 |
| -0.2660 ± 0.0000 | tau23^2 |
| -0.3727 ± 0.0000 | tau23 |

**Fig. 23 Permutation Importance scores for all possible input features.**

| Weight | Feature |
|---|---|
| -0.0316 ± 0.0000 | sl |
| -0.0535 ± 0.0000 | tau13*sig22 |
| -0.0622 ± 0.0000 | yc |
| -0.0634 ± 0.0000 | tau23*sig22 |
| -0.0671 ± 0.0000 | tau23*tau13 |
| -0.0765 ± 0.0000 | tau13*sig33 |
| -0.0955 ± 0.0000 | tau12*sig22 |
| -0.0966 ± 0.0000 | yt |
| -0.1205 ± 0.0000 | tau12*tau13 |
| -0.1322 ± 0.0000 | tau12*sig33 |
| -0.1328 ± 0.0000 | tau23*sig33 |
| -0.1400 ± 0.0000 | tau12^2 |
| -0.1522 ± 0.0000 | sig22^2 |
| -0.1602 ± 0.0000 | sig33^2 |
| -0.1779 ± 0.0000 | sig22*sig33 |
| -0.1808 ± 0.0000 | tau13^2 |
| -0.2526 ± 0.0000 | tau23^2 |
| -0.2599 ± 0.0000 | tau12*tau23 |

**Fig. 24 Permutation Importance scores for refined features.**

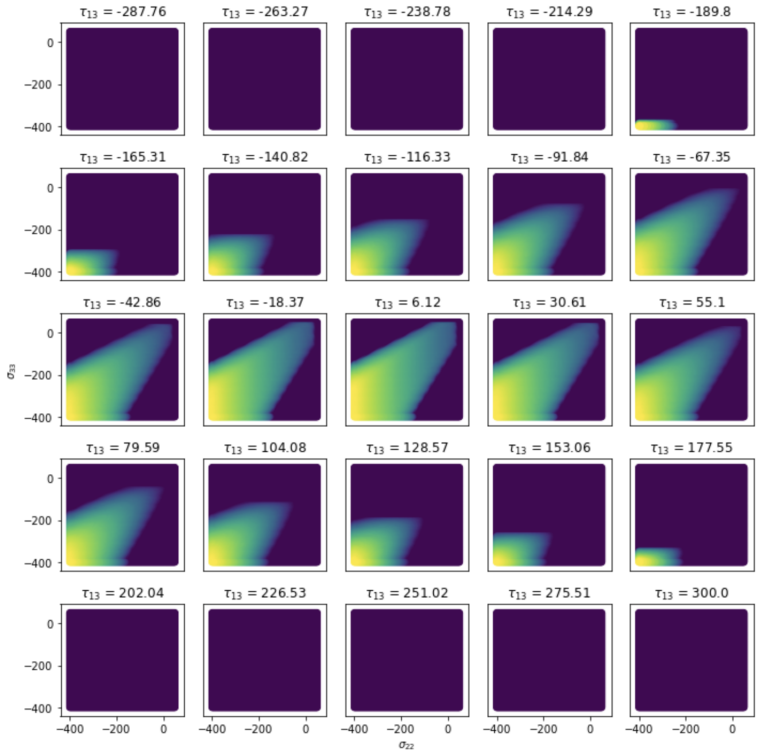


**Fig. 25 Failure Envelope generated by the competing algorithm for various $\tau_{13}$ values.**

## C. Failure Envelope

### *1. Visualization*

The failure envelope for a full stress state is more difficult to visualize than for plane stress due to the added dimensions and more figures are needed to effectively evaluate the results. Each generated failure envelope is presented as a sequence of plots. Each individual plot shows a surface with $\sigma_{22}$ on the x-axis, $\sigma_{33}$ on the y-axis, and the color indicating the corresponding value of $\tau_{12}$ at each point. Each collection of plots shows how that failure envelope changes as other stress values change. For example, figure 25 and figure 26 are generated using the same Competing Algorithm results. Figure 25 shows the failure envelope across a certain range of $\tau_{13}$ values with $\tau_{23}$ held constant at zero while figure 26 shows the opposite - varied $\tau_{23}$ values with $\tau_{13}$ held constant at zero.

