DIFFERENCE-BASED DEEP LEARNING FRAMEWORK FOR STRESS PREDICTIONS IN HETEROGENEOUS MEDIA

Haotian Feng  
Doctoral Student  
Dept. of Civil & Environmental Engineering  
University of Wisconsin-Madison  
Madison, WI 53706  
hfeng47@wisc.edu

Pavana Prabhakar  
Assistant Professor & Salmon Fellow  
Dept. of Civil & Environmental Engineering  
Dept. of Engineering Physics  
Dept. of Materials Science and Engineering  
University of Wisconsin-Madison  
Madison, WI 53706  
pavana.prabhakar@wisc.edu

ABSTRACT
Stress analysis of heterogeneous media, like composite materials, using finite element method (FEM) has become commonplace in design and analysis. However, calculating stresses and determining stress distributions in heterogeneous media using FEM can be computationally expensive in situations like optimization and multi-scaling, where several design iterations are required to be tested iteratively until convergence. In this paper, we utilize deep learning and develop a set of Difference-based Neural Network (DNN) frameworks based on engineering and statistics knowledge to determine stress distribution in heterogeneous media with special focus on discontinuous domains that manifest high stress concentrations. To evaluate the performance of DNN frameworks, we consider four different types of geometric models that are commonly used in the analysis of composite materials: plate with circular cutout, square packed fiber reinforced, hexagonal packed fiber reinforced and hollow particle reinforced models. The proposed DNN structure consists of a normalization module (DNN-N) for all geometries considered, while we additionally introduce a clean module with DNN-N, named DNN-NC, for geometries with discontinuities. Results show that the DNN structures (DNN-N and DNN-NC) significantly enhance the accuracy of stress prediction compared to existing structures for all four models considered, especially when localized high stress concentrations are present in the geometric models.

Keywords  Machine Learning · Stress Prediction · Reinforced Composites · Finite Element Analysis · Micromechanics

1 INTRODUCTION
Stress analysis is an important discipline within engineering, where the primary objective is to determine stresses and strains in structures and materials subjected to external loads. It is primarily used for analyzing how structures and materials respond to applied loads and for designing structures and materials that can withstand anticipated loads, and investigate failures of the same. Stress analysis is primarily used in mechanical, civil and aerospace disciplines for designing structures from bridges to aircraft and materials ranging multiple length scales from micro to macro scales. Within stress analysis, we typically start with a geometrical description of a structure or material and the expected
load acting on it. Typical output of stress analysis is the quantitative distribution of stresses, strains and deformations. There are several approaches for stress analysis of solids, like classical mathematical closed form solutions for partial differential equations, computational simulation, experimental testing or a combination of these methods. Among these, Finite Element Method (FEM)\cite{1, 2} is a commonly used computational tool for stress analysis, and is used extensively for designing of structures and materials. By reformulating governing partial differential equations (PDE) from strong form to weak form, and implementing these in discrete form within FEM, the response of solids subjected to external loads and boundary conditions can be determined. Several FEM software (both commercial and open-source) are popular and widely used in academia and industry, including ABAQUS\cite{3}, NASTRAN, ANSYS, LS-DYNA, FEniCS and Deal-II. These software are capable of generating reliable stress distribution outputs for a given model with loads and boundary conditions.

FEM is used extensively for analyzing composite materials\cite{4}, which are heterogeneous and usually made of individual constituent materials with unique properties that are combined together to result in improved physical properties as compared to the individual materials. Composite materials typically consist of matrix and reinforcing constituent materials, where the reinforcing material is stiff and strong, and matrix is made of homogeneous and monolithic material that binds the reinforcements together. Typically, several length scales exist within composites, with reinforcements at the micrometer scale and composites at the meter scale. Detailed FEM analyses at multiple length scales are widely used to analyze and design these composites\cite{5, 6, 7}.

Although commonly used for stress analysis, FEM can be expensive when used for optimization and multi-scale analysis of structures and materials. Hence, the past few years have witnessed a few attempts for substituting traditional FEM with Neural Network (NN), a method within Machine Learning (ML) framework, for structural optimization\cite{8, 9} and multi-scale analysis\cite{10, 11}. ML is a branch of artificial intelligence that allows computers to learn from past experience and detect complex behaviors from either large, noisy or incomplete data sets using a variety of statistical, probabilistic, and optimization techniques\cite{12, 13, 14}. Within ML, algorithms are capable of identifying patterns from complex data sets by a training process, and then use the trained algorithm to predict unknown data. ML has been used for a wide range of applications in recent years, including but not limited to spam detection\cite{15} and pattern recognition\cite{16}. In each of these ML algorithms, there are no hard-coded rules for classifying data into certain categories, instead, these algorithms learn from a given data set and extract knowledge to classify newer data sets available.

Past researches have shown great potential for ML and Deep Learning (DL) method as a surrogate for predicting mechanical properties within Computational Solid Mechanics and Computational Fluid Dynamics (CFD), without having to perform Finite Element Analysis (FEA). Liang et al.\cite{17} first introduced machine learning model into stress distribution estimation in human tissues and proved the feasibility of establishing the linkages between shape features and FEA-predicted results with machine learning approach. Gao et al.\cite{18} proposed Encoder-Decoder structure in CFD to predict non-uniform steady laminar flow in vehicle aerodynamic analysis and showed that it is considerably faster than traditional Lattice Boltzmann methods solvers. Bhatnagar et al.\cite{19} further improved encoder-decoder structure’s capability for predicting velocity and pressure field in unseen flow conditions and geometries. Nie et al.\cite{20} first introduced Stress-Net by implementing Residual Network (ResNet)\cite{21} and Convolutional Neural Network (CNN)\cite{22} into Encoder-Decoder structure to predict von Mises stress distribution. They validated their framework using 2D linear elastic homogeneous cantilevered geometric structures. These past researchers have focused on developing ML approaches for simple structures with homogeneous material. However, the heterogeneity introduced by multiple phases in reinforced composites necessitates a revolutionary method for predicting stress distributions accurately. Thus, in this paper, we have presented a novel Neural Network that is capable of predicting stress distributions in heterogeneous materials.

Inspired by the ability of CNN to extract high-level features\cite{22} and several previous successful Encoder-Decoder models\cite{23}, we introduce a new Difference-based Neural Network (DNN) structure in this paper that focuses on geometric and associated stress differences between different samples for stress prediction in heterogeneous materials. Instead of directly using several Finite Element model geometries and stresses as input, DNN focuses on highlighting the differences in stress distribution between different input samples by using the geometry and stress differences between training samples and a reference model (determined based on training samples) as input data for training the NN. Highlighting these differences in the input sample data is expected to improve the accuracy of prediction. Specifically, we demonstrate the capabilities of DNN using 2D hollow structures and reinforced composite materials. This is the first attempt, to our best knowledge, towards predicting stress contour distributions in heterogeneous media like composite materials that possess severe stress concentrations.
Figure 1: An overview of the proposed ML framework: Composite models are first generated and solved using Finite Element solver to obtain stress distribution contours. The meshed geometry and stress contours are interpolated onto Cartesian Maps and further used for training the DNN structure. Finally, the trained DNN is used for predicting stress distribution contours for new composite models.

2 Overview of the Proposed Machine Learning Framework

In Figure 1, we present an overview of the proposed machine learning framework for composite structures and materials. Throughout the paper, we have used Intel Core™ i7-9700 Processor with 4.70 GHz maximum frequency for performing FEA and post-processing. We performed the machine learning step on NVIDIA GeForce RTX 2080 SUPER with 3072 CUDA cores with 1815 MHz frequency.

First, several target composite geometries are randomly generated and meshed, and are solved numerically under quasi-static loading using Finite Element software to obtain the spatial stress distribution for each geometry. We consider linear elasticity in this study, where we solve steady-state or stationary problems described below within solid mechanics. We consider a two-dimensional domain $\Omega \subseteq \mathbb{R}^2$, and the boundary value problem (BVP) solved is as follows:

$$\nabla \cdot \sigma + f = 0 \quad \text{in} \quad \Omega$$
$$u = \bar{u} \quad \text{on} \quad \partial \Omega_u$$
$$\sigma n = \bar{\sigma} n \quad \text{on} \quad \partial \Omega_n$$

(1)

Here, $\sigma$ is the stress tensor, which is a function of displacement $u$ related by the linear elastic constitutive equation, $\sigma = C \epsilon$ and $\epsilon = \frac{1}{2}[\nabla u + (\nabla u)^T]$. This BVP is numerically solved within FEM to determine stress distribution in the domain. This is a very well established method for solving linear elastic BVPs within solid mechanics. A brief overview of the finite element method is provided in Appendix A. Next, the meshed geometry and nodal stress distribution contours are interpolated onto a uniform global map called Cartesian Map (CM)[19] shown in Figure C.1. This makes the interpolated geometry and the stress models to be of the same size, making it easier for training of the Neural Network and learning the parameters through the training steps. Upon training the CNN structure, the learned model is used for predicting the spatial stress distribution contours and evaluating its performance.

3 Model definition

3.1 Introduction to Different Geometric Models Considered

We considered four different structures shown in Figure 2 as the geometric models in this study: 1) Plate with a circular cutout model, 2) square packed fiber reinforced model, 3) hexagonal packed fiber reinforced model, 4) hollow particle reinforced model. Overall external dimensions of each model is 10 x 10 $\mu$m.

