Inferring Functional Properties from Fluid Dynamics Features

Andrea Schillaci, Maurizio Quadrio
Dipartimento di Scienze e
Tecnologie Aerospaziali
Politecnico di Milano
Milano, Italy
Email: andrea.schillaci@polimi.it
maurizio.quadrio@polimi.it

Carlotta Pipolo
Unità di Otorinolaringoiatria
ASST Santi Paolo e Carlo,
Dip. Scienze della Salute
Milano, Italy
Email: carlotta.pipolo@unimi.it

Marcello Restelli, Giacomo Boracchi
Dipartimento di Elettronica,
Informazione e Bioingegneria
Politecnico di Milano
Milano, Italy
Email: marcello.restelli@polimi.it
giacomo.boracchi@polimi.it

***Abstract*—In a wide range of applied problems involving fluid flows, Computational Fluid Dynamics (CFD) provides detailed quantitative information on the flow field, at variable level of fidelity and computational cost. However, CFD alone cannot predict high-level functional properties that are not easily obtained from the equations of fluid motion. In this work, we present a data-driven framework to extract these additional information, such as medical diagnostic output, from CFD solutions. This is a challenging task because of the huge data dimensionality of CFD, and the limited training data that can be typically gathered due to the large computational cost of CFD. By pursuing a traditional Machine Learning (ML) pipeline of pre-processing, feature extraction, and model training, we demonstrate that informative features can be extracted from CFD data. Two experiments, pertaining to different application domains, support our claim that the convective properties implicit into a CFD solution can be leveraged to retrieve functional information that does not admit an analytical definition. Despite the preliminary nature of our study and the relative simplicity of both the geometrical and CFD models, for the first time we demonstrate that the combination of ML and CFD can diagnose a complex system in terms of high-level functional properties.**

***Index Terms*—Computational Fluid Dynamics, Data-driven models, Inference, Nasal Breathing Difficulties**

I. Introduction

Computational Fluid Dynamics (CFD), i.e., solving the differential equations of the fluid motion with the aid of a digital computer, plays a crucial role in a large number of applications, ranging from industry to health. Nowadays CFD is relied upon as much as (sometimes more than) the traditional wind-tunnel testing, and its accuracy (determined by the amount of discretization as well as by the models employed) can be increased at will, provided the computational cost remains affordable.

Quite often, however, the final goal of the CFD analysis, i.e. the diagnosis of the system, remains elusive: the ultimate information that is relevant for the end-user might not be directly provided by the CFD itself, or might not be expressed as a function of the CFD solution. In particular, the complex interplay between fluid dynamics and the geometry of interest, prevents us to formulate (and solve) the design of the best geometry as an optimization problem involving CFD outcomes.

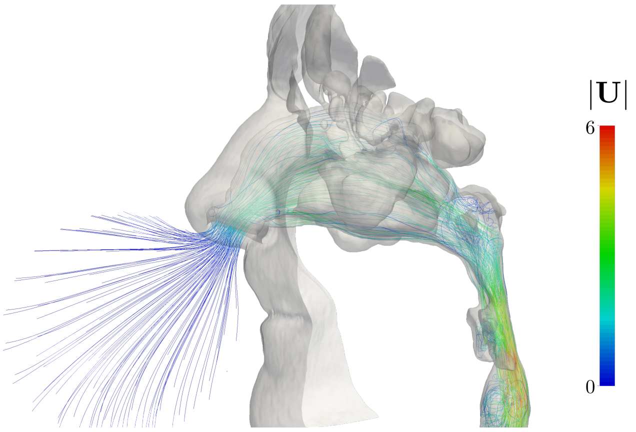

Fig. 1. CFD solution of the airflow in the upper human airways during an inspiration: streamlines are colored with the magnitude of the local velocity in m/s.

Illustrative cases exist in the medical domain [1], and we consider the diagnosis of Nasal Breathing Difficulties (NBD) as a running example. NBD represent an extremely widespread pathological condition of the human upper airways and often requires corrective surgery: a precise diagnosis is troublesome and the failure rate of surgery is up to 50% [2], [3]. A detailed CFD solution for the nasal airflow for a specific patient (see Fig. 1), is certainly important and useful to diagnose NBD, but *per se* it does not help the surgeon to make a rational decision as to whether and how to perform a specific surgical manoeuvre. Several other similar examples could be made, ranging from flood control in rivers, to aerodynamics in the transport sector, to a large number of industrial problems such as probe placement in wind tunnels. In fluid dynamics, the strong non-linearity of the governing equations makes a small geometrical detail potentially result in significant flow changes far away (for example a small imperfection on the wing surface can compromise the aerodynamic performance of the entire aircraft). On the other hand, a large geometrical modification sometimes leads to little or no consequences (for example a large deviation of the nasal septum may be compatible with normal breathing). The diagnosis of these complex systems can benefit from CFD outcomes, as for instance to determine whether and where to perform surgery, where to best prevent coastline erosion, where to optimally place a probe. The answers to these questions are indeed contained within and dictated by the CFD-computed flow field, but an analytical link between the flow field itself and the required information is not available. We believe that pattern recognition techniques [4] and data-driven models in particular [5] have a large potential in this relatively unexplored class of problems.

Using data-driven models on CFD data is particularly challenging for several reasons. First and foremost, there is a dimensionality problem: CFD invariably leads to large data sets, which are costly to produce and difficult to analyse. Such a huge amount of data is not amenable to be directly handled by Machine Learning (ML) models. To set the stage, we mention that a simple two-dimensional CFD simulation of the time-averaged flow field around an airfoil – i.e. a basic configuration of aeronautical interest addressed with the simplest of the CFD approaches – requires the discretization of space into no less than 10^6 cells. Since several flow variables (two velocity components, pressure, auxiliary turbulence variables) are computed for each cell, a single CFD simulation easily produces hundreds of Megabytes of data. This figure grows by orders of magnitude when three-dimensional configurations are considered, and/or higher-fidelity simulations are used. Furthermore, it is very difficult to gather large training sets of annotated simulations, due to their large computational cost and the difficulty of gathering a representative set of experts' decisions in domains such as medicine.

