

GARTEUR RESEARCH GROUP AG60 EXPERIENCE AND ACTIVITIES ON MACHINE LEARNING AND DATA-DRIVEN APPROACHES FOR AERODYNAMICS AND UNCERTAINTY QUANTIFICATION

Esther Andrés Pérez[1], Domenico Quagliarella[2], Derrick Hines Chaves[3], Philipp Bekemeyer[3], Tommaso Taddei[4], Jacques Peter[5], Olivier Amoignon[6], Alessandro Alaia[7], Angela Scardigli[7], Rodrigo Castellanos[8] & Javier Nieto-Centenero[1,8]

[1]INTA — Instituto Nacional de Técnica Aeroespacial
[2]CIRA — Italian Aerospace Research Centre
[3]DLR — German Aerospace Center
[4]Sapienza University of Rome
[5]ONERA — Office national d'études et de recherches aérospatiales
[6]FOI — Swedish Defence Research Agency
[7]OPTIMAD Engineering, now ESTECO
[8]UC3M — Universidad Carlos III de Madrid

Abstract

This article provides an overview of the activities and results achieved in the GARTEUR AD/AG60 project, titled "Machine Learning and Data Mining Techniques for Aerodynamic Analysis and Uncertainty Quantification." The project partners developed and compared several machine learning models for surrogate modeling and uncertainty quantification and applied them to the AIRBUS XRF1 aircraft configuration of industrial interest. They assessed the feasibility of advanced Machine Learning methods for predicting aerodynamic behaviors in industrially relevant configurations. The article details the models created and the different approaches adopted, along with a comprehensive analysis and discussion of the results, highlighting their implications for advancing aerodynamic predictions in practical industrial scenarios.

Keywords: Machine Learning, Artificial Intelligence, Aerodynamics, Uncertainty Quantification, Optimization

1. Introduction

The main objective of GARTEUR AD/AG60 has been to advance research in machine learning and data-driven methodologies for aerodynamic analysis and uncertainty quantification. This project was specifically aimed at conducting in-depth research into the effectiveness of various techniques, such as deep learning, surrogate models, and other machine learning methods, for aerodynamic analysis and prediction. The objective of the research was to provide a comprehensive comparative analysis of these methodologies, highlighting their strengths and weaknesses in aerodynamics. Furthermore, the project aims to leverage data fusion, specifically a multi-fidelity approach, in surrogate modeling. This involves the expert management of heterogeneous data from various sources, including Computational Fluid Dynamics (CFD) simulations with varying accuracy, wind tunnel experiments, and flight-test-derived data. The integration of these disparate data sources is crucial for improving the accuracy and reliability of aerodynamic analyses, thereby advancing the understanding and optimization of aerospace systems. GARTEUR AD/AG60 was structured into the following tasks:

- **Task1 - Aerodynamic analysis and prediction of aerodynamic features**: Assessment of machine learning and data-driven techniques for aerodynamic analysis and prediction of aerodynamic features.

- **Task 2 - Data fusion and multi-fidelity heterogeneous data management**: Explore how to manage heterogeneous multi-fidelity data from different sources efficiently.
- **Task 3 - Uncertainty quantification and management**: Compare different methods to propagate uncertain inputs (Mach, Alpha) to obtain output distributions (C_p). Also, the uncertainty due to changes in the shape parameters can be analyzed.

In the following article, we will summarize the main contributions of each organization involved in this action group, while referring to the final report and the articles produced by the participants for a more in-depth analysis of the topics addressed.

2. AIRBUS

Airbus has made an essential contribution to the AG60 by providing part of the numerical experimental database for its XRF-1 (eXternal Research Forum) configuration to partners. The XRF-1 is a virtual aircraft and standard research model developed by Airbus for academic and research purposes, representative of a long-range wide-body aircraft; its three-view drawing is shown in Figure 1.

2.1 XRF-1 Databases

Two databases have been made available to AG60 partners and are briefly described below.

2.1.1 Database 1

Database 1 includes data obtained from CFD simulations, WTT (wind tunnel tests) with pressure-sensitive paint, and WTT with pressure taps. Table 1 illustrates the fields included in this database.
For the CFD data, the database incorporates several CFD campaigns (Table 2) representing stinger and non-stinger cases (representing WTT conditions), encompassing two different turbulence models and two Reynolds numbers. The CFD calculations to generate the database were run using the DLR TAU code [2].
As for the WTT data, a test campaign was conducted in the European Transonic Wind Tunnel, ETW, with a full-span, high-speed model representing the Airbus XRF-1 geometry at 1/37 scale. Test data were obtained for eight configurations and at Reynolds numbers ranging from just over 5 million to 40 million for Mach numbers between 0.5 and 0.96. The wind tunnel data includes data from 317 constant pressure shots and pressure sensitive paints (the latter was used in two of the cases as a technique development exercise). The ranges of test section flow conditions are reported in Table 3.

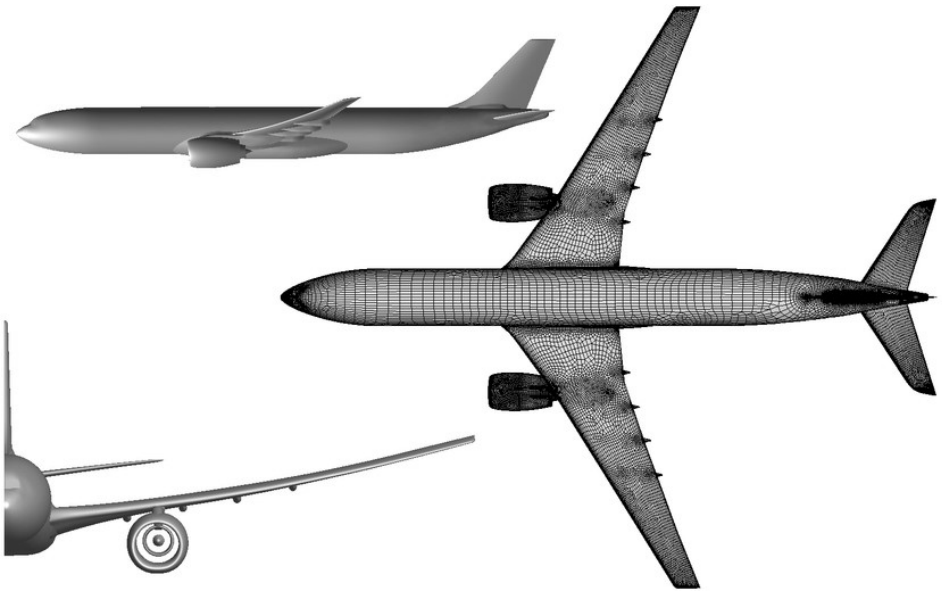


Figure 1 – XRF-1 Model. Source: [1]

Field	Description
Re	Reynolds number (float)
Mach	Mach number (float)
Alpha	Angle of Attack (float)
X, Y, Z	Coordinates for the given C_p value (float)
C_p	Pressure coefficient value at the given coordinates (float)
DATA SOURCE	CFD (data from CFD simulation) WTT_PSP (data from WTT using pressure sensitive paint) WTT_TAPPINGS (data from WTT using pressure tappings)

Table 1 – Database components.

Shape	Turbulence model	Re	Mach
XRF-1 1g shape, sting off, clean wing	SST	25e6	0.50, 0.60, 0.70, 0.80, 0.82, 0.84, 0.86, 0.88, 0.90, 0.92, 0.94, 0.96
		40e6	0.82, 0.86, 0.90, 0.94
	RSM-g	25e6	0.50, 0.70, 0.82, 0.88, 0.90, 0.94
XRF-1 1g shape, sting on, clean wing	SST	25e6	0.50, 0.60, 0.70, 0.80, 0.82, 0.84, 0.86, 0.88, 0.90, 0.92, 0.94, 0.96
		40e6	0.82, 0.86, 0.90, 0.94

Table 2 – CFD simulations. For each Mach number, a polar is run for $\alpha = [-12^\circ : 0.25^\circ : 15^\circ]$.

Parameter	Symbol / Unit	Experimental Range
Total temperature	T_0 (K)	115 and 296
Total pressure	p_0 (kPa)	180 to 400
Mach number	M	0.50 to 0.96
Reynolds number	Re	5.14×10^6 to 40×10^6 (at transonic Mach numbers)
Lift coefficient	C_L	$C_{L_{\min}}$ to $C_{L_{\max}}$

Table 3 – Ranges of Test Section Flow Conditions.

2.1.2 Database 2

In Database 2, a simplified version of the XRF-1 wing geometry, derived from the original Computer-Aided Design (CAD) model, is employed, in which the Flap Track Fairing (FTF) and the engine were previously removed. However, several geometric modifications are allowed relative to the baseline design (represented as the subscript '0'). These modifications include variations in the wing-span ratio b/b_0, the sweep angle increment $\Delta\Lambda = (\Lambda - \Lambda_0)$, and the thickness ratio t/t_0. A total of 60 different wing geometries are simulated, each using a structured mesh composed of $6,519$ grid points distributed across 41 wing sections. Table 4 summarizes the flow and geometric characteristics, including their respective intervals. Figure 2 illustrates the sketched representations of the geometric variations.

The aerodynamic data analyzed in Database 2 is obtained from computational simulations using RapidCFD software, which implements the BLWF code developed at the Central Aerohydrodynamic Institute (TsAGI) [3, 4]. The BLWF code solves the boundary-value problem for the complete velocity-potential equation, accounting for viscosity within the boundary-layer approximation while keeping the transition from laminar to turbulent flow fixed. This approach effectively captures local supersonic areas, shock waves, and flows with small separation zones. Extensive verification of the method has demonstrated its high reliability.

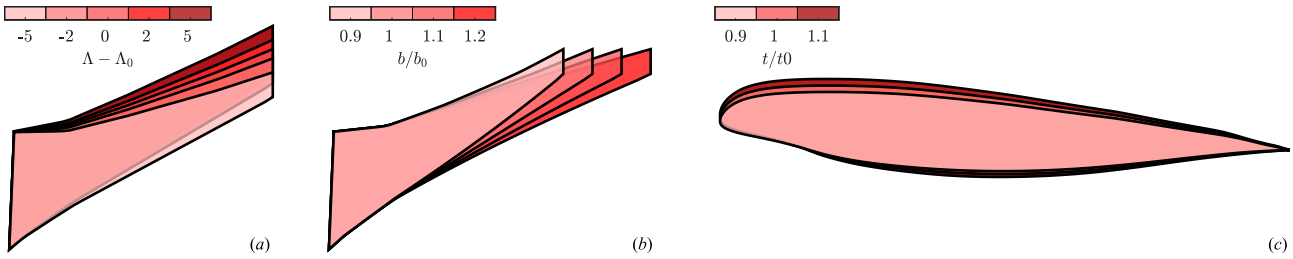


Figure 2 – Schematic representation of the different geometric parameters used to generate the database. The geometric parameters (denoted with subscript 0), describes an alteration of the nominal geometry, which are: (a) Increment of sweep angle $\Delta\Lambda = \Lambda - \Lambda_0$ in degrees, (b) scaling of the wing span b/b_0, and (c) scaling of thickness ratio t/t_0.

3. CIRA and INRIA

INRIA and CIRA, together with OPTIMAD, focused on applying Machine Learning (ML) techniques to improve the efficiency of aerodynamic design loops, with particular emphasis on data fusion to enhance the response fidelity of surrogate models.

3.1 Aerodynamic analysis and prediction of aerodynamic features

Within this task, dedicated to a detailed assessment of machine learning and data-driven techniques for aerodynamic analysis and prediction of aerodynamic features, CIRA focused its attention on the problems relating to the design of aerodynamic configurations with the support of surrogate models based on both supervised and unsupervised machine learning techniques. In this context, a group of CIRA, INRIA, and OPTIMAD was set up to explore and compare the potential of machine learning in improving the computational efficiency of aerodynamic design procedures. The intent was to compare approaches to surrogate modeling with machine learning based on proper orthogonal decomposition (POD), optimal transport theory (OT), and machine learning (ML). The approach stems from the consideration that POD is a powerful tool for storing information in a compressed and efficient way but has substantial limitations in dealing with non-linearities; instead, optimal transportation theory can deal effectively with non-linearities, but finding the optimal transportation map can be difficult and computationally intensive; finally, machine learning approaches (in particular, Neural Networks, have been widely used to build approximator and surrogate models but may lack, in some cases, precision or generalization capabilities. OPTIMAD, INRIA, and CIRA aimed to explore if the combination of POD, OT, and ML might improve the overall efficiency of an aerodynamic shape design process.
INRIA has focused on demonstrating the effectiveness of integrating Machine Learning (ML) techniques with Proper Orthogonal Decomposition (POD) and Optimal Transport (OT) theory to dramatically increase computational efficiency in shape optimization cycles, overcoming traditional limitations inherent in the strong nonlinearities of transonic flows. The need to develop a synergistic approach based on POD+OT+ML arises from analyzing the structural limitations of the individual methodologies taken separately:

1. **Limitations of POD:** Although POD is ideal for reconstructing smooth fields, it has substantial limitations in handling strong nonlinearities. In the presence of moving shock waves in the

Flight condition	**Geometric features**
$\alpha = 0°, 2°, 3°, 4°, 5°, 6°, 7°, 8°$	$b/b_0 = 0.9, 1.0, 1.1, 1.2$
$M = 0.70, 0.74, 0.78, 0.80, 0.82, 0.84, 0.86$	$\Delta\Lambda = -5°, -2°, 0°, +2°, +5°$
	$t/t_0 = 0.9, 1.0, 1.1$

Table 4 – Definition of the different flight conditions and geometric features used to build the database. A total of 56 flight conditions and 60 wing geometries, summing up to 3,360 simulations.

transonic spectrum, linear interpolation of POD modes introduces spurious oscillations and artificial numerical damping, compromising the local accuracy of the solution.

2. **Optimal Transport Challenges:** Optimal Transport theory can capture and move discontinuities (such as shock fronts) in a geometrically consistent manner without generating numerical artifacts. However, numerically determining the optimal transport map (Monge-Kantorovich) is inherently complex, nonlinear, and computationally intensive when solved *ex novo* during the optimization cycle.

3. **Limitations of Pure Machine Learning:** Machine learning approaches are widely used to build approximators and surrogate (black-box) models. Despite their high speed, when trained directly on high-dimensional vectors or without physical/geometric constraints, they can suffer from chronic lack of precision or poor generalization capabilities outside the training dataset.

The union **POD + OT + ML** aims to overcome this state of the art by combining the efficiency of dimensionality reduction, the geometric robustness of optimal transport, and the nonlinear interpolation capability of neural networks.
INRIA methodological proposal reconstructs the transonic fluid field using Optimal Transport. To correctly map the POD modes to the computed solution in the presence of shock waves, a discrete transformation map is defined:

$$\text{Design Parameters} \longrightarrow \text{Discrete Transport Map (Matrix)} \longrightarrow \text{Fluid Dynamics Solution}$$

As can be appreciated from Figure 3, Rather than performing standard convex interpolation ($\hat{\rho}^{\text{co}}$), which produces an artificial double shock or blurred density profiles, the reconstruction using Optimal Transport ($\hat{\rho}$) moves the shock wave continuously to the correct position, preserving the sharpness of the discontinuity front and approximating much better the computed shock wave (ρ).