- We considered the plate with a circular cutout geometric model for developing the DNN framework.
Figure 2: Meshed geometry (top row) and corresponding stress distribution contours (bottom row) of models considered: (a,e) plate with circular cutout (b,f) square packed fiber reinforced composite (c,g) hexagonal packed fiber reinforced composite (d,h) hollow particle reinforced composite. Blue and pink regions in the geometry models are different materials and the white region are hollow.

- Next, we considered the square and hexagonal packed fiber reinforced composite models to verify the efficacy and efficiency of the DNN framework in view of micro-mechanical analysis of heterogeneous media, which here are fiber reinforced composites.
- Finally, we demonstrated that the DNN framework we have proposed can accurately predict stress distributions in composite material models with large stress concentration, which in this paper is hollow particle reinforced composites.

3.2 Importance and Relevance of Each Model Considered

We considered two dimensional plane stress analysis in this paper. We focused on predicting von Mises stresses as both matrix and reinforcement regions have in-plane isotropic properties. We varied the geometry by choosing different fiber or cutout radius \( r \), which is calculated based on different volume fractions \( V_f \) using the equation \( r = \sqrt{\frac{A_{\text{square}} \times V_f}{\pi}} \). \( V_f \) describes the volume percentage of one part to the whole model. We randomly generated different values of \( V_f \) within a target range, where \( A_{\text{square}} \) is the area of the bounding square region.

3.2.1 Plate with circular cutout

Plate with cutout designs are widely used in mechanical industries, for example airplane cabin window and screw-bolt designs. Under externally applied loads, plate with a cutout experiences high stress concentrations in the vicinity of the cutout. These high stress regions are candidate for localized damage and failure under external loads, and largely affect the mechanical performance of structures. We used the plate with a circular cutout model for establishing the DNN structure and validating the accuracy of local stress concentration prediction for isotropic materials. The cutout diameter is usually relatively small, and hence we assume that the cutout region has a volume fraction range between 5% to 25%. A sample meshed plate model with a circular cutout is shown in Figure 2(a).

3.2.2 Fiber reinforced polymer composites - Square and Hexagonal Packing

Fiber reinforced polymer composite (FRPC) materials are widely used in aerospace, automotive, marine and construction industries due to its higher strength comparing to pure polymer matrix. FRPCs typically consist of two parts: stiff reinforcing fibers and a less stiff binding matrix. In this paper, we choose FRPCs as a model system. One key design feature in FRPCs is the fiber volume fraction \( V_f \). Changing the values of \( V_f \) influences the resulting stress distribution...
within FRPCs, which contributes directly to their mechanical properties. Fibers within FRPCs typically have diameters in the range of few micro meters, for example, carbon fibers are approximately 6 \( \mu m \) in diameter. Since a structure made of composite is in the order of few meters, representing each fiber in a computational domain is not practical. Often, we resolve to micromechanics for analyzing such composites.

In micromechanical analysis of composites, a large composite domain can be represented by arrays of small repeating unit cells (RUC). These RUCs can be square packed or hexagonal packed models of fibers in matrix with fibers having its actual diameter and the entire fiber-matrix domain has a fiber volume fraction equal to that of the macro scale composite domain. Thus, the micromechanics models are in the order of micrometers while effectively capturing the mechanical behavior of large composites. Square-packed RUC of FRPCs has fiber in the center and a square-shaped matrix surrounding it as shown in Figure 2(b), while hexagonal packed RUC has a full fiber in the center and four quarter fibers in each corner of the square matrix domain as shown in Figure 2(c). For validation purposes, we simply assume hexagonal packed composite to be a square shaped domain as in the case of a square fiber packed model. Fiber volume fraction of 40% to 60% are most commonly considered in real FRPC materials. Hence for this paper, we consider the same volume fraction range to generate random RUC models.

3.2.3 Hollow Particle Reinforced Composite

Hollow particle reinforced composite materials, also referred to as syntactic foams, are gaining traction in lightweight applications due to their low density, high compressive energy absorption capability and large strains to failure[24, 25, 26, 27]. Syntactic foams typically consist of stiff hollow particles (often at the micro-scale) randomly dispersed in a softer matrix region, which results in lightweight closed cell foams. A unit cell representation of hollow particle reinforced composite material consists of a square block with circular cutout and a thin reinforced layer at inner surface of the cutout. The reinforced layer is typically made of stiffer materials, like glass, carbon or ceramics. Compared to the plate with circular cutout model, hollow particle reinforced composite models manifest more significant stresses and concentrations due to stiffer material and thinner thickness of the reinforcement. In this paper, we consider a similar structure as that of plate with circular cutout model, but with a thin circular ring inside the cutout region as shown in Figure 2(d). We choose the wall thickness of the reinforcing ring to be approximately 1/22.5 of the inner diameter (hole diameter) of reinforced hollow particle based on prior experimental research performed by Jayavardhan et al. [27]. In this paper, we consider the volume fraction of the cutout region to be in the range from 40% to 60% with a ring thickness of 0.18 \( \mu m \).

3.3 Finite Element Analyses for Training Data Generation

All the domains mentioned above are 2D plane stress models with the same external dimensions. Linear elastic mechanical properties of polymer matrix (\( E = 3.2 \) GPa, \( \nu = 0.31 \)) are assigned to the blue regions in each model, which are typical of epoxy resin used in fiber reinforced polymer matrix composites. For reinforced composites, linear elastic carbon fiber properties (\( E = 8.0 \) GPa, \( \nu = 0.35 \)) in the plane perpendicular to the fiber direction are assigned to the pink regions. The external boundaries of each domain are defined as \( \Gamma_1 \), \( \Gamma_2 \), \( \Gamma_3 \) and \( \Gamma_4 \), respectively, for the top, left, bottom and right edges. The boundary conditions on each boundary is described below in terms of horizontal (\( u \)) and vertical (\( v \)) displacements as Equation 2. Essentially, each model is subjected to a positive displacement along the vertical direction subjecting them to tension.

\[
\begin{align*}
    v &= \text{0.1} \mu m & \text{on } \Gamma_1 \\
    u &= 0 & \text{on } \Gamma_2 \\
    v &= 0 & \text{on } \Gamma_3 \\
    u &= \text{constant} & \text{on } \Gamma_4 \\
\end{align*}
\] 

(2)

Using the above mentioned inputs to each domain, we performed mesh convergence analysis to determine the maximum (max) mesh size that provides converged stress predictions. A max mesh size of 0.2 \( \mu m \) for particle reinforced composite model and a max mesh size of 0.3 \( \mu m \) for the other three models was determined and used. After meshing the geometric models with the mesh sizes determined above and assigning material properties along with above-mentioned boundary conditions, we generated stress distribution contours using static solver within FEM in a commercially available software, ABAQUS. Sample results from stress analysis of the four models are shown in Figure 2(e)-(h).
4 Difference-based Neural Network Framework

Within CNN, an Encoder-Decoder Neural Network\[33\] is typically trained using geometry and stress contours directly as inputs. Past researchers have demonstrated that residual learning can significantly improve the accuracy of prediction. Nie et al.\[20\] proposed the Stress-Net structure based on residual learning algorithm, which was shown to improve the accuracy of stress distribution prediction. However, the existing Stress-Net structure does not provide high prediction accuracy when localized high stresses exist, like stress concentration, especially within composite materials. In this paper, a novel Neural Network structure is developed that embeds engineering and statistical knowledge for stress prediction. During engineering design iterations, engineers typically use an initial design as the reference model and then refine the subsequent designs based on that reference model. Similar to this idea, a known geometry model and corresponding stress distribution contour are chosen as the reference model when training the target Neural Network. Then, the differences between different geometry contours $G_i$ are used for directing the Neural Network to focus on the differences and train the stress difference contours $\sigma_i$. Here, $i = 1,..., \text{total number of training samples}$. We refer to this as a Difference-based Neural Network (DNN) structure and is shown in Figure 3. The DNN structure consists of three modules: sample processing, Encoder-Decoder and stress prediction. Difference-based Neural Network with normalization (DNN-N) is built upon DNN by adding additional normalization and de-normalization blocks (orange color). Further, Difference-based Neural Network with normalization and clean module (DNN-NC) is developed based on DNN-N structure by adding two additional clean modules (blue color). An example of how DNN predicts the stress distribution contours on a square packed composite model is shown in Figure C.3.

4.1 Sample processing module

The sample processing module mainly extracts information from geometry and stress contours from the training data. Mean geometry and stress contours across all training models are picked as the reference sample for training the Neural Network (labelled as Ref Geometry and Ref Stress). The geometry differences, stress differences and mean stress distribution contours are used next for training the Neural Network. To avoid covariate shifting and to improve training efficiency, a normalization module is added to the geometry difference contour set and a denormalization module is added after the DeConv Block within the final prediction module (defined in section 4.3). Normalization block is developed based on Min-Max Feature scaling functions\[34\], described in Equation 3, where $G$ refers to the labelled
geometry difference contour. During the Neural Network training stage, normalization block ensures that all inputs are bounded within similar ranges in order to enhance prediction accuracy and efficiency.

\[
\text{Normalization: output} = \frac{\text{input} - \min(G)}{\max(G) - \min(G)}
\]  

(3)

### 4.2 Encoder-Decoder module

Encoder-Decoder module consists of three types of blocks: Conv-SE (yellow), ResNet-SE (purple) and DeConv block (red), as shown in Figure 3. Each Conv-SE block, as shown in Figure 4(a), consists of one 2D convolutional layer with ReLU and Batch-normalization, and one Squeeze-and-Excitation (SE) block [35]. The convolutional layer aims at extracting high level key features of geometry contour and the SE block adaptively re-calibrates channel-wise feature responses by modelling inter-dependencies between different channels. Following the Conv-SE blocks are the ResNet-SE blocks, where each ResNet-SE block is constructed based on ResNet architecture and consists of two convolutional blocks and one SE block to enhance the extracted inter-dependent high level features, as shown in Figure 4(b). After ResNet-SE blocks, the result is further passed into three DeConv blocks, with each of them consisting of one 2D deconvolutional layer [36] with Batch-normalization, as shown in Figure 4(c). The DeConv block expands key features and finally back to the original input dimensions. Since the differences between the original and mean contours generally have a zero mean and certain variations, we assume that it follows a Gaussian distribution, and hence use Glorot initialization for weight initialization[37]. Stochastic gradient descent (SGD)[38] is used as an optimizer with learning rate set as 0.001. Loss function is defined based on mean squared error (MSE). Training epoch number and steps are chosen when prediction accuracy is converged. In this paper, we choose epoch number to be 80 and 40 steps in each epoch.

