Steel and Composite Structures, *Vol. 25, No. 6 (2017) 663-670*
DOI: https://doi.org/10.12989/scs.2017.25.6.663

Failure estimation of the composite laminates using machine learning techniques

Alexandru Şerban*

Faculty of Mechanical Engineering, University of Gh. Asachi, Iasi, 700050, Romania

(*Received January 7, 2017, Revised August 20, 2017, Accepted September 2, 2017*)

Abstract. The problem of layup optimization of the composite laminates involves a very complex multidimensional solution space which is usually non-exhaustively explored using different heuristic computational methods such as genetic algorithms (GA). To ensure the convergence to the global optimum of the applied heuristic during the optimization process it is necessary to evaluate a lot of layup configurations. As a consequence the analysis of an individual layup configuration should be fast enough to maintain the convergence time range to an acceptable level. On the other hand the mechanical behavior analysis of composite laminates for any geometry and boundary condition is very convoluted and is performed by computational expensive numerical tools such as finite element analysis (FEA). In this respect some studies propose very fast FEA models used in layup optimization. However, the lower bound of the execution time of FEA models is determined by the global linear system solving which in some complex applications can be unacceptable. Moreover, in some situation it may be highly preferred to decrease the optimization time with the cost of a small reduction in the analysis accuracy. In this paper we explore some machine learning techniques in order to estimate the failure of a layup configuration. The estimated response can be qualitative (the configuration fails or not) or quantitative (the value of the failure factor). The procedure consists of generating a population of random observations (configurations) spread across solution space and evaluating using a FEA model. The machine learning method is then trained using this population and the trained model is then used to estimate failure in the optimization process. The results obtained are very promising as illustrated with an example where the misclassification rate of the qualitative response is smaller than 2%.

Keywords: failure estimation; layup optimization; machine learning; finite element analysis; numerical analysis

1. Introduction

The mechanical behavior and failure mechanism of fiber reinforced plastics are highly influenced by the fibers orientation. This particularity becomes critical when we bring into question layered composite materials built by stacking up unidirectional or bidirectional reinforced layers with different orientations and possibly different mechanical properties as in the case of hybrid composite laminates-i.e., mixing carbon and Kevlar reinforced layers. The main cause of this characteristic is the orthotropic behavior exhibited by the reinforced layers.

For different type of applications it is desirable to use a composite laminate which have a small weight (a critical example is the aeronautics industry) and/or the material cost is kept at a low level. This single/multi objective goal is usually achieved by an optimization process with the purpose to find the configuration with the smallest number of layers or the smallest cost able to carry out the loads for which it was designed. The layup optimization is very complex due to the multidimensional search space which cannot be exhaustively explored because of the computational limitations. For this reason, heuristic methods are usually addressed of which the most reliable are the genetic algorithms (Topal 2013, Cho *et al.* 2013, Ghiasi *et al.* 2009, Ghiasi *et al.* 2010, Bagheri *et al.* 2011, Badallo *et al.* 2013, Chatzi *et al.* 2011, Montagnier and Hochard 2013). During the optimization process a lot of configurations are evaluated with the purpose to converge to the global optimum. In order to keep the convergence time range to an acceptable level it becomes mandatory that the evaluation of a single configuration to be very fast.

On the other hand, the mechanical behavior analysis of the composite laminate requires complex numerical tools like finite element analysis (Sliseris and Rocens 2013, Lee *et al.* 2012, Yong *et al.* 2010, Pohlak *et al.* 2010, Murugan *et al.* 2012, Yong *et al.* 2008, Falzon and Faggiani 2012). Because FEA is a time consuming method a lot of optimization studies are restricted to simple geometries (i.e., rectangular) and simple boundary condition for which analytical closed form solutions are available (Kayikci and Sonmez 2012, Sharma *et al.* 2014, Papadopoulos and Kassapoglou 2007, Park *et al.* 2008). To overcome this issue Serban (2016) proposed a very fast finite element model specially developed for layup optimization based on the observation that a lot of calculations can be reused for all the configurations evaluated during the optimization process. However, the lower bound of the execution time of a FEA model is determined by the global linear system solving which might be unacceptable big in some complex applications. Also, there are situations in which it is highly

*Corresponding author, Ph.D.
E-mail: ser.alexandru@gmail.com

ISSN: 1229-9367 (Print), 1598-6233 (Online)

preferred to shorten the optimization time with the cost of a small reduction in the analysis accuracy.

Motivated by the advances in machine learning, we propose in this paper a novel method for estimating the failure of a composite laminate with a very small computational cost while maintaining the precision level from FEA. To achieve this purpose we try to model the mechanical behavior of the composite laminate using various supervised learning algorithms. In this respect we experimented with regression and regularization algorithms, instance-based algorithms, dimensionality reduction algorithms, decision trees, artificial neural networks and support vector machines (Hastie *et al.* 2009, Bishop 2006, Murphy 2012, Mohri *et al.* 2012). At the end of this paper we demonstrate with a practical example that our procedure performs very well having a misclassification rate smaller than 2 % in predicting qualitative failure.