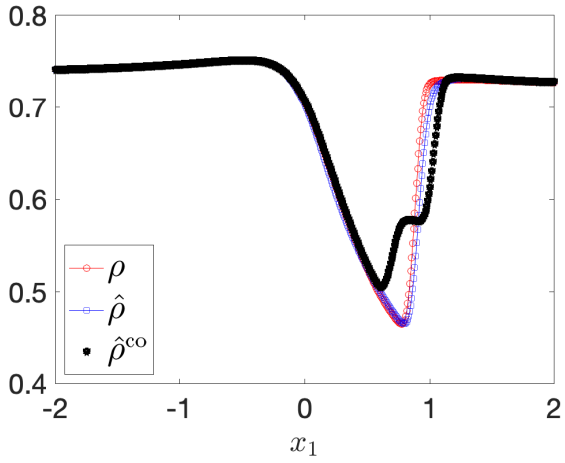


Figure 3 – Optimal transportation reconstruction.

A neural network is trained to learn the non-linear relationship between the **geometric/operational parameters** of the airfoil and the **discrete optimal transport matrix**, thereby serving as an instantaneous predictor of the field map.
The outline of the conceived transonic airfoil design optimization procedure is sketched in Table 5.
CIRA, instead, focused its attention on the study of the potential of POD and Physics-Informed Neural Networks (PINN), analyzing the potential and comparing the performance of a surrogate-based aerodynamic design procedure in the two cases in which the surrogate model it is built using the Gappy POD [5, 6, 7, 8, 9, 10] and a particular neural network, namely the Deep Operator Network (DeepONet) [7], made up of two sub-networks: one for encoding the input function at a fixed number of sensors (branch net), and another for encoding the locations for the output functions (trunk net). Gappy POD is a modification of the standard POD method that handles incomplete data sets used in applications where distributed sensors provide only partial field measurements, while DeepONet

Step	Phase	Description & Technical Details
1	Coarse Sampling	Define the initial parameter space: airfoil geometry, Mach number (M), lift coefficient (C_L), and Reynolds number (Re).
2	POD Analysis	Perform Proper Orthogonal Decomposition (POD) on the flow field snapshots, capturing mesh nodes and flow state variables.
3	ML & Non-Linearity	**Limitation:** Standard POD lacks accuracy near shock waves. **Mitigation:** Apply **Optimal Transportation Theory** to map target airfoils outside the initial basis. *Action:* Compute transport matrices and use them to train an ML approximator (Neural Network / Gaussian Process).
4	Surrogate Model	Establish the functional mapping: DESIGN PARAMETERS $\rightarrow$ TRANSPORT MAP/MATRIX to construct the surrogate.
5	Optimization Loop	Deploy the fast surrogate model within the aerodynamic design optimization cycle.
6	Solution Selection	Select the best candidate solution along with other significant configurations based on error metrics.
7	Dataset Enrichment	Run the high-fidelity CFD flow solver on the selected solutions and append the new snapshots to the database.
8	Iteration / Convergence	Re-run the POD analysis with the expanded database and repeat the loop until convergence or budget exhaustion.

Table 5 – Transonic design optimization procedure.

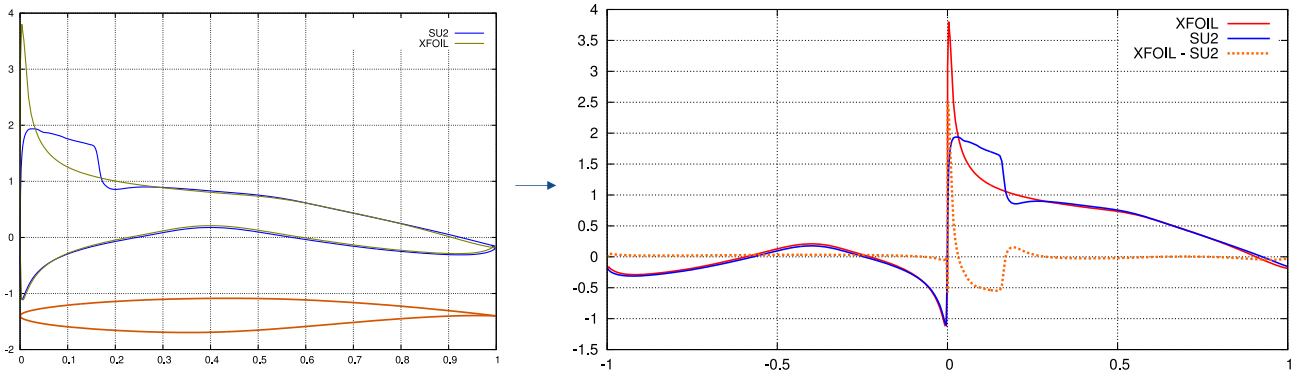


Figure 4 – Hi-fi and lo-fi pressure coefficients.

is an innovative approach capable of implementing effective approximations of non-linear operators. The basic idea is that of embedding physics information through multi-fidelity modeling in the approximators, i.e., inserting information on the physics of the problem using a low-fidelity model and ensuring that the approximator returns the difference between the low-fidelity model and the high-fidelity model (details of the multi-fidelity model are reported in the section related to ML4AERO Task 2). In other words, we exploit the availability of a hierarchy of increasingly complex solvers, namely a full potential with boundary layer correction (XFOIL) and a RANS solver (SU2), and use the low-order model to improve the performance of a surrogate model trained on the high-order model. The physical information that is used, in particular, is the pressure coefficient. Both approximators, namely GappyPOD and DeepONet, are trained with the pairs of low- and high-fidelity calculated pressure coefficient distributions and generate, after the training phase, the high-fidelity pressure distribution with the relative global coefficients, receiving as input the distribution of low-fidelity pressure coefficients and the relative global coefficients.
The procedure was tested on the optimization problem shown in 1:

$$\begin{aligned} &\min c_d \\ &\text{s.to}: \quad c_l \geq c_{l_0} \qquad\qquad c_{l_0} = 0.724 \\ &\qquad \text{maxthickness} = 12.11\% \\ &\qquad \text{Mach} = 0.65 \qquad\qquad \text{Reynolds} = 6500000 \end{aligned} \tag{1}$$

The baseline airfoil is the RAE 2822. The Initial database setup was filled with 100 elements calculated in high and low fidelity. An incremental strategy was then adopted for eventual subsequent iterations of the procedure, based on the high-fidelity recalculation of the envelope of the best individuals in the convergence history. The CMA-ES evolutionary optimization procedure [11] was used in all the design optimization runs.
Figures 5 and 6 show the convergence history and the envelope of the best solutions from the two runs carried out with GappyPOD and DeepONet, respectively. Note how the graphs of the solutions chosen for the recalculation highlight the quality of the approximator in terms of accuracy and outliers.

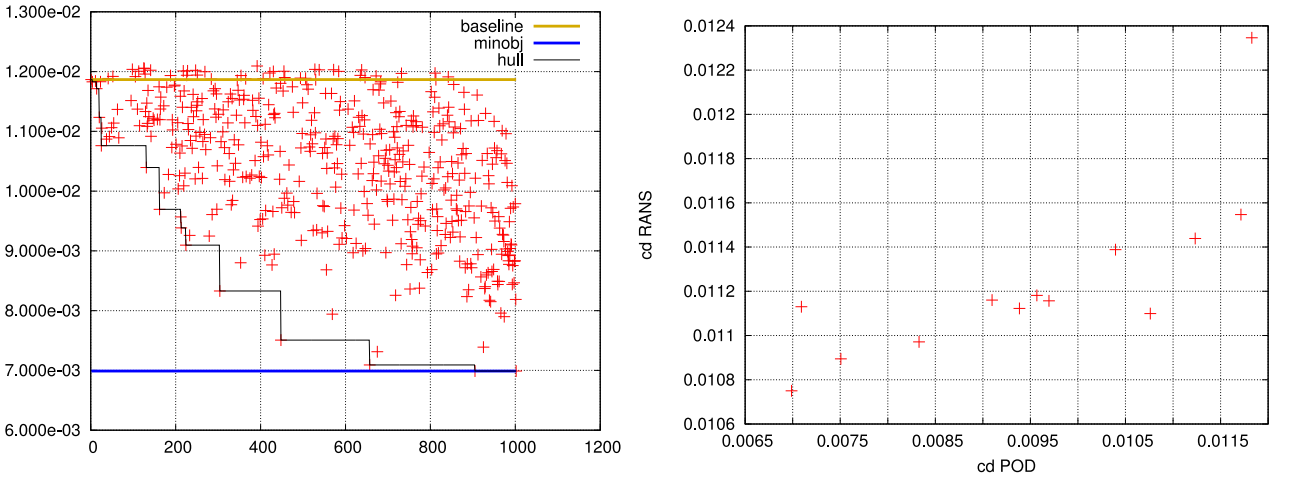


Figure 5 – GappyPOD-based optimization run.

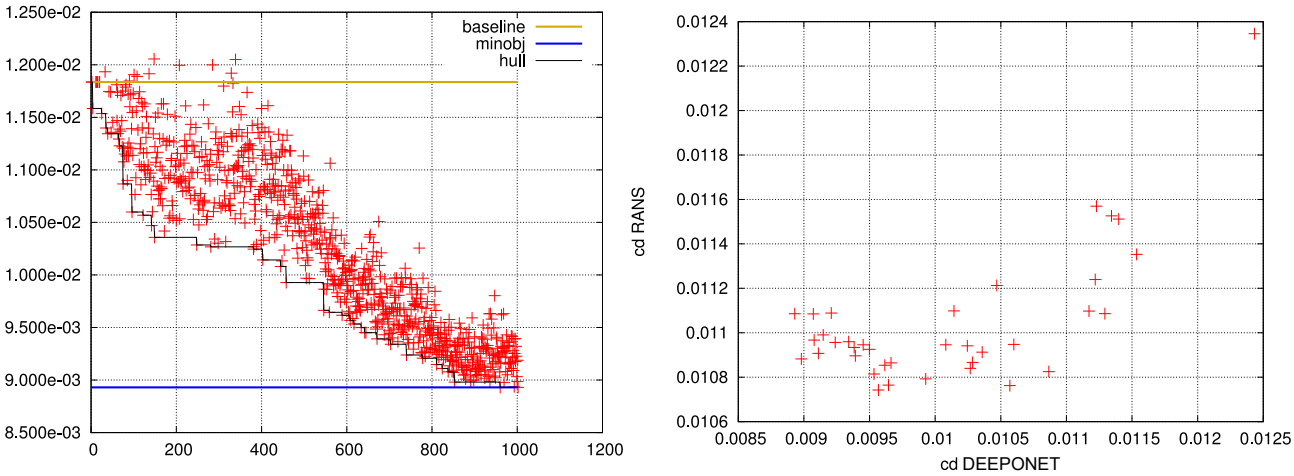


Figure 6 – DeepONet-based optimization run.

3.2 2.2 Data fusion and multi-fidelity heterogeneous data management

Within this task, CIRA focused on the consideration that developing machine learning applications very often requires huge amounts of data. Contrary to what happens to applications for facial and speech recognition, which can, mainly thanks to the web, rely on gigantic and heterogeneous databases, most engineering applications, such as analysis and aerodynamic design, can rely on much fewer data and, above all, enormously more expensive to generate and manage. With this in mind, CIRA introduced an almost entirely automated tool to allow the RANS analysis of simple airfoils. This tool allows the generation of a large amount of good-quality aerodynamic data that

can be used in training neural networks or ML applications that require a significant amount of data for their setup. This tool is not an alternative but complementary and preparatory to the data and computations relating to XRF-1 database. It allows the parametric wing section shape handling, the mesh generation, and the RANS flow field equation solution to be fully automated. The framework comprises a shape handler block, a grid generator, and the numerical flow solver. The wing section shape is handled using the in-house developed tool wg2aer. Once the baseline geometry is modified, the computational mesh is generated using an automated procedure based on gmsh software tool [12] software. Finally, the aerodynamic characteristics can be evaluated by choosing between two different flow solvers: the Stanford SU2 RANS solver [13, 14] for high-fidelity and MIT Drela's XFOIL as a low-fidelity solver [15].

3.3 Best practices guidelines

As regards best practices, the considerations linked to the work carried out can be summarized in two points:

1. The application to aerodynamic shape design has highlighted how vital it is to find an appropriate balance between the need for machine learning approaches to have an extensive database available for training and the need for design procedures to use the least possible number of high-fidelity evaluations. The path that needs to be explored and, consequently, the best practice to follow is that of using adaptive approaches (in-line training) for constructing and improving surrogate models based on machine learning. The potential of architectures, such as PINNs, to include information on the physical model of the problem to reduce the amount of data needed for training drastically must also be explored. Furthermore, using low-fidelity solvers to train a neural network on the difference in response fidelity is an avenue worth exploring.

2. The use of heterogeneous data sources is undoubtedly one of the possible ways to allow machine-learning-based procedures to work on small and heterogeneous data sets. However, tuning these neural networks can be critical, and it is therefore essential to have rapid tools for calculating aerodynamic solutions at different levels of fidelity to explore these approaches comprehensively.

4. DLR

In the context of the GARTEUR group DLR focused on the prediction of surface pressure distributions for 3D configurations. This task is highly relevant for industrial applications since accurately predicting the pressure coefficients is essential for the determination of aerodynamic loads, transition prediction and inverse design. Besides long-standing conventional models, especially deep-learning approaches have gathered significant interest in the aerodynamics community [16]. In the following results will be compared between three deep learning methods, a Multi-layer perceptron for distribution predictions, a Multi-layer perceptron for pointwise predictions and an Autoencoder coupled with an interpolation technique to Proper Orthogonal Decomposition and Isomap with latent space interpolation. The test case was based on the XRF1 half-body configuration provided by Airbus and available within the GARTEUR group. Further DLR investigated Graph Neural Networks for the Prediction of Aircraft Surface Pressure Distributions and Physics-based Regularization of Neural Networks for Aerodynamic Flow Predictions. For further details on these results interested readers are referred to the publications [17] and [18], respectively.

4.1 Data-driven methods for aerodynamic flow predictions

Various methods have been proposed for data-driven modeling of aerodynamic flows. Within the GARTEUR group DLR compared the following deep- and non-deep-learning based approaches.