### 4.3 Stress prediction module

Stress prediction module shown in Figure 3 follows the Encoder-Decoder Neural Network module. De-normalization block is the first block within this module, which reverts the effect of normalization as defined in Equation 4. Here, \( \sigma \) represents the stress difference contour. A stress processing block is added after the De-normalization block, which generates the final stress prediction. Details of this stress processing block is shown Figure 4(d). This module consists of one dense block for linearly combining the predicted stress differences (input) and the reference stress contour, followed by one 2D De-Convolution block to smooth the prediction before presenting the output. The kernel size of the De-convolution layer is selected based on the uniformity in stress distributions. A kernel size of [2,2] is used for the De-Convolution block if large stress concentrations occur (DNN-NC) and a kernel size of [1,1] is used otherwise (DNN-N). Here, we consider that large stress concentration exists if the stress ratio defined as \( R_\sigma = \sigma_{\text{max}}/\sigma_{\text{mean}} \) is larger than 2. Adding a [2,2] kernel size would consider more information about nearby pixels compared to linear element-wise multiplication. Hence, using [2,2] kernel size is beneficial in models with large stress concentrations as the differences between pixel values are higher.

\[
\text{Denormalization: output} = \text{input} \ast (\max(\sigma) - \min(\sigma)) + \min(\sigma)
\]  

(4)

### 4.4 Clean module (Only for geometry with cutout region)

For geometric models that have regions of no material, like in the case of plate with circular cutout model and hollow particle reinforced model, we need to add another step to DNN-N, which gives the DNN-NC structure. Varying the size of the cutout region can introduce undesired negative values during subtraction with the reference contours, which can influence the accuracy of the Neural Network prediction (refer to section 5.2). Hence, we introduce an additional module called “clean module” for these two models specifically. This module performs element-wise multiplication between the target contour and material contour \( M \), whose regions with material are labelled as ‘1’ and regions without material (cutout) as ‘0’. The multiplication can be represented with Hadamard product[39] shown in Equation 5, manually forcing regions without material to be zero valued. Here, \( A_{ij} \) represents the input contour and \( C_{ij} \) represents the output contour after passing through the clean module.

\[
C_{ij} = (A \circ M)_{ij} = A_{ij}M_{ij}
\]  

(5)

\[
M_{ij} = \begin{cases} 
1, & \text{if material exists at a node} \\
0, & \text{if material is absent at a node}
\end{cases}
\]
4.5 DNN-N versus DNN-NC

The key difference between DNN-N and DNN-NC is the addition of a clean module within DNN-NC. Specifically, DNN-N structure can be viewed as a special case of DNN-NC, where the material contour $M$ in Equation 5 is matrix with all-ones as no cutout region exists. A Hadamard product with all-ones matrix will output the exact same matrix as the input. Besides, to account for the presence of a clean module, we choose difference kernel sizes, as discussed in section 4.3.
5 Interpolation on Cartesian Map

5.1 Data Pre-processing

Meshed geometry of the models and corresponding stress contours obtained from FEM analysis are used for training and evaluating different Neural Network frameworks. Each training sample can have different shape due to different fiber volume fractions or different cutout diameters considered, which can result in different meshes in the domain. This poses difficulty for the Neural Network to learn. In order to render the geometry as well as the stress contour trainable for Neural Network, these contours are further interpolated onto a global ‘map’ called the Cartesian Map, such that all contours have the same size. Inspired by Bhatnagar’s [19] idea of creating a uniform map, both meshed geometries and stress contours are interpolated onto a Cartesian Map that has the same size as that of the geometric models, which are 10 μm-by-10 μm.

Triangulation-based linear interpolation[40] is widely used for map-to-map interpolation due to its simplicity and efficiency. This algorithm searches for three nearest nodes on the Cartesian mesh to be mapped on for each node in the original FEM mesh to form a triangulation. This can introduce large artificial errors due to matrix illness in FEM meshed models when stress concentrations are present that may require non-uniform mesh with smaller mesh size. To avoid this in the DNN framework, we use Barycentric coordinate system[41], also known as area coordinates, which normalizes each axis and generates homogeneous coordinates. Due to this characteristic, Barycentric coordinates are extremely useful in making the interpolation more stable in triangular sub-domains. Barycentric coordinates can help in accurately determining nodal location with respect to triangular mesh, as well as interpolation coefficient with three vertices. Each node in the Cartesian Map is projected onto the FEM mesh model to find the triangle it falls within. Then, the geometry label and stress value of each node on the Cartesian Map are further interpolated using the corresponding values of the three vertices of that triangle. Relative positions between a node and a triangle are determined by values of the three Barycentric parameters $\lambda_1$, $\lambda_2$, $\lambda_3$, which are determined using the equations shown below. Here, $\{x_i, y_i\}$ represents the coordinate of each vertex of the triangle and $\{x, y\}$ represents the coordinate of a target node on the Cartesian Map. The relative position of a node with respect to the vertices of the triangle can be visualized in Fig C.2.

\[
\lambda_1 = \frac{(y_2 - y_3)(x - x_3) + (x_3 - x_2)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)} \quad (6)
\]

\[
\lambda_2 = \frac{(y_3 - y_1)(x - x_3) + (x_1 - x_3)(y - y_3)}{(y_2 - y_3)(x_1 - x_3) + (x_3 - x_2)(y_1 - y_3)} \quad (7)
\]

\[
\lambda_3 = 1 - \lambda_1 - \lambda_2 \quad (8)
\]

Interpolated nodal values on the Cartesian Map can then be determined as:

\[
S = \lambda_1 S_1 + \lambda_2 S_2 + \lambda_3 S_3 \quad (9)
\]

where, $S$ is the nodal value on the Cartesian Map containing geometry label or physical information like stress. $S_i$ is the $i^{th}$ nodal value of the triangular element within which the node on the Cartesian Map falls.

Next, we performed a comparison of von-Mises stress contours obtained using linear interpolation (with nearest three nodes and nearest five nodes) and Barycentric coordinate interpolation on a square packed model. It is observed (Figure C.4) that the nearest three node linear interpolation generates negative values, whereas von-Mises stresses are non-negative values based on their definition. This implies that the nearest three node linear interpolation introduces singularities during interpolation. The nearest five node linear interpolation manifests several noisy points with minimum (min) stress approximately equal to zero, which is also inaccurate. On the other hand, Barycentric coordinate interpolation can successfully represent the stress distribution contour and stress ranges after interpolation.

In addition to the interpolation method, Cartesian Map density is a key factor that contributes towards interpolation values. In order to ensure that the Cartesian Map captures important statistical features of the geometry and stress distribution contours after interpolation, we run a stress interpolation analysis to establish the Cartesian Map density that gives reasonable interpolation accuracy and efficiency. From this analysis (Figure C.5), we observe that the interpolation accuracy increases with increasing Cartesian Map density, while the interpolation speed decreases. To strike a balance between the accuracy and efficiency, especially accounting for large sample sizes, we select a Cartesian Map density of 79-by-79, which has the shape of 80-by-80 after converting to nodal matrix. This Cartesian Map density can provide an interpolation accuracy above 99% in max stress and a reasonable interpolation speed of 26 samples/minute. An example of interpolation onto Cartesian Map is shown in Figure C.6.
5.2 Statistical Property Analysis

As compared to the existing Stress-Net structure, the objective of our proposed DNN structure is to improve the accuracy of prediction based on a reference data set and render the training process more statistically stable. To that end, we calculate the mean and skewness in the training samples by considering a subset of the data points in the regions where the geometry of the models change (Fig C.7), that is in the vicinity of cutout or fiber. We considered nine points (A-I) in this region and calculated skewness values based on Pearson’s second skewness coefficient shown in Equation 10 [42].

\[
\text{skewness} = \frac{3(\text{mean} - \text{median})}{\text{standard deviation}}
\]  

EQ. 10

Skewness in training data can cause imbalance problem and eventually reduce the prediction accuracy of Neural Network by introducing unbalanced data [43, 44]. There are typically two ways of reducing skewness: 1) model-oriented (reduce skewness in the structure) and data-oriented (reduce skewness by pre-processing)[45]. The sign of skewness manifests in the data distribution, where positive skewed data has tail on the right while negative skewed data has tail on the left. To reduce the skewness in data, element-wise root operation on the training samples is performed for positive skewed data and element-wise power is used to deal with negative skewed data[46].