2. The procedure

The procedure proposed in this paper has the purpose to build and train a statistical model using various supervised learning algorithms. The scope of the model is to accurately predict the failure factor for the layup configurations having the same topology and boundary conditions-as it is the case of the layup optimization process. The procedure has three main steps:

1. Randomly generate a population of layup configurations (observations) spread across the convoluted search space. For any random observation, the number of layers is chosen in a plausible range, the orientation of each layer is chosen from a set of discrete orientations and the material of each layer is chosen from a set of materials (defined by their mechanical properties).

2. Compute the failure factor for each observation in the population generated at step 1 using a fast FEA model, for instance the model proposed by Serban (2016) which was specially developed for the layup optimization problem.

3. Train the model on the population using the failure factor as the response variable. Regarding the predictors, a detailed description of their derivation will be given in the next section.

To summarize, in the training process (step 3) it is used a random population of layup configurations (step 1) with the failure factors accurately computed using FEA (step 2). The trained model is subsequently used to predict the failure for the new observations coming from the layup optimization process. The estimated response can be either qualitative (the configuration fails or not) or quantitative (the value of the failure factor). We can mention the following advantages of our procedure:

1. The time to predict the failure of a layup configuration using the trained model is negligible in comparison with the time required by the FEA model.

2. The trained model can be successfully reused in multiple sessions of genetic optimization. As an example we can mention the convex multi objective optimization based on two conflicting objectives: weight and cost. In order to obtain a pareto-optimal set of layup configurations, the balance between the contribution of weight and cost to the objective function is modified using small increments. This means that the optimization procedure runs for a lot of times, maybe hundreds.

3. The trained model need not necessarily be updated when some small changes occur (i.e., small adjustment of Young modulus or layer thickness for a certain material).

4. The time spent to evaluate the random population is significantly smaller than the time needed by a single session of genetic optimization using the FEA model. Moreover, this time is not wasted because the random population can be used to generate the initial population for GA by selecting the individuals with the best fitness results. Having a healthy initial population increases the convergence speed of the optimization procedure.

Obviously, as a drawback we have to mention the reduction in the analysis accuracy. However, we will prove that this accuracy reduction is fairly acceptable.

3. Strategy and predictors

We shortly review a finite element mathematical model-also used by Serban (2016) - in order to explain how we chose the *basic set* of our predictors. Usually, in the case of composite laminates the numerical analysis is carried out using two-dimensional theories, like *first-order shear deformation theory* (FSDT).

FSDT assumes that the form of $(\Delta_x, \Delta_y, \Delta_z)$displacements is

$$\begin{cases} \Delta_x(x,y,z,t) = \Delta_x^m(x,y,t) + z \cdot \phi_y(x,y,t) \\ \Delta_y(x,y,z,t) = \Delta_y^m(x,y,t) + z \cdot \phi_x(x,y,t) \\ \Delta_z(x,y,z,t) = \Delta_z^m(x,y,t) \end{cases} \quad (1)$$

where $(\Delta_x^m, \Delta_y^m, \Delta_z^m, \phi_x, \phi_y)$ represents the displacements and rotations around O_x and O_y axes of the midplane points of the laminate.

The compact form of the FSDT finite element model is

$$[K^e]\{\Delta^e\} - \{F^e\} = \{0\} \quad (2)$$

where:

- (K^e) represents the element stiffness matrix;
- (Δ^e) represents the displacements and rotations for the element nodes at the laminate midplane. Δ^e are the unknowns in the system (2);
- (F^e) represents the force vector associated with the element and is computed by applying the Neumann boundary condition.

System (2) can be written in expanded form

$$\sum_{\beta=1}^{5}\sum_{j=1}^{n} K_{ij}^{\alpha\beta}\Delta_j^{\beta} - F_i^{\alpha} = 0, \quad \left(\alpha = \overline{1,5} \; and \; i = \overline{1,n}\right) \quad (3)$$

where:

- (n) is the number of nodes of the element;
- $K_{ij}^{\alpha\beta}$ are the blocks of the elemental stiffness matrix:

$$K_{ij}^{1\alpha} = \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} N_{1j}^{\alpha} + \frac{\partial \psi_i}{\partial y} N_{6j}^{\alpha} \right) dx\, dy$$

$$K_{ij}^{2\alpha} = \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} N_{6j}^{\alpha} + \frac{\partial \psi_i}{\partial y} N_{2j}^{\alpha} \right) dx\, dy$$
$$K_{ij}^{3\alpha} = \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} Q_{1j}^{\alpha} + \frac{\partial \psi_i}{\partial y} Q_{2j}^{\alpha} \right) dx\, dy$$
$$K_{ij}^{4\alpha} = \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} M_{1j}^{\alpha} + \frac{\partial \psi_i}{\partial y} M_{6j}^{\alpha} + \psi_i Q_{1j}^{\alpha} \right) dx\, dy \quad (4)$$
$$K_{ij}^{5\alpha} = \int_{\Omega^e} \left(\frac{\partial \psi_i}{\partial x} M_{6j}^{\alpha} + \frac{\partial \psi_i}{\partial y} M_{2j}^{\alpha} + \psi_i Q_{2j}^{\alpha} \right) dx\, dy$$

where:

- (ψ_i)represents the *Lagrange* shape functions of the element;
- $(N_{ij}^{\alpha})$are defined as

$$N_{1j}^1 = A_{11}\frac{\partial \psi_j}{\partial x} + A_{16}\frac{\partial \psi_j}{\partial y},\quad N_{1j}^2 = A_{16}\frac{\partial \psi_j}{\partial x} + A_{12}\frac{\partial \psi_j}{\partial y}$$
$$N_{1j}^4 = B_{11}\frac{\partial \psi_j}{\partial x} + B_{16}\frac{\partial \psi_j}{\partial y},\quad N_{1j}^5 = B_{16}\frac{\partial \psi_j}{\partial x} + B_{12}\frac{\partial \psi_j}{\partial y}$$
$$N_{2j}^1 = A_{12}\frac{\partial \psi_j}{\partial x} + A_{26}\frac{\partial \psi_j}{\partial y},\quad N_{2j}^2 = A_{26}\frac{\partial \psi_j}{\partial x} + A_{22}\frac{\partial \psi_j}{\partial y}$$
$$N_{2j}^4 = B_{12}\frac{\partial \psi_j}{\partial x} + B_{26}\frac{\partial \psi_j}{\partial y},\quad N_{2j}^5 = B_{26}\frac{\partial \psi_j}{\partial x} + B_{22}\frac{\partial \psi_j}{\partial y} \quad (5)$$
$$N_{6j}^1 = A_{16}\frac{\partial \psi_j}{\partial x} + A_{66}\frac{\partial \psi_j}{\partial y},\quad N_{6j}^2 = A_{66}\frac{\partial \psi_j}{\partial x} + A_{26}\frac{\partial \psi_j}{\partial y}$$
$$N_{6j}^4 = B_{16}\frac{\partial \psi_j}{\partial x} + B_{66}\frac{\partial \psi_j}{\partial y},\quad N_{6j}^5 = B_{66}\frac{\partial \psi_j}{\partial x} + B_{26}\frac{\partial \psi_j}{\partial y}$$

- (M_{ij}^{α}) are defined as

$$M_{1j}^1 = B_{11}\frac{\partial \psi_j}{\partial x} + B_{16}\frac{\partial \psi_j}{\partial y},\quad M_{1j}^2 = B_{16}\frac{\partial \psi_j}{\partial x} + B_{12}\frac{\partial \psi_j}{\partial y}$$
$$M_{1j}^4 = D_{11}\frac{\partial \psi_j}{\partial x} + D_{16}\frac{\partial \psi_j}{\partial y},\quad M_{1j}^5 = D_{16}\frac{\partial \psi_j}{\partial x} + D_{12}\frac{\partial \psi_j}{\partial y}$$
$$M_{2j}^1 = B_{12}\frac{\partial \psi_j}{\partial x} + B_{26}\frac{\partial \psi_j}{\partial y},\quad M_{2j}^2 = B_{26}\frac{\partial \psi_j}{\partial x} + B_{22}\frac{\partial \psi_j}{\partial y} \quad (6)$$
$$M_{2j}^4 = D_{12}\frac{\partial \psi_j}{\partial x} + D_{26}\frac{\partial \psi_j}{\partial y},\quad M_{2j}^5 = D_{26}\frac{\partial \psi_j}{\partial x} + D_{22}\frac{\partial \psi_j}{\partial y}$$
$$M_{6j}^1 = B_{16}\frac{\partial \psi_j}{\partial x} + B_{66}\frac{\partial \psi_j}{\partial y},\quad M_{6j}^2 = B_{66}\frac{\partial \psi_j}{\partial x} + B_{26}\frac{\partial \psi_j}{\partial y}$$
$$M_{6j}^4 = D_{16}\frac{\partial \psi_j}{\partial x} + D_{66}\frac{\partial \psi_j}{\partial y},\quad M_{6j}^5 = D_{66}\frac{\partial \psi_j}{\partial x} + D_{26}\frac{\partial \psi_j}{\partial y}$$

- (Q_{ij}^{α}) are defined as

$$Q_{1j}^3 = A_{55}\frac{\partial \psi_j}{\partial x} + A_{45}\frac{\partial \psi_j}{\partial y},\quad Q_{2j}^3 = A_{45}\frac{\partial \psi_j}{\partial x} + A_{44}\frac{\partial \psi_j}{\partial y}$$
$$Q_{1j}^4 = A_{55}\psi_j,\quad Q_{2j}^4 = A_{45}\psi_j \quad (7)$$
$$Q_{1j}^5 = A_{45}\psi_j,\quad Q_{2j}^5 = A_{44}\psi_j$$

where (A_{ij}, D_{ij}, B_{ij}) denotes the components of the *extensional stiffness matrix*, *bending stiffness matrix* and *bending-extensional coupling stiffness matrix*, respectively. The blocks of the elemental force vector are computed using

$$F_i^1 = \int_{\Gamma^e} \psi_i\, N_n ds,\quad F_i^2 = \int_{\Gamma^e} \psi_i\, N_{ns} ds$$
$$F_i^3 = \int_{\Gamma^e} \psi_i\, Q_n ds + \int_{\Omega^e} \psi_i\, q\, dx\, dy$$
$$F_i^4 = \int_{\Gamma^e} \psi_i\, M_n ds,\quad F_i^5 = \int_{\Gamma^e} \psi_i\, M_{ns} ds \quad (8)$$

where (N_n, N_{ns}, M_n, M_{ns}, Q_n, q) represent the corresponding edge normal and tangential forces and moments, transverse force and transverse distributed load, respectively.