- Proper Orthogonal Decomposition plus latent space interpolation
- ISOMAP plus latent space interpolation
- Autoencoder plus latent space interpolation

Metric	POD+I	Isomap+I	AE+I	MLP-Dec.	MLP-Pointwise
MAE	0.0293	0.0324	0.0382	0.0362	0.0309
AE q.95	0.1256	0.1424	0.1338	0.1250	0.1163
AE q.99	0.3242	0.3160	0.3047	0.2822	0.2780
Max. AE	1.1824	1.0187	0.9988	1.0603	1.0815
RMSE	0.0748	0.0723	0.0734	0.0706	0.0638
r^2-score	0.9659	0.9681	0.9671	0.9696	0.9752

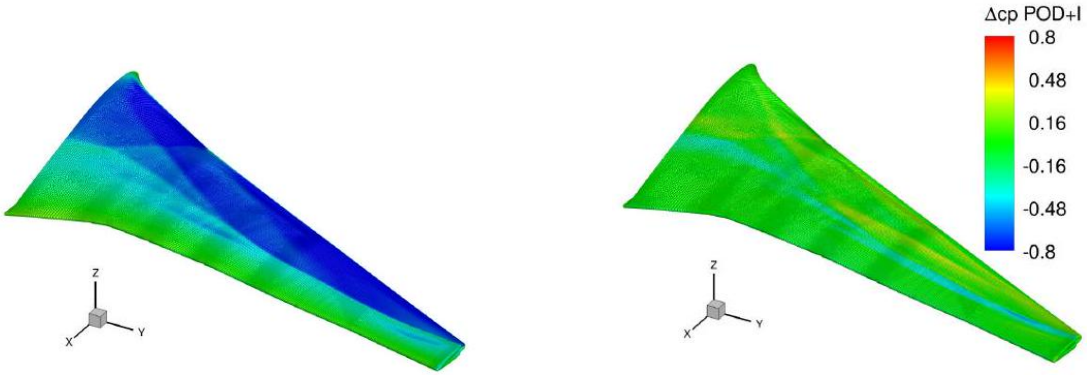
Table 6 – C_p test metrics

- MLP decoder
- MLP yielding pointwise predictions

An in-depth introduction of methodologies is beyond scope for this overview paper and interested readers and instead referred to the available literature [19], [17] and [18].
For all models hyperparameters need to be selected and they have a crucial influence on the model performance. Hence, DLR employed two methodologies for efficient hyperparameter determination. The two approaches used to find the best hyperparameters are a Full Factorial Grid Search for POD $+I$, Isomap $+I$, and $AE+I$ models and a Surrogate-Based Optimization for the hyperparameters of the NN models.
The reduced order modeling methods are part of the DLR Surrogate Modeling for AeRo data Toolbox in Python (SMARTy) [20]. SMARTy is a toolbox for predictive modeling of aerodynamic scalar or high-dimensional quantities using state-of-the-art data-driven methods. It allows, in a very user-friendly way, the construction of reduced order models from CFD data in a unified layer that includes several preprocessing and postprocessing options, training of neural networks with PyTorch [21] and TensorFlow [22] as backends, and surrogate-based hyperparameter optimization. In addition, it features a model selection module that provides grid search functionality, automatically building and training various models, and selecting the best one. For the all deep learning models presented herein, TensorFlow is used as backend.

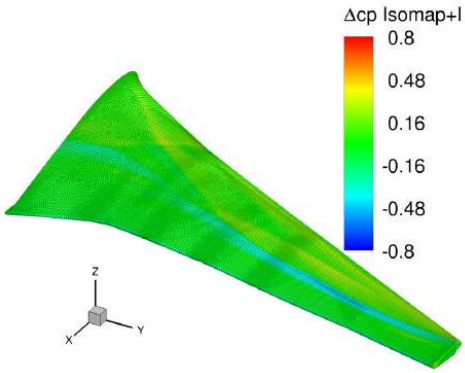
4.2 Test case and results

The test case configuration considered here is an industrially relevant aircraft configuration known as XRF1. The XRF1 is an Airbus provided industrial standard multi-disciplinary research test case representing a typical configuration for a long range wide body aircraft. It is used by Airbus to engage with external partners on the development and demonstration of relevant capabilities and technologies. Details on the dataset are given in 2.. A total of $m = 89$ pressure coefficient (c_p) distributions were computed for four different Mach numbers, 0.82, 0.86, 0.90, and 0.96, and angles of attack in the range of $[-7.6°, 8.1°]$. The Reynolds number was fixed to $Re = 25 \cdot 10^6$. Out of the $m = 89$ total samples, 6 are selected for testing, 17, approximately twenty percent, are selected for validation, while the remaining 66 samples are used for training.
Next, results obtained predicting pressure coefficient distributions at various combinations of Mach number and angle of attack for the test case presented in the previous section are discussed. In a production environment, it is key that the models perform well at new unseen parameter combinations. Accordingly, the performance of the models is assessed using the 6 test samples, which were used neither for training nor validation or hyperparameter selection. The predictive capabilities are evaluated quantitatively using the mean absolute error (MAE), the 95% and 99% quantiles of the absolute error (AE $q.95$, AE $q.99$ resp.), the maximum absolute error (Max. AE), the root mean squared error (RMSE), and the r^2-score.
POD $+$ I achieves the lowest MAE among all models, while MLP-Pointwise yields the best performance in terms of all other error metrics. MLP-Pointwise outperforms POD $+$ I by 7.6% and 14.25% in terms of the 95% and 99% error quantiles respectively. This suggests that MLP-Pointwise performs better at capturing localized phenomena such as strong shocks or flow separation. A possible

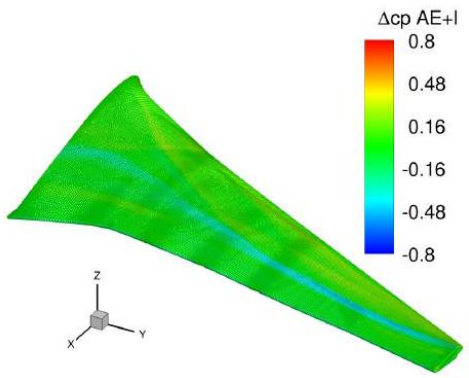


(a) Reference Solution

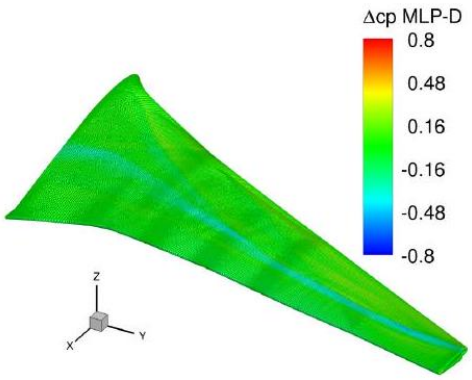
(b) POD + I



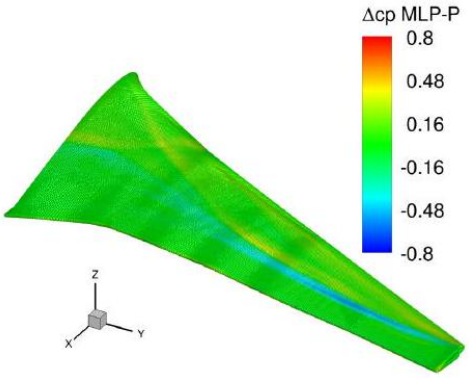
(c) Isomap +I



(d) AE + I



(e) MLP-Decoder



(f) MLP-Pointwise

Figure 7 – Comparison between the prediction accuracy for M: 0.82 and AoA: 3.105.

explanation is that the mini-batch stochastic optimization algorithm with a mean squared error loss helps the deep learning methods by forcing them to learn local behaviour and increasing their generalization performance. Moreover, the use of point coordinates provides MLP-Pointwise with additional information that can turn out to be useful for learning such local phenomena by implicitly gathering information on local phenomena and connectivity. All test metrics with respect to the six test samples are shown in Table 6.

Figure 7 shows the CFD reference result together with all model predictions at a Mach number of 0.82 and an angle of attack of 3.105. In this case, MLP-Decoder attains the best performance with respect to all metrics. It outperforms POD + I by 26.85% and 8.03% with respect to the 95% and 99% error quantiles, respectively. However, for other cases, especially even more challenging aerodynamic conditions, the MLP-Pointwise model has shown superior performance.

4.2.1 Computational cost

For practical applications, it is not only important that the data-driven models provide accurate predictions of aerodynamic quantities, but also that they do so in significantly lower time than the time-consuming high-fidelity CFD computations. MLP-Pointwise was trained using an NVIDIA Quadro P4000 GPU, whereas all other models were trained exclusively on an Intel(R) Xeon(R) W-2135 CPU @ 3.70GHz with 12 cores. POD + I and Isomap + I require the least time to train, 9 and 11 seconds, respectively. For POD + I, most of the computation time is required for the regularized TPS interpolation, while for Isomap + I the dimensionality reduction is the most time-consuming step. The training of AE + I and MLP-Decoder is in the order of minutes, each taking 126 and 166 seconds. MLP-Pointwise has a training time in the order of hours, taking 5839 seconds to complete 250 epochs. Regarding the evaluation time, for all models it is significantly under one second and is thus negligible with respect to the CFD simulation time. Table 7 shows for each model the approximate time per epoch, the training time, and the time it takes to perform an evaluation of one sample to yield a prediction of the complete pressure coefficient distribution.

Phase	POD+I	Isomap+I	AE+I	MLP-Dec.	MLP-Pointwise
One Epoch	–	–	0.74	0.42	23.84
Training	9	11	126	166	5839
One Evaluation	0.0024	0.0028	0.0126	0.1901	0.2184

Table 7 – Approximate computational cost in seconds

4.2.2 Conclusions

In the context of the GARTEUR group, DLR focused on the prediction of surface pressure distributions for 3D configurations leading to three publications, in which DLR explored the use of different deep learning methods for the prediction of surface pressure distributions for both wings and full aircraft configurations. These models provide fast approximations and can be seen as a surrogate for time-consuming RANS-CFD simulations. Further details are available in [19], [17] and [18].

5. FOI

The objective is to investigate the performance of methods from machine learning (ML), algorithms from statistical learning and artificial intelligence, to supplement the results from heavy computer simulations by producing models which can produce estimations at points where simulations or experiments have not been conducted, so-called missing point estimations [16].

Here we evaluate the capacity of ML in building models of the map between free-stream data (Mach number and angle-of-attack), or free-stream data together with geometrical parameters, and the pressure field on a wing typical of a long range passenger aircraft.

The benefits of such a surrogate model are cost reductions of a design process: replacing computer intensive simulations in Monte Carlo type of process or providing better estimates than computer simulations in case experimental data are available.

The challenge [23] is to produce accurate models when flow features are sensitive to small changes of the input parameters, which is the case of shocks in transonic flight conditions; their location and intensity strongly varying for any change in free-stream conditions or in shape.

In the context of black-box data-driven modelling (Which does not require any insight about the process which produces the data, here CFD or experiments), the cost is the number of training data which is required to achieve a targeted accuracy.

In the AG60 project the focus is on steady aerodynamic flows obtained either from CFD or from wind-tunnel tests around the Airbus XRF1 geometry, for Mach numbers ranging from 0.5 up to 0.96 which covers subsonic up to transonic flow conditions.

The approach that we adopt involves a dimension reduction of the pressure coefficient distribution as a first stage; the most common algorithm to perform this task is Proper Orthogonal Decomposition (POD) [5, 23, 24].

The dimension reduction provides a latent space representation of the pressure coefficients, and a regression algorithm maps the latent space data from the input data, which are the free-stream conditions and eventual geometrical parameters.
In this section, we evaluate alternative schemes to POD for the dimension reduction stage: Auto-Encoder neural networks [25] and Kernel-POD [26]. Because of the infinite variety of designs of neural networks, tests were conducted to identify the potential impact of Auto-Encoder topology, with both linear and non-linear activation functions.

5.1 Methodology

Databases A preliminary study is conducted on five databases of airfoils:

- Varying airfoil geometry at constant free-stream conditions: two flow regimes (subsonic and transonic) and two parameterizations (FFD [27] of the RAE2822 airfoil with 3dof and 6dof).
- Varying free-stream conditions (Mach number and angle of attack) around a fixed airfoil geometry (RAE2822).

The tests on the XRF1 concern Database 2 (DB2) with varying Mach number, angle-of-attack, sweep-angle, global thickness and span-width: our first test concern the entire span of those input parameters in the database DB2, in particular the Mach numbers are in the interval $[0.5, 0.86]$ and the angle-of-attack in the interval $[-4, 8]$ degrees, and in a second series of tests we restrict the span of Mach numbers to $[0.7, 0.86]$ and the span of angle-of-attack to $[2, 8]$ degrees.

Models description

Feature data modelling: dimension reduction As described above the approach to build the models uses a dimension reduction of the pressure coefficients $(y_i = C_p^i)$, as illustrated in Figure 8 for the POD, followed by an interpolation or regression of the latent data with respect to the input of the model which can be the flight conditions, the design or a combination of both.

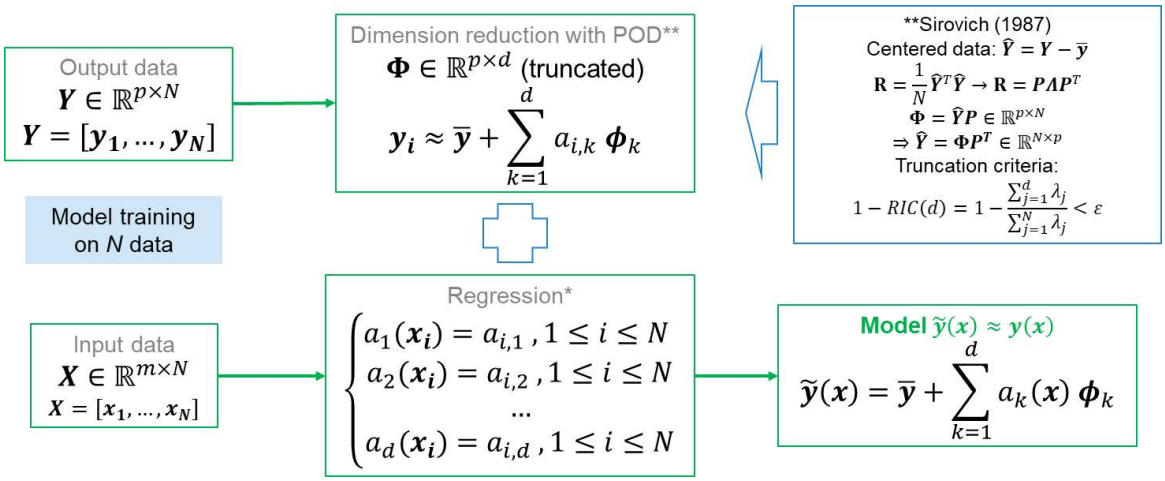


Figure 8 – The workflow to produce models using POD as a dimension reduction algorithm.

