33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

ICAS 2022 SWEDEN

AN ASSESSMENT OF REDUCED-ORDER AND MACHINE LEARNING MODELS FOR STEADY TRANSONIC FLOW PREDICTION ON WINGS

Rodrigo Castellanos[1,2], Jaime Bowen Varela[1], Alejandro Gorgues[1] & Esther Andrés[1,*]

[1]Theoretical and Computational Aerodynamics Branch, Flight Physics Department, Spanish National Institute for Aerospace Technology (INTA), Torrejón de Ardoz 28850, Spain
[2]Aerospace Engineering Research Group, Universidad Carlos III de Madrid, Leganés 28911, Spain
[*] corresponding author: eandper@inta.es

Abstract

An assessment of reduced-order models (ROMs) combined with machine learning regression algorithms for aerodynamic data prediction is presented. The analyses focuses on the prediction of pressure distributions on a 3D wing flying in the transonic regime based on computational fluid dynamics (CFD) data and given the angle of attack and the Mach number as flight conditions. Proper orthogonal decomposition (POD) and Isomap are the considered ROMs and are compared against a direct regression. A random forest (RF) and a Deep Neural Network (DNN) are explored to predict approximate CFD solutions at untried flight conditions. The paper describes the performance of efficient ROMs to ensure an improved treatment of the data to obtain an accurate prediction of pressure distributions at a reduced computational cost. The nonlinear features of Isomap as a manifold learning model combined with the DNN highlight the accurate determination of local, nonlinear events. A comparative assessment of the proposed ROM+Interpolator predictor against the direct interpolation is addressed, featuring the strengths and weaknesses of each approach.

Keywords: surrogate model; machine learning; reduced-order model; Isomap; POD; deep neural network; random forest; aerodynamics.

1. Introduction

The modern aerodynamic design is heavily supported by computational fluid dynamics (CFD) simulations, which are very demanding in terms of computational resources and time budgets. As of today, preliminary and intermediate technological development stages are driven by simplified models that provide a reasonable quality of the aerodynamic data at a fraction of the cost of high-accuracy strategies such as direct numerical simulations (DNS). The irruption of novel mathematical models in the realm of reduced-order modelling (ROM) and machine learning (ML) pursue a more agile design process and reduces the derived costs from using expensive computational resources [1]. However, most of these models have shown a considerable lack of accuracy in the presence of complex flows, such as wall-bounded turbulence due to its stochasticity, or shockwaves because of the abrupt change in the flow state, and a lack of robustness when tested at different circumstances than the reference, as a modification of Reynolds number or geometrical shape of the model.

Despite the intrinsic nonlinear nature of transonic phenomena, such flows are characterised by recurrent flow patterns and physical features that can be learned from simulation or empirical data. A powerful tool classically used for the order reduction of large-scale systems is proper orthogonal decomposition (POD) [2], also known in the field of statistics as principal component analysis (PCA) [3]. The POD provides the most efficient orthogonal basis to decompose the data in terms of energy content, i.e. the variance of the quantity to be analysed. Its application in aeronautics and fluid mechanics spans from a simplification of the Navier-Stokes equations into a system of linear differential equations employing Galerkin projections [4], to the improvement of regression problems

by combining it with an interpolator in the low-dimensional space [5] or combined with a CFD flux residual minimisation scheme to improve the simulation in transonic regimes [6]. Low-order models obtained from POD open a door to a vast space of applications; however, its underlying assumption that the flow solutions lie in a low-dimensional linear subspace of the high dimensional space makes highly nonlinear features insufficiently reproduced as could be the case of transonic three-dimensional flows.

There exist a vast variety of dimensionality reduction algorithms. The Gauss-Newton with approximated tensors (GNAT) method uses the GANT in a projection-based framework related to the residual minimisation [7]. The discrete empirical interpolation (DEIM) [8] relies on a classical POD-Galerkin approach although it facilitates the evaluation of nonlinear terms of the governing equations with an additional POD basis. The Multi-dimensional scaling (MDS) is based on the singular value decomposition (SVD) of the data distance matrix to project the data in a low-dimensional space preserving the distance between the snapshots in the high-dimensional space [9].

In this study, Isomap [10], as a nonlinear manifold learner, is compared against POD and combined with a regression model to predict transonic three-dimensional flows. Manifold learning aims at recognising the topologically closed surface (namely, the manifold) over which the data lies or near it. The manifold is a geometrical representation of the intrinsic relations that connect snapshots. This order reduction technique finds the nonlinear degrees of freedom that underlie complex natural observations thanks to the dimensionality reduction based on geodetic distances. Surprisingly, the application of Isomap in fluid mechanics and aeronautics is not widely exploited, with some contributions identifying manifold from flow-visualisation data [11], in the combustion field [12], and, recently, to comprehensively understand the physics in shear flows [13]. Within the state of the art of Isomap applications to surrogate modelling in aerodynamics, Franz et al. [14] developed *Isomap+I*, a parametric ROM to predict shock waves on a 3D wing in the transonic regime. The encoding part of the algorithm to reduce the dimensionality of the data is based on Isomap while a linear interpolation method is then used to predict the manifold coordinates of unexplored flight conditions in the manifold space. The inverse mapping to the high-dimensional space (decoder) is performed by a weighted average of the high-dimensional snapshots based on the distance of their projection in the low-dimensional embedding.

The present contribution relies on the previous work on manifold learning applied to aerodynamic data [14], aiming at further improving the performance of the interpolation and the decoder, based on a $k-$NN method to interpolate the pressure distribution of the nearest neighbours in the manifold space based on a weighted average by the distance. The Isomap is combined with two regression models (Isomap+I), namely a Random Forest [15] and a Deep (artificial) Neural Network [16]. Among the wide variety of data fit surrogate models, DNNs outstand as an alternative to extract the nonlinear features of the data. DNNs have been applied to optimisation in the design process of airfoils [17], the control of shedding flows using a DNN as the control agent [18] or in nonlinear system identification techniques such as NARMAX [19]. Several algorithms have been proven sturdy for aerodynamic data prediction [20]; hence, this study will evaluate the usage of deep neural networks (DNN) compared to more conventional methods such as tree-based regression models. A comparative assessment of the proposed *Isomap+Interpolator* method against the *POD+Interpolator* interpolator method is performed. The selected test case is a database of CFD simulations of a 3D-wing in the transonic regime. The performance of each method is evaluated and described, pointing out the main strengths and weaknesses of each approach.

The paper is structured as follows: First, the methodology is presented in Section 2., starting with a brief theoretical background of Isomap and POD in §2.2, and followed by a concise description of Random Forest and Deep Neural Networks in §2.3. Thereafter, the performance of the proposed ROM+I is discussed and compared. A set of results is presented in Section 3. for customary flight conditions and finally, the conclusions are drawn in Section 4..

2. Methodology

The employed methodology and database are described in this section. First, the database and the geometry of the considered model are outlined. Next, the POD and Isomap ROMs are described. A

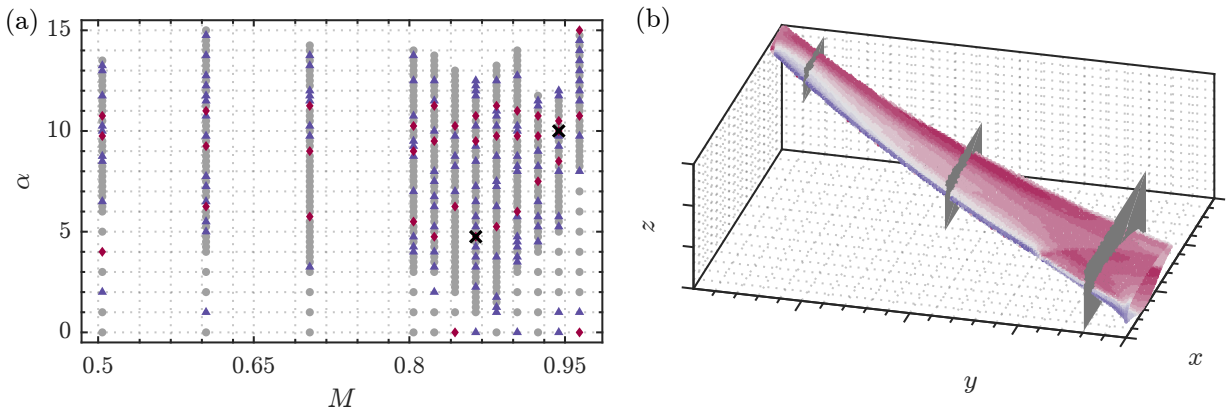

Figure 1 – (a) Distribution of the data set within the flight envelope: training (●), test (▲), and validation (◆) sets. Visualisation cases at $(\alpha_1, M_1) = (4.75°, 0.864)$ and $(\alpha_2, M_2) = (10.0°, 0.944)$ are highlighted (×). (b) Three-dimensional representation of the wing geometry coloured with the C_p distribution for visualisation case 1. Visualisation planes at $\eta = 0.1, 0.5, 0.9$ are highlighted (▬)

brief description of Random Forest and Deep Neural Networks is also provided, including the process to optimise the hyperparameters driving the models.

