



An ABAQUS® plug-in for generating virtual data required for inverse analysis of unidirectional composites using artificial neural networks

Yaser Ismail¹ · Lei Wan² · Jiayun Chen^{2,4} · Jianqiao Ye³ · Dongmin Yang² 

Received: 1 April 2021 / Accepted: 14 October 2021 / Published online: 31 October 2021
© The Author(s) 2021

Abstract

This paper presents a robust ABAQUS® plug-in called Virtual Data Generator (*VDGen*) for generating virtual data for identifying the uncertain material properties in unidirectional lamina through artificial neural networks (ANNs). The plug-in supports the 3D finite element models of unit cells with square and hexagonal fibre arrays, uses Latin-Hypercube sampling methods and robustly imposes periodic boundary conditions. Using the data generated from the plug-in, ANN is demonstrated to explicitly and accurately parameterise the relationship between fibre mechanical properties and fibre/matrix interphase parameters at microscale and the mechanical properties of a UD lamina at macroscale. The plug-in tool is applicable to general unidirectional lamina and enables easy establishment of high-fidelity micromechanical finite element models with identified material properties.

Keywords Plug-in · Unidirectional lamina · Artificial neural networks · Periodic boundary conditions · Finite element modelling

1 Introduction

Fibre-reinforced polymer (FRP) composite laminates have been widely used in aerospace, automotive, and wind energy industry due to their excellent material properties such as high stiffness-to-mass ratio, high strength, and light weight. Applications of FRP composite laminates to create engineering structure models fundamentally require mechanical properties as inputs. Experimental tests are ideal solutions to evaluate the mechanical properties of a composite lamina. However, it must be repeated whenever the constituents (fibre and matrix) and/or microstructure characteristics (fibre volume fraction) are altered. This procedure may, for instance, costs millions of dollars and lasts for years to

generate the experimental data of mechanical properties for the design of aircraft structures [32].

To overcome the aforementioned drawbacks associated with experimental tests, various micromechanical approaches have been proposed to establish a closed-form relationship between elastic properties at the lamina scale and the elastic properties at the constituent scale. These methods fall generally into two categories, i.e., analytical and numerical methods. Analytical methods, such as the Rule of Mixture method [6], the Halpin–Tsai semi-empirical method [10], the Mori–Tanaka method [19], and the Chamis method [5], facilitate the calculation of elastic properties by a direct mathematical, empirical expression between constituent properties and elastic properties of the lamina. However, these methods have an inherent limitation in describing the stress and strain fields at microstructural scale mainly due to neglecting fibre interaction.

With the development of computing capacities, numerical methods, in particular the finite element method (FEM), have become widely used tools for studying the behaviour of composites, including inverse analysis [8, 15, 26], elastic moduli [27], failure of composite lamina [29, 33, 34], and the effective coefficients of thermal expansion [13]. Inverse analysis has been used to identify fibre mechanical properties and fibre thermal expansion coefficient, and

✉ Dongmin Yang
Dongmin.Yang@ed.ac.uk

¹ Groundshore, Barleycastle Ln, Warrington WA4 4RB, UK

² Institute for Materials and Processes, School of Engineering, University of Edinburgh, Edinburgh EH9 3FB, UK

³ Department of Engineering, Lancaster University, Lancaster LA1 4YW, UK

⁴ College of Mechanical and Vehicle Engineering, Hunan University, Changsha 410082, People's Republic of China

evaluate the factors of analytical methods, and so on. [26] determined the elastic and thermal properties of graphite fibre using inverse analysis. [3] predicted fibre properties using finite element analysis of hexagonal and random representative volume element (RVE) through inverse analysis. Similarly, [14] conducted an inverse analysis to predict fibre mechanical properties. However, they used quasi-analytical gradients derived from analytical models such as Chamis or Halpin–Tsai to reduce the computational cost. [15] utilised an inverse method to identify the mechanical properties of T300 carbon fibre as well as the interphase region parameters based on a computational homogenisation approach together with experimental results and Kriging metamodeling. [8] carried out an inverse analysis in the framework of FEM to estimate the reinforcement parameter ξ of the Halpin–Tsai models which is used to calculate transverse stiffness E_2 . A total number of 67 FE models of 2D square, 2D hexagonal, and 3D random fibre distributions were used to obtain a new value of ξ with a high level of confidence.

Regardless of the inverse methods used or the purpose they are used for, a large number of FE analyses are required for converged solutions. However, constructing a micromechanical FE model is not a straightforward task and requires special treatments to impose boundary conditions, and generate microstructures including fibre distribution and fibre/matrix interphase, and extract outputs, etc. These complexities impose a barrier of using inverse analysis by engineers and researchers. Recently, several ABAQUS plug-ins have been developed for the ease of creating micromechanical FE models. These plug-ins were developed either by the functions available in ABAQUS or by external software. An ABAQUS plug-in named MultiMech was developed to perform multi-scale finite-element analysis (FEA) with the capability of simulating nonlinear behaviours of composites [16]. Another ABAQUS plug-in for multilevel modelling of linear and nonlinear behaviour of composite structures [7, 28]. The plug-in developed using Python scripts for analysing an RVE at microscopic level to obtain macroscopic parameters for structural analysis by user-defined FORTRAN subroutines in ABAQUS. Composite Micro-Mechanics (COMM) toolbox was developed in Matlab for micromechanical analysis of composites [17]. The toolbox creates an input file that can be read by ABAQUS which performs the FEA. Recently, EasyPBC plug-in was developed for ABAQUS to estimate effective elastic properties of a pre-prepared and meshed RVE [21]. While, the ABAQUS plug-in proposed by [24] is capable of generating an RVE with random fibre distribution using Random Sequential Adsorption (RSA) technique.

The aforementioned plug-ins have shown outstanding benefits and capabilities to create and simulate complex RVEs of unidirectional (UD) FRP composite lamina. However, they are designed to generate and analyse a single

model. Therefore, this paper aims to develop an open-source ABAQUS plug-in named Virtual Data Generator (*VDGen*) that automates the time-consuming manual task requires to create a large number of virtual data for inverse analysis. The plug-in uses Latin-Hypercube (LH) sampling methods and supports the unit cell of square and hexagonal fibre arrays. In addition, the plug-in incorporates Artificial Neural Networks (ANN) to explicitly parameterise the relationship between fibre mechanical properties and fibre/matrix interphase parameters and the mechanical properties of a UD lamina. The data required here were created in advance by the plug-in and used to train the ANN model.

2 Main plug-in GUI

The concept of the plug-in arises from the need for a tool that helps to perform a large number of micromechanical FE simulations in a few simple steps. ABAQUS has different ways to increase its capabilities such as subroutines and/or adding new plug-ins. ABAQUS/CAE plug-in is one of the most powerful tools that can be used to perform pre- and post-processing via functions written in Python programming language in the kernel. The current plug-in operates through a series of user-friendly GUI commands send to the kernel to carry out tasks. The plug-in interface is shown in Fig. 1. It consists of six tab items that allow the user to navigate between them to edit input and output commands. For computational micromechanics modelling, the plug-in supports square and hexagonal unit cell fibre arrays. Despite fibres are usually randomly distributed in the matrix, it has been concluded by [31] that micromechanical modelling of the unit cell is accurate enough to predict the elastic properties of a UD lamina, while an RVE with randomly distributed fibres is essential to compute the local failure.