### *2. Evaluation*

The Competing Algorithm is used as a baseline for the full stress state. This is different from the plane stress results because the BS algorithm is not applicable for the full stress state. Figure 25 and figure 26 show the baseline failure envelope and figure 27 and figure 28 show the failure envelopes predicted by the neural network. The neural network is able to clearly capture the shape of the failure envelopes across the different input dimensions with only slight variations. In terms of

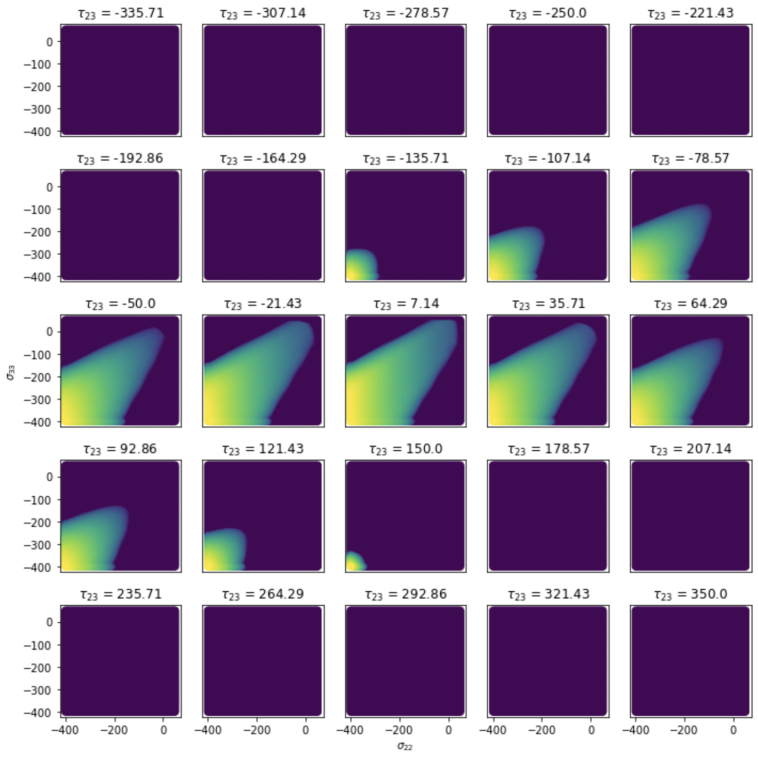


**Fig. 26 Failure Envelope generated by the competing algorithm for various $\tau_{23}$ values.**

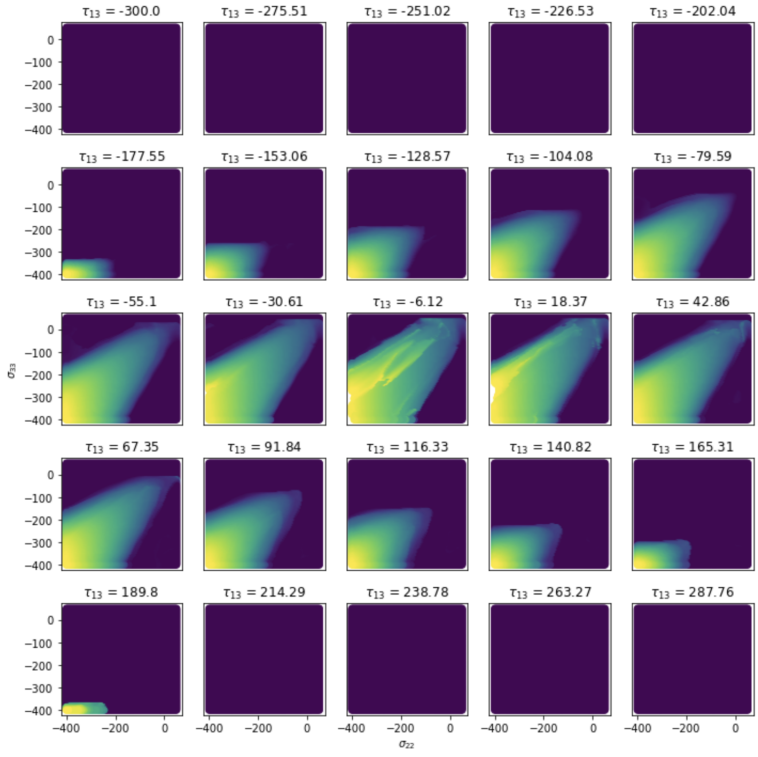


**Fig. 27 Failure Envelope generated by neural network for various $\tau_{13}$ values.**

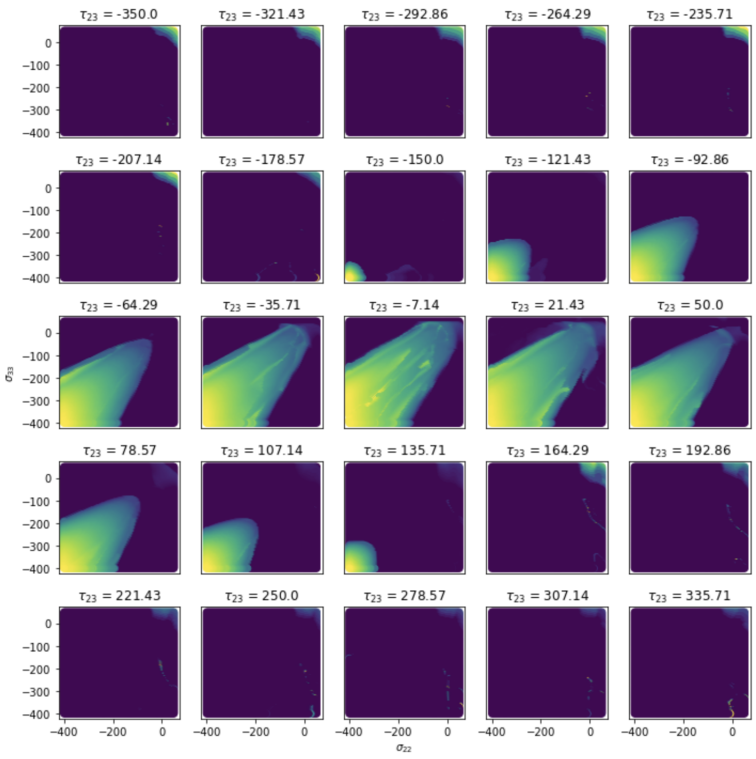


**Fig. 28 Failure Envelope generated by neural network for various $\tau_{23}$ values.**

computational efficiency, the neural network ran approximately eight times faster than the competing algorithm, as shown in figure 29.

## VIII. Conclusion

The Mohr-Coulomb and Hashin models provide high-fidelity results for damage initiation, but due to the iterative nature of current implementations, they can have a high computational cost. For 2D stress states, improvement in computational efficiency can be achieved in multiple ways. By replacing the competing algorithm with Binary Search, run time can be reduced dramatically while still