Data-driven design of a supercompressible meta-material

Martin van der Schelling (m.p.vanderschelling@tudelft.nl)[1] and Miguel A. Bessa (miguel_bessa@brown.edu)[2]

[1]Doctoral Researcher at Delft University of Technology, the Netherlands
[2]Associate Professor at Brown University, USA

ENGN2350 Data-Driven Design and Analysis of Structures and Materials — October 9, 2024

Objective: Use machine learning and optimization to design a supercompressible meta-material.
General Instructions: Each group has to deliver **one PDF report** and a **ZIP-file including their code**. The code should be easy to read, properly commented and it should be possible to replicate the results of the report.

Introduction

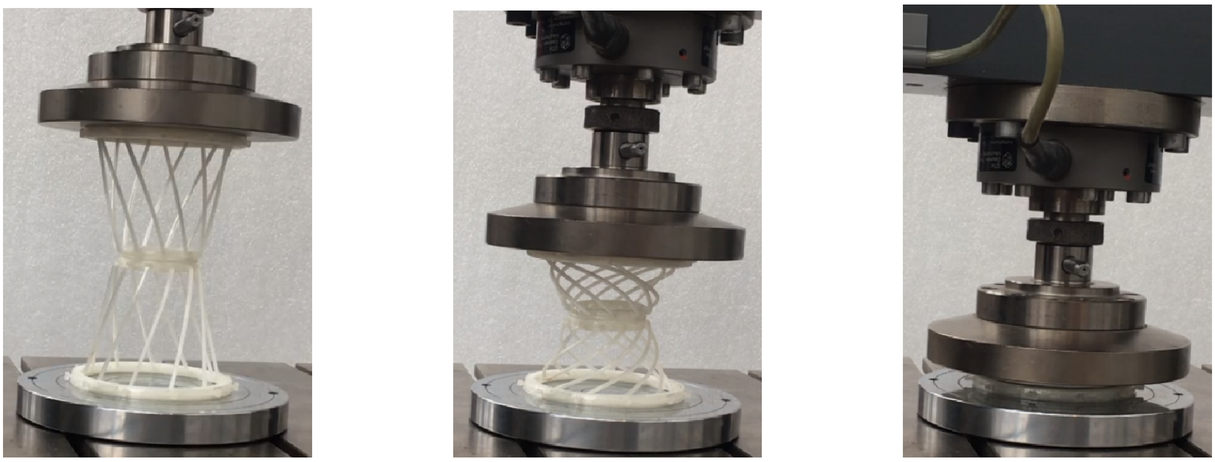
For years, creating new materials has been a time consuming effort that requires significant resources because we have followed a trial-and-error design process. Now, a new paradigm is emerging where machine learning is used to design new materials and structures with unprecedented properties. Using this data-driven process, a new super-compressible meta-material was discovered despite being made of a fragile polymer.

Figure 1 shows the newly developed meta-material prototype that was designed with the above-mentioned computational data-driven approach [1] and where experiments were used for validation, not discovery. This enabled the design and additive manufacturing of a lightweight, recoverable and super-compressible meta-material achieving more than 90% compressive strain when using a brittle base material that breaks at around 4% strain. Within minutes, the machine learning model was used to optimize designs for different choices of base material, length-scales and manufacturing process. Current results show that super-compressibility is possible for optimized designs reaching stresses on the order of 1 kPa using brittle polymers, or stresses on the order of 20 MPa using carbon like materials.

F3DASM framework

The framework for data-driven design and analysis of structures and materials (`f3dasm`) is an attempt to develop a systematic approach of inverting the material design process [2]. The framework integrates the following fields:

- **design of experiments**, where input variables describing the microstructure, properties and external conditions of the system to be evaluated are sampled and where the search space is determined;



(a) Undeformed metamaterial. (b) 50% deformation. (c) More than 90% deformation.

Figure 1: 3D printed super-compressible meta-material designed with machine learing [1].

