Augmented patient-specific functional medical imaging by implicit manifold learning

Robert Rapadamnaba, Franck Nicoud, Bijan Mohammadi

Augmented patient-specific functional medical imaging by implicit manifold learning

Robert Rapadamnaba[1,*], Franck Nicoud[1] and Bijan Mohammadi[1]

[1] *IMAG, Université de Montpellier, CNRS, CC051, 34095 Montpellier, France.*

SUMMARY

This paper uses machine learning to enrich magnetic resonance angiography and magnetic resonance imaging acquisitions. A convolutional neural network is built and trained over a synthetic database linking geometrical parameters and mechanical characteristics of the arteries to blood flow rates and pressures in an arterial network. Once properly trained, the resulting neural network can be used in order to predict blood pressure in cerebral arteries noninvasively in nearly real-time. One challenge here is that not all input variables present in the synthetic database are known from patient-specific medical data. To overcome this challenge, a learning technique, which we refer to as implicit manifold learning, is employed: in this view, the input and output data of the neural network are selected based on their availability from medical measurements rather than being defined from the mechanical description of the arterial system. The results show the potential of the method and that machine learning is an alternative to costly ensemble based inversion involving sophisticated fluid structure models.

KEY WORDS: machine learning, transfer learning, convolutional neural network, hemodynamic problems, noninvasive pressure estimation.

*Correspondence to: Robert Rapadamnaba, IMAG, Université de Montpellier, CNRS, CC051, 34095 Montpellier, France. Email: robert.rapadamnaba@umontpellier.fr

1. INTRODUCTION

At the junction of optimization, statistics and computer science, machine learning (ML) has gained tremendous interest throughout all scientific disciplines. One can say without exaggeration that it is a revolution in the way solutions of scientific problems are considered. This revolution is very noticeable in biomedical applications with the works of Rahul, Liang et al, Adam et al, Pozin et al, Koeppl et al, Koprowski et al, Cang et al, Luo et al, Cilla et al [1–9], and also more recently, with the works of Goceri, Zhang et al, Chen et al, and Nguyen and Wei [10–13], to name just a few. According to several researchers, ML will undoubtedly continue to significantly revolutionize biomedical research and global health care in the future [1, 14, 15]. Indeed, fueled by increasingly powerful computers and available storage capacities, ML methods are able to deal with large, complex and heterogeneous data typically found in biomedical applications [1, 13–17]. Thus, ML can contribute to identify and engineer features from the data, perform more robust predictions and therefore lead to more accurate diagnostic algorithms and individualize patient treatments [15, 18–22].

ML-based strategies most often rely on models applied to a carefully constructed dataset so that the training and test data are drawn from the same feature space and the same marginal probability distribution [23]. As pointed out by Ruder [24], these models sorely lacks the ability to generalize to conditions that are different from those encountered during learning and for which the models might be unable to make reliable predictions. This is the case, for instance, when a model learns from simulations and is being applied to deal with actual data. As data from the real world is messy and may contain several new scenarii, many of which were not present during the learning step, the model tuned from numerical data suffers from a significant loss in performance or even breaks down completely when used in practice [24]. In the field of ML, the ability of a model to transfer knowledge to new conditions is generally known as transfer learning (TL) and this is one of the key features used in this paper.

Many examples exist in the ML literature where TL has been applied successfuly: text categorization, text and Web-document classification [25–28], learning in real-time strategy game [29], text mining [27, 30], natural language processing [31], WiFi-based Indoor Localization [32], reinforcement learning [33–37], to cite a few. In this paper, we apply TL, in particular learning from simulations, to the prediction of blood pressure in a cardiovascular system.

Learning from simulations is one particular application of TL commonly used in ML when one needs to deal with data, which is either rare, incomplete, expensive or simply dangerous to obtain [24]. In all these situations, analytic models or numerical simulations can fill the gap. In previous works, the use of an algorithm based on ensemble Kalman filter (EnKF) technique coupled to a 0D compartement network has allowed to noninvasively estimate patient-specific blood pressure in cerebral

arteries [38, 39]. However, this procedure, which requires inverse problem solving, is too long, expensive and difficult to integrate in a medical acquisition device. The present work aims at remedying that with focus on the following three key messages:

- Use supervised ML together with a Convolutional Neural Network (CNN) — a type of ML algorithms used with great success in miscellaneous image recognition applications and prediction tasks, such as handwriting recognition and face detection [40–43] — as an alternative to the ensemble based inversion procedure by EnKF used in our previous works for cerebral blood pressure estimation. In particular, on the basis of magnetic resonance angiography and magnetic resonance imaging (MRA & MRI) acquisitions provided by physicians and using a learning database generated with the same numerical procedure as that used in the inversion procedure, the aim is to recover the EnKF outcomes previously achieved by Lal et al [38] through the solution of an inverse problem. This allows to augment/enrich already available information in MRA & MRI images.
- Use "implicit learning on a manifold" to deal with situations where part of the information in the input variables are missing while observations are available on some output variables. This is why the term "implicit" is introduced. The term "manifold" is used to reflect the fact that among all the variables contained in the database, only a subset of them will be used for the learning. This is like working on a vector subspace and consists of reorganizing the database into *structured* variables in order to take into account unavailable patient-specific data. Therefore, "implicit learning on a manifold" should not be seen as a particular Artificial Intelligence (AI) algorithm. In fact, it should be regarded as just a multi-output convolutional neural network. Thus, any multi-output regression method (e.g. linear models, multi-output random forests or other forms of neural nets with autoencoding) could have been used instead of this multi-output convolutional neural network implementation.
- Apply the outcome of the learning from simulations to patient-specific data and we use the term "transfer" in that sense. The network and its coefficients obtained during the learning from simulations are kept unchanged when applied on patient-specific data.

The remainder of the paper is organized as follows. First, an overview of the methodology followed in this work is provided in Section 2. Section 3 details the procedure for building the synthetic database. Section 4 presents the state of the art in ML and discusses some mathematical aspects of the CNN approach used in this study. Section 5 illustrates the effectiveness of the method by considering a very simple problem while Section 6 shows its application for the estimation of blood pressure in cerebral

arteries. Finally, Section 7 discusses some of the limitations of the present study and Section 8 provides concluding remarks.

2. PRELIMINARIES

2.1. General methodology

Figure 1 presents the general flowchart for this work with the following main steps:

1. **Hemodynamic and morphological data extraction**: first, from magnetic resonance angiography and magnetic resonance imaging acquisitions and segmentation of a 3D time of flight magnetic resonance angiography (3D-TOF-MRA) — `dicom` files provided by the Department of Neuroradiology of the *Centre Hospitalier Régional Universitaire de Montpellier (CHRU), Montpellier, France*, — blood flow rates in ascending aorta (AA), right and left internal carotid arteries (R-ICA and L-ICA) have been extracted using the `GTFlow` sofware (`http://www.gyrotools.com/products/gt-flow.html/`) together with the morphological data relevant to geometric measurements and the morphology of some arteries through the use of the RadiANT DICOM Viewer software (`http://www.radiantviewer.com/`).
2. **Patient-specific arterial network construction**: on the basis of these images, a patient-specific arterial network of 33 arteries as shown in Figure 2 consisting of the aorta, vertebral, carotid and brachial arteries together with the complete circle of Willis has been then constructed. For more details, the reader is referred to previous works [38, 39].
3. **Machine learning**
   (a) **Database generation**: the next step has consisted in generating synthetic data by performing a series of forward simulations of the blood flow model ($\mathcal{M}$) described in detail below in Section 2.2. The outcome of these simulations relates the *model input parameters* (namely, the blood flow rates in AA, the geometrical (radius r, thickness h, and length l) and mechanical characteristics (Young moduli) of the 33 arteries as well as the Windkessel three-element parameters) to the *model output parameters* (namely, the blood flow and pressure in the 33 arteries.)
   (b) **The use of CNN as ML algorithm**: subsequently, a multi-output CNN-based approach was introduced in order to learn on *structured* variables from the database, the relationship between CNN input and output parameters, and then, apply the acquired knowledge to patient-specific data for blood pressure estimation.

4. **Validation**: Finally, to test the accuracy of the method, pressure estimated using CNN was compared with that previously predicted in Lal et al [38] by solving an inverse problem using EnKF.

Compared with the method used in previous works [38, 39], the novelty in the general framework proposed here is the use of ML method instead of EnKF-based parameter estimation algorithm.

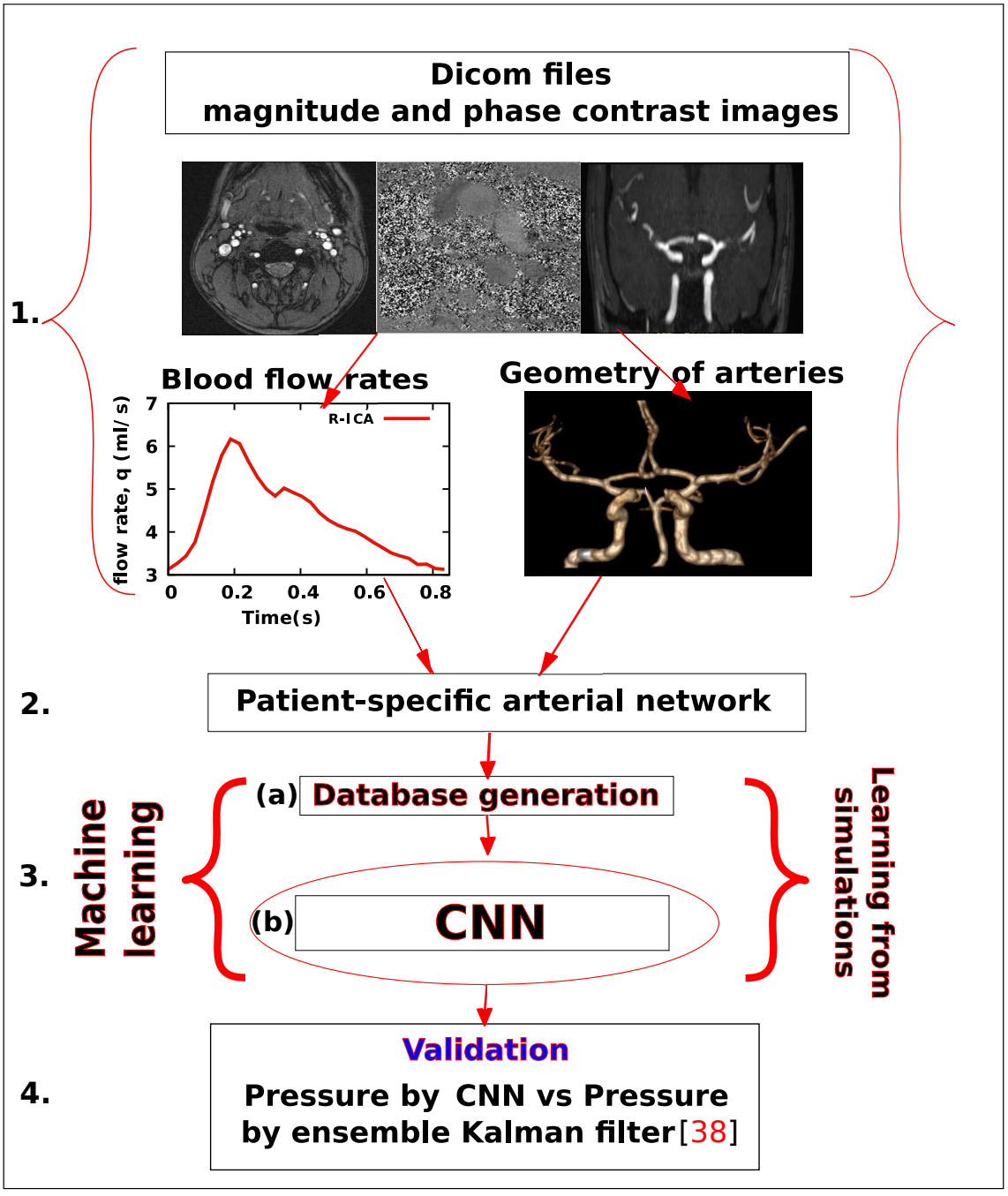

Figure 1. The general framework of the proposed methodology. The numbers on the left side refer to the four steps of the general methodology described above.