Here we propose a ML methodology to diagnose a complex system whose physics is governed by fluid dynamics. The class of problems we consider relies on the ability of the flow field to convey information, especially of the geometrical type, from an *a priori* unknown location to a predetermined sensing location. Crucially, the success of this endeavour hinges upon the convective properties of the flow. In particular, we aim at using data-driven models to arrive at important information that cannot be computed via the simulation itself, such as a diagnostic output in medicine. We identify and describe low-dimensional features that can be realistically extracted from CFD data and then used in a ML pipeline. These features, namely the field values measured at predefined locations or streamlines arrival time, will be demonstrated to be effective in two different application scenarios, where they enable accurate inference of the target variable even with rather small training sets. Since, to the best of our knowledge, no CFD dataset on parametric geometries is publicly available to date, we develop two case studies in distant application domains: studying the airflow in the human nose, and the airflow around a two-dimensional section of an airplane wing. The airfoils dataset is publicly available for download at https://doi.org/10.5281/zenodo.4106752.

For reasons related to the computational cost of creating the database, both experiments are quite simplified in terms of geometric and CFD models, without compromising the validity of the ML procedure. Both problems share an identical structure, insofar as the interest lies in retrieving non-local information (pathological anatomic anomalies of the airways, or shape characteristics of the profile) from simple features extracted from the computed flow field.

II. Related work

In the last 5-10 years, the application of ML to fluid mechanics has bloomed. This is witnessed by the quantity and quality of the published material. Recent researches and authoritative surveys can be found in [6]–[10]. Most often ML is used to model fluid equations using CFD as input (or, equivalently, physical realizations of a flow), and expecting fluid mechanical quantities as output. Hence ML models are often used to predict the complex input-output relationship typical of fluid flows governed by highly non-linear equations. To the best of our knowledge, however, there is no previous work that shares our goal of inferring quantities that cannot be computed by the CFD itself.

A clearly identifiable strand of work aims at improving turbulence models [8], which is needed in CFD approaches where the small-scale details and the unsteady behaviour of a turbulent flow cannot be computed. Indeed, a universal and accurate turbulence model is still lacking. Recent developments are leading to bound uncertainties in existing turbulence models via physical constraints and to adopt statistical inference to characterize the empirical coefficients of existing models. Among the several examples, Ling et al. [11] were the first to employ a deep neural network to enforce a correction to the popular Spalart-Allmaras RANS turbulence model [12], by embedding the required Galilean invariance into the model-predicted tensor of the turbulent stresses. Along similar lines, Wang et al. [13] used random forests to identify large discrepancies in model-based turbulent stresses. Fukami *et al.* [14] applied supervised ML to solve a number of regression problems for reconstruction and estimation. Example applications were the estimation of time-varying force coefficients and flow reconstruction from a limited number of sensors. They also considered convolutional neural networks for super-resolution, training the ML model with direct numerical simulations to extract key features from the training data.

Another class of works attempts to bypass the use of the differential equations that govern the fluid motion to get rid of the simulation stage altogether. For example, a physics-informed deep-learning framework was developed [15] to learn the velocity and pressure fields from the flow visualizations; it shows potential also for biomedical applications, in cases where quantitative measurements are unavailable. Srinivasan *et al.* [16] illustrated the potential of neural networks to predict the dynamical evolution of a simple model of a temporally-evolving turbulent shear flow, training multilayer perceptron and long short-term memory networks.

It is important to note that in all the aforementioned works the fluid dynamics quantities are used as both input *and* output of the ML algorithm. In other words, ML is typically used as a surrogate of the Navier–Stokes governing differential equations, either to speed up or replace the computation, or to improve the turbulence modeling required by CFD.

III. Problem Formulation

The output of a CFD simulation is a set of scalar or vector fields defined over a domain $\Omega \subset \mathbb{R}^3$ which in CFD always undergoes discretization, for example into many small volumes or a computational mesh. These fields are obtained by solving the discretized Navier–Stokes equations (sometimes in a simplified form supplemented by a turbulence model) together with boundary conditions applied at the geometrical boundary $\Gamma \subset \mathbb{R}^3$. For instance, for the human nose, Γ includes the internal geometry of the nasal cavities extracted from the CT scan of the patient, as shown in Fig. 1.

A CFD simulation results in several output fields, which in general are also time-dependent. However, the present work only considers time-averaged quantities, in particular the vector field of the mean velocity $\mathbf{U}(x, y, z)$ and the scalar field of the mean pressure $p(x, y, z)$ (expressed in a Cartesian reference system without loss of generality):

$$\mathbf{U}(x,y,z) = \begin{bmatrix} u(x,y,z) \\ v(x,y,z) \\ w(x,y,z) \end{bmatrix}, \quad p(x,y,z). \tag{1}$$

All the flow quantities referring to the generic i-th cell resulting from the discretization of Ω can be stacked into a vector $\mathbf{Q}_i \in \mathbb{R}^4$:

$$\mathbf{Q}_i = \begin{bmatrix} u_i \\ v_i \\ w_i \\ p_i \end{bmatrix}, \tag{2}$$

where for conciseness $u_i = u(x_i, y_i, z_i)$ being $(x_i, y_i, z_i) \in \Omega$ the cell center. Since the spatial domain Ω is discretized over n cells, which in our elementary case studies is already $n \sim 10^6$, the CFD output is a (very large) matrix $\mathbf{C} \in \mathbb{R}^{4\times n}$, which contains all the flow quantities in every cell.