Figure 2 shows a typical 3D unit cell of square and hexagonal fibre arrays of the fibre reinforced composite that the plug-in supports. The mechanical properties of each constituent, i.e., fibre, matrix and interphase, can be modified in the material section. There are two ways to input the value of constituent properties, either by a single value or using a domain of lower and upper bounds (lower–upper). A material property assigned with a single value remains unchanged throughout simulations. While for others, a random value in the range of (lower–upper) is generated at each training point using Latin-Hypercube sampling technique.

The fibres and matrix are meshed using eight-node brick element with reduced integration (C3D8R). There were also a relatively small amount of six-node linear triangular prism elements (C3D6) due to the free meshing technique used. The interphase region is meshed with eight-node cohesive elements (COH3D8). To maintain matched meshes between the cohesive elements and the

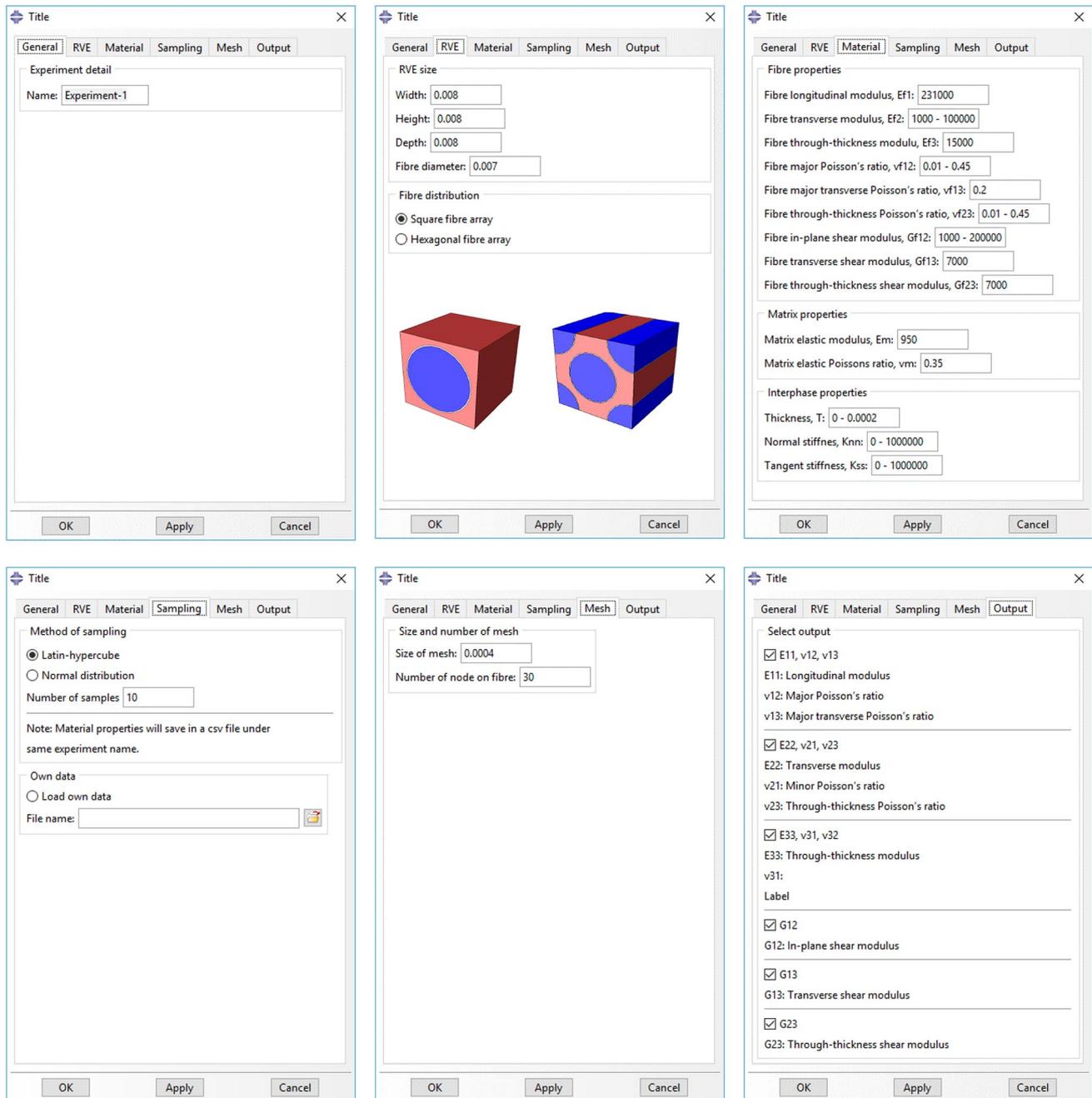


Fig. 1 The graphical user interface (GUI) of the plug-in

fibres and matrix elements, a suitable number of nodes are seeded to the interphase and its neighbours. The elastic behaviour of the cohesive elements is written in terms of a stiffness matrix that relates the nominal stresses to the nominal strains across the interphase. The nominal traction stress vector \mathbf{t} consists of three components, t_n , t_s , t_t , which represent the normal and two shear tractions, respectively. The corresponding separations are denoted by δ_n , δ_s , and δ_t , and the original thickness of the cohesive

element is denoted by T . Then, the nominal strains can be defined as

$$\epsilon_n = \frac{\delta_n}{T}, \epsilon_s = \frac{\delta_s}{T}, \epsilon_t = \frac{\delta_t}{T}. \tag{1}$$

Therefore, the elastic behaviour of the cohesive element can be written in Eq. (1). For simplicity of computation, uncoupled behaviour between the normal and shear