The global linear system is assembled using all the elements from the topology and provides the solutions for $(\Delta_x^m, \Delta_y^m, \Delta_z^m, \phi_x, \phi_y)$ at each node at the laminate midplane. Using the interpolation functions, Eq. (1), von Karman strain-displacements and Hooke's law it can be computed the displacements, strains and stresses at each point in the entire laminate. In other words the solutions from the laminate midplane are extrapolated to the entire laminate using FSDT assumptions.

Now we define (λ) to be the failure factor which is calculated using various failure criteria such as maximum strain, maximum stress, Tsai-Hill or Tsai-Wu. For example in the case of maximum strain criterion the laminate failure occurs when the value of λ is bigger than 1.

In the case of maximum strain criterion, λ is computed as

$$\lambda = \max_i \lambda_i^l \quad (9)$$

where (λ_i^l) indicate the maximum failure factor at layer (*i)* and is computed as follow

$$\lambda_i^l = \max_j \lambda_j^{li,n} \quad (10)$$

where $(\lambda_j^{li,n})$ indicate the value of the failure factor at layer (*i*) and node (*j*).

From the finite element mathematical model we can easily observe that for different layup configurations:

1. Lagrange shape functions remain the same because the topology doesn't change;

2. Global force vector remains the same because the Neumann boundary conditions doesn't change and due to observation 1.

3. The coefficients of the global stiffness matrix can be obtained by linear combination of (A_{ij}, D_{ij}, B_{ij}) coefficients and some integrals over the elements. But the values of these integrals remain constant due to observation 1.

The layup configuration implies that we can select as a *basic set* of predictors the (A_{ij}, D_{ij}, B_{ij}) coefficients because they strongly depend to the layup configuration and we expect that they explains a great proportion of the response variability. This expectation is actually confirmed with the example considered in this paper. We refer to this set of predictors as a *basic set* because it is not necessary to include all the coefficients in the statistical model - instead we include some polynomial and interaction terms. The procedure to select the *actual set* of predictors based on (A_{ij}, D_{ij}, B_{ij}) coefficients is described in the next section.

4. Predictors selection

As we already mention the *actual set* of predictors which are actually used to train the statistical models is derived from the *basic set* which contains the (A_{ij}, D_{ij}, B_{ij})

coefficients. Some of these coefficients are highly correlated and can cause severe problems for few statistical methods like linear regression by the induced collinearity. Also, having correlated predictors doesn't help the model to better explain the response variability instead can lead to overfitting the training observations. On the other hand we observe that the models perform better when polynomial coefficients and interaction terms are considered.

Given the above observations we developed an intuitive and relatively simple procedure for predictors selection which is described in the follow steps:

1. An *extended set* of predictors is built as a union of the *basic set*, the set of quadratic coefficients and the set of all interaction terms.

2. *Backward stepwise selection* is applied to a random subset of 90 % of the training observations using regression on the *extended set* of predictors.

3. Step 2 is repeated few times (5 or 10) with different random subsets. At each repetition a minimal set *o* predictors is selected applying cross validation.

4. *Best subset selection* is applied to the union of all the minimal sets of predictors found in step 3. If this union contains too many elements then *forward stepwise selection* can be applied.

The predictors selected at step 4 along with the response variable (λ-failure factor) will be used in the statistical models. However, a log transformation is considered for λ in order to obtain a normal distribution as illustrated in the numerical example considered in this paper.

5. Statistical models

As artificial intelligence has seen tremendous progress it was naturally integrated in engineering (Ozcan *et al*. 2017, Kocak *et al*. 2015, Samui 2011a, b). In this section we present a very short review of the machine learning techniques that we have experimented to model the failure response of a certain layup configuration. More details about these techniques can be found in (Hastie *et al*. 2009, Bishop 2006, Murphy 2012, Mohri *et al*. 2012). Also, we remember that the response of a certain model can be either qualitative (the configuration fails or not) or quantitative (an estimation of the failure factor).

We review some common notations used in this section:

- *N*-number of observations-configurations in our case;
- *p*-number of predictors;
- λ_i-failure factor of the *i*-th observation;
- $\hat{\lambda}_i$-predicted failure factor of the *i*-th observation;
- x_i-vector of predictors with *p* components for the i^{th} observation.

5.1 Regression algorithms

Linear regression model is based on the assumption that the relationship between the response variable and the predictors is linear

$$\lambda = \beta_0 + \sum_{j=1}^{p} \beta_j x_j + \epsilon \tag{11}$$

where β_j are the regression coefficients (β_0 is the intercept) and ϵ is the error term which is assumed to have a normal distribution and to be unrelated to the predictors *x*.

According to the linear model the predicted response is

$$\hat{\lambda} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j \tag{12}$$

The regression coefficients are computed by minimizing the residual sum of squares (*RSS*)

$$RSS(\beta) = \sum_{i=1}^{N} (\lambda_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 \tag{13}$$

Ridge regression is a regularization model which shrinks the regression coefficients by adding a penalty on their size and minimizes the penalized sum of squares

$$\hat{\beta}^r = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(\lambda_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \alpha \sum_{j=1}^{p} \beta_j^2 \right\} \tag{14}$$

where $\alpha \geq 0$ is a factor which controls the amount of shrinkage and is usually selected by cross validation.