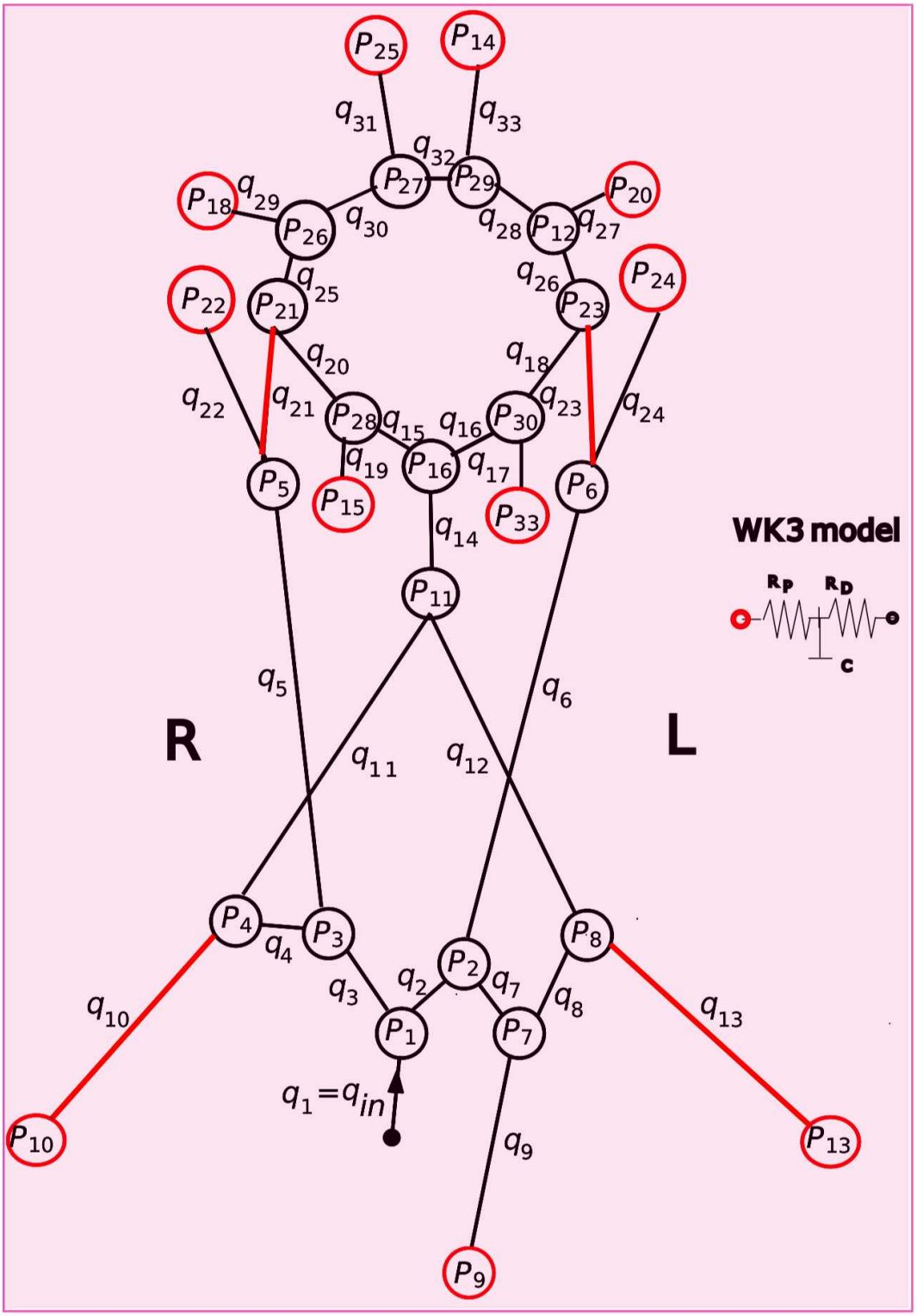

Figure 2. The 0D arterial network model [38] of the patient-specific upper body arteries and the circle of Willis. The arterial segments in red and the red circles represent the four arteries of interest examined in the rest of paper and the WK3 model, respectively.

2.2. *The blood flow model*

Mathematical modeling is a powerful tool for a better understanding of the cardiovascular system in an inexpensive and noninvasive way. There are several cardiovascular models (0D, 1D, 3D) available in the literature, each coming with its advantages and drawbacks. In this work, we adopted a 0D formulation, also known as lumped-parameter model, due to its simplicity and effectiveness in describing first order features of the human cardiovascular system.

In general, the mathematical formulation of the lumped-parameter models is derived by decomposing the cardiovascular system into single arterial segments or compartments, each segment being represented with a single electrical circuit, consisting of resistance R, inductance L, and compliance C, as shown in Figure 3. In this way, the full arterial network is built by connecting single electrical compartments together.

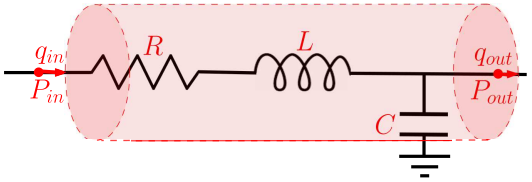

Figure 3. A single compartment circuit illustration.

Applying the laws of conservation of mass and momentum to a single compartment filled with an incompressible Newtonian fluid, the following linear first order ordinary differential equations can be obtained:

$$\begin{cases} C\dot{P}_{out} = & q_{in} - q_{out} \\ L\dot{q}_{in} = & P_{in} - P_{out} - Rq_{in}, \end{cases} \tag{1}$$

where (P_{in}, q_{in}) and (P_{out}, q_{out}) refer to the pressure and flow rate at the inlet and outlet of the vascular segment, respectively. For arteries with a radius $r < 0.2$ cm, the inertial effect is ignored [44] and the flow rate is given by $q_{in} = (P_{in} - P_{out})/R$. Let r, h, l, μ, ρ, and E be the radius, the wall thickness, the length, the blood viscosity, the blood density and the Young modulus of the arterial segment, respectively. The three parameters R, L, C are obtained from physical principles (momentum and mass conservation) applied to the arterial sector and read [45]:

- the viscous flow resistance, R, is derived from the Poiseuille's equation, which describes the relation between pressure drop, ΔP, and the steady blood flow, q, through a uniform and rigid blood vessel: $R = 8\mu l/\pi r^4$;
- the inductance of the vessel, L, which represents the acceleration and deceleration of the inertial blood at each heart beat, is equal to $L = \rho l/\pi r^2$;

- the vessel compliance, C, which represents the ability of a vessel wall to expand and contract with changes in pressure, is equal to $C = 3\pi r^3 l/2Eh$.

In order to close the arterial network and accommodate the cumulative effects of vessels beyond the terminal segments, the three-element Windkessel model (WK3-lumped parameter model) [46, 47] is prescribed at the outlet of each terminal segment. As a reminder, the equation relating the instantaneous blood pressure and the flow rate reads as follows:

$$\dot{p}(t) + \frac{p}{R_D C} = R_P\, \dot{q}(t) + \frac{qR_T}{R_D C}, \tag{2}$$

where p, q, C, R_P, and R_D refer to the instantaneous pressure at the inlet of the WK3 model, the instantaneous flow rate, the compliance, the proximal resistance, and the distal resistance of the vascular beds, respectively, and R_T refers to the total peripheral resistance meaning that $R_T = R_P + R_D$. For simplicity, the blood flow model — let us refer it to as $(\mathcal{M})$ — can be formally expressed as follows :

$$(\mathcal{M}) : \begin{cases} \dot{Y}(t) = & F_X(Y(t)) \\ Y(0) = & Y_0, \end{cases} \tag{3}$$

where $Y(t) = (P(t), q(t))^\top$, Y_0 is the initial condition, X represents the geometrical parameters of the problem (i.e r, h, l), the inlet flow rate from AA, the Young moduli, the Windkessel WK3 boundary conditions [38] at the same time, and F_X is a linear or nonlinear function defined by the blood flow model describing the relation between $Y(t)$ and its time derivatives $\dot{Y}(t)$. Here and for the rest of the paper, the notation 'function with suscript X', for examle F_X, means that the function F depends on the parameter X. This is a generic formalism for any 0D arterial network and therefore includes the 0D formulation of the cardiovascular network investigated in the current study (Figure 2). This network is taken from Lal et al [38] because both studies focus on the same patient.

3. IN SILICO DATABASE GENERATION

In this section, we describe the procedure for generating synthetic data. This data consists of geometrical parameters (the radius, r, the wall thickness, h, and the length, l), physical parameters (the Young moduli, E), blood flow rates, q, blood pressures, P for the 33 elements of the arterial model (Figure 2) on top of the eleven WK3 models used as boundary conditions (the proximal resistances, R_P, the distal resistances, R_D, and the compliances, C).

For a forward simulation using the blood flow model ($\mathcal{M}$) presented in Section 2.2, this data is interrelated in the sense that the variables r, h, l, E, R_P, R_D, C of the network associated with the variable q_{in}, the inlet flow rate (the blood flow rate in AA), represent the input parameters of the model, which we refer to as *model input parameters*, and the variables q and P, the output parameters, which we refer to as *model output parameters*. As a reminder, the Young moduli E are given by $Eh = r(k_1\, e^{k_2 r} + k_3)$ [48] where k_1, k_2, k_3 are model constants and the inlet flow rate q_{in} is defined using two components α and τ as follows [38, 39]:

$$q_{in}(t) = \begin{cases} \alpha \sin(\pi t/\tau) & \text{if } t < \tau, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where α = 302 ml/s and τ = 0.3 s are nominal values taken from Lal et al [38]. These will be randomly perturbated below following a Gaussian distribution with means equal to these nominal values and standard deviations σ, which will be specified later, to produce samples of q_{in}. For more details, the reader is referred to Lal et al [38].

To generate the database, blood rheological parameters were fixed as: the blood density ρ = 1050 kg·m^{-3} and the blood viscosity μ = 0.004 Pa·s, and the same WK3 boundary conditions for left and right pairs of terminal compartments were assumed. This means that the terminal compartments such as the segment connecting P_{27} and P_{25} and the one linking P_{29} and P_{14} in Figure 2, for instance, are assigned with the same WK3 boundary conditions. A series of forward simulations of the blood flow model ($\mathcal{M}$) is then performed using a series of perturbated *model input parameters* ($\tilde{q}_{in}$, $\tilde{r}$, $\tilde{h}$, $\tilde{l}$, $\tilde{E}$, $\tilde{R}_P$, $\tilde{R}_D$, $\tilde{C}$). The series of outcomes obtained from these simulations ($P = \tilde{P}_1, \cdots, \tilde{P}_{33}$, $q = \tilde{q}_1, \cdots, \tilde{q}_{33}$) associated with the series of perturbated *model input parameters* constitutes the database. This is illustrated on Figure 4. Note that, while not expressly stated, all the parameters in the database representing blood flow rates and pressures are time-dependent. It is also worth noting that the parameters in the database can be subdivided into two categories: the class of known parameters (easy to measure) and the class of unknown parameters (difficult to measure). Table I provides a general overview of these different parameters listed by category.

To obtain the series of perturbated *model input parameters* for each forward simulation of the blood flow model ($\mathcal{M}$), we randomly perturbate each *model input parameter* following a Gaussian distribution with a mean equals to the nominal value of each input parameter taken from Lal et al [38], and a standard deviation σ of 5% of this nominal value. The standard deviation σ of 5% of each nominal value is chosen in order to obtain perturbated *model input parameters* tending to realistic values encountered in the literature. E being given by the relation $Eh = r(k_1\, e^{k_2 r} + k_3)$, $\tilde{E}$ is obtained by perturbating the

parameters appearing in the relation giving E meaning that $\tilde{E}\tilde{h} = \tilde{r}(\tilde{k}_1\, e^{\tilde{k}_2\tilde{r}} + \tilde{k}_3)$, where each of the 3 constants $\tilde{k}_1$, $\tilde{k}_2$, and $\tilde{k}_3$ is also obtained from a Gaussian distribution with a mean equals to the nominal value of each constant k_1, k_2, and k_3 taken from Lal et al [38] and a standard deviation of 5% of this nominal value.