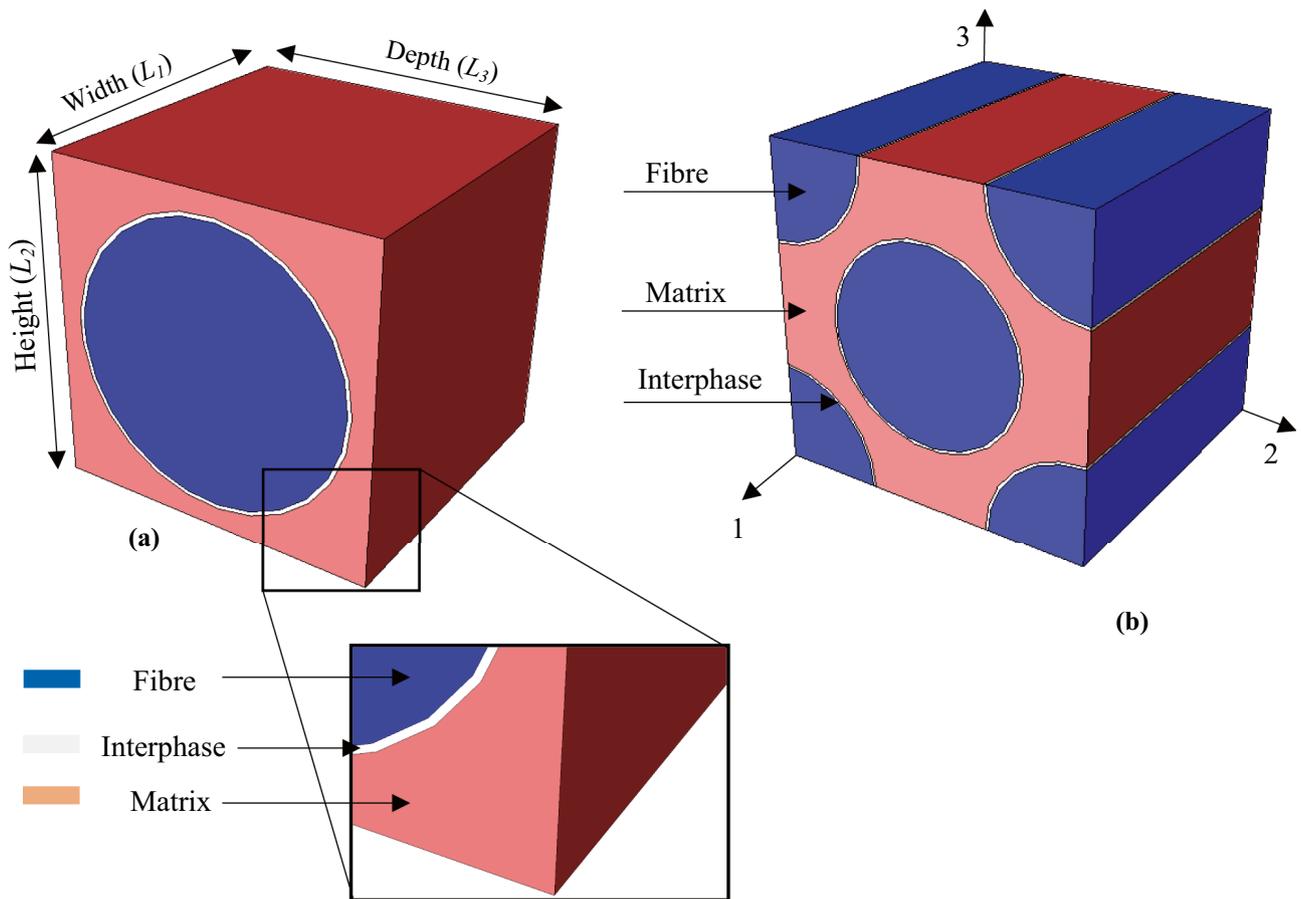


Fig. 2 3D unit cell models of regular fibre arrays: **a** Square and **b** Hexagonal

components is desired, so the off-diagonal terms in the elasticity matrix are set to be zero and the stiffness in the two shear directions are assumed to be equal [1, 33]

$$t = \begin{Bmatrix} t_n \\ t_s \\ t_t \end{Bmatrix} = \begin{bmatrix} K_{nn} & K_{ns} & K_{nt} \\ K_{ns} & K_{ss} & K_{st} \\ K_{nt} & K_{st} & K_{tt} \end{bmatrix} \begin{Bmatrix} \epsilon_n \\ \epsilon_s \\ \epsilon_t \end{Bmatrix} = K\epsilon. \tag{2}$$

The plug-in imposes Periodic Boundary Conditions (PBC) on the corresponding surfaces of the unit cell to ensure the compatibility of strain and stress at the macroscale level. These consist of a series of constraints in which the deformation of each pair of nodes on the opposite surfaces of the unit cell is subject to the same amount of displacements. The PBCs are expressed in terms of the displacement vectors \bar{U}_1 , \bar{U}_2 , and \bar{U}_3 that are related to the displacements between the opposite surfaces by

$$\begin{cases} \bar{u}(0, x_2, x_3) - \bar{u}(L_1, x_2, x_3) = \bar{U}_1 \\ \bar{u}(x_1, 0, x_3) - \bar{u}(x_1, L_2, x_3) = \bar{U}_2 \\ \bar{u}(x_1, x_2, 0) - \bar{u}(x_1, x_2, L_3) = \bar{U}_3 \end{cases}, \tag{3}$$

where L_1 , L_2 , and L_3 are the lengths of the unit cell along with three orthogonal directions, respectively. PBC requires matching nodes on opposite sides of the unit cell. Hence, elements of equal size are assigned to the edges of the unit cell to ensure periodic mesh required for PBC.

The *Output* tab allows the user to select appropriate results that suit the work. The macroscopic normal and shear strain components are calculated by

$$\epsilon_{ii} = \frac{\bar{U}_i}{L_i} \text{ (normal strain)}, \tag{4}$$

$$\epsilon_{ij} = \frac{\bar{U}_i}{L_j} + \frac{\bar{U}_j}{L_i} \quad (i \neq j) \text{ (shear strain).} \tag{5}$$

The macroscopic stress is calculated as

$$\sigma_{ij} = \frac{\sum F_i}{A_j}, \tag{6}$$

where F_i is the resultant force on the i th surface which represents the reaction force at a reference point where the displacement is applied, and A is the area of the surface. Therefore, Young’s modulus, Poisson’s ratio, and shear modulus are, respectively, calculated from Eqs. (7), (8), and (9)

$$E_{ii} = \frac{\sigma_{ii}}{\epsilon_{ii}} \tag{7}$$

$$G_{ij} = \frac{\sigma_{ij}}{\epsilon_{ij}} \tag{8}$$

$$\nu_{ij} = -\frac{\epsilon_{ij}}{\epsilon_{ii}}. \tag{9}$$

The flowchart of the pre- and post-processing procedure of the plug-in is described in Fig. 3. The user is to define the analysis of data and the required outputs, as given in Fig. 3. When the ‘OK’ or ‘Apply’ button is clicked, the plug-in creates three files to be called in the subsequent steps. *ExperimentNAME.dat* file contains all input commands which are given by the user in the GUI interface. These commands are vital to creating the FE model. Also, this file provides an opportunity to modify the input commands using exact plug-in keywords in the file. *~Temp.txt* is dedicated to storing the experiment name the user wants to run. Material properties values created by Latin-hypercube sampling are stored in a tabular format in *ExperimentNAME.csv* file, which makes them easier to read and process by external software. Once these files are created,

the Python script (*Execute.py*) can be submitted for analysis from ABAQUS command environment using either *dos* (*'abaqus cae script=Execute.py'*) or *dos* (*'abaqus cae noGUI=Execute.py'*) command. It is strongly recommended to execute it using the latter in which ABAQUS/CAE runs commands in *Execute.py* without the added expense of running a GUI display. Whichever option adopted to perform the analysis, the plug-in continuously provides the user with useful information, e.g., number of jobs done, number of jobs remain and approximate time to complete. Upon running, the plug-in instantly creates two files to store outputs after completion of each job and to record errors that occurred during the analysis.

