Complex & Intelligent Systems (2021) 7:1991–2021
https://doi.org/10.1007/s40747-021-00307-y

ORIGINAL ARTICLE

On the application of surrogate regression models for aerodynamic coefficient prediction

Esther Andrés-Pérez[1] · Carlos Paulete-Periáñez[1]

Received: 15 October 2020 / Accepted: 16 February 2021 / Published online: 10 March 2021

Abstract

Computational fluid dynamics (CFD) simulations are nowadays been intensively used in aeronautical industries to analyse the aerodynamic performance of different aircraft configurations within a design process. These simulations allow to reduce time and cost compared to wind tunnel experiments or flight tests. However, for complex configurations, CFD simulations may still take several hours using high-performance computers to deliver results. For this reason, surrogate models are currently starting to be considered as a substitute of the CFD tool with a reasonable prediction. This paper presents a review on surrogate regression models for aerodynamic coefficient prediction, in particular for the prediction of lift and drag coefficients. To compare the behaviour of the regression models, three different aeronautical configurations have been used, a NACA0012 airfoil, a RAE2822 airfoil and 3D DPW wing. These databases are also freely provided to the scientific community to allow other researchers to make further comparison with other methods.

Keywords Machine learning · Aerodynamic analysis · Computational fluid dynamics · Surrogate modelling · Regression · Support vector machines for regression

Introduction

The aeronautical sector is in a continuous fierce competition to shorten time-to-market cycles and to offer greener and more efficient aircraft. This is because aircraft manufacturers want to keep their industrial leadership, but also due to the targets that governments impose to the sector. For instance, in Europe, the High Level Group on Aviation Research (HLGAR) provided several guidelines stated at the ACARE (Advisory Council for Aeronautics Research in Europe) 2050 [1] flight path. To advance in the design of better aircraft in reduced times, it is required to introduce new tools and technologies in the design process. In particular, the CFD simulations of complex configurations usually require several hours using hundreds of processors. In an earlier stage of the design process, computational fluid dynamics (CFD) simulations could be substituted by surrogate models able to predict the aerodynamic performance within reasonable precision margins. In the last few years, as will be detailed in next section, there have been several applications of certain surrogate models to predict pressure curves or global aerodynamic coefficients, such as lift or drag, of aeronautical configurations. However, to our knowledge, there is not an integrated comparison of several surrogate models, as the one proposed in this paper, using the same aircraft configurations and databases, so conclusions on the performance of each model can be extracted.

Therefore, the objective of this paper is to make a comparison of different surrogate regression models for aerodynamic coefficient prediction in different aeronautical configurations. In particular, three aeronautical configurations have been used, a NACA0012 airfoil, a RAE2822 airfoil and 3D DPW wing from the AIAA (American Institute of Aeronautics and Astronautics) Drag Prediction Workshop.

The paper is structured as follows: "Brief review of the state-of-the-art" presents a review of the state-of-the-art, focusing on surrogate modelling applications for aerodynamic analysis and design. "Surrogate modelling approaches for regression" theoretically describes the methods to be

✉ Esther Andrés-Pérez
eandper@inta.es

Carlos Paulete-Periáñez
pauletepc@inta.es

[1] Theoretical and Computational Aerodynamics Branch, Aircraft Department, Spanish National Institute for Aerospace Technology (INTA), Ctra. Ajalvir, km. 4, 28850 Torrejón de Ardoz, Spain

compared. "Numerical results" shows the numerical results and finally, "Conclusions and future research" presents the conclusions. As annexes at the end of this paper, the complete databases information is provided.

Brief review of the state-of-the-art

Generally speaking, surrogate modelling refers to a group of techniques that make use of previously obtained sampled data to build surrogate models, which are subsequently used to predict the value of variables at new points in the design space. The use of machine learning methods to make predictions is not new in markets such as finances or insurance. For instance, more than 10 years ago, it is possible to find applications such as [2] where authors explore the performance of credit scoring using two data mining techniques, classification and regression tree (CART) and multivariate adaptive regression splines (MARS). In [3] it is performed a comparative study of prediction performances of an artificial neutral network (ANN) model against a linear prediction model like a linear discriminant analysis (LDA) with regards to forecasting corporate credit ratings from financial statement data. In addition, other scientific publications [4–6] focused on stock market prediction using ANNs.

In the aeronautical sector, there have been also applications of surrogate modelling techniques mainly for aerodynamic analysis and optimization. The application to the aerospace field, and particularly, to aerodynamic data prediction based on CFD, wind tunnel and flight testing data, can allow a first stage exploration of new areas in the design space, without the need of expensive simulations, wind tunnel or flight testing, and in this way, reduce the number of required experiments. For instance, with respect to the so called physics models, Kriging [7–10] and co-Kriging [11–13] based models have been applied to multi-objective optimization or multi-disciplinary optimization problems, including also uncertainties management and quantification as in [14, 15].

On the other hand, within the machine learning field, there have been also some application of models based on ANNs or support vector machines (SVMs) for aerodynamic coefficient predictions [16–18], aerodynamic design [19–22] and uncertainty quantification or/and robust design [23, 24]. Other supervised learning methods, such as Bayesian automatic relevance determination (ARD) regression or Bayesian ridge have been applied in [25–27] mainly for aerodynamic design and optimization and decision tree-based models have been used in [28, 29].

More recently, deep learning techniques have been also applied to the wing airfoil pressure calibration in [30], and to a multi-fidelity surrogate-based robust optimization framework in [31]. In addition, approximation models based on convolutional neural networks (CNNs) are proposed for flow field predictions in [32–35].

In summary, machine learning entails powerful information processing algorithms that are relevant for modelling, optimization, and control of fluids. Currently, machine learning capabilities are advancing at an incredible rate, and fluid mechanics is beginning to tap into the full potential of these powerful methods. Many tasks in fluid mechanics, such as reduced-order modelling, shape optimization, uncertainty quantification, and feedback control, may be posed as optimization and regression tasks. Machine learning can dramatically improve optimization performance and reduce convergence time. Machine learning is also used for dimensionality reduction, identifying low-dimensional manifolds and discrete flow regimes, which benefit understanding.

This explains why, in the last few years, there have been an increasing number of applications of surrogate models to predict pressure curves or global aerodynamic coefficients, such as lift or drag, of aeronautical configurations. However, to our knowledge, there is not an integrated comparison of several surrogate models using the same aircraft configurations and databases, so conclusions on the performance of each model can be extracted.

Therefore, the objective of this paper is to make a deep comparison of different surrogate regression models for aerodynamic coefficient prediction in different aeronautical configurations. In particular, three different aeronautical configurations have been used, a NACA0012 airfoil, a RAE2822 airfoil and 3D DPW wing.

The novelty of this work is on the application of surrogate regression models for the aerodynamic coefficients prediction of aeronautical configurations. Although the regression models applied in this paper already exist and have been applied in other sectors such as finances or risk analysis, its application in aerodynamics is still in its infancy. The importance of this research resides on the high computational cost of the computational fluid dynamic simulations. If this computational cost could be reduced using the proposed regression models, it would be possible to speed up the design process of new aircraft configurations, and, moreover, to consider also unconventional aircraft configurations since it would be feasible to consider a high number of design parameters, which is not possible nowadays. However, the success of these techniques in the aeronautical sector and, in particular, in computational aerodynamics, is still not clear and requires further research. If machine learning methods could be successfully used to substitute computational fluid dynamics tools for aerodynamic simulations it would constitute a huge improvement for the aeronautical industry which could use these methods to obtain fast predictions of aircraft or components aerodynamics features and, therefore, speed up the time-to-market of their products. In addition, these methods have also a great potential to exploit aerodynamic data already existing at industries, as for instance from previous simulations, wind tunnel experiments or even flight testing. Therefore, it is worth to investigate the feasibility of these methods for the aeronautical sector and in particular, for aerodynamic prediction.

Surrogate modelling approaches for regression

This section theoretically describes the methods that will be compared in this paper.

Linear models

Linear models are one of the simplest regression algorithms available to data scientists. They predict data based on a linear relationship with the features, such that:

$$\hat{y}(\vec{w},\vec{x}) = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n, \tag{1}$$

where $\hat{y}$ is the predicted value, x_i are the features of the dataset and w_i are the coefficients of the linear regression ($i=1$, 2…, n). All linear models share this characteristic; however, differences appear in how the coefficients are obtained from the labelled data. In the following paragraphs, some of the most prevalent and used methods within this category will be explained.

Least squares

This method involves fitting the training data to the model by minimizing the sum of the squares of the error of the prediction with respect to the training label, that is:

$$\min_{w} \left|\left|X\vec{w} - \vec{y}\right|\right|_2^2. \tag{2}$$

With X being the matrix built by the vectors of features, w is the vector of coefficients and y is the training label. Although simple, this objective function poses difficulties when the features of the data to fit are correlated. In this case, the design matrix becomes singular or close to singular and the model becomes highly sensitive to errors in the observed label.

Ridge regression

Ridge regression [36] caters to the problems encountered with multicollinearity in least squares. It does so by introducing a parameter α that penalizes the absolute value of the coefficients. The new objective function is:

$$\min_{w}(\left|\left|X\vec{w} - \vec{y}\right|\right|_2^2 + \alpha\left|\left|\vec{w}\right|\right|_2^2). \tag{3}$$

The parameter α will reduce the value of the resulting coefficients as it increases, augmenting the robustness against correlated features.

LASSO (least absolute shrinkage and selection operator)

Least absolute shrinkage and selection operator [37] is mathematically, a linear model with an added regularization term, in such a way that the objective function becomes:

$$\min_{w} \frac{1}{2n}\left|\left|X\vec{w} - \vec{y}\right|\right|_2^2 + \alpha\left|\left|\vec{w}\right|\right|_1. \tag{4}$$

The l_1 norm that appears in the objective function makes it so that less non-zero coefficients appear in the final model, by automatically performing feature selection on the dataset combined with a regularization, as seen in ridge regression. Due to the feature selection strategy of the model, LASSO gives further insight into the dataset compared to other similar models, which can be useful when, for instance, the dataset is small and there is model accuracy to be gained by removing non-relevant features to simplify the problem.

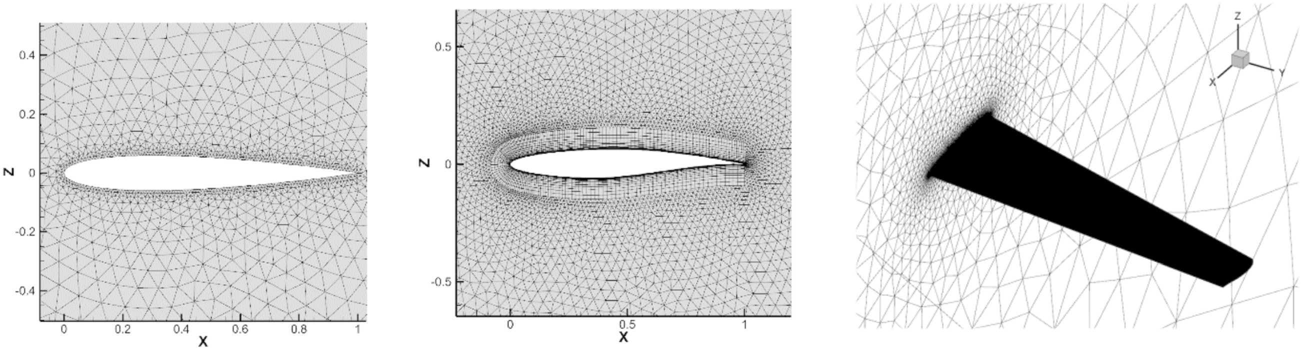


Fig. 1 NACA0012, RAE2822 and DPW grids

LARS (least angle regression)

Least angle regression [38] takes a different approach to other linear methods covered here, as a linear model, it fits a linear combination of coefficients and features to a certain label; however, LARS does so using an iterative algorithm.

Initially, all the coefficients are set to zero, and an analysis is carried out to determine which coefficient is the most correlated with the labels. Then, this coefficient is increased along the slope given by its correlation until some other coefficient has as much correlation with the residual. At that point, one increases both coefficients in their combined least squares direction until there is another coefficient with as much correlation with the residual as the pair. This process is repeated until all the coefficients are included.

LARS is particularly useful when operating with high-dimensional small datasets, efficient and fast to run on high-dimensional small datasets, however, it is particularly sensitive to noise.

Bayesian ridge

Bayesian regression [39], contrary to least squares for instance, does not assume that there is an optimal set of coefficients to satisfy the linear relationship, instead, it yields a posteriori distribution of the model parameters. In this way, a priori knowledge of the coefficients can be included in the model to make a better estimator.

Bayesian ridge, is a Bayesian implementation of the ridge model described in this section. For this model, the prior of the coefficients w are given by a spherical Gaussian as:

$$p(w|\lambda) = \mathcal{N}(w|0, \lambda^{-1}I_{\mathrm{p}}), \tag{5}$$

where the priors w and λ are chosen as gamma distribution. Bayesian ridge models tend to produce very similar results

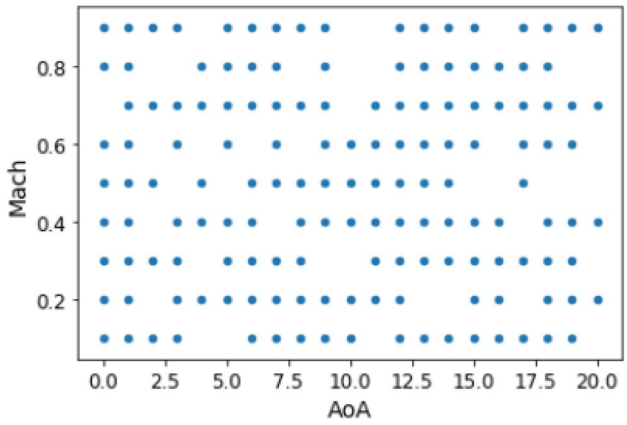


Fig. 2 Exploring NACA0012 database (Mach versus AoA distribution of the database samples)

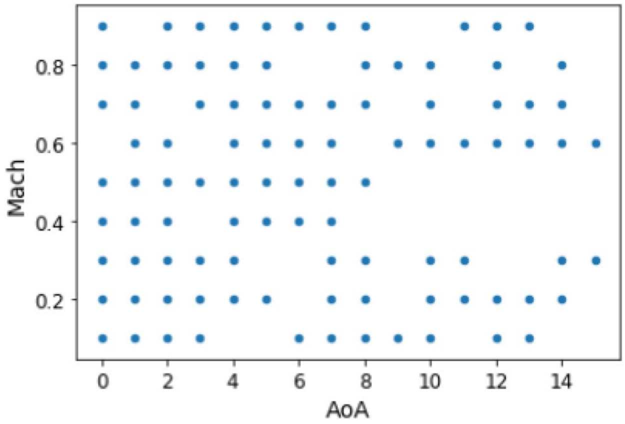


Fig. 4 Exploring RAE2822 database (Mach versus AoA distribution of the database samples)

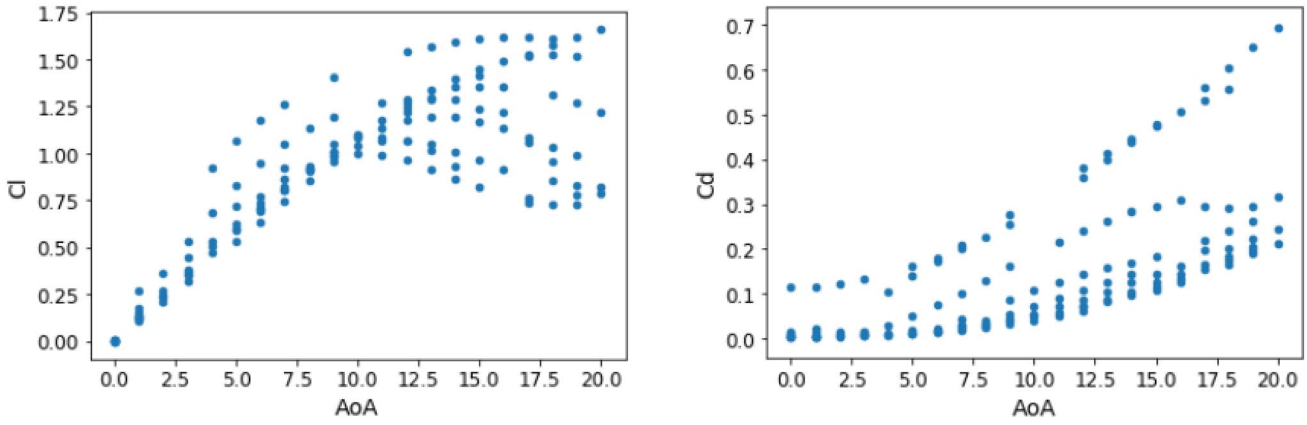


Fig. 3 Exploring NACA0012 database (left: representation of the database samples in the lift-AoA space, right: representation of the database samples in the drag-AoA space)

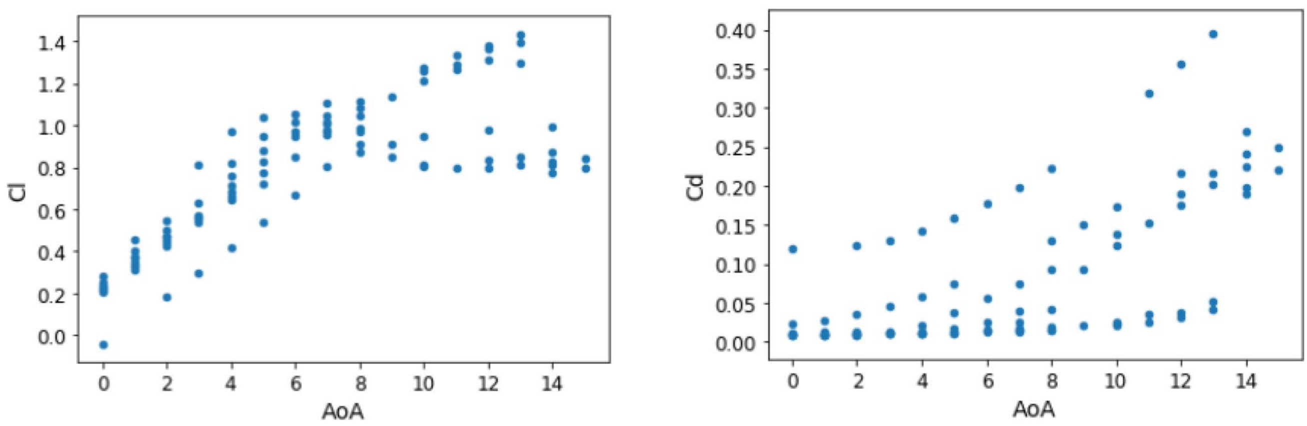


Fig. 5 Exploring RAE2822 database (left: representation of the database samples in the lift-AoA space, right: representation of the database samples in the drag-AoA space)

to ridge models; however, they also tend to be more robust in cases when limited data is available. Moreover, this model can incorporate prior knowledge of the system and compute the uncertainty associated with it.