Our goal is to train a model $\mathcal{K}$ that predicts a target value Y associated to the matrix $\mathbf{C}$ provided by CFD:

$$\mathcal{K} : \mathbf{C} \mapsto Y. \tag{3}$$

The target variable can be either categorical (as for a classifier that identifies the most suitable surgery for NBD), or ordinal/real (as for a regressor that estimates some geometric quantities from Γ). To this purpose, we assume that a training set of l labelled pairs $\{(\mathbf{C}_j, Y_j), j = 1, \dots, l\}$ is provided.

The major challenges to be addressed in our settings are *i*) the large dimensionality of each input (namely large n); and *ii*) the limited number of training samples l, due to the high computational cost of each CFD simulation. To tackle the latter challenge, we opted for a computationally cheap CFD approach – i.e. solving the Reynolds Averaged Navier–Stokes (RANS) equations. The available alternatives would lead to a prohibitive computational cost for dataset generation, even though more accurate results may contain additional important information. RANS equations are fast to solve (around 10-12 computing hours per case in our simple 3D application), but they only provide information on the mean fields.

IV. Proposed solution

We describe now our approach for training a model and performing inference over the CFD output $\mathbf{C}$. It consists of a concatenation of rather customary steps of ML pipelines [4], namely *pre-processing*, *feature extraction*, and *model training*; however, the first two steps are customized to CFD data and are therefore described in detail below.

A. Pre-processing

The CFD output $\mathbf{C}$ is first pre-processed to compute *streamlines*. By definition, streamlines are locally tangent to the velocity vector and can be thought of as massless tracer paths. A number of streamlines is drawn connecting a start region $\mathcal{S} \subset \Omega$ to an end region $\mathcal{E} \subset \Omega$. For example, Fig. 2 shows streamlines for the nasal airflow starting from $\mathcal{S}$, a spherical surface placed in front of the nostrils, and ending at $\mathcal{E}$, a plane crossing the downstream end of the computational domain, beneath the larynx. Fig. 3 shows the streamlines pattern for the two-dimensional flow around an airfoil: in this case, $\mathcal{S}$ is a vertical line upstream of the profile and $\mathcal{E}$ is a similar line placed downstream. Streamlines provide a compact view of the flow field in the domain Ω, and can highlight vortical structures, recirculation zones, and high-velocity regions (where the streamlines approach each other).

Each streamline is defined by its tangent, which is locally parallel to the velocity vector $\mathbf{U}$. Hence, once the velocity field is known, streamlines are computed by selecting s locations over the region $\mathcal{S}$, and by numerically integrating their trajectory. In detail, we set an initial location for the k-th streamline $({x_0}^k, {y_0}^k, {z_0}^k) \in \mathcal{S}$ and initialize its velocity as $\mathbf{U}({x_0}^k, {y_0}^k, {z_0}^k)$. Then, trajectory is integrated until the end region $\mathcal{E}$ is reached; the velocity $\mathbf{U}$ is obtained by linear interpolation out of the mesh nodes.

B. Feature Extraction

Due to its large size, the CFD output $\mathbf{C}$ cannot be fed to the classifier $\mathcal{K}$ directly. Therefore, we perform feature extraction to dramatically reduce the number of inputs of the classifier, while preserving the information content of the CFD. We propose two kinds of expert-driven features, which are inspired by engineering practice in the analysis of flow fields: distribution of *streamline arrival times* and *regional averages* of flow variables.

Distribution of Streamline Arrival Times: Once the s streamlines connecting $\mathcal{S}$ to $\mathcal{E}$ have been computed, we measure the time required to travel from $\mathcal{S}$ to $\mathcal{E}$ along each streamline at the local mean velocity. The arrival times are then considered as realizations of a random variable with unknown distribution, of which we estimate mean μ_1 and centered moments up to fifth order, i.e. $\mu_2, \dots, \mu_5$. The statistics of the arrival times provide an extremely compact and meaningful description of the flow. For example, streamlines entangled by vortices would take longer to reach $\mathcal{E}$ than straight streamlines; similarly, streamlines passing through highly turbulent regions would result in outliers with respect to the distribution of normal trajectories. Besides arrival times, additional quantities

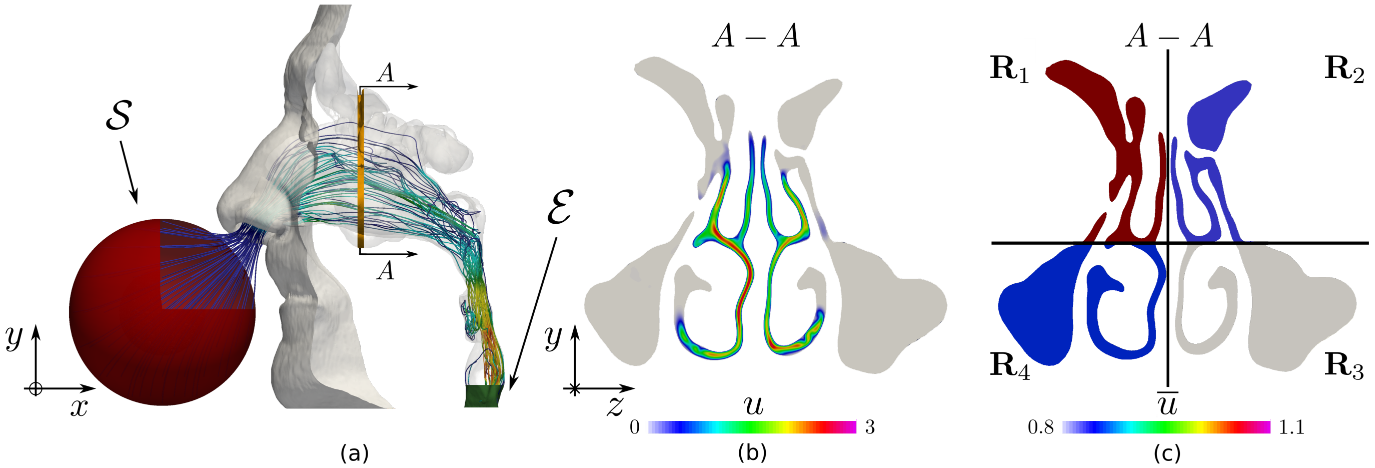

Fig. 2. Airflow in the human nasal cavities during inspiration. (a) Streamlines start from region $\mathcal{S}$ and end in region $\mathcal{E}$. The orange slice indicates the cross-sectional cut plotted in panels (b) and (c). (b) Mean velocity component normal to the cross-sectional cut. (c) Division of the plane in 4 regions $\{\mathbf{R}_{1-4}\}$, colored with the value of the regional average velocity $\overline{u}_k$.

can be extracted from streamlines, e.g. by integrating flow quantities (like velocity or pressure) along the streamlines and computing the sample moments of their distribution.

