Machine learning and reduced order modelling for the simulation of braided stent deployment

Beatrice Bisighini[1,2,3], Miquel Aguirre[4,5,1],
Marco Evangelos Biancolini[3], Federica Trovalusci[3],
David Perrin[2], Stéphane Avril[1]* and Baptiste Pierrat[1]

[1]Mines Saint-Étienne, University Lyon, University Jean Monnet, INSERM, Saint-Étienne, France,
[2]Predisurge, Grande Usine Creative 2, Saint-Etienne, France, [3]Department of Enterprise Engineering,
University Tor Vergata, Rome, Italy, [4]Laboratori de Càlcul Numèric, Universitat Politècnica de
Catalunya, Barcelona, Spain, [5]International Centre for Numerical Methods in Engineering (CIMNE),
Gran Capità, Barcelona, Spain

Endoluminal reconstruction using flow diverters represents a novel paradigm for the minimally invasive treatment of intracranial aneurysms. The configuration assumed by these very dense braided stents once deployed within the parent vessel is not easily predictable and medical volumetric images alone may be insufficient to plan the treatment satisfactorily. Therefore, here we propose a fast and accurate machine learning and reduced order modelling framework, based on finite element simulations, to assist practitioners in the planning and interventional stages. It consists of a first classification step to determine *a priori* whether a simulation will be successful (good conformity between stent and vessel) or not from a clinical perspective, followed by a regression step that provides an approximated solution of the deployed stent configuration. The latter is achieved using a non-intrusive reduced order modelling scheme that combines the proper orthogonal decomposition algorithm and Gaussian process regression. The workflow was validated on an idealized intracranial artery with a saccular aneurysm and the effect of six geometrical and surgical parameters on the outcome of stent deployment was studied. We trained six machine learning models on a dataset of varying size and obtained classifiers with up to 95% accuracy in predicting the deployment outcome. The support vector machine model outperformed the others when considering a small dataset of 50 training cases, with an accuracy of 93% and a specificity of 97%. On the other hand, real-time predictions of the stent deployed configuration were achieved with an average validation error between predicted and high-fidelity results never greater than the spatial resolution of 3D rotational angiography, the imaging technique with the best spatial resolution (0.15 mm). Such accurate predictions can be reached even with a small database of 47 simulations: by increasing the training simulations to 147, the average prediction error is reduced to 0.07 mm. These results are promising as they demonstrate the ability of these techniques to achieve simulations within a few milliseconds while retaining the mechanical realism and predictability of the stent deployed configuration.

KEYWORDS

braided stent, beam elements, contact mechanics, reduced order modelling, machine learning

1 Introduction

Intracranial aneurysms (IAs) are local dilations of the arteries in the brain caused by a degenerative weakening of the arterial wall. Saccular, or blister-like, are the most common IAs. Their prevalence among the general population is estimated to be around 2%–3% (Rinkel et al., 1998). With an incidence of 10/100,000 person-years, IAs rupture leads to subarachnoid haemorrhage, a life-threatening type of stroke with high morbidity and mortality (Wiebers, 2003; Vlak et al., 2011). Therefore, when IAs with a diameter larger than 5/7 mm are detected, they are often recommended for early treatment to exclude the aneurysm sac from the cerebral circulation (Rahman et al., 2010). Nowadays, endovascular options have become the preferred intervention thanks to the lower rate of complications with respect to invasive techniques (e.g., clipping) (Liu and Huang, 2015).

Endoluminal reconstruction through flow diverters represents a novel paradigm for the mini-invasive treatment of IAs, as an alternative to endosaccular occlusion through coiling (Durso et al., 2011; Pierot, 2011). Flow diverters are self-expanding devices consisting of a low-porosity braided stent. Thanks to this structure, they are highly flexible and resistant to kinking. Accordingly, they are well suited for tortuous vessels and wide-neck IAs. These stents come in different sizes, in terms of radius and length. Nowadays, surgeons choose the size based only on clinical experience and measurements taken on medical volumetric images (e.g., 3D rotational angiography or computed tomography angiography), acquired shortly before surgery.

However, the configuration assumed by these devices once deployed within the parent vessel is not easily predictable and routine 3D medical images alone may be insufficient to plan the treatment satisfactorily. The related difficulties can lengthen the interventional times. Moreover, higher doses of angiographic contrast agents and anaesthetic drugs are needed. All these factors increase the risk of postoperative complications for the patient. Therefore, there is a compelling need to develop computational models capable of simulating, in real time, the deployment of flow diverters within patient-specific vessels to assist practitioners in the planning and interventional stages (Karmonik et al., 2005).

The mechanical behaviour of braided stents is typically simulated using a finite element (FE) model where beam elements are used to discretise the wires (Auricchio et al., 2011; Shiozaki et al., 2020; Zaccaria et al., 2020; McKenna and Vaughan, 2021); however, only a few studies modelled numerically flow diverters (Ma et al., 2012; Fu and Xia, 2017). Due to the large amount of degrees of freedom (DOFs) and the necessity to solve the contact between the device and the vessel wall, the computational time required by these traditional techniques is very high. To overcome this limitation and make computational models suitable for clinical use in real time, fast virtual stenting (FVS) methods have been reported in the literature (Larrabide et al., 2012; Spranger et al., 2015; Zhong et al., 2016). They predict the stent deployed configuration by simplifying its mechanical behaviour and/or the contact against the vessel wall (e.g., by using simplex deformable meshes, mass-spring models or active contour models).

Reduced order modelling is also gaining interest in computer-aided surgery thanks to its capability of reducing the computational complexity and cost of numerical problems while preserving their inner physics (Niroomandi et al., 2012a; Niroomandi et al., 2012b; Mena et al., 2015; Santo et al., 2020). One of the most powerful and widespread techniques to build reduced-order models (ROMs) is the reduced-basis (RB) method. The RB method is adapted to non-linear problems which need to be solved a large number of times for different parameter values (Hesthaven et al., 2016; Quarteroni et al., 2016). In biomechanics, the parametrization can concern boundary and initial conditions (Chang et al., 2017; Bridio et al., 2022; Girfoglio et al., 2022), loads (Biancolini and Valentini, 2018) and the geometrical domain under investigation (Ballarin et al., 2016; Biancolini et al., 2020; Kardampiki et al., 2022). The latter is sometimes handled with a statistical shape model, which enables the comparison of different anatomies in terms of the same characteristics (Lauzeral et al., 2019; Cosentino et al., 2020; Buoso et al., 2021).

The real-time simulation of such parametrized problems is achieved by exploiting the intrinsic similarities between their solutions. The most important features of the original, full-order model (FOM), i.e., the RBs, can be extracted through proper orthogonal decomposition (POD) from a set of high-fidelity (HF) solutions. The construction of such a dataset and the extraction of the RBs is referred to as offline stage. Thereafter, approximated solutions for unseen parameter values are determined as linear combination of the RBs (online stage). The powerful advantage of these methods is that, if the online stage is completely decoupled from the offline one, the computations performed in the online stage are independent of the dimension of the FOM. RBs-based methods differ in the implementation of the online stage: Intrusive methods rely on a projection onto the RBs space to generate the ROM (Ballarin et al., 2015; Ballarin et al., 2016); non-intrusive methods employ a regression model trained to learn the mapping from parameters to the solution expressed in the RBs space on the HF dataset (Guo and Hesthaven, 2018; Hesthaven and Ubbiali, 2018; Guo and Hesthaven, 2019; Fresca and Manzoni, 2022). Non-intrusive methods outperform intrusive methods in terms of efficiency, as they do not require solving a system of non-linear equations in the online phase, but only evaluating the regression model. However, they require a very large training dataset to ensure accurate solutions. Recently, physics-informed neural networks (PINNs) emerged as a promising alternative (Liu et al., 2020; Buoso et al., 2021; Chen et al., 2021).

In this study, we present the first implementation of a machine learning (ML)-based ROM scheme for the prediction of the stent deployed configuration. To assess its feasibility, we created a parametric synthetic geometry that allows control of the vessel radius, curvature and aneurysm size. Surgical decisions on the deployment site are also considered when creating the HF dataset. If given the deployment conditions the stent does not land inside the aneurysm and is well positioned against the vessel wall, the deployment is considered successful from a clinical perspective. Since there is no clinical need to predict the stent configuration after an unsuccessful deployment, the virtual framework here proposed consists of two steps (**Figure 1**): A first classification step that allows *a priori* determination of whether a simulation will be successful or not, followed by a regression step that provides an approximated solution of the deployed stent configuration. Moreover, in continuation with our previous work (Bisighini et al., 2022), we propose a fast strategy to perform FE simulations of

braided stent deployment to reduce the computational time required to build the HF dataset for training.

The remainder of the paper is organized as follows. Section 2.1 summarizes the methods used to generate the HF dataset with particular emphasis on the description of the flow diverter model (Section 2.1.1), the synthetic aneurysm model (Section 2.1.2) and the scheme followed to perform FE simulations of stent deployment (Section 2.1.3). Section 2.2 describes the classification model, Section 2.3 the ML-based ROM. In Sections 3, 4, the results of testing these models against unseen scenarios are shown and discussed. Finally, Section 5 presents some concluding remarks.

2 Methods

2.1 High-fidelity simulations

2.1.1 Braided stent modelling

The braided stent is modelled as a tubular net of thin wires with circular cross-section (**Figure 2A**). For a stent with radius R_s, length L_s, composed by N_w wires with radius R_w and presenting N_{cells} repetitive units, the nodal positions are defined by the following set of equations:

$$\begin{cases} x_{n,i} = (R_s + R_w) \cdot \cos(orient \cdot i \cdot d\theta + \Theta_n), \\ y_{n,i} = (R_s + R_w) \cdot \sin(orient \cdot i \cdot d\theta + \Theta_n), \\ z_{n,i} = n \cdot \tan(\phi), \end{cases} \quad (1)$$

with $n \in [0, N_w/2]$ and $i \in [0, N_{cells}]$ and where $orient$ is either 1 or -1 respectively for clockwise and counter-clockwise wires, $d\theta = 2\pi/(N_w/2)$, $\Theta_n = n \cdot 2\pi/(N_w/2)$ and $\phi = L_s/N_{cells}$ is the pitch angle.

In this work, the values assigned to these geometrical parameters refer to a generic flow diverter: $N_w = 48$, $R_s = 2.6$ mm, $R_w = 0.014$ mm, $L = 15$ mm, $N_{cells} = 70$. The stent is made of Phynox, a cobalt-chromium alloy, which is modelled as a linear elastic material with Young modulus $E = 225$ GPa, Poisson coefficient $\nu = 0.33$ and density $\rho = 9.13 \cdot 10^3$ kg/m^3.

2.1.2 Artery and aneurysm modelling

The stent is released within a parametric idealised model of an intracranial artery presenting a saccular aneurysm. The vessel centerline is defined using a planar quadratic Bézier curve:

$$\mathbf{B}(t) = (1-t)^2 \mathbf{P}_0 + 2t(1-t)\mathbf{P}_1 + t^2\mathbf{P}_2 \quad t \in [0,1], \quad (2)$$

where $\mathbf{P}_0$, $\mathbf{P}_2$ are fixed and $\mathbf{P}_1$ is the control point which will be included in the parametrisation. These points lie in a 2D plane, so the only DOFs of this spline are the y, z coordinates of $\mathbf{P}_1$. The curve $\mathbf{B}(t)$ starts from $\mathbf{P}_0$ in the direction of $\mathbf{P}_1$, then bends to reach $\mathbf{P}_2$. Bézier curves allow defining smooth, continuous curves that resemble the curvature of intracranial arteries (Zyłkowski et al., 2018; Danu et al., 2019). The Visualization Toolkit (VTK) software is employed to build the model (Schroeder et al., 2006). The artery is created using the vtkTubeFilter() function that generates a tube around a line. Its diameter (D_v) is considered constant along the centerline. Following, a spherical idealised aneurysm with diameter D_a is created using vtkSphereSource(), the sphere centre $\mathbf{C}_a$ is positioned in the middle

of the vessel centerline and the relative y-distance is parametrised. Through a Boolean union, the sphere is added to the artery model (**Figure 3A**). In summary, the vessel geometry is fully parametrised by 5 parameters: $y_{\mathbf{P}_1}, z_{\mathbf{P}_1}, D_v, D_a, y_{\mathbf{C}_a}$.

2.1.3 FE simulation of braided stent deployment

The simulations are performed using EndoBeams.jl, an in-house and open-source FE modelling framework for the numerical simulation of frictional contact interactions between beams and rigid surfaces (Bisighini et al., 2022). This software has been validated against some public benchmark tests and the commercial software Abaqus (Simulia, Dassault Systems, Providence, RI, United States).

