# Report UGAL – project D3T4H2S

prof. Viorel MÎNZU, prof. Eugen RUSU, postdoctoral researcher Ana CHIROȘCĂ

February 1, 2025 – June 14, 2025

# Implementation of Machine Learning Models for Composite Materials — with the Same Carbon Fiber but Different Matrices —

## Table of Contents

## 1. Introduction

This project report covers only a specific phase of the project "A Data-Driven Digital Twin for Improved Hydrogen Storage Vessels Addressing Challenges for the Energy Transition", Europe Horizon – LEAP-RE program. This report aims to demonstrate how machine learning (ML) algorithms can model composite materials, particularly structures composed of fibers and matrices. The developed ML models establish the relationship between stress and strain values while incorporating additional physical parameters of composite materials. This report examines three composite materials that utilize *the same carbon fiber but different matrices. ML models address two or three composites simultaneously to reduce the data points needed for the training procedure while ensuring good prediction accuracy.* In this context, a conjecture is proposed and gradually solved using the developed ML models. The main result is decreasing the time and cost of simulations that generate datasets. Readers of the report can access all the scripts, functions, and auxiliary files used by the authors in this work.

This project report outlines a modeling approach to predict the mechanical behaviour of specific composite materials. The scientific opportunities of this topic and the use of composite materials for fabricating hydrogen storage vessels are largely described in many papers (for example, [1]).

The project consists of multiple phases, including the modeling of composite materials in two contexts:

- Modeling composite laminates with various fibers, already presented in [2].
- Modeling composite materials, such as fiber and matrix structures, is the topic of this presentation.

Moreover, the above-mentioned project deals with the modelling of composite materials within two scientific contexts:

- The composite materials science involves mechanical, physical, and chemical knowledge working together to achieve models. Finite element numerical integration is used to construct these models.
- Machine learning algorithms are used to create models of composite materials (see Liang [3]) based on data generated through physical tests or precise simulation campaigns. These models enable stress-strain analysis and predict mechanical behavior. This adjacent context emerges from the first one, the composite materials science, which offers the foundation of rigorous results analysis.

The objective of this presentation is solely to demonstrate how machine learning algorithms were employed for modeling composite materials, specifically structures made up of fibers and matrices. *This report does not cover the analysis based on composite materials science elaborated by the partners in this project, nor the implications derived from this analysis.*

This report focuses on three composite materials: CF/PA6 (Carbon Fiber/Polyamide 6), CF/PA12, and CF/PA11. The results can also apply to other materials. The developed ML models describe the relationship between stress and strain while incorporating additional physical parameters to identify composite materials.

Machine learning (ML) models of the stress-strain relationship can be created using data collected from simulations or mechanical tests. Because the last ones are very expensive, simulations assisted by the DIGIMAT–VA software generate the datasets using finite element calculations. Generally, the simulation of material performance under real-world conditions enables a precision of up to 95% correlation with physical testing results (Dotoli et al. [4]). Simulations provide high-quality data but take much more time than predictions.

For any composite material of this kind, we are interested in describing the prediction of the stress value as a function

**stress = *f* (strain, other physical parameters)**.

We will create ML regression models [5–9] that will enable us to both implicitly and explicitly determine the function *f*. When called repeatedly, the function *f* can generate the complete profile of the composite material's mechanical behavior. These models will be suitable tools for materials science specialists to analyze the mechanical behavior of a composite material with the same carbon fiber (CF) but a different polyamide.s

The main objective is to address a specific problem: to provide an answer to a proposed conjecture. We assume sufficient datasets are available to develop the ML prediction models for CF/PA6 and CF/PA12. Moreover, a joint ML model for CF/PA6 and CF/PA12, referred to as ML/PA6–PA12, can be developed.
Since all three materials utilize the same carbon fiber, we will investigate whether the mechanical behavior predictions of CF/PA11 can be derived from those of ML/PA6–PA12. We are interested in this hypothesis to avoid conducting complete calculations with DIGIMAT-VA each time predictions for PA11 are required. This approach will reduce the analysis or optimization time.
If the answer is negative, we will relax the problem: Can several data points be added to the ML/PA6–PA12 training data set to enable accurate predictions for CF/PA11?

The primary contribution of this work is the gradual resolution of the proposed conjecture. Our research was directed to find a small number of data points added to the ML/PA6–PA12 training data set. The solution is not unique; it depends on the desired level of prediction accuracy.

As we shall present, aggregating predictions can define a relationship between stress and strain. The general implementation of function *f* in the current programming language for any composite material represents a collateral contribution that springs from the creation of the ML models. This will be necessary in the upcoming phases of the D3T4H2S project.

This report could be of interest to project partners involved in developing ML models for composite materials, particularly regarding their mechanical behavior. As we shall see, the Regression Neural Network (RNN) and Gaussian Process Regression (GPR) fulfill the accuracy requirements for predicting stress values based on strain and other parameters.

In a broader context, project partners interested in ML predictions can find implementation details about data organization and the application of ML algorithms in practical, real-world settings. Mutatis mutandis, they could apply aspects of the data organization and implementation to other specific models and predictions.

Integrating Artificial Intelligence (AI) into composite material science has significantly enhanced predictive modeling and optimization. *Here, we mention only a few papers relevant to our work, considering the specifics of a project report*.

Neural networks can enhance the prediction of dynamic mechanical properties, achieving up to 30% greater accuracy compared to older methods (Vahed et al. [10]). This accuracy significantly reduces the experimental workload by efficiently optimizing material properties.

Young's modulus of polymer composites reinforced with carbon nanotubes was predicted using ML models (Ho et al. [11]). The accuracy exceeds traditional testing methods by approximately 25%, facilitating the advancement of new materials.

Convolutional neural networks combined with principal component analysis can accurately predict stress-strain behaviors from microstructural images of composites (Yang et al. [12]). This method has decreased computational time by 40% compared to traditional finite element analysis, offering a fast and scalable tool for material design. Campbell [13] has emphasized the role of AI in predicting the mechanical behavior of composites under varied stress conditions.

Models of composite materials developed using ML algorithms [14–15] enable accurate simulations and predictions of their performance. This significantly reduces both time and costs compared to simulations and experimental testing. In [2], ML models for composite laminates with various fiber orientations allow a comparative analysis of these materials.

Following this introduction, Section 2 discusses the materials and methods utilized in this report. First, it explains how ML will be utilized to create specific models through responses to several questions: Who generates the datasets for the ML algorithms? What ML models will be developed? What are the practical objectives? Section 2.2 (Data preparation) addresses this crucial topic extensively and may be of interest for organizing the application. ML models for single composite materials, such as CF/PA6 and CF/PA12, are developed and validated in subsection 2.3; they will serve as the basic building blocks for implementing subsequent models.

Section 3 aims to develop machine learning models that accurately represent three composite materials and predict stress based on strain and other parameters of the matrix. These models should be trained using data points for CF/PA6 and CF/PA12, along with the minimum necessary data points for CF/PA11. Their implementation is presented so readers can reproduce and adapt our work to their applications. The results of our approach are presented naturally as outcomes of the models' implementation and testing scripts; other details complement their presentation.

Section 4 briefly recalls the gradual resolution of our conjecture and analyzes the models and results obtained in each step of our work.

All MATLAB scripts, functions, and other files used in this work are available to readers in the "`Autom_Mtlb`" folder, which has been archived and attached to this report.

## 2. Machine Learning Regression Models and Data Preparation

In the following subsections, besides the presentation of the materials, the methods will be described more directly and in correlation with their implementations.

### *2.1 Machine Learning Models for the Mechanical Behavior of Composite Materials*

The project, referred to in the sequel as D3T4H2S, addresses various topics, including the mechanical behavior of composite materials used in constructing liquid hydrogen storage tanks. The primary goal is to analyze the stress-strain relationships of these materials, which consist of the same carbon fiber (CF) combined with different matrices: polyamide PA6, PA12, and PA11. It is crucial to clarify several aspects from the outset:

(1) Who generates the data sets for the ML algorithms?

Before constructing the machine learning models, we used the specialized finite element simulator DIGIMAT-VA (version 2023.3) to analyze the stress-strain relationships of three materials: CF/PA6, CF/PA12, and CF/PA11. Physical parameters and assessed rupture criteria offered insights into the materials' response to mechanical loading. The simulation software calculates the stress values for many strain values spanning undamaged (elastic) and damaged (plastic) zones, excluding the rupture zone.
Datasets are collected and used by ML algorithms.

(2) What ML models is this work aiming to develop?

Machine learning algorithms utilize imported datasets to train and validate regression models representing the relationship between stress and strain. The ML algorithms are implemented as MATLAB functions and scripts, version R2024a. Based on the strain and other parameter values, this model predicts the stress value much faster than conducting a finite element calculation. We can consider the prediction as a function in equation (1).

$$\text{stress} = f\,(\text{strain}, \ldots). \tag{1}$$

If the predictions are accurate, the function $f$ can generate the complete profile of the composite material's mechanical behavior when called repeatedly.

