NASA/TM-20230005410

Prediction of Stiffness and Fatigue Lives of Polymer Matrix Composite Laminates Using Artificial Neural Networks

Subodh K. Mital
The University of Toledo, Toledo, Ohio

Steven M. Arnold, Pappu L.N. Murthy, and Brandon L. Hearley
Glenn Research Center, Cleveland, Ohio

May 2023

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., "quick-release" reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 757-864-6500

- Telephone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Program
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM-20230005410

Prediction of Stiffness and Fatigue Lives of Polymer Matrix Composite Laminates Using Artificial Neural Networks

Subodh K. Mital
The University of Toledo, Toledo, Ohio

Steven M. Arnold, Pappu L.N. Murthy, and Brandon L. Hearley
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

May 2023

Acknowledgments

Primary development was funded through the NASA Aeronautics Research Mission Directorate's (ARMD's) Transformational Tools and Technologies (TTT) Project. Thanks also to Josh Stuckner, Multiscale and Multiphysics branch of NASA Glenn Research Center for many helpful discussions.

Level of Review: This material has been technically reviewed by technical management.

This report is available in electronic form at https://www.sti.nasa.gov/ and https://ntrs.nasa.gov/

Prediction of Stiffness and Fatigue Lives of Polymer Matrix Composite Laminates Using Artificial Neural Networks

Subodh K. Mital
The University of Toledo
Toledo, Ohio 43606

Steven M. Arnold, Pappu L.N. Murthy, and Brandon L. Hearley
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

Machine learning (ML) models are increasingly being used in many engineering fields due to the advancements in ML algorithms and availability of high-speed computing power. One of the most popular ML class of models is artificial neural networks (ANN). ML is increasingly being used in the design and analysis of composite materials and structures, specifically in the constitutive modeling of composite materials with the focus on greatly accelerating multiscale analyses of composite materials and structures through development of surrogate models. Towards that end, both Python and MATLAB-based neural nets have been developed to predict initial stiffness and fatigue life of an eight-ply symmetric polymer matrix composite laminate. Two types of neural networks, a Multilayer Perceptron (MLP) and a Recurrent Neural Network (RNN), have been developed for both platforms. Results show that the both neural net types can provide an excellent estimate of initial stiffness as well as fatigue life of eight-ply symmetric polymer matrix composite laminate. RNNs are better able to capture the shape of the fatigue curve of a laminate. This tool can be very useful for system level studies to obtain an estimate of desired properties and life of PMC composite laminates. The associated surrogate models could also be used in composite multiscale analyses to replace the actual physics-based calculations at lower scales and thereby significantly increase the computational efficiency of such analyses and thus make multiscale analyses a viable industrial tool for large scale structural problems.

Introduction

In recent years, the phraseology, artificial intelligence (AI), machine learning (ML) and deep learning artificial neural network (ANN) appear in countless articles with a promise of achieving self-driving cars, intelligent chatbots, and virtual assistants, etc. (Ref. 1). ANN are a subset of ML, which itself is a subset of AI. AI can be thought of as a process that tries to automate intellectual tasks that are normally expected to be performed by humans. As mentioned, AI is much more than ML. Previously, research was done in symbolic AI, which does not involve any learning. Previously, researchers believed that by programming a sufficiently large number of explicit rules, human level intelligence could be achieved (Ref. 2). Such an approach seemed to work to solve logical problems such as playing chess, however, it proved insufficient in solving complex problems such as image classification, speech recognition, language translation etc. which resulted in developing new approaches now known as ML.

ML started to flourish in the 1990s to become the most popular and most successful subfield of AI, driven by the availability of faster computers and larger datasets. ML gives computers the ability to learn without being explicitly programmed. It explores the study and construction of algorithms that can learn

from and make predictions on data by making data driven decisions and predictions without any specific static programming instructions. Today, machine learning is being employed in a wide range of computing tasks where designing and programming explicit algorithms is infeasible (Ref. 3).

The term deep learning is also mentioned in relation to machine learning and neural networks. Broadly stating, in machine learning, the relationships between the data are formed by the learning system. Data is input along with the results related to that input which is often obtained from physical models. The machine learning system maps the data to the results and comes up with rules that become part of the system. When new data is introduced, it can predict new results that were not part of the training sets.

Deep learning refers to systems with more than one layer of neurons between the input and output layers and associated weights applied to each neuron. It has been observed that multi-layer networks can learn and interpret relations that single layer networks cannot learn. The elements of a neural network are nodes or neurons, where weighted signals are combined, and biases added. In a single layer, the inputs are multiplied by weights and then added with a bias, before passing through a non-linear activation function. In a multi-layer or "deep learning" network, the inputs are combined in the second layer before being output (Ref. 4). It is highly linked to optimization techniques as it's trying to find the most optimum weights for each node that will minimize the error between the prediction and the target result. Figure 1 shows the anatomy of a deep learning neural network. It is also worth mentioning that previously developing deep learning neural networks required significant expertise in computer science that few people possessed. Nowadays, simple Python scripting knowledge is sufficient to develop deep learning artificial neural networks. So, in that sense, this process has been highly democratized.

Another fast-growing discipline with emphasis on reducing the cost and time to market of new materials is Integrated Computational Materials Engineering (ICME). ICME is an integrated approach to the design of products and the materials which comprise them by linking multiple models at different time and length scales (Refs. 5 and 6). At present, such analyses that link atomistic scale models to structural scale models are extremely resource intensive, often requiring the use of high-performance computing platforms. If some of these analyses at some scales could be replaced by artificial neural nets, it has the potential to greatly enhance the computational efficiency without sacrificing the accuracy required.

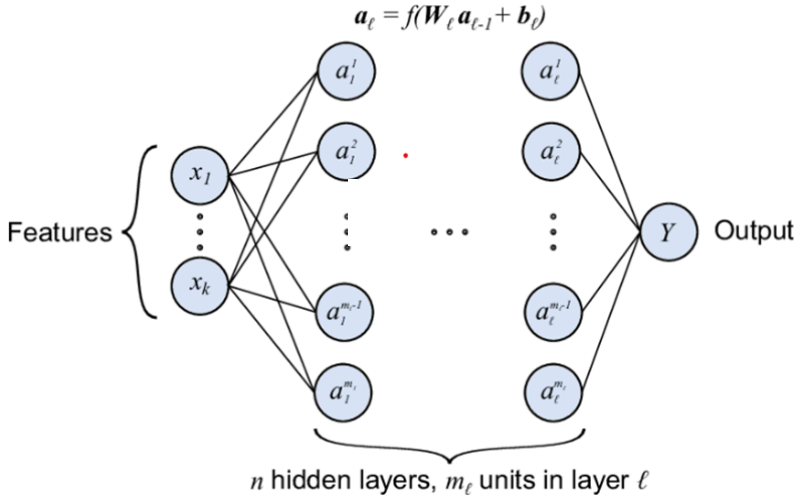


Figure 1.—Anatomy of a deep learning neural network.

ML is increasingly being used in the design and analysis of composite materials and structures, specifically in the constitutive modeling of composite materials with the focus on greatly accelerating multiscale analyses of composite materials and structures (Refs. 7, 8, and 9). Towards that end, our objective is to provide designers a Fatigue Life estimation tool, based on MATLAB or Python-based ANN, that can rapidly predict the initial stiffness (ABD matrices) and fatigue lives (S-N Curve) of polymer matrix composite (PMC) laminates. Since ML models typically require large amounts of data for training and validation and this quantity of measured data is not readily available, synthetic (virtual/simulated) data is generated using NASA's MAC/GMC (Micromechanics Analysis Code using Generalized Method of Cells) computer code (Ref. 10). MAC/GMC is a comprehensive, physics-based, composite material and laminate analysis tool that utilizes the method of cells family (MOC, GMC and HFGMC) of micromechanics theories (Refs. 11 and 12). These theories provide access to not only the effective composite response but the local (in-situ constituent) stress and strain fields in the composite material that are crucial for assessing damage initiation and progression in composite structures. MAC/GMC provides a user-friendly framework, in which a user can access a library of local inelastic, damage, and failure models as well as various microstructures, idealized as repeating unit cells (RUCs). The current ANN model has focused on predicting initial laminate stiffnesses and fatigue life of thermoelastic eight-ply, symmetric, PMC laminates subjected to stiffness reduction progressive cyclic damage. Results show that the trained surrogate ML model provides reasonable estimates of the desired composite behavior for a fraction (10^{-4}) of the computational cost of the corresponding physics-based model. Consequently, such an approach will lead to efficient, robust, and accurate data-driven design and analysis of composite materials and structures.

Overview of MAC/GMC Computer Code

GMC, first developed by Paley and Aboudi (Ref. 13) and HFGMC, first developed by Aboudi et al. (Ref. 14), are semi-analytical in nature, and their formulation involves application of several governing conditions (e.g., traction and displacement compatibility) in an average sense – within the RUC. They provide the local fields in composite materials, allowing incorporation of arbitrary inelastic constitutive models with various deformation and damage constitutive laws. The microstructure of a periodic multiphase material, within the context of GMC and HFGMC, is represented by a doubly periodic (continuously reinforced) or triply periodic (discontinuously reinforced) RUC consisting of an arbitrary number of subcells, each of which may be a distinct material (Figure 2). In the case of GMC, the displacement field is assumed linear, whereas in the case of HFGMC the displacement approximations are assumed quadratic, thus leading to a constant and linear subcell strain field, respectively. In fact, it is precisely this higher order assumption in the displacement field that enables HFGMC to retain its ability to compute nonzero transverse shear stress distributions within the composite (i.e., normal and shear coupling) when global tensile loading is applied. This shear coupling is very important when dealing with disordered microstructures (Refs. 15 and 16). However, it is also this high-order field assumption which makes HFGMC more computationally expensive and subject to subcell discretization dependence as compared to GMC.

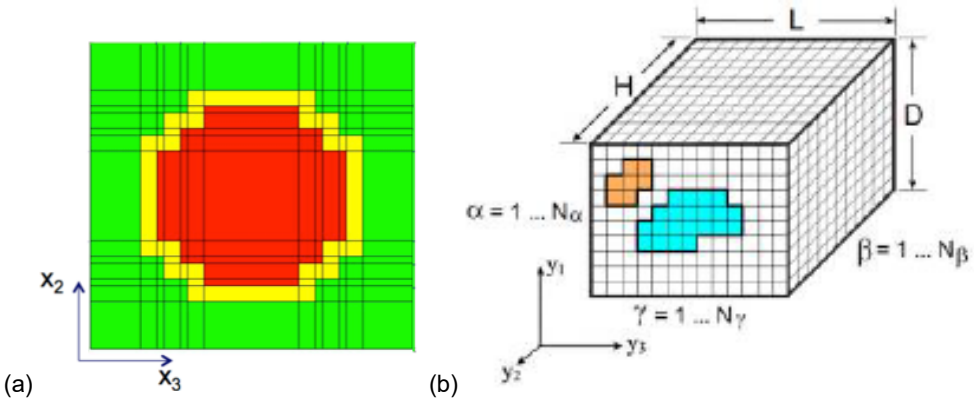


Figure 2.—Composite with repeating microstructure and arbitrary constituents. (a) Doubly periodic. (b) Triply periodic.

Displacement and traction continuity are enforced in an average, or integral sense at each of the subcell interfaces and the periodic boundaries of the RUC. These continuity conditions are used to formulate a strain concentration matrix **A,** which gives all the local subcell strains ($\boldsymbol{\epsilon_S}$) in terms of the global, average, applied strains $\boldsymbol{\epsilon_{applied}}$ (i.e., $\boldsymbol{\epsilon_S} = \mathbf{A}\ \boldsymbol{\epsilon_{applied}}$). The local subcell stresses ($\boldsymbol{\sigma}$) can then be calculated using the local constitutive law and the local subcell strains. Finally, the overall RUC stiffness is obtained utilizing the local constitutive law and the strain concentration matrix averaged over the RUC dimensions. The detailed methodology of GMC and HFGMC and the formulation to be embedded within classical laminate theory is described thoroughly in Aboudi et al. (Refs. 11 and 12). Also, in these references the superior accuracy of HFGMC over that of GMC is demonstrated, consequently in this study HFGMC will be assumed to provide the most accurate predictions.