3 Numerical example: prediction of effective elastic properties

To validate the newly developed plug-in, the effective elastic properties of carbon fibre/epoxy (T300/PR-319) and glass fibre/epoxy (E-Glass/MY750) were determined and compared with EasyPBC plug-in developed by [21] as well as the experimental data [12]. The mechanical properties of the fibre, matrix, and interphase are given in Table 1. Since T300 carbon fibre is classified as a transversely isotropic material, the elastic properties highlighted by an asterisk * symbol in the table are obtained by applying the following relations:

$$G_{f12} = G_{f13}, G_{f23} = \frac{E_{f2}}{2(1 + \nu_{f23})}. \tag{10}$$

The E-glass is considered as isotropic material. It is important to note that the input data for the interphase region are not accurately known as they are difficult to measure from simple laboratory experiments. However, an initial stiffness K_i of 10^5 GPa/mm is used in [2, 18, 25, 30] to simulate the elastic behaviour of the RVE model. In this paper, the elastic parameters from [15] are used to for the interphase as an approximation.

Table 2 shows the comparison of the predicted effective elastic properties determined by VDGen and EasyPBC plug-ins. It is noted among the prediction results that VDGen provides reliable results that are identical to those from EasyPBC. However, an obvious discrepancy exists between experimental results and those predicted by the two plug-ins. This is mainly due to the inaccurate parameters used for the interphase region.

Figure 4a shows the stress contours of a loaded unit cell under transverse and Fig. 4b shows the stress contours for in-plane shear loading conditions. It can be seen the periodic stress contours distribution which is additional verification of the PBC.

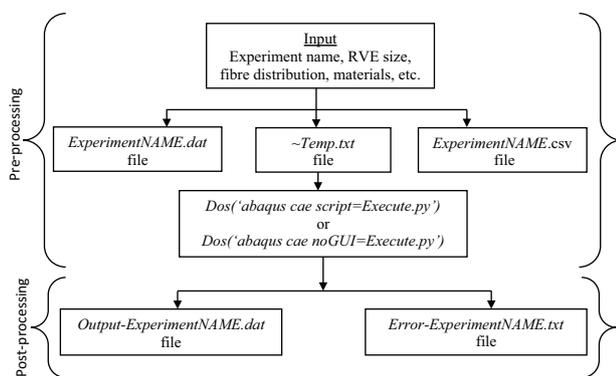


Fig. 3 Flowchart of the plug-in

Table 1 Elastic properties of fibre, matrix [12] and interphase [15]

Fibre elastic property	T300 carbo fibre	E-Glass fibre
Longitudinal modulus, E_{f1} (GPa)	231	74
Transverse modulus, E_{f2} (GPa)	15	74
Through-thickness modulus, E_{f3} (GPa)	15	74
In-plane Poisson's ratio, ν_{f12}	0.2	0.2
Major transverse Poisson's ratio, ν_{f13}	0.2	0.2
Through-thickness Poisson's ratio, ν_{f23}	0.07143*	0.2
In-plane shear modulus, G_{f12} (GPa)	15	30.8
Transverse shear modulus, G_{f13} (GPa)	15*	30.8
Through-thickness shear modulus, G_{f23} (GPa)	7	30.8
Matrix elastic property	PR-319	MY750 matrix
Elastic modulus, E_m (GPa)	4.0	3.35
Poisson's ratio, ν_m	0.35	0.35
Interphase region parameters	Interphase	Interphase
Thickness, T (μm)	0.139	0.139
Normal stiffness, K_m (GPa/mm)	846.8	846.8
Tangent stiffness, K_{ss} (GPa/mm)	535.6	535.6

f stands for fibre, m stands for matrix

Table 2 Comparison of predicted effective elastic properties determined by VdGen, EasyPBC and experimental data [12, 15]

Elastic property	T300/PR-319			E-Glass/MY750		
	VdGen	EasyPBC	Experimental data	VdGen	EasyPBC	Experimental data
E_{11} (GPa)	127.54	127.54	138	42.11	42.11	45.6
E_{22} (GPa)	3.97	3.97	11	14.86	14.86	16.2
E_{33} (GPa)	3.98	3.98	11	14.87	14.87	16.2
ν_{12}	0.24	0.24	0.28	0.24	0.24	0.278
ν_{13}	0.24	0.24	0.28	0.24	0.24	0.278
ν_{23}	0.25	0.25	0.4	0.25	0.25	0.4
G_{12} (GPa)	1.36	1.36	5.5	4.52	4.52	5.83
G_{13} (GPa)	1.36	1.36	5.5	4.52	4.52	5.83
G_{23} (GPa)	0.89	0.89		3.21	3.21	5.7

4 Application: identifying fibre and interphase parameters using ANN

4.1 Background

In this section, a machine learning (ML) technique Artificial Neural Network (ANN) is used to construct a relationship between fibres and interphase parameters and effective elastic properties of the lamina. It is inspired by the animal brain's structure and function, which learns from former examples. ANN consists of three main layers: an input layer, one or more hidden layers, and an output layer. Each layer has several neurons which are responsible for

transmitting weight and biases (equivalent to chemical and electric signals in the animal brain) between two layers. Figure 5 illustrates a typical structure of single neurons where each input (x) comes from the previous layer multiplied with its individual weight of the connection (w_i) and then summed up with biases [23]. Then, this sum is composed with activation function (f), resulting in another vector (a) as

$$a = f(x_i \cdot w_i + b). \quad (11)$$

Another key step of ANN is a defined objective function that is to be minimised during the training process. Mean Squared Error (MSE) and Sum Squared Error (SSE) among

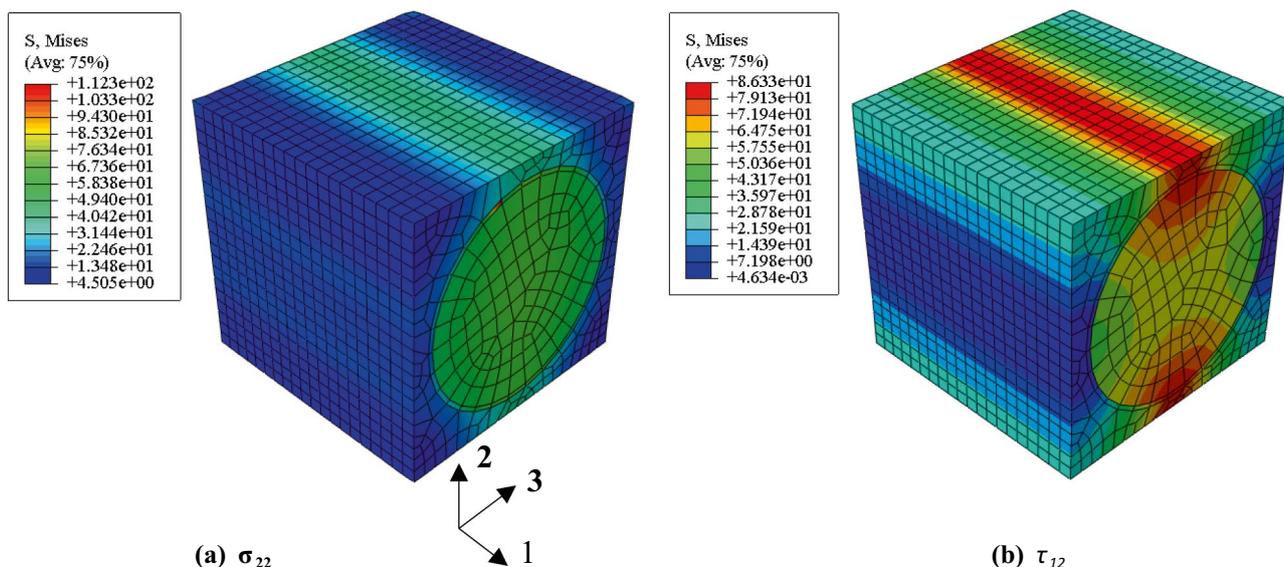


Fig. 4 Stress contour plot of the unit cell under a Transverse loading and b In-plane shear loading