The lasso regression is also a regularization model very similar to ridge regression which shrinks the regression coefficients by minimizing the penalized sum of squares

$$\hat{\beta}^l = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(\lambda_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \alpha \sum_{j=1}^{p} |\beta_j| \right\} \tag{15}$$

where $\alpha \geq 0$ is a factor which controls the amount of shrinkage. The main difference between lasso and ridge regression is the penalty - ridge penalty is the L_2: $\sum_{j=1}^{p} \beta_j^2$ while lasso penalty is the L_1: $\sum_{j=1}^{p} |\beta_j|$.

Logistic regression is a classification model which estimates the probability that a response belongs to a certain class given the predictors vector. The model is based on the logistic function

$$P(\lambda > \lambda_{crit} | x) = \frac{e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_j}}{1 + e^{\beta_0 + \sum_{j=1}^{p} \beta_j x_j}} \tag{16}$$

where λ_{crit} is the critical value for the failure factor. In order to fit the model (16) it is used the maximum likelihood method.

5.2 Instance based algorithms

K nearest neighbors is a method which uses those observations from the training set closest to the predictors vector. The response is predicted as

$$\hat{\lambda}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} \lambda_i \tag{17}$$

where $N_k(x)$ represents the neighborhood of *x* containing the closest *k* observation. Euclidean distance is used as a distance measure for the observations. The value of *k* is usually selected using cross validation.

In case of classification, *k* nearest neighbors method

uses Bayes rule and assigns an observation to the most common class from the neighborhood

$$P(\lambda > \lambda_{crit}|x) = \frac{1}{k}\sum_{x_i \in N_k(x)} 1_{\lambda_i > \lambda_{crit}} \quad (18)$$

where $1_{\lambda_i>\lambda_{crit}}$ is and indicator which returns 1 if $\lambda_i > \lambda_{crit}$ and 0 otherwise.

5.3 Dimensionality reduction algorithms

Principal component regression uses first $M<p$ principal components in a linear regression setup such as (12). The idea is based on the observation that often a small number of principal components suffices to explain the relationship between the response and the predictors. Also, it is assumed that the directions in which the predictors show the most variation are the directions associated with the response.

The regression model can be summarized as

$$\hat{\lambda} = \theta_0 + \sum_{m=1}^{M} \theta_m Z_m \quad (19)$$

where Z_m are the principal components defined as a linear combination of the initial set of predictors

$$Z_m = \sum_{j=1}^{p} \emptyset_j^m x_j \quad (20)$$

Partial least squares is very similar to principal component regression having the same form as in (19) with the main difference that the reduced set of features Z_m is obtained in a supervised way making use of the response variable λ. Also, this is the reason that the assumption of direction variability from principal components regression setup can be eliminated.

5.4 Discriminant analysis

Linear discriminant analysis is a classification model which uses the Bayes rule to assign an observation to the most likely class with the assumptions that the predictors vector is drawn from a multivariate normal distribution and all the *k* classes have the same covariance matrix. The posterior probability that an observation belongs to a certain *k* class is modeled as follow

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + \log\pi_k \quad (21)$$

where μ_k is the class specific mean vector, Σ is the common covariance matrix and π_k is the prior probability that an observation belongs to the *k* class.

Quadratic discriminant analysis is similar to the linear discriminant analysis except that the assumption of all *k* classes having the same covariance matrix is eliminated. That being said, the posterior probability that an observation belongs to a certain *k* class is modeled as follow

$$\delta_k(x) = -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k + \log\pi_k \quad (22)$$

For our problem we define two classes: $\lambda \leq \lambda_{crit}$ and $\lambda > \lambda_{crit}$.

5.5 Decision tree ensemble algorithms

Random forest is a method used for both classification and regression which involves the construction of *B* decision/regression trees grown deep and not pruned. The trees are generated based on *B* bootstrapped training sets obtained from the original training set. Also, at each split in the tree construction algorithm it is considered only a random subset of $m<p$ predictors (typically $m = \sqrt{p}$) with the purpose to obtain high diversity. The estimated value for a given observation is then computed by averaging the estimations of all *B* trained trees (23-for regression) or by the majority vote (for classification)

$$\hat{\lambda}(x) = \frac{1}{B}\sum_{i=1}^{B} \lambda_i(x) \quad (23)$$

where $\lambda_i(x)$ is the estimated value with the i^{th} tree for the predictors vector *x*.

Boosting with decision/regression trees involves the cascaded construction of *B* small trees each one fitted to the current updated residuals. With this method the quality of the fit is slowly improving from the previous to next tree and the risk of overfitting is eliminated. The method is controlled by three parameters: *B*-the number of trees, α-shrinkage parameter which controls the rate at which boosting learns and *d*-the number of splits in each tree (usually a small positive integer).