guaranteeing the correct fracture angle is found as defined by the MC failure criterion. Using linear regression as a surrogate model it is difficult to accurately predict the fracture angle but using a more complex neural network model, 99.9% prediction accuracy is found while reducing runtime by as much as 99%.

For 3D stress states, distributions of failure angles are less predictable and so more efficient algorithms are not applicable. Similarly, linear regression was unable to successfully predict failure angles with a reasonable accuracy. Surrogate models must be more complex to capture the unpredictable patterns in the data - the neural network architecture contained many more layers and nodes. With the more complex architecture, a neural network was able to produce 94.7% prediction

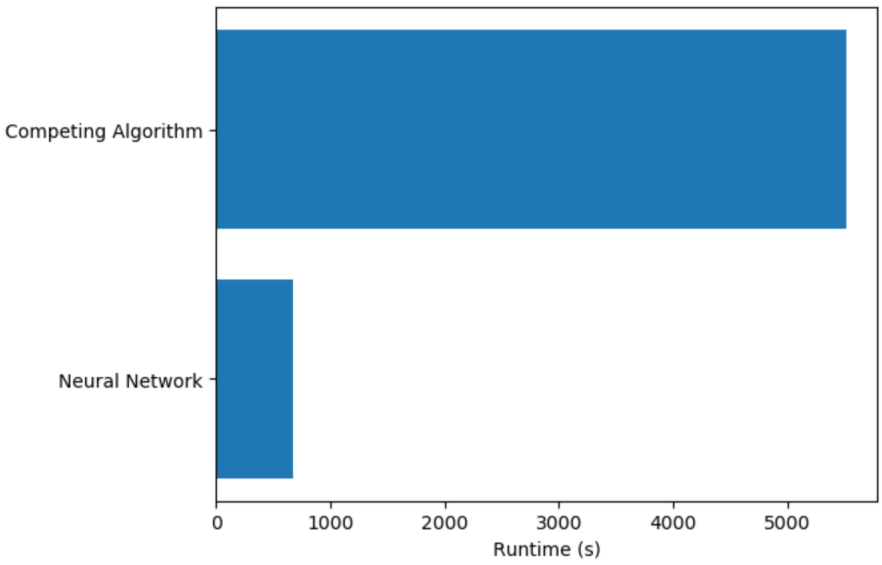


**Fig. 29 Performance comparison between Neural Network and Competing Algorithm**

accuracy with respect to failure angles and substantially faster run times compared to the competing algorithm. Since the machine learning predictive models include material properties and stress components as input parameters, the trained models will be applicable to the damage initiation determination of any composite material system with any load case. Using these surrogate models can allow engineers to model more complex problems, improve the rate of experimental iteration, or simply reduce computational cost without sacrificing any accuracy.