Constitutive Models

The most well-known and widely used constitutive model, Hooke's law, is written as

$$\sigma_{ij} = C_{ijkl}\,\varepsilon_{kl} \tag{1}$$

which describes time-independent, linear (proportional) reversible material behavior, where C_{ijkl} is the classic stiffness tensor and ε_{kl} is the elastic component of the strain tensor. Extension into the irreversible regime has been accomplished by assuming an additive decomposition of the total strain tensor into three components, that is a reversible mechanical strain (i.e., elastic/viscoelastic) ε_{ij}; an irreversible (i.e., inelastic or viscoplastic) strain ε_{ij}^{I}; and a reversible thermal strain, ε_{ij}^{th} component.

$$\varepsilon_{ij}^{total} = \varepsilon_{ij} + \varepsilon_{ij}^{I} + \varepsilon_{ij}^{th} \tag{2}$$

or

$$\varepsilon_{ij} = \varepsilon_{ij}^{total} - \varepsilon_{ij}^{I} - \varepsilon_{ij}^{th} \tag{3}$$

After substituting expression (3) into Equation (1) we arrive at a stress strain relation (generalized Hooke's law) that incorporates irreversible strains as well as reversible ones, that is:

$$\sigma_{ij} = C_{ijkl}(\varepsilon_{kl} - \varepsilon_{kl}^{I} - \varepsilon_{kl}^{th}) \tag{4}$$

where numerous models describing the evolution of the inelastic strain have been proposed in the literature (Refs. 17 to 19). Herein we will confine this initial study to elastic constituent materials only.

Continuum Fatigue Damage Model

The fatigue life of the composite will be predicted utilizing micromechanics and the multiaxial, isothermal, continuum damage mechanics model of Arnold and Kruch (Ref. 20) for the matrix constituent. When reduced to its isotropic form (parameters ω_u, ω_{fl}, ω_m, η_u, η_{fl}, and η_m are set equal to one) this model reduces to the Non-Linear Cumulative Damage Rule (NLCDR) developed at ONERA (Ref. 21). This model assumes a single scalar internal damage variable, *D*, that has a value of zero for undamaged material and one for a completely damaged (failed) material. The implementation of the damage model within GMC and HFGMC has been performed on the local scale, thus damage evolves in each subcell based on its local stress state and number of cycles. For a given damage level, the *stiffness of the subcell* is degraded by (1 – *D*). Further, the implementation allows the application of a local damage increment ΔD, and then calculates the number of cycles, *N*, required to achieve this local increment of damage. This approach allows the model to determine the stress state in the composite, identify the controlling subcell that will reach the desired damage level in the fewest cycles, apply that number of cycles to all subcells, and calculate the damage that arises throughout the remainder of the composite. Then the composite can be reanalyzed, and a new stress state determined based on the new, spatially varying, damage level throughout the composite RUC. In this way, the local and global stress and damage analyses are coupled. As the damage in the composite evolves, the stress field in the composite is redistributed, which then affects the evolution of damage.

For an isotropic material, the damage parameters that must be selected reduce to *M*, β and *a*, while the pertinent equation relating the fatigue life of the isotropic material to the cyclic stress state is,

$$N_F = \frac{(\sigma_u - \sigma_{\max})\left(\dfrac{M}{\sigma_{\max} - \bar{\sigma}}\right)^{\beta}}{\hat{a}(1+\beta)\left(\sigma_{\max} - \bar{\sigma} - \sigma_{fl}\right)} \quad \text{for } N_F > 0 \tag{5}$$

where σ_u is the material ultimate strength, σ_{fl} is the material fatigue limit (stress below which damage does not occur), σ_{max} is the maximum stress during a loading cycle, $\bar{\sigma}$ is the mean stress during a loading cycle, and N_F is the number of cycles to failure. Note that, in the terminology of Arnold and Kruch (Ref. 20), $\hat{a} = a\frac{\sigma_{fl}}{\sigma_u}$. Utilizing Equation (5), the damage model parameters *M*, β, and *a* can be selected for an isotropic material based on the material's S-N curve (stress level vs. cycles to failure). Both the fatigue limit and the scaling parameter M are general enough to account for the effect of mean stress. However, in this study this additional effect is ignored since only one R ratio (R = –1, fully reversed) is examined. A representative S-N curve for an epoxy matrix was obtained, and the corresponding fatigue damage model parameters were found to be *M* = 150 MPa, β = 9, and a = 0.05, with σ_u = 80 MPa, and σ_{fl} = 27.0 MPa. A plot showing the resulting matrix S-N curve is given in Figure 3.

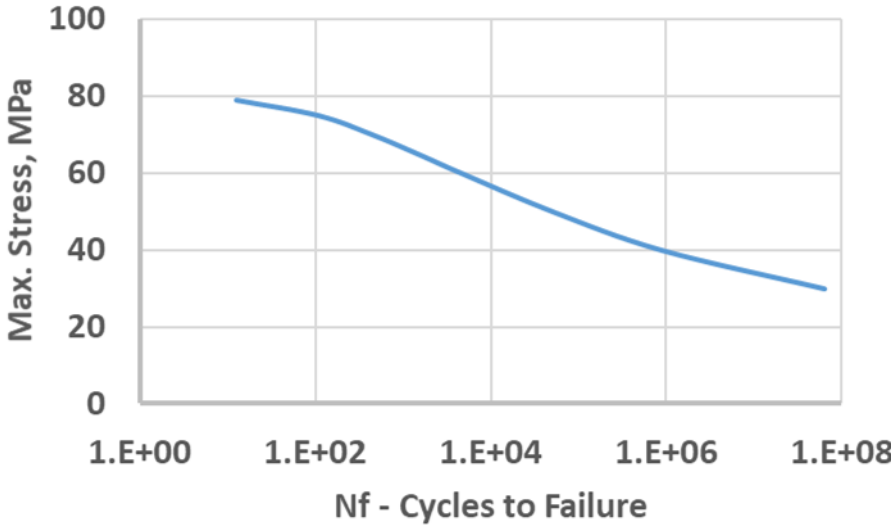


Figure 3.—*Stiffness reduction* fatigue damage model representation for epoxy matrix.

A second damage model within GMC and HFGMC is much simpler and involves degradation of a material's strength due to cyclic loading. As shown by Wilt et al. (Ref. 22), this type of damage model can be used to simulate the fatigue behavior of fibers that occurs in-situ during fatigue of a composite. The model assumes a logarithmic relation between the material's strength and the number of cycles within a certain range such that:

$$\begin{aligned} \sigma_u &= \sigma_{u1} && 0 \le N \le N_1 \\ \sigma_u &= \sigma_{u1} \frac{(\sigma_{u1} - \sigma_{u2}) \log(N/N_1)}{\log(N_2/N_1)} && N_1 \le N \le N_2 \\ \sigma_u &= \sigma_{u2} && N_2 \le N \end{aligned} \tag{6}$$

This *strength degradation* model (Eq. (6)) was employed in the present example to model the longitudinal fatigue behavior of the graphite fiber. The necessary parameters for the model are σ_{u1}, σ_{u2}, N_1, and N_2, see Example 5d in (Ref. 23). The values of these parameters chosen for a graphite fiber are shown in Figure 4. Note that these data were not correlated with experiment, but rather chosen based on the expected trend. Given these required parameters for the fatigue damage models for each phase in the PMC, macroscopic or composite fatigue life of both unidirectional and laminate composites can be simulated. Note although creep-fatigue interaction can be incorporated in the above theory, see Reference 24, it is not included in the present calculations.

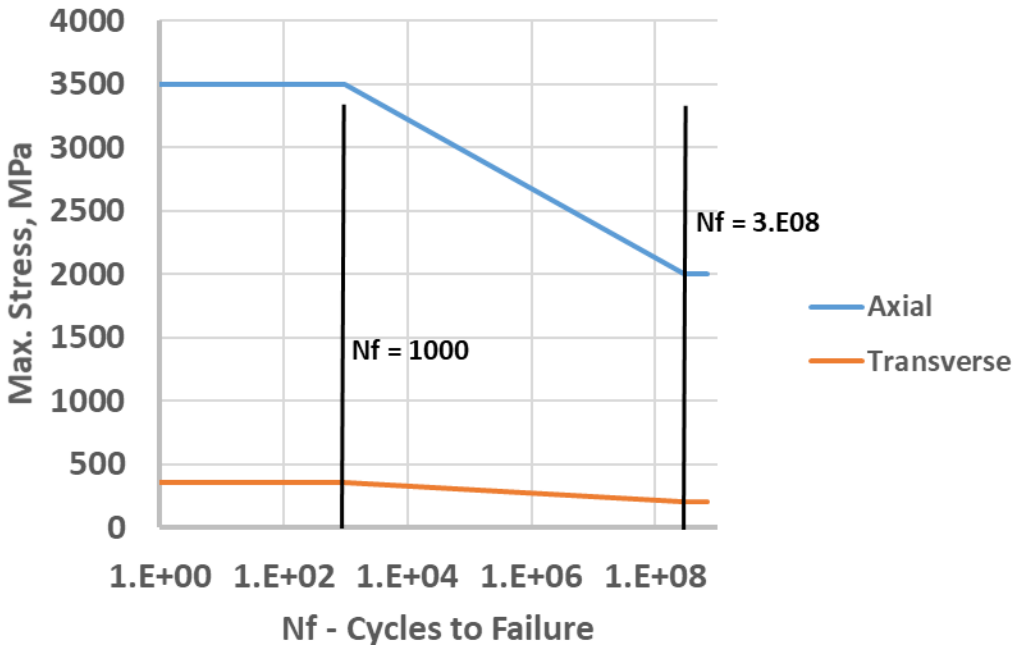


Figure 4.—Strength reduction fatigue model parameters assumed for the graphite fiber.

Neural Networks

There are different types of deep learning neural networks that have applications in different areas (Ref. 25). In general, they work well with large amounts of data and have multiple layers to learn the input/output relationships accurately. Generally, the anatomy of a neural network can be described by the following objects – (a). Layers, which are combined into a network, each containing a distinct number of neurons, (b) Input data and the corresponding targets, which is divided into training, validation, and test sets, (c) the loss function, which defines the feedback signal used for training, most common being mean-squared error (MSE), and (d) Optimizer, which really determines how learning proceeds and finds the most optimal weights for each neuron that will minimize the error. It is worth mentioning that picking the right network architecture is more an art than science. However, overall architectural choices and types of networks (CNN, RNN, MLP etc.) can be determined by what is being modeled. There are some best practices and principles that one can rely upon, but only by trying different architectures, can one find a proper neural net architecture.

Although there are many types of neural nets, two types of deep learning neural nets have been investigated here – first is a Multi-Layer Neural Network, also referred to as Multi-Layer Perceptron (MLP) architecture and the second is a Recurrent Neural Network (RNN). MLP is the most common and basic deep learning neural network that consists of input neurons, multiple layers of hidden neurons, and output neurons. Although uncommon, different layers can have different activation functions or can even be different types of layers in functionality. They are fully connected feed-forward artificial neural network or sometimes loosely referred to as simply ANN. These are the most basic or “vanilla” type of deep learning neural networks. The other RNN is a type of recursive neural network which work on structured data (or data that has some functional relationship) and generally have time as the structuring element. RNN have a loop within that combines the previous time step’s data with the hidden or intermediate layer to represent the current time step. They can be useful in modeling sequences of data to predict things such as future earthquakes and stock market performance and are very useful in language translation, speech recognition and many other applications. In our case, for fatigue life prediction (S-N

curve) of polymer matrix composites, time is not the sequential element. Even though each stress-life pair are independent from each other yet are totally dependent upon the specific laminate defined, thus they exhibit a functional relationship between each point. Consequently, an RNN can potentially be useful in predicting the fatigue life curve (i.e., S-N curve) as it is a series of stress – life data, and thus it is a sequential series data. A brief description of various types of deep learning neural networks is provided in Reference 26.

Description of PMC Laminates and Data Generation

Neural nets inherently require a significant amount of training data depending upon the number of inputs and outputs. Typically, in material science the amount of experimental data available doesn't exist in sufficient quantity due to the expense involved in generating this data. Consequently, in this study, the physics-based GMC micromechanics method was used to generate the large amount of synthetic training data. HFGMC micromechanics (a more accurate but significantly more computationally expensive, by 2 to 3 orders of magnitude, compared with GMC) was also used to generate synthetic data as well. It was observed that for most laminates considered, there was no significant difference between GMC and HFGMC predictions particularly for fatigue lives. Stiffness matrices [A] and [D] ([B] =0 for symmetric laminates) and fatigue lives (S-N curve) were generated for eight-ply, symmetric, regular[1] PMC laminates with varying fiber volume fractions as well as varying constituents (fiber and matrix) material properties. A 7x7 RUC, which is available within the MAC/GMC's internal library of repeating unit cells (Ref. 10), was used to represent the microstructure of the composite as shown in Figure 5. Fatigue lives were predicted under a fully reversible cyclic uniaxial load applied in the global X direction. Fiber volume fractions and constituent material properties were randomly generated between some pre-defined ranges and following all applicable laws of material behavior (see Table 1, first two columns for the limits used). Twelve input values consisting of ply angles, fiber volume ratio, fiber and matrix stiffness-related properties are needed to predict the [A] and [D] matrices for the resulting laminate. To predict fatigue life, 7 additional fiber and matrix fatigue parameters (see Table 1) are required as well as N1, N2, and "a" (which were held constant at 1,000, 10^9, and 0.5, respectively) in running the MAC/GMC computer code. Note a reduction in the number of input parameters from 26 (5 for RUC/laminate, 5 for fiber elastic, 2 for matrix elastic, 8 for fiber fatigue, 6 for matrix fatigue) to 19 was achieved by assuming that the fiber strength parameters are related to each other via a Von Mises relationship (i.e., $\sqrt{3}$), see Table 2. In the development of surrogate model(s), various neural net methods contained within two different toolsets were considered: 1) the neural net formulations available in Python scripting environment (Ref. 27) and 2) the Machine learning toolbox that is available within the MATLAB programming environment (Ref. 28). Note it is also essential that an informatics infrastructure be utilized to store not only the data (albeit virtual or real) but also more importantly the meta data for both data and ML model(s) to maximize traceability and minimize misuse, see Reference 29. Finally, the applied load is also required.