The choice of Auto-Encoder (AE) is natural because it is known [28] that in its simplest form, with one hidden layer and linear activation functions, and with a latent space dimension equal to the number of components retained for the POD (d), it achieves the same mean-square error (MSE) as the POD on representing new data and the columns of its matrix of weights (W in Figure 9) spans the same space than the retained components of the POD.
Based on this observation, we investigate the effect of having AE with several hidden layers and AE with non-linear activation functions. However, in order to restrict the number of combinations (infinite) of possible networks we adopt a parameterization of the hidden layers topology inspired by another

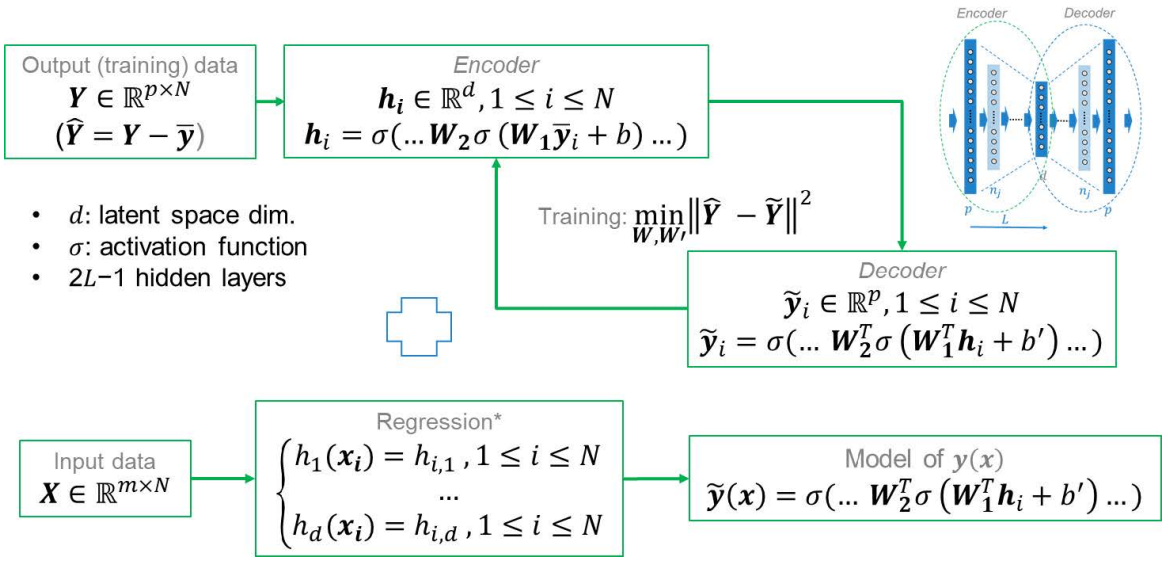


Figure 9 – The workflow to produce models using Auto-Encoder neural networks as dimension reduction algorithm.

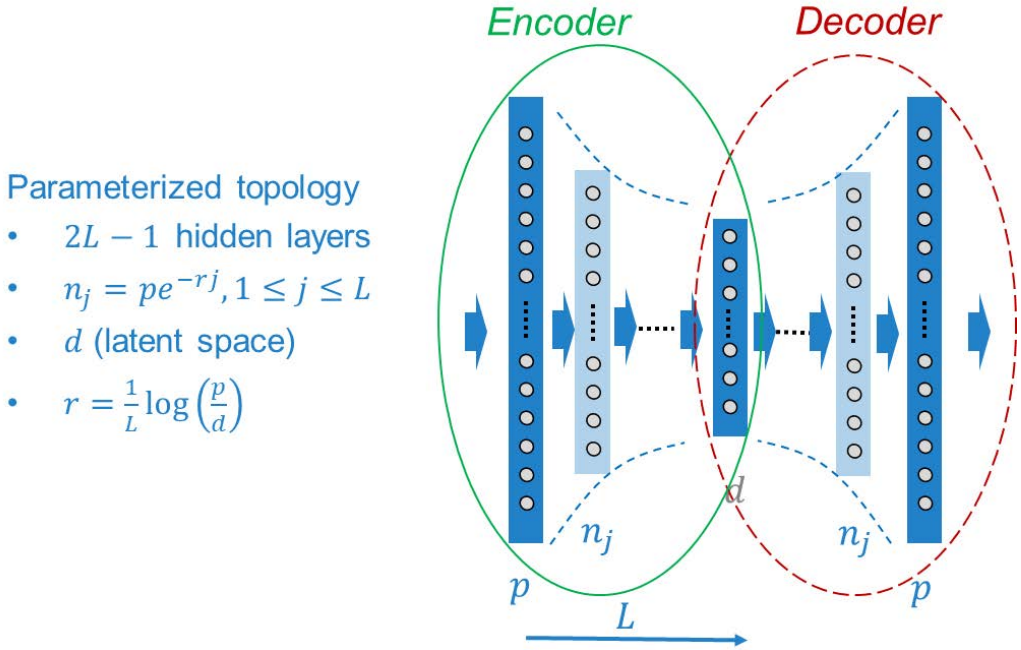


Figure 10 – Auto-Encoder neural network: we investigate symmetric networks with hidden layers which sizes (n_j) are given by a function which depends on the position (j) of the layer, the half-width (L) of the network and the latent space dimension (d).

study [29], which is described in Figure 10. Also, the results with non-linear AE presented concern the $\tanh$ activation function.

Concerning the Kernel-POD we refer to Haykin [30] for a description of the kernel trick. The process is similar to using the POD or the AE: after a transformation of the output data (pressure coefficient) towards a higher-dimensional space (the kernel trick) the Kernel-POD algorithms uses SVD, as it is done in POD, to identify principal components; the interpolation or regression can then take place to map the input data to the latent space variables in order to obtain a complete model relating the pressure distribution to the input data.

Hyperparameters for the KPOD implementation in Scikit learn [31]: RBF kernel, $\gamma = 5.0 \times 10^{-6}$ (shape factor), $\alpha = 10^{-12}$ (inversion).

Latent space data modelling Here we use the Kernel Ridge Regression (KRR) implemented in Scikit learn [32] with grid search for the selection of the hyperparameters γ (shape factor), α (regularization), using a 5-fold cross-validation and `neg_mean_absolute_error` for scoring [33].

Method evaluation

Achieving consistency of a method of approximation is an objective that is not guaranteed: the prospect is to obtain a method which produces models which accuracy increases for an increasing number of training data, in a statistical sense. The motivation is that the cost being proportional to the amount of training data, we would be guaranteed that adding new training data would improve the predictive capability of the new model, may it be in a statistical sense only.
We therefore present our results in terms of statistical distributions represented in the graphics by boxplot and, regarding the databases of AG60, in terms of tables with the correlation factor (R^2), explained variance (EVS), mean square error (MSE) and maximum absolute error (ME).

5.2 Results

Preliminary investigations

The results reported on the 2D databases may guide further investigations regarding the impact on the data-driven modelling process:

- Free-stream data as input variables.
- Geometrical data, such as airfoil shape, as input variables.
- Kernel-POD as an alternative to POD.
- Auto-Encoder, linear or non-linear, as an alternative to POD.
- Topology of the hidden layers in Auto-Encoder neural networks.
- Impact of the mapping from input variables to the latent space variables.

In Figure 11 we summarize results obtained with both POD and Kernel-POD at a Mach number where compressibility should not have a large impact (Mach 0.5) and at a Mach number in the transonic region (Mach 0.8). Showing the RMSE instead of boxplot makes the results more readable as we compare four different models in each graph.
From Figure 11, we observe similar trends and levels of accuracy for the parameterization with 3 dof compared to the parameterization with 6 dof when a large amount of training data is used, above 60 to 100. Also, the differences between the models are small when using POD compared to Kernel-POD for dimension reduction.
In Figure 12 we compare models using POD to models using linear Auto-Encoder (left) or non-linear Auto-Encoder (right). The results of RMSE for latent space dimensions (d) from 15 and above overlap with each other in the case of linear Auto-Encoders.

XRF1 applications

DB2 (4320 CFD simulations) The first series of test concerned the part of the database with 4320 CFD results for varying Mach number, angle-of-attack, variation of sweep-angle, variation of global thickness and variation of span-width, with the Mach values in $[0.5, 0.7, 0.74, 0.78, 0.8, 0.82, 0.84, 0.86]$, and the angle-of-attack in $[-4, 0, 2, 3, 4, 5, 6, 7, 8]$ degrees. Models were built with POD, Kernel-POD, linear Auto-Encoder networks and non-linear Auto-Encoder networks.
We start by a comparison of POD and Kernel-POD on two versions of this database:

- with only the Mach number and the angle-of-attack changing, meaning without varying geometrical parameters,
- with the entire database of 4320 CFD simulations.

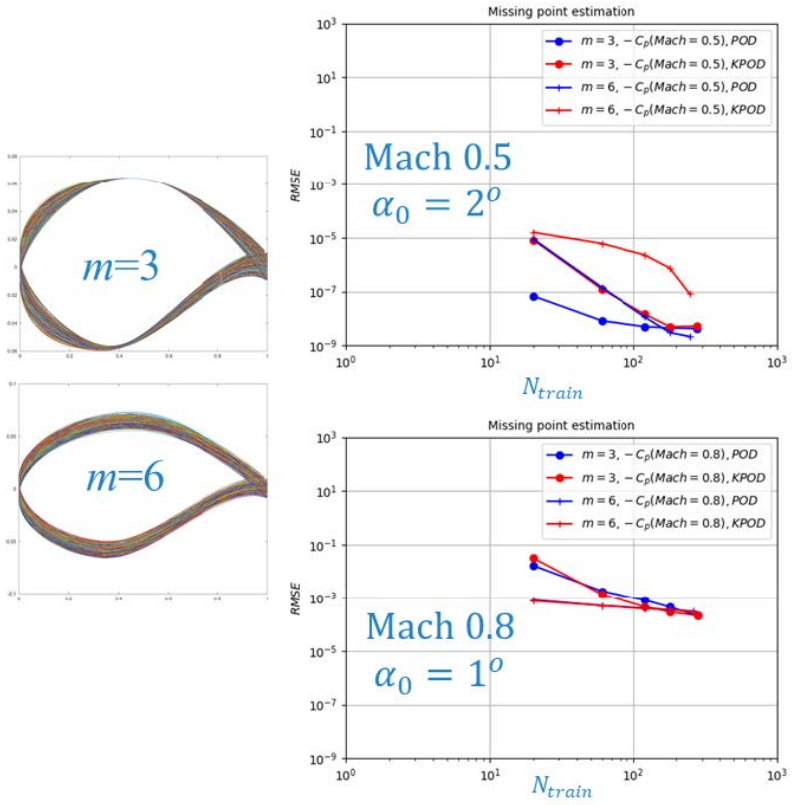


Figure 11 – RAE2822 test cases with fixed free-stream and variable airfoil design as input variables: error statistics in terms of RMSE of models of the pressure coefficients built by POD with KRR regression (POD) or with Kernel-POD with KRR regression (KPOD) for a varying number (N_{train}) of training data, for two different FFD parameterizations, with 3 dof ($m = 3$) or with 6 dof ($m = 6$).

Those results, summarized in Figure 13, confirm the trend observed in the preliminary study on 2D CFD databases that POD and Kernel-POD supplemented with Kernel Ridge Regression (KRR) show very close results. This is why in the remaining of this section we only comparing POD with Auto-Encoder neural networks.

DB2 (2940 CFD simulations) The second series of test regarding DB2 concerned the part of the database with 2940 CFD results for varying Mach number, angle-of-attack, variation of sweep-angle, variation of global thickness and variation of span-width, with the Mach number values in $[0.7, 0.74, 0.78, 0.8, 0.82, 0.84, 0.86]$, and the angle-of-attack in $[2, 3, 4, 5, 6, 7, 8]$ degrees.
Models were built here as well with POD, Kernel-POD, linear Auto-Encoder networks and non-linear Auto-Encoder networks. We give only a few results comparing the performances of Auto-Encoders and POD for the sake of conciseness. Similar observations can be done regarding the contribution to the modelling error levels due to the dimension reduction algorithms and due to the regression used to map the variations of the input parameters to the latent space variables, see Figure 14 and Figure 15.

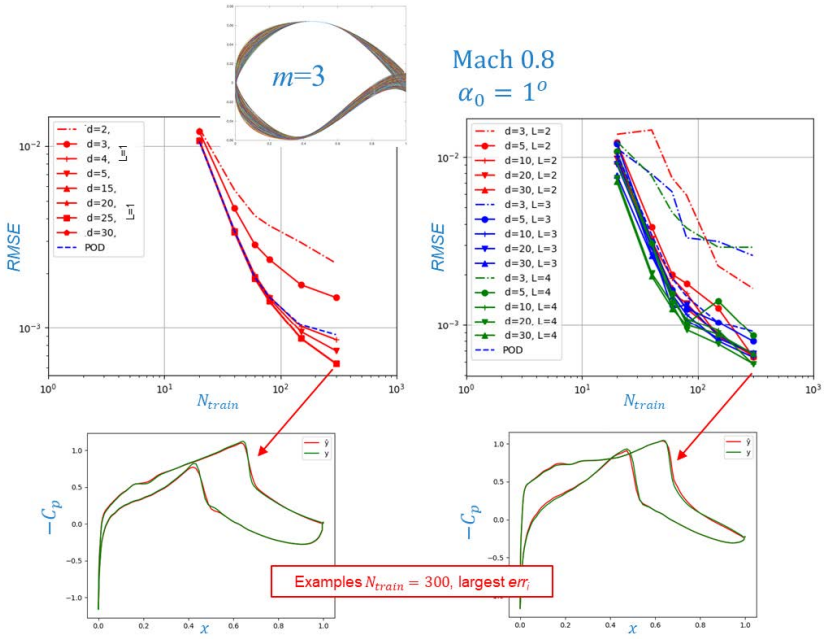


Figure 12 – RAE2822 test case at Mach 0.8 with 3 FFD parameters as input variables: the plots at the top represent the errors of each model with respect to the training cost (N_{train}). On the left POD is compared to 8 different types of linear Auto-Encoders, each having a different latent space dimension (d) and a single hidden layer. On the right POD is compared to 15 different non-linear Auto-Encoders, each having a different combination of latent space dimension (d) and hidden layer parameter (L). At the bottom worst cases predictions for each of the best model: linear Auto-Encoder on the left, non-linear Auto-Encoder on the right, where symbols denote the model's prediction and the reference (CFD) data.