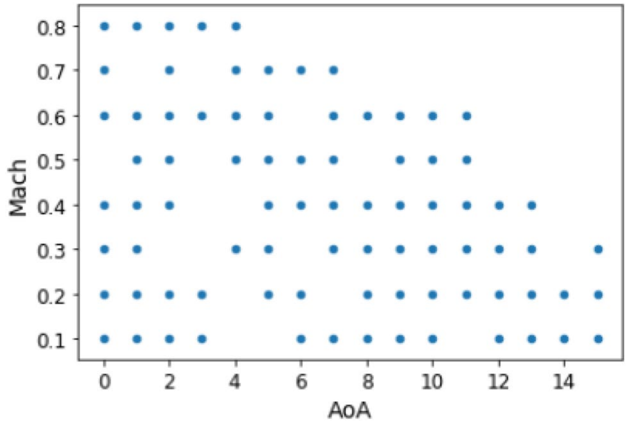


Fig. 6 Exploring DPW database (Mach versus AoA distribution of the database samples)

Huber regression

Huber regression [40] implements a ridge model with the l_2 norm regularization parameter but introducing a different loss for outliers. The objective function then becomes:

$$\min_{w,\sigma} \sum_{i=1}^{n} \left(\sigma + H_{\epsilon}\left(\frac{X_i w - y_i}{\sigma}\right)\sigma + \alpha||w||_2^2\right), \tag{6}$$

with

$$H_{\epsilon}(z) = \begin{cases} z\text{\^{}}2, & |z| < \epsilon \\ 2\epsilon|z| - \epsilon^2, & \text{otherwise} \end{cases},$$

where σ is a scaling parameter to be optimized as well.

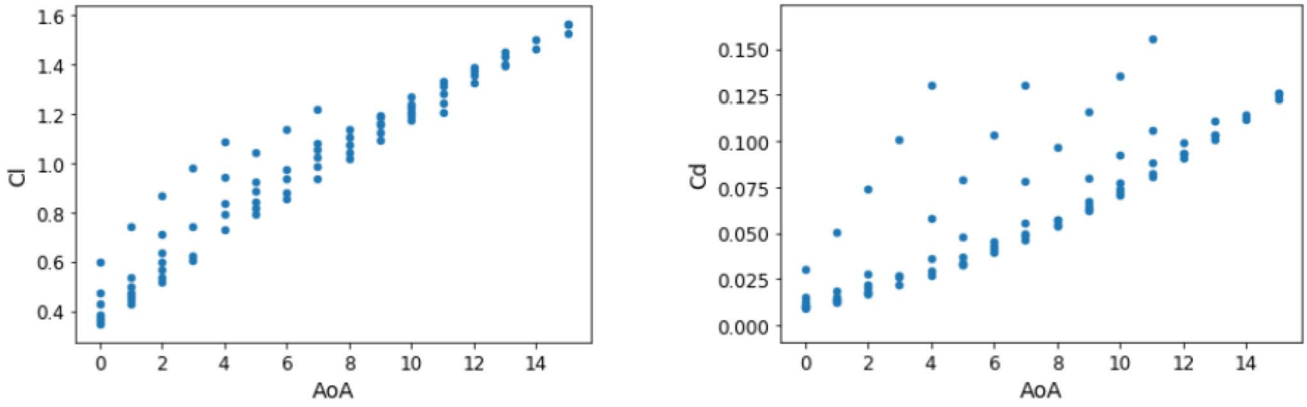


Fig. 7 Exploring DPW database (left: representation of the database samples in the lift-AoA space, right: representation of the database samples in the drag-AoA space)

Huber regression brings outlier tolerance to the ridge model, by introducing a linear loss instead of a quadratic one to reduce the effect of the outliers. A sample is classified as an outlier if the absolute error associated with it has a value larger than a parameter ϵ the smaller this parameter is, the more robust the model is to outliers.

ARD (automatic relevance determination) regression

Automatic relevance determination regression [39] is similar to a Bayesian ridge regression with some modifications to the prior introduced for that model. For ARD, the assumption of w as a spherical Gaussian is dropped, instead, it is assumed an axis parallel, elliptical Gaussian distribution. Then

$$p(w|\lambda) = \mathcal{N}(w|0, A^{-1}), \tag{7}$$

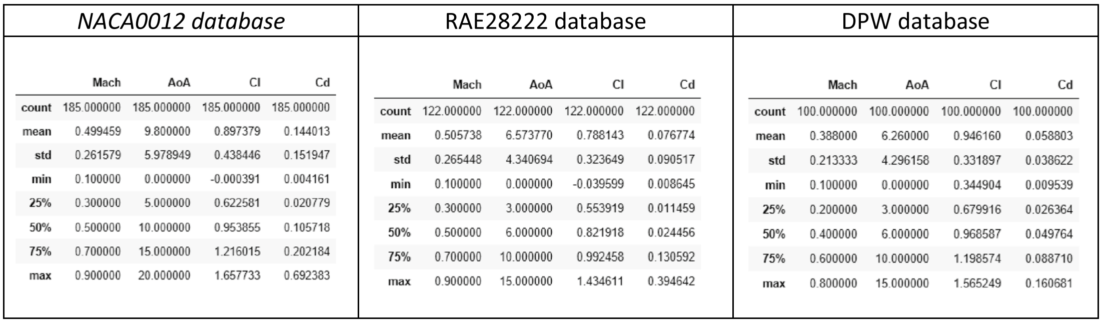
where A is a diagonal matrix of coefficients λ_i, therefore, every coefficient w has its own standard deviation contrary to Bayesian ridge. ARD regression in practice, leads to sparser coefficients than Bayesian ridge.

Decision trees

Decision trees [41] are supervised learning methods that can be used for classification and regression. They attempt to fit the data by devising a set of decision rules on its features. They can be used for classification and regression problems, modifying only the output data type.

Intuitively, they can be thought of as a series of splits on the features of the dataset, for instance, when classifying a vehicle dataset based on the number of wheels, an initial split may be, if #Wheels < 3 → Motorcycle. On the else part of this if statement, one may include #Wheels < 5 → Car

Table 1 Statistics of the aerodynamic databases



	NACA0012 database				RAE28222 database				DPW database			
	Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd
count	185.000000	185.000000	185.000000	185.000000	122.000000	122.000000	122.000000	122.000000	100.000000	100.000000	100.000000	100.000000
mean	0.499459	9.800000	0.897379	0.144013	0.505738	6.573770	0.788143	0.076774	0.388000	6.260000	0.946160	0.058803
std	0.261579	5.978949	0.438446	0.151947	0.265448	4.340694	0.323649	0.090517	0.213333	4.296158	0.331897	0.038622
min	0.100000	0.000000	-0.000391	0.004161	0.100000	0.000000	-0.039599	0.008645	0.100000	0.000000	0.344904	0.009539
25%	0.300000	5.000000	0.622581	0.020779	0.300000	3.000000	0.553919	0.011459	0.200000	3.000000	0.679916	0.026364
50%	0.500000	10.000000	0.953855	0.105718	0.500000	6.000000	0.821918	0.024456	0.400000	6.000000	0.968587	0.049764
75%	0.700000	15.000000	1.216015	0.202184	0.700000	10.000000	0.992458	0.130592	0.600000	10.000000	1.198574	0.088710
max	0.900000	20.000000	1.657733	0.692383	0.900000	15.000000	1.434611	0.394642	0.800000	15.000000	1.565249	0.160681

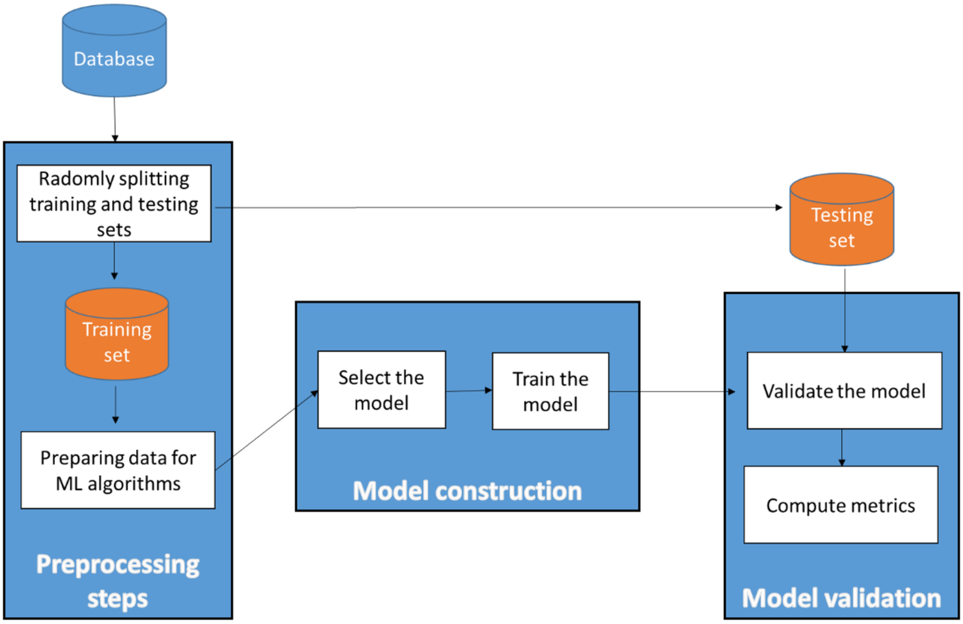


Fig. 8 First step of the followed strategy (full models comparisons with training and testing split)

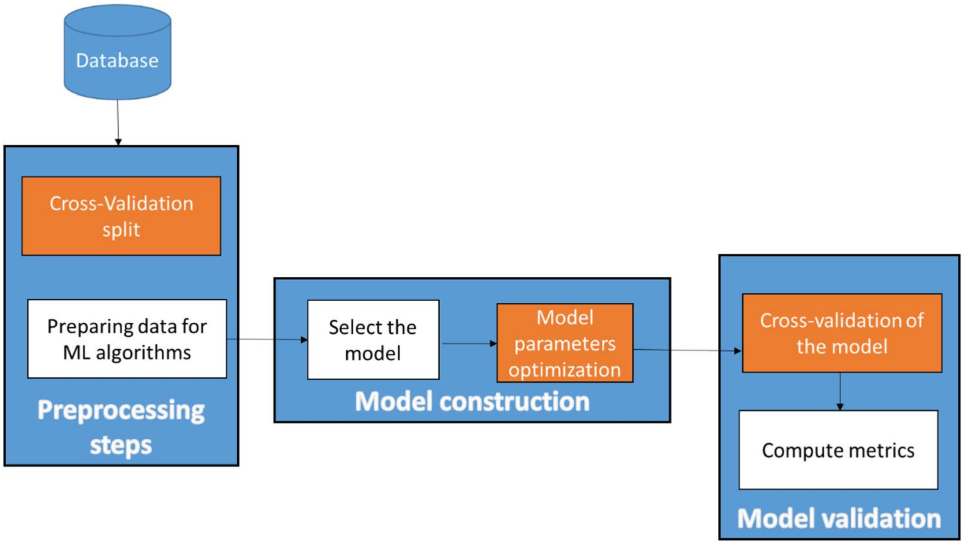


Fig. 9 Second step of the followed strategy (best models comparisons with cross-validation and parameters tuning)

Table 2 Brief description of the specific methods used in the comparison

Method	Description
SVR_rbf	Support vector regression (SVR) using a radial basis function (RBF) kernel
SVR_poly	Support vector regression (SVR) using a polynomial kernel
SVR_linear	Support vector regression (SVR) using a linear kernel
Decision tree	A decision tree regressor
Extra tree	Extra tree method (extremely randomized tree)
Lars	Least angle regression method
Lasso	Lasso regression method
MLP_tanh	Multi-layer perceptron (MLP) regression method with activation equals to the hyperbolic tan function
MLP_id	Multi-layer perceptron (MLP) regression method with activation equals to the identity function
MLP_log	Multi-layer perceptron (MLP) regression method with activation equals to the logistic sigmoid function
MLP_relu	Multi-layer perceptron (MLP) regression method with activation equals to the rectified linear unit function
HuberRegressor	Hurber regression method
ARDRegression	Automatic relevance determination regression (ARDR) method
LinearReg	Linear regression method
BayesianRidge	Bayesian Ridge regression method

Table 3 Brief description of the specific metrics used in the comparison

Metric	Description
R^2	Coefficient of determination: $R^2(y,\hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
MAE	Mean absolute error: $\text{MAE}(y,\hat{y}) = \frac{1}{n_\text{samples}} \sum_{i=0}^{n_\text{samples}-1} \lvert y_i - \hat{y}_i \rvert$
RSME	Root squared mean error: is the square root of the MSE: $\text{MSE}(y,\hat{y}) = \frac{1}{n_\text{samples}} \sum_{i=0}^{n_\text{samples}-1} (y_i - \hat{y}_i)^2$
ME	Maximum error: $\text{Max Error}(y,\hat{y}) = \max(\lvert y_i - \hat{y}_i \rvert)$
EVS	Explained variance score: $\text{explained\_variance}(y,\hat{y}) = 1 - \frac{\text{Var}\{y - \hat{y}\}}{\text{Var}\{y\}}$

and finally anything else may be a truck. The same logic can be applied to regression by assigning numerical values instead of classes to each decision.

One of the main parameters associated with decision trees is the maximum depth. It represents the maximum number of consecutive decisions allowable. This parameter is highly dependent on the size of the dataset and the system complexity. It is important to reach an appropriate depth for the problem at hand, too low depth would yield lower accuracy than acceptable, but also, it is easy to overfit the model by introducing too many splits.

Overfitting is of great concern when working with decision trees. To avoid this issue, usually a tuning process is applied for the model to find the best depth. There are

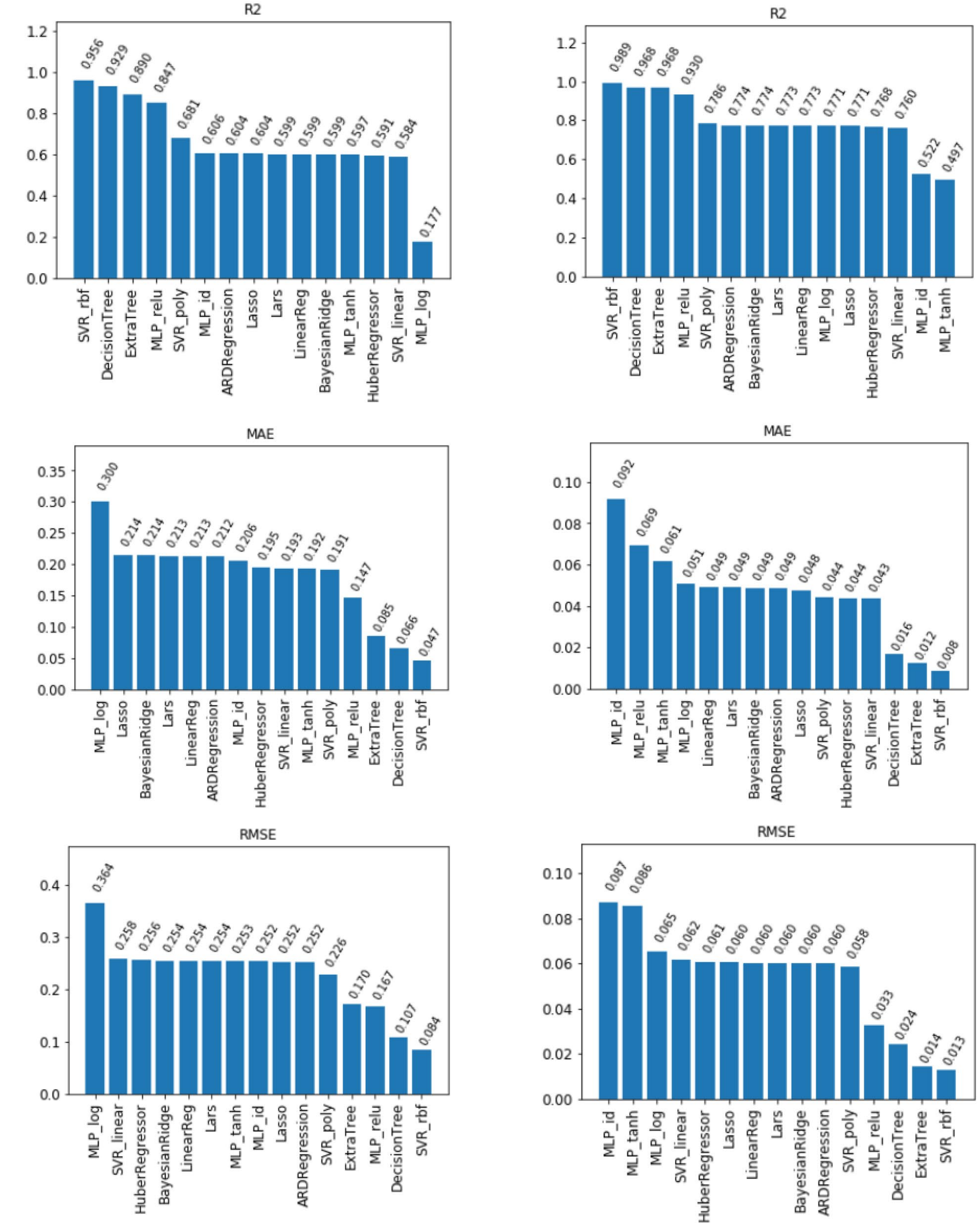


Fig. 10 Models comparison results on the NACA0012 database (left: Cl prediction, right: Cd prediction). Metrics are R^2 (determination coefficient), MAE (mean absolute error), RMSE (root mean squared error), ME (max error) and EVS (explained variance score)

