Report

Inferring Functional Properties from Fluid Dynamics Features

POLITECNICO
MILANO 1863

Francesco Montanaro

17/11/2021

1. Features Extraction

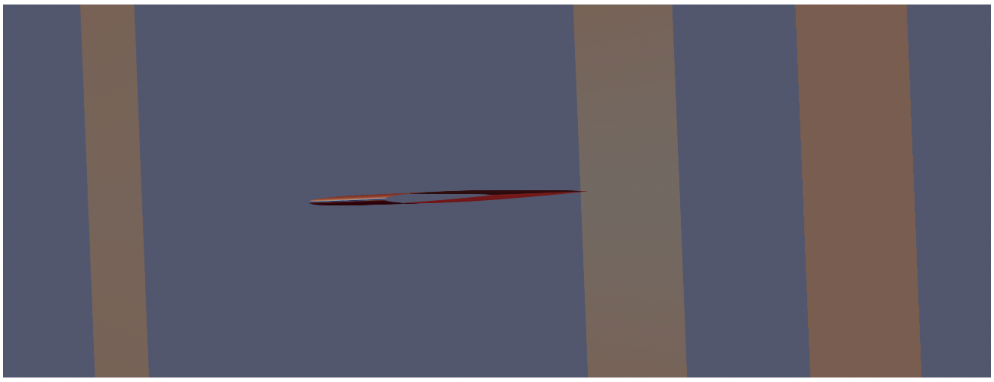
To allow Machine Learning models to predict the NACA codes of the airfoils, two features were considered and extracted from the CFD outcomes of the data available:

- **Regional averages** of the flow quantities.
- **Streamlines** arrival time.

The main tools used in the feature extraction part are the *Numpy* and *Vtk* libraries with the *Python* programming language.

1.1. Regional Averages

To extract region-averaged flow quantities, 3 sections were selected and drawn perpendicular to the airfoil chord ***c*** (as shown in the figure below), each one consisting of 3 regions for an overall of 24 regions. The first eight segments (regions from *1 to 8*) lay on a vertical section placed at *x=-c* with respect to the origin of the reference system; The second eight segments (regions from *9 to 16*) lay on a vertical segment placed at *x=c*, and the last eight segments (regions from *17 to 24*) lay on a segment placed at *x=10c*.
On each segment, the regions are symmetrically placed with respect to *y=0*, and their boundaries have y coordinates of *[500, 10, 1, 0.1, 0, 0.1, 1, 10, 500]*.
In particular, the original 3D mesh was cut at the x coordinates of interest in order to obtain three two-dimensional sections, to be divided into the corresponding regions.



The three sections and the airfoil in the space

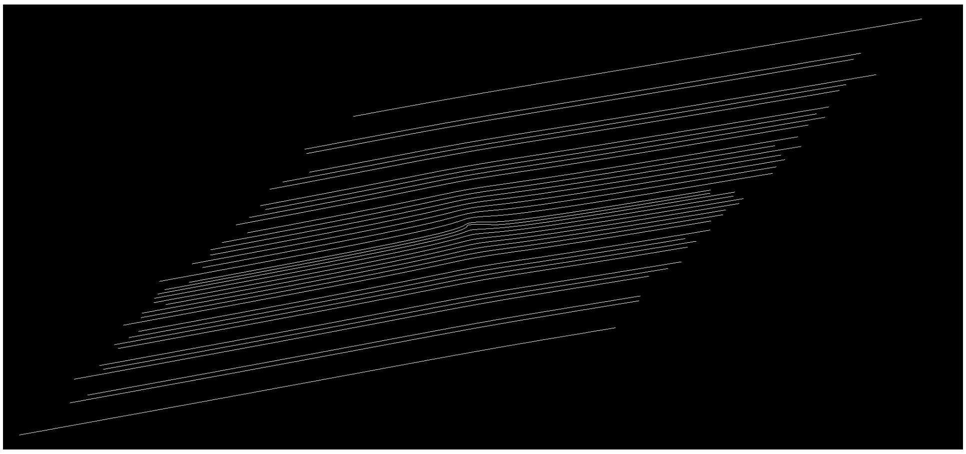
For each of the three sections, the cells of the mesh were readapted in the form of triangles from which the values of the flow quantities were extracted. Then, the centroid and the area of each triangle was computed by using the semiperimeter formula.