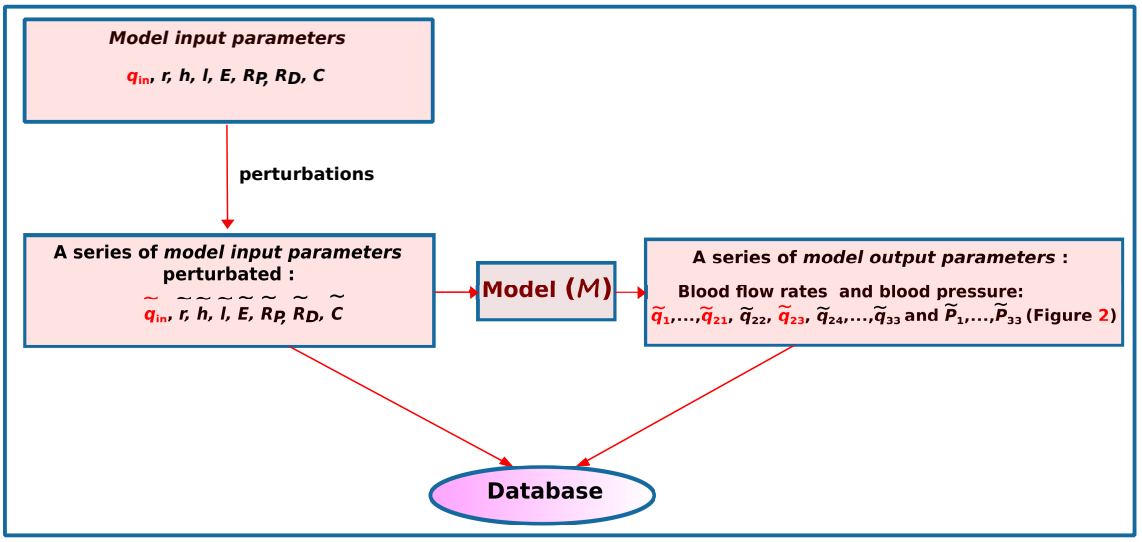

Figure 4. Illustration of the database generation procedure. In the case studied here, the parameters in red are considered as known parameters (easy-to-measure parameters) while the remainder is considered to be unknown.

Table I. The two categories of parameters in the database: the known parameters (easy-to-measure parameters) and the unknown parameters (not-easy-to-measure parameters).

Known parameters	Unknown parameters
blood flow rates in AA, $\tilde{q}_{in} = \tilde{q}_1$	geometrical parameters of the 33 arteries, $\tilde{r}_1, \cdots, \tilde{r}_{33}, \tilde{h}_1, \cdots, \tilde{h}_{33}, \tilde{l}_1, \cdots, \tilde{l}_{33}$
blood flow rates in R-ICA, $\tilde{q}_{21}$	Young moduli of the 33 arteries, $\tilde{E}_1, \cdots, \tilde{E}_{33}$
blood flow rates in L-ICA, $\tilde{q}_{23}$	blood pressures in the 33 arteries, $\tilde{P}_1, \cdots, \tilde{P}_{33}$
—	blood flow rates in 30 arteries, $\tilde{q}_2, \cdots, \tilde{q}_{20}, \tilde{q}_{22}, \tilde{q}_{24}, \cdots, \tilde{q}_{33}$

4. THE STATE OF THE ART IN MACHINE LEARNING

A tremendous number of ML algorithms have been developed, which all share the same goal: to provide sufficient flexibility to minimize training error but, at the same time, allow generalization to new data sets, all in a computationally efficient way [1]. Among these algorithms, the most commonly used ones are Linear Regression, Logistic Regression, Linear Discriminant Analysis, Decision Tree, Support Vector Machine (SVM), Naive Bayes, k-Nearest Neighbors (kNN), Learning Vector Quantization (LVQ), K-Means, Bagging, Random Forest, Dimensionality Reduction Algorithms, Convolutional Neural Networks (CNN), Gradient Boosting algorithms and AdaBoost [49–52]. These algorithms tasks mainly

include regression, classification, predictive modelling, clustering, association or survival analysis, link mining, and dimensionality reduction [52–54]. The details of these different algorithms are beyond the scope of this paper. However, CNN is discussed below because of its ability to manage complicated relationships between input and output data that are not easily captured by manual measurement [14], its relevance to biomedical research, its frequent use within the literature [15] and its use within the current work.

4.1. The implicit CNN : notation and database destination

The first step in our approach is to receive from a user raw data made of a set of scenarii linking input and output variables (let us refer to them as input data and output data, respectively). In the sequel, as we are interested in biomedical applications, the considered database will consist of a set of N scenarii linking n input real variables (these are typically blood flow rates values at temporal discretization points) to m output real variables (these also typically correspond to blood pressure values at temporal discretization points): $\{X \in \mathbb{R}^n \rightarrow Y \in \mathbb{R}^m\}$ where $X = (X_1, ..., X_n)$ and $Y = (Y_1, ..., Y_m)$. Hence, the required data is as follows:

$$\begin{gathered} X^1 = (X_1^1, ..., X_n^1) \rightarrow Y^1 = (Y_1^1, ..., Y_m^1), \\ \vdots \\ X^N = (X_1^N, ..., X_n^N) \rightarrow Y^N = (Y_1^N, ..., Y_m^N). \end{gathered}$$

In the applications we are interested in, data is often rare, expensive to obtain or confidential. The aim, therefore, is not to handle very large amount of data, but rather see the potential of learning of CNNs based on small amount of data in opposition to big data situations. The destination of the data is rather classical. The database is splitted in three subsets to be used to train, validate and a posteriori test the network. In this splitting, the major part of data is used for learning. A second subset is used to make sure to avoid over-fitting during learning (this is called validation). The third subset is for a posteriori test of the quality of the network. This kind of splitting is very classical in statistical data treatment and widely described in dedicated manuscripts [55].

4.2. The implicit CNN algorithm construction

Classical CNN algorithms are usually built to mimic and replicate the investigated physical model process. This means that in the classical CNN, the input and output data of the neural network are exactly the same as those used in the physical model considered. This is illustrated in Figure 5 in the case examined here, where the input data and the output data of the network are directly derived from the series of perturbated *model input parameters* and from the series of *model output parameters*,

respectively. Note that except the blood flow rates $\tilde{q}_{in}$ and $(\tilde{q}_{21}, \tilde{q}_{23})$ in the input data and the output data, respectively, the other parameters constituting the input/output data in the classical CNN are unknown (not easy to measure). This prevents us from performing the a posteriori test of the quality of the network, hence the idea of changing the structure of the classical CNN algorithm so that all the known parameters in the database can be gathered as consisting of the network input data.

The CNN algorithm proposed here, which we refer to as implicit CNN, is not usual. One of its particularities is that its input data is neither usual nor classical. Indeed, in contrast to the classical approach, the input and output data of the network in the implicit CNN algorithm does not mimic those from the physical model. They are rather reorganized in *structured* variables. The example shown in Figure 6 demonstrates such organizational change in the implicit CNN : the input data consists of one subset of the series of perturbated *model input parameters*, $\tilde{q}_{in}$, and one subset of the series of *model output parameters* (blood flow rates $\tilde{q}_1, \cdots, \tilde{q}_{33}$), and the output data consists of the remaining part of the series of *model output parameters* (blood pressure $\tilde{P}_1, \cdots, \tilde{P}_{33}$). However, in practice, we are interested in predicting the blood pressure in a specific artery and not in all at once. Thus, instead of using all available information as shown in Figure 6, a simpler construction where the input and output data of the implicit CNN will consist of $\tilde{q}_{in}$ and the series of blood flow rates in the artery of interest on the one hand, and on the other hand, the series of blood pressure in the same artery (see Figures 10 – 13 below) takes over from it. This unusual construction is motivated by the available parameters from the patient's MRA & MRI images. Indeed, only the parameters available (known parameters) within the patient-specific data (see Table I), namely the parameters extracted from MRA & MRI images and not requiring the resolution of an inverse problem — the blood flow rates in AA, R-ICA and L-ICA, $\tilde{q}_{in}$, $\tilde{q}_{21}$, and $\tilde{q}_{23}$ (see Figure 2), respectively are used as input data of the network. The idea behind this construction is to obtain a network capable of linking only the easy-to-measure parameters (known parameters in *model input parameters*) from the patient's MRA & MRI images to the *model output parameters*. This implies that the information in the unknown parameters from *model input parameters* which are not used for building the network is implicitly contained in the *model output parameters* used for the network construction and can therefore be indirectly recovered. This means that exploiting only known parameters from *model input parameters* as input data of the network allows to access to hidden information in the unknown parameters not used. In addition to allowing the a posteriori test of the quality of the network, this construction avoids the problem of unknown parameter estimation through inverse problem solving before performing the convolutional neural network construction.

Another particularity of our implicit CNN concerns the definition of the network. In the classical CNN techniques, the number of layers and the number of hidden variables in each layer must be a priori defined by the user before the parameters of the network are found through error backpropagation. This is known

to require lots of know-how by the user. We avoid this a priori definition introducing an incremental procedure for the definition of the network structure. We proceed in a systematic way combining an incremental construction of the network with the use of the information available through activation functions. This means that different network depths are tested with increasing number of layers. For each of these networks, a maximum number of variables per layer is prescribed. In our case, we consider a maximum allowance of 200 hidden variables for all layers. Eventually, after this optimization procedure, only the variables which are found to sufficiently contribute to the network outcome are retained. These correspond to those with the activation function beyond a given user-defined threshold. The network we retain is the one with the best fitting capability over the learning database while avoiding overfitting. This optimization of the network structure is interesting for both the optimization and inference steps. For the former, once a given variable is found not being of sufficient importance, it is removed from the optimization set. This means that search activities along this variable are abandoned hence reducing the size of the optimization space. For the latter, a lighter network will obviously permit more effective inference steps in term of computational effort. Figure 7 illustrates this procedure for the simple example given in Section 5. It shows the evolution of the mean average error over the test database for increasing network depth and the evolution of the cumulative number of the variables of the network. The network, which shows the best compromise between accuracy and complexity is found to be of depth 20: adding extra layers and variables do not significantly improve the quality of the results. The results presented below in Figure 9 corresponds to the outcome of this network. Note also that in our implicit CNN, 1D convolution operations with Gaussian kernels are used in convolution layers as we handle time series. The supports of the convolutions are also automatically identified in the same way as the optimal number of layers of the network.