Since geometry labels are uniform for individual regions in stress prediction analysis of heterogeneous media, nodal stress values are the main source of skewness. Table C.1 shows how the DNN structure changes the max-min data range. For square packed and hexagonal packed models, max-min range changes slightly while data is more zero-centered. For plate with circular cutout and hollow particle reinforced model, max-min ranges increase due to the presence of cutout regions. This effect is more significant for the hollow particle reinforced model since more severe stress concentrations exist. Tables C.2, C.3, C.4 and C.5 summarize the mean, median and skewness values for 1000 different models. Since the difference based structure only performs mean contour subtraction, the data has a zero mean value. On the other hand, since the skewness value does not change when subtracting a constant, the difference contours will have the same skewness value as that of the original contour with changes only in the mean and median values. For plate with circular cutout and hollow particle reinforced models, we add the clean module within DNN, which alters the statistical properties for the nodes inside the cutout region, resulting in few nodes that do not have a zero mean value. Even though we have a few non-zero mean valued points in the hollow region, the clean module switches all the skewness values to positive as compared to the original sample and the difference based sample. This helps in improving the accuracy of the DNN prediction, and also making it easier to implement further steps to reduce the effect of skewness if needed.

6 Machine Learning Model Inputs

As discussed in the previous section, both nodal stress contours and geometry contours for four different models are interpolated onto Cartesian Maps for training the Neural Network. The geometry contours can have different shapes and lead to different effects on Neural Network training. Unique labels are assigned to each region, including cutout region, fiber region, particle ring region and matrix region. In this paper, the same boundary conditions and loads are maintained in all analyses, and hence, the effects of boundary condition labelling are not discussed.

To train the Neural Network and test its performance, we randomly generate a total of 2000 samples for each model based on different cutout or fiber volume fractions. When training the Neural Network for each model, the total samples are randomly split into 80% for training, 10% for cross-validation and 10% for testing. The random split method is controlled based on the pseudorandom number generator[47] developed in scikit-learn[48] to ensure different Neural Network frameworks are trained and tested with the same Finite Element input samples for comparison purposes.

6.1 Plate with circular cutout for ML Model Development

Plate with circular cutout model consists of matrix and cutout region. To identify the two regions, matrix region is labelled as ‘1’ and cutout as ‘0’, as expressed in Equation 11, where \(G_{ij}\) represents the labelled geometry contour.

\[
G_{ij} = \begin{cases} 
1 & \text{if material exists at a node} \\
0 & \text{if material is absent at a node}
\end{cases}
\]  

EQ. 11

Figure 5(a) and (b) shows an example of labelled geometry contour and the interpolated stress contour from FEM mesh onto Cartesian Map, showcasing that the Cartesian Map density sufficiently captures stress contour features for this model.
Figure 5: Geometry contour labelling and Stress interpolation from Finite Element output to Cartesian Map for plate with circular cutout model (a and b), square packed reinforced composite model (c and d), hexagonal packed reinforced composite model (e and f), and hollow particle reinforced composite model (g and h)

6.2 Square and Hexagonal Packed Fiber Reinforced Composites for ML Model Verification

Square packed reinforced composite model consists of a squared shape matrix with fiber embedded at the center, where fiber is made of a stiff material and matrix is made of a relatively softer material. Hexagonal packed reinforced composite consists of a square shaped matrix with fibers embedded at the center as well as four quarter fibers at each corner of the square. In both of these models, the fiber region is labelled as '1' and matrix as '0' in the geometry contour, as described in Equation 12. Figure 5(c) and (e) show an example of labelled geometry contours for square and hexagonal packed models. Their corresponding interpolated stress contours from FEM mesh onto Cartesian Map are shown in Figure 5(d) and (f).

\[
G_{ij} = \begin{cases} 
1 & \text{if the material at a node is fiber} \\
0 & \text{if the material at a node is matrix} 
\end{cases}
\] (12)

6.3 Hollow Particle Reinforced Composite for ML model Verification

Hollow particle reinforced composite model is similar to that plate with circular cutout model, but with an additional reinforcing material ring along the cutout that represents the hollow reinforcing particle. In this case, we have three different regions, labeled as 0, 1 and 2 for no material, reinforcing ring and matrix regions, respectively as shown in
Equation 13. The geometry and corresponding interpolated stress contours from FEM to Cartesian Map are shown in Figure 5(g) and (h).

\[
G_{ij} = \begin{cases} 
1, & \text{if the material at a node is reinforcing particle} \\
2, & \text{if the material at a node is matrix} \\
0, & \text{if material is absent at a node}
\end{cases}
\]

(13)

6.4 Analyzing the Effect of Skewness

We discussed the effect of skewness in training data on the prediction accuracy of Neural Network in section 5.2. We noticed that all the skewness values become positive after adding the clean module to DNN. To determine the optimal solution for reducing the effect to positive skewness and enhancing the prediction accuracy, we introduce a skewness correction factor \( p \). By considering different root \( p \) over the stress contour values prior to training the Neural Network, our goal is to reduce the influence of skewness on the accuracy of prediction. Element-wise root values are calculated before extracting the mean stress contour and correspondingly element-wise power is introduced after obtaining the prediction from the Deconv block. The predicted stress contour can be expressed using Equation 14, where the original input geometry contour to the Neural Network is \( G \). \( NN(w,b) \) is the Neural Network with parameters \( w \) and \( b \). \( \sigma_{\text{average}} \) is the average stress contour used as the reference contour during training, which is calculated using Equation 15. Different values of \( p \) are considered for investigating the hidden relationships within DNN using the hollow particle reinforced composite model. The result of this analysis is discussed later in Section B.

\[
\sigma_{\text{output},ij} = [(G * NN(w,b))_{ij} + \sigma_{\text{average},ij}]^p \quad (0 < p \leq 1)
\]

(14)

\[
\sigma_{\text{average},ij} = \frac{1}{N} \sum_{n=1}^{N} \sqrt{\sigma_{n,ij}}
\]

(15)

7 Results and Discussion

We constructed our Neural Network within Tensorflow 2.0.0 and trained it on GPU as discussed in section 2. To test the prediction capability of different Neural Network frameworks for four different geometric models considered, we used the Stress-Net structure [20] for comparison, which has already been proven to predict accurate stresses with higher efficiency compared to Finite Element method. The accuracy of stress prediction within each component of composite materials, Neural Network training duration and training loss are used for evaluating each model.

7.1 Prediction error definition

We evaluated the prediction accuracy of max stress based on the max stress error rate (MER) as defined in Equation 16. We evaluated the training loss based on the mean squared error (MSE) in stress prediction as defined in Equation 17. Here, \( N \) is the total number of samples in the testing set and \( n \) is the total number of nodes in the Cartesian Map. \( Y_i \) is the nodal stress predicted using the Neural Network and \( \hat{Y}_i \) is the ground truth nodal stress mapped onto the Cartesian map from FEM stress analysis. \( Y_{i,j} \) and \( \hat{Y}_{i,j} \) represent the \( j^{th} \) nodal stress value in \( i^{th} \) sample obtained from the Neural Network and FEM stress analysis, respectively. Training accuracy and efficiency of different Neural Network structures on the four geometric models considered in this paper are summarized below.

\[
MER = \frac{1}{N} \sum_{i=1}^{N} \frac{\max(\hat{Y}_i) - \max(Y_i)}{\max(Y_i)} \times 100%
\]

(16)

\[
MSE = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{1}{n} \sum_{j=1}^{n} (Y_{i,j} - \hat{Y}_{i,j})^2 \right)
\]

(17)

7.2 Neural Network prediction results

Next, we evaluated different Neural Network structures, including a baseline Stress-Net structure and three types of difference-based structures developed by us: DNN, DNN-N and DNN-NC, on each of the four geometric models.
considered in this paper. DNN-NC structure is used only when cutout region exists in the geometry. That is, all three
difference-based structures are evaluated for plate with circular cutout model and hollow particle reinforced composite
model, whereas, only two difference-based structures (DNN and DNN-N) are evaluated using square and hexagonal
packed fiber reinforced composite models. Each Neural Network is trained with total of 1000 and 2000 samples
separately. Figure C.8 to C.11 show the training loss profile for all four geometric models with respect to training
epochs on a linear and log scale with different sample sizes. These plots indicate that the loss converges rapidly, and 80
epochs is sufficient for training the Neural Network for both sample sizes. Figure 6 to 9 shows an example of predicted
stress contour compared to that generated by FEM solver for each geometric model as well as the errors (MER and
MSE) and training duration while considering 1000 total number of samples. Tabular data of Figure 8 to 9 along with
that corresponding to 2000 total samples is provided in Figure C.12 to C.15 and Tables C.7 to C.23.

Based on the training results obtained for these four geometric models, we conclude the following (prediction results
for 1000 total samples are used for illustration purposes):

1. Training accuracy for models without cutout region:
   • Square packed fiber reinforced composite model (Figure 6): Stress-Net structure manifested average MER of
     0.74% and 0.73% for fiber and matrix, respectively, and MSE of 0.14. The DNN structure is able to reduce both
     fiber and matrix MER to 0.56%. DNN-N structure further reduced the fiber and matrix MER to 0.46% and 0.45%,
     resulting in about 38% and 20% reduction in MER for max stress prediction of fiber and matrix compared to
     Stress-Net structure, respectively. A stabilized reduction in MSE value from 0.14 to 0.11 is also obtained from
     DNN-N structure. Contours plots of the difference in predicted stress distributions using Stress-Net and DNN-N
     structures compared to FEA stress output are shown in Figure 6 (b). We observe large stress differences in
     Stress-Net prediction (in dark red and blue), which reduced when DNN-N was used.
   • Hexagonal packed fiber reinforced composite model (Figure 7): Stress-Net structure manifested average MER
     of 0.78% and 1.10% for fiber and matrix, respectively, and MSE of 0.13. The DNN structure is able to reduce
     both fiber and matrix MER to 0.59% and 0.65%. DNN-N structure further reduced the fiber and matrix MER to
     0.46% and 0.63%, respectively, resulting in about 41% and 43% reduction in MER for max stress prediction of
     fiber and matrix compared to Stress-Net structure. Similar to the square packed model, a stable reduction in MSE
     value from 0.13 to 0.11 is obtained from DNN-N for this case. Contours plots of the difference in predicted stress
     distributions using Stress-Net and DNN-N structures compared to FEA stress output are shown in Figure 7 (b).
     Similar to the square packed fiber reinforced composite model, the large stress differences indicated by dark red
     and blue colors are reduced when DNN-N framework was used.
   • In general, DNN-N significantly reduces the max stress error rate in both fiber and matrix, as well as marginally
     lower MSE value compared to the baseline model. Hence, our proposed DNN-N structure is shown to be the best
     structure for stress prediction in composite models without cutout region.