(3) What practical objectives does implementing machine learning models aim to achieve?

3-1. The prediction function $f$ is generally implicitly available when the ML model is already created, as an attribute of the considered object. The implementation's first objective is to define $f$ by its specific implementation in the programming language. This will be done using the object representing the ML model. These functions, $f6$, $f12$, and $f11$, will characterize the mechanical behavior of the three composite materials, especially when the predictions' accuracy is very good. The benefit of having these functions in this form is that they will be largely used in the next D3T4H2S stages.

3-2. The second objective is to address a specific problem. We assume sufficient datasets exist to develop the ML prediction models for CF/PA6 and CF/PA12.

Given that the three materials share the same carbon fiber, we will investigate whether the predictions for the mechanical behavior of CF/PA11 can be derived from a joint ML model for both CF/PA6 and CF/PA12, referred to as ML/PA6–PA12. This time, the data points characterizing the three composite materials will include variables reflecting the material properties of the matrices. The goal is to avoid complete calculations with DIGIMAT-VA each time we analyze the mechanical behavior of a composite material with the same CF but a different polyamide.

We will add a certain number (*n*11) of datapoints characterizing the material CF/PA11 to the datapoints for CF/PA6 and PA12 to generate, after training and testing, a new ML model, referred to as ML/PA6–PA12–pa11. Using lowercase "pa11" instead of "PA11" symbolizes that its training dataset includes only a few additional data points related to CF/PA11. We will investigate how small *n*11 could be so that the predictions for CF/PA11 would achieve good accuracy.

### 2.2 Data preparation.

The data used to develop the ML models for the stress-strain relationships of the three composite materials—CF/PA6, CF/PA12, and CF/PA11—are contained in three files: "PA6.xlsx," "PA12.xlsx," and "PA11.xlsx." These results were obtained from simulations conducted using the software DIGIMAT–VA. The material parameters for carbon fiber and matrices are provided in the "Demanded Data.xlsx" file.

These files contain data that describe stress-strain relationships for six load directions; they are processed similarly to prepare for the learning process. In this presentation, we will only consider the processes corresponding to PA6, as the other data undergo similar transformations. At the same time, all the columns of the "PA6.xlsx" file are loaded into table T6, but only the first two columns are retained in Table T6m, which is presented in Table 1. These columns contain the strain and stress values when the composite material's load direction aligns with the direction of the fiber (situation coded by the sequence 11). In the sequel, only this load direction is considered; the other directions are treated similarly, but different ML models are produced.

**Table 1**. The first ten lines of table T6m out of its 203 lines

| Strain11 | Stress11<br>(MPa) |
|---:|---:|
| 5e-14 | 6.0014e-09 |
| 0.0005 | 60.014 |
| 0.000853 | 102.38 |
| 0.001353 | 129.17 |
| 0.001853 | 153.49 |
| 0.002353 | 175.29 |
| 0.002853 | 194.54 |
| 0.003353 | 211.3 |
| 0.003853 | 225.68 |
| 0.004353 | 237.84 |

Table 1 is sufficient for constructing the ML model for the stress-strain relationship corresponding to this load direction. However, the material parameters must be added to the data points to obtain the joint ML model in the sequel. In addition to the load direction ("LoadD"), eight material parameters are listed in the adjacent columns:

- Density (tonne/mm3) [20-25°C],
- Matrix Young Modulus Em (MPa),
- Poisson ratio, Yield stress (MPa),
- Hardening modulus (MPa),
- Hardening exponent,
- Linear hardening modulus (MPa),
- Volume fraction $Vf$.

Table 2 presents the structure of Table T6mB, which consists of 203 lines and 11 columns, along with the values of the material parameters required by the joint ML model.

**Table 2**. The first ten lines of table T6mB out of its 203 lines.

| LoadD | Density | Em | Poisson | YieldS | Hm | He | Lhm | Vf | Strain11 | Stress11 |
|---|---|---|---|---|---|---|---|---|---|---|
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 5e-14 | 6.0014e-09 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.0005 | 60.014 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.000853 | 102.38 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.001353 | 129.17 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.001853 | 153.49 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.002353 | 175.29 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.002853 | 194.54 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.003353 | 211.3 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.003853 | 225.68 |
| LD11 | 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.6 | 0.004353 | 237.84 |

Readers of the report can examine the script "S1_Table6_12_11B.m" located in the "Autom_Mtlb" folder to understand the data processing of the three composite materials. Lastly, the relevant tables are saved as workspace files (".mat").

The script "S2_TablesTrainTest.m" plots the stress-strain relationship in Figure 1 as it results from data supplied by the DIGIMAT-VA simulations. It utilizes the tables constructed in the previous script. The elastic and plastic zones are highlighted.

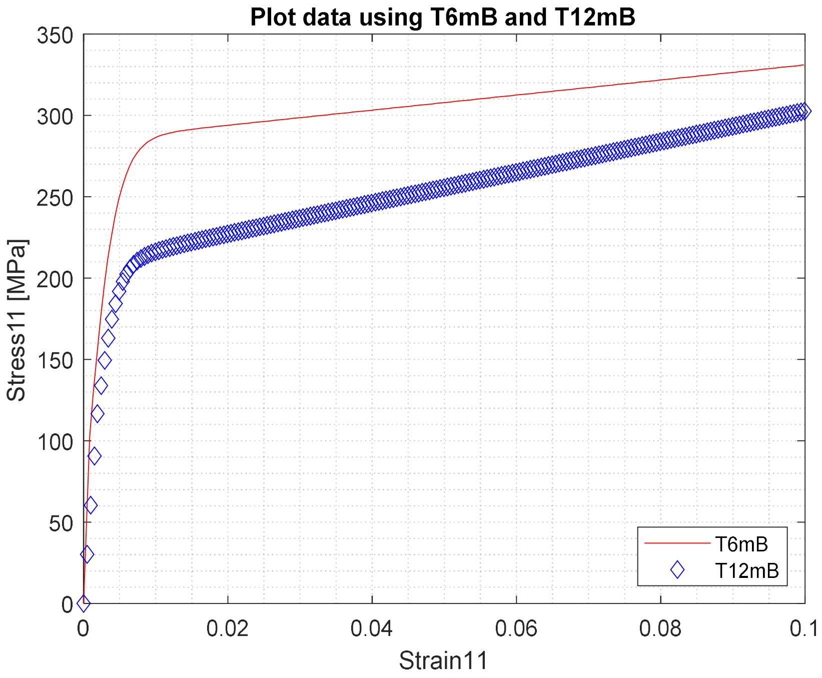


**Figure 1.** Undamaged and damaged zones for PA6 and PA12

The primary goal of this script is to create the tables that will be utilized during the training and testing processes. T6m1 is a new table derived from T6m, to which a "categorical" type column has been added, containing the value "PA6."

Two tables, DTrain and DTest, are created using lines from T6m1. The lines from the latter are added alternately to DTrain and DTest, four and one, respectively. Ultimately, tables DTrain and DTest contain 161 and 40 lines, respectively. Table 3 presents the first 5 and last 6 lines of the table "DTrain." The table "DTest" has the same variables but consists of only 40 lines.

**Table 3.** The first 5 and last 6 lines of table DTrain out of its 161 lines

| Matrix | Strain11 | Stress11 |
|---|---|---|
| PA6 | 5e-14 | 6.0014e-09 |
| PA6 | 0.0005 | 60.014 |
| PA6 | 0.000853 | 102.38 |
| PA6 | 0.001353 | 129.17 |
| PA6 | 0.002353 | 175.29 |
| ............... | ............... | ............... |
| PA6 | 0.096353 | 329.33 |
| PA6 | 0.097353 | 329.79 |
| PA6 | 0.097853 | 330.03 |
| PA6 | 0.098353 | 330.26 |
| PA6 | 0.098853 | 330.49 |
| PA6 | 0.099853 | 330.95 |

Finally, the tables "DTrain" and "DTest" are saved in a workspace file. For PA12, similar tables are created and stored in a comparable workspace.

### *2.3 ML Model for a Single Composite Material.*

The initial ML model developed by the authors is a parametric one, specifically a multiple linear regression model. This model can incorporate nonlinear terms, such as interactions, which are the products of predictor variables. Despite these possibilities, the model preserves a linear relationship concerning its coefficients.

Among the linear regression models developed in our work, we present only the one based on a stepwise regression method, which involves adding or removing features from a constant model. In our MATLAB implementation, this method is executed using a specialized function called "stepwiselm(T)," which provides a model that fits the dataset in T [38].

Details regarding the construction of such a model can be found in the script "S3_PA6Model_SW.m" and its execution list (Appendix A, Table A1). The resulting model, referred to as "mdlsw", is used to predict stress values corresponding to the strain values belonging to "D6Test". These predicted results are then compared to the values in the "D6Test.Stress11" column. The matrix "MatC" and the table "TableC" are created for this comparison (see Appendix A, Table A2).