The stress versus cycles fatigue curve (i.e., S-N curve) and [A] and [D] matrices of approximately 10,000 different laminates were computed and stored as the synthetic data. They were divided into training (80 percent), validation (10 percent), and test data (10 percent). It was also noticed that when training either the conventional MLP neural net or the RNN, training is much better for fatigue prediction when the data covers the full range of the fatigue curve (life between 1,000 and 1×10^9 cycles). However, the MAC/GMC code is designed in such a way that the user is required to specify the fatigue load so that the code will predict the number of cycles to failure corresponding to that load. Therefore, it is difficult to

[1] Each ply of the laminate has the same thickness.

know a priori, the load range that will result in an S-N curve covering the entire fatigue life range (between 1,000 and 1×10^9 cycles). In practice, it becomes an iterative procedure to find that load range for a specific laminate. To overcome this issue, a MATLAB optimization script (see Appendix A) was written that can predict the fatigue loads (upper limit and lower limit) at lifetimes of 1,000 and 1×10^9 cycles for a given laminate. Once those limits were computed that range (upper – lower) was divided into 10 segments and the whole S-N curve was predicted for a given laminate. To facilitate the prediction of the ABD matrices and the full range of the S-N curve a Python GUI and MATLAB GUI was developed, see Appendix B and C respectively.

TABLE 1.—DATA GENERATION: INPUT PARAMETER RANGES

Property	Upper	Lower
RUC/Laminate properties		
FVR	0.4	0.7
Θ_1	–90	90
Θ_2	–90	90
Θ_3	–90	90
Θ_4	–90	90
Constituent deformation properties		
Fiber		
Efa, GPa	70	700
Eft, GPa	70	200
νfa	0.2	0.4
νfa	0.2	0.4
Gfa, GPa	25	200
Matrix		
Em, GPa	2.5	4.5
νm	0.2	0.45
Constituent fatigue parameters		
Fiber		
SU11, MPa	2,500	9,500
SU21, MPa	300	2,500
Matrix		
Epsm1	0.015	0.03
Epsm2	0.015	0.03
β	4	12
SFL, MPa	15	30
XML, MPa	80	160

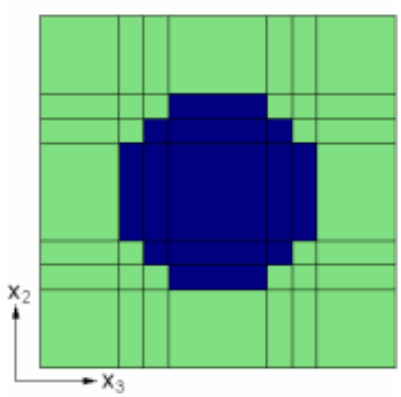


Figure 5.—7x7 RUC used for data generation.

TABLE 2.—IMPLICIT RELATIONSHIP BETWEEN FIBER PROPERTIES AND MATRIX PROPERTIES

Fiber properties:	**Matrix properties:**
❑ Et ≤ Ea ❑ Et/(2(1+vt))≤ Ga ≤ Ea/(2(1+va)) ❑ SU12 = SU13 = (SU11/Ea)*Et ❑ SU14 = SU12/sqrt(3) ❑ SU15 = SU16 = SU11/sqrt(3) ❑ SU21 < SU11 ❑ SU22 = SU23 = (SU21/Ea) * Et ❑ SU24 = SU22/sqrt(3) ❑ SU25 = SU26 = SU21/sqrt(3) ❑ X11=SU11, X22=SU12, X33=SU13, X23=SU14, X12=SU15 X13=SU16	❑ Su = Em * Epsm1 (failure is based on strain and is assumed linear until failure)

Results

Two types of neural nets MLP and RNN were developed to predict the laminate stiffness and fatigue curves. Both types of networks provided results with excellent accuracy and speed. Both the neural net formulations available in Python scripting environment and those available in the AI toolbox within the MATLAB programming environment where investigated. Both approaches gave very similar results but early in the development of neural nets surrogate models, having both results from both toolsets (Python and MATLAB) provided a good check. Following results are generally based on what was obtained using the Python scripting environment for neural net development.

Stiffness Prediction

A standard MLP neural net was trained to predict the stiffness (i.e., A and D matrices) for an eight-ply symmetric laminate (i.e., B = 0). There are 12 input parameters – four angles, five fiber elastic properties for a transversely isotropic fiber (axial and transvers modulus, axial and transverse Poisson's ratios, axial shear modulus) and two matrix elastic properties for an isotropic matrix (modulus, Poisson's ratio). There were 12 output properties – six components associated with the A matrix (3×3 symmetric matrix) and six components of the D matrix (also a 3×3 symmetric matrix). The specific neural net for stiffness prediction had five hidden layers and 26 nodes in each layer. The mean square error (MSE) for stiffness prediction was 0.02. Figure 6 shows the comparison between the ANN stiffness prediction of a random laminate, $[–17°/35°/–72°/5°]_S$, with that produced from MAC/GMC (labelled as actual in Figure 6). Results show excellent agreement between the predicted and actual values. It should be noted that when the target values are relatively small (e.g., very close to zero), somewhat amplified errors can occur.

Laminate Stiffnesses

	Predicted	Actual	% Error
A11	2.7824e+05	2.7839e+05	0.06
A12	55536	55654	0.21
A13	13624	13660	0.26
A22	1.3209e+05	1.3204e+05	-0.04
A23	-15535	-15488	-0.3
A33	50057	50008	-0.1
D11	1.0162e+05	1.0178e+05	0.16
D12	21948	21955	0.03
D13	-8392.7	-8378.8	-0.17
D22	28278	28220	-0.2
D23	673.36	707.9	4.88
D33	20118	20073	-0.22

Figure 6.—Comparison of NN (predicted) and GMC (actual) output for a random laminate, $[-17°/35°/-72°/5°]_S$.

Development of a neural net for stiffness prediction of an eight-ply symmetric polymer matrix composite laminate with random ply angles turned out to be a rather straight-forward process.

Prediction of Fatigue (S-N) Curve

Training of a neural net to predict the fatigue curve for a given laminate configuration turned out to be a much more complex process. The ANN for the fatigue life prediction has added complexity where the number of inputs increase from 12 to 19 as the inputs for fatigue related properties must be considered as well, see Table 1 and Table 2. The calculation of fatigue life is where the development of ANN estimator shows its true benefit as physics-based model prediction of fatigue life can take anywhere from minutes to hours to compute the entire S-N curve, depending upon the laminate specification. Consequently, all fatigue synthetic data (training data instances) were run on a Linux cluster. The number of data sets that were generated were close to 100,000 (requiring an equivalent of 15,000 h of computation on a window personal computer). That amounted to analysing 10,000 laminate IDs with 10 load levels (points) per fatigue (S-N) curve. The results shown in the following sections were obtained from neural nets developed in the Python scripting environment. The neural nets developed in the MATLAB environment showed similar results.

As mentioned before, two different types of neural nets were trained to predict the fatigue life of a given laminate as described below.

Multi-Layer Perceptron (MLP) Network

This is also sometimes loosely referred to as simply an artificial neural network (ANN). This is the most common and the simplest form of a deep learning neural network. An ensemble deep learning neural net was employed. An ensemble approach is used to reduce the variance of neural network models by training multiple models instead of a single model and to combine the predictions from these models. This is called ensemble learning and it not only reduces the variance of predictions but also can result in predictions that are better than any single model. A schematic of an ensemble model is shown in Figure 7.

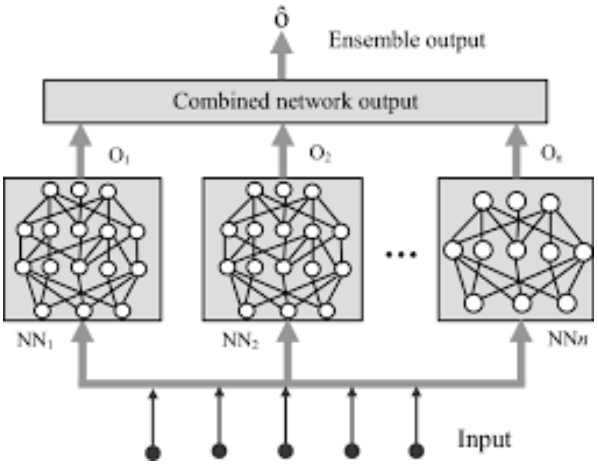


Figure 7.—Deep learning ensemble neural net.

Thirty submodels or neural nets were trained to predict the log of fatigue life (Log N) that is expected to be between 3 and 9 given all the inputs for the laminate including the fatigue load. Each individual neural net (submodel) consists of 6 hidden layers with 48 neurons in each layer. The mean square error (MSE) ranged from 0.25 to 0.32 for the 30 submodels. The output from these 30 submodels was then combined using a simple linear stacked model and the overall error (MSE) reduced to 0.104 with the mean absolute error (MAE) in Log N being 0.181.

The fatigue (S-N) curve for a laminate is defined by 10 points (stress vs. number of cycles to failure pairs). About 5,000 laminates (50,000 rows of data) was used for training and validation while the test data consisted of approximately 600 laminates (6,000 rows of data each). These laminates consisted of angles that were purely random as well as those with custom angles such as [0°], [90°], cross-ply [0°/90°], quasi-isotropic [0°/±4°5/90°], etc. Each laminate had a full range of stress values which provided number of cycles to failure ranging from approximately 1×10^3 to 1×10^9. Also, the training set consisted of 40 percent random and 60 percent custom laminates. When the trained neural net was tested on validation and test data, it showed that the

- Probability that a prediction will lie within ±10 percent log of target was 84.4 percent
- Probability that a prediction will lie within ±20 percent log of target was 94.6 percent
- Probability that a prediction will lie within ±30 percent log of target was 97.4 percent

Note that ±10 percent log (N) is approximately equivalent to ±300 percent of target N, which is within typical experimental error, which is in the range of 2 to 4 times.

Figure 8 shows the comparison between the neural net prediction of the fatigue curve (blue line) vs. the simulated (red line) curves using the physics-based model MAC/GMC, for 16 laminates with a mixture of custom and random laminates. It shows that the ANN predictions are a good estimate of the actual fatigue curves, with an MSE range of 0.01 to 1.27 for these 16 laminates. However, these predictions are made at a fraction of the cost (CPU time and effort) compared to the actual physics-based model (i.e., MAC/GMC). Note the worst error occurs for the random angled laminates.