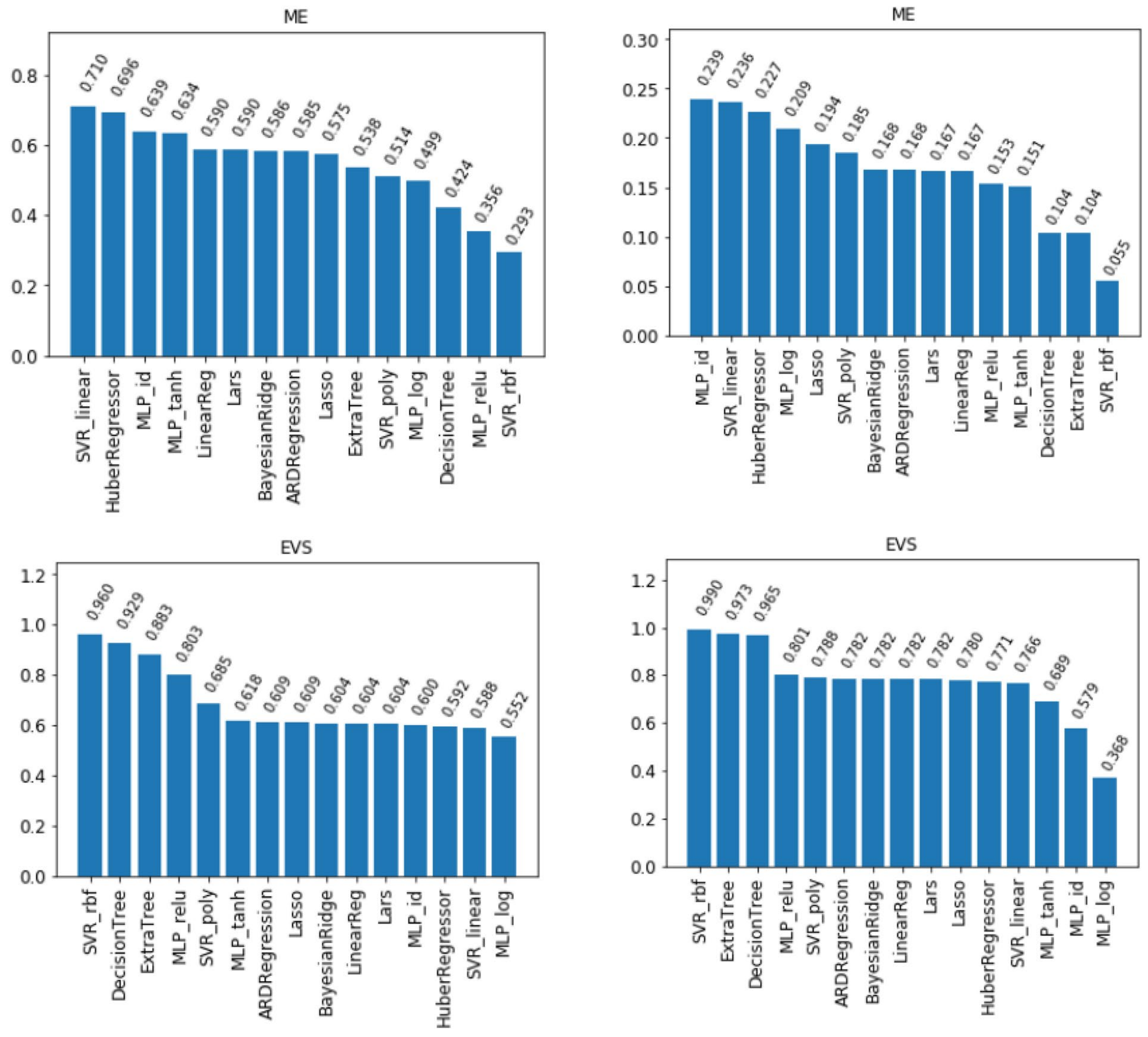


Fig. 10 (continued)

alternative strategies to increase accuracy while avoiding overfitting, these typically fall within the scope of the ensemble learning methods. Ensemble methods employ a combination of models so that the final combination of output is better than any single model by itself reducing the variance. These methods, when applied to decision trees are appropriately called random forests [42] because they combine several trees.

To train a random forest, each tree is trained on a subset of the complete training set and using only a portion of the features available. The reason for using only a subset of the features on any given tree is that there is enough variability between the trees avoiding them being too focused on any given feature in the data. After training each tree, its output is averaged for the output of the forest and a regression problem or a majority voting process is used for classification.

Finally, to introduce even more randomness in the training process to reduce variance, extremely randomized trees or Extra trees [43] can be trained. These differ from random forests in the training process of each tree. Typically, the choice of splitting is decided based on the most discriminative point, however, in an Extra tree model, the splitting rule is chosen as the best from set of randomly generated thresholds. This typically reduces variance at the cost of increasing slightly the bias.

Neural networks

Neural networks have been one of the hot topics in the machine learning community in the recent years and will probably be kept that way for the foreseeable future. They were devised from inspiration of biological neural networks.

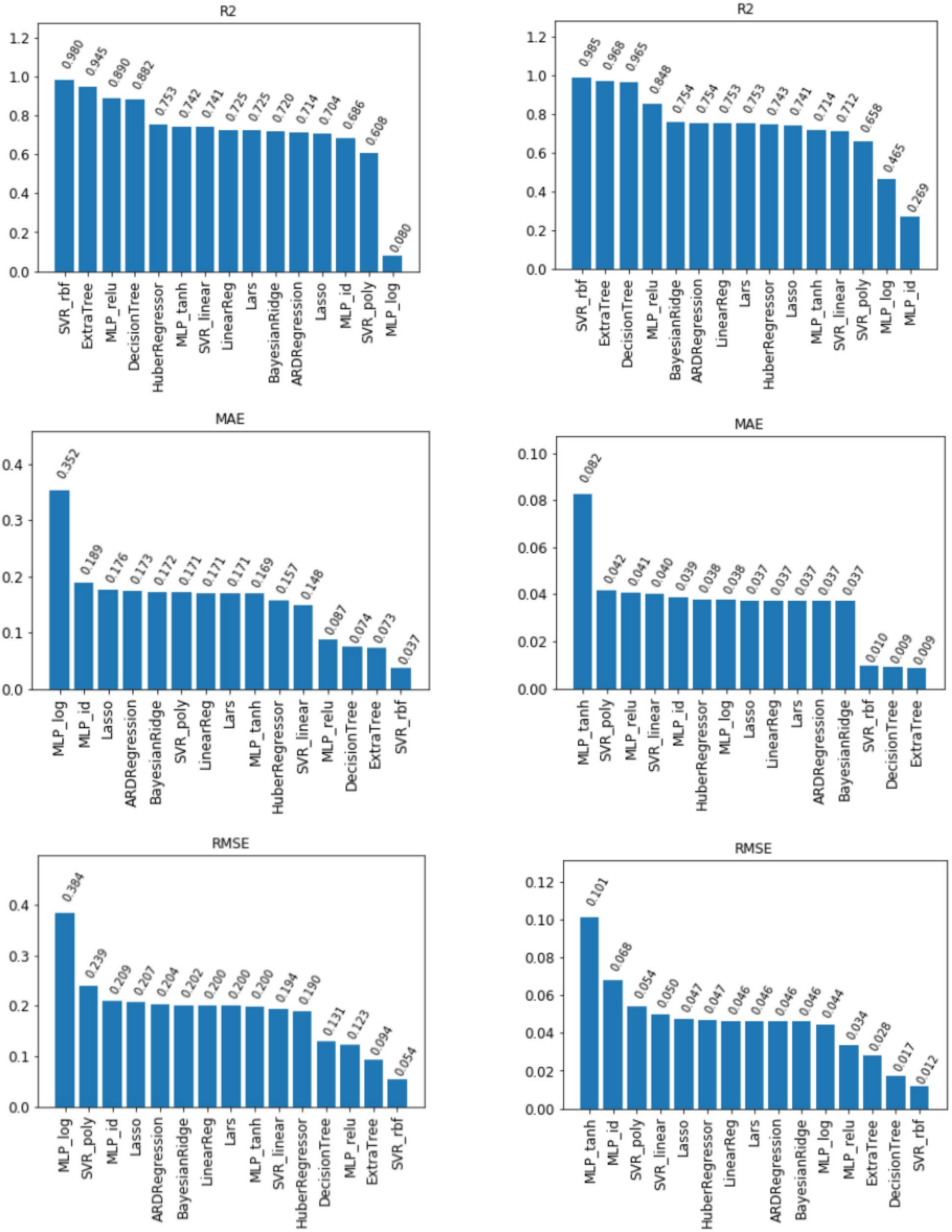


Fig. 11 Models comparison results on the RAE2822 database (left: Cl prediction, right: Cd prediction). Metrics are R^2 (determination coefficient), MAE (mean absolute error), RMSE (root mean squared error), ME (max error) and EVS (explained variance score)

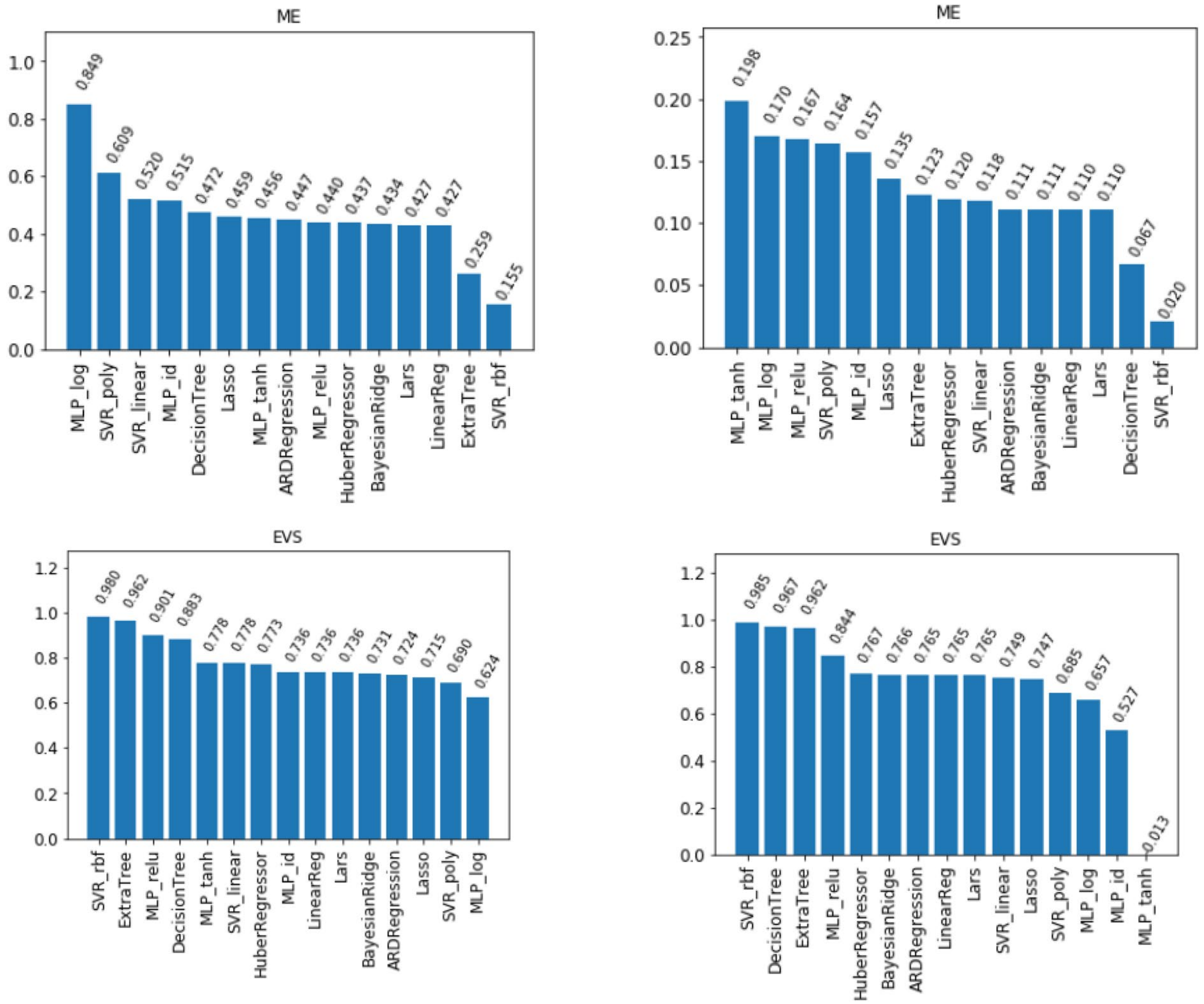


Fig. 11 (continued)

The basic unit within a neural network [44] is what is called a neuron. Neurons are very simple units that are characterized by a set of weights with size equal to the number of inputs to the neuron, during execution, the output of the neuron is the linear combination of the weights and the input. One neuron by itself is not a very powerful estimator; however, the power of the neural networks comes from using many of this simple units creating complex relationships which can extend to very difficult classification or regression problems.

Although there are many configurations for networks of neurons, it is outside the scope of this paper to consider other configurations rather than the simple dense network. In this type of network, the neurons are organized in what is known as layers. The first layer, known as the input layer, receives the features of a sample, and each neuron computes the linear combination of its weights with the feature set. This result is passed on to the next layer of the network for an arbitrary number of layers until the last layer or output layer is reached.

The output layer may perform an additional operation known as activation function. These are used in classification problems to produce categorical data from a real output. Among these functions such as sigmoid or tan*h* can be found.

To train such network, one must define a cost function such as the square error (many others can be used, the cost function is one of the hyperparameters of the model) between the output and the test label. The objective then, is to generate a network that can reduce this error. To that end, one can compute the gradient of the cost function with respect to each weight in the network. Using this, an optimization algorithm may be able to reduce this error.

Finding the weights that minimize the error is not trivial, and there are a wide array of methods and techniques to obtain the best performance possible, from regularization to

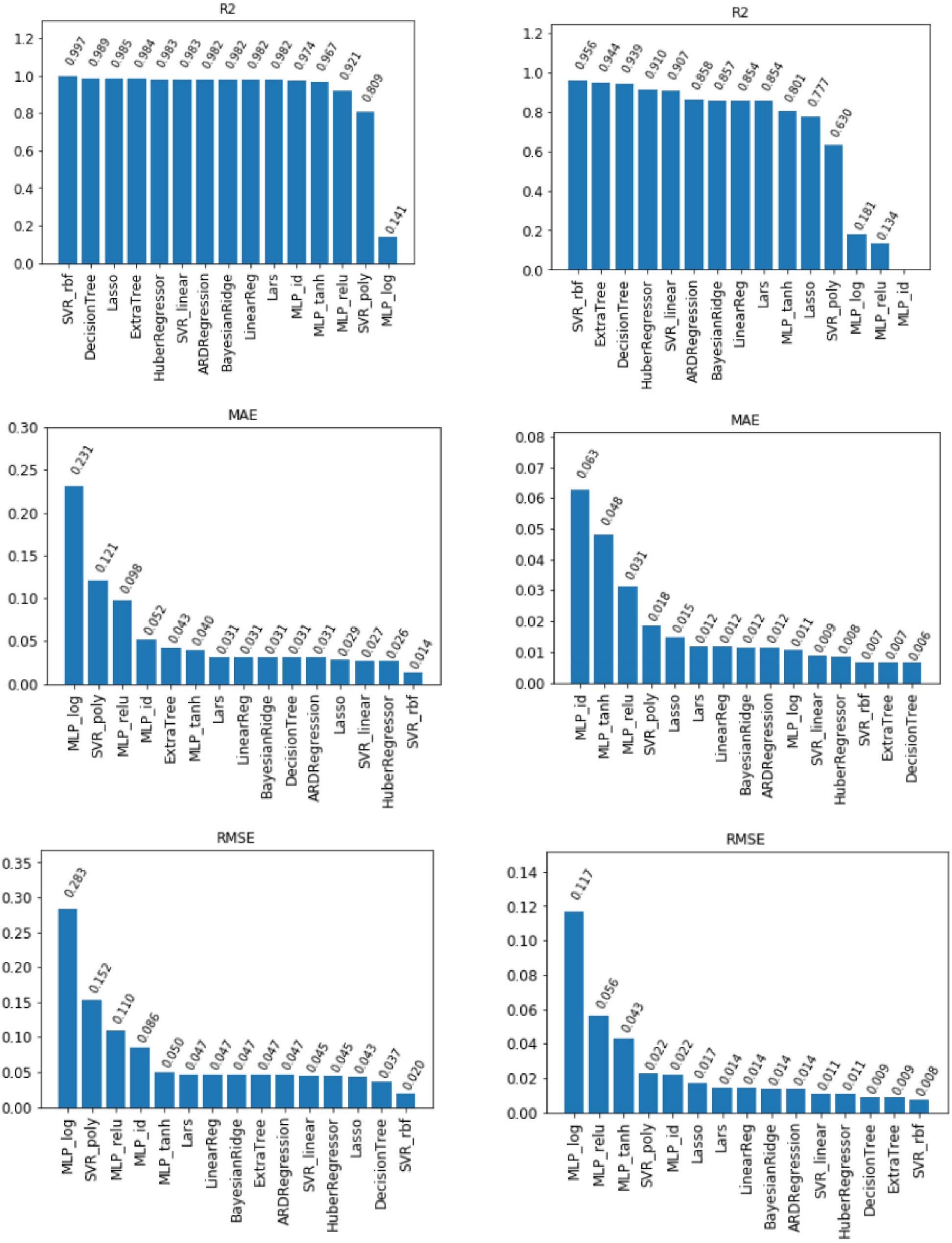


Fig. 12 Models comparison results on the DPW database (left: Cl prediction, right: Cd prediction). Metrics are R^2 (determination coefficient), MAE (mean absolute error), RMSE (root mean squared error), ME (max error) and EVS (explained variance score)