The predictions are less accurate at the extremes of the strain's range, and we will try to construct regression models using ML algorithms that are better suited to our problem (Support Vector Machines, decision trees, Gaussian process regression, and neural networks). This report presents only Regression Neural Networks and Gaussian Process Regression as the ML algorithms that produced the best results.

We utilized the Regression Learner application in the MATLAB environment to analyze and identify the most accurate models for the stress-strain relationships of both PA6 and PA12. As an example, for PA6, the result of this investigation is the MATLAB function "trainRegNN4.m" that generates a trained ML model, which is an RNN (in our implementation referred to as "mdlNN4"). The training is achieved by leveraging the "fitrnet" function, which is called with the parameters described in Appendix B, Table A3.

For PA12, we also created an RNN training function.

The algorithm (method) for creating an RNN and evaluating its predictions on the test dataset is detailed in the following pseudocode; it uses PA6 as a generic matrix.

**Table 4**. Method for creating an ML model based on an RNN.

```
Create an RNN and compare the real and predicted stress values
1. Load the tables "T6m1", "D6Train", and "D6Test" from the appropriate workspace.
2. [mdlNN4, vrmse] ← trainRegNN4(D6Train).
3. StressPredValid ← mdlNN4.predictFcn(D6Train).
4. RMSEValid ← rmse(StressPredValid, D6Test.Stress11)
5. Display vrmse and RMSEValid.
6. StressPred ← mdlNN4.predictFcn(D6Test).
7. Create the table "TabD" with two columns: "D6Train.Stress11" and "Stress-Pred".
8. Display table "TabD".
9. Plot Figures 2 and 3 that compare the actual and predicted stress values.
10. Save the ML model for PA6 in a workspace file.
end
```

The "vrmse" value is the RMSE (root mean square error) calculated during the training process (the validation phase). The "RMSEValid" refers to the RMSE computed between the predictions and the actual values during the testing phase. The "rmse" and "predictFcn" are functions in MATLAB. The "rmse" function calculates the RMSE between two vectors, while "predictFcn" is a method of the "mdlNN4" object that performs predictions.

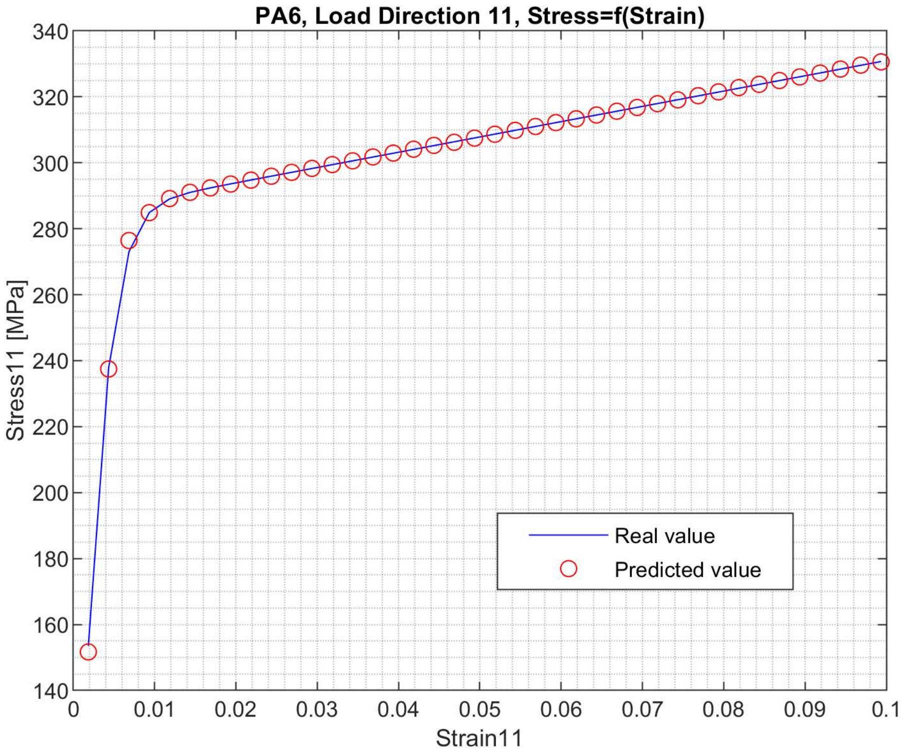


**Figure 2.** Stress-strain functions: real and predicted by the ML model "mdlNN4".

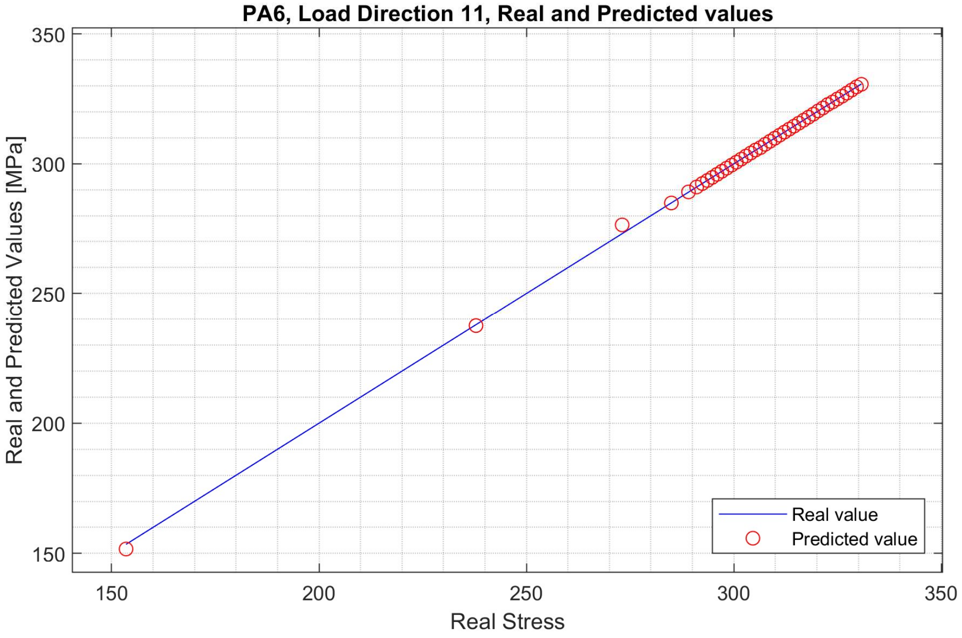


**Figure 3.** Real and predicted stress values using the ML model "mdlNN4".

The algorithm in Table 4 is implemented by the script "S5_PA6DamageRNN4.m," whose execution listing is provided in Appendix B, Table A4. This table and Figures 2 and 3 demonstrate that the ML model "mdlNN4" predictions are highly accurate, reinforcing our prediction approach.

## 3. Implementation and Results

### *3.1 Representation of the stress-strain relationship for a given composite*

The representation of the function stress = *f*(strain, …) is crucial due to its implications in the subsequent stages of the D3T4H2S project. Considering the graphic representation of the measurements or simulations, the most natural way to represent the stress–strain relationship is to encode this graphic representation so that it can be reproduced at each point. Below are the possibilities examined to achieve such a representation:

- A vector of pairs (strain, stress) from the DIGIMAT-VA simulation; the number of points should be sufficient and tailored to the specific application.
- A vector that contains pairs of values (strain, stress), including all angular points up to the rupture point.
- Another representation based on the observation that in the MATLAB system, the prediction function is available implicitly; it can be defined through the object's method that implements the composite material's ML model.

In this work, we utilized the third option among those mentioned above, taking into account the following features of the work already presented:

- The file "PA6.xlsx" includes the strain and stress values along the various load directions of the different simulations.
- The values in this file enable constructing an ML regression model, which "learns" the stress-strain relationship. The ML model can be any type among those that can be constructed.
- Regardless of the model type, it includes a prediction "function" that allows us to specify a stress value for a given strain value.

Generally, the prediction function is a method within the object that implements the ML model (in our case, "mdlNN4"). For various applications, we can represent the stress-strain relationship as a function, in the sense of the programming language at hand:

stress = *f*(strain, ML_model, other parameters).

Here is an example of the structure of the function "F_PA6" in our context:

**Table 5**. Definition of the function *f* according to the programming language in use.

```
Function stress = F_PA6(strain,mdl)
            % The function F_PA6 serves as a general representation of f.
            % strain: the strain's value
            % mdl: the ML model as an object reference.
 1.  Define the table Tabj with one line and two columns.
            % tabj=table('Size',[1,2],'VariableTypes',{'categorical','double'},...
            % 'VariableNames',{'Matrix','Strain11'});
2.  Tabj{1,2} ← strain;
3.  Tabj{1,1} ← "PA6";
4.  stress=mdl.predictFcn(tabj);
end
```

The file "F_PA6.m" implements this function; "PA6" is a categorical constant suggesting the function's name. This function could be modified to include the matrix type (PA6, PA12,...) or the load direction in its arguments.

The script "S8_TestF_PA6.m" tests the function "F_PA6.m" by verifying the predictions for PA6 using the model mdlNN4, which is loaded from the appropriate workspace. Like the script "S5_PA6DamageRNN4.m," it produces the same results and plots figures identical to Figures 2 and 3.