## Acknowledgments

I gratefully acknowledge the support and advice received from Professor Ilyas Ustun, who has advised me for the past year and without his encouragement this thesis would never have been written. Additionally, Professor Anthony M. Waas, Felix Pawlowski Collegiate Professor, and Shiyao Lin, PhD student, from the University of Michigan Aerospace Engineering department have provided significant domain knowledge and expertise in the creation of this thesis.

## References

[1] Hashin, Z., "Failure criteria for unidirectional fiber composites," 1980.

[2] Pineda, E. J., and Waas, A. M., "Numerical implementation of a multiple-ISV thermodynamically-based work potential theory for modeling progressive damage and failure in fiber-reinforced laminates," *International journal of fracture*, Vol. 182, No. 1, 2013, pp. 93–122.

[3] Puck, A., and Schürmann, H., "Failure analysis of FRP laminates by means of physically based phenomenological models," *Failure criteria in fibre-reinforced-polymer composites*, Elsevier, 2004, pp. 832–876.

[4] Hinton, M., Kaddour, A., and Soden, P., "A comparison of the predictive capabilities of current failure theories for composite laminates, judged against experimental evidence," *Composites Science and Technology*, Vol. 62, No. 12-13, 2002, pp. 1725–1797.

[5] Pinho, S. T., Dávila, C. G., Camanho, P. P., Iannucci, L., and Robinson, P., "Failure Models and Criteria for FRP Under-In-Plane or Three-Dimensional Stress States Including Shear Non-Linearity," 2005.

[6] Dávila, C., and Camanho, P., "Failure Criteria for FRP Laminates in Plane Stress," 2003.

[7] Pinho, S., Iannucci, L., and Robinson, P., "Physically-based failure models and criteria for laminated fibre-reinforced composites with emphasis on fibre kinking: Part I: Development," *Composites Part A: Applied Science and Manufacturing*, Vol. 37, No. 1, 2006, pp. 63–73.

[8] Patel, D. K., Waas, A. M., and Yen, C.-F., "Compressive response of hybrid 3D woven textile composites (H3DWTCs): An experimentally validated computational model," *Journal of the Mechanics and Physics of Solids*, Vol. 122, 2019, pp. 381–405.

[9] Jones, R. M., *MacroMechanical Behavior of a Lamina*, CRC Press, 2018.

[10] Reddy, J. N., *Introduction to the finite element method*, McGraw-Hill Education, 2019.

[11] Ferreira, A. J. M., *MATLAB codes for finite element analysis: solids and structures*, Springer Science and Business Media, 2009.

[12] Chen, C.-T., and Gu, G. X., "Machine learning for composite materials," *MRS Communications*, Vol. 9, No. 2, 2019, pp. 556–566.

[13] Gu, G. X., Chen, C.-T., and Buehler, M. J., "De novo composite design based on machine learning algorithm," *Extreme Mechanics Letters*, Vol. 18, 2018, pp. 19–28.

[14] Tiryaki, S., and Aydın, A., "An artificial neural network model for predicting compression strength of heat treated woods and comparison with a multiple linear regression model," *Construction and Building Materials*, Vol. 62, 2014, pp. 102–108.

[15] Yang, Z., Yabansu, Y. C., Al-Bahrani, R., Liao, W.-k., Choudhary, A. N., Kalidindi, S. R., and Agrawal, A., "Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets," *Computational Materials Science*, Vol. 151, 2018, pp. 278–287.

[16] Davidson, P., and Waas, A. M., "Probabilistic defect analysis of fiber reinforced composites using kriging and support vector machine based surrogates," *Composite Structures*, Vol. 195, 2018, pp. 186–198.

[17] Hinton, M., and Soden, P., "Predicting failure in composite laminates: the background to the exercise," *Composites Science and Technology*, Vol. 58, No. 7, 1998, pp. 1001–1010.

[18] Knuth, D. E., "Big omicron and big omega and big theta," *ACM Sigact News*, Vol. 8, No. 2, 1976, pp. 18–24.

[19] Breiman, L., "Random forests," *Machine learning*, Vol. 45, No. 1, 2001, pp. 5–32.