- **data generation**, typically through computational analyses, resulting in the creation of a material response database;
- **machine learning**, in which a surrogate model is trained to fit experimental findings;
- **optimization**, where we try to iteratively improve the design.

In this assignment, the data generation part consists of running ABAQUS simulation scripts. These simulation have been done beforehand[1]. Instead, we will focus on the machine learning part of the framework. The package is written in Python.

Aim of the project

This project focuses on a hands-on approach to understand how to use machine learning and optimization methods to design this meta-material using a data-driven framework. At the end of this project, you will gain:

- Introductory knowledge on design of experiments: uniform sampling vs. Sobol sequences.
- Basic knowledge on machine learning: distinction between supervised regression and classification, simple dataset preprocessing, and selection of an appropriate algorithm.
- Elementary knowledge on optimization

Provided data and resources

Now that you covered the basics of Machine Learning, you want to use this knowledge to design your own supercompressible meta-materials. To help you accomplish your task, we will provide two datasets that you should use to explore the design space and come up with creative solutions:

- A dataset with 50000 experiments that is parametrized by 7 parameters (`supercompressible_7d`)
- A dataset with 1000 experiments that is parametrized by 3 parameters (`supercompressible_3d`)

The data has been created with the `f3dasm` framework package. You will not be able to generate additional data points.

Design-of-experiments

The supercompressible meta-material is parameterized by 5 geometric parameters and 2 material parameters [1]. The geometry is defined by the top and bottom diameters, D_1 and D_2, the height P and the cross-section parameters of the vertical longerons: the cross-sectional area A, moments of inertial I_x and I_y, and torsional constant J, see figure 2. The isotropic material is defined by its elastic constants: Young's modulus E and shear modulus G.

Due to the principle of superposition both the geometric and material parameters can be scaled by one of its dimensions/properties (here D_1 and E). Therefore, the variables that you will find in the dataset are:

$$\frac{D_1 - D_2}{D_1}, \ \frac{P}{D_1}, \ \frac{I_x}{D_1^4}, \ \frac{I_y}{D_1^4}, \ \frac{J}{D_1^4}, \ \frac{A}{D_1^2}, \frac{G}{E}$$

This is a 7-dimensional problem and learning the response surface may require a significant amount of training points[2]. Therefore, you will also consider a simpler version of the problem in 3 dimensions, defined by constraining the longerons' cross-section to be circular with diameter d, and choosing a particular material, leading to the following 3 features:

$$\frac{d}{D_1}, \frac{D_2 - D_1}{D_1}, \ \frac{P}{D_1}$$

The bounds and parameter names for each design variable can be found in the `domain.pkl` *pickle* files.

[1] You will not need to setup nor run computer simulations to create new data.

[2] Remember the "curse of dimensionality"!

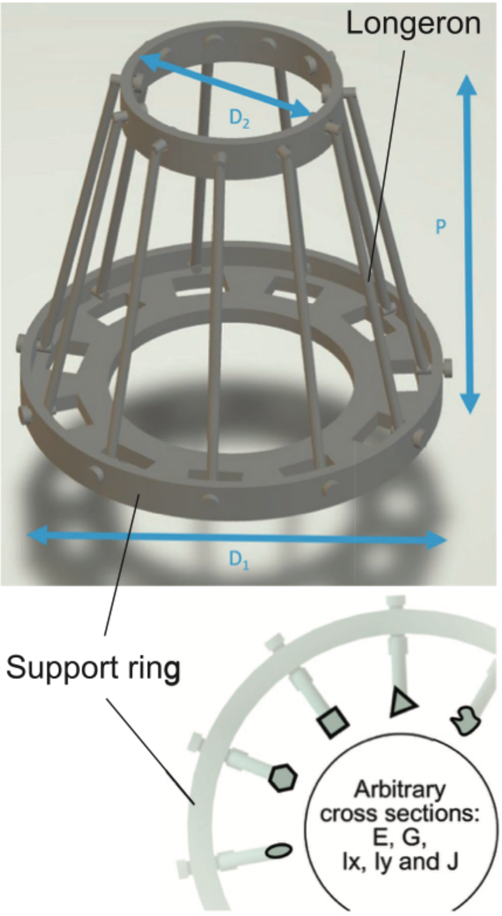


Figure 2: Supercompressible metamaterial building block with generalized cross-section for the longerons (x is the radial direction; y is the tangential direction) [1].