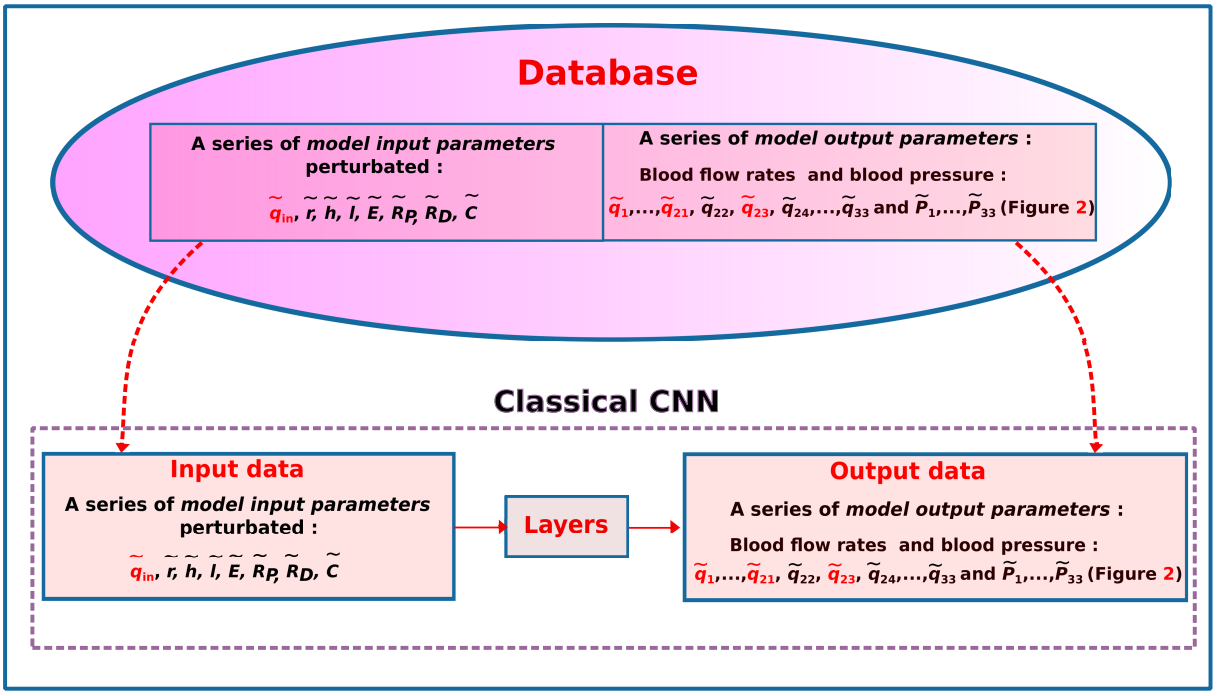

Figure 5. The classical CNN construction. The network input data consists of a series of *model input parameters* perturbated $\tilde{q}_{in}, \tilde{r}, \tilde{h}, \tilde{l}, \tilde{E}, \tilde{R}_P, \tilde{R}_D, \tilde{C}$, and the network output data is composed of the corresponding series of *model output parameters* ($\tilde{q}_1, \cdots, \tilde{q}_{33}$ and $\tilde{P}_1, \cdots, \tilde{P}_{33}$ in Figure 2). The known parameters are in red and the unknown ones are in black.

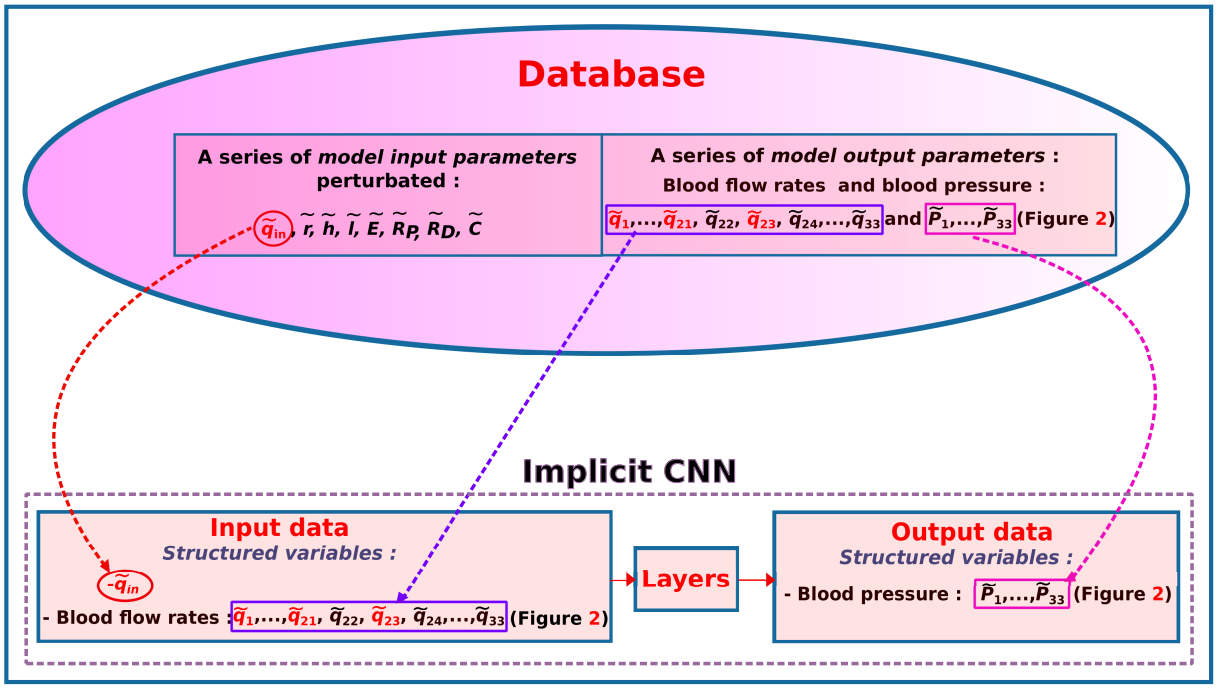

Figure 6. The implicit CNN construction. The network input data consists of a series of *structured* variables: the series of known parameters from the series of *model input parameters* perturbated, namely the perturbated inlet flow rate, $\tilde{q}_{in}$, and the corresponding series of blood flow rates from the series of *model output parameters* ($\tilde{q}_1, \cdots, \tilde{q}_{33}$ in Figure 2), and the network output data is composed of the corresponding series of blood pressure from the series of *model output parameters* ($\tilde{P}_1, \cdots, \tilde{P}_{33}$ in Figure 2). The known parameters are in red and the unknown ones are in black.

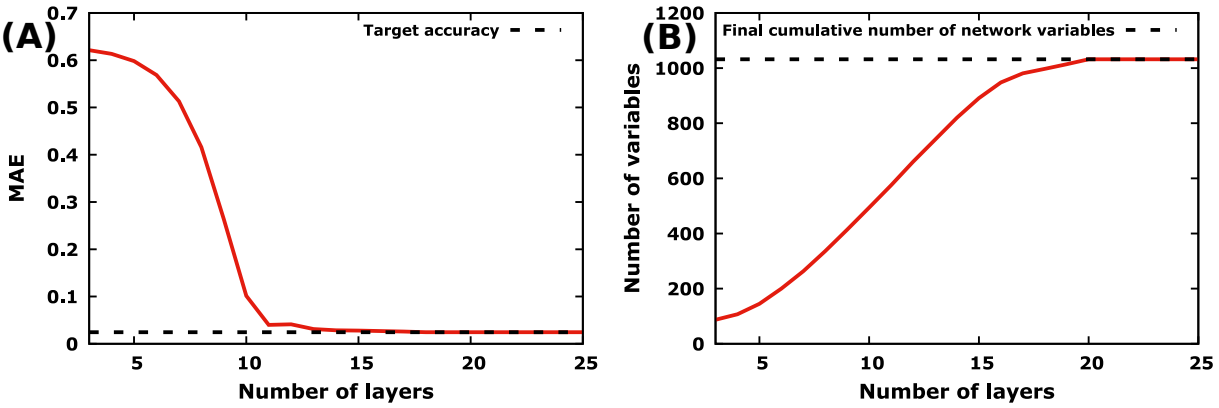

Figure 7. Incremental implicit network construction for the simple example given in Section 5. A: Mean average errors versus number of layers and B: total number of network cumulative variables versus number of layers.

5. ILLUSTRATION ON A SIMPLE EXAMPLE: CNN AND SOLUTION OF A FOWARD PROBLEM

Let us illustrate the proposed approach on the solution of an ordinary linear differential equation:

$$\begin{cases} \dot{p}(t) &= f_X(t), \\ \dot{q}(t) &= g_X(t), \\ p(0) &= p_0, \\ q(0) &= q_0, \end{cases} \tag{5}$$

with $X = (X_1, X_2)$, $f_X(t) = X_1 \cos(X_1 t + X_2)$, $g_X(t) = -X_1 \sin(X_1 t + X_2)$, $p_0 = \sin(X_2)$, and $q_0 = \cos(X_2)$.

Setting $Y = (p, q)^\top$, $H_X(t) = (f_X(t), g_X(t))^\top$, $Y(0) = (p(0), q(0))^\top$ and $Y_0 = (p_0, q_0)^\top$, this ordinary linear differential equation can be restated as:

$$\begin{cases} \dot{Y}(t) &= H_X(t), \\ Y(0) &= Y_0. \end{cases} \tag{6}$$

The solution Y_* of the latter differential equation is trivial and we have $Y_*(t) = (p_*(t), q_*(t))^\top = (\sin(X_1 t + X_2), \cos(X_1 t + X_2))^\top$. In order to apply CNN-based technique to this simple problem, a database containing samples of $X \in \mathbb{R}^n$ and $Y \in \mathbb{R}^m$, is digitally generated.

For the classical CNN, we have $n = 2$, the size of input data X. Moreover, the solution Y_* is represented by Y (output data), a vector of size $m = 86$ corresponding to the number of points in a discrete representation of the solution Y_* (43 for each of the two components of Y). In this test case,

the database contains $N = 200$ samples of Y obtained for 200 values of X. In the applications we are interested in, $N = 200$ different scenarii is already quite large as data is often not that easy to obtain. This choice of 200 samples is not coerced. This is arbitrary and we could, of course, have chosen any other reasonable number of samples.

For the implicit CNN, data is reconstructed as follows: the input data X' is built from a subset of X and a subset of Y. For instance, one can consider $X' = (X_1, q)$. The output data Y' is made of the remaining part of Y, p. Thus, the size of input data is now $n = 44$ (1 for X_1 and 43 for q), and that of output data is $m = 43$. Note that X_2 belongs neither to the input nor the output but its effect on the output is present, although not explicitly written.

The results obtained using the classical CNN and the implicit CNN for this simple problem are illustrated in Figures 8 and 9, respectively. In these Figures, the upper parts (Figures 8A and 9A) show the networks determined to properly represent the actual “physical" system described in Eq. 6 : a network consisting of 30 locally fully connected layers with up to 155 hidden variables for the classical CNN, and a network of 20 locally fully connected layers with up to 84 hidden variables for the implicit one. By locally fully connected we mean that the variables of two successive layers are connected. In the remaining part of these Figures (Figures 8B, C and 9B), the desired solution is referred to as Target, the prediction of the desired solution referred to as CNN, and data used for learning referred to as Learning data. Comparing the Target and CNN data, the results are very good with a relative error of less than 3 % and mean absolute errors (MAE) of 0.05 and of 0.024 for the classical and implicit CNN, respectively. Note that the implicit CNN is able to recover the partial output p without using the explicit knowledge of the X_2 input. This is a big advantage compared to the classical CNN formulation, especially in any case where X_2 is difficult to measure.