The stent structure is discretised using beam elements and the resulting mesh is composed of 3,408 nodes and 3,360 beam elements. To avoid rendering beam-to-beam contacts, a penalty-based constraint is imposed at each interconnection between the interlaced wires so that the position of nodes pairs is forced to be the same (Zaccaria et al., 2020). The contact algorithm implemented in EndoBeams.jl relies on an implicit representation of the contact surface, the signed distance field (SDF) (Aguirre and Avril, 2020). The SDF of the artery model is computed using the Julia package SignedDistanceField.jl (SignedDistanceField.jl, 2022), which only requires as input the triangular mesh in the form of a. *stl* file (**Figure 3B**). The vessel wall is assumed rigid, thus the SDF is constant along the simulation. The normal contact pressure is treated using a viscoelastic model evaluated by means of a regularised penalty method while the tangential force is determined with a regularised Coulomb friction law considering slip-stick behaviour (Wriggers, 2006; Aguirre and Avril, 2020; Shiozaki et al., 2020).

The strategy followed to perform simulations of stent deployment was inspired by the work proposed in (Perrin et al., 2015; Spranger et al., 2015; Hemmler et al., 2018). This approach is compatible with the use of the SDF to manage the vessel-stent contacts and represents an alternative to the use of a virtual catheter, as typically done before in the literature (Bock et al., 2012). The simulations are carried out in three steps: 1) crimping; 2) positioning and 3) deployment. Since our goal is to obtain the stent configuration at the end of the deployment simulation, we focus on quasi-static simulations. Moreover, as commonly done in literature, the stent is assumed to be positioned along the vessel centerline at the beginning of the deployment (Auricchio et al., 2011; Bock et al., 2012; Hemmler et al., 2018; Leng et al., 2018).

First, the braided stent is crimped by blocking its circumferential DOFs and imposing a radial displacement equal to $(R_{stent} - R_{crimped})$ to all its nodes, where $R_{crimped}$ is the stent target radius after the crimping (**Figure 2A**).

The braided stent nodes lying on the same z-plane can be grouped in N_r rings and a central point can be introduced for each of these rings, $\mathbf{R}_i$ with $i = 1 \dots N_r$ (**Figure 2B**). These points define the initially straight centerline of the crimped stent (C_0). The stent is displaced so that $\mathbf{R}_0$ coincides with the chosen deployment site along the vessel centerline. The final centerline of the stent (C_T) is computed as the projection of C_0 along the vessel one (**Figure 2C**) so as to maintain constant the total length of the stent centerline. C_0, and so C_T, can be subdivided into segments, i.e., vectors connecting subsequent points along the centerline ($\mathbf{seg}_i = \|\mathbf{R}_i - \mathbf{R}_{i-1}\|$). The first point $\mathbf{R}_0$ is considered aligned with the z-axis. A rotation is
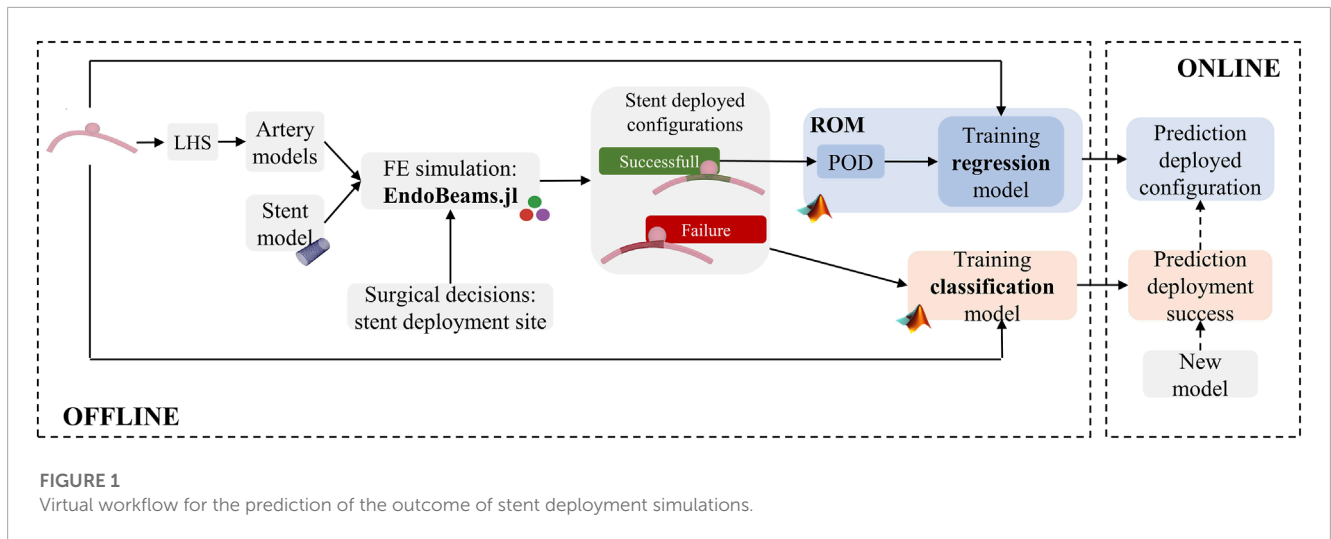
FIGURE 1
Virtual workflow for the prediction of the outcome of stent deployment simulations.

associated with each of these segments to realign it to the preceding one. The angle (θ_i) and axis ($\mathbf{ax}_i$) of rotation for each segment can be computed as follows:

$$\mathbf{ax}_i = \frac{\mathbf{seg}_i \times \mathbf{seg}_{i-1}}{\|\mathbf{seg}_i\|\|\mathbf{seg}_{i-1}\|} \quad (3)$$

$$\theta_i = \cos^{-1}\left(\frac{\mathbf{seg}_i \cdot \mathbf{seg}_{i-1}}{\|\mathbf{seg}_i\|\|\mathbf{seg}_{i-1}\|} \right) \quad (4)$$

Thus, a rotation matrix $\mathcal{M}_i$ can be defined for each segment. By cumulatively applying these rotations, we can align C_T with the z-axis, obtaining C_0:

$$\mathcal{M}_{tot,i} = \prod_{k=1}^{i} \mathcal{M}_i \quad (5)$$

The advantage of this technique is the possibility of obtaining intermediate positions (C_t) between C_T and C_0 by dividing θ_i by the number of desired configurations and applying only this partial rotation to C_T (**Figure 2D**). These intermediate positions are used to drive a physical simulation where kinematics constraints are implemented between all the stent nodes lying on the same ring and their corresponding points $\mathbf{R}_i$ on C_0; the difference of the coordinates between C_t and C_0 is computed and the resulting displacements are applied to each point of C_0 in a successive way leading the stent to bend (**Figure 2E**).

Finally, the stent is allowed to freely deform within the vessel and activate the contact against the wall (**Figure 2F**). The simulations are stopped when the kinetic energy falls below a certain threshold (10^{-12} mJ), which represents static mechanical equilibrium.

2.1.4 Creation of the high-fidelity dataset

The impact of anatomical characteristics and surgical decisions on the final stent configuration is studied. Therefore, we consider a set of geometric features describing both the artery and aneurysm geometry and the stent deployment site along the vessel centerline; we refer to this generic set of parameters as *simulation* parameters, collected in the vector $\boldsymbol{\mu}$.

For the creation of the HF dataset, we considered the following simulation parameters:

$$\boldsymbol{\mu}_B = \left[y_{\mathbf{P}_1}, \quad z_{\mathbf{P}_1}, \quad D_v, \quad D_a, \quad y_{\mathbf{C}_a}, \quad \eta \right]. \quad (6)$$

where $y_{\mathbf{P}_1}$ and $z_{\mathbf{P}_1}$ are the y and z coordinates of the middle Bézier curve point, $y_{\mathbf{C}_a}$ the y coordinate of the aneurysm centre point and η the stent position along the vessel centerline. Since the impact of stent misplacement is one of the objectives of this study, we consider deployment sites in which one of the ends of the stent falls in the aneurysm neck area.

A Latin hypercube sampling (LHS) method is used to generate N_s different values for $\boldsymbol{\mu}_B$; the corresponding artery models are created and a stent deployment simulation is performed within each model as explained in Section 2.1.3. The LHS plan is created using the Julia package LatinHypercubeSampling.jl (LatinHypercubeSampling.jl, 2020). The simulation parameters are evaluated within a range that resembles that observed in the literature (Krejza et al., 2006; Li et al., 2013): for D_v, we consider a range of [2,4] mm; for D_a, a range of [5,10] mm.

Alternatively, the simulation parameters $\boldsymbol{\mu}_B$ are substituted as input of the ML models with $\boldsymbol{\mu}_{cl}$ where, instead of the middle point of the Bézier curve ($\mathbf{P}_1$) and the stent deployment site (η), the y and z coordinates of N_{cl} points ($\mathbf{Q}_i$) on the positioned stent centerline (C_T) are used:

$$\boldsymbol{\mu}_{cl} = \left[y_{\mathbf{Q}_1}, \quad z_{\mathbf{Q}_1}, \quad \cdots, \quad y_{\mathbf{Q}_{N_{cl}}}, \quad z_{\mathbf{Q}_{N_{cl}}}, \quad D_v, \quad D_a, \quad y_{\mathbf{C}_a} \right]. \quad (7)$$

The $\boldsymbol{\mu}_{cl}$ vector components are calculated geometrically from the deployment conditions and no FE simulation is required.

2.2 Binary classification

Within the HF dataset, deployment solutions are considered "successful" from a clinical perspective if the stent extremities are in contact with the artery wall and do not land within the aneurysm sac; otherwise, the simulation outcome is labelled as "failure" (**Figure 4**). This classification is done automatically by checking if, at the end of the simulation, one or more nodes from the stent extremities are inside the aneurysm. Being it modelled as a sphere, the SDF of the aneurysm can be computed analytically and, thus, a node $\mathbf{x}_p$ is located inside the aneurysm if its distance to the surface is
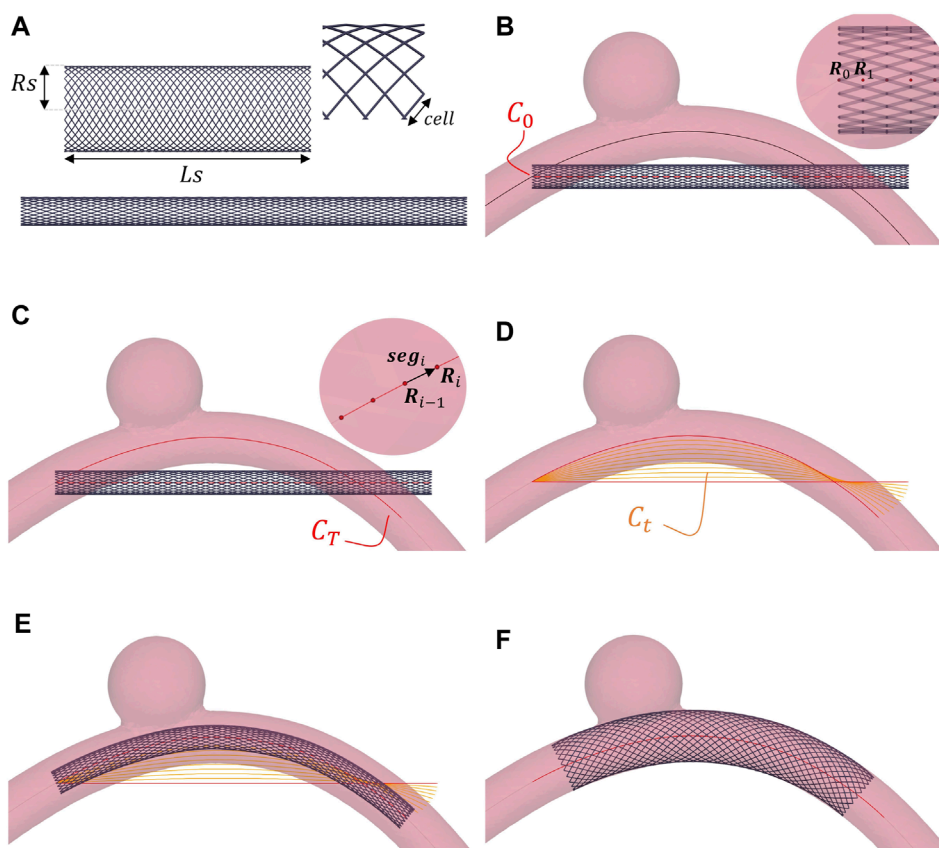
FIGURE 2
Stent deployment strategy. The wire thickness is magnified (2×) to better visualize the stent. **(A)** Free and crimped stent. **(B)** Crimped stent and initially straight centerline (C_0). **(C)** Final centerline of the stent (C_T). **(D)** Intermediate centerlines (C_t). **(E)** Positioned stent. **(F)** Deployed stent.

positive, i.e.,:

$$\|\mathbf{C}_a - \mathbf{x}_p\| - R_a - r_w \geq 0. \tag{8}$$

Once the results are labelled, a supervised machine-learning algorithm is trained to learn the relationship between the simulation parameters $\boldsymbol{\mu}$ and the corresponding solution output: when the model is built, it allows the assignment of new, unseen scenarios to one of the two categories. The classification is done in Matlab (MathWorks, Natick, MA, United States). For this purpose, the performance of the following classifiers is compared (Singh et al., 2016):

- A logistic regression (LR) model is created and fitted using the fitctree() function;
- A k-Nearest Neighbour (k-NN) model is created and fitted using the fitcknn() function;
- A naive Bayes (NB) model is created and fitted using the fitcnb() function;
- A decision tree (DT) model is created and fitted using the fitctree() function;
- An artificial neural network (ANN) model with three layers of size [10, 10, 10] and the hyperbolic function tanh as activation function is created and fitted using the fitnn() function;

- A support vector machine (SVM) model with a polynomial kernel of order 2 is created and fitted using the fitsvm() function.