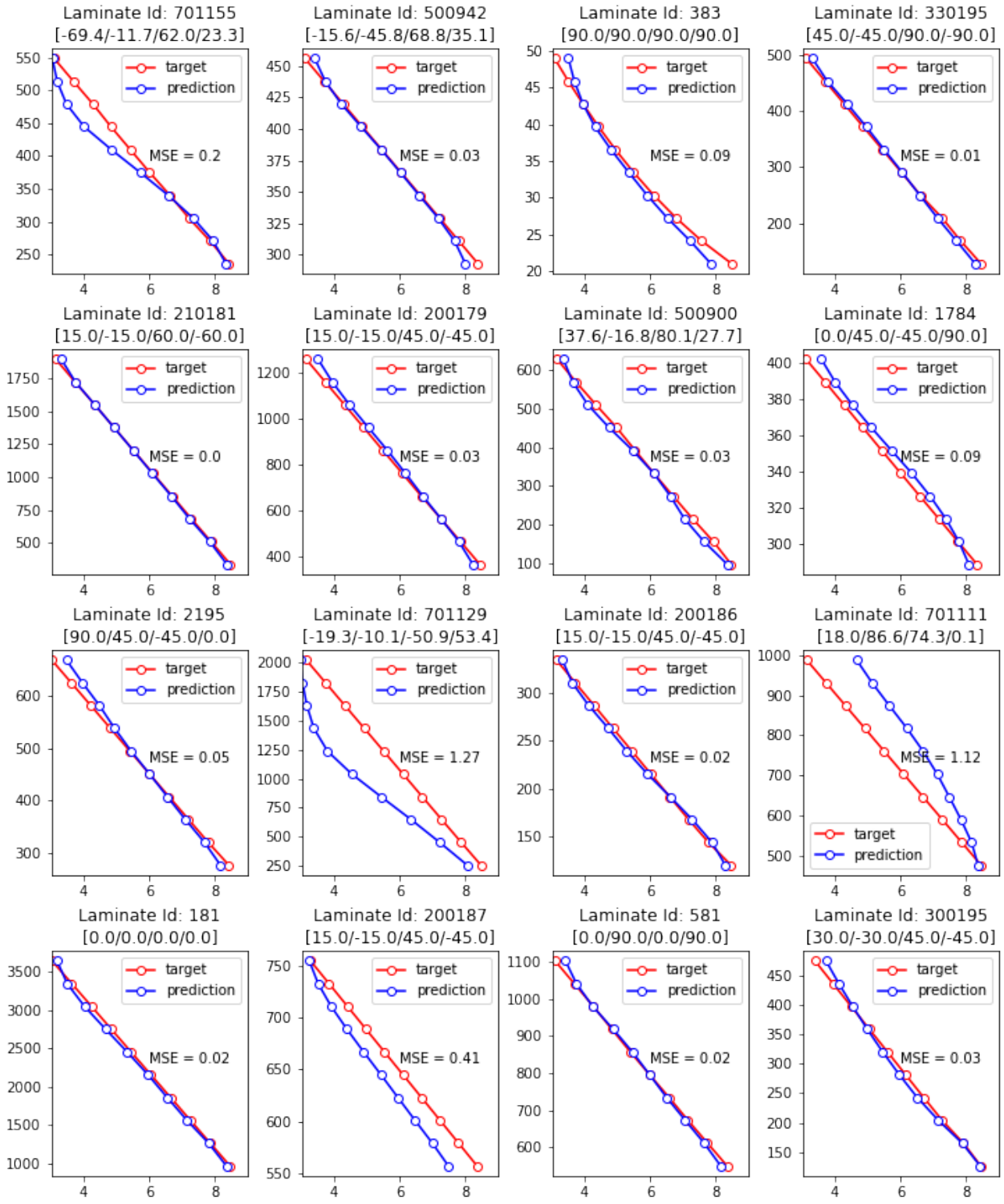


Figure 8.—Comparison of predicted and actual fatigue curves of random and custom laminates.

Although, results are not shown here, analyses have shown that if we fix the angles in the laminates and vary other parameters – like volume fraction, material properties and fatigue parameters, the neural nets can be trained to predict very accurate results with only a small amount of data (e.g., generally just with a few hundred laminates in the training set). Introduction of random ply angles in the laminates,

increases the complexity of the problem significantly, thus requiring larger amounts of training data. With this in mind, and to assess the error between custom and random; two sets of laminate S-N curves were predicted using the neural net and compared with the results from the actual physical model MAC/GMC. Material properties, volume fraction and fatigue parameters in both sets were held fixed with only the ply angles of the laminate varying. The first set, shown in Figure 9, consists of custom angles that are commonly used. The second set, shown in Figure 10, consists of random angles. The comparison in the case of custom angles with the target curves appear to have excellent agreement, in that the MSE range for these 12 laminates is between 0.01 to 0.34. However, for the case of 12 random angled laminates the predictions appear to be quite poor compared to the physics-based simulations, with an MSE range of 0.03 to 2.78.

Note that the actual benefit of this fatigue estimation tool is its ability to predict the custom angle laminates specifically, because in actual applications, it is unlikely anyone would use one of the random angle cases. Rather, the random angles are used here to cover the full design space and show robustness of the tool itself.

Recurrent Neural Network (RNN)

As mentioned before, a RNN was also trained using the same virtual data set even though our model and data do not fit the traditional definition of RNN. However, given a specific laminate (volume fraction, ply angles, constituent material properties) MAC/GMC predicts the fatigue life (number of cycles to failure) for a given fatigue load, thus providing a series of implicitly related points (S vs. N) for a given laminate. Previous work (Ref. 30) has shown similar success.

In this section, results obtained from Python based RNN are presented (MATLAB results were very similar). The RNN consisted of one dense and one LSTM (Long Short-Term Memory) layer with 20 neurons each, which was a relatively simple architecture. The MSE for fatigue life prediction was 0.01 and MAE was 0.06 (for log N) for validation cases, which is remarkably better than what was achieved using an MLP network. The training/validation data that was used to develop the MLP was also used here. The results on validation cases showed that the probability that the predicted life –

- Probability that a prediction will lie within ±5 percent log of target was 92 percent
- Probability that a prediction will lie within ±10 percent log of target was 98 percent
- Probability that a prediction will lie within ±10 percent log of target was 100 percent
- Probability that a prediction will lie within ±10 percent log of target was 100 percent

Note that ±5 percent log (N) is approximately equivalent to ±200 percent of target N, which is well within the typical experimental error of 2 to 4 times.

These are well within the scatter range that is observed in experimental data. The predicted and target S-N for 16 cases randomly selected from the validation set is shown in Figure 11. Results indicate that the RNN predictions are a very good estimate of the actual fatigue curves, with an MSE range of 0.0 to 0.08 for these 16 laminates. Again, the worst error being associated with the random angle case. Clearly, the RNN model provides overall significantly better results than what was observed using the MLP network.

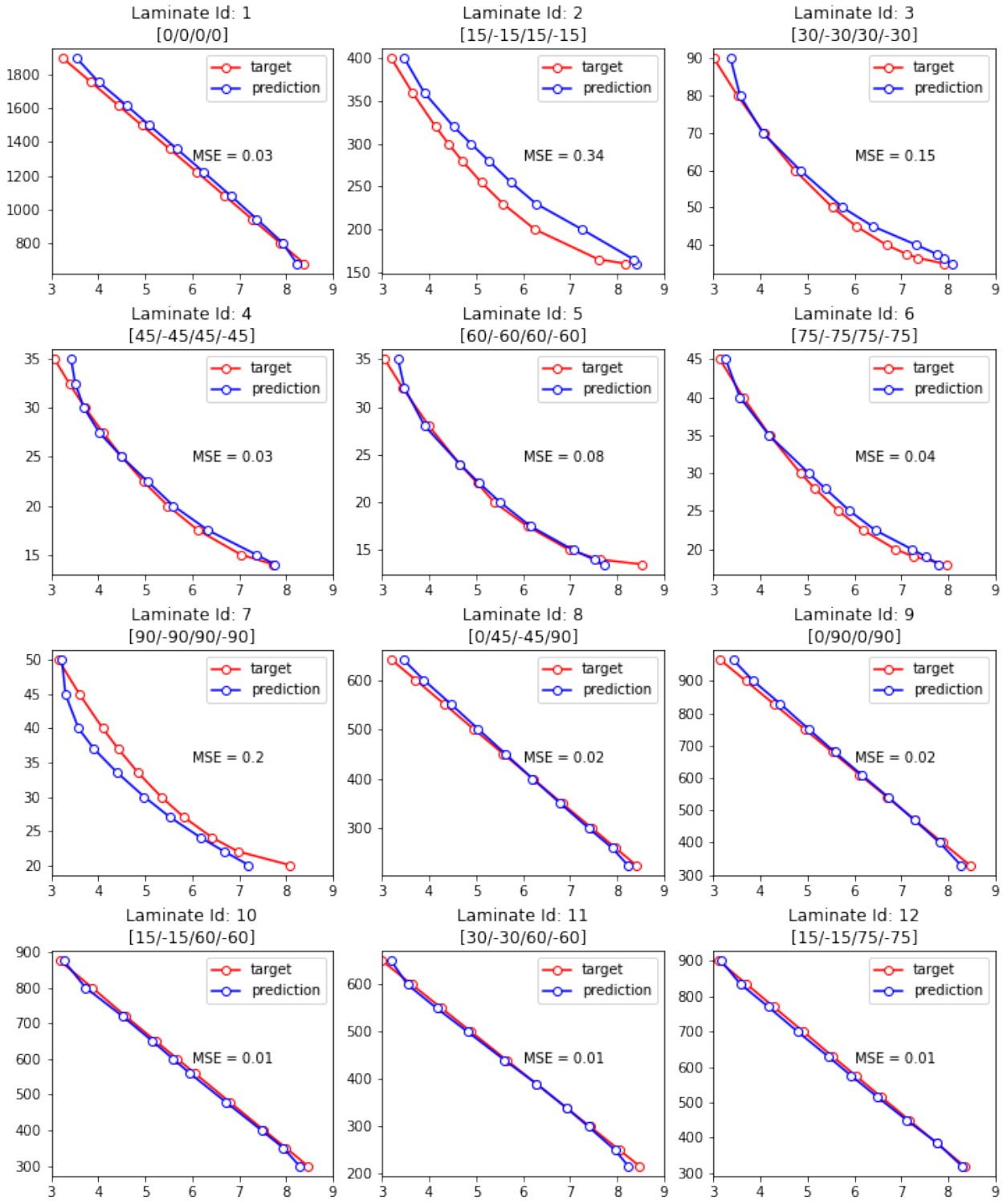


Figure 9.—Predicted and target fatigue curves of 12 selected laminates with custom angles.

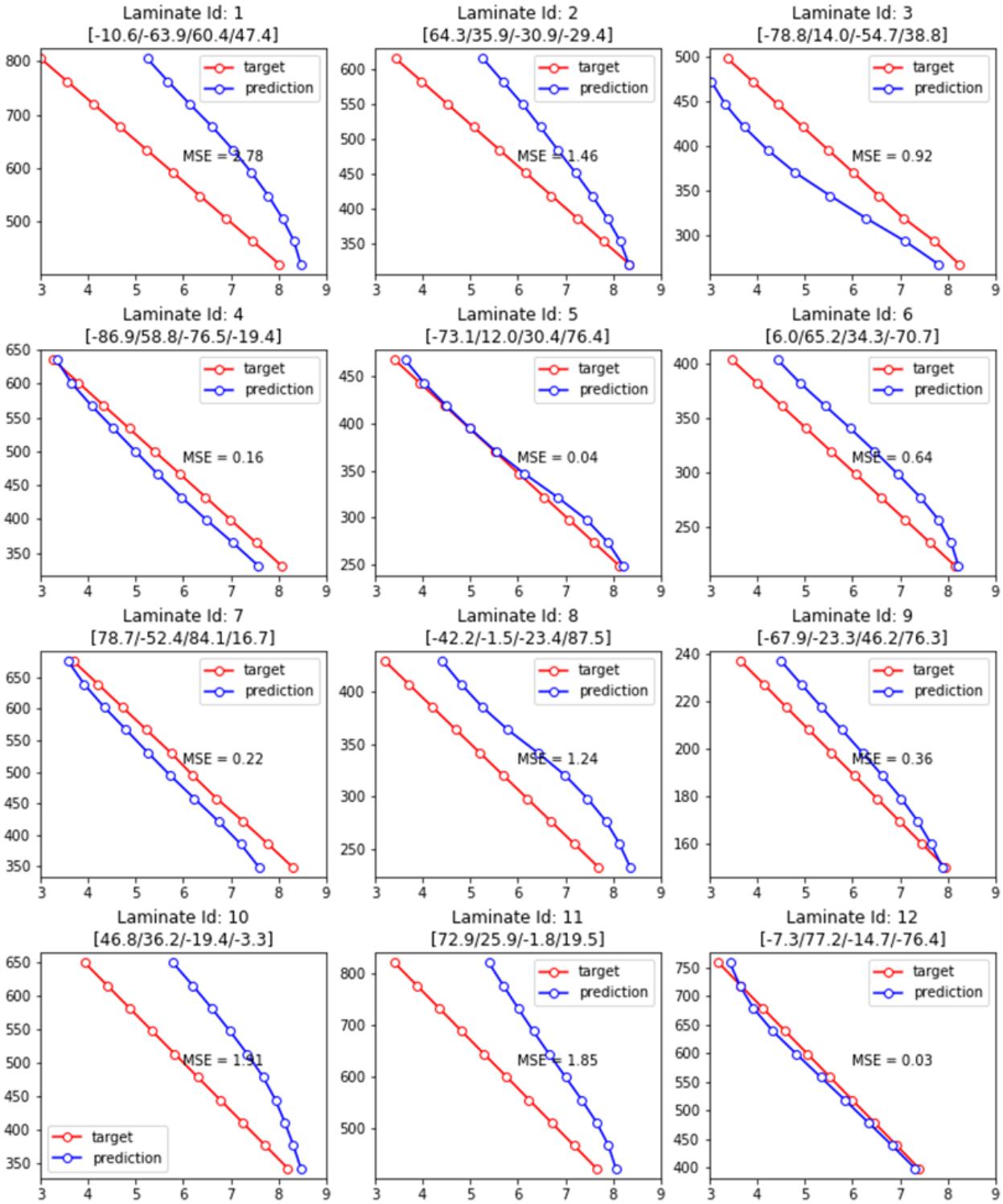


Figure 10.—Predicted and target fatigue curves of 12 selected laminates with random angles.