The cells belonging to the i-th region were extracted considering all the triangles whose centroid is contained within its boundaries.
Finally, the regional averages were computed by averaging the sum of the flow quantities by the sum of the areas of the triangles belonging to the considered region.

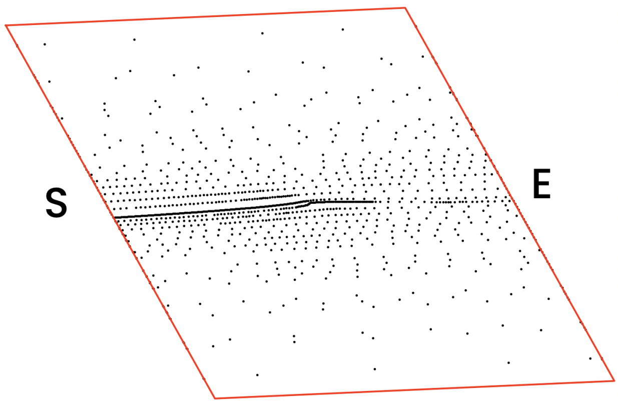
$$\bar{p}_k = \frac{\sum_i A_i p_i}{\sum_i A_i} \qquad \bar{U}_k = \frac{\sum_i A_i U_i}{A_i}$$

1.2. Streamlines

The streamlines were extracted by integrating the velocity fields of the points of the mesh by using a *VtkStreamTracer* object.
The resolution and the propagation of the streamlines was decreased, distributing the streamlines around the region of interest, in order to discard useless information and reduce the computational load.



The cutting sections, with respect to which consider the arrival time of each streamline, were then extracted. In particular, two sections were considered: the starting section *S* and the arrival section *E*, both orthogonal to the direction of the free stream (of 10 degrees). These, both of length *10c*, were placed respectively at a distance of *-3c* and *+3c* with respect to the origin of the reference system.



For each streamline, the velocity of the points that compose it was extracted, interpolating them with the values of the ones of the original mesh.

The temporal distance of the consecutive points was computed dividing their euclidean distance with the average of the magnitude of their velocity values:

$$\Delta t_{i,j} = \frac{2\sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}{|U_j| + |U_i|}$$

Finally, the arrival time was computed by summing the time distance of the consecutive points of the streamline belonging to the region delimited by the cutting sections S and E:

$$T_k = \sum_{i=1}^{n-1} \Delta t_{i,i+1}$$

The following table shows the distribution of the arrival times for different airfoils, described by their respective NACA numbers, related to the streamlines closest to the airfoils themselves (the ones affected by more perturbations).

NACA numbers	Streamline #33 arrival time	Streamline #34 arrival time	Streamline #35 arrival time
(0.0, 0.0, 0.05)	0.014	0.111	0.107
(0.03, 0.5, 0.14)	0.106	0.114	0.110
(0.05, 0.3, 0.14)	0.106	0.116	0.112
(0.06, 0.9, 0.07)	0.106	0.126	0.111
(0.08, 0.3, 0.09)	0.107	0.122	0.111

2. Regression

To infer functional properties given the features extracted from the available data, a neural network for regression was built. The architecture of the model is characterized by a hidden layer of 64 neurons and an output layer of three neurons, which represent the NACA numbers to predict. The input shape of the NN varies according to the features used: 72 neurons for the regional averages and 68 neurons for the streamlines.

The dataset was splitted into training and test sets each one with a percentage of *80%* and *20%* of the available samples. *20%* of the training samples were used for the validation set. Then, the data were normalized according to the mean and standard deviation values of the entire dataset.

The loss function used to optimize the model was the Mean Square Error (MSE):

$$L = \frac{1}{n}\sum_{i=1}^{n}(y - \widehat{y_i})^2$$

To limit overfitting several techniques were adopted such as dropout, with a rate of *0.2*, and early stopping with a patience of *10* epochs on the validation set loss.

2.1. Performance evaluation

To evaluate the model, a study was carried out by varying the quantity and the type of the samples to be used in the training phase. In particular, two approaches were adopted to observe how the cardinality and the composition of the training set could affect the performances. The model was then evaluated using both the features extracted from the available data.