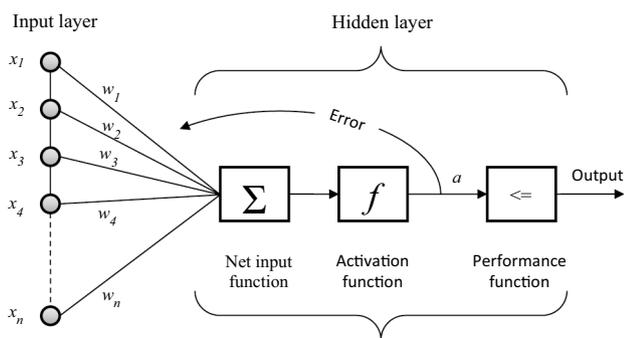


Fig. 5 A typical one neuron structure of the ANN

others are examples of functions used to assess the network’s behaviour by measuring the errors between the output and the target. The errors are reduced through tuning the values of the weight and biases by the so-called back-propagation. Back-propagation is widely recognised as a powerful tool for the training of ANN very efficiently. Several algorithms have been proposed to address the slow convergence associated with the back-propagation. However, it is quite difficult to decide which algorithm is more computationally efficient as it depends on many factors. Readers may refer to a comparative study carried out by researchers to evaluate accuracy and convergence time for different algorithms [4, 9].

4.2 ANN model to identify micro-parameters

ANN model is developed to identify the micro-parameters, e.g., fibre and fibre/matrix interphase parameters. The relationship between micro- and macro-properties in UD lamina

is fairly complex and nonlinear. Moreover, the number of micro-parameters to be identified is usually more than the number of macro-properties, which makes the ANN a complicated task. Therefore, to ease the training process, the micro-parameters are set to be the input layer of the neural networks model and the macro-properties are of the output layer. However, the calculation of optimal micro-parameters becomes difficult when they are in the input layer as it is not possible to obtain an analytical inverse response solution with the ANN model that has multiple neurons in the hidden layer. This issue is overcome using trained ANN to enlarge the dataset. Details of model building are explained in the following section.

4.2.1 Model building

The whole procedure of the fibre and interphase parameters identification using ANN is illustrated in Fig. 6. Firstly, a total of $n = 500$ FE models were created by VGen using the procedure outlined in Sect. 3. In each model, a random value of the parameters to be identified is created by LH sampling within the range given in Table 3. The remaining fibre properties were obtained by applying the transversely isotropic material relationships

$$E_{f3} = E_{f2}, \nu_{f13} = \nu_{12f}, G_{f13} = G_{f12}, G_{f23} = \frac{E_{f2}}{2(1 + \nu_{f23})}. \tag{12}$$

E_{f1} remained unchanged in all samples and its value was 230GPa . For all samples, the fibre volume fraction is 60% and matrix properties are given in Table 1. By the end of

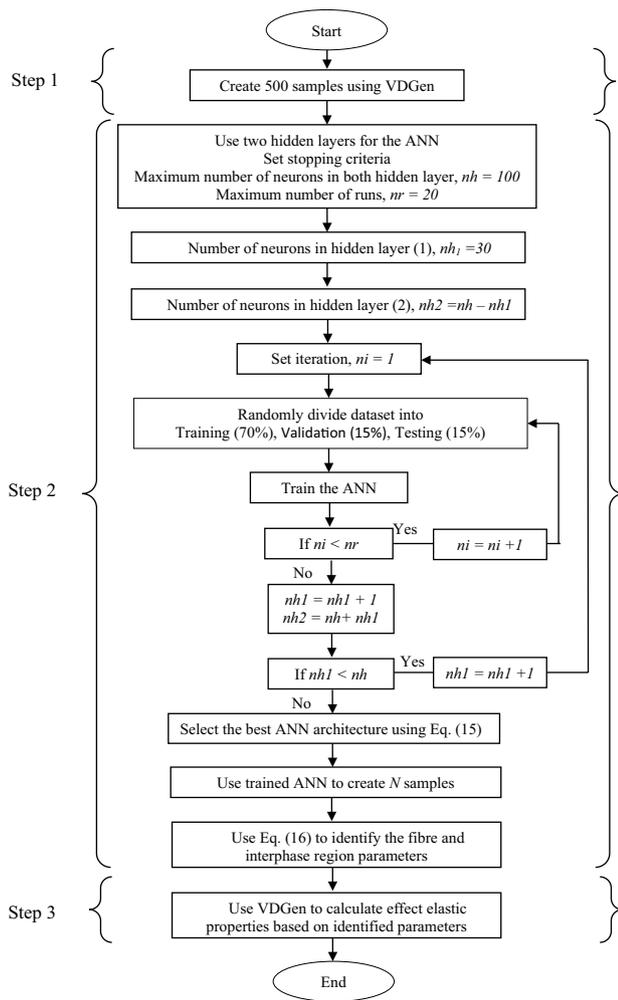


Fig. 6 Fibre and interphase parameters identification through ANN

Table 3 Ranges of fibre and interphase elastic properties

Elastic property	T300 carbon fibre
E_{f2} (GPa)	1–100
ν_{f12}	0.01–0.45
ν_{f23}	0.01–0.45
Interphase	
T (μm)	0.009–0.2
K_m (GPa/mm)	1–1000
K_{ss} (GPa/mm)	1–1000

this phase (Step 1), a dataset of 500 samples containing the inputs ($x = [E_{f2}, \nu_{f12}, \nu_{f23}, G_{f12}, T_i, K_m, K_{ss}]$) and the targets ($t = [E_{11}, E_{22}, \nu_{12}, \nu_{23}, G_{12}]$) required to train ANN is obtained (Fig. 7).