5.6 Support vector machines

They were invented by Vladimir N. Vapnik and Alexey Ya. Chervonenkis. They are an extension of the support vector classifier which are using kernels to enlarge the feature space and to obtain a non-linear decision boundary. For our problem the radial kernel (24) proves to have the best results

$$K(x, x') = e^{-\gamma\sum_{j=1}^{p}(x_j - x'_j)^2} \quad (24)$$

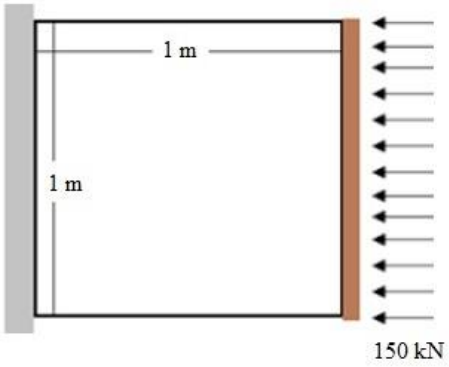
where γ is a positive constant. Even support vector machines are naturally binary classifiers it can be extended for problems involving more than two classes or quantitative estimation.

5.7 Artificial neural networks

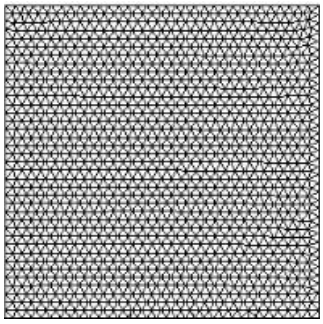
They are algorithms inspired from the neural structure of the cortex. Here we use back-propagation neural networks. The name comes from the backwards propagation of errors which is a strategy that performs *gradient descent* over the vector space of solutions with the purpose to find the global optimum along the steepest vector of the error surface.

6. Numerical example

In this section we present the results obtained with our proposed method using various machine learning algorithms. For our numerical example, the geometry and



(a) Geometry and boundary condition



(b) mesh

Fig. 1 Numerical example

Table 1 Mechanical properties of a carbon fiber lamina

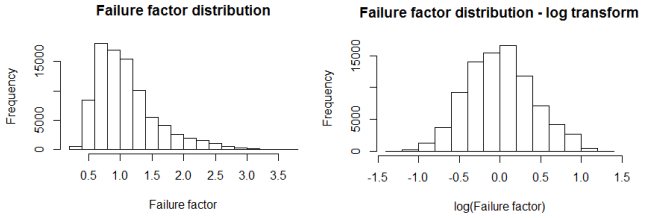
E_1 (GPa)	E_2 (GPa)	G_{12} (GPa)	G_{23} (GPa)	G_{13} (GPa)	ϑ_{12} (-)	ϑ_{21} (-)
147	10.3	7	3.7	7	0.27	0.0189
h (mm)	ρ (kg/m^3)	UTS_t^1 (%)	UTS_c^1 (%)	UTS_t^2 (%)	UTS_c^2 (%)	UTS^{12} (%)
0.5	1600	1.05e-03	0.85e-03	0.5e-03	2.5e-03	1.4e-03

boundary condition are illustrated in Fig. 1 along with FEA mesh.

We simulate a square plate with the edge length equal to 1 m, simply supported with one edge clamped and a compression load applied on the opposite edge. The layup consist of 10 carbon fiber plies symmetrically distributed around the laminate midplane having random discrete orientations from −90° to +90° with increments of 15°. In Table 1 are presented the mechanical properties of the carbon fiber lamina.

Where (E, G, ϑ) represents Young modulus, shear modulus and Poisson ratio, respectively and (h, ρ, UTS) represents layer thickness, material density and ultimate tensile strain at traction and compression, respectively.

We generate a population of 86953 observations with random plies orientations and evaluate the failure factors using FEA model from Serban (2016) and maximum strain criterion. This population is split into a training dataset and a test dataset. The training dataset is used to train the machine learning algorithms while the test dataset is used to evaluate the performance of the trained algorithms. We apply a log transformation to the failure factors in order to normalize its distribution. This transformation is illustrated in Fig. 2.



(a) (b)

Fig. 2 (a) λ distribution, (b) log λ distribution

Table 2 Classification results on the test dataset

Algorithm	False *Not Fail* count [-]	False *Fail* count [-]	Classification error rate [%]	Algorithm parameters
K nearest neighbor	563	462	1.33	K=1
Random forest	543	710	1.62	m=3, B=500
Support vector machines	716	616	1.73	γ=1.9, cost=300
Artificial neural networks	2113	2171	5.56	2 hidden layers with 5 and 3 neurons
Logistic regression	2419	2581	6.49	-
Linear discriminant analysis	3501	2481	7.77	-
Quadratic discriminant analysis	3725	3268	9.08	-

Based on the fact that the laminates are symmetrical around the midplane all the B_{ij} will be equal to 0 and consequently can be eliminated from the basic set of predictors. Using the method described in the section dedicated to the predictors selection the following features have been selected for this example $: A_{26}, A_{44}, D_{12}, A_{26}^2, A_{44}^2, A_{45}^2, A_{11}: A_{44}, A_{26}: A_{45}$, where ":" denotes interactions between terms.

We randomly selected 10000 observations for the training dataset and the remaining 76953 will represent the test data set. All the results presented in Tables 2 and 3 represent the performance of the trained algorithms evaluated with the test dataset. We have used R language to run the described machine learning techniques.