### *3.2 Construction of a joint ML model for CF/PA6, CF/PA12, and CF/PA11*

The previous sections have shown that ML models can be effectively developed for the composite materials CF/PA6 and CF/PA12. Since the composite material CF/PA11 uses the same fiber as the other two materials, we will explore the following approach to predict the stress for CF/PA11:

- We shall construct a single ML model characterizing both composite materials simultaneously, CF/PA6 and CF/PA12. This model will be referred to as ML/PA6–PA12.
- We assume that the composite material CF/PA11 follows the ML/PA6–PA12 model since it uses the same fiber. We will test whether this model applies to the behavior of PA11 and if the predictions align with the actual stress values. If the predictions are inaccurate, we will include a specific number of data points (*n*11) that characterize the CF/PA11 material.

The processed data should be unified and refined to create the dataset for the joint model. This is achieved in the script "S9_BigDataSet.m," which is easy to follow as it involves only sequential actions. Table 6 summarizes these processes.

**Table 6.** Data processing steps to create the dataset for the joint model**.**

```
>> S9_BigDataSet
1. Load "T6mB" and "T12mB" from the appropriate workspace file.
2. The matrix "Mat6" is the image of the table "T6mB" after eliminating the first
   2 columns.
3. Mat6s is a selection of Mat6's lines (one-third of its lines).
4. The matrix "Mat6b" is created by adding three perturbed lines for each line
   of "Mat6s." The perturbation ranges uniformly from -2% to +2% of the
   matrix values.
5. – 7. Carry out actions similar to steps 2 to 4, but concentrate on the data
   related to PA12 (the matrices bear the index 12).
8. Create the matrix "Mat11b" using the first three lines of "T11mB" and adding
   nine perturbed lines as before.
9. Create "Mat6_12_11" by concatenating the lines from "Mat6b," "Mat12b,"
   and "Mat11b."
```

10. Transform "Mat6_12_11" into a table called "Tab6_12_11."
11. Create the tables "Tab6_12_11train" and "Tab6_12_11test" using 75% of the datapoints from "Tab6_12_11" for training and 25% for testing.
12. Save the tables "Tab6_12_11", "Tab6_12_11train", and "Tab6_12_11test" to a workspace file.

The data-generating process must utilize a probability distribution to satisfy the assumption of independent and identically distributed samples. We have acknowledged that uniformly distributed noise perturbs the data points. This adjustment, considered at points 4 and 8, enhances parameter diversity and improves the generalization.

Remark 1: From the beginning, we have added a very small number of datapoints (12) concerning CF/PA11 to the datapoints (552) characterizing CF/PA6 and CF/PA12. Certainly, the new model, ML/PA6–PA12–pa11, will recognize CF/PA6 and CF/PA12 just as the model ML/PA6–PA12 does.

## *3.3 Implementation of joint ML models and the outcomes of their predictions.*

### 3.3.1 The model ML/PA6–PA12

This section presents a regression model that produced good results, ML/PA6–PA12. Generally, successful models are those based on Regression Neural Networks (RNN) and Gaussian Process Regression.

The function "trainRegMod5.m" is called by the script "S10_Training.m," which generates a trained ML model, "mdlNN5," an RNN (wide NN). The MATLAB "fitrnet" function is called with the following parameters:

```
mdlNN5 = fitrnet(...
    predictors, ...
    response, ...
    'LayerSizes', 100, ...
    'Activations', 'relu', ...
    'Lambda', 0, ...
    'IterationLimit', 1000, ...
    'Standardize', true);
```

The tables used in this script are retrieved from the workspace "WsTab61211.mat", which contains the following data:

- The global data set: "Tab6_12" (552 data points);
- Training data: "Tab6_12train" (414 data points);
- Testing data: "Tab6_12test" (138 data points).

The execution of the script "S10_Training.m" generates Figure 4 and the data from Table 7.

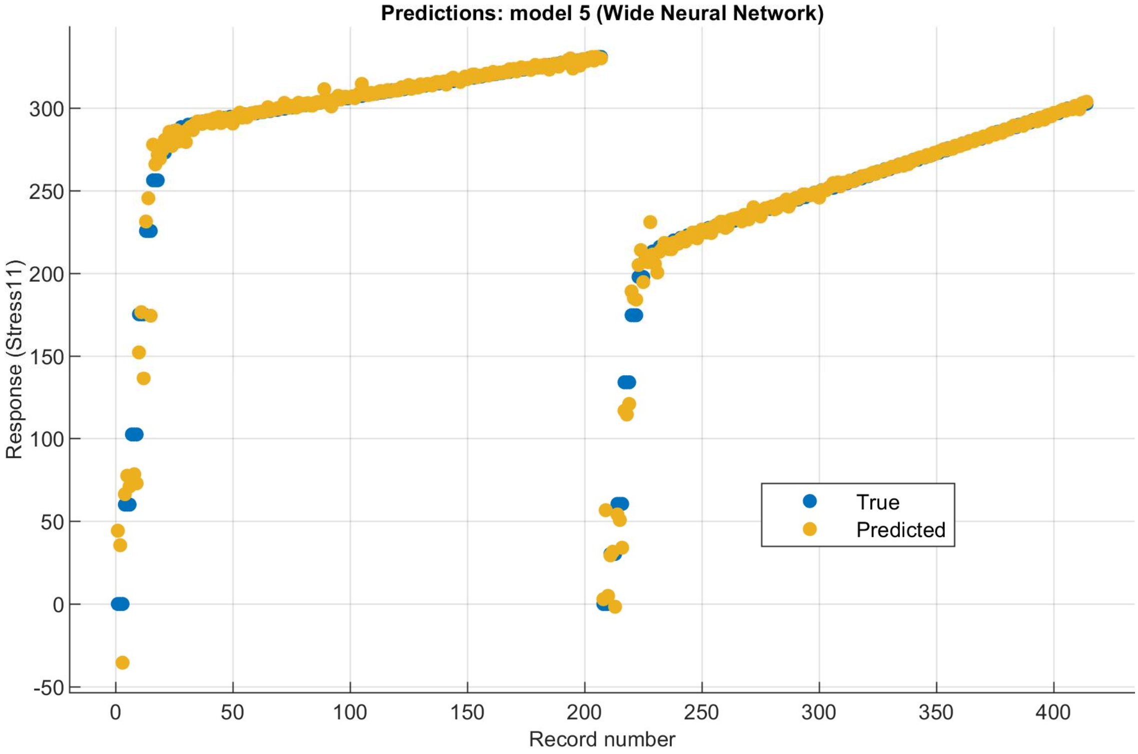


**Figure 4.** Predicted versus actual values for the same training dataset and ML model "mdlNN5".

The predicted values for the testing data are shown in Table 7, column "StressPred".

**Table 7.** * The last two columns of Table "Tab6_12test," along with the predicted values generated by "mdlNN5."

| Nr | Strain11 | Stress11 | StressPred |
|---:|---:|---:|---:|
| 1 | 5e-14 | 6.0014e-09 | 41.894 |
| 2 | 0.0005 | 60.014 | 53.935 |
| 3 | 0.00085301 | 102.38 | 60.448 |
| 4 | 0.002353 | 175.29 | 168.04 |
| 5 | 0.003853 | 225.68 | 190.92 |
| 6 | 0.005353 | 256.33 | 271.62 |
| 7 | 0.006853 | 273.07 | 275.44 |
| 8 | 0.008353 | 281.68 | 283.34 |
| 9 | 0.009853 | 286.1 | 284.16 |
| 10 | 0.011353 | 288.5 | 287.12 |
| : | : | : | : |
| 128 | 0.08493 | 288.49 | 287.24 |
| 129 | 0.08643 | 289.91 | 289.4 |
| 130 | 0.08793 | 291.32 | 290.66 |
| 131 | 0.08943 | 292.73 | 291.98 |
| 132 | 0.09093 | 294.14 | 295.29 |
| 133 | 0.09243 | 295.56 | 296.73 |
| 134 | 0.09393 | 296.97 | 297.55 |
| 135 | 0.09543 | 298.38 | 299.57 |
| 136 | 0.09693 | 299.79 | 299.04 |
| 137 | 0.09843 | 301.21 | 300.23 |
| 138 | 0.09993 | 302.62 | 303.32 |

* A fragment of the execution listing for the script "S10_Training.m".

The stress values for CF/PA6 and CF/PA12 are accurately predicted for all data points, as demonstrated by Figure 4, Table 7, and the statistics.

vrmse = 5.9084,

RMSEValid = 6.7645.

We used this model for 50 data points characterizing CF/PA11. The predictions were inaccurate. Moreover, by repeating the model's call for predictions, the model returns other values, which means it has no constant tendency to treat these data points.

Remark 1. Although the carbon fiber is the same, the matrix has different parameters. The model ML/PA6–PA12 can only make predictions based on the data it was trained on, which is a logical fact. In the next subsections, we propose a method to utilize the fact that the three materials share the same carbon fiber.