Features extracted from streamlines are very practical, since they compactly convey flow information while sampling most of the volume Ω with minimal knowledge of the geometry Γ. In fact, only the initial and final regions $\mathcal{S}$ and $\mathcal{E}$ need to be identified: no accurate registration is required for the rest of the surface.

Regional Averages: Other informative features can be extracted by averaging the flow quantities over r pre-defined regions $\mathbf{R}_k \subset \mathbf{\Omega}, k = 1, \cdots, r$. To take into account the uneven layout of the samples in Ω, these averages are volume-weighted. For example, the region-averaged pressure p over region $\mathbf{R}_k$ is referred to as $\overline{p}_k$ and is defined as

$$\overline{p}_k = \frac{\sum_i p_i V_i}{\sum_i V_i} \tag{4}$$

where the index i includes all the cells $(x_i, y_i, z_i) \in \mathbf{R}_k$, and V_i denotes their volumes.

Fig. 2(a) illustrates a thin orange slice $(A - A)$ as a meaningful choice for a set $\{\mathbf{R}_k\}$. This coronal section (Fig. 2(b)) intersects large areas exhibiting little or no flow (the paranasal sinuses), as well as narrower areas delimited by the turbinates, where most of the flow rate is concentrated. Fig. 2(c) shows how this section has been divided into four regions ($k = 1, \ldots, 4$), with the color indicating the computed value $\overline{u}_k$ in each region.

Information conveyed in regionally-averaged features obviously depends on whether the set of selected regions $\{\mathbf{R}_k\}$ is meaningful. The selection of these regions might not be straightforward in the medical domain, where $\mathbf{R}_k$ typically refers to landmarks that cannot be detected automatically or that require sophisticated registration procedures to align the input surface Γ with a common reference where regions can be defined.

Regional averages mimic procedures often used in wind-tunnels measurement campaigns, where probes like hot-wire anemometers or Pitot tubes are placed in the flow beforehand. Our experiments suggest that averages over a few significant regions in Ω might be discriminative enough to solve our inference problems.

C. Model training

The pre-processing and feature-extraction steps map the output $\mathbf{C} \in \mathbb{R}^{4\times n}$ of each CFD to a feature vector $\mathbf{f} \in \mathbb{R}^m$, which stacks m features being either the streamline moments or the regional averages of velocity and pressure. Overall, we expect $m \ll 4 \times n$, so that these two steps yield a substantial reduction in the dimensionality of the problem. Depending on the nature of the target variables, any classifier or regressor $\mathcal{K}$ can be trained from the set of labeled feature vectors $\{(\mathbf{f}_j, \mathbf{Y}_j), j = 1, \cdots, l\}$. In the experiments described below, we adopt Neural Networks trained to perform regression over the space of target variables and we show that a limited number of features is often enough to provide very accurate predictions.

V. Experiments

We describe two experiments to show that a handful of informative features are sufficient to infer quantities that cannot be computed directly from a CFD simulation. To demonstrate the flexibility of the method presented in Section IV, the two case studies belong to distant application domains: prediction of geometrical parameters of an airfoil (subsection V-A) and prediction of the severity of an anatomical anomaly of a human nose (subsection V-B). From a fluid-dynamic perspective, the two case studies are far away from each other: the airfoil case is two-dimensional and involves an external fully turbulent flow, in which the inertia forces dominate. The human nose case is three-dimensional and involves an internal, mostly laminar or transitional flow. However, in both cases the goal is to retrieve geometrical information from far away CFD data.