In the classification case we have defined 2 classes: $\lambda \leq \lambda_{crit}$ which we'll refer as *Not Fail* class and $\lambda > \lambda_{crit}$ which we'll refer as *Fail* class. As we use the maximum strain criterion the value of λ_{crit} is 1. For the algorithms where probabilities are estimated we have used the 0.5 probability to separate the 2 classes. The results are shown in Table 2.

From Table 2 it can be observed that the classification error rate is very small-under 2 % - for k nearest neighbor, random forest and support vector machines while for the rest of algorithms the error rate is still acceptable. Motivated by the very good results we run the top 3 algorithms with a training dataset reduced to only 2000 observations. As a consequence the classification error rate increase to 7.13% for k nearest neighbor, to 5.39% for random forest and to 6.16% for support vector machines, remaining still acceptable.

Table 3 Quantitative estimation results on the test dataset

Algorithm	MSE [-]	Mean of errors [-]	Standard deviation of errors [-]	Algorithm parameters
Support vector machines	0.0034	0.0032	0.0588	γ=0.18, cost=300
Random forest	0.0057	0.0081	0.075	m=3, B=500
Artificial neural networks	0.0058	0.0008	0.076	2 hidden layers with 5 and 3 neurons
Boosting	0.0079	0.0022	0.089	α=1.0, d=3, B=500
K nearest neighbor	0.008	0.0041	0.0895	K=1
Linear regression	0.0145	0.0041	0.1206	-
The lasso	0.0145	0.0044	0.1205	α=5.41e-05
Principal components regression	0.0145	0.0041	0.1206	M=8
Partial least squares	0.0145	0.0041	0.1206	M=8
Ridge regression	0.0593	0.0377	0.24	α=0.03733

The computational effort necessary to generate and evaluate a training dataset using the FEA model from Serban (2016) is around 3 minutes for a population of 2000 observations and around 15 minutes for a population of 10000 observations using an i3 processor while the time spent to train the machine learning algorithm and the time spent for genetic optimization became insignificant.

In Table 3 are shown the results for the quantitative estimation of the failure factor. As performance measures we use the mean square error (MSE), mean of the errors and standard deviation of the errors.

From Table 3 it can be observed that the best performance can be obtained with support vector machines while random forest and artificial neural networks provides very good estimates. The mean value of estimation errors is very close to 0 and the smallest standard deviation of estimation errors is 0.058. This means that aproximatively 95.5% from the estimation errors are in the range of [−0.116 0.116] having a normal distribution with mean 0. Motivated by the very good results we run the support vector machines with a training dataset reduced to only 2000 observations. As a consequence the test MSE increase to 0.0079 and the standard deviation of the errors to 0.089, ramaining still acceptable.

Even if the results obtained with linear regression are inferior we take the opportunity to extend our analysis based on the interpretability of the linear regression model. All the p-values associeted with the regression coefficients are virtually 0 (<2e-16) which means that the probability to observe a true linear relationship between the response and the predictors by chance is 0. Also, it means that all the predictors are strongly related to the response. This is also confirmed by the value of the adjusted R squared which is very high - 0.946-which means that the selected predictors explain a great proportion of the response variability, confirming our assumption about (A_{ij}, D_{ij}, B_{ij}) coefficients.

Another interesting but not surprising fact is that the value of the shrinkage parameter α is almost 0 for the lasso regression. The value of α is obtained by cross validation and it means that all the selected predictors are strongly related to the response. Also, the lasso model becomes in this case just a linear regression model because the penalty factor from (15) vanishes. This is confirmed by the almost identical values obtained with linear regression and the lasso. This observation remains true also for principal component regression and partial least squares by replacing the value of α with the value of m which is maximal meaning that all the components are important to estimate the response.

7. Conclusions

In this paper we presented a procedure for qualitative and quantitative prediction of a composite laminate failure. We explored the performance of some machine learning techniques which were trained using a random population of configurations evaluated with a FEA model. We proved with a numerical example that our procedure is accurate. The misclassification rate obtained with three algorithms is under 2% and the smallest is 1.33%. Also, the failure factor estimation is very good, 95% of the deviations being smaller than 0.116 in absolute value-in case of support vector machines. We proposed a simple procedure for feature selection given that some predictors can be correlated and we statistically proved that the relationship between the response and the predictors is very strong and that a lot of response variability is explained by the selected predictors. Our procedure is very useful for the layup optimization problem where the FEA model can be replaced by a trained machine learning model which drastically reduces the optimization time to a negligible value. The time reduction is even more important when the optimization procedure should run a lot of times like in the case of obtaining the pareto-optimal set of designs. Also, the time needed to generate and evaluate the population of configurations used to train the machine learning model is not wasted. The reason is that from this population it can be selected the best individuals to create the initial population for the optimization algorithm. As further work we intend to explore with more machine learning techniques to increase the estimation accuracy. Also, we take into account to develop o controlled procedure to generate the training population in order to better cover the region in the search space we are most interested with the expectation to increase the prediction accuracy for that specific region.

References

Badallo, P., Trias, D., Marin, L. and Mayugo, J.A. (2013), "A comparative study of genetic algorithms for the multi-objective optimization of composite stringers under compression load", *Compos. Part B*, **47**, 130-136.