The architecture and hyperparameters values are chosen for each ML model using the ClassifierLearnerApp.

2.2.1 Metrics

Five metrics are considered to evaluate the performance of the trained classifiers. They are computed by counting true positives (TPs), true negatives (TNs), false positives (FPs) and false negatives (FNs), which are collected in the confusion matrix. Their expressions are.

- Accuracy $= \frac{TP+TN}{TP+FN+TN+FP}$: it tells how good is the classifier, regardless of the label meaning;
- Sensitivity (or recall or TP rate) $= \frac{TP}{TP+FN}$: it tells how good the classifier is at predicting successful deployment conditions;
- Specificity (or TN rate) $= \frac{TN}{TN+FP}$: it tells how good the classifier is at predicting failed deployment conditions;
- Precision $= \frac{TP}{TP+FP}$: it tells how close predicted values are to each other;
- F1-score $= 2\frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}$: it is a more general measure of accuracy that combines precision and recall in a single metric.

FIGURE 3
Artery and aneurysm modelling. **(A)** Parametric idealised model of an intracranial artery presenting a saccular aneurysm: Bézier points P_0, P_1 and P_2, artery diameter D_v, aneurysm diameter D_a and centre C_a. **(B)** SDF section of the idealised artery an aneurysm model: the signed distance from a voxel to the surface is reported as a colour map where negative values (blue) correspond to points outside the surface and positive values (red) inside. The maximum internal distance in correspondence of the aneurysm sac coincides with its radius (5 mm in this case).

FIGURE 4
Examples of successful and unsuccessful deployment outcomes depending on the deployment site along the vessel centerline, η. The wire thickness is magnified (2×) to better visualise the stent. **(A)** Stent positioned at η = 0.3. **(B)** Unsuccessful deployment: landing zone inside the aneurysm space. **(C)** Stent positioned at η = 0.5. **(D)** Successful deployment: good conformity between stent and vessel wall.

Another useful tool to asses the classification performance is the receiver operating characteristic (ROC), a plot showing the performance of the classifier in terms of TP rate (sensitivity) against FP rate (1-specificity) as a function of the cut-off threshold. The metric related to the ROC curve is the area under the curve (AUC). The closer the ROC curve is to the upper left corner of the graph (and thus the higher the AUC value), the more accurate the classifier is.

2.3 Reduced order modelling

In case of successful deployment, a ML-based reduced order modelling method is employed to compute an approximated solution for the stent deployed configuration within the considered vessel. At first, one might consider training a ML algorithm to learn the relationship between the simulation parameters and the vector of nodal displacements at the end of the simulation. However, given the large number of DOFs of the stent model (in our case, ~10,000 DOFs), the size of such an output vector is very large and would result in a very long training time. Reduced order modelling allows for the reduction of the original problem dimension by extracting the most important features, the RBs, from the training dataset through POD (Guo and Hesthaven, 2018; Han et al., 2020; Bridio et al., 2022; Fresca and Manzoni, 2022). A supervised ML algorithm is then used to establish the relationship between simulation parameters and the solution expressed in the RBs space. Finally, an approximated solution of the stent deployed configuration can be recovered in real time for any combination of simulation parameters.

TABLE 1 Number of train and test samples of the binary classifiers ($N_{train, class}$, $N_{test, class}$) and the ML-based ROM ($N_{train, regr}$, $N_{test, regr}$).

N_s	$N_{success}$	$N_{failure}$	$N_{train, class}$	$N_{test, class}$	$N_{train, regr}$	$N_{train, regr}$
150	97	53	50	100	47	50
300	196	104	200	100	146	50
600	396	204	500	100	346	50
900	605	295	800	100	555	50

2.3.1 Proper orthogonal decomposition

The theory behind the POD algorithm is here only introduced: if interested, the reader is suggested to refer to (Chatterjee, 2000). Once the HF dataset is computed, the snapshots matrix **S** is built by arranging the HF solutions $\mathbf{u}_h(\boldsymbol{\mu})$ as columns of a matrix:

$$\mathbf{S} = \begin{bmatrix} \mathbf{u}_h(\boldsymbol{\mu}_1) & | & \mathbf{u}_h(\boldsymbol{\mu}_2) & | & \ldots & | & \mathbf{u}_h(\boldsymbol{\mu}_{N_s}) \end{bmatrix}. \quad (9)$$

Each of these vectors represents one *snapshot* and contains the nodal displacements at the end of the quasi-static deployment simulation:

$$\mathbf{u}_h(\boldsymbol{\mu}) = \begin{bmatrix} u_{x,1}, & u_{y,1}, & u_{z,1}, & \ldots, & u_{x,N_n}, & u_{y,N_n}, & u_{z,N_n} \end{bmatrix}, \quad (10)$$

where N_n is the number of nodes in the stent mesh, thus the dimension of the HF problem is $N_h = 3 \cdot N_n$.

The POD algorithm relies on performing the singular value decomposition (SVD) of **S**:

$$\underset{N_h \times N_s}{\mathbf{S}} = \underset{N_h \times N_h}{\mathbf{U}} \quad \underset{N_s \times N_h}{\boldsymbol{\Sigma}} \quad \underset{N_s \times N_s}{\mathbf{Z}^T}, \quad (11)$$

where $\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 | \mathbf{u}_2 | \cdots | \mathbf{u}_{N_h} \end{bmatrix}$ is the left singular vectors matrix, $\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 | \mathbf{z}_2 | \cdots | \mathbf{z}_{N_s} \end{bmatrix}$ is the right singular vectors matrix and $\boldsymbol{\Sigma} = diag(\sigma_1, \sigma_2, \ldots, \sigma_{N_s})$ contains the singular values of **S**, sorted from the largest to the smallest ($\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{N_s} \geq 0$).

The Schmidt-Eckhart-Young theorem states that the columns of **S**, Col(**S**), can be well approximated by the first L left singular vectors of **S**, i.e., Col(**U**), if the singular values decay rapidly. Thus, given a tolerance ε_{POD}, L can be found as the minimum integer such that:

$$\frac{\sum_{i=1}^{L} \sigma_i}{\sum_{i=1}^{N_s} \sigma_i} \geq 1 - \varepsilon_{POD}. \quad (12)$$

The column vectors $\begin{bmatrix} \mathbf{u}_1 | \mathbf{u}_2 | \cdots | \mathbf{u}_L \end{bmatrix}$ represent the RBs of the model and are assembled in the matrix **V**.

The HF solution $\mathbf{u}_h(\boldsymbol{\mu})$ can be now projected onto the reduced space defined by **V**:

$$\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{U}\mathbf{U}^T\mathbf{u}_h(\boldsymbol{\mu}) \approx \mathbf{V}\mathbf{V}^T\mathbf{u}_h(\boldsymbol{\mu}) = \mathbf{V}\mathbf{u}_L(\boldsymbol{\mu}) = \mathbf{u}_{rb}(\boldsymbol{\mu}), \quad (13)$$

where $\mathbf{U}\mathbf{U}^T = \mathbf{I}$ since **U** is orthogonal, $\mathbf{u}_L$ are the L projection coefficients associated with the column bases of **V** and $\mathbf{u}_{rb}(\boldsymbol{\mu})$ is the solution projected onto the reduced space.

2.3.2 Gaussian process regression

As proposed by Guo in (Guo and Hesthaven, 2018), Gaussian process regression (GPR) is adopted to approximate the HF solutions for any simulation parameters combination. A theoretical introduction to GPR is provided in **Appendix 1**, here we simply introduce the implementation aspects and variables necessary for a complete interpretation of the results.

In this work, a GPR model $\hat{f}$ is constructed using the Matlab function fitrgp() with the Matérn 5/2 kernel from a set of training data where the predictors are the simulation parameters $\boldsymbol{\mu}$ and the outputs are the projection coefficients $\mathbf{u}_L(\boldsymbol{\mu})$ computed from the HF solutions as:

$$\boldsymbol{\mu} \rightarrow \hat{\mathbf{u}}_L(\boldsymbol{\mu}) \approx \hat{f}(\boldsymbol{\mu}) \quad \text{trained from} \quad \left(\boldsymbol{\mu}_i, \mathbf{V}^T\mathbf{u}_h(\boldsymbol{\mu}_i) \right). \quad (14)$$

Once the model is trained, the function predictExactWithCov() is used to predict the projection coefficients for any desired unseen value of the simulation parameters $\boldsymbol{\mu}^*$, i.e., $\hat{\mathbf{u}}_L(\boldsymbol{\mu}^*)$. This allows recovering the full-order solution as follows:

$$\mathbf{u}_p(\boldsymbol{\mu}^*) = \mathbf{V}\hat{\mathbf{u}}_L(\boldsymbol{\mu}^*) \approx \mathbf{u}_h(\boldsymbol{\mu}^*). \quad (15)$$

2.3.3 Metrics

For the test cases, the ML-based ROM results are evaluated against the HF solutions by computing three absolute errors at each node of the stent mesh:

- The order reduction error, $\mathrm{E}_{rb} = \|\mathbf{u}_{rb}(\boldsymbol{\mu}) - \mathbf{u}_h(\boldsymbol{\mu})\|$;
- The prediction error, $\mathrm{E}_p = \|\mathbf{u}_p(\boldsymbol{\mu}) - \mathbf{u}_h(\boldsymbol{\mu})\|$;
- The GPR error, $\mathrm{E}_{gpr} = \mathrm{E}_p - \mathrm{E}_{rb}$.

At the nodal level, these errors correspond to the distance between pairs of nodes. The accuracy of a single solution can be evaluated as the average error (AE) or maximum error (ME) on the mesh nodes: e.g., for the order reduction error, we get AE_{rb} and ME_{rb}. Global variables can be computed by averaging these values on the test dataset: e.g., for the order reduction error, we refer to $\overline{\mathrm{AE}}_{rb}$ as the average of AE_{rb} over the test solutions and to $\overline{\mathrm{ME}}_{rb}$ as the average of ME_{rb} over the test solutions.

3 Results

3.1 Binary classification: Prediction of deployment success

In this section, the results of the classification models are presented. To study the influence of the dataset size on the prediction capability of the ML models, we first created a dataset of $N_s = 900$ simulations and then, by using the subLHCoptim() function from the LatinHypercubeSampling.jl package, we defined three optimal subspaces with $N_s = \{150, 300, 600\}$. Following an approach similar to (Hesthaven and Ubbiali, 2018), we considered a fixed number of samples ($N_{test} = 100$) for testing the four different datasets. In **Table 1**, the number of samples considered for ML training and testing is reported. The input data were standardised before training.

TABLE 2 Analysis of classifiers performance. Evaluation metrics of the six ML models for $N_{train} = \{50, 200, 500, 800\}$. The best result for each category is highlighted.