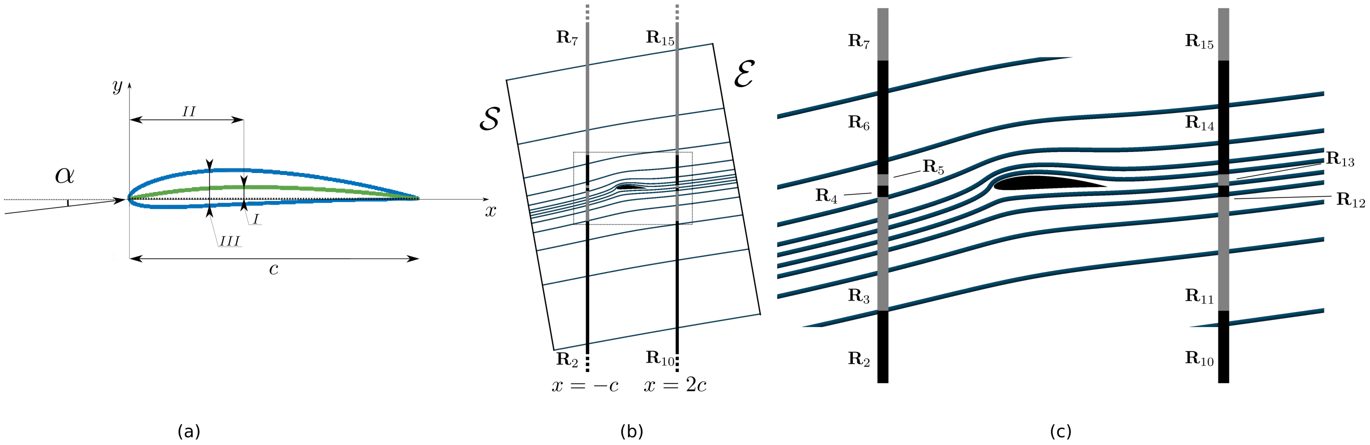

Fig. 3. Flow field around an airfoil at incidence (flow is from left to right). (a) Sketch of the airfoil, indicating chord c (the segment connecting the leading edge to the trailing edge), angle of incidence α formed between chord and free-stream velocity, the leading edge at $x = 0$, and the trailing edge at $x = c$. The green line is the camber line. First number of NACA code: maximum camber I. Second number of the NACA code: position II of maximum camber along the cord. Third number of the NACA code: maximum thickness III. (b) Streamlines connecting start region $\mathcal{S}$ to end region $\mathcal{E}$, with part of the regional sets $\mathbf{R}_k$ (which in a two-dimensional case reduce to lines). (c) Zoom around the airfoil. Smaller regions around $y = 0$ like $\{\mathbf{R}_{4-5}\}$ can be appreciated.

The numerical simulations are carried out with OpenFOAM [17], a popular open-source C++ CFD toolbox. We choose the most simple and computationally affordable CFD approach by solving the Reynolds-Averaged Navier–Stokes (RANS) equations using the Spalart-Allmaras [12] turbulence model to generate the airfoil dataset, and the $k - \omega$ SST turbulence model [18] to generate the human nose dataset.

Overall, the best features are found to be the regional averages, with accuracy varying according to the distance between $\{\mathbf{R}_k\}$ and the geometry of interest. Table I shows that in the airfoil dataset, the overall accuracy exceeds 95% when the regional sets are not too far away from the profile.

A. *Prediction of Geometrical Features of an Airfoil*

1) Dataset and task: We consider a popular family of airfoils four digit NACA (National Advisory Committee for Aeronautics). Our goal is to train a multivariate regressor $\mathcal{K}$ to predict the NACA numbers, i.e. the shape of the airfoil itself, starting from the CFD solution.

The shape of a NACA airfoil is described by their four-digits code, which corresponds to three integer numbers, and the length of the chord c (see Fig. 3 a). The first number in the NACA code corresponds to the first digit (integer, [0-9]) and quantifies the maximum camber of the airfoil in units of $c/100$; the second number corresponds to the second digit (integer, [0-9]) and locates the point of maximum camber along the chord measured from the leading edge, expressed in $c/10$; the third number has two digits (integer, [05-50]) and quantifies the maximum thickness of the airfoil expressed in $c/100$.

The two-dimensional CFD domain Ω is centered on the airfoil and has a radius larger than $500c$; the angle of incidence α (Fig. 3 a) is set at 10 degrees, the free-stream velocity is 30 m/s. A database of CFD solutions is built by considering 3025 different combinations of digits, hence 3025 different airfoil shapes.

2) Feature Extraction: Streamlines connecting $\mathcal{S}$ to $\mathcal{E}$ are shown in Fig.3 (a). $\mathcal{S}$ is a straight segment of length $10c$ orthogonal to the free-stream velocity, whose center is $3c$ distant from the leading edge; $\mathcal{E}$ is identical to $\mathcal{S}$ with center shifted $3c$ downstream from the trailing edge. Along $\mathcal{S}$, the streamlines starting points $({x_0}^k, {y_0}^k, {z_0}^k)$ are non-uniformly spaced, with finer spacing towards the center, as shown in panel (c) of Fig. 3.

To extract region-averaged flow quantities, 24 regions $\{\mathbf{R}_k\}$ are selected, consisting of eight portions of three vertical lines drawn perpendicular to the airfoil chord. The first eight segments for $1 \leq k \leq 8$ lay on a vertical line placed at $x = -c$ upstream of the airfoil; eight segments for $9 \leq k \leq 16$ lay on a vertical line placed $1c$ downstream (Fig. 3 b,c), and the eight segments for $17 \leq k \leq 24$ lay on a vertical line placed $10c$ downstream the airfoil trailing edge. On each segment, the regions are symmetrically placed with respect to $y = 0$, and their boundaries have y coordinates of $[-500, -10, -1, -0.1, 0, 0.1, 1, 10, 500]$. Note that in Fig. 3 the most rearward segment and the regions farthest from the profile are not displayed.

3) Model Training and Performance Assessment: We train a three-layers neural network to estimate the three numbers in the NACA code. This is a regression network with 3 output neurons, one per each number of the NACA code. Since the estimated numbers are not necessarily integers, they are rounded to yield the output code. We adopt different splitting criteria in training and test set, considering both interpolation (Table I) and extrapolation (Table II). As a figure of merit, we primarily consider $|e|$, the mean absolute error over each estimated code and also the classification accuracy a, the percentage of correctly estimated codes.

4) k-fold cross-validation experiment: The goal of this experiment is to identify the most informative features and assess our regression performance when varying the dimension of the training set. Features are initially grouped according

TABLE I

INTERPOLATION EXPERIMENTS FOR THE AIRFOIL DATASET. TRAINING SET DIMENSION FOR REGIONAL AVERAGES: 484, TRAINING SET DIMENSION FOR STREAMLINES: 2000.

Features	I		II		III		a
	$\|e\|$	σ	$\|e\|$	σ	$\|e\|$	σ	[%]
μ_{1-5}	0.24	1.16	0.41	1.16	0.89	11.54	60.79
$\overline{p}_{1-8}$	0.04	0.30	0.06	0.29	0.03	0.16	99.34
$\overline{v}_{1-8}$	0.04	0.30	0.06	0.21	0.04	0.39	97.45
$\overline{p}_{9-16}$	0.06	0.16	0.11	0.31	0.06	0.16	96.39
$\overline{v}_{9-16}$	0.03	0.32	0.04	0.13	0.04	0.54	99.47
$\overline{p}_{17-24}$	0.15	0.37	0.27	0.70	0.15	0.70	86.25
$\overline{v}_{17-24}$	0.15	0.43	0.26	0.60	0.12	0.29	85.71

to classical fluid dynamics practices, and are then selected by performing a 5-fold split over the whole training set. In particular, we select three sections, up and downstream the airfoil, where to extract features from pressure and velocity measures.