Input data

Using Sobol sequence[3] we have sampled the design space. Table 1 and 2 show the names of the paramaters for the 7D and 3D dataset respectively.

expression	parameter name
$\frac{D_1 - D_2}{D_1}$	`ratio_top_diameter`
$\frac{P}{D_1}$	`ratio_pitch`
$\frac{I_x}{D_1^4}$	`ratio_Ixx`
$\frac{I_y}{D_1^4}$	`ratio_Iyy`
$\frac{J}{D_1^4}$	`ratio_J`
$\frac{A}{D_1^2}$	`ratio_area`
$\frac{G}{E}$	`ratio_shear_modulus`

Table 1: Input variables in the 7-dimensional dataset: `input.csv`

[3] This is done with help of the SALib library.

expression	parameter name
$\frac{D_1-D_2}{D_1}$	ratio_top_diameter
$\frac{P}{D_1}$	ratio_pitch
$\frac{d}{D_1}$	ratio_d

Table 2: Input variables in the 3-dimensional dataset: `input.csv`

The samples and number of points considered can be found in the `input.csv` file.

Output data

For each data point (i.e. for each material design) we can use nonlinear finite element analyses to predict the complete buckling and post-buckling behavior. From the analyses, we can understand if a material is coilable and compute the critical buckling stress[4] σ_{crit} and the energy absorbed E_{abs}. The datasets have information about these three outputs.

For the 7-dimensional problem, the coilability of a material is simply defined by a Boolean variable, i.e. the material can either be or not be coilable. For the 3-dimensional problem, we can extract more information and understand also if within the coilable materials there is yielding (plasticity) or if they are reversible (elastic).

Concerning the output quantities (targets), the critical buckling stress σ_{crit} is deterministic and it is directly provided by the finite element analyses, while the energy absorbed E_{abs} is a stochastic variable (the uncertainty comes from geometric imperfections). This variable has to be computed from the stress-displacement curve (area under the curve) provided by the finite element analyses. The strategy to compute this variable is robust (we interpolate the stress-strain curve using a piecewise cubic Hermite interpolating polynomial and then compute the integral by applying Simpson's rule), but prone to errors.

Due to unsuccessful simulations, there are missing points in the datasets (they are stored as `NaN` in the data). You can simply ignore them or use any strategy that seems appropriate to complete the missing data. Table 3 shows the names of the parameters for both of the datasets.

expression	parameter name
coilability	coilable
σ_{crit}	sigma_crit
E_{abs}	energy

Table 3: Output variables in the datasets

The values of the `coilable` parameter are encode in the following way

- 0 = not coilable
- 1 = coilable
- 2 = coilable (but yields)

The post-processed output quantities of the simulations can be found in the `output.csv` file.

In summary, we provide you with two datasets, each containing 3 output variables (coilability, critical buckling stress σ_{crit} and energy absorbed E_{abs}): i) 7-dimensional problem (generalized longeron's cross section and different elastic material properties) and ii) 3-dimensional problem (circular longeron's cross section and fixed material properties). With this information you will design your own meta-material after finding the Machine Learning models to navigate the design space via machine learning classification and regression, followed by optimization. Note that you can optimize the meta-material with different goals in mind, e.g. reversibly coilable meta-material with highest energy absorption capability or maximum buckling strength. In order to get there, you will start by answering some questions.

[4] Critical buckling stress is defined as the critical buckling load divided by the area of the bottom ring of the meta-material.