### 3.3.2 A model ML/PA6–PA12–pa11 whose predictions indicate a trand.

This subsection aims to extend the applicability of ML/PA6–PA12 by incorporating additional data points regarding CF/PA11 at the lower end of the strain range. These data points near the null strain can impose fixed points that the function *f* will pass through. In other words, they could eliminate an indefinite tendency of the predictions. We expect the strain–stress function to pass through several points in the lower end of the strain range.

We shall see if this method could create a model ML/PA6–PA12–pa11 that would make accurate predictions for CF/PA11.

After examining other ML models in this new context, a better model is constructed using GPR (Gaussian Process Regression), which generates model 7.

The tables used in this script are recovered from the workspace "WsTab61211.mat", which contains the following data:

- The global data set: "Tab6_12_11" (564 data points);
- Training data: "Tab6_12_11train" (423 data points) (in Appendix C);
- Testing data: "Tab6_12_11test" (141 data points).

Using the function "Mod7trainGPR.m" and the training data, the script "S11_trainMod7.m" trains the model "mdlGPR7". Executing this script produces Figure 5 and the data from Table 8.

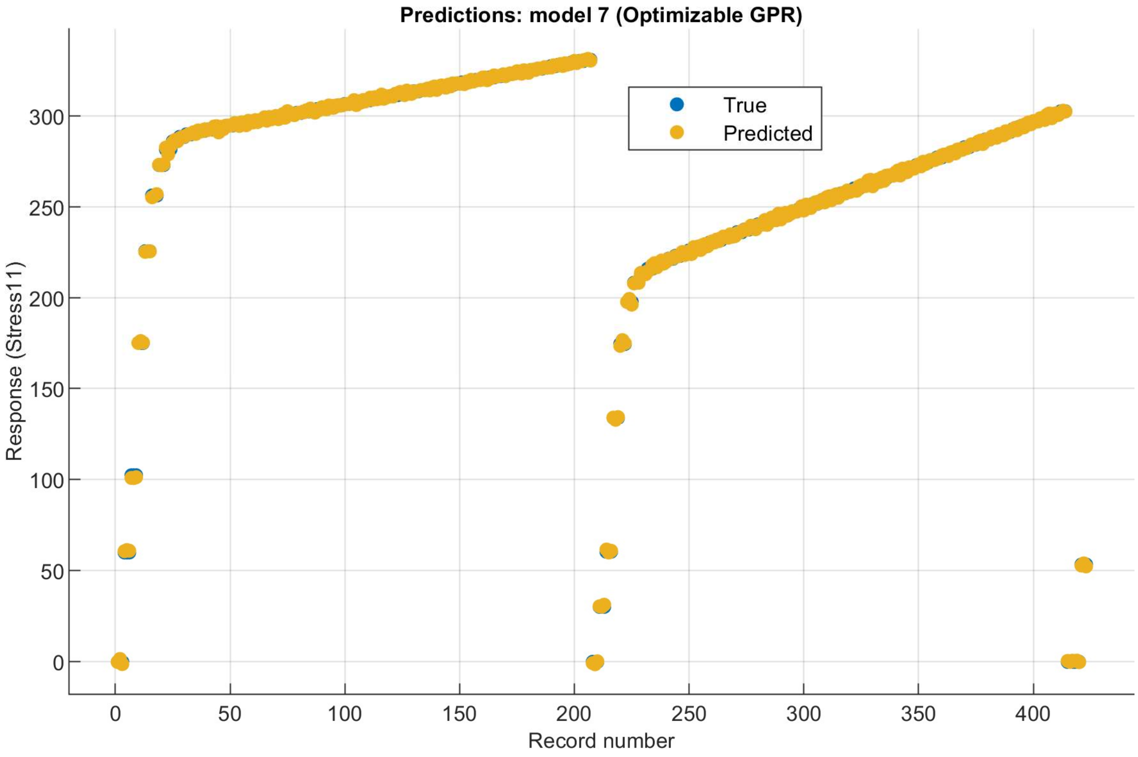


**Figure 5.** Predicted versus real values for the same training dataset and ML model "mdlGPR7".

The training and the validation give very good values for the root mean square error:
vrmse = 0.74923; RMSEValid = 0.38761

**Table 8.** * The last two columns of Table "Tab6_12_11test" and the predicted values column generated by "mdlGPR7".

| Nr | Strain11 | Stress11 | StressPred |
|---|---|---|---|
| 1 | 0 | 0 | -0.72849 |
| 2 | 0.0005 | 60.014 | 61.518 |
| 3 | 0.000853 | 102.38 | 101.05 |
| 4 | 0.002353 | 175.29 | 175.53 |
| 5 | 0.003853 | 225.68 | 225.67 |
| 6 | 0.005353 | 256.33 | 254.81 |
| 7 | 0.006853 | 273.07 | 273.44 |
| 8 | 0.008353 | 281.68 | 280.06 |
| : | : | : | : |
| 69 | 0.099853 | 330.95 | 331.33 |
| 70 | 0 | 0 | 0.4573 |
| 71 | 0.0005 | 30.21 | 29.859 |
| 72 | 0.001 | 60.42 | 61.265 |
| 73 | 0.0024301 | 133.88 | 133.33 |
| 74 | 0.0039301 | 174.71 | 173.34 |
| : | : | : | : |
| 134 | 0.09393 | 296.97 | 297.51 |
| 135 | 0.09543 | 298.38 | 298.95 |
| 136 | 0.09693 | 299.79 | 299.86 |
| 137 | 0.09843 | 301.21 | 301.11 |
| 138 | 0.09993 | 302.62 | 302 |
| 139 | 0 | 0 | 0.39246 |
| 140 | 5e-14 | 5.3419e-09 | -0.16096 |
| 141 | 0.0005 | 53.419 | 52.746 |

* A portion of the execution listing from the script "S11_trainMod7.m".

Finally, the current script selects a set of 42 data points related to PA11, distributed across the entire strain range. The model "mdlGPR7" lacks accuracy in predicting stress; however, it does show some tendencies in its predictions, as shown in Figure 6.

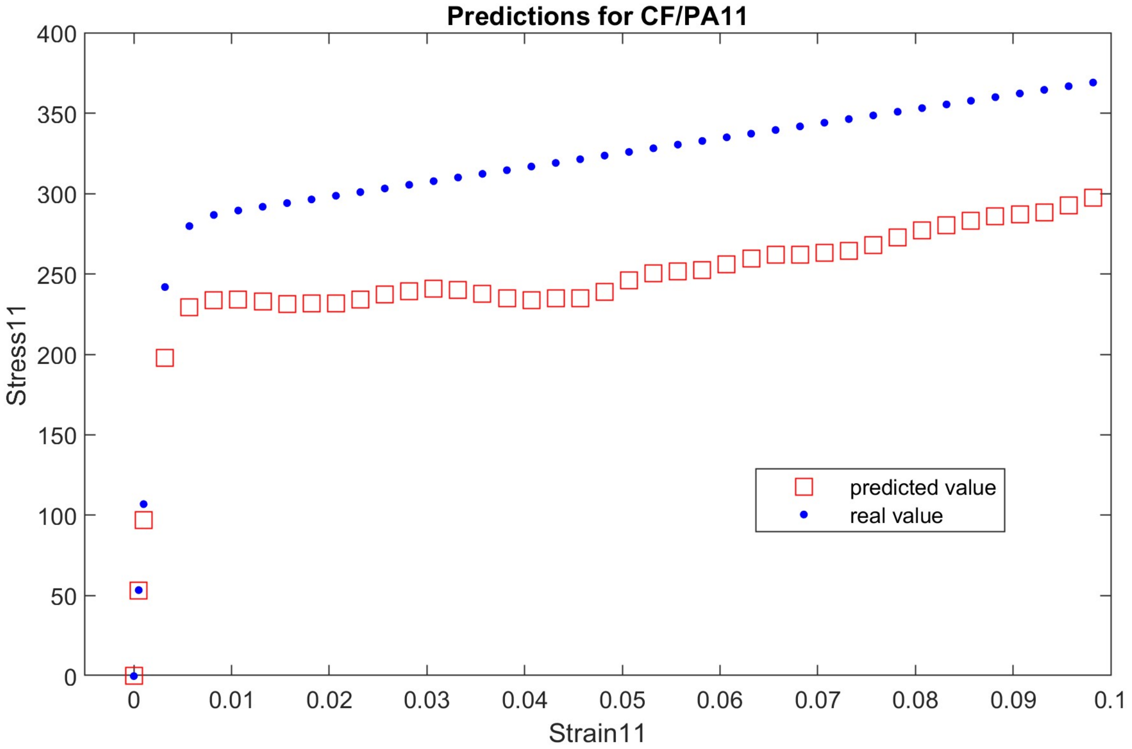


**Figure 6.** The model "mdlGPR7" predicts inaccurately CF/PA11, but it does show a tendency.

Remark 2: As we see in Figure 5 and Table 8, the model ML/PA6–PA12–pa11 implemented by "mdlGPR7" provides improved predictions for CF/PA6, CF/PA12, and the few data points related to PA11 (lines 139 – 141). But, predictions for datapoints related to PA11 and not in the training set are inaccurate. As illustrated in Figure 6, these predictions indicate a trend marking a step forward.