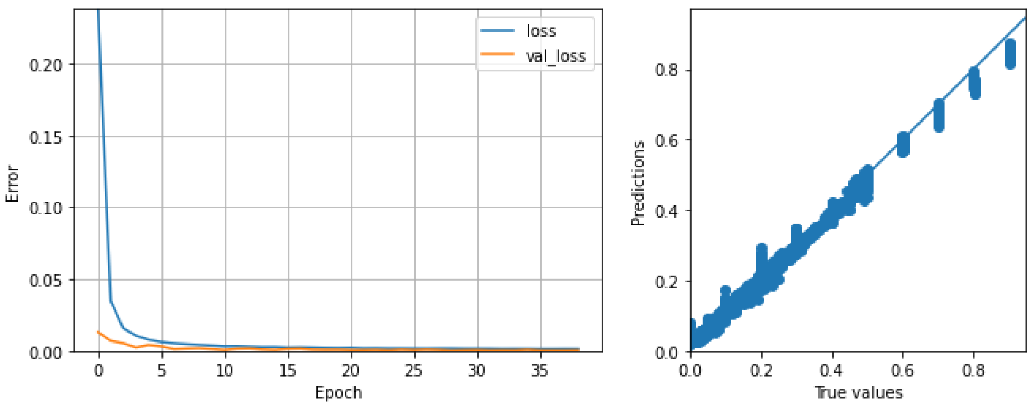
2.1.1. Interpolation

In this first approach, all the data used for the training set were sampled randomly from the original dataset. In particular, two different studies were carried out:

2.1.1.1. All data used

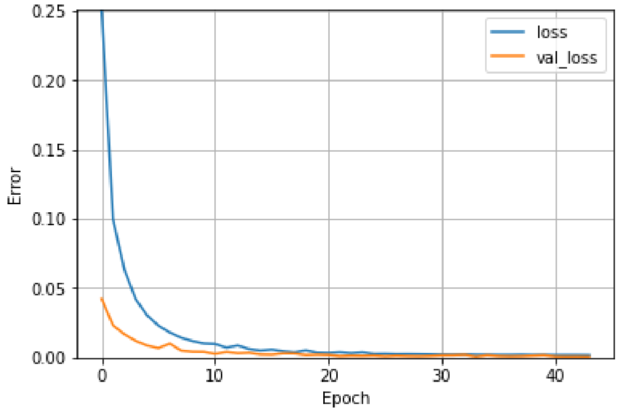
In this first part, all the data were used to train the model and sampled randomly to evaluate it in a scenario where a considerable number of samples were available. The performances obtained with the different features are the following:

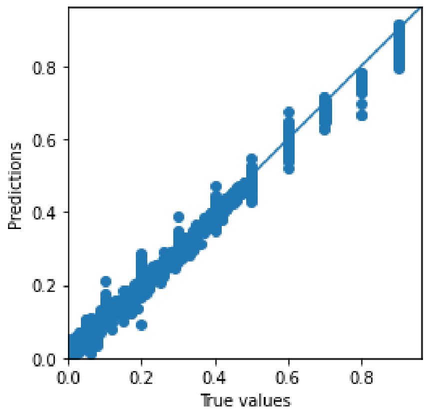
- **Regional Averages**



Loss: *0.000467*
Mean Absolute Error: *0.016030*
Mean Square Error: *0.000467*

- **Streamlines**



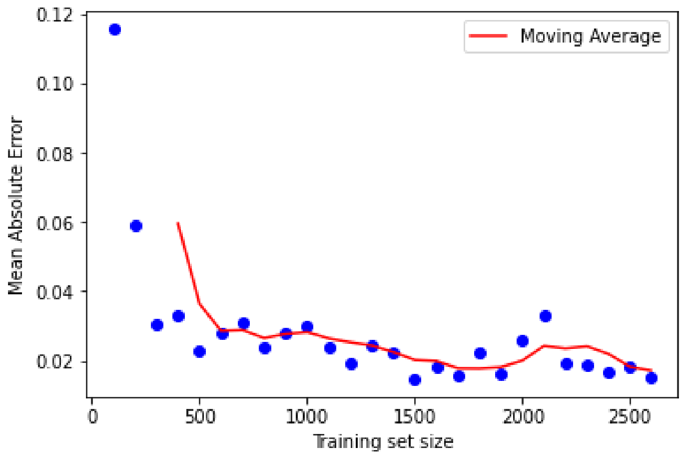


Loss: *0.000661*
Mean Absolute Error: *0.019026*
Mean Square Error: *0.000661*

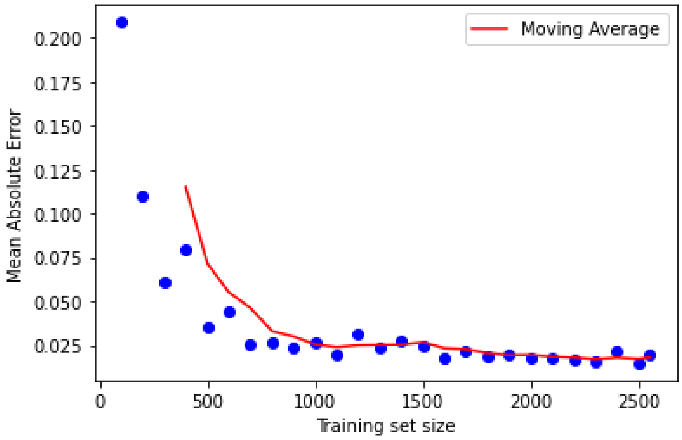
2.1.1.2. Variation of the training set size

In this case, it was studied how the training set size could affect the performance of the model. In particular, an incremental step size of 100 samples was chosen for each experiment, while the type of data used was randomly sampled as in the previous point.
Below is reported, for each feature considered, the trend of the Mean Absolute Error with respect to the training set size.

- **Regional Averages**

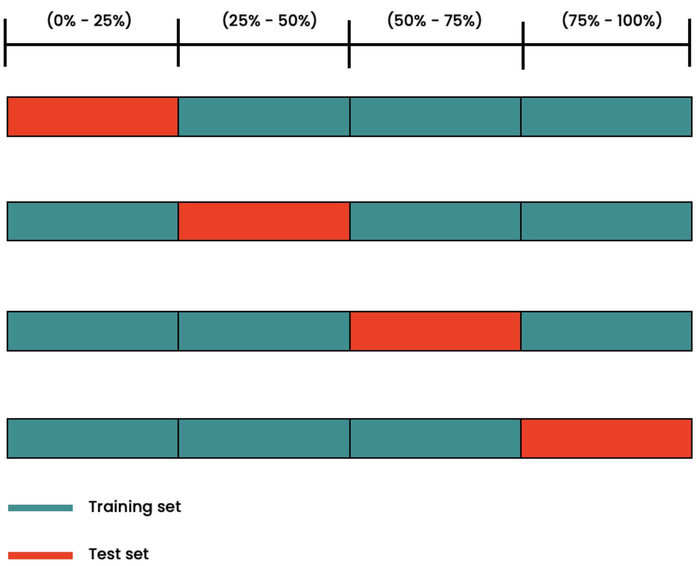


- **Streamlines**



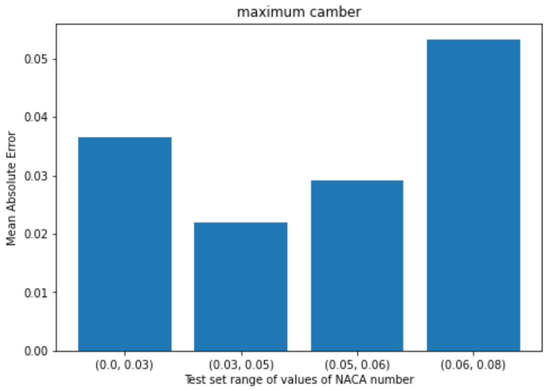
2.1.2. Extrapolation

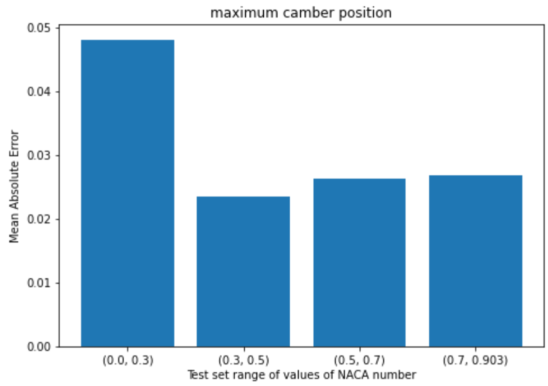
In this last part it was studied how the composition of the training set size could affect the performance. In particular, for each experiment, a subset of samples, whose NACA numbers belong to a certain range of values, was excluded from the training set and inserted in the test set for the prediction phase. For each NACA number, four not overlapping subsets were extracted, corresponding respectively to the range of values: *[(0% - 25%), (25% - 50%), (50% - 75%), (75% - 100%)]*, where the values related to the *0%* and *100%* represent the minimum and the maximum value of the specific NACA number in the dataset. In other words, the NACA codes were considered as numbers to define the four subsets into which to divide the dataset (as shown in the following figure):