In Step 2, the ANN was built and trained using the dataset created in the previous step. The training, testing, and validation of the ANN model were conducted by MATLAB R2015a software. By MATLAB default, 70%, 15%, and 15% of the original dataset were used for training, validation, and testing, respectively. Nonlinear tangent sigmoid and linear functions were employed as the activation functions in the hidden layers and the output layer, respectively.

$$f(x) = 2 / (1 + e^{-2x}) - 1 \quad \text{tansig}(\text{tangent sigmoid activation function})$$

$$f(x) = x \quad \text{purelin}(\text{linear activation function}). \tag{13}$$

Selecting a best representative ANN structure plays an important role in output prediction. In this study, two hidden layers were used and the number of neurons in each hidden layer was changed until the best possible prediction was obtained. Initially, the number of neurons in the first hidden layer (nh_1) was set to 20 then increased by one, whereas the total number of neurons in both hidden layers (nh) was retained at 100. Usually, the data are randomly divided into three subsets (training, validation, and testing), and different initial weight and bias values are used in each time the neural networks are trained. As a result, different neural networks trained for the same problem may give different outputs for the same inputs. In this study, therefore, 20 runs were performed on each ANN architecture to ensure inclusion of different data for each subset. MSE, which is the average squared difference between the output (y) vectors and the target (t) vectors, was used to compute the difference and back-propagated through the networks to update weights and biases

$$MSE = \frac{1}{n} \sum_{i=1}^n (y - t)^2. \tag{14}$$

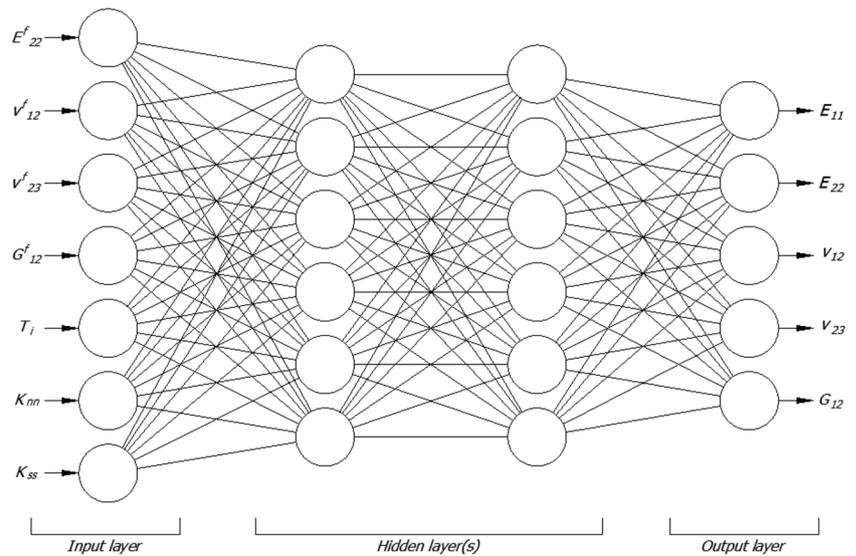
Levenberg–Marquardt (LM) back-propagation algorithm was adopted in this study. This function uses back-propagation scheme to update weights and biases according to Levenberg–Marquardt optimization algorithm, which can accurately achieve results with fewer data comparing with its counterparts.

To decide which is the best ANN taking into account the fact of dividing the input data into three main subsets (training, validation, and testing) during the training process, the sum of correlation coefficient (R-value) between the target and the output of the entire dataset was used to attain the optimal ANN structure

$$\max R(x) = R_v^{E11} + R_v^{E22} + R_v^{\nu12} + R_v^{\nu23} + R_v^{G12}, \tag{15}$$

where $R_v^{E11}, R_v^{E22}, R_v^{\nu12}, R_v^{\nu23}$, and R_v^{G12} are the regressions of the dataset of $E_{11}, E_{22}, \nu_{12}, \nu_{23}$, and G_{12} , respectively. The optimal ANN was then used to extend the dataset and

Fig. 7 ANN scheme for fibre and interphase parameters identification



creating new N samples required to accurately identify fibre and interphase parameters. This was conducted using LHS to create N random samples of the inputs within the range given in Table 3. These samples are then passed through the trained ANN to obtain the output. LH sampling method is appropriate for this work as it ensured a full coverage of the input sample space. A denser space is constructed by selection more points, which results in more reliable results. The optimum carbon fibre and interphase parameters were selected based on the smallest value obtained from Eq. (16)

$$\min F(x) = \frac{|E_{11}^f - E_{11}^{ANN}|}{E_{11}^f} + \frac{|E_{22}^f - E_{22}^{ANN}|}{E_{22}^f} + \frac{|v_{12}^f - v_{12}^{ANN}|}{v_{12}^f} + \frac{|v_{23}^f - v_{23}^{ANN}|}{v_{23}^f} + \frac{|G_{12}^f - G_{12}^{ANN}|}{G_{12}^f}, \tag{16}$$

where $E_{11}^f, E_{22}^f, v_{12}^f, v_{23}^f$ and G_{12}^f are the experimental effective elastic properties given in Table 2.

$E_{11}^{ANN}, E_{22}^{ANN}, v_{12}^{ANN}, v_{23}^{ANN}$, and G_{12}^{ANN} are corresponding effective elastic properties predicted by the ANN.

Finally, the identified parameters were used to calculate the effective elastic properties using the plug-in (Step 3) and results were compared with experimental data.

4.3 Results and discussion

The ANN which can predict fibre and fibre/matrix interphase parameters from micromechanical FE modelling dataset is designed. The number of total samples created by micromechanical FE modelling to train the ANN is 500. 350 of them is randomly assigned for training, 75 for validation set and the rest used for testing. The ANN is built and validated as explained in Sect. 4.2.1.

Table 4 presents some architecture samples used to verify the performance of the ANN in terms of R-value. Since 20

Table 4 Verification cases of the ANN architecture in terms of R-value for each output element

ANN architecture	R_v^{E11}	R_v^{E22}	R_v^{v12}	R_v^{v23}	R_v^{G12}	ΣR (Eq. 15)
7-38-52-5	0.998	0.948	0.995	0.933	0.815	4.689
7-39-51-5	0.998	0.961	0.997	0.957	0.688	4.601
7-40-50-5	0.994	0.949	0.997	0.942	0.740	4.623
7-41-49-5	0.998	0.932	0.997	0.931	0.768	4.626
7-42-48-5	0.997	0.958	0.996	0.954	0.865	4.771
7-43-47-5	0.998	0.944	0.997	0.955	0.723	4.617
7-44-46-5	0.980	0.923	0.979	0.946	0.796	4.624
7-45-45-5	0.998	0.933	0.996	0.929	0.727	4.584
7-46-44-5	0.999	0.964	0.998	0.954	0.690	4.605
7-47-43-5	0.998	0.943	0.995	0.918	0.751	4.605

runs are performed within each ANN structure to ensure the use of different data for each subset and due to the space restrictions of the paper, the table shows only the best performance run from the 20 runs. It can be seen from the table that the maximum R-value is attained when 42 neurons at the first hidden layer and 48 neurons at the second hidden layer are used.

Figure 8 shows the regression graphs for the target and the output of the verification data set only. This figure shows the closeness among the output data predicted by the ANN and the target data obtained from FEM. The dashed line in each subfigure represents the perfect results when the outputs equal targets. It can be seen that E_{11} , E_{22} , ν_{12} and ν_{23} are well predicted by ANN with an R-value between 0.91 and 0.99 and a regression slope (m) between 0.81 and 0.96. G_{12} is slightly less well predicted comparing to other effective elastic properties with an R-value and a regression slope of 0.86 and 1.08, respectively. Hence, the selected ANN is capable of providing a good correlation between the target and the output.