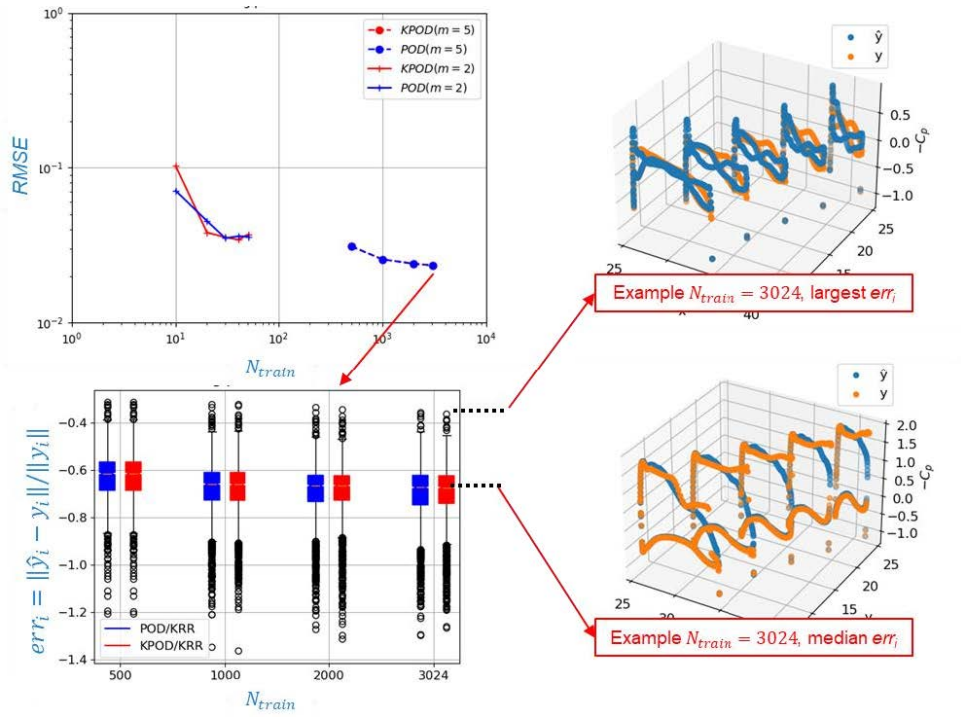


Figure 13 – DB2 modelling with POD and Kernel-POD (KPOD), both type of models using Kernel Ridge Regression (KRR) to map the input parameters variations to the latent space variables: both methods were applied on a restricted part of the database with only Mach and AoA varying ($m = 2$) and to a larger subset of DB2 with Mach, AoA, sweep angle, span-width and thickness varying as input parameters to the CFD.

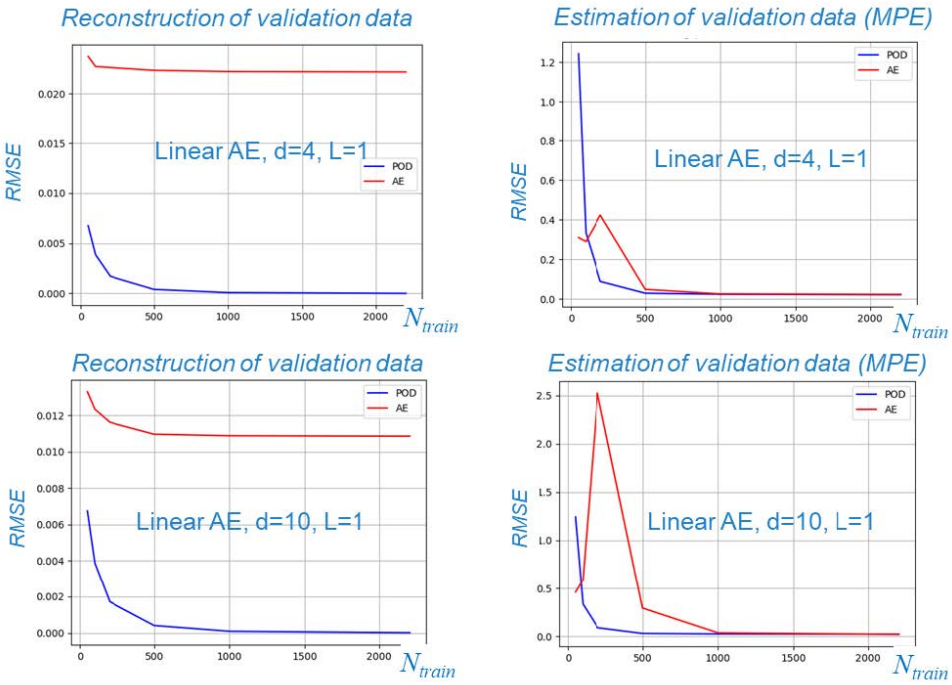


Figure 14 – Modelling of DB2 data (2940) with linear Auto-Encoder networks - Error analysis: on the left side the impact of the number of training data on the RMSE reconstruction error is the lowest achievable RMSE error that each model (POD or linear AE) can achieve and, on the right, the RMSE for missing point estimation, which are the errors made by the trained models consisting of dimension reduction and regression.

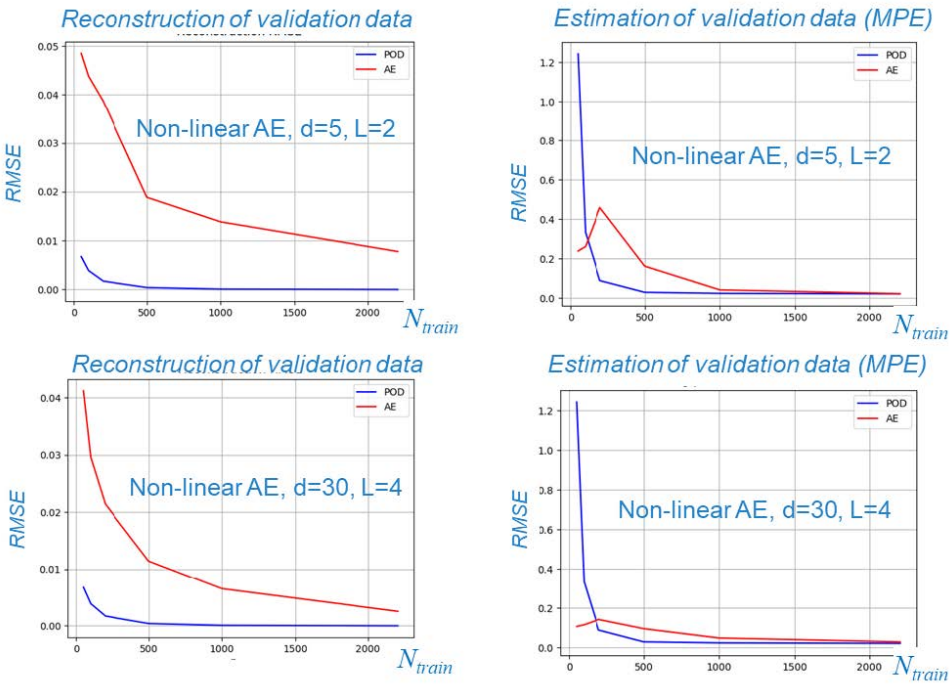


Figure 15 – Modelling of DB2 data (2940) with non-linear Auto-Encoder networks - Error analysis: on the left side the impact of the number of training data on the RMSE reconstruction error is the lowest achievable RMSE error that each model (POD or non linear AE) can achieve and, on the right, the RMSE for missing point estimation, which are the errors made by the trained models consisting of dimension reduction and regression.

6. INTA and UC3M

Within the GARTEUR AG60 research programme, INTA, with the support of Universidad Carlos III de Madrid (UC3M), carried out a series of studies focused on the application of machine learning, reduced-order modelling, and uncertainty quantification techniques to aerodynamic prediction and design. Their work aimed to reduce the computational burden associated with high-fidelity CFD simulations while maintaining sufficient accuracy for engineering applications. A central theme of their research was the exploration of nonlinear manifold-learning techniques, particularly Isometric Feature Mapping (Isomap), as an alternative to traditional reduced-order modelling approaches.

The first major contribution was the development and assessment of surrogate models capable of predicting aerodynamic pressure distributions on aircraft wings operating in the transonic regime. INTA and UC3M investigated whether machine learning could accurately reproduce CFD results for flight conditions that had not been previously simulated. To achieve this, they compared traditional Proper Orthogonal Decomposition (POD), a widely used linear reduced-order modelling technique, with Isomap, a nonlinear manifold-learning method capable of identifying hidden low-dimensional structures in complex aerodynamic datasets. These reduced-order representations were coupled with two machine learning regressors: Random Forests (RF) and Deep Neural Networks (DNN). Using a database of more than 500 CFD simulations of the Airbus XRF1 wing covering a broad flight envelope, they demonstrated that Isomap-based approaches could achieve prediction accuracies comparable to or better than POD while reducing the dimensionality of the problem much more aggressively. The resulting surrogate models were capable of predicting complete surface pressure coefficient distributions using only Mach number and angle of attack as inputs, enabling rapid aerodynamic evaluations that would otherwise require expensive CFD calculations.

A significant aspect of this work was the development of a complete encoding–regression–decoding framework. The aerodynamic pressure fields were first projected onto a low-dimensional latent space using either POD or Isomap. Machine learning models were then trained to predict the latent coordinates associated with a given flight condition. Finally, the original pressure distribution was reconstructed through dedicated decoding procedures. For Isomap, INTA and UC3M proposed a novel back-mapping strategy based on weighted k-nearest-neighbour interpolation, allowing accurate reconstruction of high-dimensional pressure fields from only a handful of latent variables. Their results showed that Isomap could compress aerodynamic information into as few as two to four coordinates while preserving the essential flow physics. Furthermore, the Isomap-based Random Forest model achieved excellent predictive performance with a dramatically smaller model size than direct interpolation approaches, highlighting the potential of manifold learning for practical engineering applications.

The second major contribution extended these ideas towards aerodynamic shape optimisation. In this study, INTA and UC3M developed a data-driven surrogate modelling framework intended for preliminary aircraft design, where engineers must evaluate many design alternatives with limited computational resources. Unlike the previous work, which focused only on flight conditions, this study incorporated both aerodynamic and geometric design variables, including wing sweep, thickness ratio, and span ratio. By applying Isomap to a database of pressure distributions generated from wings with varying geometries, the researchers identified a low-dimensional manifold that efficiently represented the aerodynamic behaviour of the entire design space. They then trained a deep neural network to map directly from the design parameters to the manifold coordinates. A reconstruction procedure based on nearest-neighbour techniques was subsequently used to recover the full pressure distributions. This Isomap+DNN framework enabled rapid prediction of aerodynamic responses across a multidimensional design space while avoiding the need to run CFD simulations for every new design configuration. The work demonstrated that manifold learning could play a key role in accelerating early-stage aerodynamic optimisation and design exploration.

The third major contribution addressed the challenge of uncertainty quantification and multi-fidelity modelling. Recognising that aerodynamic design often relies on information coming from sources of varying fidelity and accuracy, INTA and UC3M combined Bayesian Gappy Proper Orthogonal Decomposition (BGPOD) with Multi-Fidelity Gaussian Process Regression (MFGPR) to create a more robust surrogate modelling framework. In this approach, high-dimensional pressure distributions were re-

constructed from sparse experimental measurements using BGPOD, while MFGPR was used to fuse low-fidelity CFD information with sparse high-fidelity wind-tunnel data. This allowed the surrogate model not only to improve prediction accuracy but also to quantify uncertainty explicitly. The resulting framework provided estimates of both the mean prediction and its associated variance, making it possible to assess confidence levels in the predicted aerodynamic loads. The combined methodology achieved approximately an 11 reduction in prediction error compared with the use of BGPOD alone and enabled predictions at arbitrary spatial locations rather than only at CFD grid points. This work represented an important step towards integrating uncertainty-aware machine learning methods into aerodynamic design workflows.
The results Table 8 summarizes the validation performance of the surrogate models developed by INTA and UC3M for the prediction of pressure coefficient distributions over a transonic wing. The table compares different combinations of reduced-order modelling techniques (Isomap and POD) and machine learning regressors (Random Forest and Deep Neural Networks), as well as direct regression approaches without dimensionality reduction. The evaluation considers both predictive accuracy, through the coefficient of determination (R2) and mean squared error (MSE), and computational efficiency, represented by the resulting model size. These results provide a quantitative assessment of the trade-offs between accuracy, model complexity, and data compression, highlighting the effectiveness of nonlinear manifold-learning approaches such as Isomap for constructing compact and accurate aerodynamic surrogate models.

	Isomap+RF	POD+RF	RF	Isomap+DNN	POD+DNN	DNN
Model Size	360 MB	390 MB	11.4 GB	40 MB	45 MB	485 MB
Mean R^2	0.958	0.888	0.918	0.924	0.862	0.960
Mean MSE*	0.800	2.439	1.626	0.953	2.387	0.798
Mean R^2	0.720	2.209	0.924	0.847	1.737	0.756
Mean MSE*	0.965	0.898	1.550	0.959	0.908	0.962
Median R^2	0.976	0.921	0.932	–	–	–
Median MSE*	0.392	1.597	1.100	–	–	–

Table 8 – Performance metrics validation set (Mean and Median values)

The joint work of INTA and UC3M demonstrated how advanced machine learning and reduced-order modelling techniques can significantly reduce the computational cost of aerodynamic analysis while maintaining high levels of accuracy. Their research established Isomap as a powerful alternative to conventional POD-based approaches for aerodynamic data compression and prediction, showed how manifold learning can support rapid aerodynamic shape optimisation, and introduced multi-fidelity probabilistic frameworks capable of incorporating experimental uncertainty into aerodynamic predictions. Collectively, these contributions provide a foundation for next-generation data-driven aerodynamic design tools that are faster, more flexible, and better suited to handling the complexity and uncertainty inherent in modern aircraft development.

6. ONERA

During an aircraft program, a very large amount of aerodynamic data is generated in particular to evaluate the coefficients of forces and moments all over the flight envelope. High-Fidelity (HF) Computational Fluid Dynamics (CFD) codes provide a major contribution to gathering these data but as HF CFD and experiments are expensive, low fidelity CFD and Reduced Order Modeling (ROM) of HF data are both of major interest. In the ML4AERO AG60 GARTEUR group, the latter type of techniques has been extensively tested based on numerical data provided by Airbus-M.

6.1 Provided Data

ONERA's regression activity, as part of task 1 of the project, was based on large sets of TAU calculations provided by Airbus-M. Various wing-body calculations were performed by Airbus-M with or

without horizontal tail plane (HTP), vertical tail plane (VTP), sting on a XRF1 model in a wind tunnel (XRF1 configuration is close to a long range wide body Airbus aircraft). Whatever the detailed geometric configuration, the final provided data are C_p coefficients on the wing of the aircraft.
The three sets of calculations have been divided by INTA in four couples of (learning, testing) data in order to define as many regression exercises. The data, in their initial form, appear as a list of C_p values as a function of the coordinates of the wall points and the aerodynamic conditions (M_∞, AoA, Re).
Nevertheless, there is an underlying modal ordering in all series: the number of points is the same for all flow conditions (but the wall points are not the same when varying the Reynolds number). The splitting in training and testing data is also consistent with the flow conditions, classifying all C_p values of one flow condition either in the training or in the testing set. The four sets of (training/testing) data are presented in Table 9.

Dataset	Description
SST_2p5g	Contains the original dataset provided by AIRBUS without the selected validation cases: **Database** 1 **CFD data** 2p5g **Sting** On **HTP** off **VTP** on **SST** full
SST_m1g_a	Contains the original dataset provided by AIRBUS without the selected validation cases: **Database** 1 **CFD data** m1g **Sting** On **HTP** off **VTP** on **SST** full
EARSM	Contains the original dataset provided by AIRBUS without the selected validation cases: **Database** 1 **CFD data** No Sting **HTP** off **VTP** on **EARSM** full
SST_m1g_b	Contains the original dataset provided by AIRBUS without a AoA and a Mach that have been extracted for Re 25e7 (in particular, simulations with Mach 0.824 and AoA -2 have been removed from the original database): **Database** 1 *subset* **CFD data** m1g **Sting** On **HTP** off **VTP** on **SST** full

Table 9 – Datasets.