Remark 3: Only a few additional data points regarding CF/PA11 at the lower end of the strain range are not enough to make the model ML/PA6–PA12–pa1 able to predict accurately the stress values for PA11.

### 3.3.3 A model ML/PA6–PA12–pa11 able to predict the stress values for PA11.

Remarks 2 and 3 suggest we need to include additional data points related to PA11, covering a broader range of strain values. This will help the model "learn" more effectively from well-distributed points in various regions. The model presented in this subsection is constructed using GPR (Gaussian Process Regression), resulting in model 28. The data for this model is generated in a few steps:

- The script "S17_works.m" generates a new dataset for CF/PA11, using the same method as "S9_BigDataSet.m," which was applied for CF/PA6 and CF/PA12 to perturb the data points. This new dataset is called "Tab11Pert" and contains 276 data points.
- The script "S182_AddTab11" selects 56 data points from "Tab11Pert," distributed across the entire strain range, to create the table "Tab11crumb." This table is then added to the "Tab6_12train" table, resulting in "Tab6_12_11train2."
- To create a test set of data points the new model has not encountered during training, the table "Tab11unkn" selects 82 lines from "Tab11Pert." The data points are chosen from three specific zones of "Tab11Pert": the middle, lower, and upper ends of the strain range.

Using the function "trainMdlGPR28.m" and the new training dataset, the script "S193_addCrumb.m" trains the model "mdlGPR28". The tables utilized in this script are retrieved from the workspace "WsWithCrumb.mat", which contains the following data:

- Training data: "Tab6_12_11train2" (506 data points);

- Data characterizing CF/PA11 included in the training data: "Tab11crumb2" (56 data points);
- Testing data only for CF/PA11: "Tab11unkn" (82 data points) and "Tab11unkn2" (46 data points).

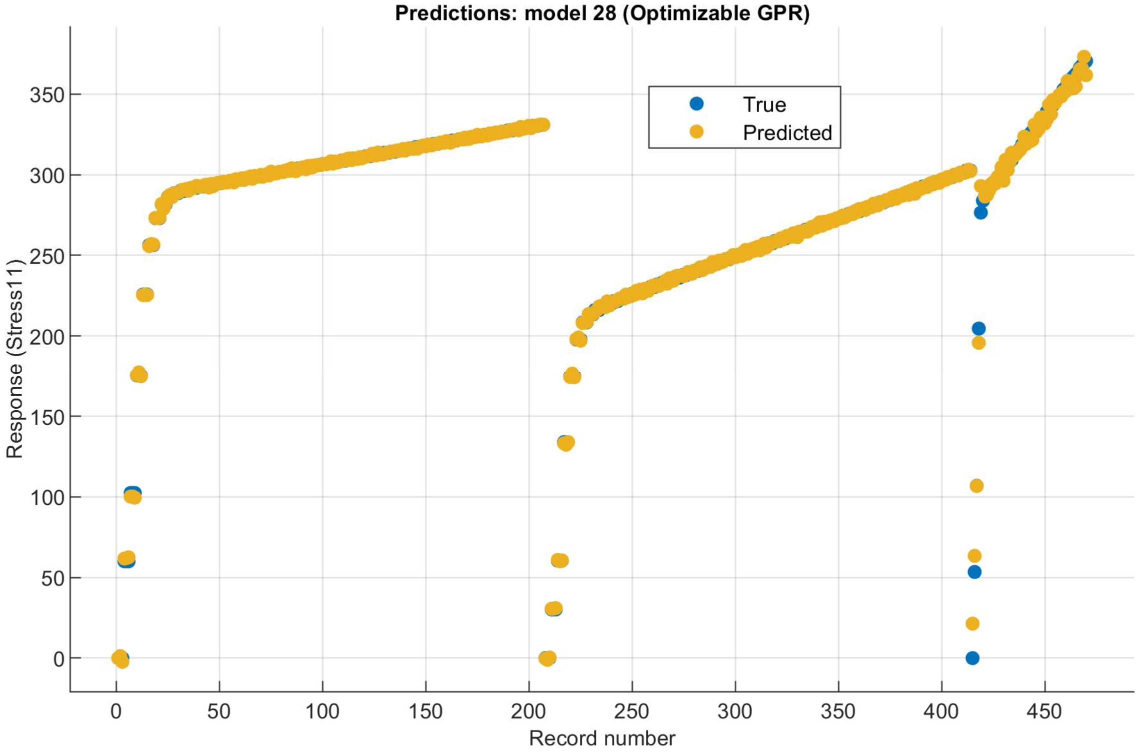


**Figure 7**. Predicted versus real values for the new training dataset and the ML model "mdlGPR28".

The training and validation provide excellent values for the root mean square error.

vrmse = 1.5285,

RMSEValid = 0.17497.

This model, "mdlGPR28," provides very good predictions across all zones of the stress-strain relationship concerning CF/PA11. This time, the predictions for data points related to PA11, although not included in the training set, are accurate. Lines 13 to 19 of the script "S193_addCrumb.m" calculate predictions for the data in "Tab11unkn" and the RMSE between the actual stress values in this table and the predictions, which is 2.25.

We have also considered a smaller table "Tab11unkn2," comprising only 46 data points, to enhance the graphical quality of Figure 8. The predictions are accurate in this instance as well. The two vectors of predicted and actual stress values have the following statistics:

- The prediction errors belong to the interval [- 5.6, 5.4] MPa;
- The root mean square error is 2.05. The values in Table 9 strongly support this assertion, even more convincingly than the graphical representation in Figure 8.

**Table 9**.* Predictions made by "mdlGPR28" for data points related to PA11 that are not part of the training set.

| Nr | Strain11 | Stress11 | Prediction |
|---|---|---|---|
| 1 | 5e-14 | 5.3419e-09 | -0.15513 |
| 2 | 0.0005 | 53.419 | 53.591 |
| 3 | 0.0021863 | 204.59 | 204.53 |
| 4 | 0.0036863 | 255.01 | 255.21 |
| 5 | 0.0066863 | 283.88 | 284.15 |
| 6 | 0.0081863 | 286.8 | 286.79 |
| 7 | 0.011186 | 290.03 | 290.36 |

```
 8   0.012686   291.42   291.44
 9   0.015686   294.16   293.94
10   0.017186   295.53   295.55
11   0.020186   298.25   298.41
12   0.021686   299.62   299.66
13   0.024686   302.34   302.63
14   0.026186    303.7   303.68
15   0.029186   306.42   306.13
16   0.030686   307.79   307.74
17   0.033686   310.51   310.55
18   0.035186   311.87   311.86
19   0.038186   314.59   314.28
20   0.039686   315.96   315.97
21   0.042686   318.68   318.34
22   0.044186   320.04   320.02
23   0.047186   322.76   322.93
24   0.048686   324.12   324.12
25   0.051686   326.85   327.14
26   0.053186   328.21   328.27
27   0.056186   330.93   331.19
28   0.057686   332.29   332.29
29   0.060686   335.02   335.41
30   0.062186   336.38   336.37
31   0.065186    339.1   339.11
32   0.066686   340.46    340.4
33   0.069686   343.19   343.01
34   0.071186   344.55   344.52
35   0.074186   347.27   347.21
36   0.075686   348.63   348.63
37   0.078686   351.35   351.29
38   0.080186   352.72   352.78
39   0.083186   355.44   355.29
40   0.084686    356.8   356.83
41   0.087686   359.52   359.25
42   0.089186   360.88   360.87
43   0.092186   363.61   363.39
44   0.093686   364.97   364.96
45   0.096686   367.69   367.39
46   0.098186   369.05   369.07
```

* A fragment of the execution listing from the script "S193_addCrumb.m".

Aside from line #1, where a null value can approximate the prediction, the columns "Stress11" and "Prediction" are nearly identical.