N_{train}	Classification model	Accuracy	Sensitivity	Specificity	Precision	F1-score
50	Logistic Regression	75.0%	87.5%	73.9%	22.5%	35.8%
	K-nearest neighbour	86.0%	81.4%	87.6%	70.9%	75.8%
	Decision Tree	89.0%	100%*	86.2%	64.5%	78.4%
	Naive Bayes	89.0%	45.4%	70.5%	30.3%	36.3%
	ANN	80.0%	61.7%	96.2%	93.5%*	74.3%
	SVM*	93.0%*	85.2%	96.9%*	93.5%*	89.2%*
200	Logistic Regression	67.0%	45.4%	73.0%	32.2%	37.7%
	K-nearest neighbour	80%	66.6%	86.5%	70.9%	68.7%
	Decision Tree	87%	82.1%*	88.8%	74.1%	77.9%
	Naive Bayes	87%	76.4%	92.4%	83.8%	80.0%
	ANN*	91.0%*	78.9%	98.3%*	96.7%*	86.9%*
	SVM	90.0%	80.0%	95.3%	90.3%	84.8%
500	Logistic Regression	74%	77.7%	73.6%	22.5%	35.0%
	K-nearest neighbour	83.0%	73.3%	87.1%	70.9%	72.1%
	Decision Tree	89.0%	88.4%	88.4%	74.1%	80.7%
	Naive Bayes	91.0%	95.8%*	89.4%	74.1%	83.6%
	ANN*	95.0%*	90.6%	97.0%*	93.5%*	92.0%*
	SVM*	95.0%*	90.6%	97.0%*	93.5%*	92.0%*
800	Logistic Regression	75.0%	87.5%	73.9%	12.1%	19.5%
	K-nearest neighbour	86.0%	81.4%	87.6%	70.9%	75.8%
	Decision Tree	89.0%	83.3%	91.4%	80.6%	81.9%
	Naive Bayes	89.0%	100.0%*	86.2%	64.5%	78.4%
	ANN*	94.0%*	87.8%	97.0%*	93.5%*	90.6%*
	SVM	93%	85.2%	96.9%	93.5%*	89.2%

The values of the metrics for the four different dataset sizes are given in **Table 2**. For N_{train} = 200, the confusion matrices of the different classification models are reported in **Figure 5**. The NB, ANN and SVM models all show an AUC larger than 96% when using N_{train} = 200. The performance of all classification models improves when the training dataset is expanded. In general, the LR model shows the worst performance: regardless of the dataset size, its metrics stay below 90%. Precision is especially low (highest value = 32.3%): as visible in **Figure 5A**, the model fails mostly in predicting false cases (failures). For the smallest dataset, the SVM model shows the best performance with accuracy, specificity and precision larger than 90%. When considering N_{train} = 500, all the validation metrics are larger than 90% for the SVM and ANN models. k-NN, DT and NB models also show good performance in terms of accuracy and specificity (between 83% and 96%); however, when compared with SVM and ANN, they have poorer precision, which is reflected in a lower F1-score (maximum 83.3%). The ANN and SVM models achieve the highest value of specificity (97%), precision (93.9%) and F1-score (92%); instead, the highest sensitivity is reached by the NB model (95.8%). When considering the largest dataset N_{train} = 800,

only a few improvements are observed, e.g., the specificity of the NB model increases from 95.8% to 100%. However, the majority of the metrics stabilise or even decrease in value: most likely, the models are undergoing overfitting. The results of the classification are related to the use of $\boldsymbol{\mu}_B$ as predictors; no improvement is observed using $\boldsymbol{\mu}_{cl}$ for classification.

3.2 Reduced order modelling: Prediction of the stent deployed configuration

In this section, the results of the ML-based ROM are presented. As proposed for the classification models, we started with a dataset with N_s = 900 simulations and then defined three optimal subspaces with $N_s = \{150, 300, 600\}$. Then, we built four datasets with only the successful deployment simulations, $N_{success} = \{97, 196, 396, 555\}$, and considered a fixed number of samples (N_{test} = 50) for testing. In **Table 1**, the number of samples considered for ML-based ROM training and testing is reported. The input and output data were standardised before training.
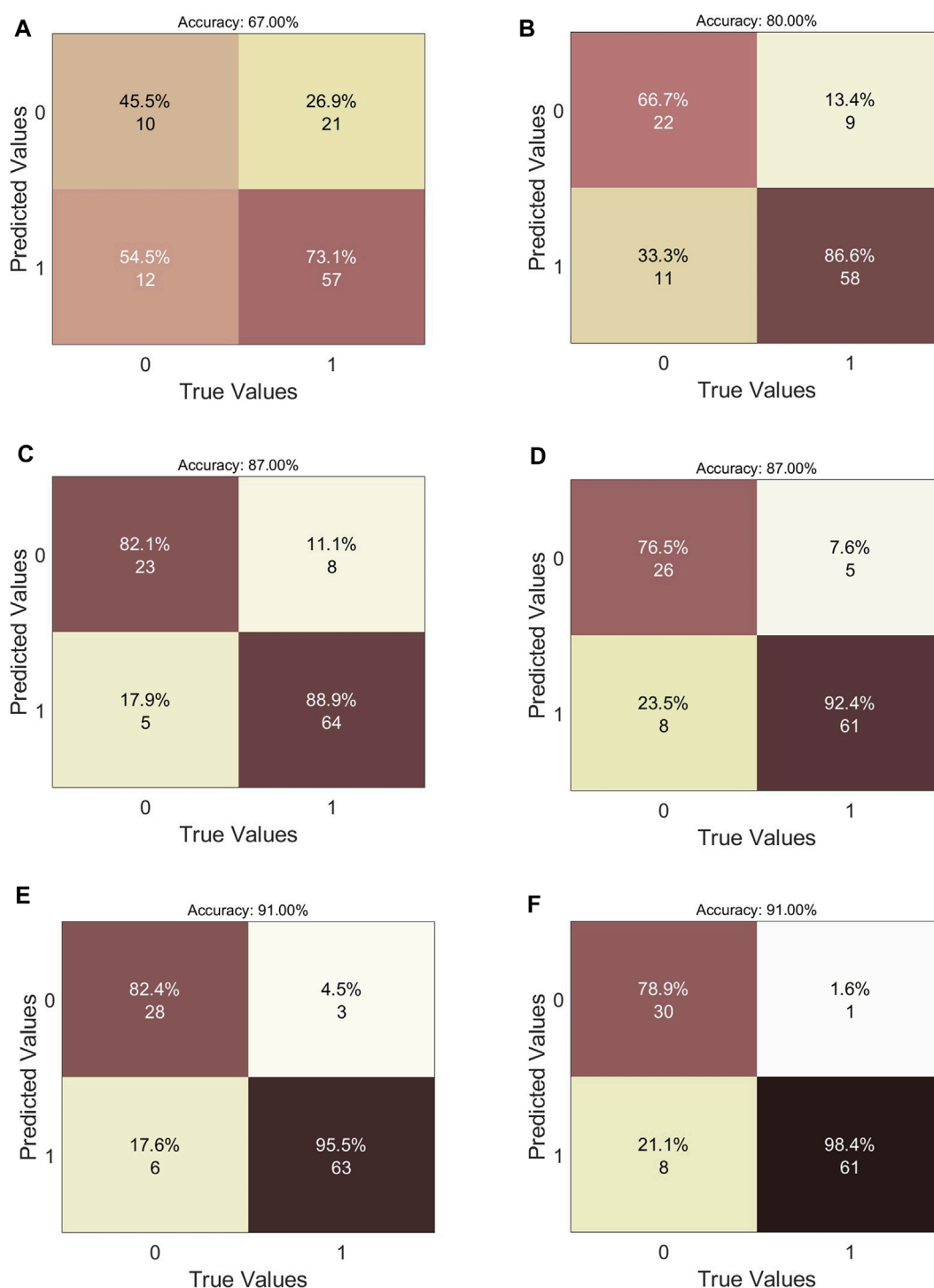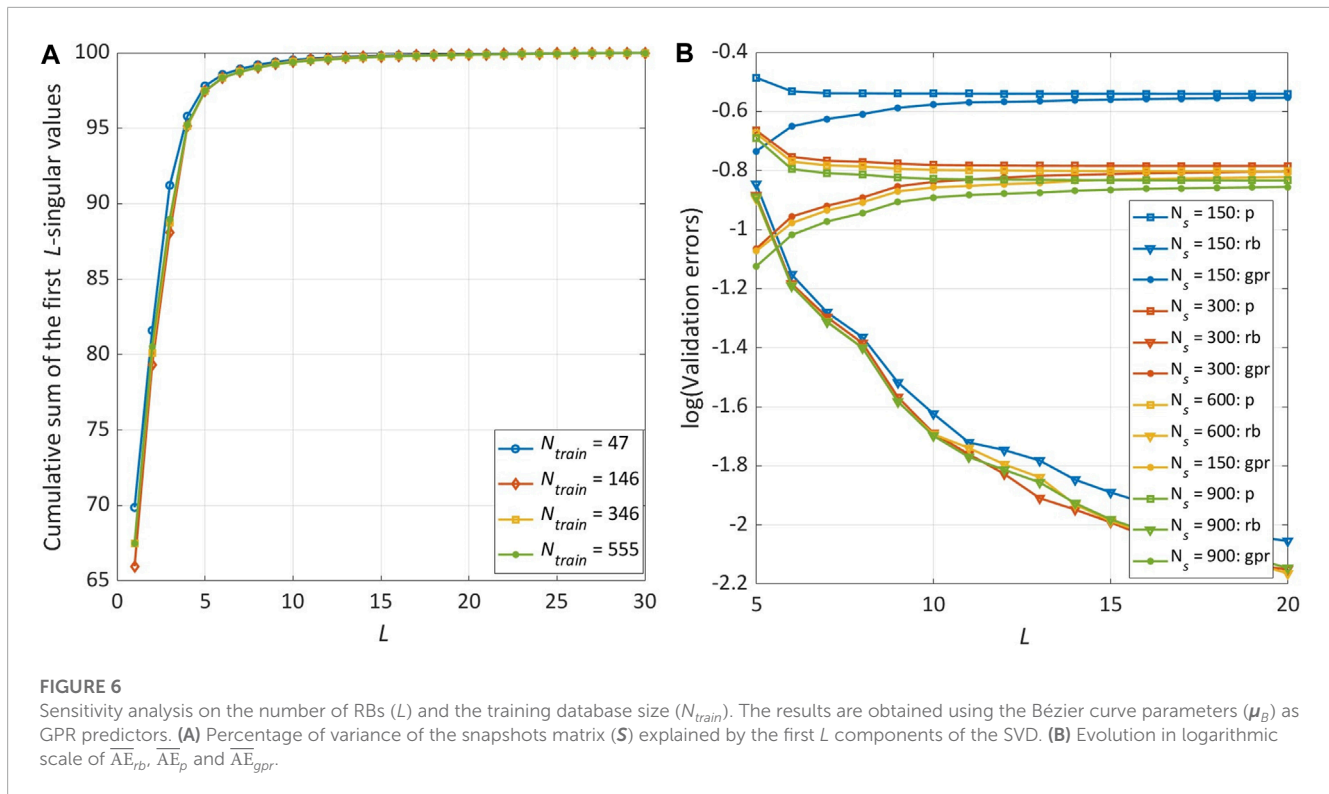
FIGURE 5
Analysis of classifiers performance. Confusion matrices of the six ML models for N_{train} = 200. **(A)** Logistic Regression. **(B)** k-Nearest Neighbour. **(C)** Naive Bayes. **(D)** Decision Tree. **(E)** Artificial Neural Network. **(F)** Support Vector Machine.

The spatial resolution of imaging techniques currently used for IAs detection and treatment was considered for evaluating the prediction errors (Hacein-Bey and Provenzale, 2011; Maupu et al., 2022): magnetic resonance angiography (MRA) = 0.6 mm–1 mm, computed tomography angiography (CTA) = 0.4 mm–0.7 mm,

digital subtraction angiography (DSA) = 0.2 mm, 3D rotational angiography (3DRA) = 0.15 mm.

As already mentioned in Section 2.3, the choice of the RBs number L is carried out considering the singular values of the snapshots matrix. In **Figure 6A**, the cumulative sum of the singular

FIGURE 6
Sensitivity analysis on the number of RBs (L) and the training database size (N_{train}). The results are obtained using the Bézier curve parameters ($\boldsymbol{\mu}_B$) as GPR predictors. **(A)** Percentage of variance of the snapshots matrix ($\boldsymbol{S}$) explained by the first L components of the SVD. **(B)** Evolution in logarithmic scale of $\overline{AE}_{rb}$, $\overline{AE}_p$ and $\overline{AE}_{gpr}$.

values normalised with respect to their total sum is reported: this plot shows the fraction of total variance retained by the first L-singular values for $N_{train} = \{47, 146, 346, 555\}$. As reported in **Table 3**, regardless of the dataset size, the first 4 RBs cover more than 90% ($\varepsilon_{POD} = 0.1$) and the first 10 RBs more than 99% ($\varepsilon_{POD} = 0.01$) of the total variance in the input data. As shown in **Figure 6B**, the prediction does not benefit considerably by considering a number of RBs greater than 15: in fact, analogously for any value of N_{train}, the average order reduction error decreases towards 0 as more RBs are considered, while the average prediction and GPR errors reach a stable plateau around $L = 15$. Therefore, henceforth $L = 15$ is considered in this analysis.

The results presented so far relate to the use of $\boldsymbol{\mu}_B$ as predictors. As reported in **Table 4** and shown in **Figure 7**, a strong reduction of the prediction error is evident when using $\boldsymbol{\mu}_{cl}$ instead: with $N_{cl} = 3$, the average prediction error is 5.65× lower than when using $\boldsymbol{\mu}_B$ while the maximum prediction error is 2.84× lower. Increasing the number of considered points N_{cl}, a further but slight decrease of the prediction error is observed: the average and maximum values are respectively 1.11× and 1.03× lower with $N_{cl} = 5$ and 1.08× and 1.04× lower with $N_{cl} = 8$. Using $\boldsymbol{\mu}_{cl}$, the percentage of test samples where the average prediction error is greater than the spatial resolution of 3DRA is zero; the same is true for the number of test samples where the maximum prediction error is greater than the spatial resolution of CTA. The number of test samples where the maximum prediction error is greater than the spatial resolution of 3DRA is reduced from 64% with $\boldsymbol{\mu}_B$ to 10% with $\boldsymbol{\mu}_{cl}$. The improvement gained by using $\boldsymbol{\mu}_{cl}$ is not only reflected in a lower variance of the prediction error within the test dataset but also within a single test solution: this is noticeable when sampling the multivariate normal distributions corresponding to the GPR outputs, as explained in **Appendix 1**. In **Figure 8**, we

reported the position of 100 points sampled within the distribution predicted by the GPR for the displacement of the first node of the stent mesh. The points are more widely distributed with $\boldsymbol{\mu}_B$ while they are highly concentrated around the mean node value with $\boldsymbol{\mu}_{cl}$, which means that the uncertainty on the predicted displacement value when using $\boldsymbol{\mu}_{cl}$ is lower than when using $\boldsymbol{\mu}_B$.