2.1 CFD model and database

The selected test case is a database of CFD simulations of the XRF1 model. The XRF1 model is an Airbus™ provided research test case to show the application of different technologies to a long-range wide-body aircraft. The work presented here has been performed in the frame of the Group for Aeronautical Research and Technology in Europe (GARTEUR) within the AD/AG60 research project. The aerodynamic data is obtained from Reynolds Average Navier-Stokes Simulations (RANS) of the full aircraft model to ensure a realistic condition by the interaction of the different aerodynamic subsystems. This work focuses on the wing with an underlying unstructured grid featured by $113,761$ points and the pressure coefficient C_p distribution on its surface for each of the grid points. The dataset is composed of 531 different flight conditions solved by the inviscid DLR TAU solver [21] at a fixed Reynolds number ($Re = 2.5 \times 10^7$). The flight condition parameters swept the whole flight envelope of the proposed aircraft, ranging the values of the Mach number (M) from 0.5 to 0.96, and computing the polar for angles of attack (α) spanning from $0°$ to $15°$ [see figure 1(a)].

The 531 flow solutions composing the database are divided in three sets: *train* set, 495 solutions used to train the ROM and regression model [(●) in figure 1(a)]; *test* set; 124 cases randomly selected from the whole database to test the trained models [(●) in figure 1(a)]; and *validation* set, 36 specific cases customarily selected to check the performance of the prediction [(●) in figure 1(a)]. The cases within the *validation* set are never considered for neither training nor optimization of hyperparameters, as later explained. These cases are selected to challenge the model, with three possible angles of attack for each considered Mach number. For now on, two cases within the *validation* set are chosen for visualisation purposes. The so-called *visualisation* cases are $(\alpha_1, M_1) = (4.75°, 0.864)$ and $(\alpha_2, M_2) = (10.0°, 0.944)$, which consider a challenging flight condition due to the high mach number and angle of attack [(×) in figure 1(a)].

It is to be noted that the load factor n_z ($= L/W$, being L the Lift and W the weight of the aircraft) implies an additional challenge for the prediction model since its value is fixed at $n_z = 2.5g$. Such a particular flight condition implies the upward deflection of the wing geometry with the consequent modification of the wing test with respect to the standard $n_z = 1g$ condition, as shown in figure 1(b). The alteration of the wing geometry prevents the irruption of shockwaves on the upper surface of the wing; however, there are strong C_p changes and separation regions for several points of the considered flight envelope.

2.2 Reduced-Order Modelling

Reduced order modelling is a mathematical field aiming at reducing the computational complexity or data handling requirement of a computational model, while preserving the expected fidelity and intrinsic

physics of the problem within a controlled error. Proper Orthogonal Decomposition is a conventional, well-established, linear ROM, whereas Isometric feature Mapping (Isomap) is a manifold learner with nonlinear features. Both proposed approaches consist of three steps: first, data is gathered from CFD simulations; second, the so-obtained data is embedded into a low-dimensional space using Isomap [22] or a POD [2]. This encoding part, which is fully data-driven, is carried out with the aim of revealing a hidden low-dimensional space that allows to relate the new coordinates to physical features of the flow. Finally, a decoding part that enables return to the high-dimensional space and to reconstruct the original flow field is developed.

For now on, consider that $N = 531$ flight conditions have been simulated, extracting the pressure coefficient on their wing surface. Each C_p distribution is an observation (point) in the high-dimensional space $\mathbb{R}^P$, where each dimension (feature) contains information about a point of C_p. Let $\mathbf{X} \in \mathbb{R}^{P\times N}$ be the data matrix containing the stated information and $\mathbf{x}_i \in \mathbb{R}^P$ be each of its rows, i.e. C_p for $i = 1,\dots,N$. The dataset in $\mathbf{X}$ is complex by nature and being able to extract a meaningful small number of coordinates that capture the main characteristics of the flow is challenging.

2.2.1 Proper Orthogonal Decomposition

The POD is a numerical, data-driven method to reduce the complexity of numerically-solved problems by reducing the dimensionality [2]. The first idea behind the POD is to find an optimally compressed description of the sequence of data $\mathbf{X} \in \mathbb{R}^{P\times N}$, which is achieved by the Singular Value Decomposition factorisation. The *compact SVD* is commonly applied due to its computational efficiency, in which the tank of the matrix $\mathbf{X}$ ($d \leq \min\{P,N\}$) determines the number of eigenvectors and eigenvalues upon decomposition,

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^* = \sum_{i=1}^{d} \sigma_i \mathbf{u}_i \mathbf{v}_i^* \tag{1}$$

where $\mathbf{U} = [u_1, u_2, \dots, u_d] \in \mathbb{R}^{P\times d}$ and $\mathbf{V} = [v_1, v_2, \dots, v_d] \in \mathbb{R}^{N\times d}$ are a orthogonal semi-unitary matrices, such that $\mathbf{U}^*\mathbf{U} = \mathbf{V}^*\mathbf{V} = \mathbf{I}_d$, and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_d) \in \mathbb{R}^{d\times d}$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_d > 0$. For the proposed database in this study, d coincides with the number of simulated flight conditions N, which is the dimension of the low-dimensional basis given by $\mathbf{U}$. Truncation of the number of modes is also possible by choosing a reduced rank $r < d$ so that only the r most energetic modes are conserved upon reconstruction. Thud, the reconstruction error is defined as the following,

$$RE = \frac{\sum_{i=1}^{r} \sigma_i}{\sum_{i=1}^{d} \sigma_i} \tag{2}$$

A conventional approach to select r is the so-called *elbow criterion* which identifies the number of POD modes n_{POD} at which adding further modes does not imply a significant improvement in the reconstruction in terms of energy content. For this work, the threshold was set to $RE \geq 99\%$ as later further described.