Bagheri, M., Jafari, A.A. and Sadeghifar, M. (2011), "Multi-objective optimization of ring stiffened cylindrical shells using a genetic algorithm", *J. Sound Vib.*, **330**(3), 374-384.

Bishop, C.M. (2006), *Pattern Recognition and Machine Learning*, Springer.

Chatzi, E.N., Hiriyur, B., Waisman, H. and Smyth, A.W. (2011), "Experimental application and enhancement of the XFEM-GA algorithm for the detection of flows in structures", *Comput. Struct.*, **89**(7-8), 556-570.

Cho, J.R., Lee, J.H., Kim, K.W. and Lee S.B. (2013), "Generalized evolutionary optimum design of fiber-reinforced tire belt structure", *Steel Compos. Struct.*, **15**(4), 451-466.

Falzon, B.G. and Faggiani, A. (2012), "The use of a genetic algorithm to improve the postbuckling strength of stiffened composite panel susceptible to secondary instabilities", *Compos. Struct.*, **94**(3), 883-895.

Ghiasi, H., Fayazbakhsh, K., Pasini, D. and Lessard, L. (2010), "Optimum stacking sequence design of composite materials Part II: Variable stiffness design", *Compos. Struct.*, **93**(1), 1-13.

Ghiasi, H., Pasini, D. and Lessard, L. (2009), "Optimum stacking sequence design of composite materials Part I: Constant stiffness design", *Compos. Struct*., **90**(1), 1-11.

Hastie, T., Tibshirani, R. and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd Edition, Springer series in statistics.

Kayikci, R. and Sonmez, F.O. (2012), "Design of composite laminates for optimum frequency response", *J. Sound Vib.*, **331**(8), 1759-1776.

Kocak, Y, Gulbandilar, E. and Akcay, M. (2015), "Predicting the compressive strength of cement mortars containing FA and SF by MLPNN", *Comput. Concrete*, **15**(5), 759-770.

Lee, D.S., Morillo, C., Bugeda, G., Oller, S. and Onate, E. (2012), "Multilayered composite structure design optimization using distributed/parallel multi-objective evolutionary algorithms", *Compos. Struct.*, **94**(3), 1087-1096.

Mohri, M., Rostamizadeh, A. and Talwalkar, A. (2012), *Foundations of Machine Learning*, MIT Press.

Montagnier, O. and Hochard, Ch. (2013), "Optimization of hybrid high-modulus/high-strength carbon fiber reinforced plastic composite drive shafts", *Mater. Des.*, **46**, 88-100.

Murphy, K.P. (2012), *Machine Learning: a Probabilistic Perspective*, 1st Edition, MIT Press.

Murugan, S., Flores, E.I.S., Adhikari, S. and Friswell, M.I. (2012), "Optimal design of variable fiber spacing composites for morphing aircraft skin", *Compos. Struct.*, **94**(5), 1626-1633.

Ozcan, G., Kocak, Y. and Gulbandilar, E. (2017), "Estimation of compressive strength of BFS and WTRP blended cement mortars with machine learning models", *Comput. Concrete*, **19**(3), 275-282.

Papadopoulos, L. and Kassapoglou, C. (2007), "Shear buckling of rectangular composites plates composed of concentric layups", *Compos. Part A.*, **38**(5), 1425-1430.

Park, C.H., Lee, W.I., Han, W.S. and Vautrin, A. (2008), "Improved genetic algorithm for multidisciplinary optimization of composites laminates", *Comput. Struct.*, **86**(19-20), 1894-1903.

Pohlak, M., Majak, J., Karjust, K. and Kuttner, R. (2010), "Multi-criteria optimization of large composite parts", *Compos. Struct.*, **92**(9), 2146-2152.

Samui, P. (2011a), "Least square support vector machine and relevance vector machine for evaluating seismic liquefaction potential using SPT", *Nat. Hazard.*, **59**(2), 811-822.

Samui, P. (2011b), "Prediction of pile bearing capacity using support vector machine", *Int. J. Geotech. Eng.*, **5**(1), 95-102.

Serban, A. (2016), "Fast and robust Matlab-based finite element model used in the layup optimization of composite laminates", *Proceedings of the 7th International Conference on Advanced Concepts in Mechanical Engineering*, Iasi, Romania, June.

Sharma, D.S., Patel, N.P. and Trivedi, R.R. (2014), "Optimum design of laminates containing an elliptical hole", *Int. J. Mech. Sci.*, **85**, 76-87.

Sliseris, J. and Rocens, K. (2013), "Optimal design of composite plates with discrete variable thickness", *Compos. Struct.*, **98**, 15-23.

Topal, U. (2013), "Pareto optimum design of laminated composite truncated circular conical shells", *Steel Compos. Struct.*, **14**(4), 397-408.

Yong, M., Falzon, B.G. and Iannucci, L. (2008), "On the application of genetic algorithms for optimizing composites against impact loading", *Int. J. Impact Eng.*, **35**, 1293-1302.

Yong, M., Iannucci, L. and Falzon, B.G. (2010), "Efficient modelling and optimization of hybrid multilayered plates subject to ballistic impact", *Int. J. Impact Eng.*, **37**(6), 605-624.

CC