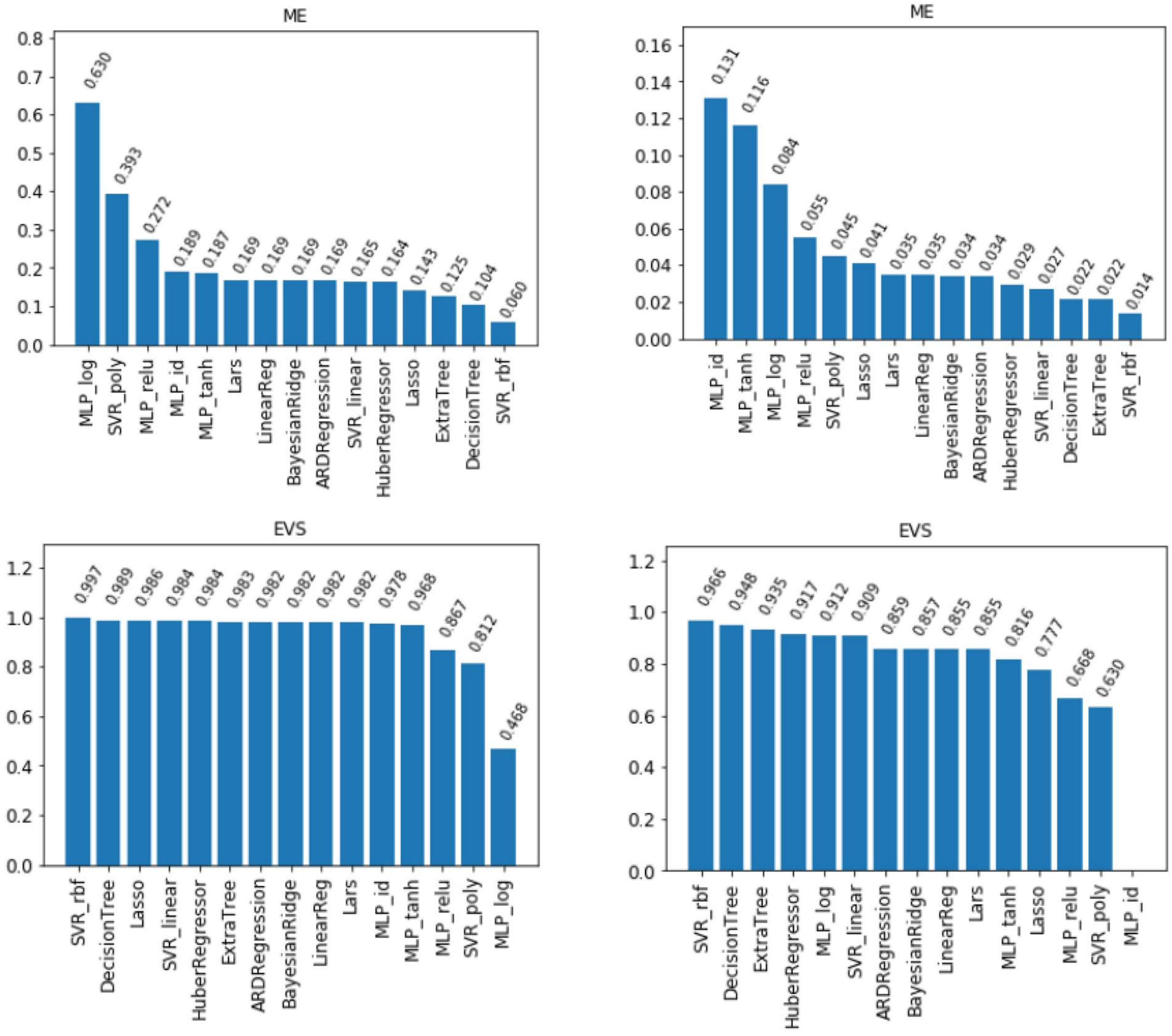


Fig. 12 (continued)

Table 4 Initial model parameters

SVR	‘C’: 1.0 ‘epsilon’: 0.01
Decision tree	‘criterion’: ‘gini’, ‘max_depth’: None, ‘splitter’: ‘best’
Extra tree	‘criterion’: ‘mse’, ‘max_depth’: None,
MLP_relu	‘alpha’: 0.0001, ‘hidden_layer_sizes’: 100, ‘learning_rate’: ‘constant’, ‘learning_rate_init’: 0.001, ‘solver’: ‘adam’, stochastic gradient-based optimizer

initialization algorithms to different optimizers. It becomes apparent that there are a large number of hyperparameters that must be chosen to yield the best possible output.

All in all, neural networks provide a lot of flexibility and complexity compared with other machine learning models. However, they are usually more time consuming and require more expertise from the user to provide the best results.

Support vector machines for regression

Support vector machines for regression (SVMr) are a powerful tool used on the machine learning field, and as a modelling tool in many regression problems in engineering. The SVMr can be solved as a convex optimization problem using kernel theory to face nonlinear problems. The SVMr consider not only the prediction error but also the

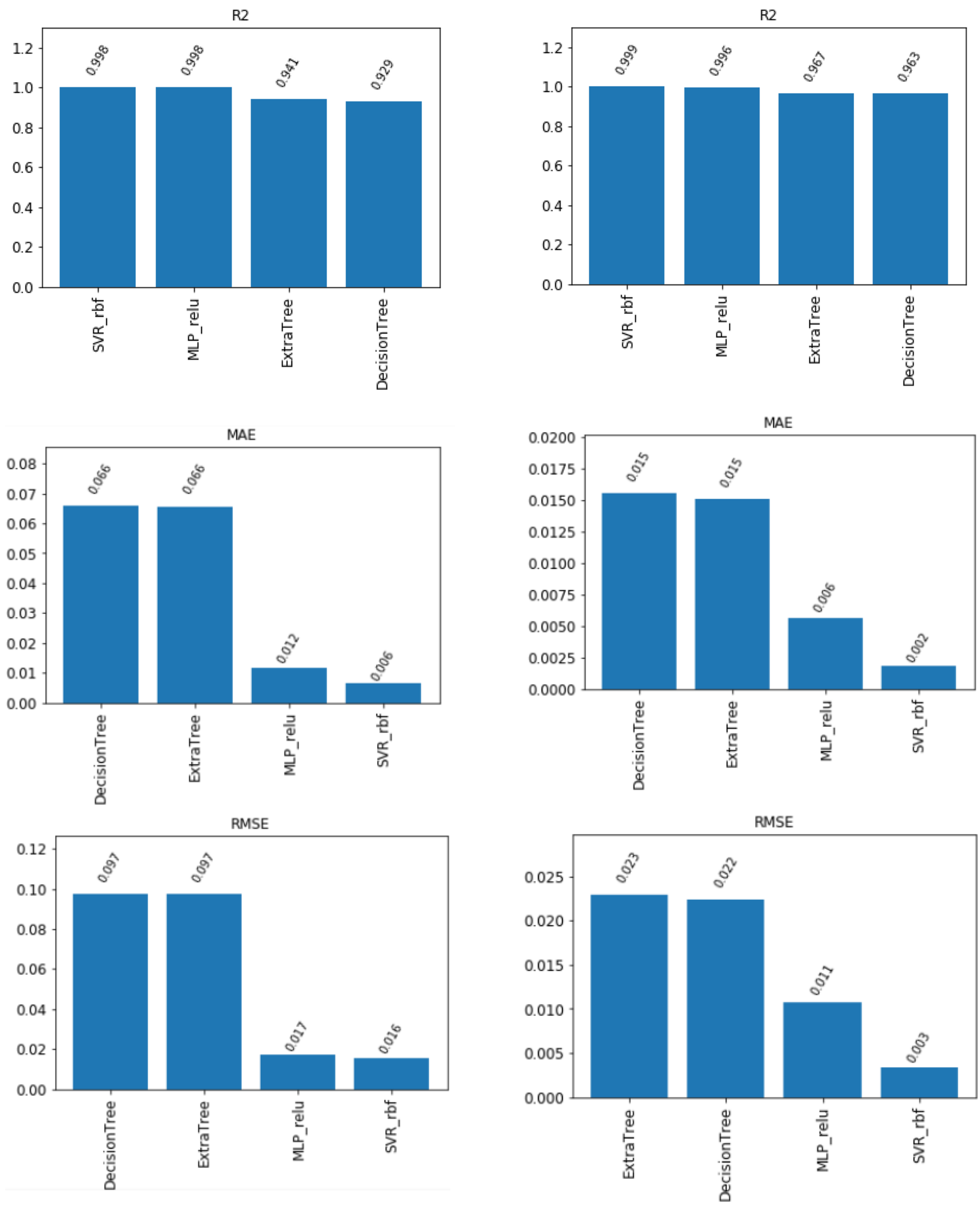


Fig. 13 Best 4 models comparison results on the NACA0012 database (left: Cl prediction, right: Cd prediction). Metrics are R^2 (determination coefficient), MAE (mean absolute error), RMSE (root mean squared error), ME (max error) and EVS (explained variance score)

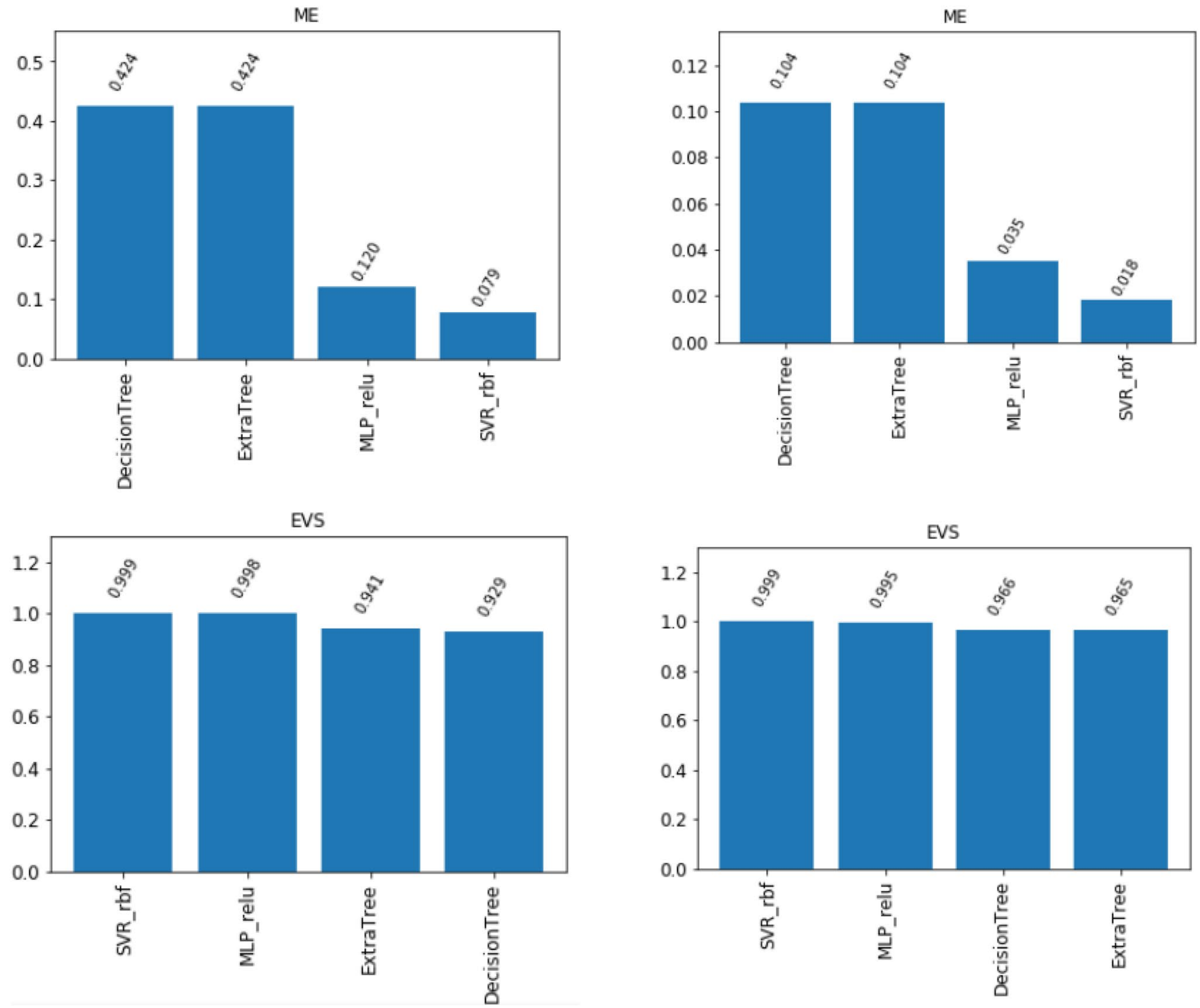


Fig. 13 (continued)

generalization of the model. The SVMr consist of training a model with the form $y = w^{\mathrm{T}}\Phi(x) + b$ given a set of parameters $C = \{(x_i, y_i), i = 1, 2, \ldots, l\}$, to minimize a general risk function of the form:

$$R[f] = \frac{1}{2}\|w\|^2 + \frac{1}{2}C\sum_{i=1}^{l} L(y_i, f(x)), \tag{8}$$

where w controls the smoothness of the model, $\Phi(x)$ is a function of projection of the input space to the feature space, b is a parameter of bias, x_i is a feature vector of the input space with dimension N, y_i is the output value to be estimated and $L(y_i, f(x))$ is the loss function selected. In this paper, the L1-SVR (L1 support vector regression) is used, characterized by an ε-insensitive loss function

Table 5 Models parameters (NACA0012 database)

	Cl predition	Cd prediction
SVR	‘C’: 10,000, ‘epsilon’: 0.0001	‘C’: 10,000, ‘epsilon’: 0.0001
Decision tree	‘criterion’: ‘mae’, ‘max_depth’: 30, ‘splitter’: ’best’	‘criterion’: ‘friedman_mse’, ‘max_depth’: 10, ‘splitter’: ’best’
Extra tree	‘criterion’: ‘mse’, ‘max_depth’: 10, ‘splitter’: ‘best’	‘criterion’: ‘friedman_mse’, ‘max_depth’: 40, ‘splitter’: ’best’
MLP_relu	‘alpha’: 0.001, ‘hidden_layer_sizes’: 200, ‘learning_rate’: ‘constant’, ‘learning_rate_init’: 1e-05, ‘solver’: ‘lbfgs’	‘alpha’: 0.001, ‘hidden_layer_sizes’: 100, ‘learning_rate’: ‘constant’, ‘learning_rate_init’: 1e-05, ‘solver’: ‘lbfgs

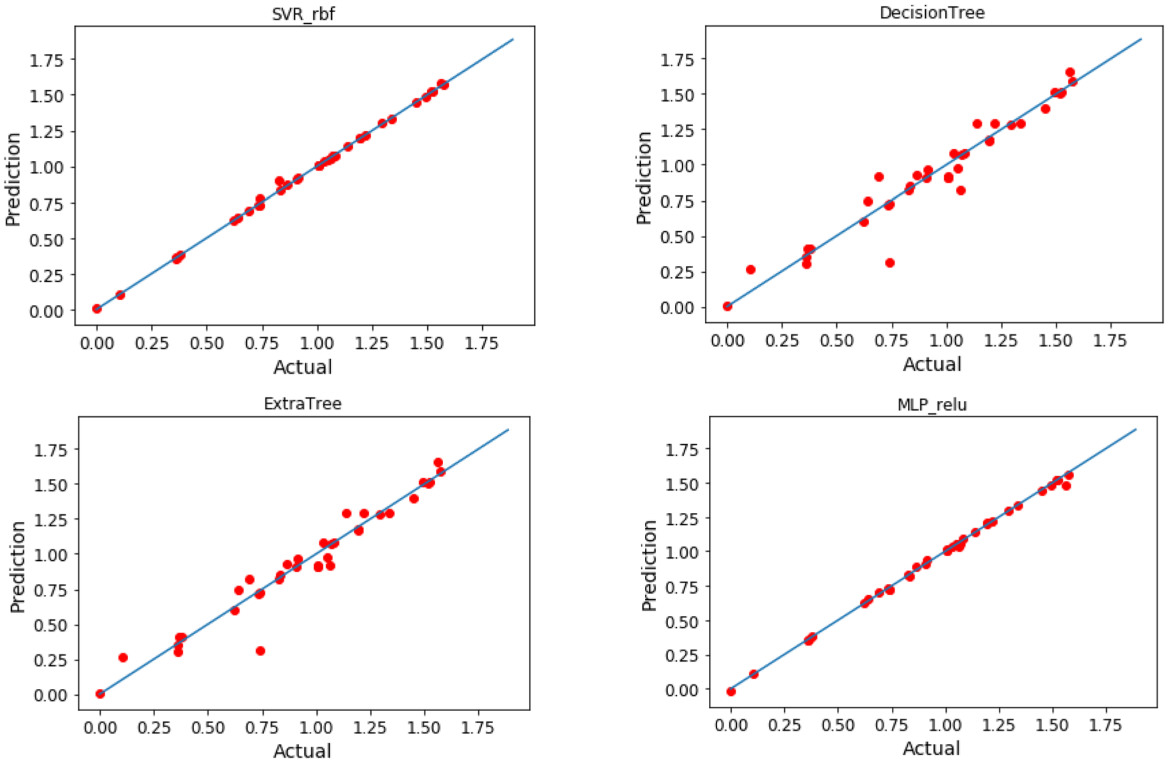


Fig. 14 Regression plots of the best four models for Cl prediction (NACA0012 database)

$$L\left(y_i,f(x)\right) = \left|y_i - f\left(x_i\right)\right|_\varepsilon. \tag{9}$$