After training the ANN, the selected model with the highest R-value is used to generate N samples. It is found that 10,000 samples of the random input parameters generated by the LH sampling method are sufficient to produce a dense space. These new input data (E_{f2} , ν_{f12} , ν_{f23} , G_{f12} , T_i , K_m and K_{ss}) are then processed by the ANN to obtain the outputs (E_{11} , E_{22} , ν_{12} , ν_{23} , and G_{12}). The closest point of the new output to the experimental data is found through Eq. (16). The corresponding carbon fibre and interphase parameters of the closest point are given in Table 5.

Finally, the identified fibre and interphase parameters (micro-parameters) obtained from the ANN are used as input for the FE model to predict the effective properties of the UD lamina, i.e., macroscale level properties. The effective elastic properties calculated by the FE model using the identified parameters are given in Table 6 (second column). The table also shows a comparison between these properties and those predicted by the ANN and the experimental data. It can be seen that the effective elastic properties agree well with the experimental data with a maximum error of about 6%. This

error is mainly due to the using fixed value for E_{11} in the training of ANN.

5 Conclusions and future improvement

An ABAQUS® plug-in (*VDGen*) has been developed for generating virtual data for identifying the uncertain materials' properties in unidirectional (UD) lamina. In combination with artificial neural networks (ANNs), the data generated from the plug-in enable the determination of the relationship between fibre mechanical properties and fibre/matrix interphase parameters at microscale and the mechanical properties of a UD lamina at macroscale. Application of the plug-in to a T300/PR-319 UD lamina has shown very good agreement between the predictions and the experimental data when using the identified constituent properties.

A few improvements of the plug-in should be considered in the future. The current plug-in is designed to support the square and hexagonal unit cell fibre arrays. Micro-mechanical FE modelling of randomly distributed fibres in the matrix is essential when studying the failure of the composite lamina. However, using random fibre distribution causes arbitrary meshing condition on opposite RVE edges [20, 35]. Further development of the plug-in to support RVEs with randomly fibre distribution and capable of generating periodic mesh on the opposite edges is under investigation by the authors.

At this stage, the plug-in is only designed to calculate the effective elastic properties of a lamina. We aim to develop it further, so that it will be capable of conducting failure analysis under uniaxial, biaxial, and multiaxial loading conditions.

The effect of fibre shape has recently been subjected to intensive studies by means of computational micromechanics [11, 22]. The current core Python scripts of the plug-in will be developed further, so that RVEs with different fibre shapes can be automatically generated.

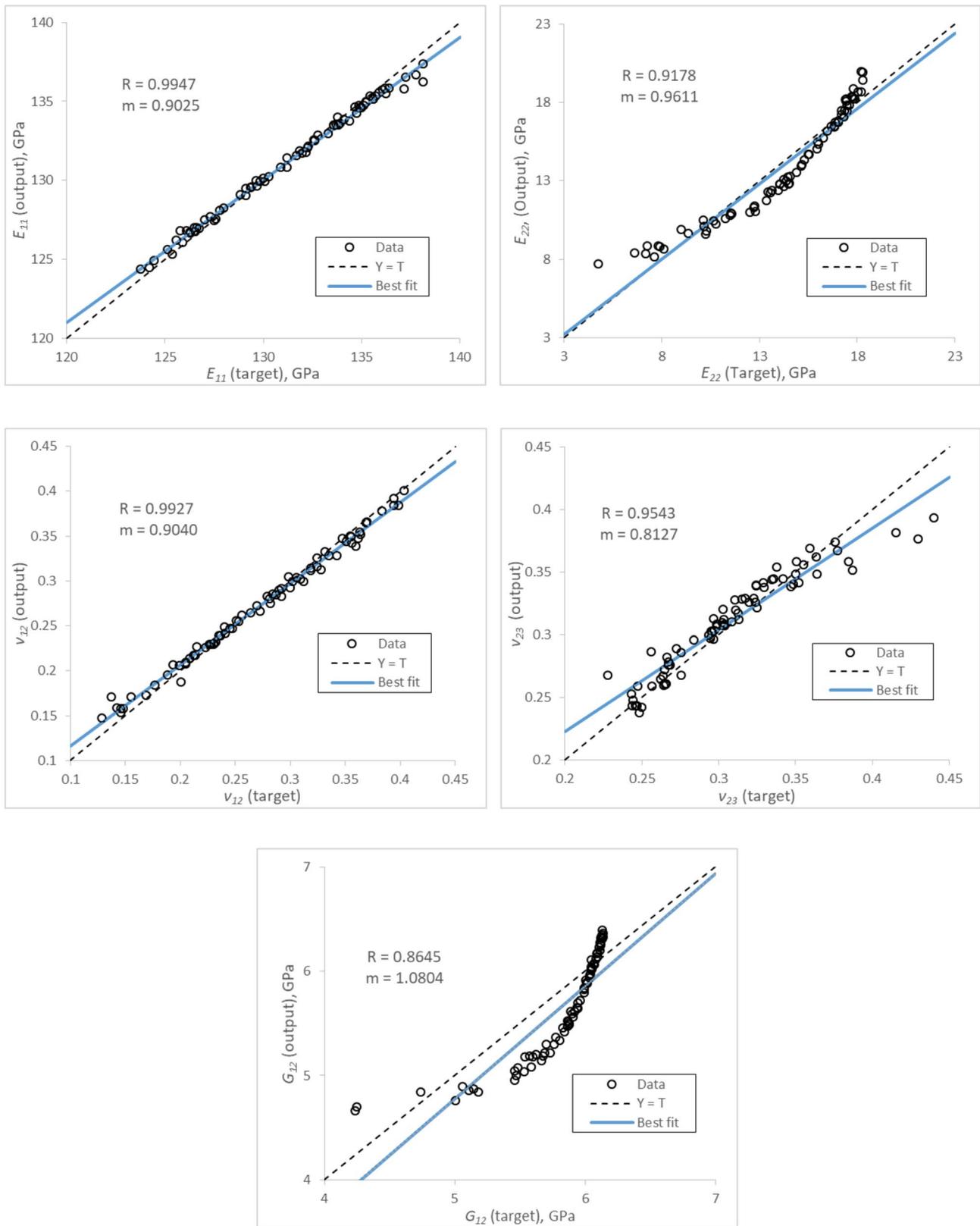


Fig. 8 ANN predictions versus FEM results for **a** E_{11} , **b** E_{22} , **c** ν_{12} , **d** ν_{23} , and **e** G_{12}

Table 5 Identified parameters for fibre and interphase region

E_{f2} (GPa)	ν_{f12}	ν_{f23}	G_{f12} (GPa)	T (μm)	K_{m1} (GPa/mm)	K_{s1} (GPa/mm)
23.66	0.249	0.437	59.57	0.121	829.81	132.50

Table 6 Macro-level properties predicted by ANN and FE model using identified parameters and experimental data [15]

Predicted by ANN					Predicted by FE model using identified parameters					Experimental data				
E_{11}	E_{22}	ν_{12}	ν_{23}	G_{12}	E_{11}	E_{22}	ν_{12}	ν_{23}	G_{12}	E_{11}	E_{22}	ν_{12}	ν_{23}	G_{12}
129.46	10.81	0.279	0.386	5.24	129.61	11.54	0.279	0.402	5.71	138	11	0.28	0.4	5.5

Author contributions YI: conceptualization, methodology, writing—original draft. LW: conceptualization, methodology, writing—review & editing. JC: conceptualization, methodology, writing—review & editing. J: supervision, writing—review & editing. DY: conceptualization, methodology, supervision, writing—original draft.