2.2.2 Isometric feature Mapping

Isomap is a nonlinear, dimensionality-reduction algorithm that computes the low-dimensional embedding of the data points that best preserve the geodesic distances measured in the high-dimensional input space. The following description is based on the Isomap algorithm implemented in the `scikit-learn` library for python [23], which has been used to carry out this investigation. This algorithm first relies on a conventional k-nearest neighbour (kNN) search to compute the matrix of Euclidean distances $d_{\mathbf{X}}(i,j)$ between data points $\mathbf{x}_i$ and $\mathbf{x}_j$ for all $i,j = 1,\dots,N$ to identify the k closest observations to $\mathbf{x}_i$, and to construct the neighbouring graph G over these data points such that two nodes i and j are connected by an edge of weight $d_{\mathbf{X}}(i,j)$ if they are neighbours. Given the pairs of vertices within G, Floyd's algorithm [24] is invoked to calculate the shortest paths between them, creating the matrix $\mathbf{D}_G$. Finally, the low-dimensional embedding is obtained $\Gamma \in \mathbb{R}^{N\times p}$, $p << P$, using a classical Multi-Dimensional Scaling (MDS) [9] on the matrix of shortest path distances $\mathbf{D}_G$ so that the Euclidean pairwise distances resemble those in the neighbouring graph $d_{\mathbf{G}}(i,j)$. From an optimisation perspective, this problem is equivalent to finding the matrix Γ that minimises the cost function

$$\mathscr{L}_{iso} = \left\| \Gamma\Gamma^{\top} - \mathbf{B} \right\|_F^2, \quad \text{where} \quad \mathbf{B} = -\frac{1}{2}\mathbf{H}^{\top}\left(\mathbf{D}_G \odot \mathbf{D}_G\right)\mathbf{H} \quad \text{and} \quad \mathbf{H} = \mathbf{I}_N - \frac{1}{N}\mathbf{I}_N, \tag{3}$$

being $\mathbf{B}$ the Gram matrix in the input space, $\mathbf{H}$ the centring matrix, $\mathbf{I}_N$ the identity matrix of dimension N, $\odot$ the Hadamard (element-wise) product and $\|\cdot\|_F$ the Frobenius norm.
The value of Γ that minimises $\mathscr{L}_{iso}$, for a given dimension p, is the matrix of the p eigenvectors $[\Gamma_1, \ldots, \Gamma_p]$ corresponding to the p largest (positive) eigenvalues of the matrix Λ arising from the eigen-decomposition of the Gram matrix $\mathbf{B}$, namely $\mathbf{B} = \mathbf{V}\Lambda\mathbf{V}^{\top}$ and $\Gamma = \mathbf{V}^{(p)}$.
The performance of the low-dimensional embedding is quantified by the *residual variance* [22] as in (4). This metric is the ratio of the residual sum of squares to the total sum of squares based on the matrix of Euclidean distances between each pair of points in the low-dimensional embedding $\mathbf{D}_\Gamma$ and the shortest distance matrix $\mathbf{D}_G$, namely

$$RV = 1 - R^2\left(\text{vec}\left(\mathbf{D}_G\right), \text{vec}\left(\mathbf{D}_\Gamma\right)\right), \tag{4}$$

where R^2 refers to the squared correlation coefficient and $\text{vec}(\cdot)$ is the vectorisation operator. Since the value of RV quantifies the information that remains unexplained by the low-dimensional embedding of the original data, the objective is to minimise for a given p and k.
The proposed back-mapping from the low- to the high-dimensional space is a purely data-driven approach based on $k-$NN method. Any data point $\mathbf{x}_i \in \mathbb{R}^P$ has its low-dimensional counterpart $\mathbf{y}_i \in \mathbb{R}^p$, $i = 1, \ldots, N$, so that $f : \mathbb{R}^p \longrightarrow \mathbb{R}^P$ is defined as the unknown back-mapping function. To reconstruct the C_p distribution for any $\mathbf{y} \in \mathbb{R}^p$, the $K-$Nearest Neighbors $\mathbf{y}_{(1)}, \ldots, \mathbf{y}_{(K)}$ and their high-dimensional counterparts, namely $\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(K)}$, are identified, computing the reconstruction as a weighted average of the neighbours with increasing importance based on the distance to the considered point.

2.3 Regression model for C_p prediction

The regression model is the mathematical "black-box" tool used to predict C_p surface distributions on the wing model under untried flight conditions. This work considers two well-established machine learning methods with different working principles: a tree-based random forest model and a fully-connected deep neural network.

2.3.1 Random Forest

The Random Forest (RF) belongs to the category of tree-based algorithms, being one of the best and mostly used supervised learning methods. In the field of aerodynamics, in particular, RF has been used to predict unsteady aerodynamic data at quasi-stall condition [25] or to determine a full three-dimensional flow around a body from CFD data [26]. Tree-based algorithms empower predictive models with high accuracy, stability and ease of interpretation, with good capabilities in capturing non-linear relationships within the data.
Decision trees are usually presented by a set of questions that then split the learning sample into smaller parts. The aim of this method is not only to find the models that produce accurate predictions but also to extract knowledge intelligently. The RF algorithm [15] works by creating a "forest" of decision trees that are randomly initialised. These trees are composed of a root (first node), internal nodes, and leaves. The main advantages of using these algorithms are that they are non-parametric and can model any complex relations between inputs and outputs without prior assumptions and they are robust to noise and outliers. An additional advantage of RF models is their adaptability to be further increased in size and complexity for progressively demanding problems, which is a feature that can not be exploited in other regression models such as neural networks.
Conversely, one of the major drawbacks of RF models is the sensibility to hyperparameter selection. RF performance is directly linked to a proper selection of the hyperparameters driving the internal optimisation process [27]. To control and optimise the learning outcomes of these algorithms, their hyperparameters must be tuned. The main hyperparameters that have been tuned are: the *number of trees* that are created for the prediction; the *bootstrap* of the data, which determines if all the data is fed to the all the trees or it is bootstrapped; the *maximum tree depth*, which determines the degree of non-linearity that the model will be capable of reproducing at the cost of possible overfitting; the

minimum samples split, which is the minimum number of samples required to spit an internal node of a tree; and the *minimum samples leaf*, which is the minimum number of samples required to be a leaf node.

2.3.2 Deep Neural Network

Deep learning (DL) [28] is part of a broader family of machine learning methods based on artificial neural networks (ANN) [16]. Deep neural networks are a specific architecture within DL, which are inspired by information processing and distributed communication nodes in biological systems. These mathematical models are characterised by using "neurons" which form layers of linear transformations with non-linear activation functions. These entities work by iterating and trying to minimise the loss function using gradient descent. The main advantages of these algorithms are their resilience to overfitting and the capacity to learn more of the data compared to other ML algorithms.

To construct a DNN, it is necessary to previously design the architecture of the network. There exist a multitude of transformations that can be applied to the inputs as linear operations, convolution operations or graph operations; as well as a multitude of activation functions. In this work, the chosen DNN is a multilayer perceptron (MLP) made up of different linear layers with the ReLU activation function. More precisely, the MLP at hand is made up of one input layer, 10 hidden layers, and an output layer. The input layer receives the flight condition in form of a vector (M, α), and this is fed to the hidden layers. Each hidden layer is identical and composed of 1024 neurons. After passing through them, the output layer returns the regression values.

To train the network it is necessary to define a number of epochs, a loss function, and a gradient descent optimiser. In this case, the number of epochs is 1.5×10^4, a high enough value to ensure convergence of the regression model while avoiding overfitting. Regarding the loss function, the classical Mean Squared Error (MSE) is chosen. Finally, the optimiser is set to be the Adam Optimiser [29], a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments. In each epoch, the loss is computed and a gradient descent optimisation is computed aiming at minimising it.

2.4 Model tuning: minimisation of reconstruction error and optimisation of hyperparameters

A maximisation of the model performance is carried out by optimising the hyperparameters driving each model. The hyperparameters, in the case of regression models, are the inner specifications of the algorithms that regulate the learning process. For the POD and Isomap algorithms, it was also decided to consider some parameters to be optimised to minimise their reconstruction error while maximising the overall performance of the surrogate model.

For POD, The number modes to be truncated depends on the data. The considered dataset does not exhibit an evident number of points at which truncation makes physical sense. It was decided to set the minimum number of modes as that required to reconstruct at least 99% of the energy content (based on Eq. 2). The minimum number of modes that need to be maintained is $n_{POD}|_{min} = 320$, computed after performing the POD decomposition of 100 randomly selected *train* set within the whole database. The upper limit in the number of POD modes is the rank d of the snapshot matrix $\mathbf{X}$ ($d = 371$) as described in §2.1 for the *train* set.