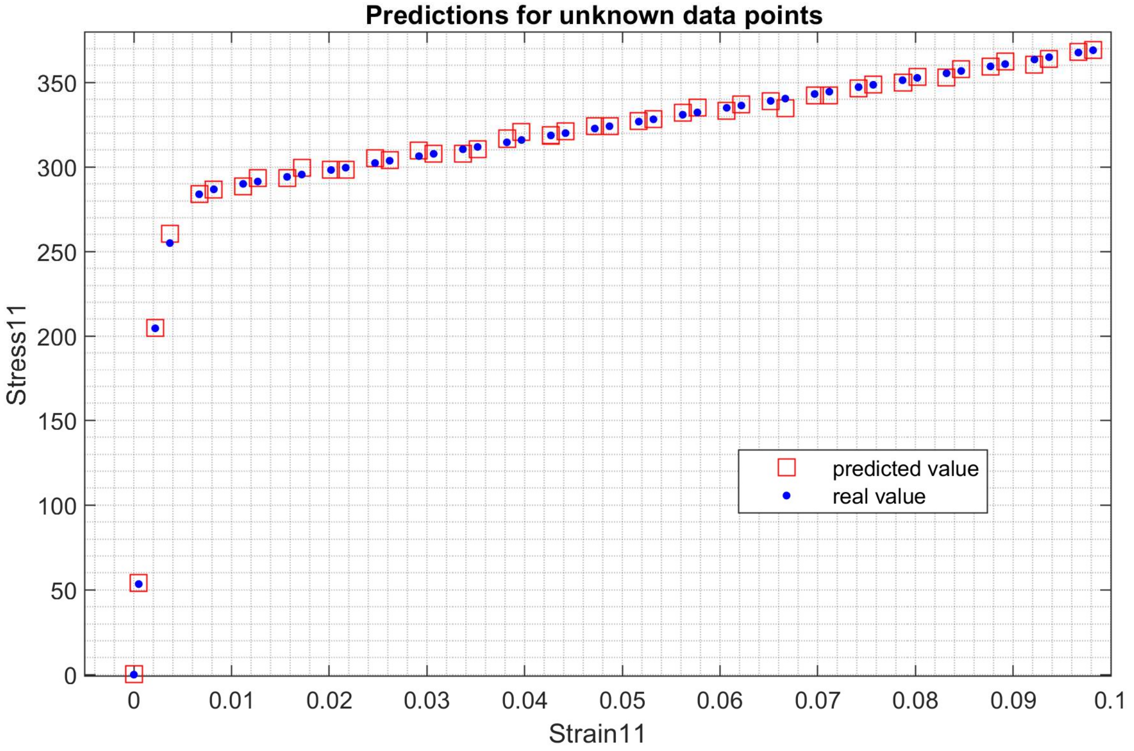


**Figure 8**. Predicted versus actual stress values for data points outside the training set (listed in Table 10)

Remark 4: The ML model "mdlGPR28" successfully made accurate predictions for CF/PA11 by utilizing a substantial dataset for the joint model ML/PA6–PA12 and only 56 additional datapoints specifically for PA11. This additional data represents 11% of the total 506 datapoints in the training set.

## 4. Discussion

The main objective of this work was to find an answer to the conjecture presented even in the Introduction section. The preparation of the data set for all the ML models is detailed in subsection 3.2, which includes sequential actions related to data importation and table construction, making it easy for readers of the report to understand.

Section 3.3 presents a gradual resolution of the proposed problem as follows:

- Step 1 (section 3.3.1): The joint model ML/PA6–PA12 ("mdlNN5") is constructed using complete and combined training data for CF/PA6 and CF/PA12.

- Step 2 (section 3.3.2): The model ML/PA6–PA12 incorporates additional data points regarding CF/PA11 at the lower end of the strain range. The Gaussian Process Regression algorithm generates the model "ML/PA6–PA12–pa11" ("mdlGPR7") using only 9 additional data points regarding CF/PA11.

- Step 3 (section 3.3.3): The model ML/PA6–PA12 incorporates additional data points related to CF/PA11, which are well-distributed across various regions of the strain range. The Gaussian Process Regression algorithm generates the model "ML/PA6–PA12–pa11" ("mdlGPR28") using 56 data points concerning CF/PA11, from a total of 506 that make up the training data.

Remark 1 highlights that Step 1 concludes with a negative result regarding the proposed conjecture. The model ML/PA6–PA12 can only predict based on the data it was trained on, which is a tautological statement. However, we presented this step to emphasize that the predictions concerning CF/PA11 are unbiased and entirely random. At the same time, this model represents a significant achievement because it can simultaneously predict both materials, CF/PA6 and CF/PA12. It replaces two RNN models, such as "mdlNN4".

Step 2 adds nine data points near the null strain, with only three data points originating from simulations, which can impose fixed points that the function *f* will pass through. We expect these fixed points to eliminate the indefinite tendency of the predictions.

Remark 2 underlines the partial success of our artifice. The model ML/PA6–PA12–pa11 is implemented by "mdlGPR7" generated by an optimized Gaussian Process Regression. Figure 6 confirms our expectation; however, the predictions for datapoints related to CF/PA11 and not included in the training set are inaccurate. These predictions only fit the stress values around the data points close to the null strain. Remark 3 indicates that only a few datapoints related to CF/PA11 are insufficient for good accuracy. We must extend the procedure to include more data points in the new training set.

To improve the accuracy of the model ML/PA6–PA12–pa11, additional data points related to PA11, covering a broader range of strain values, are added in Step 3. In the test constructed in Section 3.3.3, the data points are well distributed across various strain regions. Therefore, function *f* will be constrained by more fixed points. We presented an example where *n*11=56 data points from "Tab11Pert", distributed uniformly across the strain range, are added to the new training dataset. The result is very good: 56 data points out of a total of 506 in the training set change the behavior of the ML model. The new model, "mdlGPR28", also generated by an optimized Gaussian Process Regression, exhibits very good accuracy, as shown in Figures 7 and 8 and Table 9.

Finally, with 11% of data points related to CF/PA11 included in the training set, we obtained very good accuracy, as the statistics prove. This is not the lower bound of *n*11. A smaller number *n*11 may reduce the ML model's accuracy, but it will still provide acceptable stress predictions required by the current application.

The primary contribution of this work is the gradual resolution of the proposed conjecture. Our research was directed to find a small number of data points added to the ML/PA6–PA12 training data set, such that the model ML/PA6–PA12–pa11 allows the prediction of stress values for CF/PA11. The solution is not unique; it depends on the desired accuracy of predictions. According to the current application, the value of *n*11 may be diminished, which offers important flexibility.

Finally, we have a single ML model capable of recognizing (simultaneously) and predicting the stress value of three composite materials: CF/PA6, CF/PA12, and CF/PA11. The smaller number of data points related to CF/PA11 compared to the other two materials can reduce the cost and simulation time when preparing the data for the presented ML model.

This work also developed ML models that simultaneously address two or three composite materials; regression models recognize them and enable the prediction of their mechanical behavior. To our knowledge, this is a novel approach and could be useful in various applications.

A collateral contribution of our work that springs from creating the proposed ML models is the general implementation of function *f* in the current programming language for any composite material. However, this is applicable only when the ML model provides good prediction accuracy. The general implementation of *f* could be necessary in the upcoming phases of the D3T4H2S project, especially when addressing optimization problems, which will require fast objective function calculations.

Whether focused on composite materials or not, the project partners interested in developing ML models can find inspiration in our work for other applications. They can access the MATLAB scripts, functions, and other files related to this work, which are available in the "`Autom_Mtlb`" folder attached to this report.

*The archive "`Autom_Mtlb.zip`" contains all the files implementing the prediction models and a GUIDE to using the scripts as well.*

## Appendix A

**Table A1.** Generation of a step-wise linear regression model (execution listing)

```
>>  S3_PA6Model_SW
1. Adding Strain11, FStat = 104.439, pValue = 3.668277e-19

Linear regression model:
    Stress11 ~ 1 + Strain11

Estimated Coefficients:
                   Estimate      SE      tStat       pValue

                   ________    ______    ______    __________


    (Intercept)     252.83     5.2728    47.949    1.8763e-96
    Strain11        936.55     91.643     10.22    3.6683e-19

Number of observations: 161, Error degrees of freedom: 159
Root Mean Squared Error: 33.8
R-squared: 0.396,  Adjusted R-squared: 0.393
F-statistic vs. constant model: 104, p-value = 3.67e-19
```

**Table A2.** Comparison of real values and predicted values (using the ML model "mdlsw")

```
     i      Stress11     StressPred
     1      153.49        254.56
     2      237.84         256.9
     3      273.07        259.24
     4      284.93        261.59
     5      289.06        263.93
     6      291.02        266.27
     7      292.39        268.61
     8      293.61        270.95
     9      294.78        273.29
................................
    33      322.61        329.48
    34      323.77        331.83
    35      324.93        334.17
    36      326.09        336.51
    37      327.24        338.85
    38       328.4        341.19
    39      329.56        343.53
    40      330.72        345.87
```