Getting started

Reading

1. Read the paper Bayesian Machine Learning in Metamaterial Design: Fragile Becomes Supercompressible.

Installation

1. Make sure you have cloned and updated the `3dasm_course` GitHub repository.
2. Download the dataset from this Google Drive link.
3. In your `3dasm` environment, make sure you have installed the `f3dasm` package:

   ```
   pip install f3dasm==1.5.4
   ```

   Make sure you have version 1.5.4 installed. You can check this by running the following Python code:

   ```
   python3 -c "import␣f3dasm;␣print(f3dasm.__version__)"
   ```

   This will print the current version number to your terminal
4. In your `3dasm` environment, make sure you have installed the `f3dasm_optimize` extension package:

   ```
   pip install f3dasm_optimize
   ```

Questions to be answered

> *Important note for implementing your data-driven machine learning modeling: make sure you use the `f3dasm` package to structure your data-driven workflow and results as taught in class and practiced with the homeworks!*

Data characterization

1. An important step in data analysis[5] is the preprocessing step. Therefore, **before** considering to use any Machine Learning model, you are advised to do at least the following:

   1.1. For both datasets (3-d and 7-d) indicate the bounds of each input variable and the number of available points for each output variable.

   1.2. For the 3-d dataset plot all the points in 2-d scatter plots for all possible combinations of the input features, i.e. every pair of features as x and y of the scatter plots. Observe the sampling points and how they are distributed in the domain. Report your conclusions about the design of experiments strategy, but only show in the report the scatter plots for the first 100 points of the database (no need to plot every point in the database). Also include a 3-d scatter plot of the first 100 points (both plots should convey the same information). What can you conclude about the characteristics of the sampling method used (Sobol sequence)?

   1.3. For both datasets, create histograms for each continuous output variable (number of points whose output is within particular intervals). If you find strong outliers, propose a strategy to remove those outliers and save a new database without these outliers. Do a similar analysis for the categorical variables, but instead report the results in a table (number of points classified in each category).

   1.4. For the 3-d dataset, find the point corresponding to a material that is **reversibly coilable** (without yielding) that has maximum critical buckling stress, and another point that has maximum energy absorption capability (also for a reversibly coilable material). Report the feature values and every output value for **both** points.

Note that up to this stage you have not done any machine learning or optimization. Yet, the information you collected up to now is useful to establish a baseline for our next investigation.

Finding good machine learning models

2. Start your machine learning investigation by focusing on training different machine learning algorithms and evaluating which ones are better suited for this problem. Consider the typical split into training and testing data of 75% of data for training. Use the seed 123 for your splitting. You will do the following tasks for both datasets, but start with the 3-d dataset. Please create meaningful representations (plots) of the predictions, i.e. you should not include every plot you make during your investigation (just the key ones). Please keep your report clean and concise.

   2.1. Tasks to do for the 3-d dataset:

   2.1.1. For the **classification** problem:

   A. Fit at least the C-Support Vector Classifier from `scikit-learn` with the default hyperparameters and **2 other different classifiers** to your data (you are welcome to consider more!).

   B. Compare the performance of the trained models and select the best classifier. Explain eventual differences in the performance of the algorithms you considered.

   C. Repeat the two steps above, but now considering only two categories (*coilable*[6] and *not coilable*).

   2.1.2. For the **regression problems** (i.e. prediction of the critical buckling load and the energy absorbed):

[5] Although commonly underestimated.

[6] *Coilable (but yields)* must be considered as *coilable*

A. Fit at least the Gaussian process regressor from `scikit-learn` and **2 other regression models** to your data (you are welcome to consider more!). For the GPR, consider the Matern kernel with the smoothness parameter $\nu = 2.5$.

B. Compare the performance of the trained models and select the best regression model. Try to explain eventual differences in the performance of different algorithms.

C. Compare the best solutions found for each problem. Report the adequacy of each algorithm for noisy/noiseless problems.

2.2. Repeat the same steps (2.1.1 and 2.1.2) for the **7-dimensional problem**.

2.3. Compare the best solutions found for the 3-d and the 7-d problems. Reflect about the scalability of the algorithms that you used.