Regarding Isomap, several parameters need to be tuned with few mathematical or physical foundations on the selection criteria. Those parameters are the number of neighbours to construct the neighbouring graph, k; the number of Isomap variables, n; and the neighbours employed in the back-mapping, κ. Selecting a small k could split the manifold into a disjoint sub-manifold thus losing its real structure, while a high k may connect points that are far in the high-dimensional space due to the non-convexity of the manifold. In this work, the methodology presented by [30] is considered to determine a valid range of values $[k_{\min}, k_{\max}]$ to perform the search. The lower bound of the interval, $k_{\min}$, is selected as the smallest k so that the neighbouring graph G is connected, whereas the upper bound, $k_{\max}$, must hold the following relation based on the number of edges E and nodes N in G:

$$\frac{2E}{N} \leq k + 2, \tag{5}$$

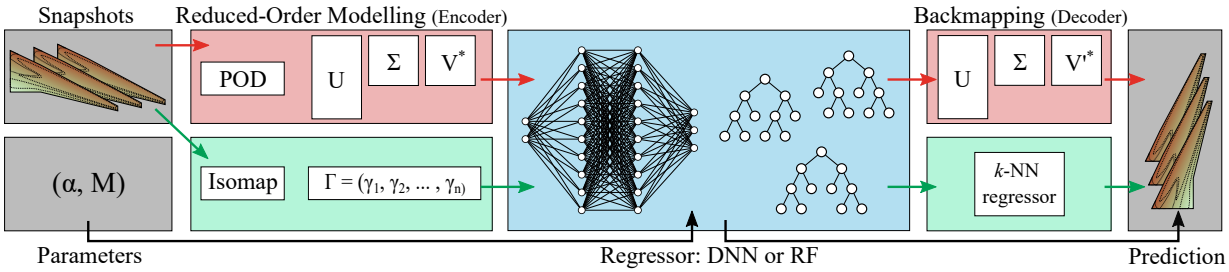

Figure 2 – Flow chart of the regression process for three ROM options: no ROM model (—), POD (—), and Isomap (—). The encoder and decoder blocks for POD (▭) and Isomap (▭) are separated. Input/Output data are depicted inside grey blocks (▭), the regression model is depicted inside the blue block (▭).

For the proposed database, the range k is highly dependant on the number of components; nonetheless, the selected range of $k \in [5,20]$ provides very small values of the residual variance. The range of valid Isomap coordinates n is chosen based on the minimisation of the residual variance. The influence of n above $n = 2$ results to be negligible for increasing number of k. Hence, the preferred range is $n \in [2,4]$. Concerning κ, the selection is merely based on experience and trial and error based on the database, selecting a range $\kappa \in [2,10]$. The lower limit, $\kappa_{\min}$, ensures a weighted average of the two closest neighbours in the low-dimensional space, while the upper limit, $\kappa_{\max}$, prevents the algorithm to find neighbours that are not close inside the manifold structure.
The second part of the hyperparameter optimisation deals with the regression algorithm. On the one hand, the DNN regression model is not further improved since the results were reasonably optimised, considering the architecture described in §2.3.2. On the other hand, for tree-based methods, the importance of hyperparameter tuning is notorious [27]. The Optuna framework [31] is used to run the Bayesian optimisation process. A total of $O(\zeta)$ (being ζ 1% of the overall dimension of the parameter space for each case) iterations are performed in which the RF is trained with a random split of the data set to prevent overfitting. The Bayesian optimiser is driven by the minimisation of the the Mean Squared Error (MSE) and the coefficient of determination (R^2).

2.5 Global surrogate model: learning process

The surrogate model learning process for the C_p prediction on the surface of a wing in the transonic regime is outlined in figure 2. The surrogate model receives as input the flight condition (α, M) and provides a surface distribution of C_p. The learning process depends on whether ROM is considered or not. For the case in which a direct interpolation is pursued (no ROM applied), the regression model (DNN or RF) is fed with the C_p snapshots of the *train* set and the corresponding flight condition (α, M). The regression model aims at minimising the mean square error (MSE) of the predicted C_p values.
The process for POD and Isomap follows a similar flowchart. The snapshots are first processed through the encoding block in which the data is projected in the low dimensional space. The low-dimensional snapshots are provided to the regression model, which learns to predict such reduced representation of C_p for a given flight condition. The predicted C_p is computed in the decoder block in which the data low-dimensional data is transformed back to the high-dimensional space of the original snapshots.
The advantage of this process lies in the considerable reduction of the amount of data to be processed by the regression model. The direct interpolation approach is constrained by the size of the data, which is given by a $\mathbb{R}^P$ space, being $P = 113,761$ the number of grid points. On the other hand, for POD, the data dimension is reduced to $\mathbb{R}^{N_{POD}}$, with $320 \leq N_{POD} \leq 371$ being the number of POD modes considered for the snapshot reconstruction. Furthermore, the Isomap model reduces the data to $\mathbb{R}^{N_{iso}}$ for $2 \leq N_{iso} \leq 4$, which is the number of Isomap coordinates to create the manifold in the low dimensional space. It is to be noted that the selection of the final value of N_{iso} and N_{POD} is driven by the Bayesian optimisation of hyperparameters described in §2.4.

3. Results

The proposed surrogate models are evaluated in this section. First, the models without ROM are analysed and their performance is discussed. Then, the same regression models are tested for POD and Isomap embedding, fitting the data in the low-dimensional space. The results are shown in the same fashion for all the evaluated cases for clearness, consisting of a regression plot, the chordwise C_p distribution at $\eta = 0.1, 0.5, 0.9$ and the surface distribution of the prediction deviation from the actual simulation data, $C_p - \tilde{C}_p$. The results are depicted for *visualisation* cases described in §2.1, namely $C_1 : (\alpha_1, M_1) = (4.75°, 0.864)$ and $C_2 : (\alpha_2, M_2) = (10.0°, 0.944)$. The global metrics for the *validation* set are given in table 1. The metrics for each validation case independently are attached in table 2 in the appendix.

	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	
	R^2	MSE*	R^2	MSE*	R^2	MSE*	R^2	MSE*	R^2	MSE*	R^2	MSE*
Mean	0.958	0.800	0.888	2.439	0.918	1.626	0.953	0.924	0.862	2.387	0.960	0.798
Mean*	0.965	0.720	0.898	2.209	0.924	1.550	0.959	0.847	0.908	1.737	0.962	0.756
Median	0.976	0.392	0.921	1.597	0.932	1.189	0.964	0.576	0.914	1.520	0.973	0.494
std	0.054	0.956	0.098	2.576	0.059	1.099	0.053	0.949	0.295	4.282	0.035	0.786

Table 1 – Key Performance Indicators of the surrogate models. Size of the trained model and descriptive global statistics of the *validation* set: mean, median and standard deviation (std) for the correlation coefficient R^2 and the Mean Square Error MSE (MSE* $=$ MSE $\times 100$). Mean* operator excludes the maximum and minimum value from the mean operation.

3.1 Direct interpolation: performance of RF and DNN

The performance of a direct interpolation of the simulation data is assessed in this section. The results are presented for cases C_1 and C_2 in figures 3 and 4, respectively. For C_1, at lower Mach number and angle of attack, the prediction is reasonably accurate for both regression models. The Random Forest, however, is not able to replicate the C_p distribution in the presence of the abrupt pressure changes as it is the case at $\eta = 0.5$ (figure 3,b) and $\eta = 0.9$ (figure 3,c). The prediction error, $C_p - \tilde{C}_p$, increases in the wing area closer to the fuselage from the mid-chord streamwise position towards the trailing edge and along the pressure depression that span through the wing. Conversely, the DNN perfectly captures the nonlinear nature of the pressure drop phenomena, predicting quite accurately the pressure changes on the upper surface of the wing. The prediction error slightly accentuates in the vicinity of these pressure depression; nonetheless, the DNN is robust where the RF fails at predicting. Regarding case C_2, the regression models follow a very similar behaviour. This case is characterised by high M and α values, almost at the edge of the flight envelope and with fewer surrounding points for a proper interpolation. The error is negligible for most of the wing's upper area towards the tip; however, for the root section, in which the high angle of attack seems to considerably affect the pressure distribution, the prediction fails to follow the C_p at the upper surface for both models.
Despite the restrictions to avoid overfitting in the RF hyperparameter tuning, it is to be remarked the considerable difference in size of the model with respect to the DNN. The trained RF for a direct C_p prediction requires 11.4GB of memory storage whilst the DNN is able to overperform at a small fraction of the memory requirement, 485MB. This technological constrain is very relevant for a realistic implementation of these kind of models and points out the incapacity of tree-based algorithms to expand from a flight condition (M, α) to a full surface distribution of C_p.