Table I shows the mean absolute error $|e|$ and the standard deviation σ for each NACA number, as well as the classification accuracy a. The network is trained by minimizing the mean square error of the estimated NACA numbers. When 8 regionally-averaged flow features are used with a training set of only 484 airfoils, the neural network achieves very small absolute errors and an overall accuracy between 85% and 99% on the NACA code (cfr. last column of Table I). The relatively large range in accuracy suggests that some regional averages are more informative than others. In particular, regions closer to the airfoil like $\overline{p}_{1-8}$, $\overline{v}_{1-8}$ (located at $x = -c$) and $\overline{p}_{9-16}$, $\overline{v}_{9-16}$ (located at $x = 2c$) achieve higher prediction scores than those further away (like $\{\mathbf{R}_{13-18}\}$ placed at $x = 11c$). This is not surprising since all the flow variables become more uniform as the distance from the airfoil increases: thus, spatial information conveyed by each flow variables decreases with the distance from the airfoil. The statistical moments of arrival times provide fairly good predictive capabilities too, especially for the first number in the NACA code. Even with a training set of 2000 airfoils, $|e|$ is relatively low for the second and third NACA numbers.

Based on these results, we restrict to regional average features extracted from pressure for studying how the performance varies as a function of the training set size. The above experiment is repeated by progressively reducing the training set size, to investigate how this solution would perform when – owing to their computational cost – only a few CFD simulations are available for training. We split the dataset into N equal segments and separately perform training and testing on each segment through a 5-fold cross validation. This procedure allows us to reliably compute the standard deviation of the regression error.

Fig. 4 illustrates the accuracy of the network classification as a function of training set size, with features $\overline{p}_{1-8}$ and $\overline{p}_{9-16}$, and indicates that about 300 training samples are enough to achieve 90% accuracy. This plot confirms that, at least when the training set is small, features located downstream ($x = 2c$) are more informative than those upstream ($x = -c$) at the same distance.

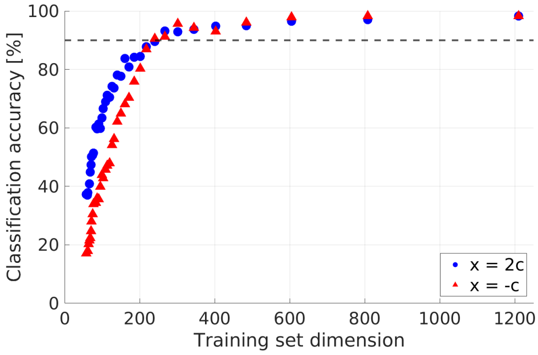

Fig. 4. Classification accuracy a versus dimension of the training set, with features $\overline{p}_{1-8}$, measured on regions at $x = -c$, and $\overline{p}_{9-16}$ measured at $x = 2c$

5) Extrapolation: In this experiment we assess the model performance at predicting NACA numbers that are out of the range of training samples. All the entries corresponding to a subrange of the third NACA number, which has a range of 05–50, are removed from the training set. In the first experiment, we test the range 30–40, and in the second experiment, we skip an internal subrange testing 05–15 and 45–50. Every experiment is carried out 5 times, to average the results. Table II shows that the first extrapolation experiments are very close to the previous k-fold cross validation tests, even though the training set is four times larger than in the k-fold cross-validation case (table I). Little difference is observed when velocity or pressure are chosen as a feature, with the far downstream regions at $x = 11c$ consistently performing slightly worse than the other two regions. The second experiment is obviously more extreme. Velocity seems to be more informative than pressure as a feature. The far regions at $x = 11c$ lead to worse performance than the others placed closer to the airfoil.

B. Prediction of pathologies in a simplified human nose

1) Dataset and task: Fig. 5 (a-c) illustrates the simplified model used to build the CFD database for the human nose. This model replicates all the essential features of a human nose as represented in Fig.1 and 2, but at the same time involves a CAD-based simplified shape which, for example, does not include paranasal sinuses. A key advantage of the simplified CAD model is its parametrization, which is used to implement controlled variations of the basic anatomy. The CFD dataset has been created by defining and modifying 7 geometrical parameters of the baseline model. These parameters mimic anatomical variations observed by Ear-Nose-Throat (ENT) doctors in their clinical practice. In particular, four of them result in "healthy" anatomical alterations of the human noses, namely that ENT doctors deem not to affect the normal breathing function. The other three parameters mimic pathological conditions at different levels of severity. These are