To train this model, the following optimization problem has to be solved

$$\min\left(\frac{1}{2}\|w\|^2 + \frac{1}{2}C\sum_{i=1}^{l}\xi_i + \xi_i^*\right), \tag{10}$$

subject to:

$$\begin{aligned} & y_i - w^{\mathrm{T}}\Phi(x) - b \le \varepsilon + \xi_i, \quad i = 1, \dots, l \\ & -y_i + w^{\mathrm{T}}\Phi(x) + b \le \varepsilon + \xi_i^*, \quad i = 1, \dots, l \\ & \xi_i, \xi_i^* \ge 0, \quad i = 1, \dots, l. \end{aligned} \tag{11}$$

To do this, a dual form is usually applied, obtained from the minimization of the Lagrange function that joins the function to minimize and the restrictions. The dual form is

$$\begin{aligned} \max & -\frac{1}{2}\sum_{i,j=1}^{l}\left(\alpha_i + \alpha_i^*\right)\left(\alpha_j + \alpha_j^*\right)K\left(x_i + x_j\right) \\ & - \varepsilon\sum_{i=1}^{l}\left(\alpha_i + \alpha_i^*\right) + \sum_{i=1}^{l} y_i\left(\alpha_i + \alpha_i^*\right) \end{aligned} \tag{12}$$

subject to

$$\sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right) = 0; \quad \alpha_i, \alpha_i^* \in [0, C]. \tag{13}$$

In addition to the restrictions, also must be taken in account the Karush–Kuhn–Tucker conditions and obtain the bias value. In the dual formulation, it is important to emphasize the apparition of the kernel function $K\left(x_i, x_j\right)$, which is equivalent to the scalar product $\langle\Phi\left(x_i\right), \Phi\left(x_j\right)\rangle$. In our case, the kernel function is a Gaussian function:

$$K\left(x_i, x_j\right) = \exp\left(-\gamma \cdot \|x_i - x_j\|^2\right). \tag{14}$$

The final form of the regression model depends on the Lagrange multipliers α_i, α_i^*, following the expression:

$$f(x)\sum_{i=1}^{l}\left(\alpha_i - \alpha_i^*\right)K\left(x_i, x\right) + b. \tag{15}$$

In this way, the SVMr model depends on three parameters, ε, C and γ. ε controls the error margin permitted for the model, as can be seen in Eqs. (10), (11), C controls the number of outliers allowed on the optimization of the function Eq. (10). Finally, γ determines the Gaussian variance for the kernel. Depending on the selection of these values, the model can have a different performance. To obtain the best SVM performance, a search of the most suitable combination of these three parameters must be carried out, usually

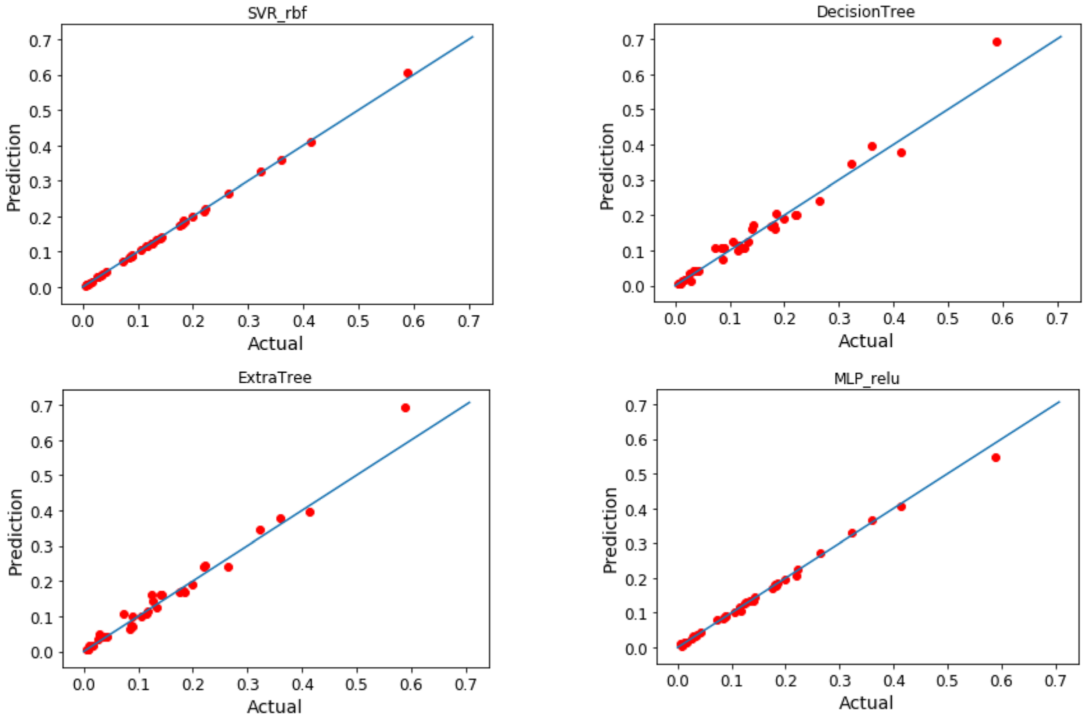


Fig. 15 Regression plots of the best four models for Cd prediction (NACA0012 database)

using cross-validation techniques over the training set. To reduce the computational time of this process, different methods have been proposed in the literature to reduce the search space related to these parameters. In this case, it has been applied the one developed by Ortiz-García et al. [45] which has proven to require pretty short search times.

Numerical results

Three test cases have been considered based on three well-known geometries (see figure below): the NACA0012 airfoil [46], the RAE2822 airfoil [47] and the DPW-W1 wing from the 3rd AIAA Drag Prediction Workshop [48] (Fig. 1).

Information about the aerodynamic databases

CFD data generation

CFD computations for the databases generation were performed with the DLR TAU code [49] and the grids were generated with Centaur [50]. The TAU-Code solves the compressible, three-dimensional Reynolds-averaged Navier–Stokes equations using a finite volume formulation. The TAU-Code is based on a hybrid unstructured-grid approach, which makes use of the advantages of semi-structured prismatic grids in the viscous shear layers near walls, and the flexibility in grid generation offered by tetrahedral grids in the surrounding flow volume. A dual-grid approach with an edge-based data structure is used to make the flow solver independent from the cell types used in the initial grid.

The TAU-Code consists of several different modules, including:

- the grid partitioner, which splits the primary grid into n number of subgrids for n processors;
- the preprocessor module, which uses the information from the initial grid to create a dual grid and second, coarser grids for multi-grid;
- The solver module, which performs the flow calculations on the dual grid;
- the postprocessing module, which is used to convert results to formats usable by popular visualization tools.

Together, all modules are available with python interfaces for computing complex application, e.g. unsteady cases, complete force polar curves or fluid–structure couplings in an automatic framework. Furthermore, it eases the usage on highly massive parallel computers to execute applications.

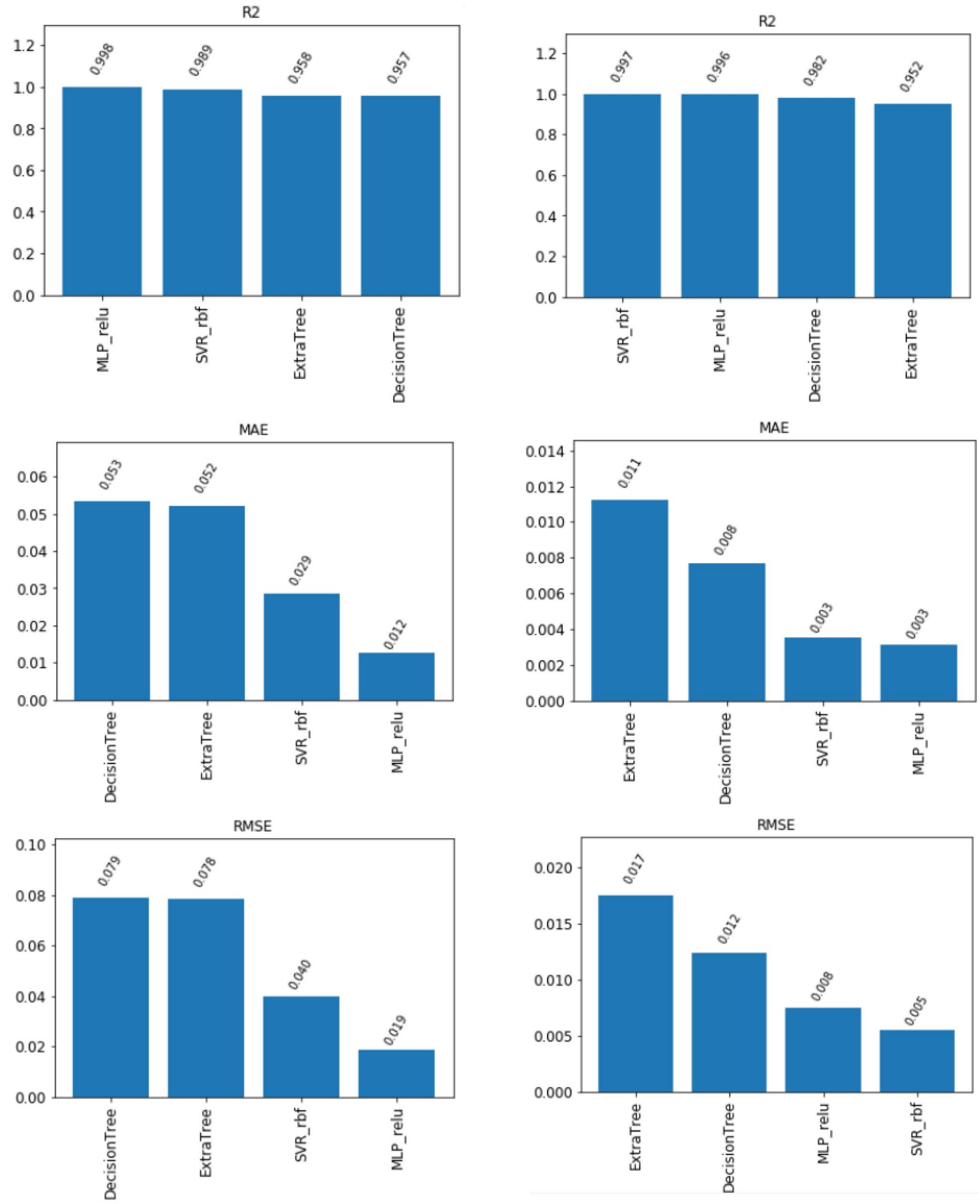


Fig. 16 Best four models comparison results on the RAE2822 database (left: Cl prediction, right: Cd prediction). Metrics are R^2 (determination coefficient), MAE (mean absolute error), RMSE (root mean squared error), ME (max error) and EVS (explained variance score)

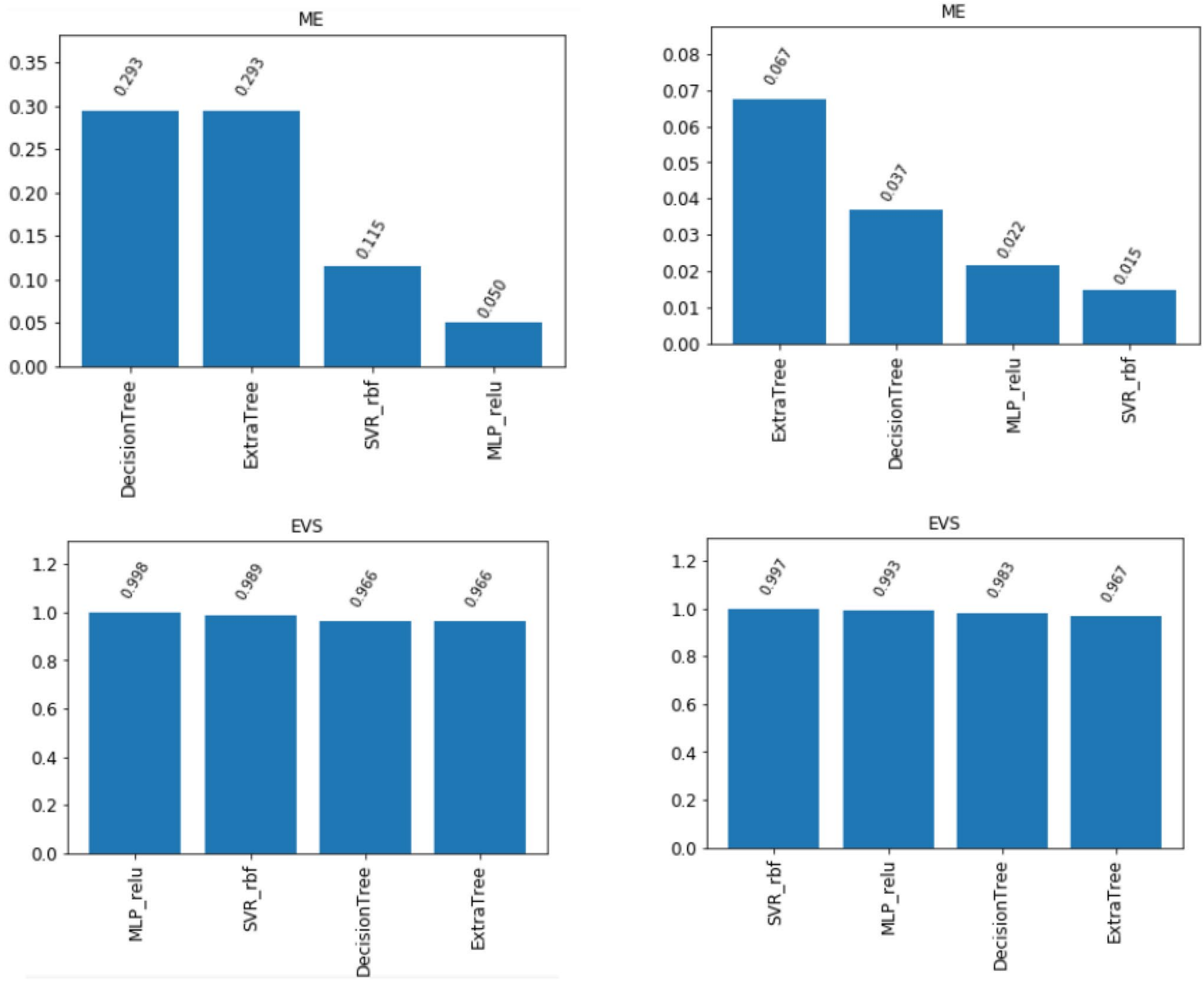


Fig. 16 (continued)

Table 6 Models parameters (RAE2822 database)

	Cl predition	Cd prediction
SVR	‘C’: 10, ‘epsilon’: 0.01	‘C’: 10, ‘epsilon’: 0.0001
Decision tree	‘criterion’: ’mae’, ‘max_depth’: 10, ‘splitter’: ‘best’	‘criterion’: ‘mae’, ‘max_depth’: 30, ‘splitter’: ‘best’
Extra tree	‘criterion’: ‘mae’, ‘max_depth’: 40, ‘splitter’: ‘best’	‘criterion’: ‘mae’, ‘max_depth’: 30, ‘splitter’: ‘random’
MLP_relu	‘alpha’: 0.001, ‘hidden_layer_sizes’: 50, ‘learning_rate’: ‘invscaling’, ‘learning_rate_init’: 0.0001, ‘solver’: ‘lbfgs’	‘alpha’: 0.01, ‘hidden_layer_sizes’: 100, ‘learning_rate’: ‘invscaling’, ‘learning_rate_init’: 1e-05, ‘solver’: ‘lbfgs’

NACA0012 database

This database contains 185 samples and was generated by LHS (Latin Hypercube Sampling) with Mach varying from 0.1 to 0.9 (incrementing 0.1 each time) and AoA (Angle of Attack) varying from 0 to 20 (incrementing 1 each time). The Mach number is a dimensionless quantity in fluid dynamics representing the ratio of flow velocity past a boundary to the local speed of sound. Angle of attack (AOA) is the angle between the oncoming air or relative wind and a reference line on the airplane or wing. It is also important to mention that if all the 185 samples had run correctly (with

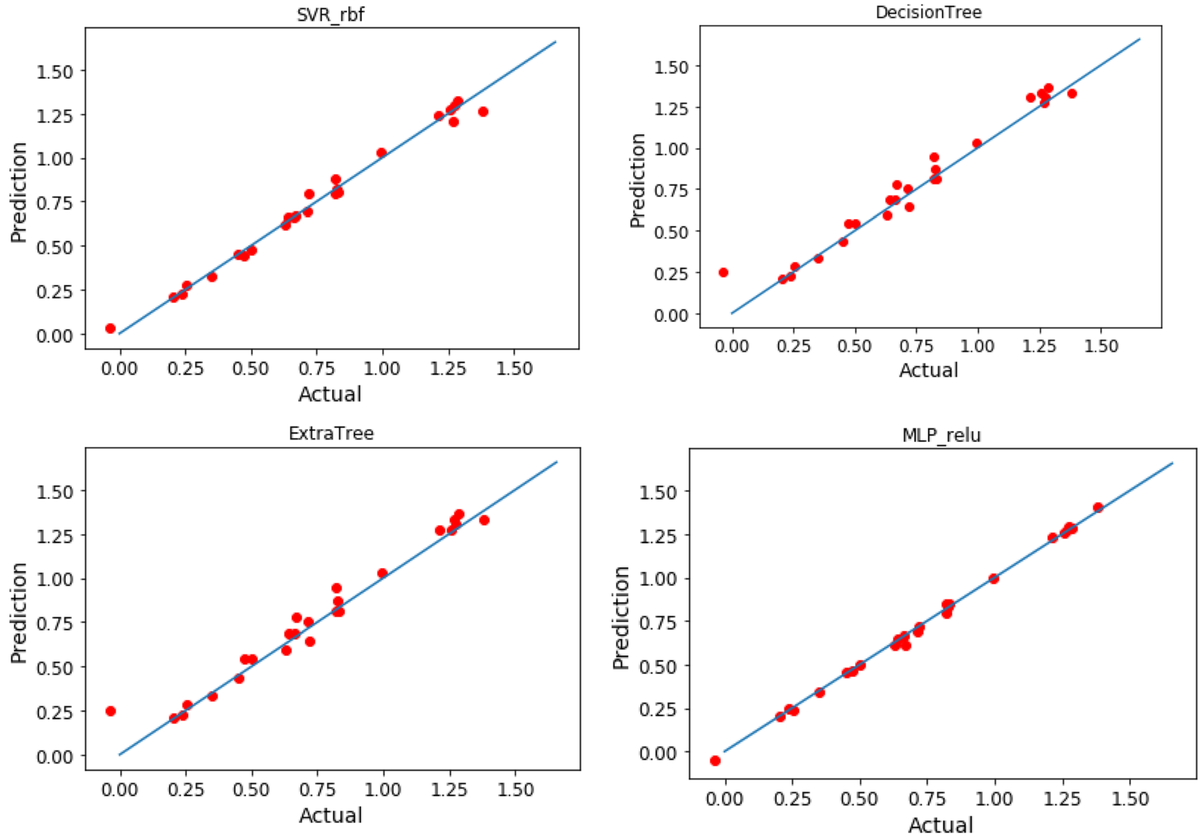


Fig. 17 Regression plots of the best four models for Cl prediction (RAE2822 database)

the CFD tool used for generating the database) there had not been empty spaces in the plane. This missing points are due to simulations that did not converge with the CFD tool, that means that the lift or drag coefficient did not reach a stable value after a considerable number of iterations of the flow solver.