## Appendix B

**Table A3.** Usage of the training function for RNNs.

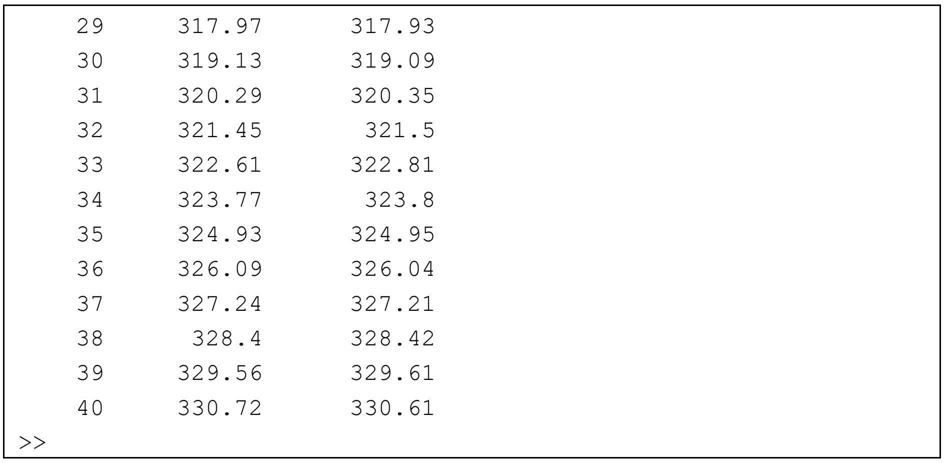
```
            The parameters of the "fitrnet" function
regressionNeuralNetwork = fitrnet(...
    predictors, ...
    response, ...
    'LayerSizes', 226, ...
    'Activations', 'relu', ...
    'Lambda', 6.366956171376093e-08, ...
    'IterationLimit', 1000, ...
    'Standardize', true);
```

**Table A4.** Execution listing of the script "S5_PA6DamageRNN4.m".

```
>>S5_PA6DamageRNN4
vrmse=
       1.4922
RMSEValid=
      0.24295

    Nr    RealValue    RNN4

    __    _________    ______
     1     153.49      151.69
     2     237.84      237.48
     3     273.07      276.44
     4     284.93      284.89
     5     289.06      289.16
     6     291.02      291.04
     7     292.39      292.38
     8     293.61      293.56
     9     294.78      294.74
    10     295.95      295.92
    11     297.11       297.1
    12     298.27       298.3
    13     299.43      299.46
    14     300.59       300.6
    15     301.75      301.77
    16      302.9      302.95
    17     304.06      304.13
    18     305.22      305.28
    19     306.38      306.22
    20     307.54      307.48
    21      308.7      308.66
    22     309.86      309.84
    23     311.02      311.01
    24     312.18      312.19
    25     313.34      313.34
    26     314.49      314.49
    27     315.65      315.64
    28     316.81      316.78
```



```
      29      317.97      317.93
      30      319.13      319.09
      31      320.29      320.35
      32      321.45       321.5
      33      322.61      322.81
      34      323.77       323.8
      35      324.93      324.95
      36      326.09      326.04
      37      327.24      327.21
      38       328.4      328.42
      39      329.56      329.61
      40      330.72      330.61
>>
```

## Appendix C

**Table A5.** * The training data set Tab6_12_11train (423 datapoints)

| Density | Em | Poisson | YieldS | Hm | He | Lhm | Strain11 | Stress11 |
|---:|---:|---:|---:|---:|---:|---:|---:|---:|
| 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 5e-14 | 6.0014e-09 |
| 1.1544e-09 | 3048.7 | 0.36448 | 25.413 | 3015.9 | 0.16726 | 991.14 | 0 | 0 |
| 1.1421e-09 | 3054.9 | 0.37688 | 24.658 | 3056.5 | 0.17311 | 999.42 | 0 | 0 |
| 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.0005 | 60.014 |
| 1.1188e-09 | 3041.9 | 0.37642 | 25.179 | 3030.9 | 0.17165 | 995.69 | 0.0005 | 60.014 |
| 1.1471e-09 | 2960.5 | 0.37305 | 24.532 | 2973.2 | 0.16691 | 983.89 | 0.0005 | 60.014 |
| 1.14e-09 | 3000 | 0.37 | 25 | 3000 | 0.17 | 1000 | 0.00085301 | 102.38 |
| | | : | : | : | : | | | |
| 1.03e-09 | 1250 | 0.37 | 51 | 1500 | 0.28 | 1680 | 0 | 0 |
| 1.0444e-09 | 1242.9 | 0.3663 | 51.064 | 1478.5 | 0.27976 | 1707.2 | 0 | 0 |
| 1.0327e-09 | 1259.3 | 0.37103 | 51.838 | 1525.5 | 0.28142 | 1672.6 | 0 | 0 |
| 1.03e-09 | 1250 | 0.37 | 51 | 1500 | 0.28 | 1680 | 5e-14 | 5.3419e-09 |
| 1.0211e-09 | 1274.3 | 0.37461 | 51.613 | 1496.3 | 0.28098 | 1689.4 | 5e-14 | 5.3419e-09 |
| 1.0478e-09 | 1248.3 | 0.36628 | 51.786 | 1480.7 | 0.28463 | 1696.1 | 5e-14 | 5.3419e-09 |
| 1.03e-09 | 1250 | 0.37 | 51 | 1500 | 0.28 | 1680 | 0.0005 | 53.419 |
| 1.0375e-09 | 1229.2 | 0.36385 | 51.487 | 1491 | 0.27603 | 1657.3 | 0.0005 | 53.419 |
| 1.0417e-09 | 1238 | 0.36733 | 51.601 | 1501.1 | 0.28184 | 1701.3 | 0.0005 | 53.419 |

* Only the last 9 lines are datapoints related to CF/PA11

## References

1. Nachtane, M.; Tarfaoui, M.; Abichou, A.; Vetcher, A.; Rouway, M.; Aâmir, A.; Mouadili, H.; Laaouidi, H.; Naanani, H.; An Overview of the Recent Advances in Composite Materials and Artificial Intelligence for Hydrogen Storage Vessels Design. J. Compos. Sci. 2023, 7, 119., https://doi.org/10.3390/jcs7030119
2. Brayek, B.E.B.; Sayed, S.; Mînzu, V.; Tarfaoui, M. Machine Learning Predictions for the Comparative Mechanical Analysis of Composite Laminates with Various Fibers. Processes 2025, 13, 602. https://doi.org/10.3390/pr13030602
3. Liang, Y. W. X. P. Y.; et al., "A review on recent applications of machine learning in mechanical properties of composites," Polymer Composites, Sep. 2024.
4. Dotoli, R.; Gerardi, A.; Polydoropoulou, P.; Lampeas, G.; Pantelakis, S.; Rovira, A.C. Virtual Testing Activities for the Development of a Hybrid Thermoplastic Composite Material for the NHYTE Project. In IOP Conference Series: Materials Science and Engineering; IOP Publishing Ltd: 2021; 1024, 012007.10.1088/1757-899X/1024/1/012025

5. Goodfellow, I.; Bengio, Y.; Courville, A. Machine Learning Basics. In Deep Learning; The MIT Press: Cambridge, MA, USA, 2016; pp. 95–161. ISBN 978-0262035613.
6. Newbold, P.; Carlson, W.L.; Thorne, B. Multiple Regression. In Statistics for Business and Economics, 6th ed.; Pfaltzgraff, M.; Bradley, A., Eds.; Pearson Education, Inc.: Upper Saddle River, NJ, USA, 2007; pp. 454–537.
7. The MathWorks Inc. (2024a). Stepwise Regression Toolbox Documentation, Natick, Massachusetts: The MathWorks Inc.; https://www.mathworks.com/help/stats/stepwise-regression.html
8. Goodfellow, I.; Bengio, Y.; Courville, A. Example: Linear Regression. In Deep Learning; The MIT Press: Cambridge, MA, USA, 2016; pp. 104–113. ISBN 978-0262035613.
9. The MathWorks Inc. (2024a). Regression Neural Network Toolbox Documentation, Natick, Massachusetts: The MathWorks Inc.; https://www.mathworks.com/help/stats/regressionneuralnetwork.html
10. Vahed, R.; Zareie Rajani, H. R.; and A. S. Milani, Can a Black-Box AI Replace Costly DMA Testing? A Case Study on Prediction and Optimization of Dynamic Mechanical Properties of 3D Printed Acrylonitrile Butadiene Styrene, Materials, vol. 15, no. 8, Apr. 2022.https://doi.org/10.3390/ma15082855
11. Ho, N.X.; Le, T.T.; Le, M.V. Development of an Artificial Intelligence-Based Model for the Prediction of Young's Modulus of Polymer/Carbon-Nanotubes Composites. Mech. Adv. Mater. Struct. 2022, 29, 5965–5978 https://doi.org/10.1080/15376494.2021.1969709SPr
12. Yang, Y. Kim, S. Ryu, and G. X. Gu, 'Prediction of composite microstructure stress-strain curves using convolutional neural networks,' Mater Des, vol. 189, Apr. 2020, https://doi.org/10.1016/j.matdes.2020.108509
13. Campbell, F.C. Structural Composite Materials; ASM International: Materials Park, OH, USA, 2010; ISBN 978-1615031405
14. Mînzu, V.; Arama, I. A Machine Learning Algorithm That Experiences the Evolutionary Algorithm's Predictions—An Application to Optimal Control. Mathematics 2024, 12, 187. https://doi.org/10.3390/math12020187
15. Mînzu, V.; Arama, I.; Rusu, E. Machine Learning Algorithms That Emulate Controllers Based on Particle Swarm Optimization—An Application to a Photobioreactor for Algal Growth. Processes 2024, 12, 991. https://doi.org/10.3390/pr12050991