We call SST_2p5g, SST_m1g_a, EARSM, SST_m1g_b the regression exercises as defined by the files in Table 9 and consistently with the selected turbulence model and the vertical load. The exact sizes of the data are then, in terms of points and flow conditions are reported in Table 10.

Exercise	$\text{points}_{\text{train}}$	$\text{points}_{\text{test}}$	$\text{flow_cond}_{\text{train}}$	$\text{flow_cond}_{\text{test}}$	points_wall
SST_2p5g	72,465,757	5,346,767	637	47	113,761
SST_m1g_a	68,484,122	5,460,528	602	48	113,761
EARSM	46,073,808	2,032,668	408	18	112,926
SST_m1g_b	72,597,118	1,365,132	638	12	113,761

Table 10 – Data dimensions for the regression exercises.

In a pointwise approach for regression, there are six inputs for SST_2p5g, SST_m1g_a and SST_m1g_b, $(x, y, z, M_\infty, \text{AoA}, \text{Re})$, and five inputs for EARSM $(x, y, z, M_\infty, \text{AoA})$. Note that no 2D reduced coordinates on the wall have been used.

6.2 Description of methods and tools

The submitted regression exercises do not involve a large input space (just $\mathbb{R}^5$ or $\mathbb{R}^6$) or output space (just $\mathbb{R}$). However, the number of C_p values for learning (tens of millions) is very large so that classical well-known regression methods like Gaussian Processes/Kriging, Support Vector Machines or Radial Basis Functions cannot be used. All of them involve a linear algebra problem scaling with data size that cannot be solved for design of experiments exceeding a few hundreds or a few thousands.
More recent approaches have been used relying mainly on their implementation in scikit-learn [34]. Both regression Trees/Random Forest and Multi-Layer Perceptron (MLP) were tuned to fit the data and were then assessed with the corresponding test set. As noticed in the previous section, the

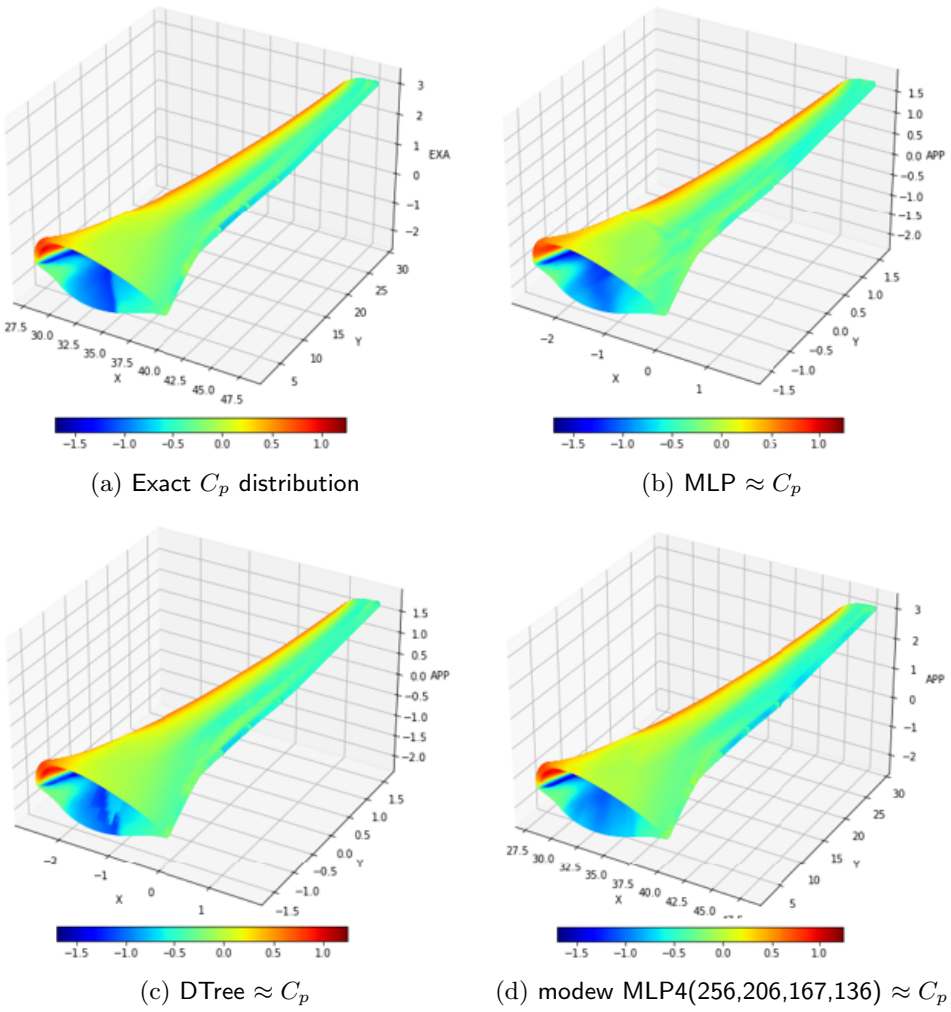
EARSM data set has the advantage with respect to those of SST_2p5g, SST_m1g_a, SST_m1g_b data sets that the wall points are the same for all flow conditions. More regressions were run with EARSM including (MLP) modewise regression where the complete C_p distribution is calculated as a function of the flow conditions (M_∞, AoA).
In the EARSM data set the M_∞ interval is $[0.5, 0.94]$, the AoA interval is $[-12, 15]$ (deg) and the constant Reynolds number per meter is 25.007×10^6.
The quality of the surrogate was evaluated with the classical R^2:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{k=1}^{k=t}(y_k - \hat{y}_k)^2}{\sum_{k=1}^{k=t}(y_k - \bar{y})^2} \tag{2}$$

where t is the size of the test set, y_k is one of the expected output value in the test set, $\bar{y}$ is the mean of the y_k and $\hat{y}_k$ is the value predicted by the surrogate at the inputs of y_k (that should hence, ideally, be equal to y_k). In addition, a series of C_p plots has been drawn on the XRF1 wing to get some intuition about the pros and cons of the various surrogates.

6.3 Results for the XRF1 EARSM regression exercise with scikit-learn

A series MLP with 1, 2, 3, 4, 5 and 10 hidden layers has been trained using scikit-learn. The selected optimization algorithm for the tuning of the weights and shifts is Adam and the selected activation function is tanh. The quality of the resulting surrogates is first assessed with the R^2 metric.
With one hidden layer, we get $R^2 = 0.760$ (100 neurons) to $R^2 = 0.800$ (180 neurons). With two hidden layers, $R^2 = 0.760$ ((30, 40) neurons) to $R^2 = 0.800$ ((80, 90) neurons). From three hidden layers with 55 neurons per layer, we get $R^2 = 0.920$, a value that is not significantly improved by more complex neural networks which training is more expensive. The regression were run on one Intel skylake node (44 cores) with the default stopping criterion and required from 1 to 15 hours depending on the number of internal parameters.
Among the regression algorithms available in scikit learn, Regression Tree and Random Forest appeared to also be able to manage as large data sets as those submitted. A regression decision tree is a set of nested decisions: the input variables (observed data) pass through a series of decisions, following a top down binary splitting process and, at the bottom level (the leaves), a unique output value is affected to the inputs casted by the successive decisions [35]. The regression appeared to be insensitive to the Decision Tree parameters but the maximum number of successive decisions (also called maximum depth) that cannot be bound without significantly altering the regression. The R^2 of the resulting surrogate is 0.850. Random Forests define a set of Decision Trees based on bagging techniques applied to the initial data. These methods did not appear to improve the regression for the problem at hand. With the same technical set up as before, the CPU time was about 400s for the Regression Tree and a few thousand seconds for the Random Forest (depending, of course, of the number of trees).
Finally, the initial data were scattered according to the flow conditions and then ordered with increasing x, then increasing y, then increasing z. This preprocessing allowed a modal approach, searching for the complete C_p distribution for the test flow conditions. In other terms, whereas the tested pointwise MLP were from an external viewpoint $[\mathbb{R}^5 \to \mathbb{R}(x, y, z, M_\infty, \text{AoA}) \to C_p]$ functions, these MLP networks are $[\mathbb{R}^2 \to \mathbb{R}^{112926}(M_\infty, \text{AoA}) \to C_{p\,\text{dist}}]$ functions. MLP with four and eight hidden layers were tested. The training appeared to be extremely fast (lasting just a few hundred seconds), probably due to the low number (18) of (very large) testing outputs. From four layers and (384, 309, 249, 204) neurons a precision $R^2 = 0.850$ was obtained, which could not be improved with NN involving more neurons.
In the end, a series of visual checks of the predicted C_p was performed. The C_p distributions presented hereafter correspond to $M_\infty = 0.9$, AoA $= -12.0$ (NB: for technical reasons, we used either the original coordinates or the scaled coordinates in the four plots). The inaccuracies, expressed by the R^2 values strictly lower than 1, also directly appear in the plots close to the root of the wing and at the trailing edge. In future works, complementary error plots will be drawn to more finely analyze the error locations. Besides, while looking at the goal C_p distribution of test data base, it appeared that some of the solutions for high positive angles of attack, like $(M_\infty, \text{AoA}) = (0.7, 15.0)$ or $(M_\infty, \text{AoA}) = (0.82, 14.0)$, are less regular than the other pressure fields, which will be a point of attention in future works.



(a) Exact C_p distribution

(b) MLP $\approx C_p$

(c) DTree $\approx C_p$

(d) modew MLP4(256,206,167,136) $\approx C_p$

Figure 16 – C_p distributions over the XRF1 wing at $M_\infty = 0.9$, AoA $= -12$.

6.4 Results for the XRF1 SST_2p5g and SST_m1g data sets. MLP with PyTorch

Pointwise MLP and Decision Tree regressors were also applied to exercises SST_2p5g, SST_m1g_a and SST_m1g_b data sets. No significant difference is to be noted with respect to EARSM, except in the best approximations obtained with Decision Trees and Neural Networks with sufficiently large number of numbers. The best R^2 values of the regressors are respectively 0.95, 0.9 and 0.97 to compare to 0.92 for EARSM data set. We understand this discrepancy as an expression of the difficulty to approximate the test data from the training data.

A series of three hidden layers MLPs has been built with PyTorch for the regression using the EARSM data set. This did not lead to improved results with respect to scikit-learn but rather to technical discussions about management of batches by both frameworks and execution speed on Intel nodes.

6.5 Conclusion

Significant experience has been gained at ONERA in the framework of the ML4AERO GARTEUR group regarding reduced-order modeling with recent Machine-Learning Techniques. Both Neural-Networks and Decision Tree/Random Forest will be further studied in internal projects. We also noted the interest of POD or Autoencoder plus classical ML techniques (Gaussian processes, RBF,...) for the coefficients of the POD modes or the inputs of the decoder. These numerical methods could not be studied by ONERA during the time span of the group but they will be in the future.

7. OPTIMAD

This section summarizes the activity performed by OPTIMAD on the Proper Orthogonal Decomposition (POD) method used in conjunction with a data-driven Machine Learning (ML) model in the

context of an aerodynamic shape optimization problem. The final goals of this experiment were to assess the accuracy of a data-driven ML model trained to predict pressure and skin friction distributions over a 2D airfoil, and to estimate the impact of noisy ML-surrogates in shape optimization problems.

7.1 Problem definition

The target problem is the constrained shape optimization of a 2D airfoil. The goal is the minimization of the overall drag force at fixed lift coefficient. Constraints are given on the momentum coefficient, on the angle of attack (AoA) and on the airfoil shape. The optimization problem can be summarized as follows:

$$\begin{aligned} &\min_{w \in W \subseteq \mathbb{R}^n} c_d \\ &\quad \text{s. to:} \quad c_l = 0.824 \qquad -0.11 \leq c_m \leq -0.01 \quad 3^\circ < \alpha < 5^\circ \\ &\qquad\qquad\quad t/c = 0.1211 \quad t_{85}/c \geq 0.02. \end{aligned} \tag{3}$$

Notation is standard. c_d, c_l and c_m are the drag, lift and momentum coefficient, α is the AoA, t/c is the ratio between the maximum thickness and the aerodynamic chord, t_{85} is the thickness at 85% of the chord.

7.2 Optimization methodology

The optimization problem was implemented in modeFRONTIER (by Esteco). A Multi Objective Genetic Algorithm (MOGA) was chosen to solve the optimization problem (3) with the maximum number of generations set to 30. This value was selected as the best compromise between running time and an exhaustive exploration of the parameters space.
Since large geometrical deformations can trigger instabilities in CFD computations, deformation parameters were constrained within suitable ranges in order to avoid both unrealistic shapes and high-curvature regions on the upper (lower) part of the airfoil.
A ground-truth optimized shape was computed using a CFD-based optimization. Each evaluation of the objective function (corresponding to a test geometry) was performed by resorting to a CFD solver. The selected CFD solver was SU2 (by Stanford University). Reynolds Averaged Navier Stokes (RANS) equations were integrated in SU2 (compressible solver) with an upstream Mach number of 0.734, Reynolds number of $6.5 \cdot 10^6$, and far-field temperature of 216.65K. In order to enforce constraints on the lift coefficient, the SU2 solver was run in "fixed C_L mode".

Surrogate optimization

In the POD+ML optimization, a POD+ML surrogate was used to approximate the objective function. Evaluations of the objective function were performed by querying the ML model in place of running expensive CFD computations.
During the online phase, the genetic optimizer produces a set of deformed airfoils, each corresponding to a set of deformation parameters. A first POD model (from now on, denoted as geometry-POD) is queried to obtain a low-dimensional encoding of the airfoil geometry. The encoding is given in terms of POD coefficients with respect to the truncated basis computed (offline) from a dataset of deformed airfoils. Coefficients from the geometry-POD are then fed into a ML model, which computes the coefficients corresponding to a POD basis for the pressure coefficient (C_p-POD) and for the friction coefficient (C_F-POD). Both POD bases are computed offline from a collection of CFD snapshots.
Once the coefficients for the C_p-POD and C_F-POD are known, pressure and friction distributions over the airfoil are reconstructed. From those, C_L, C_D and C_M can be obtained as customary in aerodynamic computations. Lift, drag, and momentum coefficients, together with the geometrical properties of the airfoil, are fed back into the optimizer for the next iteration. The resulting workflow is schematically depicted in Figure 17. The constraint on the lift coefficient is enforced using a device similar to that in SU2. Since feed-forward evaluations of the ML model are inexpensive, for each test geometry, the ML model is queried for multiple values of the AoA (uniformly sampled in the admissible range). Next, a root-finding algorithm is used to determine the AoA that realizes the target C_L. If such a value for the AoA cannot be found, the design is classified as unfeasible.

In this experiment, three different ML models were coupled with the POD method: (i) Radial Basis Function (RBF), (ii) Neural Network (NN) and (iii) Multi-fidelity Nonlinear AutoRegressive Gaussian Process (NARGP).