The following figures show the distribution of Mach versus AoA of the database samples, lift and drag coefficient curves of the database samples. The lift coefficient (Cl) is a dimensionless quantity that is used to express the ratio of the lift force to the force produced by the dynamic pressure times the area. The drag coefficient (Cd) is also a dimensionless quantity that is used to quantify the drag or resistance of an object in a fluid environment, such as air or water. It is used in the drag equation in which a lower drag coefficient indicates the object will have less aerodynamic drag. The drag coefficient is always associated with a particular surface area (Figs. 2, 3).

RAE2822 database

This database contains 122 samples and was generated by LHS with Mach varying from 0.1 to 0.9 (incrementing 0.1 each time) and AoA varying from 0 to 15 (incrementing 1 each time). It is also important to mention that if all the 122 samples had run correctly (with the CFD tool used for generating the database) there had not been empty spaces in the plane. This missing points are due to simulations that did not converge with the CFD tool, that means that the lift or drag coefficient did not reach a stable value after a considerable number of iterations of the flow solver.

The following figures show the distribution of Mach versus AoA of the database samples, lift and drag coefficient curves of the database samples (Figs. 4, 5).

DPW database

This database contains 100 samples and was generated by LHS with Mach varying from 0.1 to 0.8 (varying 0.1 each time) and AoA varying from 0 to 15 (varying 1 each time). Again, it is important to mention that if all the 100 samples had run correctly (with the CFD tool used for generating the database) there had not been empty spaces in the plane. This missing points are due to simulations that did not converge with the CFD tool, that means that the lift or drag coefficient did not reach a stable value after a considerable number of iterations of the flow solver. In this case, it can be also observed that for the highest numbers of Mach and

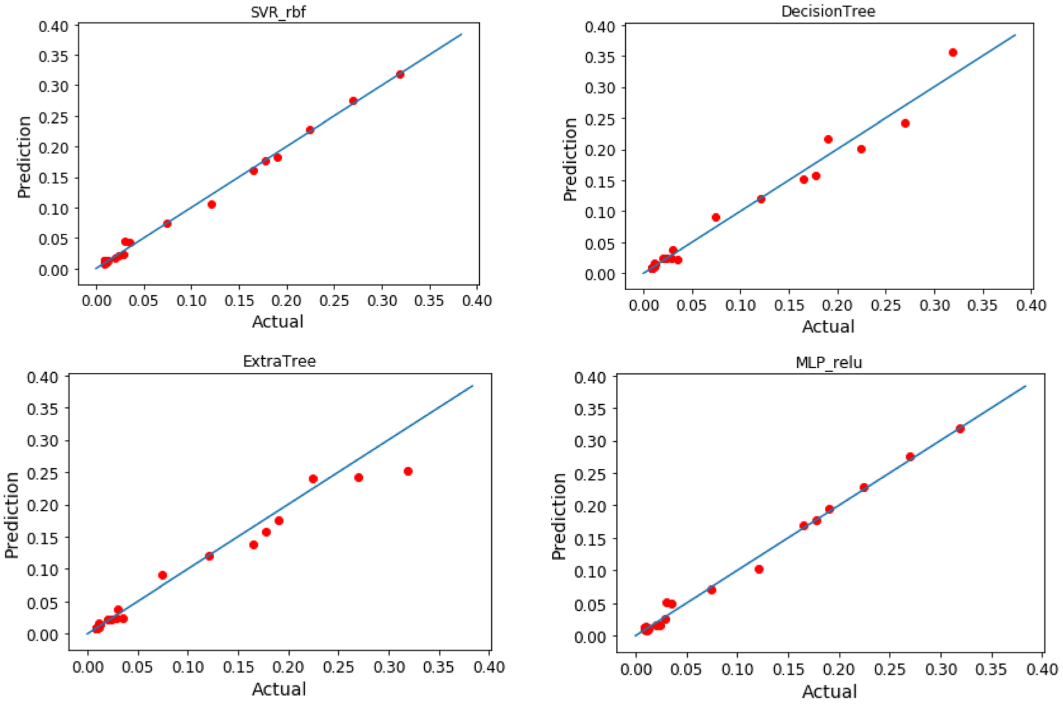


Fig. 18 Regression plots of the best four models for Cd prediction (RAE2822 database)

AoA, the probability of the solver to diverge is bigger, due to the instabilities that are expected in the flow fields.

The following figures show the distribution of Mach versus AoA of the database samples, lift and drag coefficient curves of the database samples (Figs. 6, 7).

In addition, the following table shows some additional statistics on the databases. The count, mean, min, and max rows are self-explanatory. The std row shows the standard deviation (which measures how dispersed the values are). The 25%, 50%, and 75% rows show the corresponding percentiles: a percentile indicates the value below which a given percentage of observations in a group of observations falls (Table 1).

Information about the strategy followed

In this research, the following strategy was followed:

1. First, 15 different regression models were selected to be compared on the same databases. In this step, the split between training and testing sets was done with a pure random sampling method and considering 80% of the initial samples for the train set and the other 20% for the test set. A standard scaling have been applied to the databases prior the training of the model.
2. The four best performing models were then selected for further analysis and cross-validation, to make the prediction results more robust and less dependent on the initial dataset split.

The following pictures show the main steps in this strategy (Figs. 8, 9)

Information about the methods and the comparison metrics

With the aim to provide a broad comparison of the existing regression methods, 15 different approaches have been selected. The details of these methods are displayed in the following Table 2.

The metrics that will be used for models comparison are described in the following Table 3.

Models comparison

In this section, the selected 15 methods have been applied to the three databases and compute the comparison metrics detailed in the previous section.

NACA0012 database

From these results, the following conclusions can be drawn (Fig. 10):

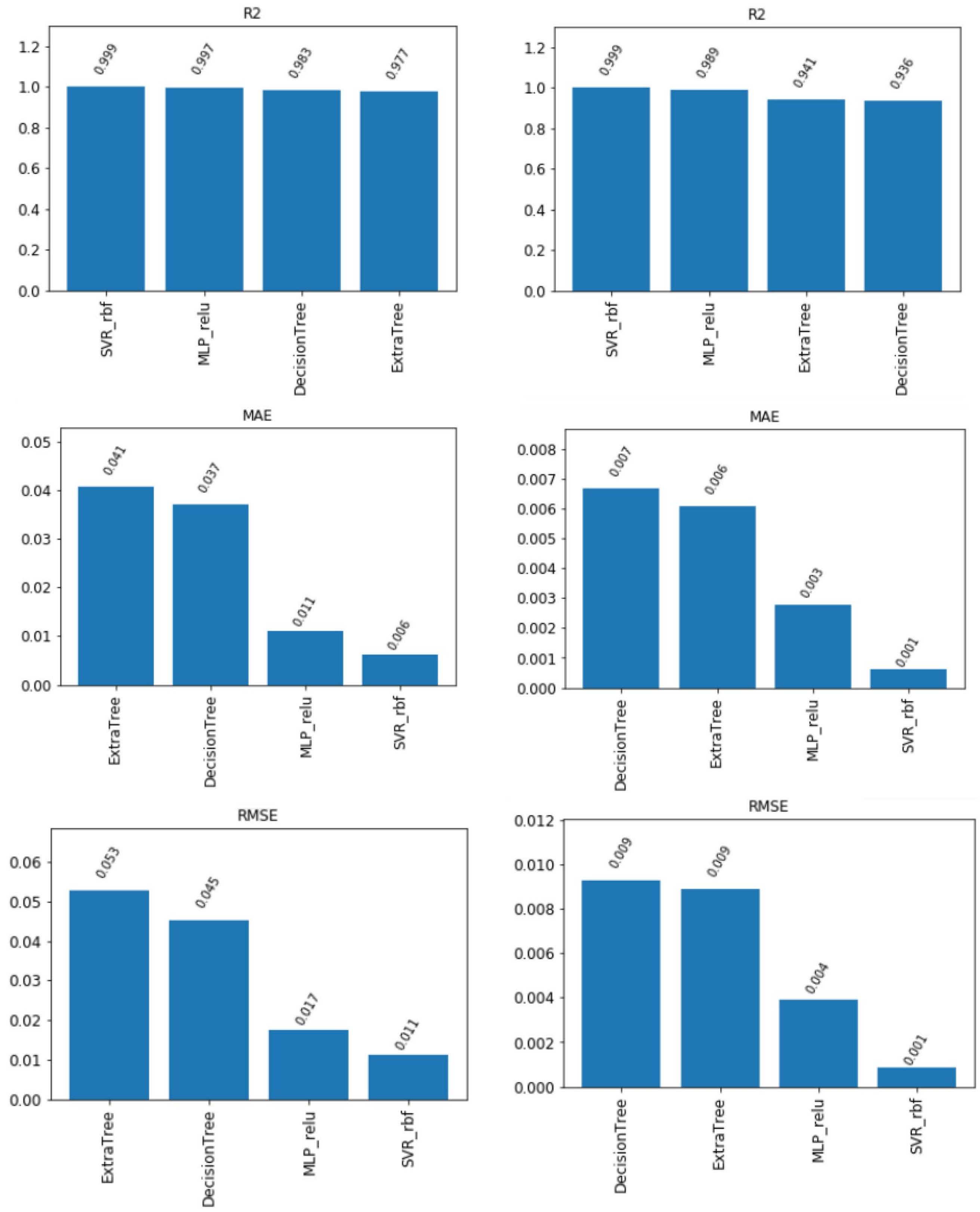


Fig. 19 Best four models comparison results on the DPW database (left: Cl prediction, right: Cd prediction). Metrics are R^2 (determination coefficient), MAE (mean absolute error), RMSE (root mean squared error), ME (max error) and EVS (explained variance score)

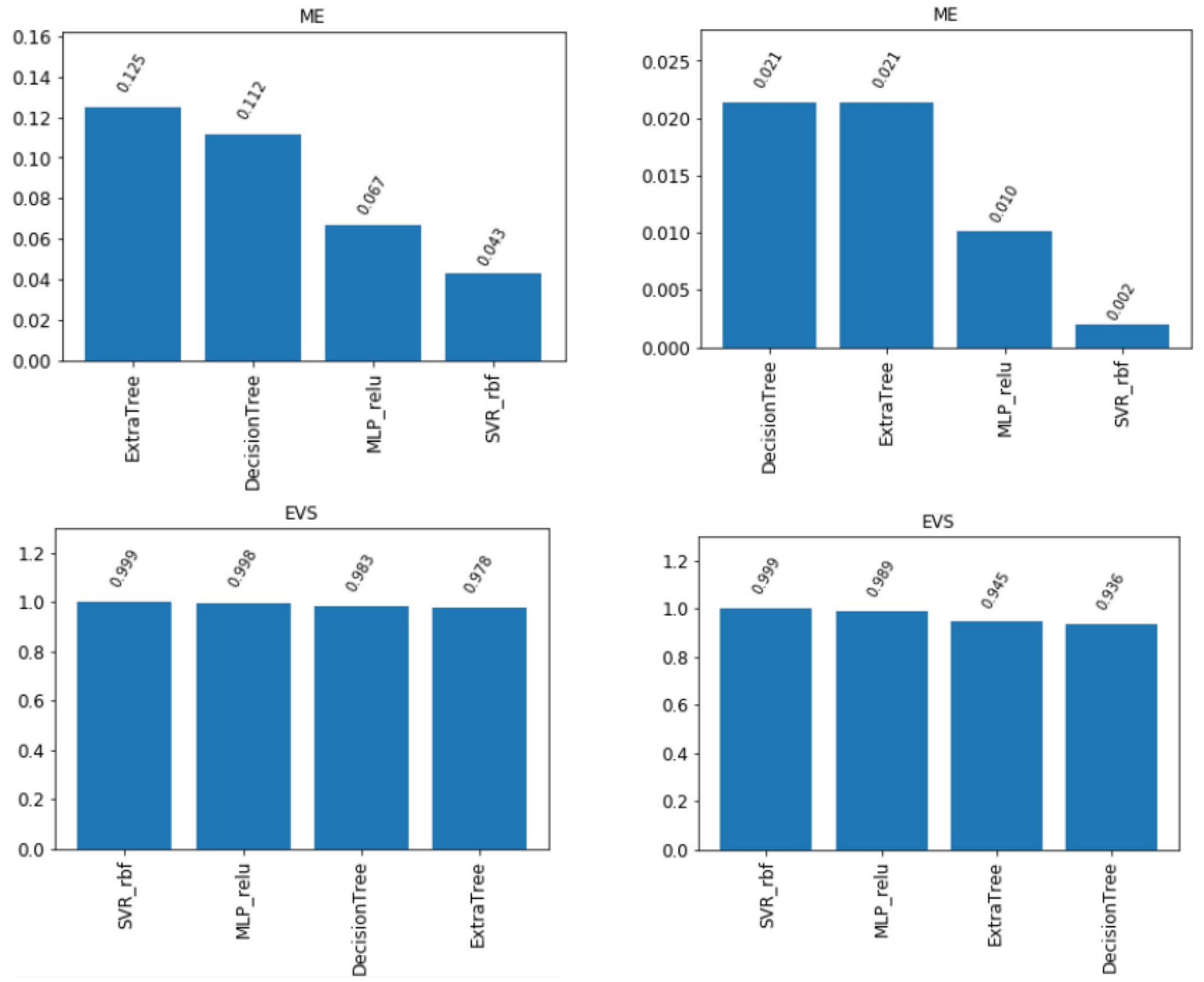


Fig. 19 (continued)

- The best model in terms of minimum MAE, RMSE and ME, and maximum R^2 and EVS is the SVR with a Radial Basis function kernel. This is the case in both Cl and Cd predictions. For this model, the RMSE is 0.084 for the Cl and 0.013 for the Cd, which is a reasonable accuracy.
- Decision tree and Extra tree models also provide very good metric results.
- The order of models in terms of performance remains almost invariable in all comparison metrics.
- Linear regression models are not able to provide good results, as it was expected.
- The kernel function used in the SVR model has a very strong influence on the metric results, for instance SVR with a polynomial kernel achieved a R^2 of 0.68 compared

Table 7 Models parameters (DPW database)

	Cl predition	Cd prediction
SVR	‘C’: 50, ‘epsilon’: 0.001	‘C’: 10, ‘epsilon’: 0.0001
Decision tree	‘criterion’: ‘mae’, ‘max_depth’: 20, ‘splitter’: ‘best’	‘criterion’: ‘friedman_mse’, ‘max_depth’: 20, ‘splitter’: ’random’
Extra tree	‘criterion’: ‘mae’, ‘max_depth’: 30, ‘splitter’: ‘best’	‘criterion’: ‘friedman_mse’, ‘max_depth’: 20, ‘splitter’: ‘random’
MLP_relu	‘alpha’: 0.001, ‘hidden_layer_sizes’: 50, ‘learning_rate’: ‘invscaling’, ‘learning_rate_init’: 0.0001, ‘solver’: ‘lbfgs’	‘alpha’: 0.01, ‘hidden_layer_sizes’: 50, ‘learning_rate’: ‘invscaling’, ‘learning_rate_init’: 0.0001, ‘solver’: ‘lbfgs’

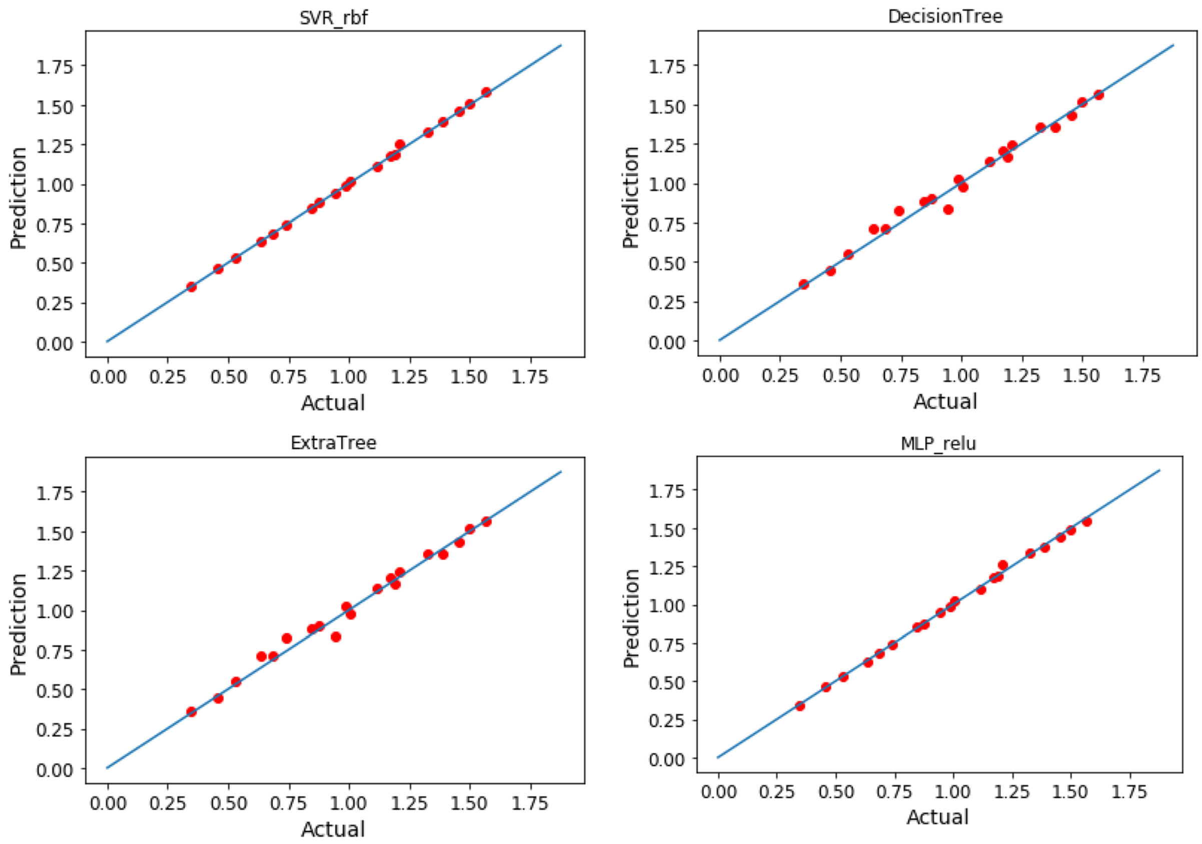


Fig. 20 Regression plots of the best four models for Cl prediction (DPW database)

to the 0.95 of the SVR with a Radial Basis function kernel.