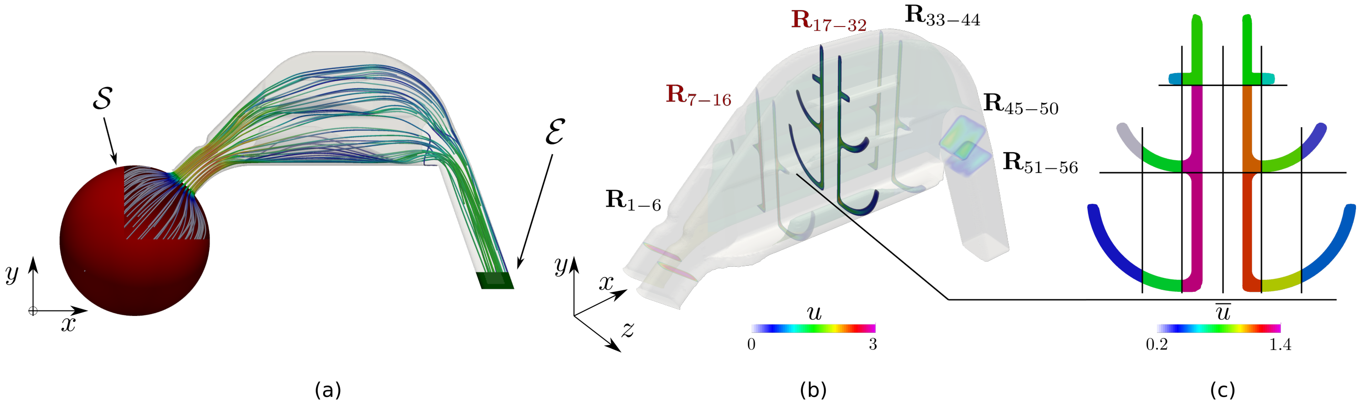

Fig. 5. Simplified model of the human nose. (a) CAD geometry which excludes paranasal sinuses, and placement of regions $\mathcal{S}$ and $\mathcal{E}$. (b) Cut planes for regional averages. Pathologies, if present, are applied in the space between the sections highlighted in red. (c) Regional averages of the x velocity component in the region set $\{\mathbf{R}_{17-32}\}$.

TABLE II
EXTRAPOLATION EXPERIMENTS FOR THE AIRFOIL DATASET. TRAINING SET DIMENSION: 2400

	Features	I		II		III		a
		$\lvert e\rvert$	σ	$\lvert e\rvert$	σ	$\lvert e\rvert$	σ	[%]
Inner	$\overline{p}_{1-8}$	0.03	0.04	0.07	0.08	0.03	0.04	99.97
	$\overline{v}_{1-8}$	0.03	0.06	0.06	0.15	0.03	0.05	98.36
	$\overline{p}_{9-16}$	0.06	0.08	0.08	0.12	0.06	0.08	99.79
	$\overline{u}_{9-16}$	0.06	0.08	0.08	0.15	0.05	0.08	98.83
	$\overline{v}_{9-16}$	0.05	0.08	0.07	0.45	0.05	0.08	98.83
	$\overline{p}_{17-24}$	0.1	0.14	0.19	0.28	0.1	0.15	92.83
	$\overline{v}_{17-24}$	0.08	0.12	0.16	0.24	0.09	0.12	95.34
Outer	$\overline{p}_{1-8}$	0.55	1.75	1.22	3.63	1.54	6.37	76.92
	$\overline{v}_{1-8}$	0.12	1.00	0.14	0.68	0.19	1.30	95.27
	$\overline{p}_{9-16}$	0.69	2.47	1.58	5.34	1.52	6.06	75.24
	$\overline{u}_{9-16}$	0.21	1.02	0.27	0.82	0.29	1.66	85.45
	$\overline{v}_{9-16}$	0.15	0.92	0.20	1.15	0.19	1.04	92.42
	$\overline{p}_{17-24}$	1.72	5.81	4.39	15.88	1.54	5.27	52.33
	$\overline{v}_{17-24}$	0.41	1.29	0.81	2.45	0.41	1.32	61.82

the anterior hypertrophy of the Inferior Turbinate, the hypertrophy of the whole Inferior Turbinate, and the hypertrophy of the anterior head of the Middle Turbinate. These parameters affect the shape between the sections labeled in red in Fig. 5 (b). While our CAD model is certainly overly simplified compared to a CT scan of the human nose and the variety of pathologies, the size of the CFD simulations is instead comparable to those that can be derived from a CT scan. To take into account anatomical variability of human noses, the CFD dataset is generated from 200 unique combinations of these 7 parameters. We address the task of estimating the three *pathological* parameters, and to this purpose we train a neural network having 3 hidden layers and 3 output neurons.

2) Feature Extraction: The regional averages are computed over the sections shown in Fig. 5 (b). The six cross-sectional planes are perpendicular to the mean flow and are further subdivided in several regions (from 6 to 16 each, depending on the surface area). The results of the experiments are reported in Table III, in terms of streamlines arrival time and pressure regional averages. Most of the regional averages achieve a small regression error, such as for $\{\mathbf{R}_{17-32}\}$ which lie on a cross plane that directly "sees" the modification of the turbinates. Since, as in the airfoils case, regional averages of velocity are found to perform similarly to regional averages of pressure, they have not been reported.

3) Model Training and Performance Assessment: In a real scenario, there is of course no guarantee to know data directly from the region where the patient's pathology is present, since this is *a priori* unknown. Thus, the most significant results in Table III are those concerning features extracted from regions far from where the pathological alterations have been applied. For example, the regional averages from regions $\{\mathbf{R}_{33-44}\}$, have a mean absolute error varying between 0.0185 and 0.0570 mm, considering that the severity of the pathologies varies with a step of $0.05mm$, it is a good result. Obviously the error is expected to be smaller when CFD information is extracted right from the regions where the pathology is present: for this reason, these values are greyed out in Table III.

This demonstrates that the ML algorithm is actually able

TABLE III
INTERPOLATION EXPERIMENTS FOR THE HUMAN NOSE DATASET. ROWS CORRESPONDING TO FEATURES EXTRACTED ON A PATHOLOGICAL SECTION ARE GREYED OUT

Features	Inf. Turb. Head		Inf. Turb. Body		Middle Turb. Head	
	$\lvert e\rvert[mm]$	σ	$\lvert e\rvert[mm]$	σ	$\lvert e\rvert[mm]$	σ
μ_{1-5}	4.478	5.581	18.556	22.214	6.008	7.3033
$\overline{p}_{1-6}$	0.113	0.181	0.083	0.140	0.087	0.1307
$\overline{p}_{7-16}$	0.023	0.038	0.012	0.022	0.020	0.038
$\overline{p}_{17-32}$	0.017	0.028	0.014	0.023	0.031	0.047
$\overline{p}_{33-44}$	0.019	0.032	0.019	0.032	0.057	0.099
$\overline{p}_{45-50}$	0.034	0.056	0.014	0.026	0.064	0.110
$\overline{p}_{51-56}$	0.038	0.060	0.018	0.029	0.072	0.119

to make accurate predictions, taking advantage of the fluid dynamic ability to transport information along the flow. Indeed the regions close to the throat such as $\{\mathbf{R}_{45-50}\}$ and $\{\mathbf{R}_{51-56}\}$ still produce rather low inference error, taking into account how far these are from the position where pathological alterations have been introduced. In comparison, streamlines arrival times do not achieve good performance, with an error of over 18.56 mm in a reference domain of 12 mm. The hypertrophy of the head of the Middle Turbinate appears to be more difficult to predict; most likely, this is due to the fact that the Middle Turbinate is interested by a smaller fraction of the global flow rate, hence its influence on the overall flow is smaller.