2. Training accuracy for models with cutout region:
   • Plate with circular cutout model (Figure 8): Stress-Net structure manifested average MER of 5.02% and MSE of
     0.64. The DNN structure is able to reduce the MER by 50% compared to Stress-Net structures, however, resulting
     in MSE values 3 times larger due to imbalanced data samples. DNN-N structure further reduced the MER and
     MSE values, however still resulted in large MSE values. This is attributed to extra noise from the cutout region
during mean contour subtraction. Finally, after adding the clean module to remove unnecessary information
during the subtraction step, the DNN-NC structure manifested the lowest MER and MSE values. That is, MER for
max stress prediction reduces to 1.06% (decreases by 76%) and the MSE value also reduced from 0.64 to 0.09.
Therefore, we conclude that DNN-NC structure is capable of providing the highest accuracy in predicting stress
distribution for plate with circular cutout model. Figure 8 (b) shows the distribution of the difference in predicted
stresses using Stress-Net and DNN-N structures compared to FEA stress output. Here, DNN-NC manifests very
little prediction differences compared to Stress-Net as indicated by the red and blue colors in the contour map,
especially in the vicinity of the circular cutout.
   • Hollow particle reinforced composite model (Figure 9): Stress-Net structure manifested average MER of 2.87%
     and 6.82% for fiber and matrix, respectively, and MSE of 2.0. Due to the effect of severe stress concentrations
     ($R_d>4$) around the cutout and within the ring (particle) as well as imbalanced data sample, the negative influence
     of noise in the hollow region is more significant as compared to the plate with circular cutout model, leading to bad
     prediction accuracy for both DNN and DNN-N structures. Hence, by introducing the clean module, the DNN-NC
     reduced the average MER to 1.75% and 1.73% in the ring and matrix regions, respectively, and correspondingly
     resulted in 39% and 75% reduction. In addition, the average MSE value also decreased significantly to 0.25 from
     2.24 as compared to that from the Stress-Net prediction. Overall, the DNN-NC resulted in the best prediction on
     hollow particle reinforced model. Similar to plate with circular cutout model, DNN-NC significantly reduces the
     large prediction differences in stress distribution compared to Stress-Net as shown in Figure 9 (b) as highlighted
     by dark red and blue colors.
In general, we have shown that the DNN-NC structure has the best performance in terms of accurate stress prediction compared to Stress-Net (baseline) and DNN-N structures for heterogeneous media with discontinuities, like hollow particle fiber reinforced composites or plate with circular cutout region, when significant stress concentrations exist ($R_{\sigma} > 2$).

Figure 6: Square packed fiber reinforced composite: (a) Comparison of DNN-N predicted and FEA stress output contours (b) Comparison of difference in predicted stress distribution of Stress-Net and DNN compared to FEA stress output (c) Neural Network prediction error (d) Training duration for 1000 samples

Figure 7: Hexagonal packed fiber reinforced composite: (a) Comparison of DNN-N predicted and FEA stress output contours (b) Comparison of difference in predicted stress distribution of Stress-Net and DNN compared to FEA stress output (c) Neural Network prediction error (d) Training duration for 1000 samples
Figure 8: Plate with circular cutout model: (a) Comparison of DNN-NC predicted and FEA stress output contours (b) Comparison of difference in predicted stress distribution of Stress-Net and DNN compared to FEA stress output (c) Neural Network prediction error (d) Training duration for 1000 samples

Figure 9: Hollow particle reinforced composite: (a) Comparison of DNN-NC predicted and FEA stress output contours (b) Comparison of difference in predicted stress distribution of Stress-Net and DNN-NC compared to FEA stress output (c) Neural Network prediction error (d) Training duration for 1000 samples

3. Training duration: In general, the training duration is the lowest for Stress-Net structure compared to the difference-based structures presented in this paper. Since the difference-based structures are built based on the Stress-Net structure, they have marginally longer training duration but in a reasonable range of approximately 150 seconds more for 800 training samples. Among the three difference-based structures presented in this paper, adding the normalization module (DNN-N) marginally reduces the training duration as it enforces the input and output of the
Encoder-Decoder structure to stay within similar limits. On the other hand, adding the clean module (DNN-NC) adds additional computational cost during the training of the Neural Network as we perform Hadamard product calculation in this module. It should be noted here that the Neural Network training stage is only for establishing the parameters and is a one-time event, and the prediction duration per sample for DNN-N and DNN-NC is on average only 0.2 seconds more compared to Stress-Net. Moreover, the Neural Network prediction speed per sample is 4 to 8 times faster compared to performing a FEM analysis using ABAQUS (shown in Table C.15). Therefore, the prediction efficiency of our proposed DNN frameworks is comparable to the baseline model and much more efficient than Finite Element software.

8 Conclusions

In this paper, we have presented a novel Neural Network framework as a surrogate for traditional FEM approach to predict stress distribution in heterogeneous media, like composite materials. Our approach consists of a set of Difference-based Neural Network (DNN) frameworks capable of predicting stress distributions with very high accuracy for different types of composite materials. Four different composite micromechanical models were considered for validating the performances of our Neural Network structures. The proposed DNN structure included a normalization module (DNN-N) for all geometries considered, while we additionally introduced a clean module with DNN-N, named DNN-NC, for geometries with discontinuities. We showed that for composite models without discontinuities (like square and hexagonal fiber packed models), DNN-N structure resulted in the best prediction accuracy. For composite models with discontinuities (like a plate with circular cutout and hollow particle reinforced composite models) when large stress concentrations exist, DNN-NC structure resulted in best prediction accuracy. The DNN frameworks presented in this paper for stress prediction in heterogeneous media can be used in future studies including mechanical property prediction and composite structure optimization at multiple length scales.

Key contributions of this paper are:

1. This is the first attempt to our best knowledge that brings Convolutional Neural Network based Machine Learning for stress distribution prediction for heterogeneous media like composite materials.
2. We introduce a novel Difference-based Neural Network framework which utilizes a set of reference models from the training set and focuses specially on training the difference contours between the target model and the reference set. This is shown to drive towards improved stress prediction accuracy when high stress concentrations manifest within heterogeneous media, with or without discontinuities like cutouts.
3. We show that the Difference-based Neural Network framework improves the stress prediction accuracy significantly compared to existing baseline structures, especially when large local stress concentrations exist. Moreover, our proposed framework has a 4 to 8 times faster prediction speed compared to that of existing Finite Element software.

Acknowledgements

The authors would like to acknowledge the support through the 2019 ONR Young Investigator Program [N00014-19-1-2206] through Sea-based Aviation: Structures and Materials Program for conducting the research presented here. The authors would also like to acknowledge the support from the University of Wisconsin Graduate Fellowship.

Author Contributions

P.P. and H.F. conceptualized and developed this study. H.F. implemented the research presented in this paper. H.F. and P.P. evaluated the outcomes of the work and drafted the manuscript.

References

[1] T. Hughes. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, volume 78. 01 2000.
[2] P. Wriggers. Nonlinear Finite Element Methods, volume 4. 01 2008.
[3] ABAQUS (2011) Dassualt Systèmes. Abaqus documentation. 2011.
[4] E. Barbero. Finite Element Analysis of Composite Materials. 01 2008.
[5] Y. Efendiev, J. Galvis, and T. Y. Hou. Generalized multiscale finite element methods (gmsfem). Journal of Computational Physics, 251:116 – 135, 2013.
[6] T. Y. Hou and X. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of Computational Physics*, 134(1):169 – 189, 1997.

[7] E. Barbero. *Finite Element Analysis of Composite Materials*. 01 2008.

[8] N. Umetani. Exploring generative 3d shapes using autoencoder networks. pages 1–4, 11 2017.

[9] E. Ulu, R. Zhang, and L. Kara. A data-driven investigation and estimation of optimal topologies under variable loading configurations. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 4:1–12, 08 2015.

[10] A. Mardt, L. Pasquali, Wu H, and F. Noé. Vampnets: Deep learning of molecular kinetics. *Nature Communications*, 9, 10 2017.

[11] G. Tribello, M. Ceriotti, and M. Parrinello. A self-learning algorithm for biased molecular dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, 107:17509–17514, 01 2010.

[12] L. Bottou and Y. LeCun. On-line learning for very large data sets. *Applied Stochastic Models in Business and Industry*, 21:137 – 151, 03 2005.

[13] J. Li, Y. Wong, Q. Zhao, and M. S. Kankanahalli. Learning to learn from noisy labeled data. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5046–5054, 2019.