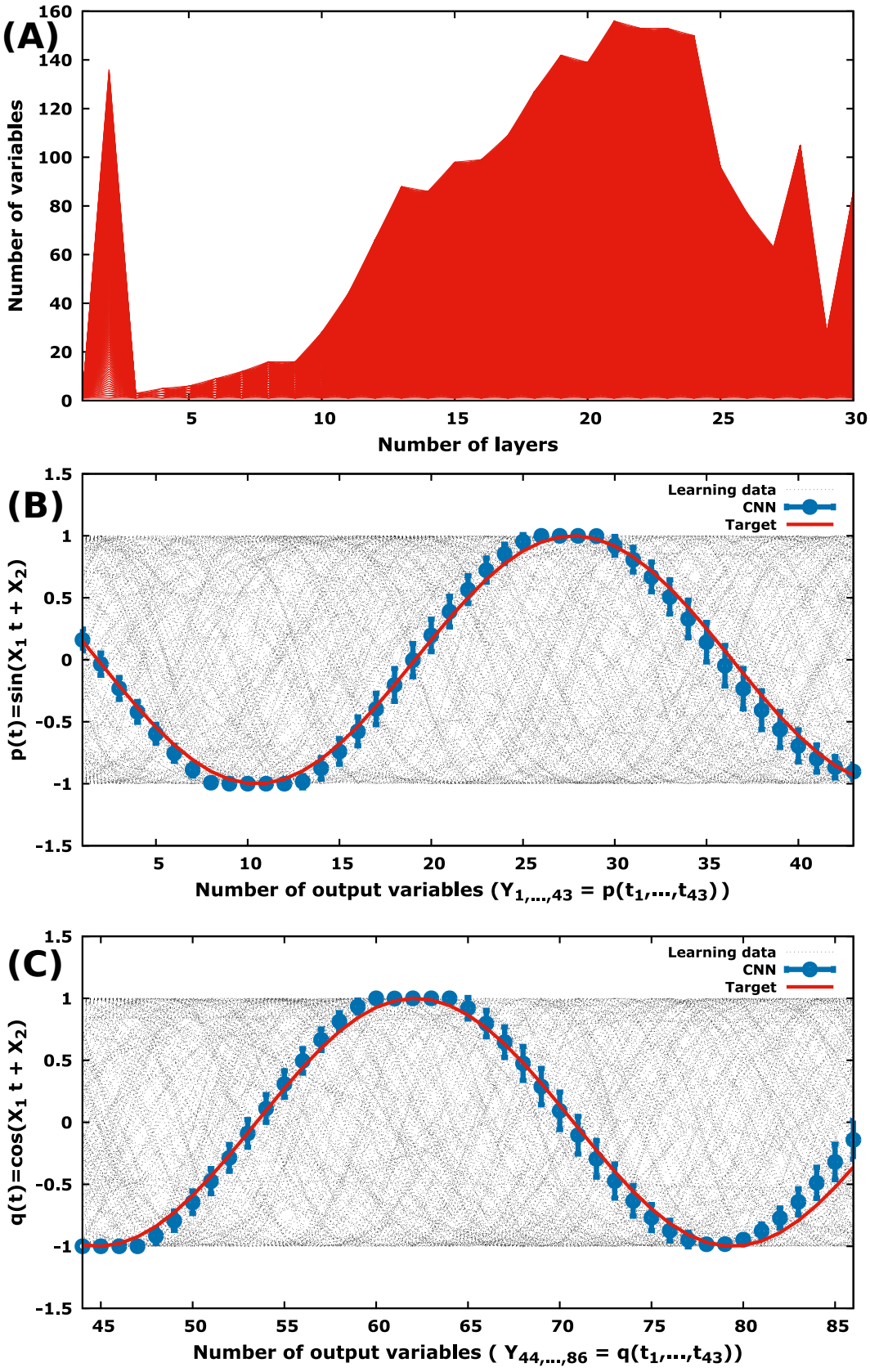

Figure 8. Learning the solution of an ordinary differential equation using a database of 200 available solutions with the classical CNN. A: the network determined for this problem consists of 28 locally fully connected hidden layers with up to 155 hidden variables, and B,C: the results of the target solution prediction using the classical CNN.

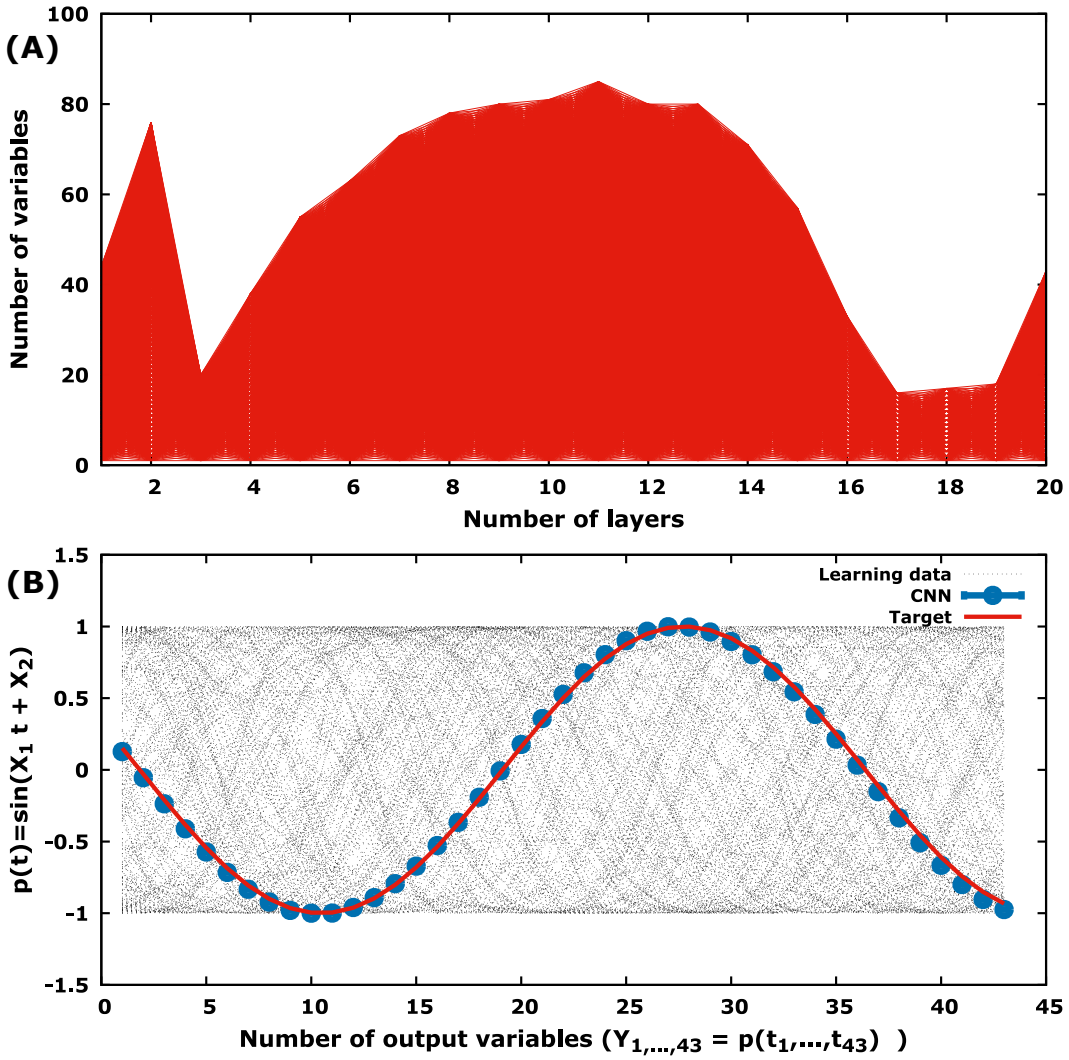

Figure 9. Learning the solution of an ordinary differential equation using a database of 200 available solutions with the implicit CNN. A: the network determined for this problem consists of 18 locally fully connected hidden layers with up to 84 hidden variables and B: the results of the target solution prediction using the implicit CNN.

6. APPLICATION OF THE IMPLICIT CNN TO 0D BLOOD FLOW MODEL

6.1. A test case with synthetic data

We are interested in the blood flow model ($\mathcal{M}$) presented in Section 2.2 and we would like to build a database allowing to estimate the blood pressure in some arteries of interest. In practice, for the hemodynamic applications we are interested in, some parameters are measurable and others not. In the case investigated here, blood flow rates in AA, R-ICA and L-ICA (q_{in}, q_{21} and q_{23}, respectively), and systolic and diastolic blood pressure in left and right brachial arteries (SBP and DBP in L.BRA and R.BRA, the maximum and the minimum value of P_{10} and P_{13}), were measured. This information led to choose both ICAs and both brachial arteries as the arteries of interest. To estimate the blood pressure in

these four arteries of interest, we first generate the database following the procedure described in Section 3. From this database, four different CNNs, one for each artery of interest and each based on $N = 100$ samples, were built following the implicit CNN structure for a single artery of interest described in Section 4.2. To build each CNN, a relationship is created between the input data of size 34 (2 for the two components $\tilde{\alpha}$ and $\tilde{\tau}$ of blood flow rates in AA, $\tilde{q}_{in}$, and 32 for the number of points in a discrete representation of the blood flow rates in the artery of interest) and the corresponding output data of size 32 representing the number of points in a discrete representation of blood pressure in the artery of interest. To be more precise, Figures 10,11,12, and 13, referred to as CNN R-ICA, CNN L-ICA, CNN R.BRA, and CNN L.BRA, respectively, show the implicit CNN structure for each of the four arteries of interest (Figures 10 and 11 for ICAs, and Figures 12 and 13 for brachial arteries). Thus, for each artery of interest, the input data size is $n = 34$, the output data size is $m = 32$, and the number of samples is $N = 100$.

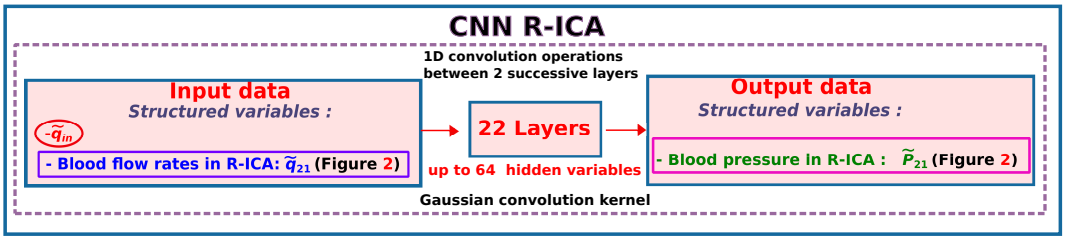

Figure 10. The implicit CNN for R-ICA. The network input data consists of a series of perturbated inlet flow rates parameters $\tilde{q}_{in}$ and of the corresponding series of blood flow rates in R-ICA ($\tilde{q}_{21}$ in Figure 2) and the network output data is composed of the corresponding series of blood pressure in R-ICA ($\tilde{P}_{21}$ in Figure 2).

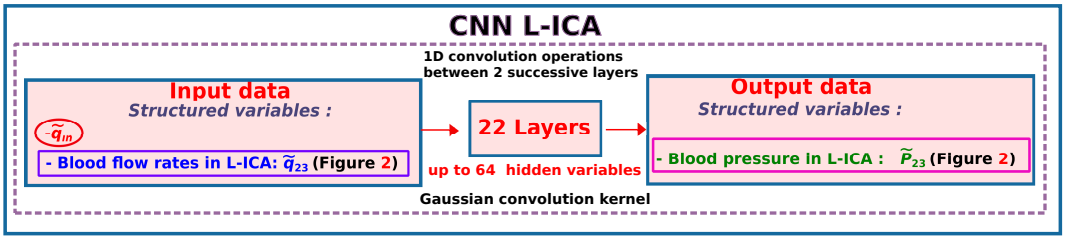

Figure 11. The implicit CNN for L-ICA. The network input data consists of a series of perturbated inlet flow rates parameters $\tilde{q}_{in}$ and of the corresponding series of blood flow rates in L-ICA ($\tilde{q}_{23}$ in Figure 2) and the network output data is composed of the corresponding series of blood pressure in L-ICA ($\tilde{P}_{23}$ in Figure 2).

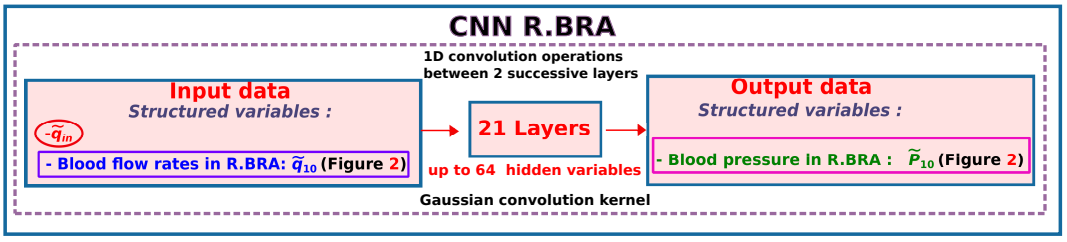

Figure 12. The implicit CNN for R.BRA. The network input data consists of a series of perturbated inlet flow rates parameters $\tilde{q}_{in}$ and of the corresponding series of blood flow rates in R.BRA ($\tilde{q}_{10}$ in Figure 2) and the network output data is composed of the corresponding series of blood pressure in R.BRA ($\tilde{P}_{10}$ in Figure 2).