Finally, we analysed the influence of the number of train samples on the ML-based ROM performance. Considering $N_{train} = 47$ instead of $N_{train} = 146$, the average prediction error is 1.92× lower while the maximum prediction error is 1.84× lower. The average and maximum prediction error slightly decrease when considering $N_{train} = 346$ instead of $N_{train} = 146$, respectively 1.14× and 1.04×. It can be observed in **Figure 9** that further expanding the training dataset, the prediction error converges around the value reached when $N_{train} = 346$ is used.

4 Discussion

In this work, a fast and accurate method for braided stent deployment analysis has been proposed. It consists of a two-step workflow where FE simulations are used, first, to train a ML model that classifies successful and unsuccessful simulations and, next, to train a ML-based ROM that approximates the stent deployed configuration within the considered vessel. This approach was validated by studying the effect of a combination of geometrical and surgical parameters on the outcome of the stent deployment simulation: to this end, we employed a parametric idealised model of an intracranial artery characterised by a saccular aneurysm. The presented model relies on previous work to develop an optimised framework for numerical simulations of frictional

TABLE 3 Analysis of ML-based ROM performance considering different tolerances (ε_{POD}), numbers of RBs (L) and training database sizes (N_{train}). The results are obtained using the Bézier curve parameters (μ_B) as GPR predictors.

N_{train}	ε_{POD}	L	$\overline{AE}_{rb} \pm SD$ [mm]	$\overline{ME}_{rb} \pm SD$ [mm]	$\overline{AE}_p \pm SD$ [mm]	$\overline{ME}_p \pm SD$ [mm]
47	0.1	3	0.730 ± 0.588	1.797 ± 1.797	0.813 ± 0.628	1.842 ± 1.343
	0.01	8	0.043 ± 0.025	0.171 ± 0.097	0.289 ± 0.238	0.460 ± 0.324
	0.001	25	0.008 ± 0.004	0.036 ± 0.025	0.288 ± 0.239	0.437 ± 0.325
146	0.1	4	0.271 ± 0.168	0.728 ± 0.483	0.331 ± 0.189	0.757 ± 0.429
	0.01	9	0.027 ± 0.016	0.111 ± 0.065	0.167 ± 0.119	0.249 ± 0.146
	0.001	58	0.003 ± 0.002	0.014 ± 0.008	0.164 ± 0.120	0.232 ± 0.147
346	0.1	4	0.271 ± 0.166	0.723 ± 0.469	0.329 ± 0.182	0.746 ± 0.413
	0.01	10	0.020 ± 0.010	0.078 ± 0.042	0.159 ± 0.119	0.217 ± 0.130
	0.001	99	0.002 ± 0.001	0.008 ± 0.004	0.157 ± 0.120	0.198 ± 0.131
555	0.1	4	0.270 ± 0.161	0.724 ± 0.463	0.323 ± 0.172	0.744 ± 0.403
	0.01	10	0.020 ± 0.010	0.077 ± 0.041	0.148 ± 0.111	0.204 ± 0.121
	0.001	124	0.001 ± 0.001	0.006 ± 0.003	0.146 ± 0.112	0.181 ± 0.121

TABLE 4 Analysis of ML-based ROM performance comparing the Bézier curve parameters (μ_B) and the centerline points parameters (μ_{cl}) as GPR predictors. The results are obtained with $L = 15$ and $N_{train} = 146$.
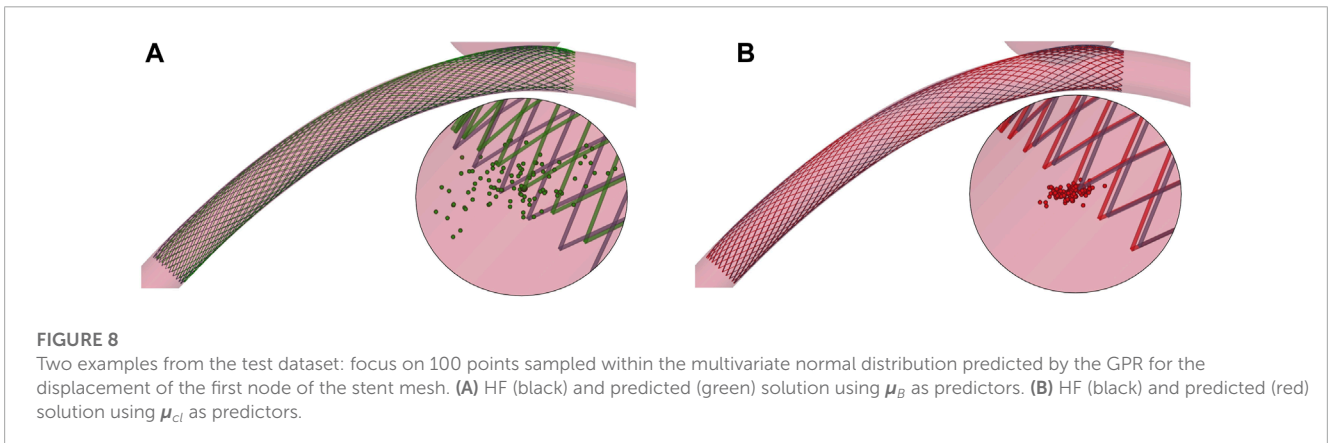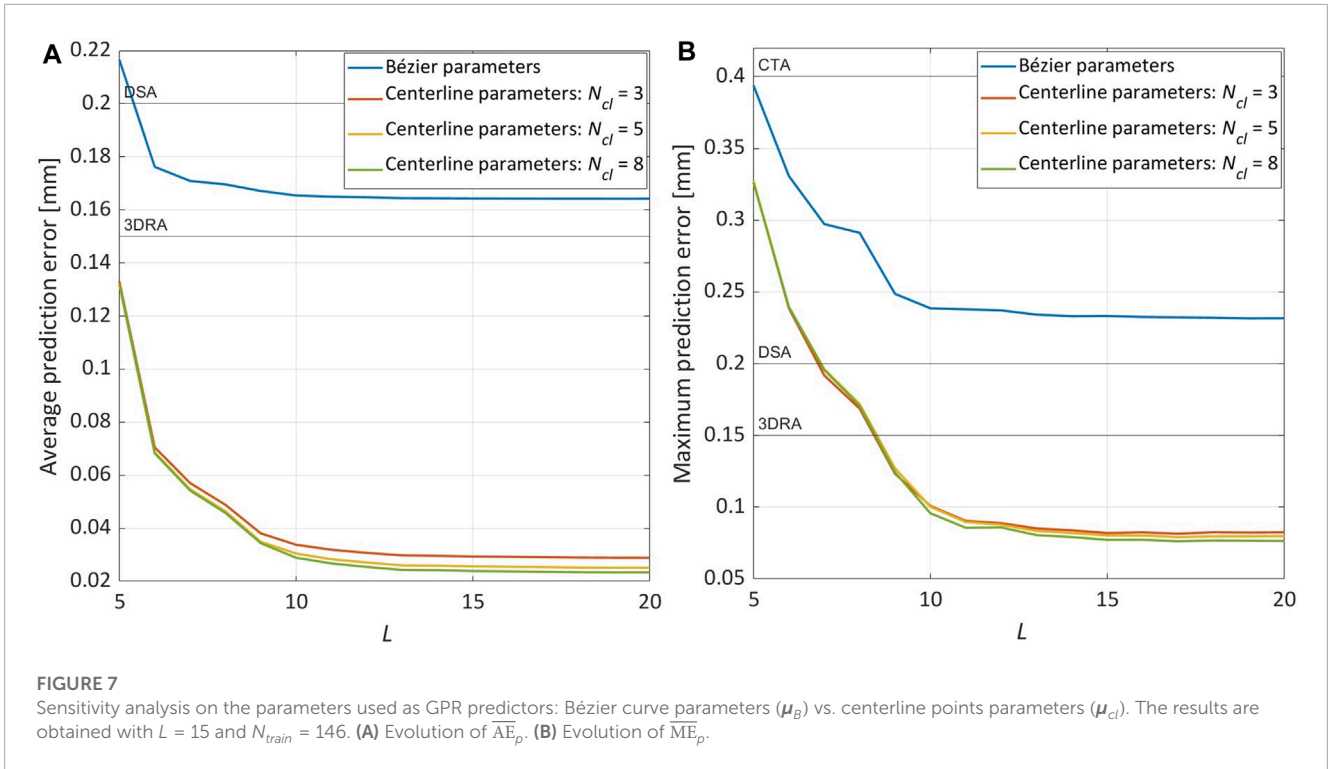
Predictors	$\overline{AE}_p \pm SD$ [mm]	$N_{AE_p>0.15mm=3DRA}$	$\overline{ME}_p \pm SD$ [mm]	$N_{ME_p>0.15mm=3DRA}$	$N_{ME_p>0.4mm=CA}$
Bézier	0.164 ± 0.120	26 (52%)	0.233 ± 0.147	32 (64%)	4 (8%)
$N_{cl} = 3$	0.029 ± 0.020	0 (0%)	0.082 ± 0.058	5 (10%)	0 (0%)
$N_{cl} = 5$	0.026 ± 0.017	0 (0%)	0.080 ± 0.059	5 (10%)	0 (0%)
$N_{cl} = 8$	0.024 ± 0.017	0 (0%)	0.077 ± 0.058	6 (12%)	0 (0%)

contact interactions between wire-like structures discretised using beam elements and rigid surfaces [Aguirre and Avril. (2020); Bisighini et al. (2022)]. Based on this, here we proposed an efficient approach to simulate braided stent deployment, which allowed us to reduce the computational time required to collect the simulations needed for ML training. As explained in Section 2.1, stent positioning is performed by connecting a subset of the stent nodes to a virtual centreline built along its main axis and imposing a pre-computed displacement field on this centreline. These kinematic constraints lead the stent to follow the centreline and bend. Thanks to this technique, the computational time to perform a FE simulation of stent deployment takes, on average, 15 min. This makes it possible to build even large datasets in an acceptable amount of time: e.g., the largest dataset $N_s = 900$ used in this analysis could be created in 12 h on 20 nodes of a cluster with 2.6 Ghz Intel Xeon Gold 6,132 CPUs and 6 Gb of RAM for each node.

We trained six ML models on a dataset of varying size and obtained classifiers that were 80%–91% accurate in predicting the deployment outcome even with a relatively small dataset ($N_{train} = 50$). Only increasing its size to $N_{train} = 500$ and using SVM and NN models, we were able to perform binary classification with all the validation metrics between 89% and 97%. The surgical needs addressed by this model require us to favour the presence of FNs