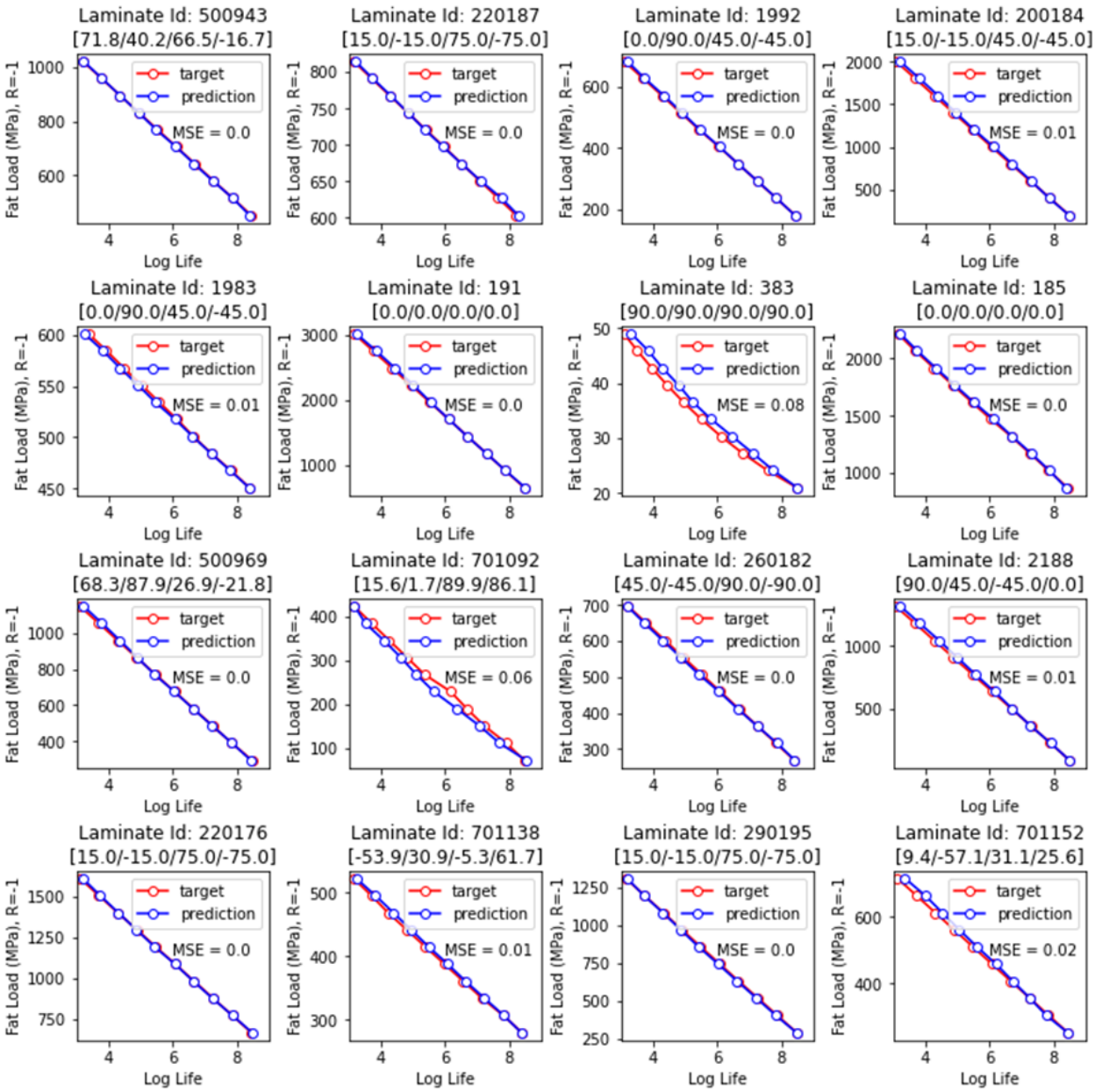


Figure 11.—Comparison of the Actual and Predicted S-N curves of Few Random Laminates.

Once again to assess the error between custom and random laminates; the same two sets of laminates examine for the MLP model are examined again. Figure 12 shows the comparison of the RNN prediction with the actual fatigue curves for the 12 custom angles previously shown in Figure 9. The accuracy for the custom angles appear to be in very good agreement, with an MSE range for these 12 laminates between 0.0 and 0.64. Comparing the same 12 random angled laminates used in Figure 10 but now predicted with the RNN model and shown in Figure 10 we see that the predictions appear to be excellent compared to the physics-based simulations, with an MSE range of 0.01 to 0.3. Also it appears that the RNN model is able to capture the endpoints of the S-N curves better than the MLP (particularly in Figure 12).

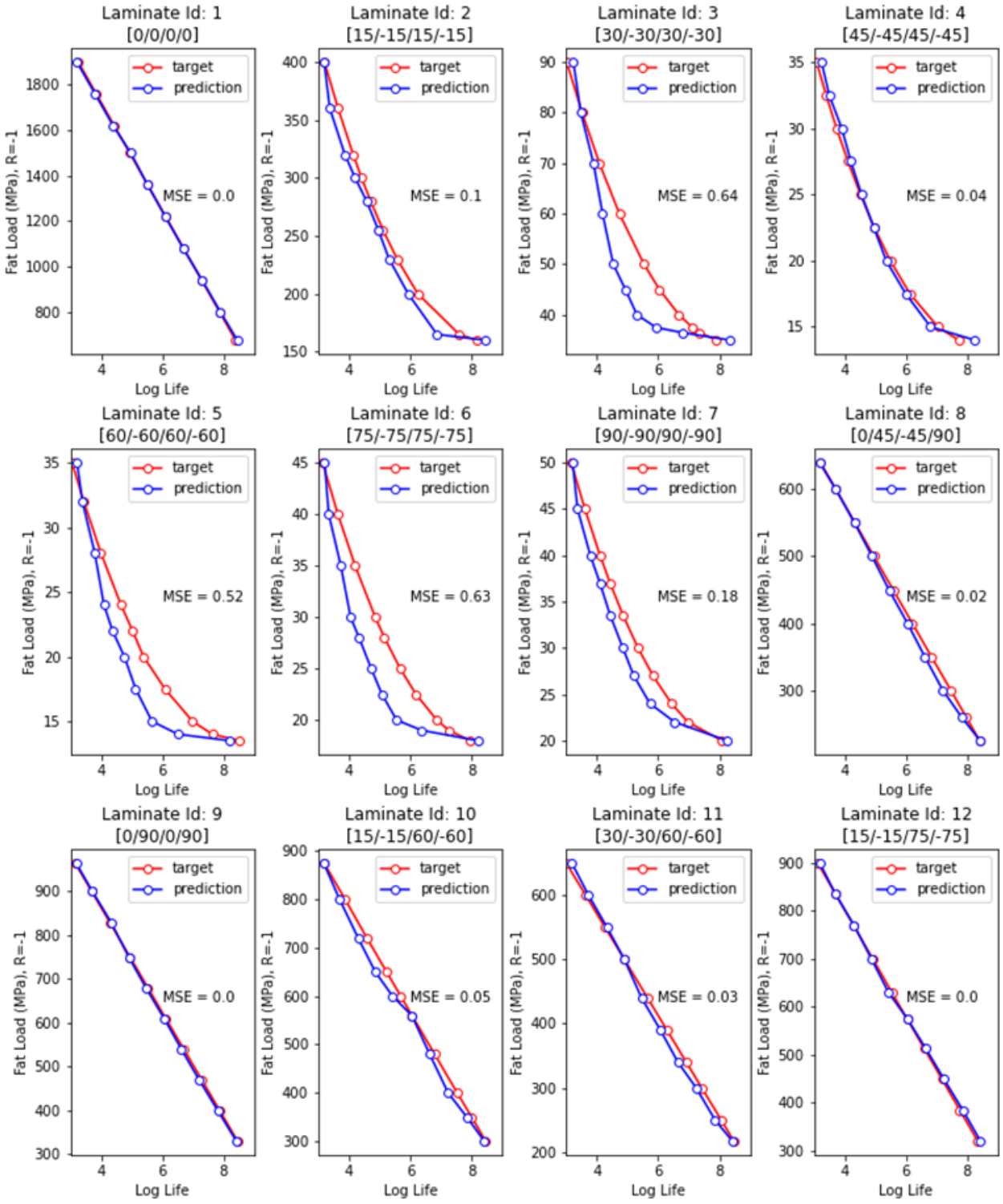


Figure 12.—Predicted and target fatigue curves of select laminates with custom angles.

Note this result is opposite to that observed in the case of MPL model where the random laminate predictions were substantially worst than the custom angled laminates. This prompted us to investigate the influence of the training data set mixture between custom and random angles. Table 3 and Table 4 show the results of this study. Results show that overall accuracy improves when laminates with random

angles are increseasd in training set. As expected, the prediction of laminates with random angles improves significantly while laminates with custom angles degrades slightly.

As can be seen, the RNN model fatigue life predictions are in general better. It is better able to capture the shape of the fatigue curve.

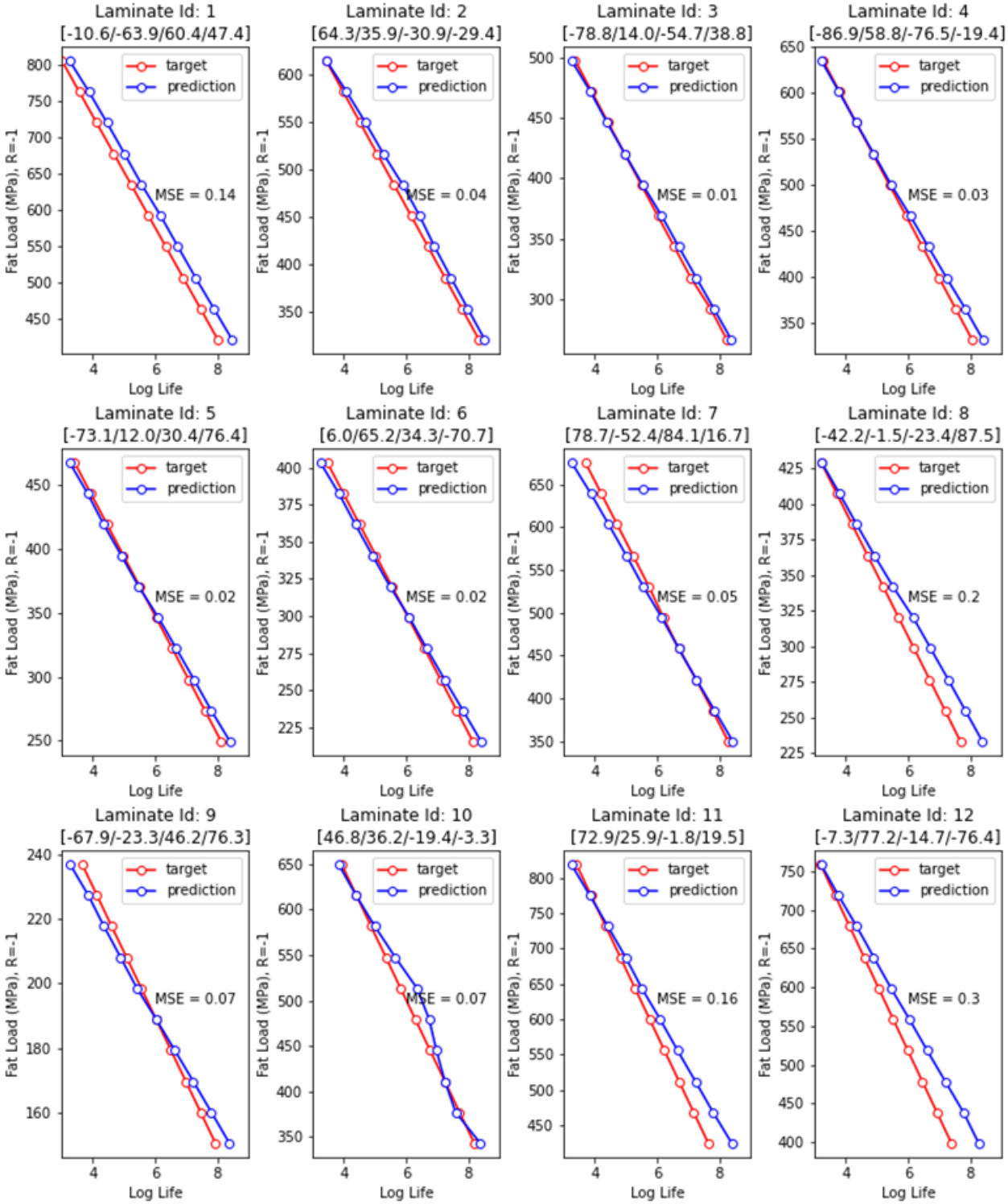


Figure 13.—Predicted and target fatigue curves of select laminates with random angles.