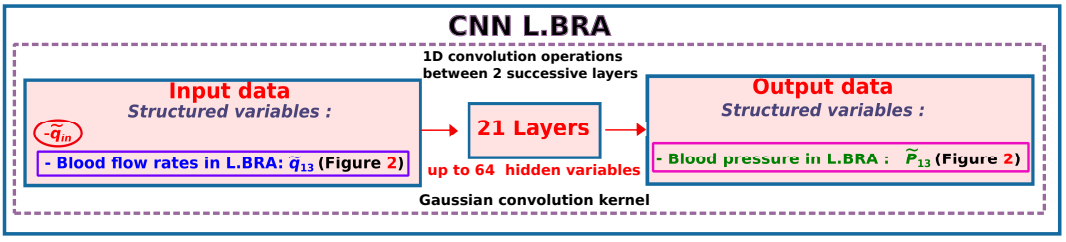

Figure 13. The implicit CNN for L.BRA. The network input data consists of a series of perturbated inlet flow rates parameters $\tilde{q}_{in}$ and of the corresponding series of blood flow rates in L.BRA ($\tilde{q}_{13}$ in Figure 2) and the network output data is composed of the corresponding series of blood pressure in L.BRA ($\tilde{P}_{13}$ in Figure 2).

Each of the four implicit CNNs built as in Figures 10-13 is trained using 99 samples whose the output data is referred to as Learning data in Figure 14. The last sample whose the output data is referred to as Target (the desired outcome) is used as testing sample. The results obtained are shown in Figure 14 which illustrates the networks used for R-ICA (Figure 14A) and for the left brachial artery (Figure 14B). Prediction of the desired blood pressure is referred to as CNN (Figure 14C, E for ICAs, and Figure 14D, F for brachial arteries). It is worth pointing out that for the two networks illustrated here, about twenty locally fully connected layers with up to 64 hidden variables have been effective in providing good results, and that using CNN with the same number of layers and hidden variables, similar results were observed for L-ICA (Figure 14E) and R.BRA (Figure 14F). From Figure 14, the predicted blood pressure recovers quite well the target blood pressure. Indeed, in all cases, the relative error between CNN prediction and the target values is less than 4%. Moreover, for the ICAs, the relative error for systolic and diastolic blood pressure is less than 1%. As for the brachial arteries, they have a relative error of less than 2% as well for systolic blood pressure as diastolic one.

To assess the relevance of these results and evaluate the skill and the performance of the four implicit CNNs on new data, a traditional approach is to undertake k-fold cross-validation. This approach consists first in splitting the $N = 100$ samples of the database into k roughly equal-sized subsamples [56]. Then, one of the k subsamples is retained as testing set, and the remaining subsamples are used as learning data. This process is then repeated k times with each of the k subsample being used for learning. Thus, all the subsamples are used for both learning and testing and each subsample is used for testing exactly once. For the choice of k, even if the choice k=10 is very common in the field of applied ML, and is recommend if one is struggling to choose a value for the dataset, there is no formal rule [16, 57, 58]. However, as pointed out by Kuhn, Gareth, and many others before them, in practice, one perfoms k-fold cross-validation using k=5 or k=10 [16, 58–60]. In fact, Both values of k have been shown empirically to yield test error rate estimates that suffer neither from excessively high bias nor from very high variance [16]. Given these considerations, k-fold cross-validation technique is performed here using k=5, 10, and 100. The case k=100, also known as leave-one-out cross-validation (LOOCV), is also taken into account

since this case generalizes the results presented above where 99 samples make up the training samples and only one sample is used as testing sample. Still, for each k-fold cross-validation run, the MAE obtained are averaged to produce a single estimation, which we refer to as the k-fold mean absolute errors (k-MAE). Table II shows typical k-MAE values obtained for blood pressure estimation using the implicit CNN in R-ICA and L.BRA and the corresponding relative errors with respect to the Target diastolic and systolic blood pressures (DBP and SBP). Just for the record, these Target values (DBP and SBP, in mmHg) were 110.67 / 161.49 for R-ICA, and 117.59 / 172.73 for L.BRA. The results in Table II shows that the k-MAE values decrease with the increase in the size of the training dataset meaning that more learning data leads to fewer errors in the prediction. This is consistent with the results in ML literature. Moreover, the relative errors with respect to the Target values for the case k=100 (less than 2%) seems to be comparable with the results obtained above without cross-validation using 99 samples for learning and a single sample for testing. These results being quite encouraging, the next step will focus on applying the implicit CNN to patient-specific data.

Table II. The k-fold mean absolute errors, k-MAE for k=5, 10, 100 and the corresponding relative errors with respect to the Target systolic and diastolic blood pressures (SBP and DBP) values in R-ICA and L.BRA. This shows typical skill scores of the implicit CNN for in-vitro pressure prediction in R-ICA and L.BRA.

Arteries of interest	**R-ICA**, mmHg			**L.BRA**, mmHg		
	k-MAE	**Relative errors**		**k-MAE**	**Relative errors**	
		DBP	**SBP**		**DBP**	**SBP**
k=5	**7.36**	**6.65%**	**4.56%**	**8.58**	**7.3%**	**4.97%**
k=10	**5.47**	**4.94%**	**3.38%**	**4.81**	**4.09%**	**2.78%**
k=100	**2.03**	**1.83%**	**1.26%**	**2.17**	**1.84%**	**1.26%**

6.2. A test case using patient-specific PC-MRA& MRI-based data.

The patient-specific data used in the current study has been provided by the Department of Neuroradiology at the *Centre Hospitalier Régional Universitaire de Montpellier (CHRU), Montpellier, France.* Arterial systolic and diastolic blood pressures at rest of a healthy volunteer were measured before and after image acquisition using a brachial automatic sphygmomanometer (Maglife, Schiller Medical). The systolic and diastolic values were 125 mmHg and 72 mmHg in the right brachial artery, and 115 mmHg and 72 mmHg in the left brachial artery. 2D phase-contrast imaging was performed on a Siemens 3T Skyra MR Scanner. The ascending aorta and the internal carotid arteries (right and left) 2D phase-contrast images were considered for the patient-specific blood flow rates extraction. For more details, the reader is referred to Lal et al [38].

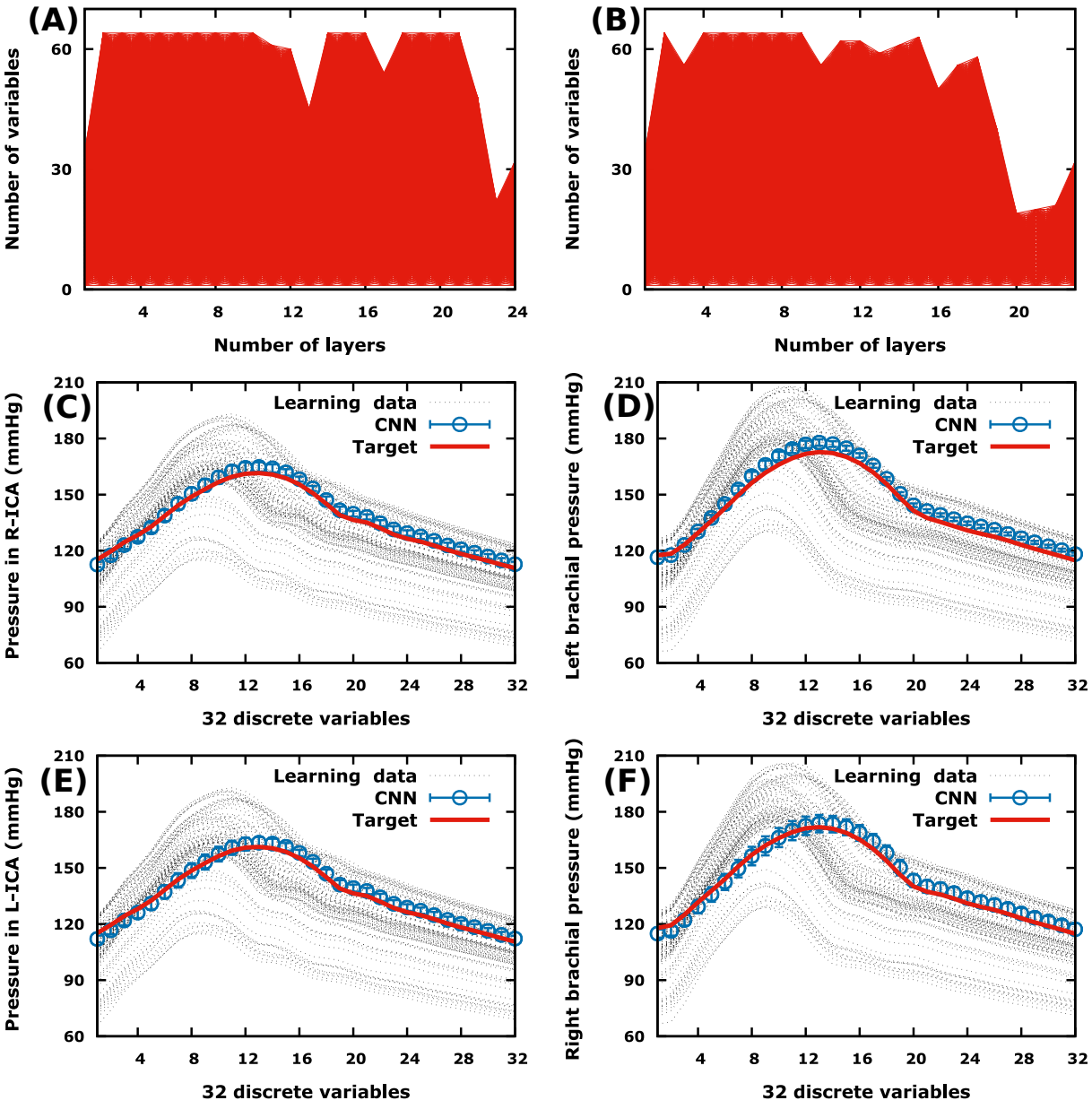

Figure 14. Typical networks determined once for all for estimating blood pressure in R-ICA (A) and in left brachial artery (B), estimated blood pressure in the four arteries of interest (in R-ICA (C), in left brachial artery (D), in L-ICA (E), and in right brachial artery (F)) using four different implicit CNNs ($n = 34, m = 32$, and $N = 100$). For R-ICA and L-ICA, the network determined has 22 hidden layers with up to 64 hidden variables, and for the brachial arteries, there are 21 hidden layers.

For each artery of interest, the implicit CNN built for the synthetic case (Section 6.1) is reused with patient-specific data. More precisely, the implicit CNN algorithm applies the acquired knowledge during learning in the synthetic case namely, the relationship between input and output data in each CNN previously built in the synthetic case (the network and its coefficients obtained during the learning from simulations are kept unchanged) to patient-specific data. In fact, the testing samples in the synthetic case are replaced with patient-specific data from the different arteries of interest. Thus, each testing sample consists now of the 2 variables (α and τ) modeling the patient-specific inlet flow rate, and the 32 points

in a discrete representation of the patient-specific blood flow rates in the artery of interest. As a reminder, R-ICA, L-ICA, R.BRA, and L.BRA are here the arteries of interest. For R-ICA and L-ICA, the implicit CNNs obtained in the synthetic case are applied to the patient-specific blood flow extracted from MRA & MRI in the ICAs, which is used as input data. The resulting output, the blood pressure estimated via CNN in the ICAs is then compared to the desired outcome, the blood pressure previously estimated by Lal et al [38] in the same arteries but through the solution of an inverse problem using EnKF.