[14] T. Hong, L. Tseng, and S. Wang. Learning rules from incomplete training examples by rough sets. *Expert Systems with Applications*, 22(4):285 – 293, 2002.

[15] Awad W.A and ELseuofi S.M. Machine learning methods for spam e-mail classification. *International Journal of Computer Science & Information Technology*, 3, 02 2011.

[16] S. Weiss and I. Kapouleas. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. volume 1, pages 781–787, 01 1989.

[17] L. Liang, M. Liu, C. Martin, J. Elefteriades, and W. Sun. A machine learning approach to investigate the relationship between shape features and numerically predicted risk of ascending aortic aneurysm. *Biomechanics and modeling in mechanobiology*, 16, 04 2017.

[18] X. Guo, W. Li, and F. Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, page 481–490, New York, NY, USA, 2016. Association for Computing Machinery.

[19] Y. Afshar, S. Bhatnagar, S. Pan, K. Duraisamy, and S. Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. 05 2019.

[20] Z. Nie, H. Jiang, and L. Kara. Stress field prediction in cantilevered structures using convolutional neural networks. *Journal of Computing and Information Science in Engineering*, page 1, 06 2019.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[22] Y. Bengio and Y. LeCun. Convolutional networks for images, speech, and time-series. 11 1997.

[23] I. Sutskever, O. Vinyals, and Q. Le. Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 4, 09 2014.

[24] P. Breunig, V. Damodaran, K. Shahapurkar, S. Waddar, M. Doddamani, P. Jeyaraj, and P. Prabhakar. Dynamic impact behavior of syntactic foam core sandwich composites. *Journal of Composite Materials*, 54(4):535–547, 2020.

[25] K. Shahapurkar, C.D. Garcia, M. Doddamani, G.C.M. Kumar, and P. Prabhakar. Compressive behavior of cenosphere/epoxy syntactic foams in arctic conditions. *Composites Part B: Engineering*, 135(June 2017):253–262, 2018.

[26] C.D. Garcia, K. Shahapurkar, M. Doddamani, G.C.M. Kumar, and P. Prabhakar. Effect of arctic environment on flexural behavior of fly ash cenosphere reinforced epoxy syntactic foams. *Composites Part B: Engineering*, 151:265–273, 2018.

[27] M. L. Jayavardhan and D. Mrityunjay. Quasi-static compressive response of compression molded glass microballoon/hdpe syntactic foam. *Composites Part B Engineering*, 149, 06 2018.

[28] R. M. Jones. Mechanics of composite materials, second edition. CRC Press, 1999.

[29] P. Prabhakar and A. M. Waas. Interaction between kinking and splitting in the compressive failure of unidirectional fiber reinforced laminated composites. *Composite Structures*, 98:85 – 92, 2013.
[30] P. Prabhakar and A. M. Waas. Micromechanical modeling to determine the compressive strength and failure mode interaction of multidirectional laminates. *Composites Part A: Applied Science and Manufacturing*, 50:11 – 21, 2013.

[31] P. Prabhakar and A. M. Waas. Upscaling from a micro-mechanics model to capture laminate compressive strength due to kink banding instability. *Computational Materials Science*, 67:40 – 47, 2013.

[32] A. E. Standage and R. Prescott. High elastic modulus carbon fibre. *Nature*, page 211, 07 1966.

[33] K. Cho, v. M. Bart, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. 09 2014.

[34] P. Juszczak, D. Tax, and R. Duin. Feature scaling in support vector data description. 05 2002.

[35] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. 2018.

[36] L. Xu, J. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1790–1798. Curran Associates, Inc., 2014.

[37] G. Xavier and B. Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS’10)*. Society for Artificial Intelligence and Statistics, 2010.

[38] L. Bottou. *On-line Learning and Stochastic Approximations*, page 9–42. Publications of the Newton Institute. Cambridge University Press, 1999.

[39] G. P. H. Styan. Hadamard products and multivariate statistical analysis. *Linear Algebra and its Applications*, 6:217 – 240, 1973.

[40] I. Amidror. Scattered data interpolation methods for electronic imaging systems: A survey. *Journal of Electronic Imaging*, 11, 04 2002.

[41] E. Hille. *Analytic Function Theory*, volume 1. Chelsea Publishing Company, New York, 2 edition, 1982. footnote 1.

[42] J. F. Kenney and E. S. Keeping, editors. *Mathematics of Statistics*. second edition. Van Nostrand, New York, 1951.

[43] Japkowicz N and Shaju S. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6:429–449, 11 2002.

[44] R. Longadge and S. Dongre. Class imbalance problem in data mining review. *CoRR*, abs/1305.1707, 2013.

[45] P. N. SubbaNarasimha, B. Arinze, and M. Anandarajan. The predictive accuracy of artificial neural networks and multiple regression in the case of skewed data: exploration of some issues. *Expert Systems with Applications*, 19(2):117 – 123, 2000.

[46] J. D. Emerson, editor. *Mathematical aspects of transformation*, pages 247–282. John Wiley, New York, 1983.

[47] P. L’Ecuyer. *Pseudorandom Number Generators*. 05 2010.

[48] 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, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
Appendix

A Brief overview of Finite Element Method

To obtain the physical property contours, the target solid domain is first discretized by meshing with different types of elements. After assigning material properties, boundary conditions and loads, this meshed domain is passed into a FEM solver. The FEM solver first calculates element stiffness matrices and force vectors based on individual element as shown in Equation 18 and 19, where $N$ is the shape function, $B$ is the strain-displacement matrix based on shape function and $C$ is the constitutive matrix defined with elastic modulus and Poisson’s ratio. $f$ represents the body force and $t$ represents external traction.

\[
K^e = \int_{V_e} B_e^T C_e B_e dV_e
\]  

(18)

\[
P^e = \int_{V_e} N^T f dV_e + \int_{S_e} N^T t dS_e
\]  

(19)

The element stiffness matrices $K^e$ and force vectors $P^e$ are assembled over all elements in meshed geometry into global stiffness matrix $K$ and force vector $P$, and solved for the nodal displacements $u$ as well as stresses $\sigma$ using Equation 20 and 21 for static state analysis. Upon extracting the nodal stresses, the 2D von Mises stress can be further calculated as defined in Equation 22.

\[
K u = P
\]  

(20)

\[
\sigma = C Bu
\]  

(21)

\[
\sigma_{vonMises} = \sqrt{\sigma_x^2 + \sigma_y^2 - \sigma_x \sigma_y + 3\tau_{xy}^2}
\]  

(22)

B Impact of skewness

To understand the effect of skewness in training samples while training the DNN and to determine the optimal value for the power $p$ in Equation 14 as discussed in section 6.4, the hollow particle reinforced model is chosen as it manifests most severe stress concentrations compared to other three models considered in this paper. We investigated the relationship between the accuracy of the Neural Network prediction and $p$ values ranging between 0 and 1. Figure B.1 shows the variation of predicted MER in the ring and matrix regions as well as MSE for different values of $p$. The results from this analysis show that the prediction error increases rapidly when $p < 0.35$ and is the lowest when $p = 0.84$. However, the improvement in prediction accuracy is marginal with $p = 0.84$ when compared to $p = 1$. That is, the DNN structure can effectively reduce the effect of skewness on stress contour prediction without considering any skewness reduction method, hence, a $p$ value of 1 is suggested for convenience. Detailed data are provided in Table C.6.
Figure B.1: Influence of skewness power $p$ on prediction error

C Figures

Figure C.1: 80-by-80 Cartesian Map matrix
Figure C.2: Relative position between target node (A,B,C) of the Cartesian Map and triangular element from the FEM mesh.

Figure C.3: An example of a Difference-based Neural Network prediction flowchart on square packed fiber reinforced model.

Figure C.4: Interpolation of stress contours on a fiber reinforced square packed model using (a) linear interpolation with nearest three nodes (b) linear interpolation with nearest five nodes (c) Barycentric coordinate.
Figure C.5: (a) Cartesian Map interpolation error rate (b) Cartesian Map interpolation speed

Figure C.6: Interpolation on Cartesian Map for plate with circular cutout model: (a) Meshed geometry (b) Cartesian Map geometry (c) Meshed stress (d) Cartesian Map stress

Figure C.7: Points selected for statistical analysis in (a) plate with circular cutout (b) square packed fiber reinforced composite (c) hexagonal packed fiber reinforced composite (d) hollow particle reinforced composite
Figure C.8: Training loss for plate with circular cutout with: (a) 1000 samples in linear scale (b) 1000 samples in log scale (c) 2000 samples in linear scale (d) 2000 samples in log scale

Figure C.9: Training loss for square packed fiber reinforced composite with: (a) 1000 samples in linear scale (b) 1000 samples in log scale (c) 2000 samples in linear scale (d) 2000 samples in log scale
Figure C.10: Training loss for hexagonal packed fiber reinforced composite with: (a) 1000 samples in linear scale (b) 1000 samples in log scale (c) 2000 samples in linear scale (d) 2000 samples in log scale

Figure C.11: Training loss for hollow particle reinforced composite with: (a) 1000 samples in linear scale (b) 1000 samples in log scale (c) 2000 samples in linear scale (d) 2000 samples in log scale
Figure C.12: Plate with circular cutout model with 2000 samples: (a) Neural Network prediction error (b) Training duration

Figure C.13: Square packed fiber reinforced composite with 2000 samples: (a) Neural Network prediction error (b) Training duration

Figure C.14: Hexagonal packed fiber reinforced composite with 2000 samples: (a) Neural Network prediction error (b) Training duration

Figure C.15: Hollow particle reinforced composite with 2000 samples: (a) Neural Network prediction error (b) Training duration
## D Tables