RAE2822 database

From these results, similar conclusions as in the previous test case can be drawn (Fig. 11):

- The best model in terms of minimum RMSE and ME, and maximum R^2 and EVS is the SVR with a Radial Basis function kernel. This is the case in both Cl and Cd predictions. For this model, the RMSE is 0.054 for the Cl and 0.012 for the Cd, which is a reasonable accuracy.
- Decision Tree, Extra tree and MLP_relu models also provide very good metric results.
- The order of models in terms of performance remains almost invariable in all comparison metrics.
- Linear regression models are not able to provide good results, as it was expected.
- As it happened also in the NACA0012 database, the kernel function used in the SVR model has a very strong influence on the metric results, for instance SVR with a polynomial kernel achieved a R^2 of 0.60 compared to the 0.98 of the SVR with a Radial Basis function kernel.
- In RMSE and ME metrics, there is a considerable difference between the two best performing models (SVR_rbf and Decision tree or Extra tree, depending of Cl or Cd prediction).

DPW database

From these results, again similar conclusions as in the previous tests cases can be drawn (Fig. 12):

- The best model in terms of minimum RMSE and ME, and maximum R^2 and EVS is the SVR with a Radial Basis function kernel. This is the case in both Cl and Cd predictions. For this model, the RMSE is 0.020 for the Cl and 0.008 for the Cd, which is a reasonable accuracy.
- Decision tree and Extra tree models also provide very good metric results.
- The order of models in terms of performance remains almost invariable in all comparison metrics.

As additional information, the parameters for the best four performing models are displayed in the following Table 4. Remember that at this stage, no optimization of the model parameters has been performed.

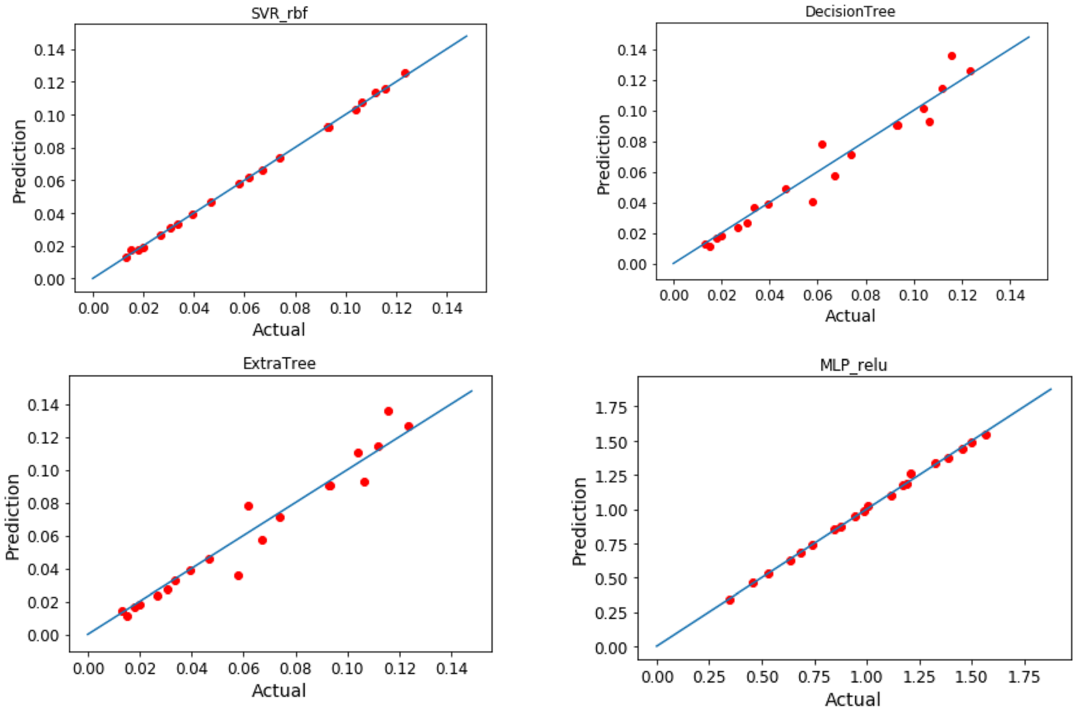


Fig. 21 Regression plots of the best four models for Cd prediction (DPW database)

Model parameters optimization and cross-validation of the best four models

Now, with the four best performing models (SVR_rbf, Decision tree, Extra tree y MLP_relu, cross-validation was applied to be sure that the results are not affected by the training and testing datasets split and optimization to the model parameters (using a grid search technique) to find out what is the maximum accuracy one can get.

The following pictures show the metrics results for each of the models in the databases.

NACA0012 database

As can be observed, SVR_rbf remains in the first position in all metrics, for instance regarding R^2 metric, which has now increased up to 0.99 for both Cl and Cd (before it was 0.95 for Cl and 0.98 for Cd) (Fig. 13).

It is also important to notice the strong performance increment of the MLP_relu model, after the parameter optimization. For instance, in terms of R^2, this model achieved values of 0.99 for both coefficients prediction and before it was 0.84 for Cl and 0.93 for Cd.

The final parameters used for the model can be observed in the Table 5 below.

The following pictures show the regression plot of these four models. Again, the outstanding behaviour of the SVR_rbf model is confirmed (Figs. 14, 15).

RAE2822 database

As can be observed, SVR_rbf and MLP_relu are in the first positions in all metrics. It is not possible to draw conclusions on which of these two models behave better since the performance varies when predicting Cl or Cd. Anyway, the differences between these two models in terms of R^2 are almost neglectable (Fig. 16).

These results were obtained with the following model parameters (Table 6).

The following pictures show the regression plot of these four models. Again, the outstanding behaviour of the SVR_rbf and MLP_relu models is confirmed (Figs. 17, 18).

DPW database

As can be observed, SVR_rbf remains in the first position in all metrics, for instance regarding R^2 metric, which has now increased up to 0.99 for both Cl and Cd (before it was only 0.95 for Cd) (Fig. 19).

It is also important to notice the strong performance increment of the MLP_relu model, after the parameter optimization. For instance, in terms of R^2, this model achieved values of 0.99 for Cl and 0.98 for Cd and before it was 0.92 for Cl and 0.13 for Cd.

The final parameters used for the model can be observed in the Table 7 below.

The following pictures show the regression plot of these four models. Again, the outstanding behaviour of the SVR_rbf model is confirmed (Figs. 20, 21).

Conclusions and future research

This paper focuses on making a deep comparison of different surrogate regression models for aerodynamic coefficient prediction in different aeronautical configurations. In particular, three different aeronautical configurations have been used, a NACA0012 airfoil, a RAE2822 airfoil and 3D DPW wing.

From the obtained results, the following conclusions can be summarized:

- The best models in terms of the metrics analysed are SVR with a radial basis function kernel and a multi-layer perceptron neural network with the rectified unit function as the activator function.
- The superiority of the support vector regression model is justified by its better generalization performance than other regression methods, which derives to a better precision accuracy. In addition, it is efficient for high-dimensional spaces and when the number of samples is limited, as it happens in computational aerodynamics, due to the computational cost of generating each sample of the training database. Moreover, the computational complexity of SVR does not depend on the dimensionality of the input space, which is also an advantage in this application field.
- Model parameters optimization is crucial to obtain good accuracy, especially for the MLP_relu model, where the metrics value drastically changed when optimizing the number of hidden layers, learning rate, etc.
- The order of models in terms of performance remains almost invariable in all comparison metrics amongst the three databases studied.
- Linear regression models are not able to provide good results, as it was expected.
- The application of surrogate regression models for aerodynamic coefficients prediction is feasible and has a tremendous potential to reduce the computational time of CFD simulations, especially when considering aerodynamic design loops.

There is still further potential to be exploited: a clever generation of the samples in the initial dataset (not LHS), the use of more robust model validation strategies, such as cross-fold validation, the combination of data with multi-fidelity within the aerodynamic database (eg. CFD, wind tunnel, flight testing data, etc.), the comparison of different regression models and tuning these parameters, etc. These issues will be undertaken in future works.

In addition, the use of deep learning techniques and the comparison against traditional machine learning techniques will be considered in the near future, since recent scientific publications have demonstrated their potential in other sectors. In particular, the application of deep learning methods to computational aerodynamics will be investigated during next years in the frame of an European project titled “Machine learning and data-driven approaches for aerodynamic analysis and uncertainty quantification” (acronym ML4AERO) with the collaboration of several research institutions (INTA, DLR, ONERA, CIRA, FOI, AIRBUS, OPTIMAD, IRT, INRIA and the University of Twente). There, the feasibility of applying deep learning methods and convolutional neural networks to aerodynamic analysis and design will be analysed.

Finally, it is important to mention that all databases used in this paper are freely available for the scientific community.

Annex: Databases

Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd
0.1	0	0.00034864	0.01355065	0.1	0	0.20478228	0.01150837	0.1	0	0.34490435	0.01532178
0.1	1	0.11786291	0.01348484	0.1	1	0.31598134	0.01169231	0.1	1	0.43094063	0.01811268
0.1	2	0.23511923	0.0140823	0.1	2	0.42641901	0.0120037	0.1	2	0.51661595	0.02160981
0.1	3	0.35163278	0.01537563	0.1	3	0.5356642	0.01246302	0.1	3	0.60181206	0.02580588
0.1	4	0.46693354	0.01738756	0.1	4	0.64335269	0.01310139	0.1	4	0.68640261	0.03069294
0.1	5	0.58047163	0.02021214	0.1	5	0.74901551	0.01395388	0.1	5	0.77025437	0.03626244
0.1	6	0.69169693	0.02392192	0.1	6	0.85205249	0.01506802	0.1	6	0.85323022	0.04250429
0.1	7	0.79989887	0.02866703	0.1	7	0.95153298	0.01652279	0.1	7	0.9351778	0.04940593
0.1	8	0.90439914	0.03453483	0.1	8	104.615466	0.01843026	0.1	8	101.593228	0.05695184
0.1	9	100.444.087	0.04161181	0.1	9	113.396707	0.02096989	0.1	9	109.530372	0.06512261
0.1	10	109.908405	0.05002468	0.1	10	12.116608	0.02446154	0.1	10	117.308781	0.07389458
0.1	11	118.733623	0.05986405	0.1	11	127.319621	0.0295291	0.1	11	124.904355	0.0832391
0.1	12	126.794121	0.07127074	0.1	12	130.757569	0.0374886	0.1	12	132.291412	0.09312363
0.1	13	133.977189	0.08428013	0.1	13	129.496163	0.05107973	0.1	13	139.439736	0.10351022
0.1	14	140.154664	0.09897053	0.2	0	0.21152056	0.01002395	0.1	14	146.316246	0.11435581
0.1	15	145.193494	0.11539774	0.2	1	0.32545539	0.01018193	0.1	15	152.884459	0.1256121
0.1	16	148.977686	0.13355583	0.2	2	0.43869458	0.01044382	0.2	0	0.35784439	0.01123022
0.1	17	1.514051	0.15344376	0.2	3	0.55085001	0.01082799	0.2	1	0.44564584	0.01414844
0.1	18	152.400049	0.17501892	0.2	4	0.66160695	0.01136067	0.2	2	0.53315257	0.01776694
0.1	19	151.933408	0.19821272	0.2	5	0.77065115	0.01206581	0.2	3	0.62025489	0.02208108
0.1	20	150.055195	0.22290925	0.2	6	0.8774021	0.0129811	0.2	4	0.70683211	0.0270855
0.2	0	0.00147839	0.00764385	0.2	7	0.98111069	0.01416774	0.2	5	0.79274664	0.03277378
0.2	1	0.12058562	0.00766263	0.2	8	108.062166	0.01571391	0.2	6	0.8778417	0.03913748
0.2	2	0.23940088	0.0081628	0.2	9	117.425782	0.01775307	0.2	7	0.96193191	0.04616506
0.2	3	0.35741302	0.00917954	0.2	10	125.932828	0.02051788	0.2	8	104.480182	0.05384133
0.2	4	0.47400585	0.01080112	0.2	11	133.095237	0.02444987	0.2	9	112.620348	0.06214616
0.2	5	0.58835445	0.01320961	0.2	12	138.026534	0.03047922	0.2	10	120.584063	0.07105415
0.2	6	0.69976032	0.01649634	0.2	13	139.096829	0.04067699	0.2	11	128.337244	0.08053246
0.2	7	0.80734698	0.02077916	0.2	14	0.77538343	0.19726277	0.2	12	135.838641	0.09053913
0.2	8	0.91003064	0.02619192	0.3	0	0.21838947	0.00934052	0.2	13	143.040181	0.10102278
0.2	9	100.649057	0.03288686	0.3	1	0.33591982	0.00949857	0.2	14	149.883792	0.11192032
0.2	10	109.515088	0.04101942	0.3	2	0.45274708	0.00975815	0.2	15	15.629855	0.12315464
0.2	11	117.424983	0.05073368	0.3	3	0.56849657	0.01013881	0.3	0	0.36874228	0.00990418
0.2	12	124.183166	0.06216799	0.3	4	0.68283083	0.0106665	0.3	1	0.45872605	0.01298114
0.2	13	129.599939	0.07537635	0.3	5	0.79542628	0.01136966	0.3	2	0.54843083	0.01678315
0.2	14	133.476649	0.0904007	0.3	6	0.90561674	0.01229446	0.3	3	0.63774344	0.02130707
0.2	15	135.644989	0.10722269	0.3	7	101.227207	0.01352695	0.3	4	0.72652433	0.02654955
0.2	16	135.983594	0.12575123	0.3	8	111.348966	0.01523225	0.3	5	0.81461785	0.03250621
0.2	17	134.467883	0.14581214	0.3	9	120.455267	0.01781221	0.3	6	0.90182815	0.03917107
0.2	18	131.234511	0.16717247	0.3	10	126.909232	0.02285614	0.3	7	0.98789493	0.04653312
0.2	19	126.688364	0.18955789	0.3	11	126.600957	0.03513813	0.3	8	107.251145	0.05457734
0.2	20	121.601526	0.21263805	0.3	14	0.80931463	0.19002002	0.3	9	115.527777	0.06328186
0.3	0	0.00166224	0.00567437	0.3	15	0.79469274	0.22054701	0.3	10	123.566974	0.07261592
0.3	1	0.12394694	0.00575906	0.4	0	0.22697387	0.00893865	0.3	11	131.298013	0.08253557
0.3	2	0.24590258	0.00628682	0.4	1	0.34981432	0.00910732	0.3	12	138.623968	0.09297645
0.3	3	0.36697165	0.00729793	0.4	2	0.47193074	0.00938471	0.3	13	145.408922	0.10384448
0.3	4	0.48580126	0.0090936	0.4	3	0.59295454	0.00979211	0.3	14	151.463925	0.1150027
0.3	5	0.60172345	0.01173898	0.4	4	0.71248721	0.01036197	0.3	15	156.524945	0.12625974

Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd
0.3	6	0.7135568	0.01540221	0.4	5	0.83001058	0.01114133	0.4	0	0.38199054	0.00953887
0.3	7	0.81978422	0.02026493	0.4	6	0.94398001	0.01225255	0.4	1	0.47511661	0.01284081
0.3	8	0.91833746	0.02658771	0.4	7	104.697298	0.01430937	0.4	2	0.56797754	0.01691617
0.3	9	100.674362	0.03461385	0.4	8	110.537962	0.02061118	0.4	3	0.66045687	0.02176407
0.3	10	108.132847	0.0447432	0.4	13	0.85517366	0.15764748	0.4	4	0.75238654	0.02738474
0.3	11	11.392022	0.05709218	0.5	0	0.23870972	0.00871551	0.4	5	0.8435564	0.03377798
0.3	12	117.834998	0.0714785	0.5	1	0.36960458	0.00890705	0.4	6	0.9336623	0.04094201
0.3	13	119.689698	0.08773689	0.5	2	0.49980719	0.00922257	0.4	7	10.222786	0.04887145
0.3	14	1.194207	0.10557819	0.5	3	0.62894137	0.00968923	0.4	8	110.869795	0.05755978
0.3	15	11.710816	0.12463177	0.5	4	0.75646706	0.01036055	0.4	9	11.916454	0.06699707
0.3	16	11.313085	0.14443285	0.5	5	0.87993837	0.01144269	0.4	10	126.828659	0.07718435
0.3	17	108.254594	0.16463531	0.5	6	0.97111817	0.01540752	0.4	11	133.268237	0.0881008
0.3	18	103.479689	0.18515759	0.5	7	100.968.497	0.02454478	0.4	12	137.858719	0.09946716
0.3	19	0.99460818	0.20597369	0.5	8	0.98585454	0.04116198	0.4	13	140.149934	0.11080493
0.3	20	0.96193387	0.22729269	0.6	0	0.25598625	0.00864509	0.5	0	0.40001585	0.00975765
0.4	0	0.00156924	0.00475901	0.6	1	0.39996208	0.00888127	0.5	1	0.49793815	0.01340333
0.4	1	0.12902259	0.00490616	0.6	2	0.5435357	0.00927495	0.5	2	0.59569143	0.0179093
0.4	2	0.25614837	0.00550179	0.6	3	0.68639769	0.0098805	0.5	3	0.69311377	0.02328386
0.4	3	0.38183434	0.00672044	0.6	4	0.82031306	0.01163186	0.5	4	0.789923	0.02954492
0.4	4	0.5044046	0.00885155	0.6	5	0.94888813	0.01607704	0.5	5	0.88528512	0.03675215
0.4	5	0.62258142	0.01207326	0.6	6	105.501.693	0.02528943	0.5	6	0.9752414	0.04521391
0.4	6	0.73415927	0.01667064	0.6	7	11.061.645	0.03897731	0.5	7	105.347332	0.05529114
0.4	7	0.83619501	0.02298766	0.6	8	102.831.852	0.05725626	0.5	8	111.687337	0.0668394
0.4	8	0.92427571	0.03152067	0.6	9	0.84746188	0.09203749	0.5	9	116.481554	0.07950771
0.4	9	0.99410253	0.04254277	0.6	10	0.80219426	0.1243578	0.5	10	119.615115	0.09280839
0.4	10	104.336401	0.0558185	0.6	11	0.79361849	0.15172702	0.5	11	120.795335	0.10614303
0.4	11	106.901774	0.07115748	0.6	12	0.79538765	0.17612524	0.6	0	0.42662636	0.01053514
0.4	12	107.033681	0.08818781	0.6	13	0.81117798	0.20116541	0.6	1	0.53215968	0.01475246
0.4	13	104.875427	0.10636047	0.6	14	0.82352243	0.22422017	0.6	2	0.63769192	0.02003221
0.4	14	100.953571	0.12504717	0.6	15	0.84541693	0.24918325	0.6	3	0.74031446	0.02685975
0.4	15	0.96219326	0.1437808	0.7	0	0.28552023	0.0088277	0.6	4	0.83611646	0.03596358
0.4	16	0.91714303	0.1626428	0.7	1	0.4552778	0.00918366	0.6	5	0.92496205	0.04759207
0.4	17	0.87976015	0.18190757	0.7	2	0.62759227	0.00981429	0.6	6	100.605016	0.06171506
0.4	18	0.85118012	0.20176305	0.7	3	0.80939577	0.01123075	0.6	7	107.851749	0.07813513
0.4	19	0.83142716	0.22208854	0.7	4	0.97088706	0.02100958	0.6	8	114.035887	0.09635543
0.4	20	0.81730939	0.24280637	0.7	5	104.109807	0.03806899	0.6	9	11.904048	0.1158151
0.5	0	0.0013092	0.00431248	0.7	6	101.928409	0.05606353	0.6	10	12.265724	0.13570755
0.5	1	0.13699071	0.00452484	0.7	7	0.96864203	0.07417348	0.6	11	124.690584	0.15516688
0.5	2	0.27220791	0.00527327	0.7	8	0.909723	0.09332397	0.7	0	0.47478709	0.01242688
0.5	3	0.40529214	0.00680007	0.7	9	0.85474593	0.11439034	0.7	1	0.59416051	0.01841556
0.5	4	0.53365078	0.0094989	0.7	10	0.8136909	0.1390286	0.7	2	0.71263573	0.02739059
0.5	5	0.65500918	0.0136707	0.7	11	0.81763773	0.16551528	0.7	3	0.82867544	0.04034131
0.5	6	0.76592039	0.01976437	0.7	12	0.83099549	0.1904208	0.7	4	0.94051449	0.05765871
0.5	7	0.85968697	0.02859707	0.7	13	0.85202205	0.21635245	0.7	5	104.452939	0.07898422
0.5	8	0.93110545	0.04048238	0.7	14	0.87184003	0.24104815	0.7	6	113.844811	0.10354212
0.5	9	0.97859639	0.05500783	0.8	0	0.25379077	0.02261738	0.7	7	122.049898	0.13032914
0.5	10	0.99966192	0.07142576	0.8	1	0.37291229	0.02757891	0.8	0	0.59842495	0.03070903
0.5	11	0.99461587	0.08896353	0.8	2	0.47440306	0.03524511	0.8	1	0.73936297	0.0501211
0.5	12	0.96501081	0.10691509	0.8	3	0.56312584	0.04564638	0.8	2	0.86822118	0.07409706
0.5	13	0.91613681	0.12481628	0.8	4	0.6460511	0.05832932	0.8	3	0.98333082	0.10115296
0.5	14	0.86503783	0.14251852	0.8	5	0.71874703	0.07377648	0.8	4	108.530449	0.13027077

Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd
0.5	15	0.81917669	0.1606226	0.8	6	0.77900244	0.09125873	0.8	5	117.487716	0.16068126
0.5	16	0.78575895	0.17937346	0.8	7	0.83005988	0.11019433				
0.5	17	0.76234041	0.19901087	0.8	8	0.87349327	0.12991298				
0.6	0	0.0009119	0.00416125	0.8	9	0.91336673	0.15089809				
0.6	1	0.15034584	0.00446971	0.8	10	0.94395678	0.17216155				
0.6	2	0.29936634	0.005434	0.8	11	0.96521429	0.19428368				
0.6	3	0.44558056	0.00743859	0.8	12	0.97446734	0.21664782				
0.6	4	0.58566612	0.01106919	0.8	13	0.97990796	0.24185741				
0.6	5	0.71641357	0.01769764	0.8	14	0.99465858	0.26980803				
0.6	6	0.83164647	0.02927649	0.8	15	102.677325	0.29960937				
0.6	7	0.92507028	0.0454219	0.9	0	− 0.03959858	0.12059924				
0.6	8	0.99671461	0.06461165	0.9	1	0.06822316	0.11970889				
0.6	9	105.085356	0.08561563	0.9	2	0.18051904	0.12310501				
0.6	10	108.157905	0.10677951	0.9	3	0.29714686	0.13081809				
0.6	11	108.386342	0.12648953	0.9	4	0.41716614	0.14252143				
0.6	12	106.537418	0.14454872	0.9	5	0.54251249	0.15831269				
0.6	13	101.799052	0.15965823	0.9	6	0.66775734	0.17707051				
0.6	14	0.9276694	0.17099549	0.9	7	0.80270165	0.19788185				
0.6	15	0.82413283	0.18302397	0.9	8	0.97275077	0.2228966				
0.6	16	0.75017574	0.20017763	0.9	9	109.793814	0.2519245				
0.6	17	0.73720931	0.21962102	0.9	11	128.626327	0.31933105				
0.6	18	0.72817532	0.24000652	0.9	12	136.381377	0.35620579				
0.6	19	0.72512852	0.26104437	0.9	13	143.461078	0.39464213				
0.7	0	0.00042779	0.00437586								
0.7	1	0.17780983	0.00485724								
0.7	2	0.35841207	0.00684577								
0.7	3	0.53344121	0.01439492								
0.7	4	0.68918301	0.02921294								
0.7	5	0.82633986	0.04998559								
0.7	6	0.94556538	0.07494196								
0.7	7	104.722633	0.10254214								
0.7	8	113.100158	0.13150703								
0.7	9	119.559795	0.16047414								
0.7	10	124.248574	0.18862919								
0.7	11	127.330107	0.21537239								
0.7	12	129.015172	0.24036068								
0.7	13	129.612152	0.26357329								
0.7	14	128.629569	0.28347853								
0.7	15	123.736911	0.2941774								
0.7	16	121.579887	0.30991999								
0.7	17	106.062252	0.29430817								
0.7	18	0.95511579	0.29199597								
0.7	19	0.77880954	0.29346789								
0.7	20	0.78938387	0.31550949								
0.8	0	− 0.00039134	0.01274311								
0.8	1	0.26636627	0.02143281								
0.8	2	0.52301696	0.0437993								
0.8	3	0.74242164	0.07303372								
0.8	4	0.92167763	0.10571765								
0.8	5	106.355444	0.13960275								
0.8	6	117.559163	0.17372668								

Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd	Mach	AoA	Cl	Cd
0.8	7	126.644313	0.20800433								
0.8	8	134.263925	0.2425298								
0.8	9	140.730755	0.27724884								
0.8	10	146.190791	0.31196262								
0.8	11	150.691821	0.34627601								
0.8	12	154.324379	0.37997316								
0.8	13	15.717274	0.4128407								
0.8	14	159.277154	0.44462681								
0.8	15	160.857777	0.47559143								
0.8	16	161.758798	0.50501055								
0.8	17	161.780059	0.53199262								
0.8	18	161.357461	0.55722501								
0.8	19	159.712192	0.57775063								
0.8	20	155.842117	0.58879609								
0.9	0	0.00199354	0.1153863								
0.9	1	0.10756093	0.11727129								
0.9	2	0.2127823	0.12281292								
0.9	3	0.31806577	0.13199071								
0.9	4	0.42396218	0.14466949								
0.9	5	0.53078849	0.16072812								
0.9	6	0.63790546	0.17989125								
0.9	7	0.74522719	0.20218415								
0.9	8	0.85139603	0.22794191								
0.9	9	0.95385464	0.25707932								
0.9	10	104.954.578	0.28910502								
0.9	11	113.714.946	0.32341146								
0.9	12	121.713735	0.35959466								
0.9	13	129.028096	0.39740223								
0.9	14	135.744052	0.43663143								
0.9	15	141.910384	0.47707385								
0.9	16	147.571004	0.51856196								
0.9	17	152.760376	0.56095865								
0.9	18	157.508078	0.60414505								
0.9	19	161.839437	0.64799499								
0.9	20	16.577327	0.69238254								

References

1. Flightpath 2050 Europe's Vision for Aviation. https://www.acare4europe.org/sites/acare4europe.org/files/document/Flightpath2050_Final.pdf
2. Lee T-S et al (2006) Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. Comput Stat Data Anal 50(4):1113–1130
3. Kumar K, Bhattacharya S (2006) Artificial neural network vs linear discriminant analysis in credit ratings forecast: a comparative study of prediction performances. Rev Account Finance 5(3):216–227
4. Birgul E (2003) Stock market prediction using artificial neural networks. Decis Support Syst 22:171–185
5. Kim K, Won BL (2004) Stock market prediction using artificial neural networks with optimal feature transformation. Neural Comput Appl 13(3):255–260

6. Yamashita T, Kotaro H, Jinglu H (2005) Application of multi-branch neural networks to stock market prediction. In: Proceedings of 2005 IEEE international joint conference on neural networks, vol. 4. IEEE
7. Jeong S, Murayama M, Yamamoto K (2005) Efficient optimization design method using kriging model. J Aircr 42(2):413–420
8. Kanazaki M et al (2006) Multi-objective aerodynamic optimization of elements' setting for high-lift airfoil using kriging model. In: 44th AIAA aerospace sciences meeting and exhibit
9. Kumano T et al (2006) Multidisciplinary design optimization of wing shape for a small jet aircraft using kriging model. In: 44th AIAA aerospace sciences meeting and exhibit
10. Liu J, Zhonghua H, Wenping S (2012) Efficient kriging-based aerodynamic design of transonic airfoils: some key issues. In: 50th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition
11. Kuya Y et al (2011) Multifidelity surrogate modeling of experimental and computational aerodynamic data sets. AIAA J 49(2):289–298
12. Han Z-H, Ralf Z, Stefan G (2010) A new cokriging method for variable-fidelity surrogate modeling of aerodynamic data. In: 48th AIAA Aerospace sciences meeting including the new horizons forum and aerospace exposition
13. Toal T, David J, Andy JK (2011) Efficient multipoint aerodynamic design optimization via cokriging. J Aircr 48(5):1685–1695
14. Rumpfkeil MP (2012) Optimizations under uncertainty using gradients, Hessians, and surrogate models. AIAA J 51(2):444–451
15. Palar PS, Koji S (2017) Multi-fidelity uncertainty analysis in CFD using hierarchical kriging. In: 35th AIAA applied aerodynamics conference
16. Secco NR, Bento SM (2017) Artificial neural networks to predict aerodynamic coefficients of transport airplanes. Aircr Eng Aerosp Technol 89(2):211–230
17. Oktay T, Çelik H, Uzun M (2017) A novel learning algorithm to estimate the optimum fuselage drag coefficient. Sakarya Univ J Sci 21(1):63–68
18. Luo C et al (2015) Adaptive space transformation: an invariant based method for predicting aerodynamic coefficients of hypersonic vehicles. Eng Appl Artif Intell 46:93–103
19. Andrés E et al (2012) Efficient aerodynamic design through evolutionary programming and support vector regression algorithms. Expert Syst Appl 39(12):10700–10708
20. Andrés-Pérez E et al (2016) Aerodynamic shape design by evolutionary optimization and support vector machines. Application of surrogate-based global optimization to aerodynamic design. Springer, Cham, pp 1–24
21. Boutemedjet A et al (2019) UAV aerodynamic design involving genetic algorithm and artificial neural network for wing preliminary computation. Aerosp Sci Technol 84:464–483
22. Sessarego M et al (2016) Aerodynamic wind-turbine rotor design using surrogate modeling and three-dimensional viscous–inviscid interaction technique. Renew Energy 93:620–635
23. Predić B et al (2018) Implementation of computationally efficient Taguchi robust design procedure for development of ANN fuel consumption prediction models. Transport 33(3):751–764
24. Tartaruga I et al (2015) Efficient prediction and uncertainty propagation of correlated loads. In: 56th AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference
25. Martinez-Cantin R (2018) Funneled Bayesian optimization for design, tuning and control of autonomous systems. IEEE Trans Cybern 99:1–12
26. Hebbal A et al (2019) Multi-objective optimization using deep Gaussian processes: application to aerospace vehicle design. AIAA Scitech 2019 Forum
27. Gaier A, Alexander A, Jean-Baptiste M (2017) Aerodynamic design exploration through surrogate-assisted illumination. In: 18th AIAA/ISSMO multidisciplinary analysis and optimization conference
28. Kumar A, Ajoy KG (2018) Decision tree- and random forest-based novel unsteady aerodynamics modeling using flight data. J Aircr 56(1):403–409
29. Hu G, Kwok KCS (2019) Predicting wind pressures around circular cylinders using machine learning techniques. arXiv preprint 1901.06752
30. Bertrand X, Frédéric T, Steeve C (2019) Wing airfoil pressure calibration with deep learning. AIAA Aviation 2019 Forum
31. Li K, Kou J, Zhang W (2019) Deep neural network for unsteady aerodynamic and aeroelastic modeling across multiple Mach numbers. Nonlinear Dyn 96(3):2157–2177
32. Bhatnagar S et al (2019) Prediction of aerodynamic flow fields using convolutional neural networks. Comput Mech 1–21
33. Duvall J, Karthik D, Yaser A (2019) Prediction of aerodynamic flow fields using spectral convolutions on graph networks. Bull Am Phys Soc
34. Zhang Y, Woong JS, Dimitri NM (2018) Application of convolutional neural network to predict airfoil lift coefficient. In: AIAA/ASCE/AHS/ASC structures, structural dynamics, and materials conference
35. Sekar V et al (2019) Inverse design of airfoil using a deep convolutional neural network. AIAA J 57(3):993–1003
36. Tikhonov AN (1943) "Об устойчивости обратных задач" [On the stability of inverse problems]. Dokl Akad Nauk SSSR 39(5):195–198
37. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc Ser B (Methodological) 58(1):267–288
38. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407–499
39. Tipping ME (2001) Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 1:211–244
40. Huber PJ (1964) Robust estimation of a location parameter. Ann Stat 53(1):73–101
41. Hastie T, Tibshirani R, Friedman J (2008) The elements of statistical learning, 2nd edn. Springer
42. Ho TK (2002) A data complexity analysis of comparative advantages of decision forest constructors (PDF). Pattern Anal Appl 5(2):102–112
43. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63:3–42
44. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Netw 61:85–117
45. Ortiz-García E, Salcedo SS, Pérez-Bellido A, Portilla-Figueras J (2009) Improving the training time of support vector regression algorithms through novel hyper-parameters searchspace reductions. Neurocomputing 72:3683–3691
46. Abbott IH, Von Doenhoff AE (1959) Theory of wing sections, including a summary of airfoil data. Courier Corporation
47. Cook PH, McDonald MA, Firmin M (1979) Wind tunnel measurements of the mean flow in the turbulent boundary layer and wake in the region of the trailing edge of a swept wing at subsonic speeds. RAE-TR-79062
48. Epstein B et al (2009) Comparative study of three-dimensional wing drag minimization by different optimization techniques. J Aircr 46(2):526–541
49. http://tau.dlr.de/fileadmin/documents/meetings/2008/pdf/1st-day-theory/Tau-Overview.pdf
50. https://www.centaursoft.com/grid-generator