3.2 Interpolation in the low-dimensional space

A first reflection regarding the computational requirement of each model is required. The ROM models comply with their task of reducing the complexity of the problem from a data management perspective and this is undeniable from the size of ROM+Regression models compared to those of the direct interpolation from the previous section. The POD reduces the output data array from the regression model from 113761 elements to N_{POD} elements, which implies a factor of ~ 300. On the other hand,

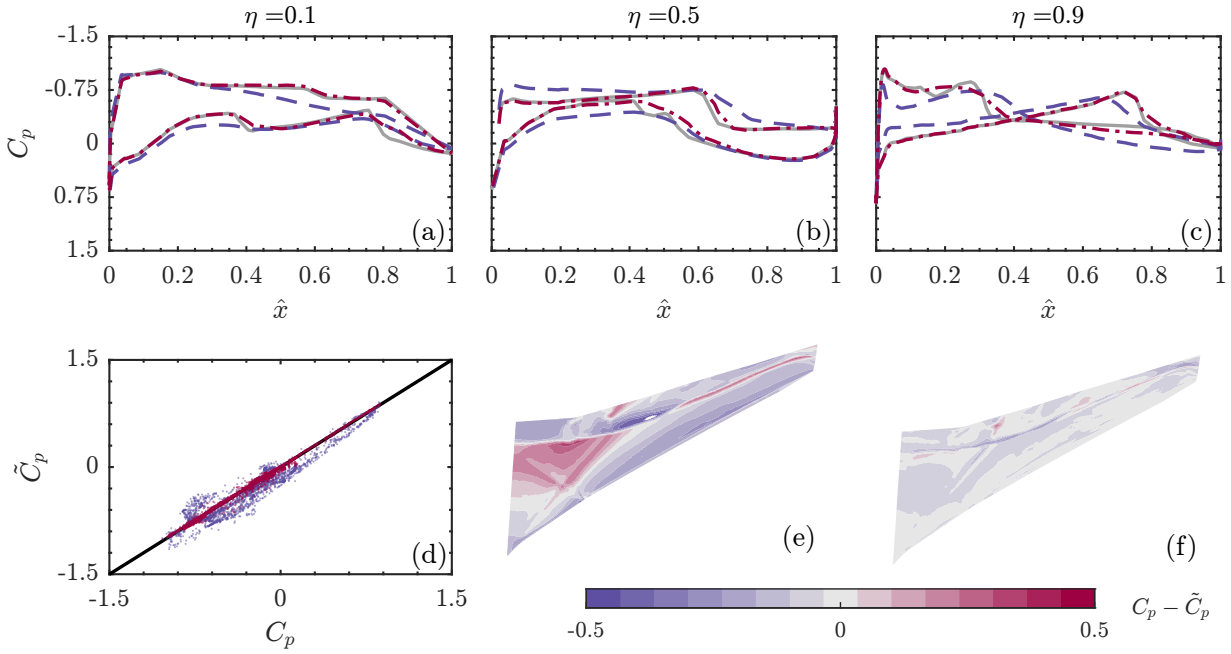

Figure 3 – C_p distributions predicted at $C_1 : (\alpha_1, M_1) = (4.75°, 0.864)$ without ROM. (a-c) Chordwise C_p distribution at $\eta = 0.1, 0.5, 0.9$ for TAU (—), RF (– –) and DNN (–·–). (d) Regression plot for RF (●) and DNN (●). (e,f) Prediction error in C_p distribution for RF and DNN, respectively.

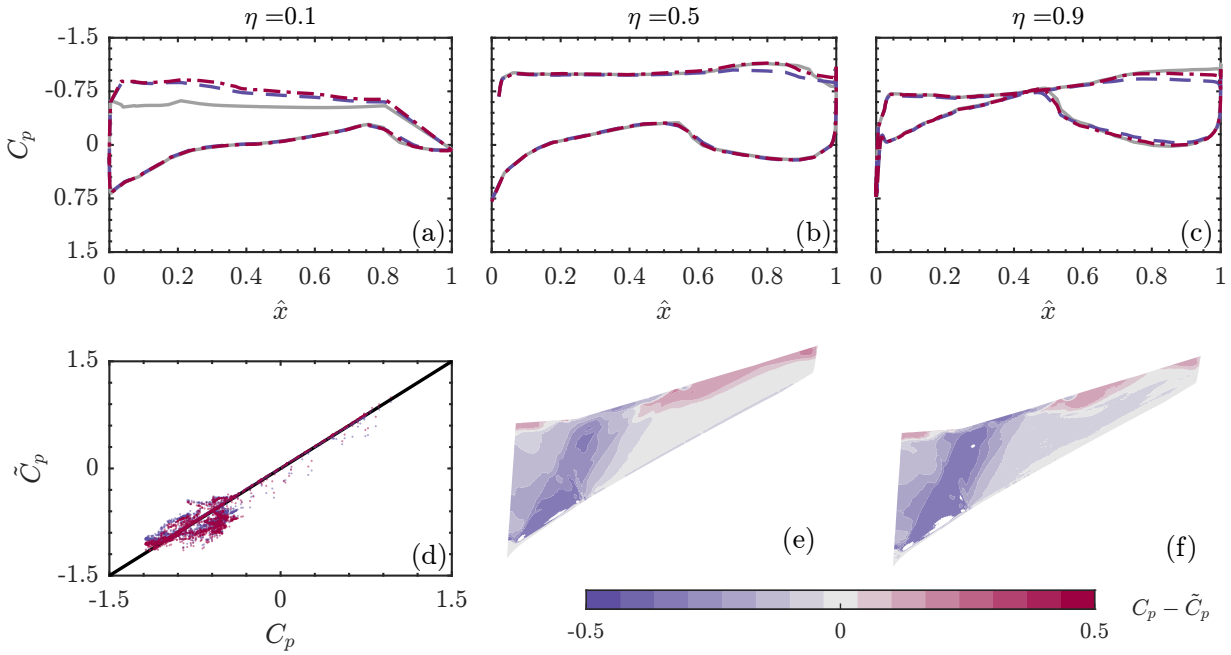

Figure 4 – C_p distributions predicted at $C_2 : (\alpha_2, M_2) = (10.0°, 0.944)$ without ROM. (a-c) Chordwise C_p distribution at $\eta = 0.1, 0.5, 0.9$ for TAU (—), RF (– –) and DNN (–·–). (d) Regression plot for RF (●) and DNN (●). (e,f) Prediction error in C_p distribution for RF and DNN, respectively.