### Table C.1: Max and min values of model samples

| Geometry max label | Stress-Net | DNN | DNN-NC | Square packed | Stress-Net | DNN | DNN-NC | Hexagonal packed | Stress-Net | DNN | DNN-NC | Hollow particle reinforced | Stress-Net | DNN | DNN-NC |
|-------------------|------------|-----|--------|--------------|------------|-----|--------|----------------|------------|-----|--------|----------------------------|------------|-----|--------|
| Stress max value  | 68.21      | 64.80 | 64.80  | 67.83        | 14.85      | 62.55| 15.47  | 122.38         | 115.47     | 115.47|
| Stress min value  | 0          | -42.03 | -15.21 | 23.29        | -16.11     | 27.98| -21.59 | 0             | -63.76     | -44.49|

### Table C.2: Statistics properties of 1000 samples for plate with circular cutout model

| Point # | Mean | Median | Skewness | DNN / DNN-N | Mean | Median | Skewness |
|---------|------|--------|----------|-------------|------|--------|----------|
| A       | 35.25| 34.30  | 0.63     | -0.95       | 0.63 | 0.00   | 0.00     |
| B       | 36.96| 35.80  | 0.61     | -1.15       | 0.61 | 0.00   | 0.00     |
| C       | 39.01| 37.79  | 0.54     | -1.22       | 0.54 | 0.00   | 0.00     |
| D       | 41.34| 40.42  | 0.35     | -0.92       | 0.35 | 0.00   | 0.00     |
| E       | 38.93| 39.98  | -0.22    | 1.05        | -0.22| 3.43   | 1.05     |
| F       | 35.34| 39.82  | -0.71    | 4.48        | -0.71| 6.82   | 4.48     |
| G       | 32.23| 38.76  | -0.90    | 6.53        | -0.90| 9.25   | 6.53     |
| H       | 28.17| 36.26  | -1.03    | 8.09        | -1.03| 11.07  | 8.09     |
| I       | 24.20| 34.31  | -1.23    | 10.11       | -1.23| 11.93  | 10.11    |

### Table C.3: Statistics properties of 1000 samples for square packed fiber reinforced composite model

| Point # | Mean | Median | Skewness | DNN / DNN-N | Mean | Median | Skewness |
|---------|------|--------|----------|-------------|------|--------|----------|
| A       | 58.39| 58.33  | 0.07     | 0.00        | -0.06| 0.07   | 0.07     |
| B       | 58.72| 58.66  | 0.07     | 0.00        | -0.06| 0.07   | 0.07     |
| C       | 59.04| 58.98  | 0.07     | 0.00        | -0.06| 0.07   | 0.07     |
| D       | 59.34| 59.27  | 0.07     | 0.00        | -0.06| 0.07   | 0.07     |
| E       | 59.62| 59.55  | 0.07     | 0.00        | -0.07| 0.07   | 0.07     |
| F       | 59.87| 59.82  | 0.06     | 0.00        | -0.06| 0.06   | 0.06     |
| G       | 60.11| 60.05  | 0.06     | 0.00        | -0.06| 0.06   | 0.06     |
| H       | 60.33| 60.27  | 0.07     | 0.00        | -0.07| 0.07   | 0.07     |
| I       | 60.53| 60.47  | 0.06     | 0.00        | -0.06| 0.06   | 0.06     |

### Table C.4: Statistics properties of 1000 samples for hexagonal packed fiber reinforced composite model

| Point # | Mean | Median | Skewness | DNN / DNN-N | Mean | Median | Skewness |
|---------|------|--------|----------|-------------|------|--------|----------|
| A       | 53.00| 52.91  | 0.12     | 0.00        | -0.09| 0.12   | 0.00     |
| B       | 52.30| 52.21  | 0.12     | 0.00        | -0.08| 0.12   | 0.00     |
| C       | 51.63| 51.52  | 0.17     | 0.00        | -0.12| 0.17   | 0.00     |
| D       | 51.02| 50.93  | 0.14     | 0.00        | -0.09| 0.14   | 0.00     |
| E       | 50.46| 50.35  | 0.18     | 0.00        | -0.11| 0.18   | 0.00     |
| F       | 49.94| 49.85  | 0.15     | 0.00        | -0.09| 0.15   | 0.00     |
| G       | 49.47| 49.39  | 0.14     | 0.00        | -0.08| 0.14   | 0.00     |
| H       | 49.05| 48.98  | 0.13     | 0.00        | -0.07| 0.13   | 0.00     |
| I       | 48.68| 48.60  | 0.14     | 0.00        | -0.08| 0.14   | 0.00     |
Table C.5: Statistics properties of 1000 samples for hollow particle reinforced composite model

| Point # | Stress-Net | DNN / DNN-N | DNN-NC |
|---------|------------|-------------|--------|
| A       | 37.54      | 37.23       | 0.30   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -3.03       | Median | -3.03 |
|         | Skewness   | 0.93        | Skewness| 0.93 |
| B       | 41.34      | 38.25       | 0.03   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -4.04       | Median | -4.04 |
|         | Skewness   | 0.93        | Skewness| 0.93 |
| C       | 52.14      | 39.69       | 1.46   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -14.45      | Median | -14.45 |
|         | Skewness   | 1.46        | Skewness| 1.46 |
| D       | 55.26      | 40.24       | 1.31   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -15.42      | Median | -15.42 |
|         | Skewness   | 1.31        | Skewness| 1.31 |
| E       | 49.19      | 39.72       | 0.71   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -9.48       | Median | -9.48 |
|         | Skewness   | 0.71        | Skewness| 0.71 |
| F       | 41.78      | 39.68       | 0.14   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -2.10       | Median | -2.10 |
|         | Skewness   | 0.14        | Skewness| 0.14 |
| G       | 33.92      | 2.28        | 2.28   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -33.92      | Median | -33.92 |
|         | Skewness   | 2.28        | Skewness| 2.28 |
| H       | 28.17      | 0.00        | 1.78   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -28.17      | Median | -28.17 |
|         | Skewness   | 1.78        | Skewness| 1.78 |
| I       | 11.75      | 0.00        | 0.99   |
|         | Mean       | 0.00        | Mean   | 0.00 |
|         | Median     | -11.75      | Median | -11.75 |
|         | Skewness   | 0.99        | Skewness| 0.99 |

Table C.6: Effect of P value on prediction accuracy

| P value | DNN-NC |
|---------|--------|
|         | Fiber MER | Matrix MER | MSE    |
| 1/6     | 5.53    | 10.06      | 3.74   |
| 2/6     | 1.03    | 3.04       | 0.28   |
| 3/6     | 1.13    | 2.52       | 0.26   |
| 4/6     | 0.80    | 2.46       | 0.24   |
| 5/6     | 0.74    | 1.90       | 0.22   |
| 1       | 0.90    | 1.93       | 0.29   |

Table C.7: Plate with circular cutout model - Neural Network training duration for 1000 total samples

|                  | Stress-Net | DNN | DNN-N | DNN-NC |
|------------------|------------|-----|-------|--------|
| Data Processing Time | 110.6 sec | 121.4 sec | 123.7 sec | 183.9 sec |
| Training Time   | 246.0 sec | 374.1 sec | 331.7 sec | 378.6 sec |
| Total Time     | 356.6 sec | 495.5 sec | 455.4 sec | 562.5 sec |

Table C.8: Plate with circular cutout model - Neural Network training duration for 2000 total samples

|                  | Stress-Net | DNN | DNN-N | DNN-NC |
|------------------|------------|-----|-------|--------|
| Data Processing Time | 267.5 sec | 278.8 sec | 280.1 sec | 375.4 sec |
| Training Time   | 266.6 sec | 409.6 sec | 377.9 sec | 421.8 sec |
| Total Time     | 534.1 sec | 688.4 sec | 658.0 sec | 797.2 sec |

Table C.9: Square packed fiber reinforced composite - Neural Network training duration for 1000 total samples

|                  | Stress-Net | DNN | DNN-N | DNN-NC |
|------------------|------------|-----|-------|--------|
| Data Processing Time | 150.8 sec | 152.1 sec | 155.6 sec | 148.4 sec |
| Training Time   | 198.8 sec | 416.0 sec | 230.2 sec | 341.7 sec |
| Total Time     | 349.6 sec | 568.1 sec | 385.8 sec | 490.1 sec |

Table C.10: Square packed fiber reinforced composite - Neural Network training duration for 2000 total samples

|                  | Stress-Net | DNN | DNN-N | DNN-NC |
|------------------|------------|-----|-------|--------|
| Data Processing Time | 294.3 sec | 306.0 sec | 309.6 sec | 322.8 sec |
| Training Time   | 220.3 sec | 412.3 sec | 250.0 sec | 389.1 sec |
| Total Time     | 514.6 sec | 718.3 sec | 559.6 sec | 711.9 sec |
Table C.11: Hexagonal packed fiber reinforced composite - Neural Network training duration for 1000 total samples

|                   | Stress-Net | DNN    | DNN-N  | DNN-NC |
|-------------------|------------|--------|--------|--------|
| Data Processing Time | 126.8 sec  | 142.3 sec | 140.5 sec | 158.9 sec |
| Training Time     | 237.3 sec  | 402.9 sec | 275.2 sec | 347.1 sec |
| Total Time        | 364.1 sec  | 545.2 sec | 415.7 sec | 506.0 sec |