over that of FPs, i.e., we prefer to mislabel a successful simulation as a failure rather than the opposite. To minimize the presence of FPs then, high-specificity models are preferred. Therefore, for $N_{train} = 50$, we would select the SVM model and for $N_{train} = 200$, the ANN model. As shown in **Figure 10**, FNs misclassification may be explained by the fact that, in these cases, the stent is in a "boundary" situation in which one of its extremities lands immediately after the aneurysm neck. Introducing deformable walls, these cases would most probably fall into the "failure" condition. A similar situation is observed also for some FPs cases, i.e., one of the extremities of the stent lands within the aneurysm immediately before the aneurysm neck, but no such clear explanation was found for all the FPs.
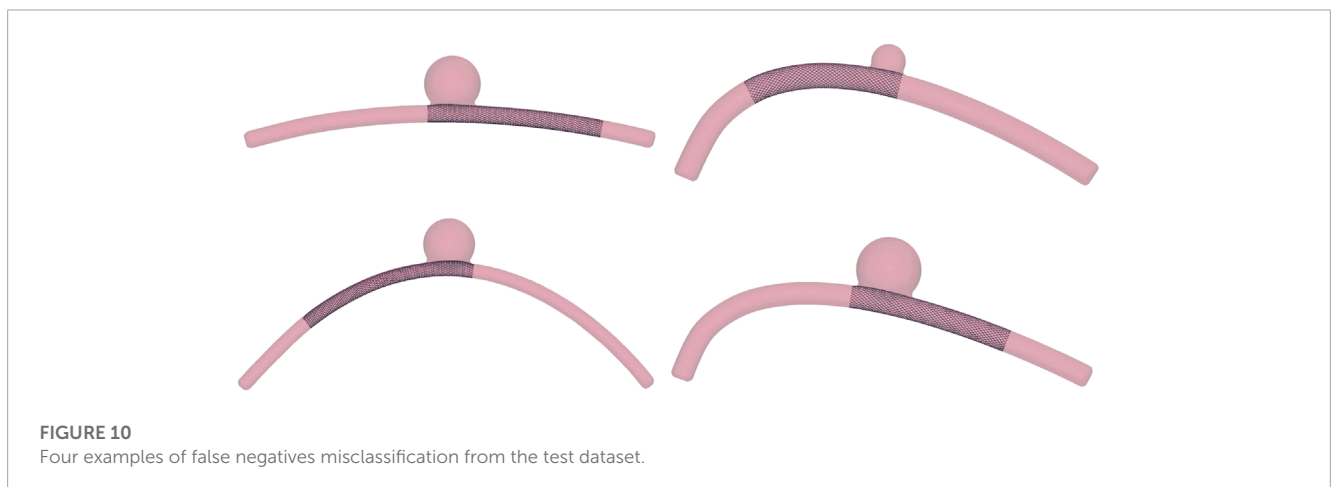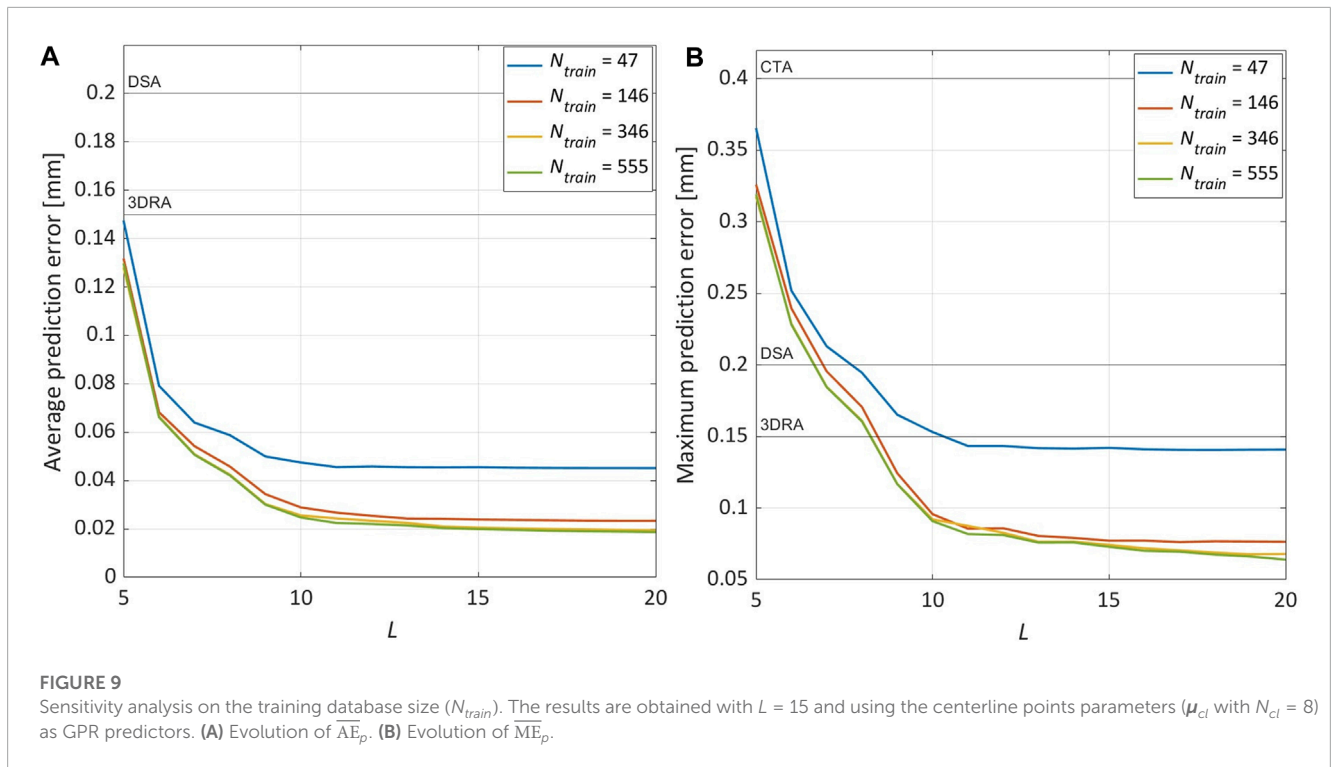
For the regression, the POD algorithm was employed to extract the RBs while a GPR model to establish a mapping between the simulation parameter values and the projection coefficients. The results showed that GPR is strongly influenced by the input parameters: maintaining the same output data and changing the predictors from μ_B to μ_{cl}, the average prediction error could be decreased by more than 5×. This reduction can be explained by the less non-linear relationship present between these alternative predictors and the output variables. We carried out a sensitivity analysis on the number of RBs to be considered for the dimensionality reduction of the problem and on the training

FIGURE 7
Sensitivity analysis on the parameters used as GPR predictors: Bézier curve parameters ($\boldsymbol{\mu}_B$) vs. centerline points parameters ($\boldsymbol{\mu}_{cl}$). The results are obtained with $L = 15$ and $N_{train} = 146$. **(A)** Evolution of $\overline{AE}_p$. **(B)** Evolution of $\overline{ME}_p$.

FIGURE 8
Two examples from the test dataset: focus on 100 points sampled within the multivariate normal distribution predicted by the GPR for the displacement of the first node of the stent mesh. **(A)** HF (black) and predicted (green) solution using $\boldsymbol{\mu}_B$ as predictors. **(B)** HF (black) and predicted (red) solution using $\boldsymbol{\mu}_{cl}$ as predictors.

dataset size. The results showed that the prediction error stabilises with a relatively low number of RBs (15). On the other hand, with 47 training samples, we were able to achieve a maximum prediction error slightly lower than the spatial resolution of 3DRA; with 147 training samples, we were able to reduce it to 0.07 mm, half the spatial resolution of 3DRA. In our analysis, the average prediction error is never greater than 0.15 mm (3DRA spatial resolution) while the maximum prediction error is never greater than 0.4 mm (CTA spatial resolution) and is lower than 0.15 mm (3DRA spatial resolution) in 90% of the total test cases. By looking at the test examples reported in **Figure 11**, it can be observed that the prediction error is evenly distributed and the stent configuration accommodates very well the vessel curvature. In light of these results, we can conclude that a prediction error in the range of 0.01 mm–0.05 mm leads to an approximated stent that is very close to the HF results and such an error is achievable even with a relatively small dataset (47 simulations). In the cases characterised
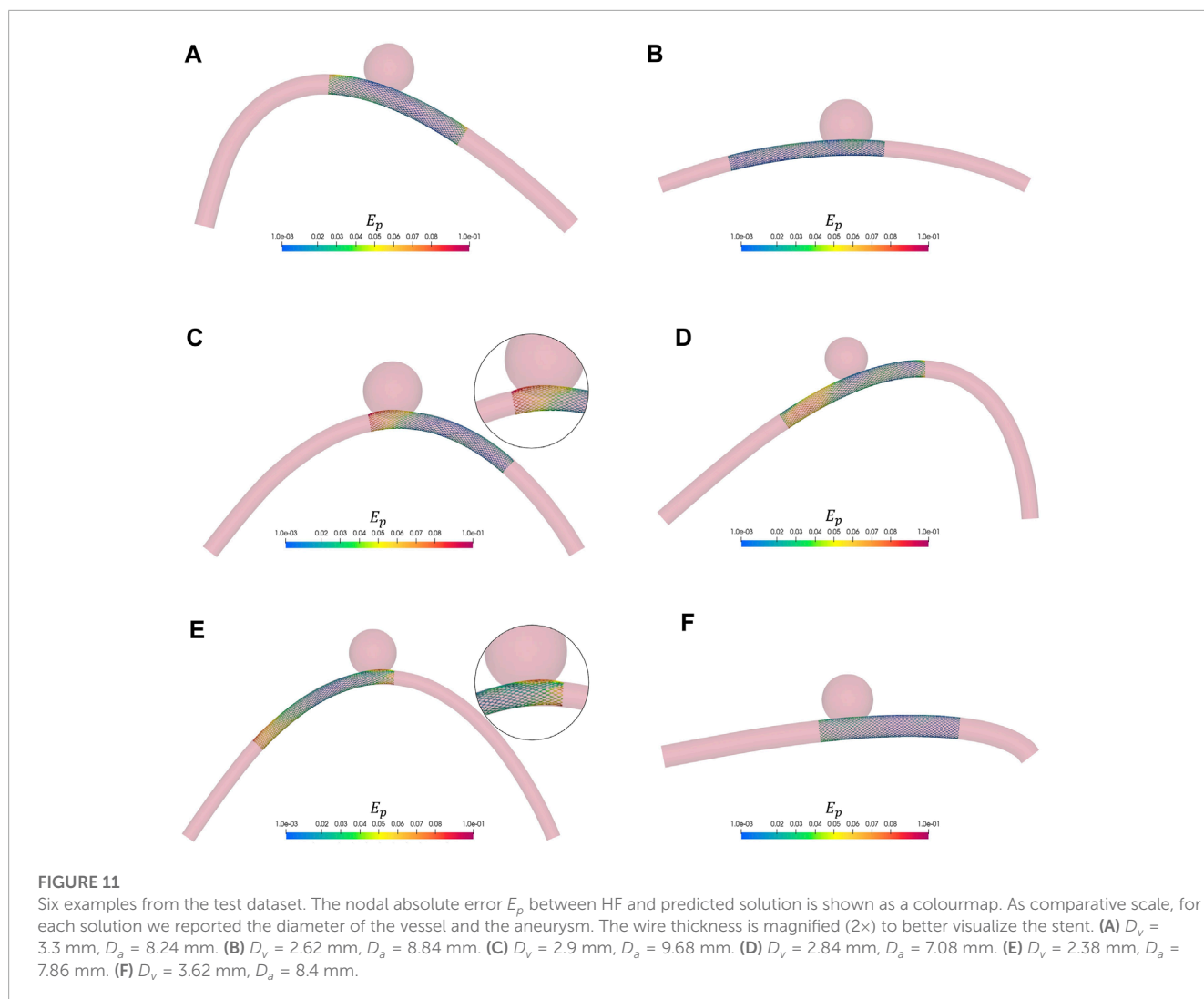
by larger prediction errors (Figures 11C, E), the greatest differences can be observed at the stent extremities. These cases correspond to the same "boundary" situation that also affects the classification models, where one of the stent extremities lands immediately after the aneurysm neck. As mentioned above, by introducing deformable walls, the stent would most likely slip within the aneurysmal space in this situation, making the deployment unsuccessful. Thanks to the complete decoupling between the offline stage (extraction of RBs and training of the regression model) and the online one (prediction of new solutions), the computational cost is decreased from ~15 min, using the FE simulation scheme proposed, to a few milliseconds.

The results here presented are promising as they demonstrate the ability of ML and reduced order modelling techniques to account for the non-linearities of the stent deployment problem and accurately model its outcome. Compared to other fast virtual stenting methods proposed in the literature, reduced order modelling has the

FIGURE 9
Sensitivity analysis on the training database size (N_{train}). The results are obtained with $L = 15$ and using the centerline points parameters ($\boldsymbol{\mu}_{cl}$ with $N_{cl} = 8$) as GPR predictors. **(A)** Evolution of $\overline{AE}_\rho$. **(B)** Evolution of $\overline{ME}_\rho$.