In the same way, for R.BRA and L.BRA, the implicit CNNs built in the synthetic case are applied to patient-specific blood flow but this time not to those extracted from MRA & MRI (measurements not available in these arteries) but to those previously predicted in the brachial arteries by Lal et al [38] with the blood flow model ($\mathcal{M}$) using EnKF. Again, in this case, the resulting outcome, the blood pressure predicted in the brachial arteries using CNN, is compared to that previoulsy predicted by Lal et al [38] in the same arteries through the solution of an inverse problem using EnKF. Comparison with the brachial pressure measurements is also performed to test the accuracy of the method used.

Figure 15 shows a comparative overview between typical results of blood pressure predicted via CNN (referred to as CNN) and typical results of blood pressure previously predicted via EnKF (referred to as EnKF). It is observed that secondary peaks in the blood pressure waveforms are well reproduced with CNN. The same trend was observed for L-ICA and R.BRA. In all cases, for the ICAs, the CNN simulated waveforms have a relative error of less than 3% in the diastolic and systolic blood pressure. Tables III and IV summarize the cardiac cycle systolic and diastolic blood pressure estimated using CNN and that simulated using EnKF, in the four arteries of interest (Table III, for the ICAs and Table IV, for the brachial arteries). From these Tables, all systolic and diastolic blood pressure predicted using EnKF and CNN results differ by less than 7%. In addition, in order to give an idea of the prediction error made by CNN with respect to available blood pressure measurements, the predicted systolic and diastolic pressure values in the brachial arteries are compared with patient-specific blood pressure measurements. As reported in Table IV, the relative error obtained using CNN is less than 3% for diastolic pressure and in the two brachial arteries, less than 6% for systolic pressure in R.BRA but up to 16 % for systolic pressure in L.BRA. One explanation for this larger error is that the input/output testing sample for this case is already from previous numerical results — the results perfomed by Lal et al [38] — and not from patient-specific data extracted from MRA & MRI. Moreover, note that there is a significant difference between the systolic blood pressure value previously predicted by Lal et al [38] using EnKF in L.BRA (129.32 mmHg) — what needs to be predicted here — and the systolic pressure measurement value (115 mmHg). Apart from this extreme case, the agreement between the predicted and measured pressure values is rather good. This shows that CNN can effectively be a good alternative to the previous method proposed by Lal et al [38] for blood pressure estimation through the solution of an inverse problem

using EnKF. This is interesting in terms of cost and computation time. Indeed, learning takes place in 2 minutes on a 3GHz CPU and a single inference in 0.05 second. The learning time includes the identification of the suitable network structure. This inference time should be compared to the 7 hours needed by the ensemble based inversion using EnKF. In other words, if the unit time is the computation time for pressure estimation by CNN, EnKF requires 240000 times more computation time. A reason for this extra cost in EnkF is that the EnKF-based technique needs to retrieve first the mechanical properties of arteries before providing the arterial pressures while machine learning does not need any identification of the mechanical properties of the arteries in order to retrieve the arterial pressures.

Table III. Validation: comparison of the cardiac cycle systolic blood pressure (SBP) and diastolic blood pressure (DBP) estimated using CNN with the model simulated values using EnKF in the ICAs. The percentage difference between the results obtained using CNN and those based on EnKF is stated in bold in parenthesis.

ICA	Systolic blood pressure (SBP), mmHg		Diastolic blood pressure (DBP), mmHg	
	CNN	EnKF	CNN	EnKF
Right	114.80 (±0.00) **(0.46%)**	114.28	69.23 (±0.04) **(2.32%)**	67.656
Left	114.54 (±0.00) **(0.46%)**	114.02	69.30 (±0.04) **(2.49%)**	67.617

Table IV. Validation: comparison of the cardiac cycle systolic blood pressure (SBP) and diastolic blood pressure (DBP) estimated using CNN with the model simulated values using EnKF in the brachial arteries. The predicted pressure values using CNN are also compared to patient-specific brachial pressure measurements. The percentage difference between the results obtained by CNN and those based on EnKF, and the percentage difference between the results obtained by CNN and the blood pressure measurements are stated in bold in parenthesis.

BRA	Systolic blood pressure (SBP), mmHg			Diastolic blood pressure (DBP), mmHg		
	CNN	measurement	EnKF	CNN	measurement	EnKF
Right	131.92 (±0.59) **(3.55%)**	125 **(5.53%)**	127.4	73.54 (±0.18) **(6.57%)**	72 **(2.14%)**	69
Left	133.71 (±0.60) **(3.39%)**	115 **(16.27%)**	129.32	73.21 (±0.17) **(6.63%)**	72 **(1.68%)**	68.66

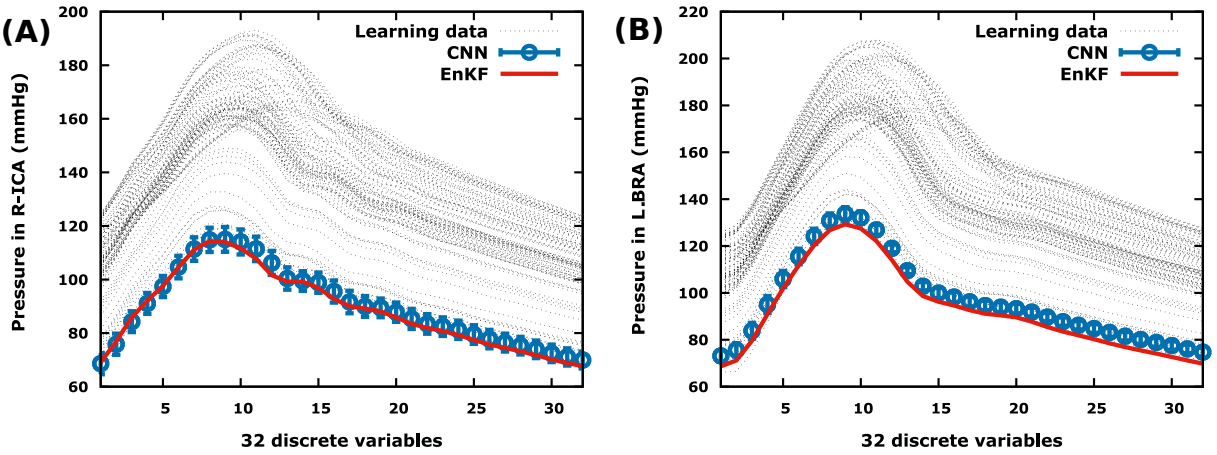

Figure 15. Prediction of the solution of a cardiovascular problem (patient-specific blood pressure) in R-ICA (A) and in L.BRA (B) using the network built during training in the synthetic case, in Section 6.1. The 99 available simulated solutions (blood pressure) are referred to as Learning data, the predicted pressure as CNN, and the pressure previously estimated using EnKF [38], as EnKF.

7. DISCUSSION AND LIMITATIONS

The objective of the present investigation is to use machine learning in order to propose an alternative to an ensemble based inversion procedure (EnKF) for cerebral blood pressure estimation. The presented implicit manifold learning technique is found to be much faster than EnKF yet providing very similar results in terms of blood pressure signals. The in vitro/in vivo results show good agreement between the target blood pressures and the outcome of the machine learning procedure. In particular, the previously predicted SBP and DBP in the four arteries of interest using EnKF are well recovered with machine learning procedure.

One of the limitations could be the use of a simple cardiovascular model comprising of 0D components and the modelling of inflow q_{in} using Equation 4. Indeed, several studies have pointed out the shortcomings of lumped models and of the use of this approximation of inflow and have proposed more complex/accurate models to better capture the characteristics of blood flow. So we are aware that more sophisticated physical models can be considered. However, to be able to make a fair comparison, we seek a supervised learning approach encapsulating the same physics as the one used in our previous studies [38, 39] with the objective to get the cerebral blood pressure in nearly real time. Again, this is why the same cardiovascular model and the arterial network have been considered as with the EnKF procedure. Of course this procedure should receive more testing in a clinical environment.

One particularity of this work is that unlike most learning techniques, the implict learning proposed here is operational with small data. This should be seen as a constraint and not something we fully control. Data might come from other teams and we do not always have control of it. Moreover, the only thing we can be sure of, regardless of the AI algorithm one uses, is that the quality of a database

decreases with its size because it simply becomes more difficult to guarantee the quality of the data when the size of the sample is becoming large. Data quality issue can also be related to possible inconsistency between scenarios (e.g. low and high fidelity prediction of a same situation). This is a classical problem with supervised learning. Therefore, we think that working with small database is desirable when this is possible. Furthermore, being able to produce meaningful results with small database will permit to operate in cases where the learning is carried out not with synthetic data but with real experiments data, which is obviously more difficult to multiply.

Another particularity of this work is that, in order to make learning and inference more efficient, learning has been performed on a manifold, using only a subset of the parameters in the database. One natural question is then how this kind of restriction can affect the results. To address this question, ablation analysis is performed in the real data case. An ablation study refers to removing some feature of a model in order to see how that affects the outcomes. As a reminder, the implicit CNN relates only the inflow q_{in} and blood flow rate in any artery of interest to blood pressure in the same artery. But, the available information from MRA & MRI images are inflow q_{in} and blood flow rates in R-ICA & L-ICA. So, when the implicit CNN is performed for predicting blood pressure in R-ICA, the blood flow rate in L-ICA is systematically ablated and vice-versa. Now all flow rates information available from MRA & MRI images (namely, inflow q_{in} and blood flow rates in R-ICA & L-ICA) and blood flow rates previously estimated by Lal et al [38] in R.BRA and L.BRA associated with inflow q_{in} are used as CNN input data, respectively. This permits to see how the networks output is affected by the addition of this extra information. The results presented in Figure 16 show that the construction is quite robust with respect to such ablation.

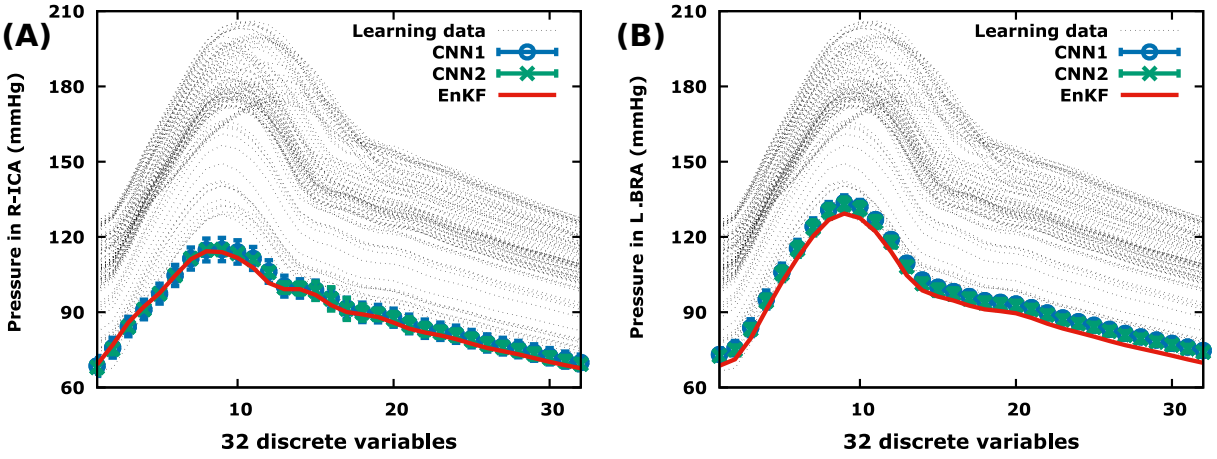

Figure 16. Ablation analysis for the patient-specific case: CNN1 and CNN2 results are practically the same. CNN1 represents the predicted blood pressure using parts of the input features (inflow q_{in} and blood flow rate in a particular artery of interest, in R-ICA (A) and in L.BRA (B) here). This corresponds exactly to what is referred to CNN in Figure 15. CNN2 is the predicted blood pressure using together inflow q_{in} and blood flow rates in R-ICA & L-ICA as input data (A) or using at the same time inflow q_{in} and blood flow rates previously estimated by Lal et al [38] in R.BRA and L.BRA as input data (B).