Note: For all the models that you train, you must report the hyperparameters, the error metrics, and the number of training and testing points used[7]. Don't be afraid to report the results of models that perform poorly: the knowledge gained from failed solutions is as valuable as the one gained from solutions that actually work. In some cases, it may not even be feasible to get a meaningful prediction with a particular algorithm (in that case, just report that no fit was possible to obtain with that algorithm). For **classification problems** you must always report *accuracy score* and, whenever possible, *precision score*, *recall score* and *F1 score*. You may consider creating a **confusion matrix** and interpreting it. For **regression problems** you must always report R^2 *score* and *mean squared error*. You should also provide a short description of a new machine learning algorithm considered in your project that was not covered in class.

Evaluating the best machine learning models you found

3. Machine Learning algorithms have several hyper-parameters and the quality of the predictions depends on the size of the training data. Understanding how the hyper-parameters and the data used affect the model prediction ability is important. Therefore, for the **3-dimensional** problem choose investigate the **C-Support Vector classifier** and the **Gaussian process regressor** algorithm to:

3.1. Study the influence of the **number of training points** in the performance of the algorithm[8].

3.2. Study the influence of the following hyper-parameters:

- for the C-Support Vector classifier the value of C
- for the GPR the smoothness parameter (ν) of the Matern kernel

3.3. Perform hyper-parameter optimization for the above mentioned hyper-parameters and relevant hyper-parameters for your best regression and classification model found in 2.1.1 and 2.1.2. Choose the testing accuracy (classification) and mean squared error (regression) as your objective. Consider **Random Search**, **TPESampler**[9] and **at least one other different optimization algorithm** and report the objective with respect to the number of hyperparameter optimization iterations for each of the runs.

Optimization

4. Commonly, the ultimate goal of having a description of the design space is to find the best solution for a given objective and under some constraints. Consider the following tasks for the **3-dimensional problem** only:

4.1. Before considering any optimization algorithm, just search[10] for the optimum points which have been classified by the C-Support Vector classifier as reversibly coilable (without yielding) that satisfy the following:

4.1.1. Higher critical buckling stress.

4.1.2. Higher energy absorption capability.

[7] Use tables whenever possible.

[8] You may not have yet gained intuition about the importance of the number of data points, but imagine that to collect the dataset you need to spend 1 minute to simulate each data point. A dataset with 50000 data points would need about 35 days of CPU-time to generate your data.

[9] You may use the built-in optimizers that can be found in the `f3dasm_optimize` package

[10] In other words: use the C-Support Vector classifier to select the points in the 3-d dataset that are reversibly coilable, and then find the maximum value for the corresponding output.

4.2. Now consider using the **Nelder-Mead optimizer**, the **CMAES optimizer** and **at least 2 other different optimization algorithms** on the Gaussian process regressor paired with the C-Support Vector classifier to determine:

4.2.1. The point with higher critical buckling stress (that is reversibly coilable).

4.2.2. The point with higher energy absorption capability (that is reversibly coilable).

4.3. Compare the solutions obtained in 1.4, 4.1 and 4.2.

4.4. Repeat 4.2 but now with your best-performing classifier and regressor in combination with the optimal hyper-parameters acquired in 3.3.

Report

Each group has to deliver **one PDF report** and a **ZIP-file including their code**.

- You can use Jupyter notebooks for the Python code. If you are using external libraries, mention them and the used version in the Jupyter notebook.
- The code should be easy to read, properly commented and it should be possible to replicate the results of the report.

References

[1] M.A. Bessa, P. Glowacki, and M. Houlder. Bayesian machine learning in metamaterial design: Fragile becomes supercompressible. *Adv. Mater.*, 0(0):1904845–, October 2019.

[2] M. P. van der Schelling, B. P. Ferreira, and M. A. Bessa. f3dasm: Framework for data-driven design and analysis of structures and materials. *Journal of Open Source Software*, 9(100):6912, 2024.