FIGURE 10
Four examples of false negatives misclassification from the test dataset.

potential of reducing the computational cost of stent deployment simulation without simplifying the underlying biomechanical model [Larrabide et al. (2012); Spranger et al. (2015); Zhong et al. (2016)]. However, the current approach presents some limitations that need to be addressed. The first limitation is the ability to consider only one stent geometry at a time: due to the analytical formulation we used to build the braided stent mesh (Eq. 1), variations in the stent radius result in a different number of nodes and, therefore, number of DOFs. This in turn would render displacement vectors of different lengths that could not be assembled in the same snapshot matrix **S**. Considering stents of various sizes would require the construction of separate ROMs. Nevertheless, flow diverters are available in a finite and limited number of sizes, so creating different models could be feasible. An alternative solution could be the introduction of a sampling step to transform the stent

mesh into a point cloud with a fixed number of points. Besides, the independent single-output GPR method used in this project does not model the correlation between outputs (**Appendix 1**): a multi-output GPR approach should be considered in the future (Liu et al., 2018). Secondly, both the stent and vessel FE models here considered are based on some simplifications: In particular, the penalty-based constraints at the wires interconnections for the stent and the rigid-wall hypothesis for the vessels. Given the high braiding angle and number of wires of flow diverters (Kim et al., 2008), the simplified contact technique employed for modelling wire-to-wire interactions reduces its radial and axial flexibility limiting the application within tortuous vessels (Zaccaria et al., 2020) and prevents modelling clinical practice as the "push-pull" technique (Ma et al., 2014). Therefore, with the aim of applying the same workflow to patient-specific geometries, the current approach

FIGURE 11
Six examples from the test dataset. The nodal absolute error E_p between HF and predicted solution is shown as a colourmap. As comparative scale, for each solution we reported the diameter of the vessel and the aneurysm. The wire thickness is magnified (2×) to better visualize the stent. **(A)** D_v = 3.3 mm, D_a = 8.24 mm. **(B)** D_v = 2.62 mm, D_a = 8.84 mm. **(C)** D_v = 2.9 mm, D_a = 9.68 mm. **(D)** D_v = 2.84 mm, D_a = 7.08 mm. **(E)** D_v = 2.38 mm, D_a = 7.86 mm. **(F)** D_v = 3.62 mm, D_a = 8.4 mm.

should be replaced by a more general contact formulation (Ma et al., 2012; Shiozaki et al., 2020). On the other hand, it is still unclear if the introduction of wall deformability would have a strong impact on the stenting simulation. Since intracranial arteries are more rigid than extracranial ones [Hayashi et al. (1980)], most studies assume rigid walls finding good agreement between the results of numerical simulations and experimental tests [Ma et al. (2012); Ma et al. (2013); Ma et al. (2014)]. However, some preliminary comparisons on idealised geometries highlighted the presence of differences in the stent deployed configuration, in particular in the radial direction (Fu and Xia, 2017; Cai et al., 2020). Moreover, deformable walls would enable modelling vessel straightening following the insertion of endovascular guides and stent sheaths (King et al., 2012; Gindre et al., 2017). Concerning the ROM construction, both of these improvements to the biomechanical model would increase the computational cost of the HF simulations, introduce a higher level of complexity in the modelled problem and, thus, require a larger training dataset. Moreover, deformable walls would most likely change the outcome of the deployment simulation, in particular in the "boundary" situations discussed above. Finally, the vessel geometry considered for this work is rather far from patient-specific

geometries. Introducing more parameters will result in the need for a larger number of bases (hence, a larger training dataset) to predict the solution well. Therefore, efficient algorithms to construct the training dataset should be explored to reduce its size as much as possible as well as alternative ML algorithms for regression, e.g., neural networks.

5 Conclusion

This work represents the first attempt to combine finite element simulations with machine learning and reduced order modelling for the analysis of braided stent deployment. Its feasibility was demonstrated using an idealised vessel model, where a set of geometrical features can be controlled. Surgical decisions were also taken into account in the creation of the high-fidelity dataset. The two-step workflow allows the classification of deployment conditions with up to 95% accuracy and real-time prediction of the stent deployed configuration with a maximum prediction error always lower than the spatial resolution of computed tomography angiography (0.4 mm) and lower than that of 3D rotational

angiography (0.15 mm) in 90% of test cases. Despite the simplified vessel shape and the assumption of rigid walls, this study is representative of the clinical scenario and can be extended to more realistic applications without modification. Current efforts are focused on understanding how many parameters are needed to fully describe patient-specific models. To represent such geometries in the reduced-order model, a statistical shape model will be used instead of the parametrization presented here. Future developments of the presented framework will also include the improvement of the clinical criterion used to assess the effectiveness of treatment: for example, the porosity of the stent at the neck of the aneurysm should be taken into account as it affects the outcome of flow diversion. In the future, a similar computational tool could be used by practitioners before and during intracranial aneurysm surgery to rule out conditions that would lead to unsuccessful deployment, visualize the stent configuration depending on the deployment site and check whether the chosen device covers one or more side branches.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Author contributions

BB: Conceptualization, data preparation, data analysis, methodology, software, validation, formal analysis, investigation, writing—original draft; MA and BP: Conceptualization, methodology, formal analysis, investigation, writing—review and editing, supervision; SA, DP, MB, and FT: Conceptualization, formal analysis, writing—review and editing, supervision.

Funding

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The reviewer EP is currently organizing a Research Topic with the author MA.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Aguirre, M., and Avril, S. (2020). An implicit 3D corotational formulation for frictional contact dynamics of beams against rigid surfaces using discrete signed distance fields. *Comput. Methods Appl. Mech. Eng.* 371, 113275. doi:10.1016/j.cma.2020.113275

Auricchio, F., Conti, M., De Beule, M., De Santis, G., and Verhegghe, B. (2011). Carotid artery stenting simulation: From patient-specific images to finite element analysis. *Med. Eng. Phys.* 33, 281–289. doi:10.1016/j.medengphy.2010.10.011

Ballarin, F., Faggiano, E., Ippolito, S., Manzoni, A., Quarteroni, A., Rozza, G., et al. (2016). Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD-Galerkin method and a vascular shape parametrization. *J. Comput. Phys.* 315, 609–628. doi:10.1016/j.jcp.2016.03.065

Ballarin, F., Manzoni, A., Quarteroni, A., and Rozza, G. (2015). Supremizer stabilization of POD-Galerkin approximation of parametrized steady incompressible Navier-Stokes equations. *Int. J. Numer. Methods Eng.* 102, 1136–1161. doi:10.1002/nme.4772

Biancolini, M. E., Capellini, K., Costa, E., Groth, C., and Celi, S. (2020). Fast interactive CFD evaluation of hemodynamics assisted by RBF mesh morphing and reduced order models: The case of aTAA modelling. *Int. J. Interact. Des. Manuf.* 14, 1227–1238. doi:10.1007/s12008-020-00694-5

Biancolini, M. E., and Valentini, P. P. (2018). Virtual human bone modelling by interactive sculpting, mesh morphing and force-feedback. *Int. J. Interact. Des. Manuf. (IJIDeM)* 12, 1223–1234. doi:10.1007/s12008-018-0487-3

Bisighini, B., Aguirre, M., Pierrat, B., Perrin, D., and Avril, S. (2022). EndoBeams.jl: A Julia finite element package for beam-to-surface contact problems in cardiovascular mechanics. *Adv. Eng. Softw.* 171, 103173. doi:10.1016/j.advengsoft.2022.103173

Bock, S. D., Iannaccone, F., Santis, G. D., Beule, M. D., Mortier, P., Verhegghe, B., et al. (2012). Our capricious vessels: The influence of stent design and vessel geometry on the mechanics of intracranial aneurysm stent deployment. *J. Biomechanics* 45, 1353–1359. doi:10.1016/j.jbiomech.2012.03.012

Bridio, S., Luraghi, G., Migliavacca, F., Pant, S., García-González, A., and Rodriguez Matas, J. F. (2022). A low dimensional surrogate model for a fast estimation of strain in the thrombus during a thrombectomy procedure. *J. Mech. Behav. Biomed. Mater.* 137, 105577. doi:10.1016/j.jmbbm.2022.105577

Buoso, S., Joyce, T., and Kozerke, S. (2021). Personalising left-ventricular biophysical models of the heart using parametric physics-informed neural networks. *Med. Image Anal.* 71, 102066. doi:10.1016/j.media.2021.102066

Cai, Y., Meng, Z., Jiang, Y., Zhang, X., Yang, X., and Wang, S. (2020). Finite element modeling and simulation of the implantation of braided stent to treat cerebral aneurysm. *Med. Nov. Technol. Devices* 5, 100031. doi:10.1016/j.medntd.2020.100031

Carè, A., and Camporeale, E. (2018). Regression. *Mach. Learn. Tech. Space Weather* 71, 71–112. doi:10.1016/B978-0-12-811788-0.00004-4

Chang, G. H., Schirmer, C. M., and Modarres-Sadeghi, Y. (2017). A reduced-order model for wall shear stress in abdominal aortic aneurysms by proper orthogonal decomposition. *J. Biomechanics* 54, 33–43. doi:10.1016/j.jbiomech.2017.01.035

Chatterjee, A. (2000). An introduction to the proper orthogonal decomposition. *Curr. Sci.* 78, 808–817.

Chen, W., Wang, Q., Hesthaven, J. S., and Zhang, C. (2021). Physics-informed machine learning for reduced-order modeling of nonlinear problems. *J. Comput. Phys.* 446, 110666. doi:10.1016/j.jcp.2021.110666

Cosentino, F., Raffa, G. M., Gentile, G., Agnese, V., Bellavia, D., Pilato, M., et al. (2020). Statistical shape analysis of ascending thoracic aortic aneurysm: Correlation between shape and biomechanical descriptors. *J. Personalized Med.* 10, 28–14. doi:10.3390/jpm10020028

Danu, M., Nita, C. I., Vizitiu, A., Suciu, C., and Itu, L. M. (2019). "Deep learning based generation of synthetic blood vessel surfaces," in 2019 23rd International Conference on System Theory, Control and Computing, ICSTCC 2019, Sinaia, Romania, 09-11 October 2019 (IEEE), 662–667. doi:10.1109/ICSTCC.2019.8885576

Durso, P. I., Lanzino, G., Cloft, H. J., and Kallmes, D. F. (2011). Flow diversion for intracranial aneurysms: A review. *Stroke* 42, 2363–2368. doi:10.1161/STROKEAHA.111.620328

Fresca, S., and Manzoni, A. (2022). POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Comput. Methods Appl. Mech. Eng.* 388, 114181. doi:10.1016/j.cma.2021.114181

Fu, W., and Xia, Q. (2017). Interaction between flow diverter and parent artery of intracranial aneurysm: A computational study. *Appl. Bionics Biomechanics* 2017, 3751202. doi:10.1155/2017/3751202

Gindre, J., Bel-Brunon, A., Rochette, M., Lucas, A., Kaladji, A., Haigron, P., et al. (2017). Patient-specific finite-element simulation of the insertion of guidewire during an evar procedure: Guidewire position prediction validation on 28 cases. *IEEE Trans. Biomed. Eng.* 64, 1057–1066. doi:10.1109/TBME.2016.2587362

Girfoglio, M., Ballarin, F., Infantino, G., Nicoló, F., Montalto, A., Rozza, G., et al. (2022). Non-intrusive PODI-ROM for patient-specific aortic blood flow in presence of a LVAD device. *Med. Eng. Phys.* 107, 103849–103923. doi:10.1016/j.medengphy.2022.103849

Guo, M., and Hesthaven, J. S. (2019). Data-driven reduced order modeling for time-dependent problems. *Comput. Methods Appl. Mech. Eng.* 345, 75–99. doi:10.1016/j.cma.2018.10.029

Guo, M., and Hesthaven, J. S. (2018). Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Comput. Methods Appl. Mech. Eng.* 341, 807–826. doi:10.1016/j.cma.2018.07.017

Hacein-Bey, L., and Provenzale, J. M. (2011). Current imaging assessment and treatment of intracranial aneurysms. *Am. J. Roentgenol.* 196, 32–44. doi:10.2214/AJR.10.5329

Han, S., Schirmer, C. M., and Modarres-Sadeghi, Y. (2020). A reduced-order model of a patient-specific cerebral aneurysm for rapid evaluation and treatment planning. *J. Biomechanics* 103, 109653. doi:10.1016/j.jbiomech.2020.109653

Hayashi, K., Handa, H., Nagasawa, S., Okumura, A., and Moritake, K. (1980). Stiffness and elastic behavior of human intracranial and extracranial arteries. *J. Biomechanics* 13, 175–184. doi:10.1016/0021-9290(80)90191-8

Hemmler, A., Lutz, B., Reeps, C., Kalender, G., and Gee, M. W. (2018). A methodology for *in silico* endovascular repair of abdominal aortic aneurysms. *Biomechanics Model. Mechanobiol.* 17, 1139–1164. doi:10.1007/s10237-018-1020-0

Hesthaven, J. S., Rozza, G., and Stamm, B. (2016). *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*. Springer International Publishing. doi:10.1007/978-3-319-22470-1

Hesthaven, J. S., and Ubbiali, S. (2018). Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J. Comput. Phys.* 363, 55–78. doi:10.1016/j.jcp.2018.02.037

Kardampiki, E., Vignali, E., Haxhiademi, D., Federici, D., Ferrante, E., Porziani, S., et al. (2022). The hemodynamic effect of modified blalock–taussig shunt morphologies: A computational analysis based on reduced order modeling. *Electron. Switz.* 11, 1930. doi:10.3390/electronics11131930

Karmonik, C., Strother, C. M., Chen, X., Deinzer, F., Klucznik, R., and Mawad, M. E. (2005). Stent-assisted coiling of intracranial aneurysms aided by virtual parent artery reconstruction. *Am. J. Neuroradiol.* 26, 2368–2370.