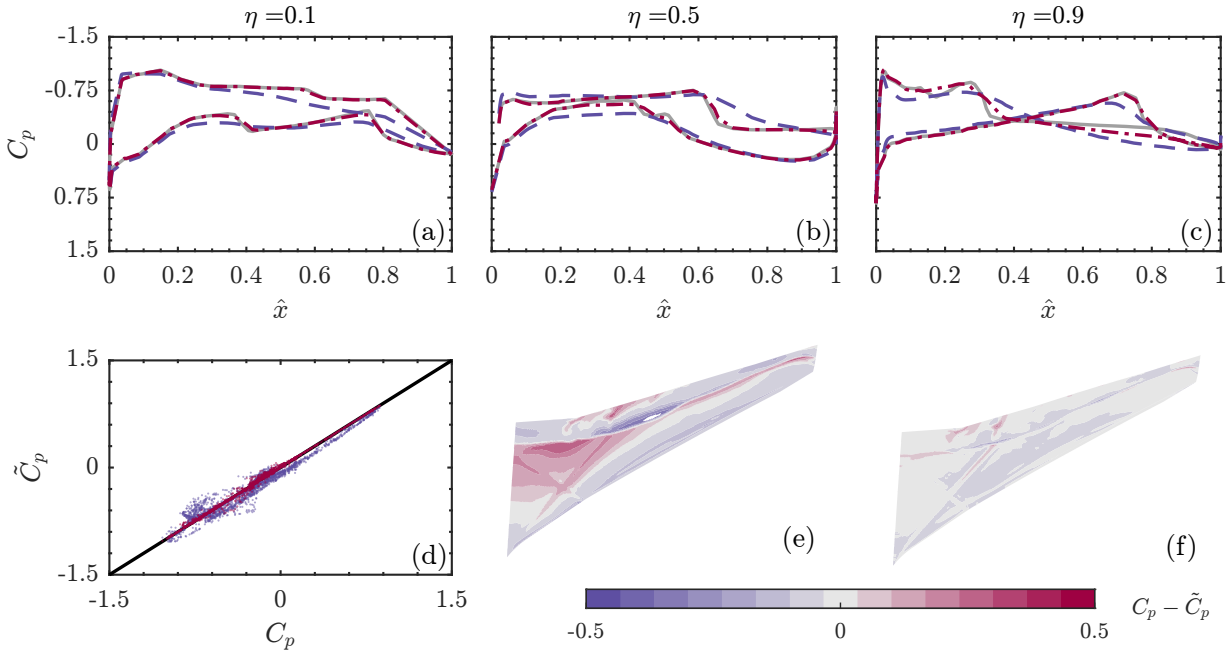

Figure 5 – C_p distributions predicted at $C_1 : (\alpha_1, M_1) = (4.75°, 0.864)$ with Random Forest regressor. (a-c) Chordwise C_p distribution at $\eta = 0.1, 0.5, 0.9$ for TAU (—), POD+RF (– –) and Isomap+RF (–·–). (d) Regression plot for POD+RF (●) and Isomap+RF (●). (e,f) Prediction error in C_p distribution for POD+RF and Isomap+RF, respectively.

the Isomap reduces the predicted array to just N_{iso} elements translated in a reduction by a factor of ~ 57000. The impact is tangible in the reduced size of the regression models and a reduction of the training and predicting computational times.

Despite the fact that the dimensionality reduction of Isomap is more abrupt than for POD, its nonlinear nature to construct the low-dimensional embedding results in a better overall performance for both regression models as depicted in figures 5 and 7 for case C_1, and figures 6 and 8 for case C_2. The performance of POD+RF is very similar to that of the direct interpolation for both cases; however, the Isomap+RF further improves the performance by being able to capture the pressure depressions and nonlinear phenomena at high Mach. The performance of the RF model is considerably improved towards the tip ($\eta = 0.9$) in which the pressure distribution at the lower surface changes significantly due to the combined effect of the load factor and angle of attack.

The DNN performs very well in case C_1 in conjunction with Isomap and with similar weaknesses to the other regression model when combined with POD. However, the combination of POD+DNN for the case C_2 is considerably detrimental when compared to POD+RF. The linear dimensionality reduction derived from POD is not able to get the proper flow distribution from midspan towards the tip, where the nonlinear phenomena associated with high Mach and geometric twist modifications become very relevant. The Isomap+DNN, on the contrary, accurately follows the pressure distribution at the three selected stations of η with a very uniform and small prediction error all over the upper wing surface.

4. Conclusions

An assessment of reduced-order modelling combined with machine-learning based regression models has been performed for a three-dimensional wing flying in the transonic regime at high load factor conditions. The considered database implies intrinsic challenges associated with the three-dimensional flow nature, the presence of nonlinear events such as shockwaves and abrupt pressure changes due to high Mach numbers; and the alteration of the wing geometry due to the flight condition at load factor $2.5g$, which modifies the nominal aerodynamic response of the wing since the local angle of attack changes along the wingspan.

The first conclusion of this work is the relevance of ROM for problems in which the data dimensionality is considerably greater than the parameters driving the prediction process. The regression model

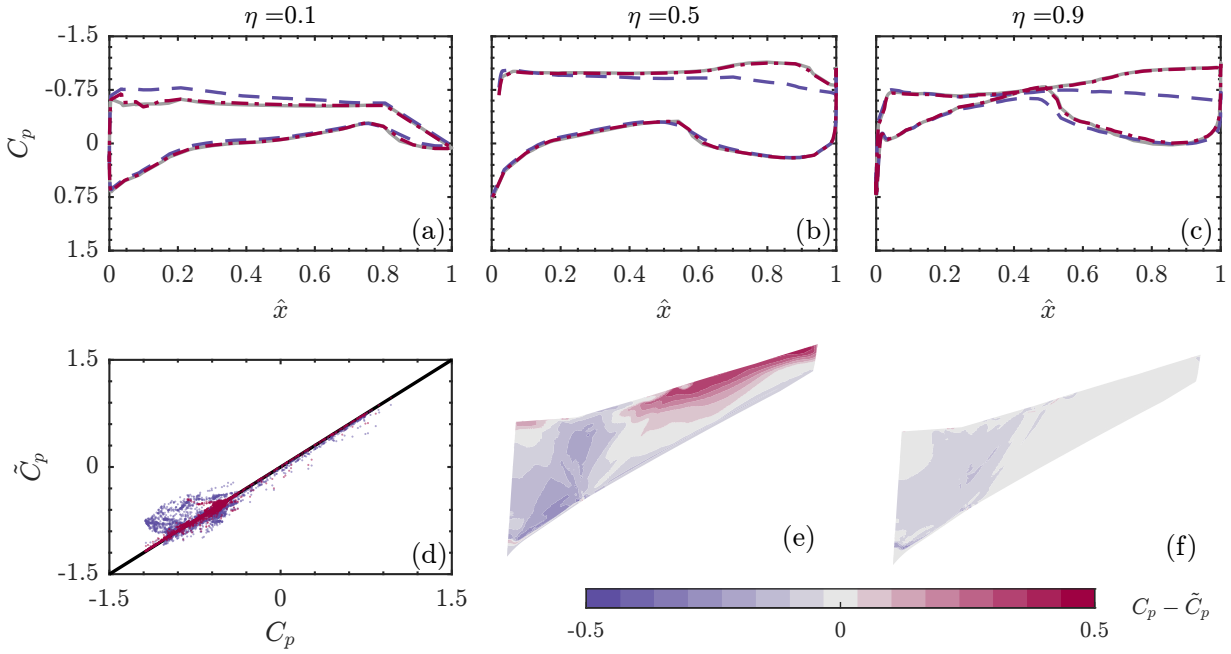

Figure 6 – C_p distributions predicted at $C_2 : (\alpha_2, M_2) = (10.0°, 0.944)$ with Random Forest regressor. (a-c) Chordwise C_p distribution at $\eta = 0.1, 0.5, 0.9$ for TAU (—), POD+RF (– –) and Isomap+RF (–·–). (d) Regression plot for POD+RF (●) and Isomap+RF (●). (e,f) Prediction error in C_p distribution for POD+RF and Isomap+RF, respectively.