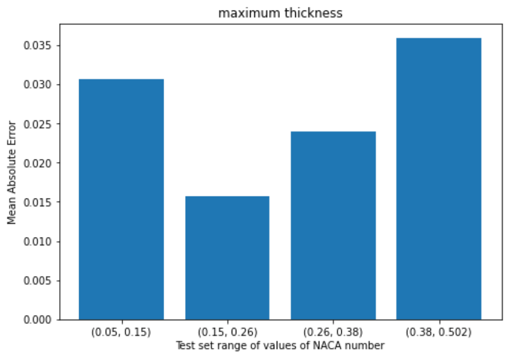


For each feature considered three bar charts are reported, one for each NACA number, representing the Mean Absolute Errors obtained excluding a specific range of values from the training set.

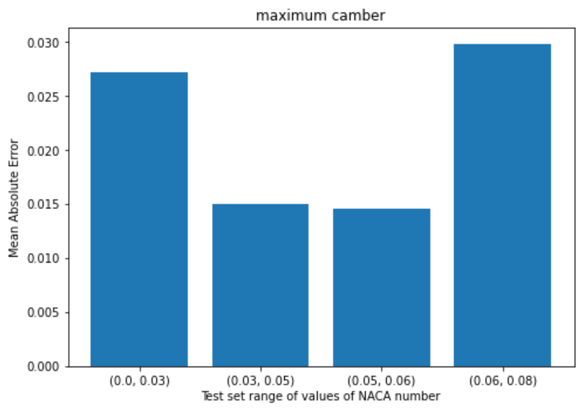
- ## Regional Averages

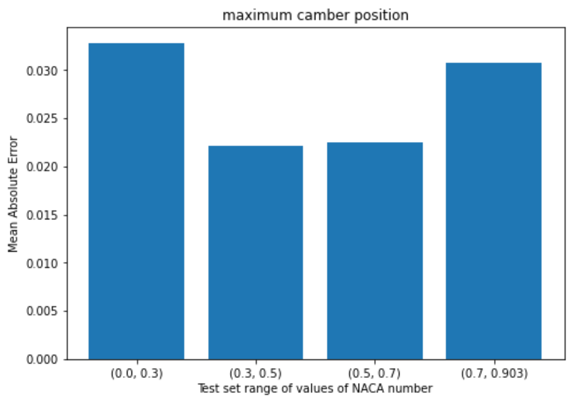


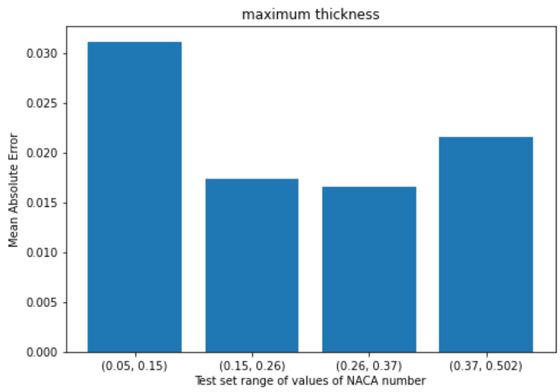




- ## Streamlines







2.1.3. Interpolation and Extrapolation comparison

In this last part it is shown how the performances vary according to the two approaches considered: interpolation and extrapolation for the same size of the training set.
In particular, for the extrapolation part, the results obtained from the different subsets were averaged in order to obtain a single result that could generalize well the overall performances:

- **Interpolation:**
  Loss: *0.000514*
  Mean Absolute Error: *0.017221*

- **Extrapolation:**
  Average Loss: *0.001427*
  Average Mean Absolute Error: *0.025125*

It is possible to notice how the performances obtained with the interpolation approach are in any case better than those obtained with the extrapolation one.