Kim, J. H., Kang, T. J., and Yu, W. R. (2008). Mechanical modeling of self-expandable stent fabricated using braiding technology. *J. Biomechanics* 41, 3202–3212. doi:10.1016/j.jbiomech.2008.08.005

King, R. M., Chueh, J. Y., Van Der Bom, I. M., Silva, C. F., Carniato, S. L., Spilberg, G., et al. (2012). The effect of intracranial stent implantation on the curvature of the cerebrovasculature. *Am. J. Neuroradiol.* 33, 1657–1662. doi:10.3174/ajnr.A3062

Krejza, J., Arkuszewski, M., Kasner, S. E., Weigele, J., Ustymowicz, A., Hurst, R. W., et al. (2006). Carotid artery diameter in men and women and the relation to body and neck size. *Stroke* 37, 1103–1105. doi:10.1161/01.STR.0000206440.48756.f7

Larrabide, I., Kim, M., Augsburger, L., Villa-Uriol, M. C., Rüfenacht, D., and Frangi, A. F. (2012). Fast virtual deployment of self-expandable stents: Method and *in vitro* evaluation for intracranial aneurysmal stenting. *Med. Image Anal.* 16, 721–730. doi:10.1016/j.media.2010.04.009

LatinHypercubeSampling.jl (2020). MrUrq/LatinHypercubeSampling.jl: Julia package for the creation of optimised Latin hypercube sampling plans. Avaliable At: https://mrurq.github.io.

Lauzeral, N., Borzacchiello, D., Kugler, M., George, D., Rémond, Y., Hostettler, A., et al. (2019). A model order reduction approach to create patient-specific mechanical

models of human liver in computational medicine applications. *Comput. Methods Programs Biomed.* 170, 95–106. doi:10.1016/j.cmpb.2019.01.003

Leng, X., Wang, Y., Xu, J., Jiang, Y., Zhang, X., and Xiang, J. (2018). Numerical simulation of patient-specific endovascular stenting and coiling for intracranial aneurysm surgical planning. *J. Transl. Med.* 16, 208. doi:10.1186/s12967-018-1573-9

Li, M., Jiang, Z., Yu, H., and Hong, T. (2013). Size ratio: A morphological factor predictive of the rupture of cerebral aneurysm? *Can. J. neurological Sci. Le J. Can. des Sci. neurologiques* 40, 366–371. doi:10.1017/S0317167100014323

Liu, A., and Huang, J. (2015). Treatment of intracranial aneurysms: Clipping versus coiling. *Curr. Cardiol. Rep.* 17, 628. doi:10.1007/s11886-015-0628-2

Liu, H., Cai, J., and Ong, Y. S. (2018). Remarks on multi-output Gaussian process regression. *Knowledge-Based Syst.* 144, 102–121. doi:10.1016/j.knosys.2017.12.034

Liu, M., Liang, L., and Sun, W. (2020). A generic physics-informed neural network-based constitutive model for soft biological tissues. *Comput. Methods Appl. Mech. Eng.* 372, 113402. doi:10.1016/j.cma.2020.113402

Ma, D., Dargush, G. F., Natarajan, S. K., Levy, E. I., Siddiqui, A. H., and Meng, H. (2012). Computer modeling of deployment and mechanical expansion of neurovascular flow diverter in patient-specific intracranial aneurysms. *J. Biomechanics* 45, 2256–2263. doi:10.1016/j.jbiomech.2012.06.013

Ma, D., Dumont, T. M., Kosukegawa, H., Ohta, M., Yang, X., Siddiqui, A. H., et al. (2013). High fidelity virtual stenting (HiFiVS) for intracranial aneurysm flow diversion: *In vitro* and *in silico*. *Ann. Biomed. Eng.* 41, 2143–2156. doi:10.1007/s10439-013-0808-4

Ma, D., Xiang, J., Choi, H., Dumont, T. M., Natarajan, S. K., Siddiqui, A. H., et al. (2014). Enhanced aneurysmal flow diversion using a dynamic push-pull technique: An experimental and modeling study. *Am. J. Neuroradiol.* 35, 1779–1785. doi:10.3174/ajnr.A3933

Maupu, C., Lebas, H., and Boulaftali, Y. (2022). Imaging modalities for intracranial aneurysm: More than meets the eye. *Front. Cardiovasc. Med.* 9, 793072–793079. doi:10.3389/fcvm.2022.793072

McKenna, C. G., and Vaughan, T. J. (2021). A finite element investigation on design parameters of bare and polymer-covered self-expanding wire braided stents. *J. Mech. Behav. Biomed. Mater.* 115, 104305. doi:10.1016/j.jmbbm.2020.104305

Mena, A., Bel, D., Alfaro, I., González, D., Cueto, E., and Chinesta, F. (2015). Towards a pancreatic surgery simulator based on model order reduction. *Adv. Model. Simul. Eng. Sci.* 2, 31–16. doi:10.1186/s40323-015-0049-1

Niroomandi, S., Alfaro, I., Cueto, E., and Chinesta, F. (2012a). Accounting for large deformations in real-time simulations of soft tissues based on reduced-order models. *Comput. Methods Programs Biomed.* 105, 1–12. doi:10.1016/j.cmpb.2010.06.012

Niroomandi, S., Alfaro, I., González, D., Cueto, E., and Chinesta, F. (2012b). Real-time simulation of surgery by reduced-order modeling and x-fem techniques. *Int. J. Numer. Methods Biomed. Eng.* 28, 574–588. doi:10.1002/cnm.1491

Perrin, D., Badel, P., Orgéas, L., Geindreau, C., Dumenil, A., Albertini, J. N., et al. (2015). Patient-specific numerical simulation of stent-graft deployment: Validation on three clinical cases. *J. Biomechanics* 48, 1868–1875. doi:10.1016/j.jbiomech.2015.04.031

Pierot, L. (2011). Flow diverters dans le traitement des anévrismes intracrâniens: Où en sommes-nous? *J. Neuroradiol.* 38, 40–46. doi:10.1016/j.neurad.2010.12.002

Quarteroni, A., Manzoni, A., and Negri, F. (2016). *Reduced Basis Methods for Partial Differential Equations*. Springer International Publishing. doi:10.1007/978-3-319-15431-2

Rahman, M., Smietana, J., Hauck, E., Hoh, B., Hopkins, N., Siddiqui, A., et al. (2010). Size ratio correlates with intracranial aneurysm rupture status: A prospective study. *Stroke* 41, 916–920. doi:10.1161/STROKEAHA.109.574244

Rinkel, G. J., Djibuti, M., Algra, A., and Van Gijn, J. (1998). Prevalence and risk of rupture of intracranial aneurysms: A systematic review. *Stroke* 29, 251–256. doi:10.1161/01.STR.29.1.251

Santo, N. D., Manzoni, A., and Pagani, S. (2020). Reduced-order modeling for applications to the cardiovascular system. *Applications* 251, 278. doi:10.1515/9783110499001-008

Schroeder, W., Martin, K., and Lorensen, B. (2006). *The Visualization Toolkit*. 4th Edn. Kitware. ISBN 978-1-930934-19-1.

Shiozaki, S., Otani, T., Fujimura, S., Takao, H., and Wada, S. (2020). Computational modeling of braided-stent deployment for interpreting the mechanism of stent flattening. *Int. J. Numer. Methods Biomed. Eng.* 1, e3335. doi:10.1002/cnm.3335

SignedDistanceField.jl (2022). SignedDistanceField.jl. Avaliable At: https://github.com.

Singh, A., Thakur, N., and Sharma, A. (2016). "A review of supervised machine learning algorithms," in Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, New Delhi, India, 16-18 March 2016 (IEEE), 1310–1315.

Spranger, K., Capelli, C., Bosi, G. M., Schievano, S., and Ventikos, Y. (2015). Comparison and calibration of a real-time virtual stenting algorithm using Finite Element Analysis and Genetic Algorithms. *Comput. Methods Appl. Mech. Eng.* 293, 462–480. doi:10.1016/j.cma.2015.03.022

Vlak, M. H., Algra, A., Brandenburg, R., and Rinkel, G. J. (2011). Prevalence of unruptured intracranial aneurysms, with emphasis on sex, age, comorbidity, country, and time period: A systematic review and meta-analysis. *Lancet Neurology* 10, 626–636. doi:10.1016/S1474-4422(11)70109-0

Wiebers, D., Whisnant, J. P., Huston, J., 3rd, Meissner, I., Brown, R. D., Jr, Piepgras, D. G., et al. (2003). Unruptured intracranial aneurysms: Natural history, clinical outcome, and risks of surgical and endovascular treatment. *Lancet* 362, 103–110. doi:10.1016/s0140-6736(03)13860-3

Wriggers, P. (2006). *Computational contact mechanics*. Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-32609-0

Zaccaria, A., Migliavacca, F., Pennati, G., and Petrini, L. (2020). Modeling of braided stents: Comparison of geometry reconstruction and contact strategies. *J. Biomechanics* 107, 109841. doi:10.1016/j.jbiomech.2020.109841

Zhong, J., Long, Y., Yan, H., Meng, Q., Zhao, J., Zhang, Y., et al. (2016). Fast virtual stenting with active contour models in intracranial aneurysm. *Sci. Rep.* 6, 21724–21729. doi:10.1038/srep21724

Zyłkowski, J., Rosiak, G., and Spinczyk, D. (2018). Semi-automatic measurements and description of the geometry of vascular tree based on Bézier spline curves: Application to cerebral arteries. *Biomed. Eng. Online* 17, 115–121. doi:10.1186/s12938-018-0547-8

Appendix: Gaussian process regression

A brief theoretical introduction to single-output GPR is presented here, more details can be found in (Carè and Camporeale, 2018). The approach employed to extend single-output GPR to multi-output problems is to model each of the outputs independently. This is referred to as independent single-output GPR (IGPR).

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ be a generic training dataset where $\mathbf{x}_i$ represents one input vector and y_i the corresponding output. We assume that y_i are noisy observations of an unknown regression function $f(\mathbf{x}_i)$:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma_i^2). \tag{16}$$

GPR consists in calculating the probability distribution over all admissible functions that fit the input data and making predictions when fed with new input data. This is possible by considering $f(\mathbf{x}_i)$ distributed as a Gaussian process (GP):

$$f(\mathbf{x}_i) \sim \mathrm{GP}(0, \kappa(\mathbf{x}, \mathbf{x}')). \tag{17}$$

A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution. It is fully described by a mean function (considered 0 for simplicity) and a covariance function $\kappa(\mathbf{x}, \mathbf{x}')$, known as *kernel*, which should be chosen carefully according to the problem under study. The most commonly used is the *squared exponential (SE)* kernel, also known as *radial basis function (RBF)* kernel:

$$\kappa(x_i, x_j) = \sigma_\kappa^2 \exp\left(-\frac{\|x_i - x_j\|^2}{2l}\right), \tag{18}$$

where the standard deviation σ_κ and the lengthscale l represent the two kernel hyperparameters.

Following the GP definition, the prior Gaussian distribution for the outputs is:

$$\mathbf{y}|\mathbf{X} \sim \mathcal{N}(0, \mathbf{K}_y), \quad \mathbf{K}_y = \kappa(\mathbf{X}, \mathbf{X}) + \sigma^2 \mathbf{I}, \tag{19}$$

where $\mathbf{y} = \{y_i, y_2, \ldots, y_N\}$ and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$.

Predictions of the noise-free outputs $\mathbf{f}^\star$ on new input data $\mathbf{X}^\star$ are made by exploiting the joint probability distribution:

$$\mathbf{f}, \mathbf{f}^\star | \mathbf{X}, \mathbf{X}^\star = \begin{bmatrix} \mathbf{f} \\ \mathbf{f}^\star \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}^\star \\ \mathbf{K}^{\star\mathrm{T}} & \mathbf{K}^{\star\star} \end{bmatrix} \right), \tag{20}$$

where $\mathbf{K}^{\star\star} = \kappa(\mathbf{X}^\star, \mathbf{X}^\star)$ and $\mathbf{K}^\star = \kappa(\mathbf{X}, \mathbf{X}^\star)$. Thus, following the conditioning theorem for Gaussians, the posterior predictive distribution of $\mathbf{f}^\star$ is:

$$\mathbf{f}^\star | \mathbf{X}, \mathbf{X}^\star, \mathbf{f} \sim \mathcal{N}(\mathbf{m}, \mathbf{C}), \quad \mathbf{m} = \mathbf{K}^{\star\mathrm{T}} \mathbf{K}_y^{-1} \mathbf{y}, \quad \mathbf{C} = \mathbf{K}^{\star\star} - \mathbf{K}^{\star\mathrm{T}} \mathbf{K}_y^{-1} \mathbf{K}^\star. \tag{21}$$

The powerful aspect of GPR is that the output of the prediction is a multivariate normal distribution, characterised by a mean vector and a covariance matrix: in our case, $\mathcal{N}(\mathbf{u}_L, \mathbf{K}_L)$. Therefore, the full-order solution is also a multivariate normal distribution, $\mathcal{N}(\mathbf{u}_p = \mathbf{V}\mathbf{u}_L, \mathbf{K}_p = \mathbf{V}\mathbf{K}_L\mathbf{V}^T)$. We can visualise the prediction uncertainty by sampling this distribution using the Matlab function mvnrnd(). Since in Matlab we are limited to IGPR, i.e., one GPR model for each projection coefficient, the covariance matrix is diagonal.