TABLE 3.—MSE RESULTS GIVEN DIFFERENT MIXTURE OF DATA SETS FOR TOTAL VALIDATION SET (μ IS MEAN AND σ IS STANDARD DEVIATION)

	Random (40 percent) and custom (60 percent)		Random (60 percent) and custom (40 percent)	
	MLP	RNN	MLP	RNN
Total	μ = 0.26 σ = 0.93	μ = 0.04 σ = 0.22	μ = 0.07 σ = 0.18	μ = 0.003 σ = 0.12
Random	μ = 0.59 σ = 1.48	μ = 0.05 σ = 0.3	μ = 0.03 σ = 0.17	μ = 0.001 σ = 0.04
Custom	μ = 0.08 σ = 0.19	μ = 0.03 σ = 0.13	μ = 0.12 σ = 0.19	μ = 0.04 σ = 0.18

TABLE 4.—MSE RESULTS GIVEN DIFFERENT MIXTURE OF DATA SETS FOR SPECIFIC CASES

	Random (40 percent) and custom (60 percent)		Random (60 percent) and custom (40 percent)	
	MLP	RNN	MLP	RNN
Random	0.03 to 2.78	0.01 to 0.3	0.03 to 3.22	0.0 to 0.3
Custom	0.01 to 0.24	0.0 to 0.64	0.0 to 0.38	0.0 to 0.91

Conclusions

ANN models using both MLP and RNN were developed to predict micromechanics-based laminate stiffnesses (ABD) and fatigue life of eight-ply, symmetric, PMC laminates. Deep learning artificial neural nets require a large amount of data for training/validation to achieve a certain level of accuracy. Generally, that amount of experimentally measured data is not available because of the cost and time involved. Therefore, synthetic/virtual data was generated using the physics-based MAC/GMC micromechanics model. The 10,000 laminates simulated required approximately 15,000 h of computations. Results indicate that the ANN based model can indeed accurately predict the laminate stiffness (i.e., ABD matrix) and fatigue life of PMC laminates at a fraction of the cost (CPU time) compared to the actual physics-based model. Models based on both ANN (MLP) and RNN were able to predict the stiffness quite accurately, but in general the RNN model was better able to capture the form of the fatigue curve with generally better accuracy and less computational time. *As such, an ML stiffness and fatigue life estimator tool was developed which designers can use for system level studies to obtain an estimate of desired properties and life of 8-ply symmetric PMC composite laminates with significantly less resources (CPU time, user effort and training). For example, given the generation of an entire typical S-N curve took approximately 1.5 h for MAC/GMC and the surrogate was approx. 1 s, this amounts to a speed up of* 1.8×10^{-4}. Similarly, the ply level Neural Net based surrogate models can be used in composite multiscale analyses to replace the actual physics-based calculations at lower scales and thereby significantly reduce the computational times of such analyses. These models thus enable multiscale analyses spanning several levels as viable industrial tools.

Appendix A.—Fatigue Load Estimator

A separate MATLAB script along with a corresponding GUI was developed to compute the upper and lower fatigue load limits automatically by running the physics-based MAC/GMC code iteratively. To utilize either tool, a reference MAC/GMC input deck (see Table A.1) is required that defines the specific laminate (ply angles and thickness), along with the constituent materials, volume fraction, and RUC that comprise the various plies, see Reference 10 for the required format and keywords. Given this reference input deck (and associated upper and lower limit initial guesses) execution of the script will determine both the upper and lower limit and then provide as output 10 MAC/GMC input decks, equally spaced between the limit loads/stresses, that can be run separately to obtain the required training data for this laminate.

Alternatively, one can run the GUI, by providing a reference input deck, inputting the assumed upper and lower limit as well as the number of points desired for displaying the associated S-N curve predicted by MAC/GMC. A sample GUI output is shown in Figure A.1. Here a sample laminate denoted LaminateB.mac (Table A.1) is chosen for illustration purpose. For this laminate, starting values for the lower and upper limits (see Figure A.1) are chosen as 10 and 60, respectively. However, in the script default values of 2 for lower bound and 7,000 for upper bound are provided to cover almost all such laminates. Default values for the load limits will increase the computational effort. Note one should select limits that are below the expected lower bound and above the upper bound for the defined laminate to find the limits in the first attempt. Also, 10 load points are chosen for computation of the S-N curve. The GUI automatically, updates the loading values highlighted in blue in Table A.1, executes MAC/GMC iteratively, until either the provided upper and lower load levels are achieved or a load that provides the prescribed upper bound load that corresponds to 1×10^3 cycles and lower bound load that corresponds to 1×10^9 cycles.

As seen in Figure A.1, the lower bound for fatigue is estimated to be 13 and the upper bound 35. In general, the lower bound calculations take a longer time to compute (minutes to hours) then the upper bound solutions (seconds). The SN curve is shown graphically between these two limits.

This script/GUI was extremely useful for generating training data associated with each laminate of interest that extended over the defined full life cycle ($N \geq 1{,}000$ and $N \leq 1\times10^9$.

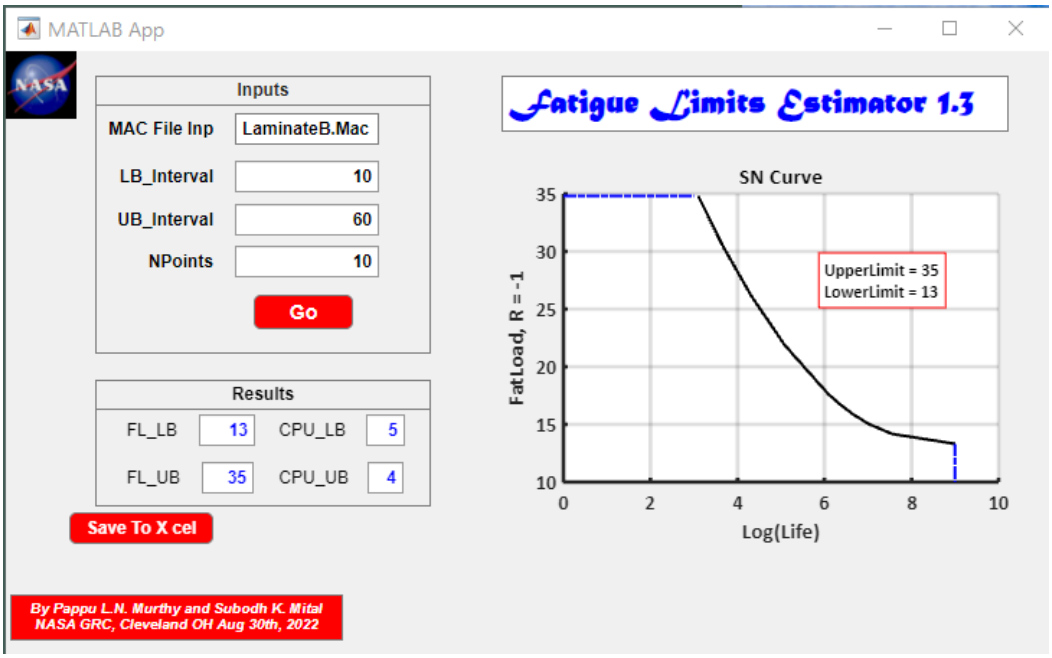


Figure A.1.—Fatigue load limit computations for LaminateB.mac.

TABLE A.1.—INPUT TO FATIGUE LOADS LIMITS ESTIMATOR (GUI)
LAMINATEB.MAC STIFFNESS ANALYSIS FOR GENERIC EIGHTPLY LAMINATE

```
*CONSTITUENTS
NMATS=2
# -- Graphite fiber
M=1 CMOD=6 MATID=U MATDB=1 &
EL= 330000.0,90000.0,0.21,0.22,100000.0,-5e-07,4.6e-06
# -- Epoxy matrix
M=2 CMOD=6 MATID=U MATDB=1 &
EL= 3800.0,3800.0,0.35,0.35,1407.4,5.2e-06,5.2e-06
#-- Define Laminate
*LAMINATE
NLY=8
LY=1 MOD=2 THK=0.125 ANG=45.0 ARCHID=6 VF=0.65 R=1 F=1 M=2
LY=2 MOD=2 THK=0.125 ANG=-45.0 ARCHID=6 VF=0.65 R=1 F=1 M=2
LY=3 MOD=2 THK=0.125 ANG=45.0 ARCHID=6 VF=0.65 R=1 F=1 M=2
LY=4 MOD=2 THK=0.125 ANG=-45.0 ARCHID=6 VF=0.65 R=1 F=1 M=2
LY=5 MOD=2 THK=0.125 ANG=-45.0 ARCHID=6 VF=0.65 R=1 F=1 M=2
LY=6 MOD=2 THK=0.125 ANG=45.0 ARCHID=6 VF=0.65 R=1 F=1 M=2
LY=7 MOD=2 THK=0.125 ANG=-45.0 ARCHID=6 VF=0.65 R=1 F=1 M=2
LY=8 MOD=2 THK=0.125 ANG=45.0 ARCHID=6 VF=0.65 R=1 F=1 M=2
*MECH
LOP=1
NPT=4 TI=0.,50.,150.,200. MAG=0.,835.0,-835.0,0. MODE=2,2,2
*SOLVER
METHOD=1 NPT=4 TI=0.,50.0,150.0,200.0 STP=10.0,10.0,10.0
NLEG=1 NINTEG=1
*DAMAGE
MAXNB=500 DINC=0.1 DMAX=0.9999 BLOCK=0.,200.
NDMAT=2
MAT=1 MOD=2 SU1=3000.0 818.2 818.2 472.4 1732.1 1732.1 &
SU2=1000.0 272.7 272.7 157.5 577.4 577.4 &
N1=1000,1000,1000,1000,1000,1000 &
N2=300000000,300000000,300000000,300000000,300000000,300000000
MAT=2 MOD=1 ANG=0. BN=0. BP=0. OMU=1. OMFL=1. OMM=1. ETU=1. &
ETFL=1. ETM=1. BE=7.0 A=0.05 SFL=20.0 XML=120.0 SU=95.0
*FAILURE_SUBCELL
NMAT=2
MAT=1 NCRIT=1
CRIT=1 X11=3000.0 X22=818.2 X33=818.2 X23=472.4 X13=1732.1 X12=1732.1 &
COMPR=SAM ACTION=1
MAT=2 NCRIT=1
CRIT=2 X11=0.025 X22=0.025 X33=0.025 X23=0.026 X13=0.026 X12=0.026 &
COMPR=SAM ACTION=1
*FAILURE_CELL
NCRIT=1
CRIT=2 X11=0.05 X22=0.05 X33=0.05 X23=0.05 X13=0.05 X12=0.05 &
COMPR=SAM ACTION=1
*PRINT
NPL=6
*XYPLOT
FREQ=1
LAMINATE=0
MACRO=0
#-- NAME=STR-STN X=2 Y=8
MICRO=0
*END
```

Appendix B.—Python Fatigue Estimator Graphical User Interface (GUI)

A python Graphical User Interface (GUI) was developed for the ABD prediction and Fatigue Life Curve prediction models for a given 8-ply, symmetric, PMC laminate. The GUI was written in Python using the tkinter module (Ref. 31) and exported to a Windows executable using the pyinstaller python function (Ref. 32), allowing users to run the GUI and machine learning models without installing Python or Tensorflow (Ref. 33) on their machines. The GUI features two pages that the user can navigate on the top left of the screen to 1) define laminate and its associated material input parameters and 2) view the results of the analysis. This Fatigue Life Estimator GUI represents the ultimate democratization toolset in that it enables a designer to obtain micromechanics-based fatigue life analysis results for an 8 ply, symmetric, PMC laminate with little or no training needed in mere seconds. The most difficult aspect of the process being the insertion of the correct life constituent properties for a given material. In a future version of this estimator the authors plan on including a library of potential material properties.

The Inputs tab, see Figure B.1, allows users to define a) the lamina stacking sequence and fiber volume fraction (see blue highlighted box) and the fiber properties (orange box) and matrix properties (green box) the composite laminate. For each category, the dark colored column can be edited by the user as input to the machine learning models. The upper and lower bounds are displayed for each input to the user in the lighter colored columns. Users are allowed to enter values outside of those bounds, but a message will appear warning that the ML models were trained within the specified limits and extrapolation beyond is not recommended. The input variables outlined in black below in Figure B.1 are required for the ABD model prediction, and all inputs are required for the Fatigue Life Curve model prediction. If values are missing for a required input, the GUI will display an error message when a user attempts to move to the Analysis page.