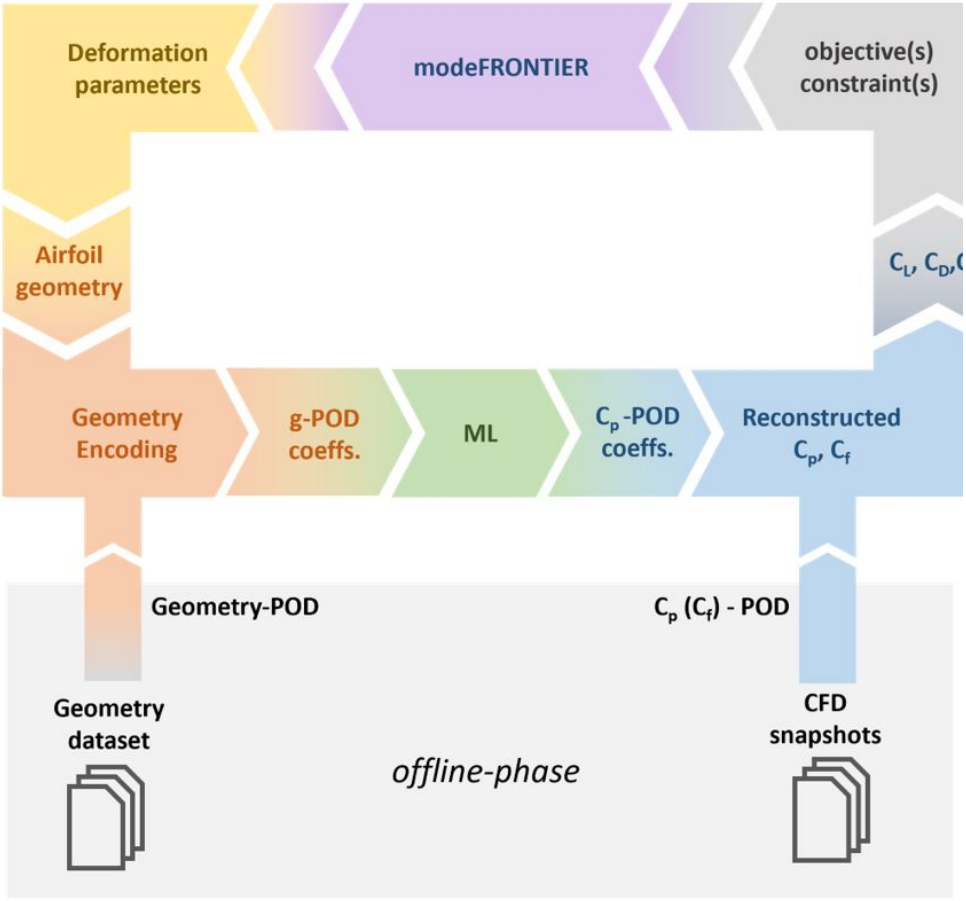


Figure 17 – Schematic depiction of POD+ML-surrogate optimization.

The training dataset for the geometry POD features 200 deformed airfoils which were sampled from the feasible region of the parameter space using a Halton sequence. For each airfoil geometry, the scaling factor (k) was set to 0.1211 to enforce the constraint on the maximum airfoil thickness. Computational meshes were generated in GMesh with roughly 57k computational cells.
The dataset of snapshots for the C_p-POD (C_F-POD) was generated by integrating RANS equations in SU2 as described in section 7.3.2. For each airfoil geometry, different simulations were performed at five AoA uniform spaced in admissible range (3-5 deg). The resulting dataset is composed of 1,000 CFD snapshots. It is worth mentioning that prior to computing the decomposition, fluid dynamic fields were mapped on a template mesh to ensure uniform size across all snapshots. This step was required since computational meshes generated by GMesh would differ slightly based on the airfoil geometry.
A validation dataset was also produced to assess the accuracy of ML predictions. The validation dataset was generated using the same procedure described above, except that validation points were created by random sampling.

ML model

The surrogate model deployed in the optimization workflow was trained to predict coefficients for the C_p-POD (C_F-POD) given an encoding of the airfoil geometry and a value of the AoA. In order not to rely on any explicit parametrization, an encoding based on the geometry-POD was chosen. It was found that this approach leads to higher prediction accuracy than an ML model trained to infer the quantities of interest from the deformation parameters. To create an encoding for the airfoil geometry, the geometry-POD basis was computed from the x, y coordinates of deformed airfoils in

the training dataset. The geometry-POD basis was truncated to ensure a low dimensionality of the encoding space, which has a beneficial impact on the training of the surrogate model. To determine the optimal dimensionality of the encoding space, an analysis of projection and reconstruction errors was performed. Based on the results of this analysis, only 15 POD coefficients were retained as the best compromise between reconstruction error of the encoded geometry and prediction accuracy of the surrogate model. The C_p-POD and C_F-POD basis were computed separately to allow fine-tuning of hyper-parameters independently for both models. In both cases, the POD basis was truncated to retain only the most energetic modes. This is customary, since injecting the noise associated to the last POD modes has a detrimental impact on the surrogate training.

7.3 Results

In order to assess the reduction on the drag coefficient, a first simulation of the baseline airfoil RAE2822 was performed at fixed C_L using the setup previously described. The aerodynamic coefficients and the angle of attack that realize the target C_L are reported in Table 11.

C_D	$2.45 \cdot 10^{-2}$
C_M	$9.68 \cdot 10^{-2}$
C_L	0.824
α [deg]	3.01

Table 11 – Main aerodynamic coefficients computed by CFD for the base airfoil, RAE2822.

CFD-based optimization

The described optimization procedure was implemented in modeFRONTIER. CFD simulations were run in SU2 (compressible solver) on 10 CPUs. The minimum C_D (reported in Table 12) is significantly smaller than the drag coefficient computed for the base RAE2822 with an overall reduction of 42.5%. As expected (see Figure 18), the pressure coefficient does not exhibit a strong drop due to the shock on the upper part of the airfoil.

Table 12 – Values of aerodynamic coefficients for the optimized airfoil.

C_D	$1.41 \cdot 10^{-2}$
C_M	$7.29 \cdot 10^{-2}$
C_L	0.824
α [deg]	3.22
t/c	0.12
t_{85}/c	0.03

RBF interpolation

An RBF interpolator was trained on the dataset of CFD snapshots using a thin plate spline kernel and a polynomial offset of degree one. Before deployment, the prediction accuracy was assessed on the validation dataset. The prediction error for the i-th validation snapshot was computed as:

$$\text{error}_i = \frac{||y_i - \hat{y}_i||_2}{||y_i||_2}, \tag{4}$$

where y_i represents the quantity of interest for the i-th snapshot while $\hat{y}_i$ is the prediction. A histogram reporting the distribution of the error across the validation dataset is shown in Figure 19.
The trained model was deployed in the described surrogate optimization. A high-fidelity CFD simulation was also run on the optimized geometry to evaluate the accuracy of the predicted aerodynamic coefficients. The comparison between aerodynamic coefficients predicted by the RBF model and computed by the CFD is reported in Table 13. It should be noted that the value of the drag coefficient for the optimal shape provided by the CFD simulation is considerably larger than the value predicted

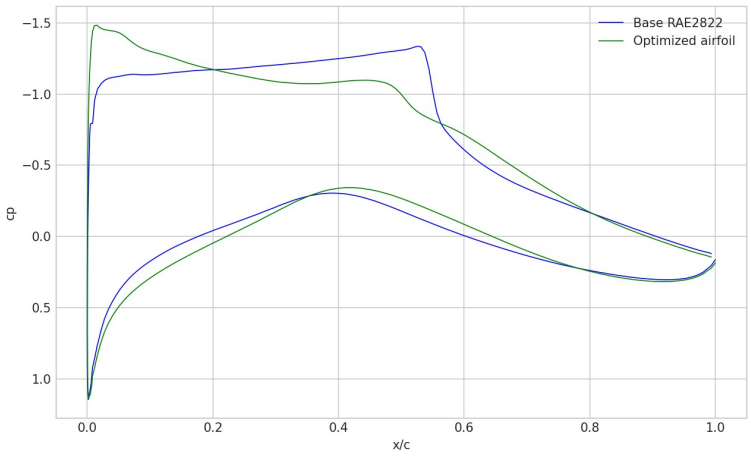


Figure 18 – Distribution of C_p on the upper (lower) surface of the optimized airfoil (green solid line) and for the base RAE2822 (blue solid line).

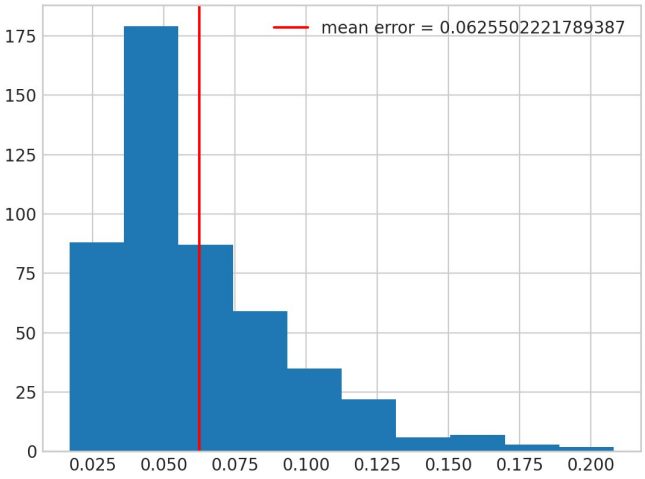


Figure 19 – Distribution of prediction error across the validation dataset. The red line indicates the average error.

by the RBF model. Figure 20 also shows that the C_p distribution resulting from CFD computations is characterized by a sharper profile of the shock as opposed to RBF predictions. Nevertheless, a reasonable reduction of 26.11% was achieved relative to the baseline airfoil.

Neural Network

The neural network model is a multi-layer perceptron composed of 5 fully-connected hidden layers of size 400 with leaky ReLU activation function. The neural network was trained using the Adam optimizer with a learning rate of 10^{-4}. Mean square error between predicted and computed C_p (C_F) POD coefficients was used as the loss function. 600 training epochs were performed with early stopping option enabled and 70-30 splitting between training and testing dataset. As shown in Figure 21, the average prediction error of the NN across the validation dataset is approximately 50% smaller than in RBF interpolation. Moreover, both shape and location of the shock are predicted with higher accuracy and sharper resolution. However NN predictions are quite noisy.

Although noisy predictions are not a desirable output, the presence of noise on the predicted distribution of C_p (C_F) did not constitute a significant problem during the optimization, since the quantities of interest (namely, aerodynamic coefficients) are computed as integral quantities. A possible reason for this behavior is that the loss function assigns the same weight to all coefficients of the C_p (C_F)-POD, including those associated to the (noisy) modes with the lowest energy content. This corresponds to inject noise during model training and eventually leads to noisy predictions.

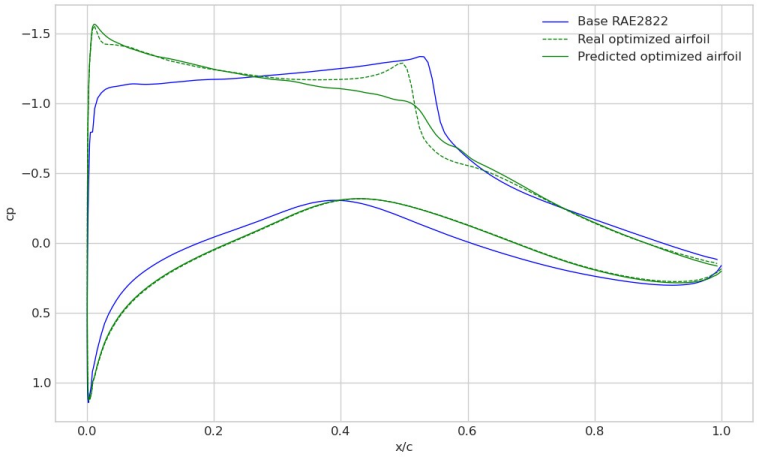


Figure 20 – Distribution of the pressure coefficient predicted by the RBF model (green solid line) compared to the distribution computed by CFD (dashed green line). For reference, C_p distribution for the base RAE2822 airfoil is shown in blue.

	RBF	**Hifi CFD**
C_D	$1.58 \cdot 10^{-2}$	$1.81 \cdot 10^{-2}$
C_M	$5.24 \cdot 10^{-2}$	$5.37 \cdot 10^{-2}$
C_L	0.824	0.824
α [deg]	3.55	3.71
t/c	0.12	0.12
t_{85}/c	0.035	0.035

Table 13 – Comparison between coefficients predicted by the RBF model and CFD.

One possible way to mitigate this problem is to first normalize the POD coefficients and then assign a weight to each term in the loss function proportional to the energy of the corresponding POD mode. In Table 14, the mean prediction error on the validation dataset for both C_p and C_F are reported for the model trained with and without a weighted loss. It should be noted that the mean prediction error obtained with a weighted loss is significantly smaller on both components of C_F, but larger on C_p. This is of course undesirable since C_p represent the main contribution to the overall drag coefficient. An alternative strategy is to train a different NN for each POD coefficient. With this strategy, a reduction of noise in the predictions was also observed. However, the overall error on C_p is increased even more. The resulting accuracy is almost equivalent to RBF interpolation.
Eventually, the original model (trained on all POD coefficients and without the weighted loss) was retained in order to have more accurate prediction on the C_p coefficient. Results for the NN-based optimization are reported in Table 15. Predictions of the C_p distribution on the upper/lower surface of the optimized airfoil are shown in Figure 22. Compared to the RBF interpolation, the reduction in the drag coefficient is considerable (39.1% with respect to the baseline RAE2822) and the optimal C_D predicted by the NN is very close to the drag coefficient computed by CFD. Moreover, the optimal point was found in almost half of the iterations required by the CFD-based optimization.

	RMSE loss	**Weighted loss**
C_p	0.04	0.05
$C_F(x)$	0.10	0.06
$C_F(y)$	0.14	0.06

Table 14 – Mean relative errors obtained with RMSE loss and weighted loss.

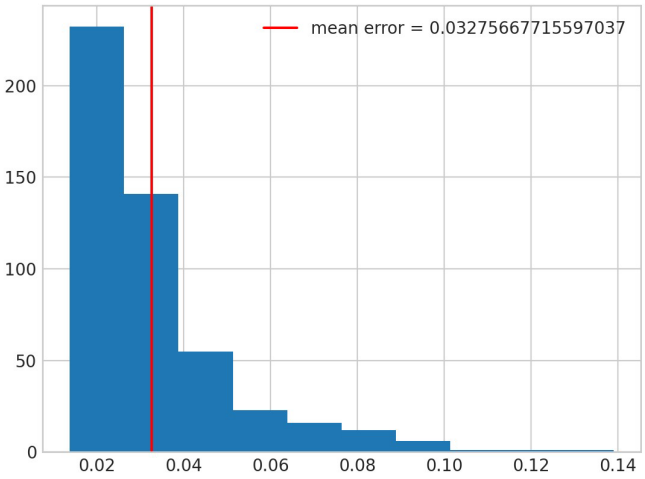


Figure 21 – Prediction error across the validation dataset. Red vertical line represents the average prediction error.