Table C.12: Hexagonal packed fiber reinforced composite - Neural Network training duration for 2000 total samples

|                   | Stress-Net | DNN    | DNN-N  | DNN-NC |
|-------------------|------------|--------|--------|--------|
| Data Processing Time | 255.6 sec  | 268.2 sec | 277.3 sec | 273.5 sec |
| Training Time     | 262.1 sec  | 458.4 sec | 284.6 sec | 383.7 sec |
| Total Time        | 517.7 sec  | 726.6 sec | 562.1 sec | 657.2 sec |

Table C.13: Hollow particle reinforced composite - Neural Network training duration for 1000 total samples

|                   | Stress-Net | DNN    | DNN-N  | DNN-NC |
|-------------------|------------|--------|--------|--------|
| Data Processing Time | 126.0 sec  | 127.8 sec | 130.5 sec | 163.8 sec |
| Training Time     | 190.4 sec  | 421.6 sec | 268.6 sec | 372.5 sec |
| Total Time        | 316.4 sec  | 549.4 sec | 399.1 sec | 536.3 sec |

Table C.14: Hollow particle reinforced composite - Neural Network training duration for 2000 total samples

|                   | Stress-Net | DNN    | DNN-N  | DNN-NC |
|-------------------|------------|--------|--------|--------|
| Data Processing Time | 285.3 sec  | 289.2 sec | 295.5 sec | 289.5 sec |
| Training Time     | 270.3 sec  | 453.4 sec | 313.7 sec | 372.5 sec |
| Total Time        | 555.6 sec  | 742.6 sec | 609.2 sec | 662.0 sec |

Table C.15: FEM analysis and Neural Network prediction duration per sample

|                   | Plate with cutout | Square packed | Hexagonal packed | Hollow particle reinforced |
|-------------------|-------------------|---------------|------------------|---------------------------|
| FEM (ABAQUS)      | 6.32 sec          | 6.67 sec      | 7.50 sec         | 8.57 sec                  |
| Stress-Net        | 1.18 sec          | 0.96 sec      | 0.83 sec         | 0.84 sec                  |
| DNN               | 1.64 sec          | 1.71 sec      | 1.79 sec         | 1.93 sec                  |
| DNN-N             | 1.38 sec          | 1.15 sec      | 1.08 sec         | 1.09 sec                  |
| DNN-NC            | 1.40 sec          | 1.25 sec      | —                | —                         |

Table C.16: Plate with circular cutout model - Neural Network prediction error rate for 1000 total samples

| Neural Network | Random Split 1 | Random Split 2 | Random Split 3 |
|---------------|---------------|---------------|---------------|
|               | Matrix Stress Prediction | Matrix Stress Prediction | Matrix Stress Prediction |
|               | MER | MSE | MER | MSE | MER | MSE |
| Stress-Net    | 5.40% | 0.63 | 4.87% | 0.61 | 2.46% | 0.71 |
| DNN           | 1.96% | 1.86 | 2.41% | 2.49 | 2.32% | 2.28 |
| DNN-N         | 1.94% | 1.63 | 2.40% | 1.86 | 1.81% | 2.09 |
| DNN-NC        | 1.09% | 0.11 | 1.13% | 0.10 | 1.20% | 0.09 |

Table C.17: Plate with circular cutout model - Neural Network prediction error rate for 2000 total samples

| Neural Network | Random Split 1 | Random Split 2 | Random Split 3 |
|---------------|---------------|---------------|---------------|
|               | Matrix Stress Prediction | Matrix Stress Prediction | Matrix Stress Prediction |
|               | MER | MSE | MER | MSE | MER | MSE |
| Stress-Net    | 6.04% | 0.71 | 4.48% | 0.57 | 4.56% | 0.65 |
| DNN           | 3.71% | 1.68 | 4.48% | 0.57 | 4.56% | 0.65 |
| DNN-N         | 2.37% | 1.27 | 1.92% | 1.94 | 1.75% | 1.27 |
| DNN-NC        | 0.98% | 0.10 | 1.02% | 0.09 | 1.17% | 0.09 |
Table C.18: Square packed fiber reinforced composite - Neural Network prediction for 1000 total samples

| Neural Network | Random Split 1 | Random Split 2 | Random Split 3 |
|----------------|----------------|----------------|----------------|
|                | Fiber MER     | Matrix MER     | MSE            | Fiber MER     | Matrix MER     | MSE            | Fiber MER     | Matrix MER     | MSE            |
| Stress-Net     | 0.81%         | 0.67%         | 0.18           | 0.66%         | 0.83%         | 0.14           | 0.76%         | 0.69%         | 0.11           |
| DNN            | 0.33%         | 0.24%         | 0.08           | 0.63%         | 0.63%         | 0.19           | 0.38%         | 0.45%         | 0.08           |
| DNN-N          | 0.37%         | 0.32%         | 0.10           | 0.44%         | 0.52%         | 0.12           | 0.57%         | 0.51%         | 0.12           |

Table C.19: Square packed fiber reinforced composite - Neural Network prediction for 2000 total samples

| Neural Network | Random Split 1 | Random Split 2 | Random Split 3 |
|----------------|----------------|----------------|----------------|
|                | Fiber MER     | Matrix MER     | MSE            | Fiber MER     | Matrix MER     | MSE            | Fiber MER     | Matrix MER     | MSE            |
| Stress-Net     | 0.88%         | 0.79%         | 0.16           | 0.58%         | 0.60%         | 0.14           | 1.00%         | 0.76%         | 0.16           |
| DNN            | 0.20%         | 0.21%         | 0.13           | 0.44%         | 0.54%         | 0.09           | 1.16%         | 1.10%         | 0.23           |
| DNN-N          | 0.41%         | 0.36%         | 0.11           | 0.55%         | 0.35%         | 0.11           | 0.47%         | 0.41%         | 0.11           |

Table C.20: Hexagonal packed fiber reinforced composite - Neural Network prediction for 1000 total samples

| Neural Network | Random Split 1 | Random Split 2 | Random Split 3 |
|----------------|----------------|----------------|----------------|
|                | Fiber MER     | Matrix MER     | MSE            | Fiber MER     | Matrix MER     | MSE            | Fiber MER     | Matrix MER     | MSE            |
| Stress-Net     | 0.97%         | 1.26%         | 0.17           | 0.79%         | 1.06%         | 0.15           | 0.57%         | 0.99%         | 0.15           |
| DNN            | 0.71%         | 0.93%         | 0.80           | 0.97%         | 0.31%         | 2.59           | 0.49%         | 0.72%         | 0.15           |
| DNN-N          | 0.33%         | 0.38%         | 0.12           | 0.47%         | 0.56%         | 0.12           | 0.47%         | 0.40%         | 0.13           |

Table C.21: Hexagonal packed fiber reinforced composite - Neural Network prediction for 2000 total samples

| Neural Network | Random Split 1 | Random Split 2 | Random Split 3 |
|----------------|----------------|----------------|----------------|
|                | Fiber MER     | Matrix MER     | MSE            | Fiber MER     | Matrix MER     | MSE            | Fiber MER     | Matrix MER     | MSE            |
| Stress-Net     | 0.44%         | 0.85%         | 0.14           | 0.71%         | 0.94%         | 0.16           | 0.79%         | 0.86%         | 0.14           |
| DNN            | 0.51%         | 0.57%         | 0.09           | 0.53%         | 0.62%         | 0.09           | 0.54%         | 0.76%         | 1.61           |
| DNN-N          | 0.47%         | 0.56%         | 0.10           | 0.48%         | 0.66%         | 0.11           | 0.43%         | 0.56%         | 0.11           |

Table C.22: Hollow particle reinforced composite - Neural Network prediction for 1000 total samples

| Neural Network | Random Split 1 | Random Split 2 | Random Split 3 |
|----------------|----------------|----------------|----------------|
|                | Ring MER      | Matrix MER     | MSE            | Ring MER      | Matrix MER     | MSE            | Ring MER      | Matrix MER     | MSE            |
| Stress-Net     | 2.29%         | 5.26%         | 1.94           | 2.97%         | 5.62%         | 2.08           | 3.51%         | 8.89%         | 2.57           |
| DNN            | 2.11%         | 3.73%         | 2.73           | 3.36%         | 5.52%         | 3.05           | 1.91%         | 5.21%         | 2.20           |
| DNN-N          | 2.85%         | 6.38%         | 5.26           | 2.92%         | 6.73%         | 5.47           | 2.90%         | 6.15%         | 5.36           |
| DNN-NC         | 1.74%         | 1.79%         | 0.25           | 1.54%         | 1.58%         | 0.25           | 1.96%         | 1.82%         | 0.28           |

Table C.23: Hollow particle reinforced composite - Neural Network prediction for 2000 total samples

| Neural Network | Random Split 1 | Random Split 2 | Random Split 3 |
|----------------|----------------|----------------|----------------|
|                | Ring MER      | Matrix MER     | MSE            | Ring MER      | Matrix MER     | MSE            | Ring MER      | Matrix MER     | MSE            |
| Stress-Net     | 2.22%         | 5.15%         | 1.68           | 4.73%         | 6.85%         | 2.17           | 2.94%         | 6.30%         | 2.08           |
| DNN            | 3.30%         | 4.11%         | 2.15           | 3.78%         | 6.30%         | 2.91           | 4.38%         | 5.82%         | 2.62           |
| DNN-N          | 3.82%         | 5.00%         | 3.37           | 4.35%         | 6.75%         | 4.13           | 5.39%         | 5.92%         | 3.44           |
| DNN-NC         | 2.00%         | 2.05%         | 0.21           | 2.32%         | 2.00%         | 0.24           | 2.30%         | 3.43%         | 0.26           |