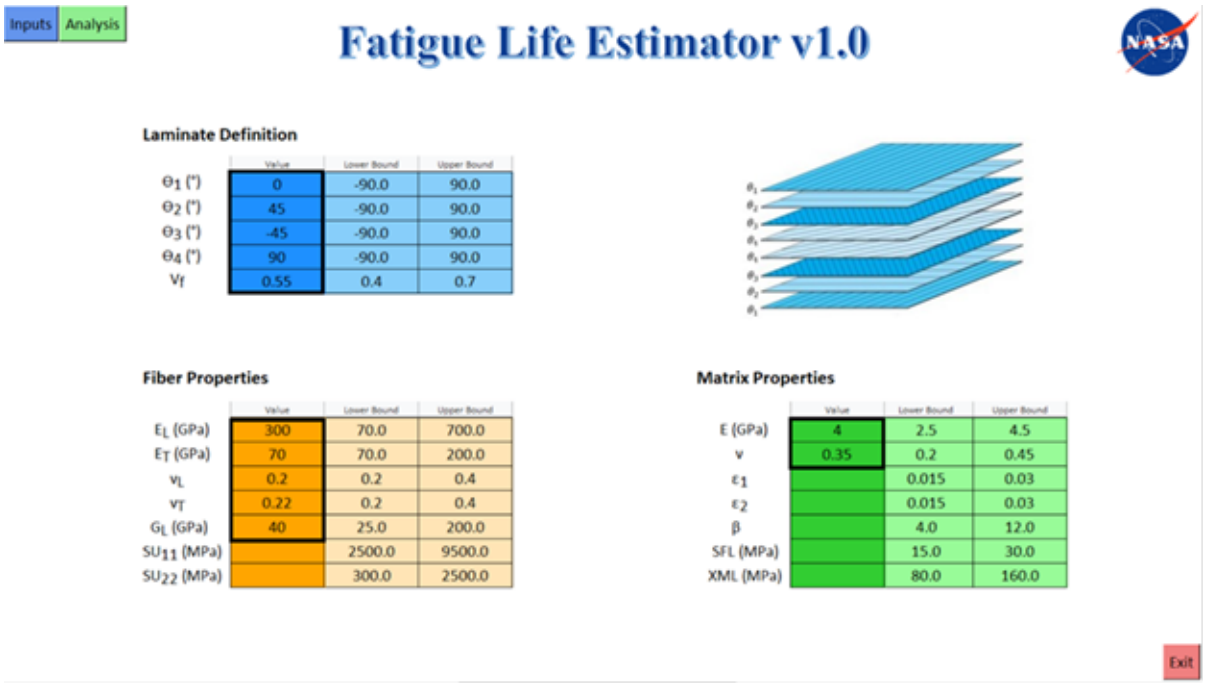
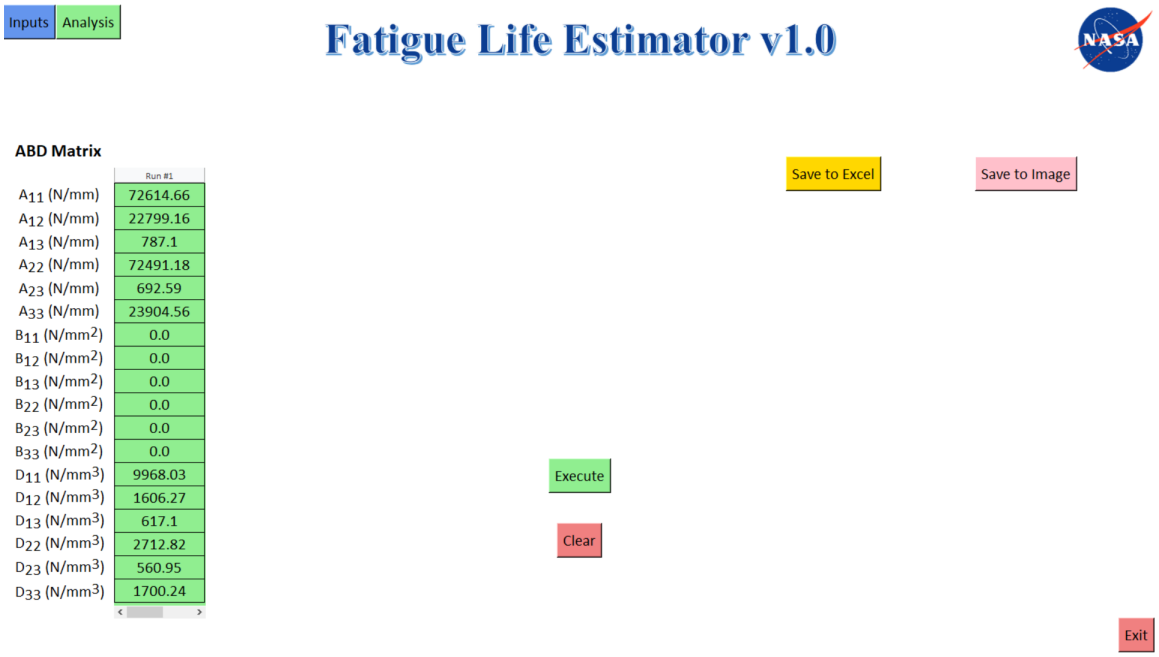
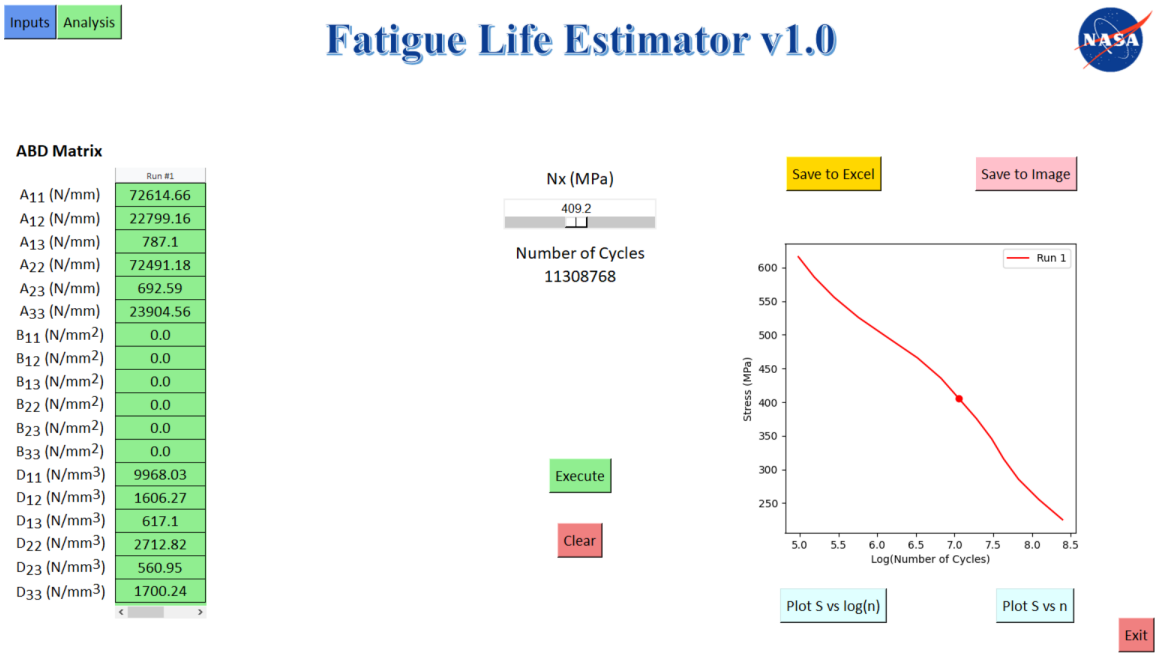


Figure B.1.—Fatigue Life Estimator GUI Inputs Page.

To run the machine learning models, the user selects the Analysis page and presses the “Execute” button, see Figure B.2. If only the inputs required for ABD prediction are populated, as is the case in Figure B.1, a table will appear on the Analysis page displaying the ABD predictions for that laminate (see Figure B.2(a)). If all inputs are populated, both the ABD table and an S-N curve plot appear on the Analysis page (see Figure B.2(b)). The S-N curve plot defaults to plotting stress values versus the log of the number of life cycles (log (n)). The X-axis parameter can be toggled between n and log (n) using the light blue buttons below the plot. The Analysis page also features a slider bar to extract an exact point from the S-N curve.



(a)



(b)

Figure B.2.—Fatigue Life Estimator GUI Analysis Page (a) ABD Prediction Only and (b) ABD and S-N Prediction

The user can return to the Inputs page using the buttons in the top left of the window and add additional laminate simulations to the analysis results. If new inputs are detected when the user returns to the Analysis page, a new column will appear in the ABD Matrix table on the left side of the page. When the user presses the Execute button, the table will populate with the new ABD components and the S-N curve will be co-plotted with all previous runs (Figure B.3). Note that the slider bar on the plot will only extract points for the *last* laminate case run (i.e., Run 2 in Figure B.3). Above the plot, the user has two options for saving results: Save to Image and/or save to Excel. The Save to Image button takes a screenshot of the analysis page and saves to a user-defined location as a Portable Networks Graphics (PNG) file. The Save to Excel buttons saves the inputs, ABD Matrix components, and S-N curve points predicted by the two ML models to a Microsoft Excel file, with each run saved on a new sheet (Figure B.4). Users can either clear the current job and erase all runs using the Clear button to start a new analysis or exit the program using the Exit button.

Future versions of this Fatigue Life Estimation Tool will enable users to select different ML models that will be applicable to other composite laminates (e.g., 16 ply, 24.ply, etc., and/or CMC and MMC systems).

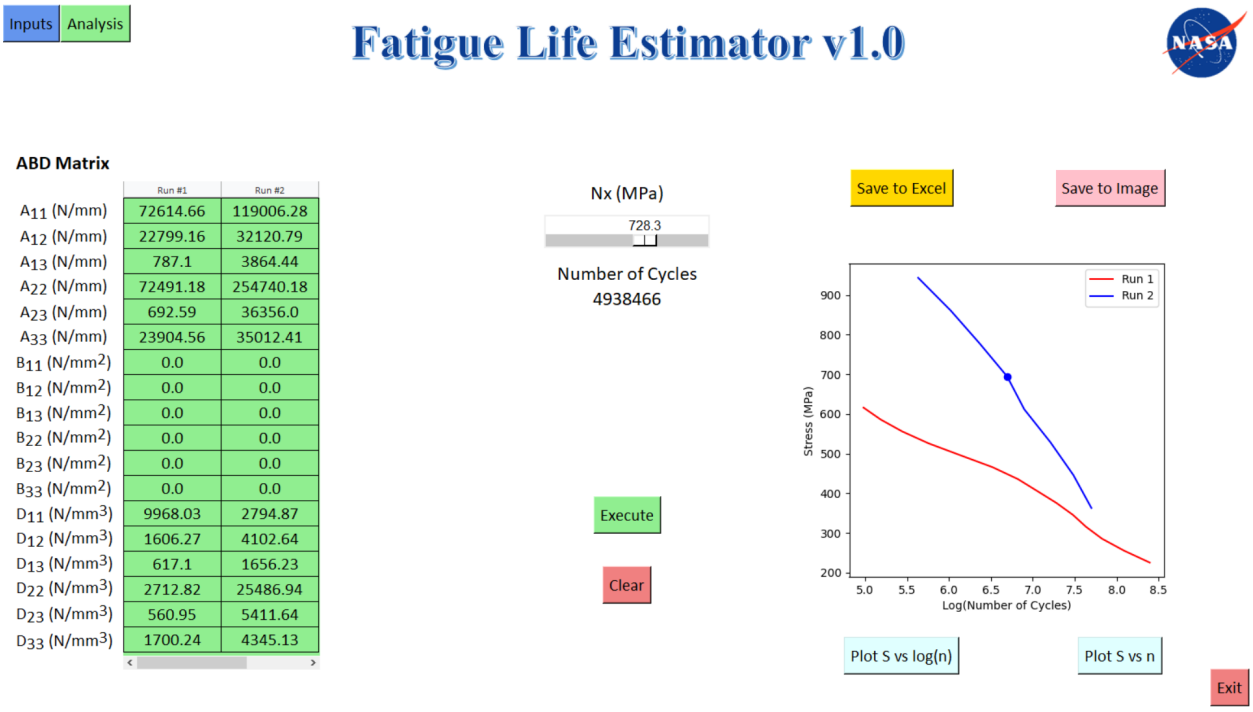


Figure B.3.—Adding multiple analyses.

	B	C	D
1	**Fiber Properties**		
2	*Mechanical Properties*		
3	Longitudinal Modulus (EL)	643	GPa
4	Transverse Modulus (ET)	116	GPa
5	Longitudinal Poisson's Ratio (v1	0.33	
6	Transverse Poisson's Ratio (v21)	0.22	
7	Longitudinal Shear Modulus (GL	133	GPa
8	*Life Properties*		
9	SU11	9000	MPa
10	SU21	1500	MPa
11			
12	**Matrix Properties**		
13	*Mechanical Properties*		
14	Elastic Modulus (E)	3.4	Gpa
15	Poisson's Ratio (v)	0.21	
16	*Life Properties*		
17	Longitudinal Strain (ε1)	0.026	
18	Transverse Strain (ε1)	0.025	
19	β	5.6	
20	SFL	30	MPa
21	XML	110	MPa
22			
23	**Laminate Properties**		
24	Ply Angle 1 (θ1)	62	°
25	Ply Angle 2 (θ2)	-80	°
26	Ply Angle 3 (θ3)	75	°
27	Ply Angle 4 (θ4)	-5	°
28	Volume Fraction (Vf)	0.64	

ABD Matrix		
A11	119006.3	N/mm
A12	32120.79	N/mm
A13	3864.44	N/mm
A22	254740.2	N/mm
A23	36356	N/mm
A33	35012.41	N/mm
B11	0	N/mm2
B12	0	N/mm2
B13	0	N/mm2
B22	0	N/mm2
B23	0	N/mm2
B33	0	N/mm2
D11	2794.868	N/mm3
D12	4102.641	N/mm3
D13	1656.225	N/mm3
D22	25486.94	N/mm3
D23	5411.641	N/mm3
D33	4345.131	N/mm3

S-n Curve	
n	S (MPa)
7.698999	362.9873
7.484299	445.899
7.20974	528.8107
6.89813	611.7224
6.693592	694.6341
6.366613	777.5458
6.021504	860.4575
5.63176	943.3692
6.52047	465.7368
6.136953	495.7895
5.756759	525.8421
5.441908	555.8947
5.184806	585.9474
4.979735	616

Run 1 | Run 2

Figure B.4.—Save to Excel Generated Output.