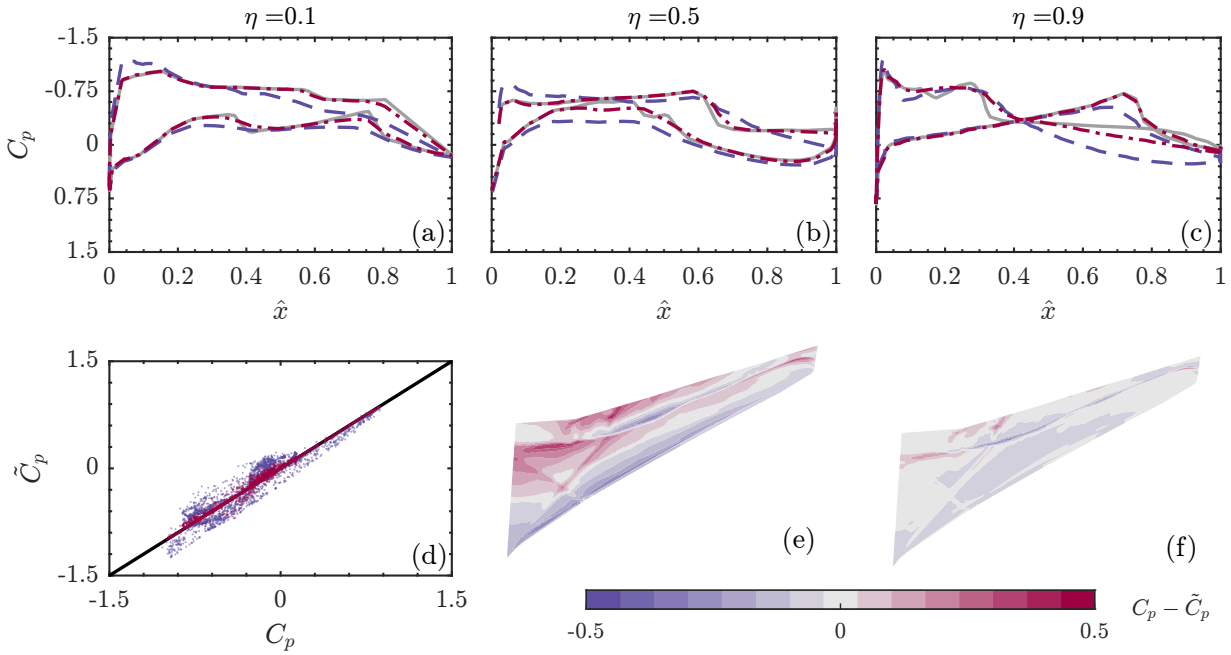

Figure 7 – C_p distributions predicted at $C_1 : (\alpha_1, M_1) = (4.75°, 0.864)$ with Random Forest regressor. (a-c) Chordwise C_p distribution at $\eta = 0.1, 0.5, 0.9$ for TAU (—), POD+DNN (– –) and Isomap+DNN ( ). (d) Regression plot for POD+DNN (●) and Isomap+DNN (●). (e,f) Prediction error in C_p distribution for POD+DNN and Isomap+DNN, respectively.

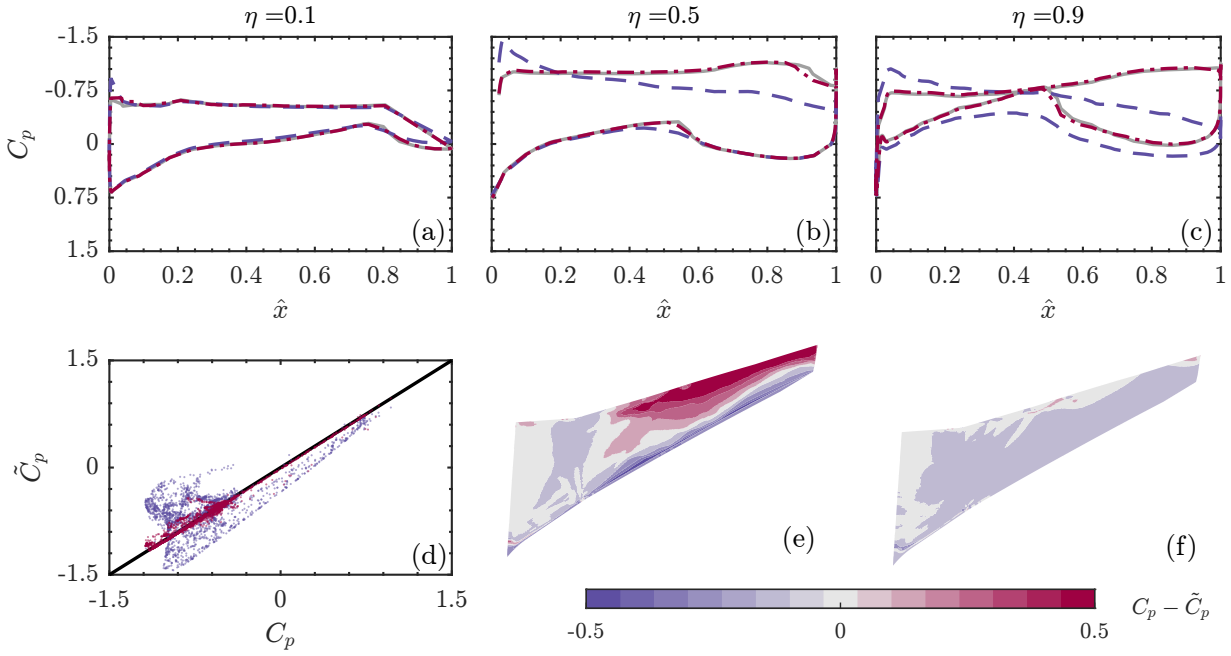

Figure 8 – C_p distributions predicted at $C_2 : (\alpha_2, M_2) = (10.0°, 0.944)$ with Random Forest regressor. (a-c) Chordwise C_p distribution at $\eta = 0.1, 0.5, 0.9$ for TAU (—), POD+DNN (– –) and Isomap+DNN ( ). (d) Regression plot for POD+DNN (●) and Isomap+DNN (●). (e,f) Prediction error in C_p distribution for POD+DNN and Isomap+DNN, respectively.

expands the information from the parameters (α, M) to the final output, $\tilde{C}_p$, which implies a considerable amount of internal operations and coefficients. The inclusion of ROM reduced the computational storage requirement by a factor of ~ 300 and ~ 57000 when using POD and Isomap, respectively.
The combination of ROM+Regression is also interesting from the physical perspective. The projection of the data in the low-dimensional space allows identifying the main features within the data, avoiding spurious information that could bias the C_p prediction. This was observed for the case $Isomap + RF$, which performs better than the direct interpolation based on RF. For the POD, however, the linear nature of the low-dimensional emending is not able to capture complex phenomena associated with the considered flight conditions. The DNN over-performs compared to RF for almost all the considered scenarios. The capabilities of DNN at a minimum computational cost and with little data available are outstanding as commented in this article. Moreover, the tuple combining Isomap and DNN springs up as an up-and-coming surrogate model.
Any comparison contains subjective biases associated with the computational load, the number of parameters, the complexity of the database, and even the experience of authors with various approaches. Also, each approach could have been further improved. e.g., the DNN architecture and RF hyperparameter tuning. Yet, this study points already to desirable features of two different machine learning surrogate models combined with dimensionality reduction algorithms.

Contact Author Email Address

mailto: eandper@inta.es

Copyright Statement

Code Availability

The Python codes used in this research are publicly published: `https://github.com/TACOMA-INTA/icas2022`.