VI. Conclusions

We have demonstrated that ML can effectively predict functional properties of complex fluid mechanical systems, when the knowledge of the flow field does not immediately provide required high-level diagnostic information. We exploit the convective properties of the fluid flow by identifying a small set of informative features extracted from CFD simulations, which provide accurate predictions of geometrical information. The required training sets are relatively small: this is an extremely important characteristic, owing to the large cost of CFD and the difficulty in gathering annotated data from experts, especially in domains such as medicine.

The flexibility of the proposed approach is demonstrated by dealing with two rather simplified examples, pertaining to applications as diverse as industry and health: the airflow around wing sections (where the goal is the prediction of the airfoil type) and the airflow within a model of the human nose (where the goal is to predict pathological anatomic deformation leading to breathing difficulties).

We identify two types of features that are potentially very informative and reconcile the massive dimensionality of a CFD dataset within a ML pipeline. One hinges upon the reconstruction of streamlines in the flow field and the integration of flow quantities along them. The other consists in averages of fluid dynamic variables over suitable regions in the flow field. Their relative merit has been assessed, with regional averages performing better than streamlines, although this is deemed to depend on the type and quality of the CFD analysis. In fact, the steady nature of the CFD simulation used here fails at providing the streamlines with the information required to successfully solve the addressed regression problems. This is particularly apparent in the human nose, where streamlines computed using RANS simulation differ much from the true ones. In contrast, streamlines are more informative in the airfoil scenario, since the flow is essentially steady. We believe that the use of unsteady CFD on an unsteady problem will unlock their full potential.

Ongoing work concerns designing effective features for addressing real-world medical scenarios, where we plan to combine ML and CFD to infer diagnostic information. Furthermore the construction of a more realistic database, using geometries from CT-scans, is ongoing. In particular, we will use our framework for surgery planning in the ENT domain, where high-fidelity and time-resolved CFD simulations will be used to analyze patient-specific CT scans. A wider target consists in adapting our framework to handle measurements derived from experimental fluid mechanics data. This opens plenty of relevant applications, such as identifying anomalies due to damages or detecting ice formation over airfoils.

VII. Acknowledgments

The Authors are grateful to G. Romani and L. Butera for preliminary experimentation with the human nose case. Computing time has been provided by the Italian CINECA Supercomputing Center under the ISCRA-B grant ML-ONOSE.

References

[1] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting clinical events via recurrent neural networks," in *Machine Learning for Healthcare Conference*, 2016, pp. 301–318.

[2] C. Sundh and O. Sunnergren, "Long-term symptom relief after septoplasty," *European Archives of Oto-Rhino-Laryngology*, vol. 272, no. 10, pp. 2871–2875, 2015.

[3] P. Illum, "Septoplasty and compensatory inferior turbinate hypertrophy: long-term results after randomized turbinoplasty," *European archives of oto-rhino-laryngology*, vol. 254, no. 1, pp. S89–S92, 1997.

[4] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[5] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.

[6] J. Kutz, "Deep learning in fluid dynamics," *J. Fluid Mech.*, vol. 814, pp. 1–4, 2017.

[7] M. Brenner, J. Eldredge, and J. Freund, "Perspective on machine learning for advancing fluid mechanics," *Phys. Rev. Fluids*, vol. 4, no. 100501, pp. 1–7, 2019.

[8] K. Duraisamy, G. Iaccarino, and H. Xiao, "Turbulence modeling in the age of data," *Annual Review of Fluid Mechanics*, vol. 51, pp. 357–377, 2019.

[9] S. Brunton, B. Noack, and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annu. Rev. Fluid Mech.*, vol. 52, pp. 477–508, 2020.

[10] M. Raissi, "Deep hidden physics models: Deep learning of nonlinear partial differential equations," *The Journal of Machine Learning Research*, vol. 19, no. 1, pp. 932–955, 2018.

[11] J. Ling, A. Kurzawski, and J. Templeton, "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance," *J. Fluid Mech.*, vol. 807, pp. 155–166, 2016.

[12] P. Spalart and S. Allmaras, "A One-Equation Turbulence Model for Aerodynamic Flows," *AIAA Paper*, no. 1992-0439, 1992.

[13] J.-X. Wang, J.-L. Wu, and H. Xiao, "Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data," *Physical Review Fluids*, vol. 2, no. 3, p. 034603, 2017.

[14] K. Fukami, K. Fukagata, and K. Taira, "Assessment of supervised machine learning methods for fluid flows," *Theor. Comput. Fluid Dyn.*, pp. 1–23, 2020.

[15] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," *Science*, vol. 367, no. 6481, pp. 1026–1030, 2020.

[16] P. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa, "Predictions of turbulent shear flows using deep neural networks," *Physical Review Fluids*, vol. 4, no. 5, p. 054603, 2019.

[17] H. Weller, G. Tabor, H. Jasak, and C. Fureby, "A tensorial approach to computational continuum mechanics using object-oriented techniques," *Computers in Physics*, vol. 12, no. 6, pp. 620–631, 1998.

[18] F. R. Menter, "Two-equation eddy-viscosity turbulence models for engineering applications," *AIAA journal*, vol. 32, no. 8, pp. 1598–1605, 1994.