	NN	**Hifi CFD**
C_D	$1.37 \cdot 10^{-2}$	$1.49 \cdot 10^{-2}$
C_M	$7.08 \cdot 10^{-2}$	$7.28 \cdot 10^{-2}$
C_L	0.824	0.824
α [deg]	3.28	3.31
t/c	0.12	0.12
t_{85}/c	0.034	0.034

Table 15 – Comparison between main aerodynamic coefficients predicted by the NN and CFD for the optimized airfoil.

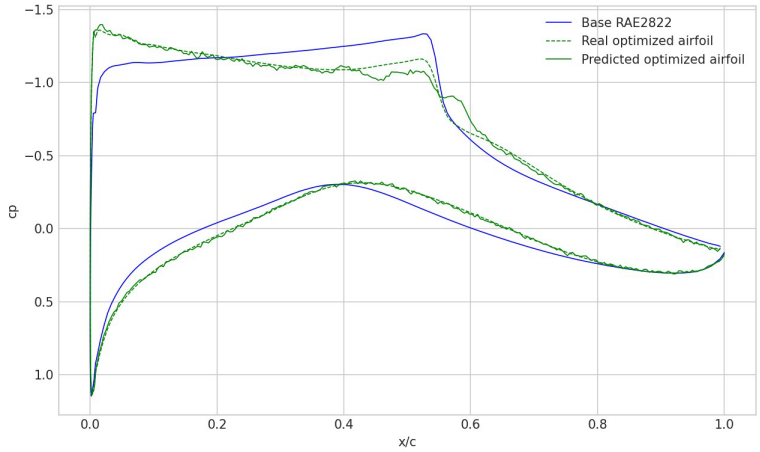


Figure 22 – C_p distribution on the upper and lower surface of the airfoil predicted by the NN (green solid line), and computed by the CFD (green dashed line) for the optimized airfoil. For reference, distribution of the pressure coefficient for the baseline RAE2822 is also shown (blue solid line).

Multi-Fidelity

The third optimization strategy tested in this experiment was a Multi-fidelity approach based on Nonlinear AutoRegressive Gaussian Process (NARGP). Here high fidelity data were taken from the first 50 training geometries of the dataset, each at 5 different values of the AoA. Hifi data points were complemented with 300 low fidelity CFD snapshots computed on a coarse mesh consisting of approximately 28k cells (as opposed to 57k computational cells used in high-fidelity computations). Parameters corresponding to low fidelity snapshots were sampled from the parameters space using a Halton sequence. Since the Halton sampling is nested, the first 50 low-fidelity samples corresponds to the high-fidelity snapshots, thus realizing a nested dataset.
NARGP model is trained on the multi-fidelity dataset. Training data are given by the multi-fidelity POD coefficients for pressure and skin friction coefficients. The error distribution across the validation dataset is shown in Figure 23. The average prediction error is comparable to that obtained with RBF interpolation. Also, shape and location predicted for the shock are comparable between the two models.

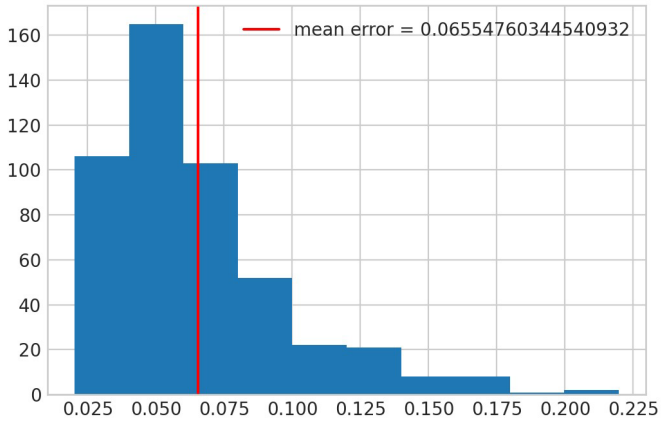


Figure 23 – Prediction error across the validation dataset. Red vertical line represents the average prediction error.

As shown in Table 16, NARGP seems to outperform RBF interpolation in terms of reduction in the drag coefficient (37.37% with respect to the base RAE2822). However, a closer inspection reveals that the C_p distribution predicted on the airfoil is quite distant from the true distribution (as shown in Figure 24).

	NARGP	**Hifi CFD**
C_D	$1.47 \cdot 10^{-2}$	$1.54 \cdot 10^{-2}$
C_M	$8.99 \cdot 10^{-2}$	$9.19 \cdot 10^{-2}$
C_L	0.824	0.824
α [deg]	3.07	3.04
t/c	0.12	0.12
t_{85}/c	0.037	0.037

Table 16 – Comparison between aerodynamic coefficients predicted by the NARGP model and computed by CFD for the optimized airfoil.

7.4 Conclusions

In this experiment, three different ML approaches were coupled with a POD reduced order model to infer distributions of pressure and skin friction coefficients from an encoding of the airfoil geometry. The three coupled ML+POD models were used as surrogates to solve a shape optimization problem. Performances were measured both in terms of prediction accuracy and in terms of reduction of the computational cost with respect to a standard CFD-based optimization. Results are summarized in

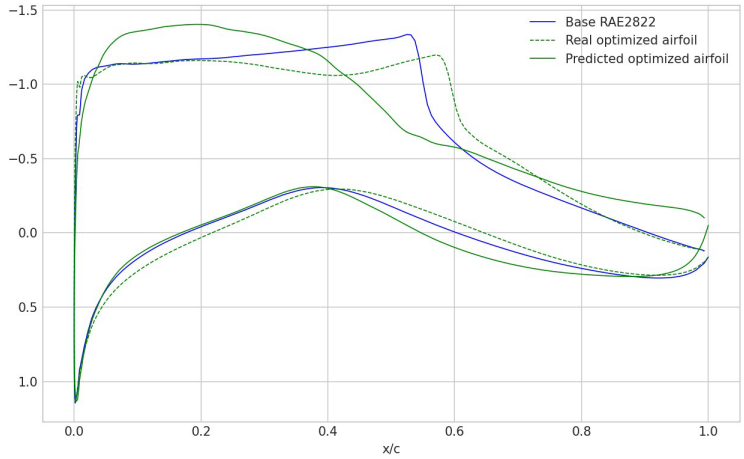


Figure 24 – C_p distribution on the upper (lower) surface of the airfoil predicted by the NARGP model (green solid line) and computed by the CFD (green dashed line) for the optimal airfoil. For reference, C_p distribution on the base RAE2822 is also shown (blue solid line).

Table 17. As expected, CFD-based optimization is the most effective but also the most expensive. The best results in terms of prediction accuracy were obtained by the NN+POD surrogate, which is characterized by the largest prediction accuracy and achieves the most significant drag reduction among all the ML surrogates tested in this experiment.

Table 17 – Performances comparison between POD+ML surrogates and CFD-based optimization.

	Validation error	C_D **reduction**	**CPU hours (incl. dataset gen.)**
CFD	–	42.5%	∼1200
RBF	6.25%	26.11%	∼287
NN	3.27%	39.10%	∼287
NARGP	6.55%	37.37%	∼287

Pertaining to the computational cost, all CFD simulations were performed at a fixed lift coefficient and required approximatively 80 CPU minutes (each). For the generation of the training dataset, five simulations were run at fixed AoA for each geometry. For each AoA (except for the first one), the flow field was initialized using the solution obtained from the previous value, resulting in a computational effort (per simulated airfoil) equivalent, on average, to a single SU2 simulation at fixed C_L. The online cost of surrogate-based optimization is negligible, since feed-forward evaluations are very fast.
Thus, we conclude that one single CFD-based optimization with approximately 200 function evaluations is equivalent (in terms of core hours) to the creation of the entire training dataset. Note that in our test case, CFD-based optimization took approximately 900 evaluations to converge.

8. Final Overview

The GARTEUR AD/AG60 project has successfully pursued its main objective of conducting advanced research in machine learning and data-driven methodologies for aerodynamic analysis. The project has comprehensively evaluated various techniques, such as deep learning, surrogate models, and machine learning, and demonstrated their applicability and effectiveness in the complex field of aerodynamics. The project findings offer valuable perspectives on the strengths and weaknesses of the various methodologies, providing a basis for informed decision making in aerodynamic analysis and prediction. The insights gained highlight the importance of a holistic approach to harnessing the full potential of machine learning in aerodynamics.
It is worth noting the valuable contribution of AIRBUS in providing the necessary databases for conducting this research. Additionally, we would like to emphasize the significance of the involvement of NLR, IRT, and AIRBUS D&S in the technical discussions held during the research group meetings.

In essence, the knowledge gained through the GARTEUR AD/AG60 project represents a significant advance in the search for optimal aerodynamic solutions based on machine learning techniques.

9. Contact Author Email Address

mailto: d.quagliarella@cira.it

10. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

11. Ackowledgements

The work reported in this paper has been conducted in the GARTEUR AG60 project “ML4AERO: Machine Learning and Data Mining Techniques for Aerodynamic Analysis and Uncertainty Quantification.”” which includes the following members: INTA, CIRA, INRIA, DLR, FOI, ONERA, NLR, OPTIMAD, IRT, UC3M, AIRBUS, AIRBUS D&S.
The authors wish to thank Esther Andrés Pérez (INTA), Rodrigo Castellanos (INTA), Javier Nieto-Centenero (INTA), Jaime Bowen (INTA), Alejandro Gorgues (INTA), Domenico Quagliarella (CIRA), Angelo Iollo (INRIA), Tommaso Taddei (INRIA), Philipp Bekemeyer (DLR), Derrick Hines Chaves (DLR), Olivier Amoignon (FOI), Boban Pavlovic (FOI), Jacques Peter (ONERA), Robert Maas (NLR), Thomas van Dooren (NLR), Haysam Telib (OPTIMAD), Alessandro Alaia (OPTIMAD), Angela Scardigli (OPTIMAD), Laura Balasso (OPTIMAD), Simona Cucchiara (OPTIMAD), Matthias de Lozzo (IRT), Anne Gazaix (IRT), Andrea Ianiro (UC3M), Stefano Discetti (UC3M), Angel Pascual (AIRBUS), Holger Barnewitz (AIRBUS), Marta Gonzalez (AIRBUS), Blanca Martinez Gallar (AIRBUS), Daniel Redondo García (AIRBUS), Xavier Bertrand (AIRBUS), Sergio De Lucas Bodas (AIRBUS D&S), and Daniel González Juárez (AIRBUS D&S) for their fruitful contribution to the GARTEUR AG60 and to the results presented in this paper.

References

[1] Philipp Bekemeyer, Reik Thormann, and Sebastian Timme. Investigation into gust load alleviation using computational fluid dynamics. 06 2019.

[2] N. Kroll, S. Langer, and A. Schwöppe. The DLR flow solver TAU - status and recent algorithmic developments. 2014.

[3] V. E. Kovalev and O. V. Karas. Calcul de l’écoulement transsonique autour d’une configuration aile-plus-fuselage compte tenu des effects visqueux et d’une région décollée mince. *La Recherche Aérospatiale*, (1):23–38, 1994.

[4] K. Zhang and M. Hepperle. Evaluation of the BLWF code - a tool for the aerodynamic analysis of transonic transport aircraft configurations. Technical report, Institut für Aerodynamik und Strömungstechnik, July 2010.

[5] L. Sirovich. Turbulence and the dynamics of coherent structures. I, II, III. *Quarterly of Applied Mathematics*, 45(3):561–590, 1987.

[6] Ian T. Jolliffe. *Principal component analysis for special types of data*. Springer, 2002.

[7] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

[8] Richard Everson and Lawrence Sirovich. Karhunen–loève procedure for gappy data. *JOSA A*, 12(8):1657–1664, 1995.

[9] T. Bui-Thanh, M. Damodaran, and K. Willcox. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal*, 42(8):1505–1516, 2004.

[10] Karen Willcox. Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition. *Computers & Fluids*, 35(2):208–226, 2006.

[11] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[12] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[13] Francisco Palacios, Juan Alonso, Karthikeyan Duraisamy, Michael Colonno, Jason Hicken, Aniket Aranake, Alejandro Campos, Sean Copeland, Thomas Economon, Amrita Lonkar, et al. Stanford university unstructured (SU2): an open-source integrated computational environment for multi-physics simulation and design. In *51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 287, 2013.

[14] Thomas D. Economon, Francisco Palacios, Sean R. Copeland, Trent W. Lukaczyk, and Juan J. Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016.

[15] Mark Drela. Xfoil: An analysis and design system for low Reynolds number airfoils. In *Low Reynolds Number Aerodynamics: Proceedings of the Conference Notre Dame, Indiana, USA, 5–7 June 1989*, pages 1–12. Springer, 1989.

[16] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52(1):477–508, 2020.

[17] Derrick Hines and Philipp Bekemeyer. Graph neural networks for the prediction of aircraft surface pressure distributions. *Aerospace Science and Technology*, 137:108268, jun 2023.

[18] Derrick Hines Chaves, Mateus Dias Ribeiro, and Philipp Bekemeyer. Physics-based regularization of neural networks for aerodynamic flow predictions. In *15th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control*. National Technical University of Athens, 2023.

[19] D. Hines Chaves and P. Bekemeyer. Data-driven reduced order modeling for aerodynamic flow predictions. In *8th European Congress on Computational Methods in Applied Sciences and Engineering*. CIMNE, 2022.

[20] Philipp Bekemeyer, Anna Bertram ..., and Nils Hoffmann. Data-driven aerodynamic modeling using the DLR SMARTy toolbox. In *AIAA AVIATION 2022 Forum*. American Institute of Aeronautics and Astronautics, jun 2022.

[21] Adam Paszke, Sam Gross ..., Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[22] Martín Abadi, Ashish Agarwal ..., Jeffrey Dean, et al. TensorFlow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[23] Thierry Braconnier, Marc Ferrier, J-C Jouhaud, Marc Montagnac, and Pierre Sagaut. Towards an adaptive pod/svd surrogate model for aeronautic design. *Computers & Fluids*, 40(1):195–209, 2011.

[24] Ralf Zimmermann. Gradient-enhanced surrogate modeling based on proper orthogonal decomposition. *Journal of Computational and Applied Mathematics*, 237(1):403–418, 2013.

[25] Michele Milano and Petros Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.

[26] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[27] Olivier Amoignon, Jiri Hradil, and Jan Navratil. A numerical study of adaptive ffd in aerodynamic shape optimization. In *52nd Aerospace Sciences Meeting*, page 0899, 2014.

[28] Hervé Bourlard and Yves Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4-5):291–294, 1988.

[29] Rohit K. Tripathy and Ilias Bilionis. Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification. *Journal of Computational Physics*, 375:565–588, 2018.

[30] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, 1998.

[31] Scikit-learn kernel PCA manual page. `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.KernelPCA.html`. Accessed: 2023-11-20.

[32] Scikit-learn kernel ridge regression manual page. `https://scikit-learn.org/stable/modules/generated/sklearn.kernel_ridge.KernelRidge.html`. Accessed: 2023-11-20.

[33] Scikit-learn grid search manual page. `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`. Accessed: 2023-11-20.

[34] F. Pedregosa ... and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] Leo Breiman. *Classification and regression trees*. Routledge, 2017.