A Key performance indicators of the surrogate models for the validation cases

M	α	Isomap + RF		POD + RF		RF		Isomap + DNN		POD + DNN		DNN	
		R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2	MSE
0.504	10.75	0.938	0.028	0.730	0.121	0.910	0.040	0.931	0.031	0.919	0.036	0.933	0.030
0.504	4	0.979	0.002	0.858	0.016	0.910	0.010	0.954	0.005	0.674	0.037	0.985	0.002
0.504	9.75	0.977	0.009	0.777	0.086	0.916	0.032	0.952	0.018	0.895	0.041	0.975	0.010
0.604	11	0.932	0.017	0.863	0.035	0.874	0.032	0.919	0.021	0.853	0.038	0.887	0.029
0.604	6.25	0.987	0.003	0.900	0.022	0.913	0.019	0.993	0.002	0.867	0.029	0.996	0.001
0.604	9.25	0.957	0.009	0.912	0.019	0.953	0.010	0.964	0.008	0.912	0.019	0.938	0.013
0.704	11.25	0.851	0.033	0.835	0.037	0.849	0.033	0.866	0.030	0.892	0.024	0.885	0.025
0.704	5.75	0.976	0.005	0.896	0.021	0.940	0.012	0.971	0.006	0.922	0.016	0.972	0.006
0.704	9	0.971	0.006	0.922	0.016	0.954	0.009	0.973	0.005	0.952	0.010	0.972	0.006
0.804	10.25	0.905	0.015	0.891	0.018	0.891	0.018	0.911	0.014	0.888	0.018	0.892	0.017
0.804	5.5	0.971	0.006	0.889	0.022	0.882	0.023	0.955	0.009	0.884	0.023	0.930	0.014
0.804	9	0.956	0.009	0.925	0.015	0.899	0.020	0.959	0.008	0.938	0.012	0.961	0.008
0.824	11.25	0.923	0.014	0.901	0.018	0.915	0.015	0.916	0.015	0.916	0.015	0.936	0.012
0.824	4.75	0.968	0.006	0.915	0.015	0.924	0.014	0.952	0.009	0.836	0.030	0.953	0.009
0.824	9.5	0.910	0.015	0.941	0.010	0.913	0.014	0.952	0.008	0.913	0.014	0.936	0.010
0.844	10.25	0.964	0.006	0.921	0.013	0.941	0.009	0.923	0.012	0.894	0.017	0.957	0.007
0.844	6.25	0.976	0.003	0.951	0.006	0.936	0.008	0.978	0.003	0.834	0.022	0.974	0.003
0.844	0	0.907	0.015	0.642	0.057	0.849	0.024	0.896	0.017	0.855	0.023	0.916	0.013
0.864	10.75	0.981	0.003	0.932	0.011	0.934	0.011	0.961	0.006	0.942	0.009	0.971	0.005
0.864	4.75	0.990	0.001	0.957	0.006	0.967	0.005	0.988	0.002	0.963	0.005	0.991	0.001
0.864	9.5	0.976	0.004	0.943	0.009	0.932	0.011	0.964	0.006	0.949	0.008	0.971	0.005
0.884	11.25	0.976	0.004	0.950	0.009	0.947	0.009	0.974	0.004	0.914	0.015	0.967	0.006
0.884	5.25	0.991	0.001	0.960	0.006	0.972	0.004	0.992	0.001	0.881	0.017	0.994	0.001
0.884	9.75	0.977	0.004	0.958	0.007	0.936	0.010	0.982	0.003	0.964	0.006	0.982	0.003
0.924	10.75	0.990	0.002	0.921	0.017	0.945	0.012	0.981	0.004	0.969	0.006	0.984	0.003
0.924	7.5	0.991	0.001	0.940	0.009	0.911	0.014	0.981	0.003	0.914	0.013	0.987	0.002
0.924	9.75	0.992	0.002	0.951	0.009	0.957	0.008	0.989	0.002	0.955	0.008	0.986	0.003
0.944	10.5	0.990	0.002	0.925	0.017	0.981	0.004	0.994	0.001	0.964	0.008	0.987	0.003
0.944	8.5	0.986	0.003	0.930	0.013	0.963	0.007	0.950	0.010	0.886	0.022	0.974	0.005
0.944	10	0.996	0.001	0.925	0.016	0.963	0.008	0.991	0.002	0.947	0.011	0.978	0.005
0.964	10.75	1.000	0.000	0.869	0.039	0.949	0.015	0.996	0.001	0.986	0.004	0.999	0.000
0.964	15	0.993	0.003	0.835	0.060	0.933	0.024	0.972	0.010	0.994	0.002	0.996	0.002
0.964	0	0.703	0.043	0.474	0.077	0.629	0.054	0.702	0.044	-0.822	0.267	0.861	0.020
0.904	11	0.966	0.007	0.952	0.010	0.894	0.022	0.980	0.004	0.948	0.011	0.976	0.005
0.904	6	0.973	0.003	0.932	0.008	0.925	0.009	0.954	0.006	0.879	0.015	0.974	0.003
0.904	9.75	0.985	0.003	0.963	0.006	0.933	0.012	0.983	0.003	0.966	0.006	0.991	0.002

Table 2 – Performance metrics of the surrogate model: the correlation coefficient R^2 and the Mean Square Error MSE. Results for each independent case of the *validation* set.

References

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

[2] L. Sirovich. Turbulence and the dynamics of coherent structures. i, ii, iii. *Quarterly of applied mathematics*, 45(3):561–590, 1987.

[3] Jonathon Shlens. A tutorial on principal component analysis, 2014.

[4] B. R. Noack, K. Afanasiev, M. Morzyński, G. Tadmor, and F. Thiele. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *Journal of Fluid Mechanics*, 497:335–363, 2003.

[5] H. V. Ly and H. T. Tran. Modeling and control of physical processes using proper orthogonal decomposition. *Mathematical and Computer Modelling*, 33(1):223–236, 2001. Computation and control VI proceedings of the sixth Bozeman conference.

[6] R. Zimmermann and S. Görtz. Improved extrapolation of steady turbulent aerodynamics using a non-linear pod-based reduced order model. *The Aeronautical Journal (1968)*, 116(1184):1079–1100, 2012.

[7] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The gnat method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242:623–647, 2013.

[8] S. Chaturantabut and D. C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[9] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[10] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[11] F. Tauro, S. Grimaldi, and M. Porfiri. Unraveling flow patterns through nonlinear manifold learning. *PLOS One*, 9(3):1–6, 03 2014.

[12] G. Bansal, A. A. Mascarenhas, and J. H. Chen. Identification of intrinsic low dimensional manifolds in turbulent combustion using an isomap based technique. Technical report, Sandia National Lab.(SNL-CA), Livermore, CA (United States), 2011.

[13] E. Farzamnik, A. Ianiro, S. Discetti, N. Deng, K. Oberleithner, B. R. Noack, and V. Guerreo. From snapshots to manifolds - a tale of shear flows. 2022. in submission process.

[14] T. Franz, R. Zimmermann, S. Görtz, and N. Karcher. Interpolation-based reduced-order modelling for steady transonic flows via manifold learning. *International Journal of Computational Fluid Dynamics*, 28(3-4):106–121, 2014.

[15] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[16] A.K. Jain, J. Mao, and K.M. Mohiuddin. Artificial neural networks: a tutorial. *Computer*, 29(3):31–44, 1996.

[17] X. Du, P. He, and J. R. R. A. Martins. Rapid airfoil design optimization via neural networks-based parameterization and surrogate modeling. *Aerospace Science and Technology*, 113:106701, 2021.

[18] R. Castellanos, G. Y. Cornejo Maceda, I. de la Fuente, B. R. Noack, A. Ianiro, and S. Discetti. Machine-learning flow control with few sensor feedback and measurement noise. *Physics of Fluids*, 34(4):047118, 2022.

[19] B. Glaz, L. Liu, and P. P. Friedmann. Reduced-Order Nonlinear Unsteady Aerodynamic Modeling Using a Surrogate-Based Recurrence Framework. *AIAA Journal*, 48(10):2418–2429, 2010.

[20] E. Andrés-Pérez and C. Paulete-Periáñez. On the application of surrogate regression models for aerodynamic coefficient prediction. *Complex & Intelligent Systems*, 7(4):1991–2021, 2021.

[21] N. Kroll, S. Langer, and A. Schwöppe. *The DLR Flow Solver TAU - Status and Recent Algorithmic Developments*. 2014.

[22] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[24] R. W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[25] A. Kumar and A. K. Ghosh. Decision tree–and random forest–based novel unsteady aerodynamics modeling using flight data. *Journal of Aircraft*, 56(1):403–409, 2019.

[26] N. Umetani and B. Bickel. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics (TOG)*, 37(4):1–10, 2018.

[27] P. Probst, M. N. Wright, and A. L. Boulesteix. Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3):e1301, 2019.

[28] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[29] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[30] O. Samko, A.D. Marshall, and P.L. Rosin. Selection of the optimal parameter value for the isomap algorithm. *Pattern Recognition Letters*, 27(9):968–979, 2006.

[31] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama. Optuna: A next-generation hyperparameter optimization framework, 2019.