Data availability The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study. Source codes for the plug-in will be provided as supplementary documents upon acceptance for publication.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abaqus Users Manual Version 6.13, Dassault Systèmes Simulia Corp., Providence, Rhode Island, USA
2. Arteiro A, Catalanotti G, Melro A, Linde P, Camanho PP (2014) Micro-mechanical analysis of the in situ effect in polymer composite laminates. *Compos Struct* 116:827–840
3. Ballard MK, McLendon WR, Whitcomb JD (2014) The influence of microstructure randomness on prediction of fiber properties in composites. *J Compos Mater* 48:3605–3620
4. Beale HD, Demuth HB, Hagan M (1996) *Neural network design*. Pws, Boston
5. Chamis CC (1989) Mechanics of composite materials: past, present, and future. *J Compos Tech Res* 11:3–14
6. Daniel IM, Ishai O (2006) *Engineering mechanics of composite materials*
7. El Hachemi M, Koutsawa Y, Nasser H et al (2016) An intuitive computational multi-scale methodology and tool for the dynamic modelling of viscoelastic composites and structures. *Compos Struct* 144:131–137
8. Giner E, Vercher A, Marco M, Arango C (2015) Estimation of the reinforcement factor ξ for calculating the transverse stiffness E_2 with the Halpin-Tsai equations using the finite element method. *Compos Struct* 124:402–408
9. Gulbag A, Temurtas F (2006) A study on quantitative classification of binary gas mixture using neural networks and adaptive neuro-fuzzy inference systems. *Sens Actuators B Chem* 115:252–262
10. Halpin JC, Tsai SW (1969) Effects of environmental factors on composite materials. Air Force Materials Lab Wright-Patterson AFB OH
11. Herráez M, González C, Lopes C et al (2016) Computational micromechanics evaluation of the effect of fibre shape on the transverse strength of unidirectional composites: an approach to virtual materials design. *Compos A Appl Sci Manuf* 91:484–492
12. Kaddour A, Hinton M (2012) Input data for test cases used in benchmarking triaxial failure theories of composites. *J Compos Mater* 46:2295–2312
13. Karadeniz ZH, Kumlutas D (2007) A numerical study on the coefficients of thermal expansion of fiber reinforced composite materials. *Compos Struct* 78:1–10
14. Lim JH, Henry M, Hwang D-S, Sohn D (2016) Numerical prediction of fiber mechanical properties considering random microstructures using inverse analysis with quasi-analytical gradients. *Appl Math Comput* 273:201–216
15. Lu J, Zhu P, Ji Q, Feng Q, He J (2014) Identification of the mechanical properties of the carbon fiber and the interphase region based on computational micromechanics and Kriging metamodel. *Comput Mater Sci* 95:172–180
16. MacKrell A (2017) Multiscale composite analysis in Abaqus: theory and motivations. *Reinf Plast* 61:153–156
17. McCarthy C, Vaughan T (2012) COMM Toolbox: A MATLAB toolbox for micromechanical analysis of composite materials. *J Compos Mater* 46:1715–1729
18. Melro A, Camanho P, Pires FA, Pinho S (2013) Micromechanical analysis of polymer composites reinforced by unidirectional fibres: Part I-Constitutive modelling. *Int J Solids Struct* 50:1897–1905
19. Mori T, Tanaka K (1973) Average stress in matrix and average elastic energy of materials with misfitting inclusions. *Acta Metall* 21:571–574
20. Nguyen V-D, Béchet E, Geuzaine C, Noels L (2012) Imposing periodic boundary condition on arbitrary meshes by polynomial interpolation. *Comput Mater Sci* 55:390–406

21. Omairey SL, Dunning PD, Sriramula S (2019) Development of an ABAQUS plugin tool for periodic RVE homogenisation. *Engineering with Computers* 35:567–577
22. Pathan M, Tagarielli V, Patsias S (2017) Effect of fibre shape and interphase on the anisotropic viscoelastic response of fibre composites. *Compos Struct* 162:156–163
23. Raschka S, Mirjalili V (2019) *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2*: Packt Publishing Ltd
24. Riaño L, Joliff Y (2019) An AbaqusTM plug-in for the geometry generation of Representative Volume Elements with randomly distributed fibers and interphases. *Compos Struct* 209:644–651
25. Romanowicz M (2012) A numerical approach for predicting the failure locus of fiber reinforced composites under combined transverse compression and axial tension. *Comput Mater Sci* 51:7–12
26. Rupnowski P, Gentz M, Sutter J, Kumosa M (2005) An evaluation of the elastic properties and thermal expansion coefficients of medium and high modulus graphite fibers. *Compos A Appl Sci Manuf* 36:327–338
27. Sun C, Vaidya R (1996) Prediction of composite properties from a representative volume element. *Compos Sci Technol* 56:171–179
28. Tchalla A, Belouettar S, Makradi A, Zahrouni H (2013) An ABAQUS toolbox for multiscale finite element computation. *Compos B Eng* 52:323–333
29. Totry E, González C, LLorca J (2008) Failure locus of fiber-reinforced composites under transverse compression and out-of-plane shear. *Compos Sci Technol* 68:829–839
30. Totry E, González C, LLorca J (2008) Prediction of the failure locus of C/PEEK composites under transverse compression and longitudinal shear through computational micromechanics. *Compos Sci Technol* 68:3128–3136
31. Trias D, Costa J, Mayugo J, Hurtado J (2006) Random models versus periodic models for fibre reinforced composites. *Comput Mater Sci* 38:316–324
32. Tsai SW, Melo JDD (2014) An invariant-based theory of composites. *Compos Sci Technol* 100:237–243
33. Wan L, Ismail Y, Zhu C, Zhu P, Sheng Y, Liu J, Yang DM (2020) Computational micromechanics-based prediction of the failure of unidirectional composite lamina subjected to transverse and in-plane shear stress states. *J Compos Mater* 2020:0021998320918015
34. Yang L, Yan Y, Liu Y, Ran Z (2012) Microscopic failure mechanisms of fiber-reinforced polymer composites under transverse tension and compression. *Compos Sci Technol* 72:1818–1825
35. Zhang C, Curiel-Sosa J, Bui TQ (2017) Comparison of periodic mesh and free mesh on the mechanical properties prediction of 3D braided composites. *Compos Struct* 159:667–676

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.