Appendix C.—MATLAB Fatigue Life Estimator Graphical User Interface (GUI)

A MATLAB Graphical User Interface (GUI) was developed with Neural Net surrogate models for the stiffness (A, B and D matrices) and the Fatigue Life (S-N Curve) for a given 8-ply, symmetric, PMC laminate. The GUI was developed with the aid of MATLAB's app designer and was later compiled including the MATLAB runtime libraries so that one can run it stand alone on any Windows OS platform. The installation of MATLAB code is not required. The app platform is shown in Figure C.1, and it has several input blocks that a user is required to fill in. These include transversely isotropic fiber (5) and isotropic matrix (2) elastic properties; fiber fatigue (2) and matrix fatigue model (5) model parameters; ply layup details (4 angles and fiber volume fraction) and the load limits (upper and lower bounds) for the estimation of the S-N fatigue curve. The user is given a choice to either chose one of the nine predefined laminates or a custom laminate with any lay-up. To guide the user with reasonable inputs, the appropriate lower and upper limits for every property is shown in red, so that the user can edit values for all the properties. Furthermore, default values are already listed in blue so if one does not have a good idea regarding a specific property, they may simply choose the default value given. This Fatigue Life Estimator GUI represents the ultimate democratization toolset in that it enables a designer to obtain micromechanics-based fatigue life analysis results for an 8 ply, symmetric, PMC laminate with little or no training needed in mere seconds. The most difficult aspect of the process being the insertion of the correct life constituent properties for a given material. A fully executed code output is shown in Figure C.1 for a ± 30 laminate, i.e., predicted S-N curve and the predicted A, D matrices along with the MAC/GMC outputs.

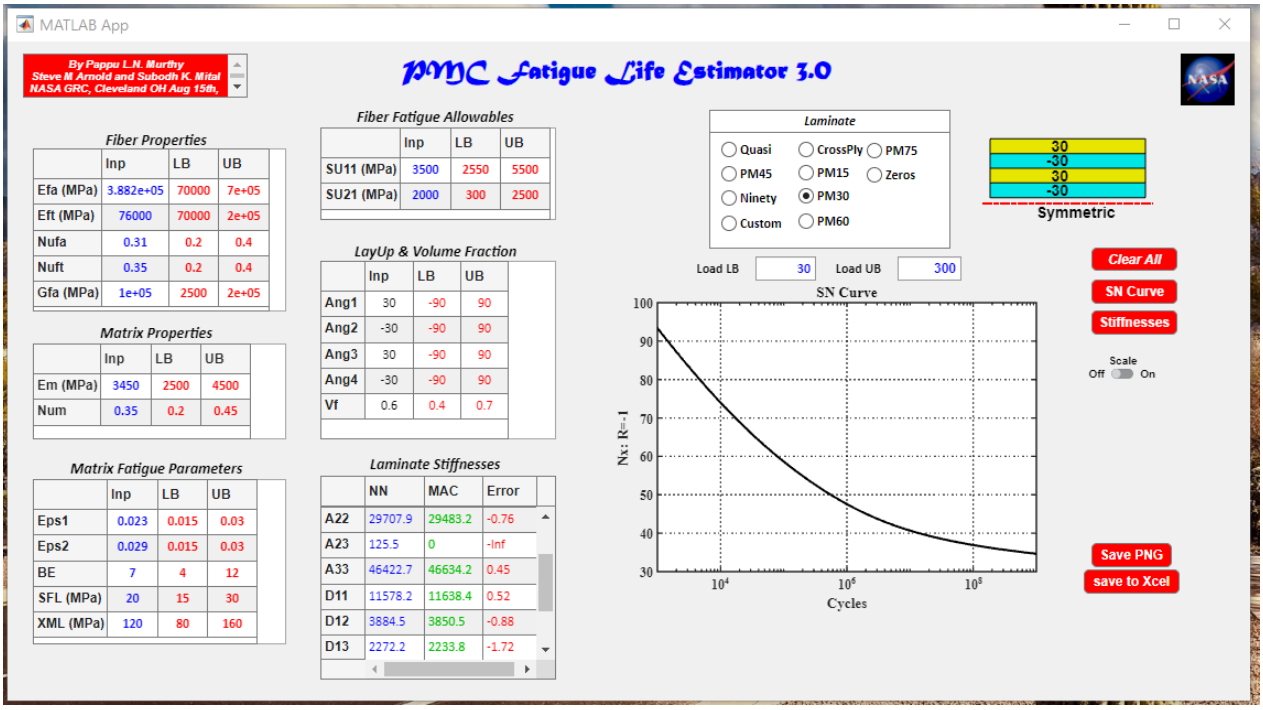


Figure C.1.—MATLAB based PMC fatigue life estimator GUI output.

User executes the code optionally by either pushing the button “SN Curve” for the fatigue life or “Stiffnesses” for A and D matrix computations or both. The user can predict multiple S-N curves and plot them for several laminates. To show them clearly separated, an option of choosing log scale for the ‘y’ coordinate is also provided. This would be necessary for example if a user chose to compare a unidirectional $[0_8]$ and angle ply (e.g., $[\pm 45°]_S$) laminates. The fatigue loads for the latter are more than an order of magnitude smaller compared to a unidirectional loaded in the fiber direction. In general, the stiffnesses are predicted within 1 percent error and the S-N curve predicted within 5 percent error. The surrogate model used for this GUI is trained with a MLP net for the randomly chosen lay-ups. The custom plies are all trained separately with different MLP nets. By choosing multiple surrogate models per laminate type, rather than training all laminates at once in one net, one can reduce the training data by at least an order of magnitude and improve the accuracy as well by an order of magnitude. The GUI also gives options to save the figure, as well as results to an Excel file if desired.

References

1. Brynjolfsso, Erik and Mitchell, Tom, "What Can Machine Learning Do? Workforce Implications", Science, Vol. 358, No. 6370, 2017. DOI: 10.1126/science. aap8062
2. Jackson, P., Introduction to Expert Systems, 3rd ed., Addison-Wesley, 1999.
3. Marsland, S. Machine Learning. CRC Press, Taylor & Francis Inc., Boca Raton, FL, 2014.
4. Haykins, S. Neural Networks. Prentice-Hall, 1999.
5. Wong, T.T., "Advances in ICME Implementations: Concepts and Practices, JOM Vol. 69, No. 879, 2017.
6. Arnold, S.M., Holland, F.A., Bednarcyk, B.A., Pineda, E.J., "Combining Material and Model Pedigree is Foundational to making ICME a Reality", Integrating Materials and Manufacturing Innovation, 4, 2015. DOI:10.1186/s40192-015-0031-2.
7. Liu, X., Tian, S., Tao, F., Yu, w., "A Review of Artificial Neural Networks in the Constitutive Modeling of Composite Materials", Composite Part B: Engineering, Vol. 224, Nov. 2021. https://doi.org/10.1016/j.compositesb.2021.109152.
8. Arnold, S.M., Piekenbrock, M., Ricks, T.M., Stuckner, J., "Multiscale Analysis of Composites Using Surrogate Modeling and Information Optimal Designs", Proc. 2020 AII SciTech Conference, Orlando, FL, Jan. 6-10, 2020. DOI:10.2514/6.2020-1863.
9. Stuckner, J., Graeber, S., Weborg, B., Ricks, T. M., & Arnold, S. M. (2021). Tractable Multiscale Modeling with An Embedded Microscale Surrogate. In AIAA SciTech 2021 Forum, DOI: 10.2514/6.2021-1963.
10. Bednarcyk, B. A., and Arnold, S. M. ; "MAC/GMC 4.0 User's Manual, Volume 2: Keywords Manual", TM 2002-212077/Vol 2, 2002.
11. Aboudi, J., Arnold, S.M., and Bednarcyk, B.A., ; *Micromechanics of Composite Materials: A Generalized Multiscale Analysis Approach*, Elsevier, Inc., 2013.
12. Aboudi, J., Arnold, S.M., and Bednarcyk, B.A., (2021); Practical *Micromechanics of Composite Materials*, Elsevier, Butterworth-Heinemann, 2021.
13. Paley, M. and Aboudi, J., (1992) "Micromechanical Analysis of Composites by the Generalized Cells Model" Mechanics of Materials, 14:127–139.
14. Aboudi, J., Pindera, M.J. and Arnold, S.M.; "Higher-Order Theory for Periodic Multiphase Materials with Inelastic Phases", NASA/TM 2002-211469. Also see IJP, 2003, 19: 805-847.
15. Romanov, V., Lomov, S. V., Swolfs, Y., Orlova, S., Gorbatikh, L., and Verpoest, I. (2013). Statistical Analysis of Real and Simulated Fibre Arrangements in Unidirectional Composites. Composites Science and Technology, 87:126-134.
16. Garnich, M. R., Fertig, R. S., and Anderson, E.M.; "Random Fiber Micromechanics of Fatigue Damage", 54th AIAA/ASME/ASCE/AHS/SC Structures, Structural Dynamics, and Materials Conference, AIAA 2013-165, Boston, MA, April 8-11, 2013.
17. N. E. Dowling, Mechanical Behavior of Materials: Engineering Methods for Deformation, Fracture, and Fatigue, Prentice Hall, New Jersey. 1999.
18. J. Lemaitre and Chaboche, J.L., Mechanics of Solid Materials, Cambridge University Press, 1990.
19. Skrzypek, J. and Hetnarski, R. Plasticity and Creep Theory Examples and Problems. Boca Raton, FL.: CRC Press, 1993.
20. Arnold, S. M. and Kruch, S., ''A Differential CDM Model For Fatigue of Unidirectional Metal Matrix Composites", Int. Journal of Damage Mechanics, Vol. 3, No. 2, pp.170-191
21. Chaboche, J.L. and Lesne, P.M.; "A Non-Linear Continuous Fatigue Damage Model", Fatigue Fract. Engnr. Mater. Struct., 11(1): 1-7., 1988.

22. Wilt, T.E., Arnold, S.M., and Saleeb, A.F.; "A Coupled/Uncoupled Computational Scheme for Deformation and Fatigue Damage Analysis of Unidirectional Metal-Matrix Composites", Applications of Continuum Damage Mechanics to Fatigue and Fracture, ASTM STP 1315, D.L. McDowell (Ed.), 65-82, 1997.
23. Bednarcyk, B. A., and Arnold, S. M. ; "MAC/GMC 4.0 User's Manual, Volume 3: Example Problem Manual", TM 2002-212077/Vol 3, 2002.
24. Kruch, S. and Arnold, S. M., ''Creep Damage and Creep-Fatigue Damage Interaction for Metal Matrix Composites ''Applications of Continuum Damage Mechanics to Fatigue and Fracture, ASTM STP 1351, D.L. McDowell, Ed., American Society for Testing and Materials, pp. 7-28.
25. Chollet, Francois, Deep Learning with Python. Manning Publications Co, New York. 2018.
26. Paluszek, M., Thomas, S., Ham, E., Practical MATLAB Deep Learning – A Projects-Based Approach, 222nd ed., Apress, 2022.
27. Anaconda Python for Windows version 3.9, 2022.
28. MATLAB, 2018, version R2018, The MathWorks, Inc., Massachusetts, United States.
29. Hearley, B., Arnold, S.M., Stuckner, J.; A Robust Machine Learning Schema for Developing, Maintaining, and Disseminating Machine Learning Models, NASA/TM-20220017137, 2022
30. Hearley, B., Stuckner, J., Pineda, E., and Murman, S.; Predicting Unreinforced Fabric Mechanical Behavior with Recurrent Neural Networks, NASA/TM-20210023708, 2021.
31. "Graphical User Interfaces with Tk," Python Software Foundation, 17 October 2022. [Online]. Available: https://docs.python.org/3/library/tk.html.
32. D. Cortesi, "PyInstaller Manual," PyInstaller Manual, 23 March 2023. [Online]. Available: https://pyinstaller.org/en/stable/
33. M. Abadi et al., "TensorFlow: A System for Large-Scale Machine Learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, 2016.