8. CONCLUDING REMARKS

The main challenge in this paper has been to use machine learning for blood pressure prediction in cerebral arteries. We aim our proposal to be faster and more practical than when the pressure is recovered through the solution of an inverse problem using ensemble methods such as EnKF, yet as accurate. Hence, machine learning has been applied to the solution of cardiovascular problems together with a transfer learning strategy using in-silico databases. Indeed, as patient data is rare, a mathematical model has been used to simulate the relationship between blood flow rates and blood pressure in an human arterial network. The machine learning tool has been applied to patient-specific blood pressure estimation to provide a direct link in any couple of arterial blood flow rates and pressure in the upper body region. The proposed method demonstrated compelling results on synthetic and patient-specific data. The results suggest that the proposed ML strategy is well-suited for the considered cardiovascular problem providing real-time estimation of the pressure in patient-specific arteries, which is missing in standard MRA & MRI acquisition. In comparison with the EnKF-based pressure estimator, the suggested ML-based estimator provides a substantial gain as regards calculation time. In that sense, we can consider inference by this machine learning approach to provide pressure distribution in real time augmenting the MRA-MRI information on the fly.

ACKNOWLEDGMENT

The authors greatly thank Prof.-Dr. V. Costalat, Dr. E. Le Bars and Dr. J. Deverdun from the Department of Neuroradiology of the *Centre Hospitalier Régional Universitaire de Montpellier, Gui de Chauliac, Montpellier, France* and Dr. J. Siguenza for providing the medical data and for helpful discussions. The blood flow model ($\mathcal{M}$) used for the generation of the database has been solved using the implicit numerical integration solver `DVODE` [61, 62] available on `http://www.radford.edu/~thompson/vodef90web/`.

REFERENCES

1. Deo" RC. "machine learning in medicine". *"Circulation"* 2015; **"132"**("20"):"1920–1930, doi: 10.1161/CIRCULATIONAHA.115.001593. URL https://www.ahajournals.org/doi/abs/10.1161/CIRCULATIONAHA.115.001593.

2. Liang L, Kong F, Martin C, Pham T, Wang Q, Duncan J, Sun W. Machine learning–based 3-d geometry reconstruction and modeling of aortic valve deformation using 3-d computed tomography images. *International Journal for Numerical Methods in Biomedical Engineering* 2017; **33**(5):e2827, doi:10.1002/cnm.2827. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.2827, e2827 CNM-Jan-16-0019.R1.

3. Adam NR, Wieder R, Ghosh D. Data science, learning, and applications to biomedical and health sciences. *Annals of the New York Academy of Sciences* 2017; **1387**(1):5–11, doi:10.1111/nyas.13309. URL https://nyaspubs.onlinelibrary.wiley.com/doi/abs/10.1111/nyas.13309.

4. Pozin N, Montesantos S, Katz I, Pichelin M, Vignon-Clementel I, Grandmont C. Predicted airway obstruction distribution based on dynamical lung ventilation data: A coupled modeling-machine learning methodology. *International Journal for Numerical Methods in Biomedical Engineering* 2018; **34**(9):e3108, doi:10.1002/cnm.3108. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3108, e3108 CNM-Jul-17-0180.R2.

5. Koeppl T, Santin G, Haasdonk B, Helmig R. Numerical modelling of a peripheral arterial stenosis using dimensionally reduced models and kernel methods. *International Journal for Numerical Methods in Biomedical Engineering* 2018; **34**(8):e3095, doi:10.1002/cnm.3095. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3095, e3095 cnm.3095.

6. Koprowski R, Foster KR. Machine learning and medicine: book review and commentary. *BioMedical Engineering OnLine* Feb 2018; **17**(1):17, doi:10.1186/s12938-018-0449-9. URL https://doi.org/10.1186/s12938-018-0449-9.

7. Cang Z, Wei GW. Integration of element specific persistent homology and machine learning for protein-ligand binding affinity prediction. *International Journal for Numerical Methods in Biomedical Engineering* 2018; **34**(2):e2914, doi:10.1002/cnm.2914. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.2914, e2914 cnm.2914.

8. Luo Y, Fan Z, Baek S, Lu J. Machine learning–aided exploration of relationship between strength and elastic properties in ascending thoracic aneurysm. *International Journal for Numerical*

Methods in Biomedical Engineering 2018; **34**(6):e2977, doi:10.1002/cnm.2977. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.2977, e2977 cnm.2977.

9. Cilla M, Pérez-Rey I, Martínez MA, Peña E, Martínez J. On the use of machine learning techniques for the mechanical characterization of soft biological tissues. *International Journal for Numerical Methods in Biomedical Engineering* 2018; **34**(10):e3121, doi:10.1002/cnm.3121. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3121, e3121 cnm.3121.

10. Goceri E. Diagnosis of alzheimer's disease with sobolev gradient-based optimization and 3d convolutional neural network. *International Journal for Numerical Methods in Biomedical Engineering* 2019; **35**(7):e3225, doi:10.1002/cnm.3225. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3225, e3225 CNM-Feb-19-0064.R1.

11. Zhang B, Dai N, Tian S, Yuan F, Yu Q. The extraction method of tooth preparation margin line based on s-octree cnn. *International Journal for Numerical Methods in Biomedical Engineering* 2019; **35**(10):e3241, doi:10.1002/cnm.3241. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3241, e3241 CNM-Feb-19-0059.R1.

12. Chen J, Huang H, Hao W, Xu J. A machine learning method correlating pulse pressure wave data with pregnancy. *International Journal for Numerical Methods in Biomedical Engineering* 2020; **36**(1):e3272, doi:10.1002/cnm.3272. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3272, e3272 cnm.3272.

13. Nguyen DD, Wei GW. Dg-gl: Differential geometry-based geometric learning of molecular datasets. *International Journal for Numerical Methods in Biomedical Engineering* 2019; **35**(3):e3179, doi:10.1002/cnm.3179. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3179, e3179 cnm.3179.

14. Henglin M, Stein G, Hushcha PV, Snoek J, Wiltschko AB, Cheng S. Machine learning approaches in cardiovascular imaging. *Circulation: Cardiovascular Imaging* 2017; **10**(10):e005 614, doi:10.1161/CIRCIMAGING.117.005614. URL https://www.ahajournals.org/doi/abs/10.1161/CIRCIMAGING.117.005614.

15. Handelman GS, Kok HK, Chandra RV, Razavi AH, Lee MJ, Asadi H. edoctor: machine learning and the future of medicine. *Journal of Internal Medicine* 2018; **284**(6):603–619, doi:10.1111/joim.12822. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/joim.12822.

16. James G, Witten D, Hastie T, Tibshirani R. *An introduction to statistical learning. With applications in R.*, vol. 103. New York, NY: Springer, 2013.

17. Başţanlar Y, Özuysal M. *Introduction to Machine Learning*, chap. 7. Humana Press: Totowa, NJ, 2014; 105–128, doi:10.1007/978-1-62703-748-8_7. URL https://doi.org/10.1007/978-1-62703-748-8_7.

18. Cruz JA, Wishart DS. Applications of machine learning in cancer prediction and prognosis. *Cancer Informatics* 2006; **2**:59–77, doi:10.1177/117693510600200030. URL https://doi.org/10.1177/117693510600200030.

19. Weiss JC, Natarajan S, Peissig PL, McCarty CA, Page D. Machine learning for personalized medicine: Predicting primary myocardial infarction from electronic health records. *AI Magazine* 12 2012; **33**(4):33–45, doi:https://doi.org/10.1609/aimag.v33i4.2438. URL https://aaai.org/ojs/index.php/aimagazine/article/view/2438.

20. Weiss JC, Kuusisto F, Boyd K, Liu J, Page D. Machine learning for treatment assignment: improving individualized risk attribution. *American Medical Informatics Association (AMIA) Annual Symposium* 11 2015; **2015**:1306–1315. eCollection 2015.

21. Kourou K, Exarchos TP, Exarchos KP, Karamouzis MV, Fotiadis DI. Machine learning applications in cancer prognosis and prediction. *Computational and Structural Biotechnology Journal* 2015; **13**:8 – 17, doi:https://doi.org/10.1016/j.csbj.2014.11.005. URL http://www.sciencedirect.com/science/article/pii/S2001037014000464.

22. Goldstein BA, Navar AM, Carter RE. Moving beyond regression techniques in cardiovascular risk prediction: applying machine learning to address analytic challenges. *European Heart Journal* 07 2016; **38**(23):1805–1814, doi:10.1093/eurheartj/ehw302. URL https://dx.doi.org/10.1093/eurheartj/ehw302.

23. Pan SJ, Yang Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* Oct 2010; **22**(10):1345–1359, doi:10.1109/TKDE.2009.191.

24. Ruder S. (2017, march 2017). transfer learning - machine learning's next frontier 2017; URL https://ruder.io/transfer-learning/index.html/, retrieved February 1, 2019, from https://ruder.io/transfer-learning/index.html.

25. Fung GPC, Yu JX, Lu H, Yu PS. Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering* Jan 2006; **18**(1):6–20, doi:10.1109/TKDE.2006.16.

26. Al-Mubaid H, Umair SA. A new text categorization technique using distributional clustering and learning logic. *IEEE Transactions on Knowledge and Data Engineering* Sep 2006; **18**(9):1156–1165, doi:10.1109/TKDE.2006.135.

27. Dai W, Xue GR, Yang Q, Yu Y. Co-clustering based classification for out-of-domain documents. *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, ACM: New York, NY, USA, 2007; 210–219, doi:10.1145/1281192.1281218. URL http://doi.acm.org/10.1145/1281192.1281218.

28. Sarinnapakorn K, Kubat M. Combining subclassifiers in text categorization: A dst-based solution and a case study. *IEEE Transactions on Knowledge and Data Engineering* Dec 2007; **19**(12):1638–1651, doi:10.1109/TKDE.2007.190663.

29. Sharma M, Holmes M, Santamaria J, Irani A, Isbell C, Ram A. Transfer learning in real-time strategy games using hybrid cbr/rl. *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 2007; 1041–1046. URL http://dl.acm.org/citation.cfm?id=1625275.1625444.

30. Raina R, Ng AY, Koller D. Constructing informative priors using transfer learning. *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, ACM: New York, NY, USA, 2006; 713–720, doi:10.1145/1143844.1143934. URL http://doi.acm.org/10.1145/1143844.1143934.

31. Blitzer J, McDonald R, Pereira F. Domain adaptation with structural correspondence learning. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, Association for Computational Linguistics: Stroudsburg, PA, USA, 2006; 120–128. URL http://dl.acm.org/citation.cfm?id=1610075.1610094.

32. Pan S, Wenchen Zheng V, Yang Q, Hao Hu D. Transfer learning for wifi-based indoor localization. *AAAI Workshop - Technical Report* 01 2008; .

33. Taylor ME, Stone P. Cross-domain transfer for reinforcement learning. *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, ACM: New York, NY, USA, 2007; 879–886, doi:10.1145/1273496.1273607. URL http://doi.acm.org/10.1145/1273496.1273607.

34. Ramon J, Driessens K, Croonenborghs T. Transfer learning in reinforcement learning problems through partial policy recycling. *Proceedings of the 18th European Conference on Machine Learning*, ECML '07, Springer-Verlag: Berlin, Heidelberg, 2007; 699–707, doi:10.1007/

978-3-540-74958-5_70. URL http://dx.doi.org/10.1007/978-3-540-74958-5_70.

35. Taylor ME, Stone P. Transfer learning for reinforcement learning domains: A survey. *J. Mach. Learn. Res.* Dec 2009; **10**:1633–1685. URL http://dl.acm.org/citation.cfm?id=1577069.1755839.

36. Lazaric A. *Transfer in Reinforcement Learning: A Framework and a Survey*, chap. 5. Springer Berlin Heidelberg: Berlin, Heidelberg, 2012; 143–173, doi:10.1007/978-3-642-27645-3_5. URL https://doi.org/10.1007/978-3-642-27645-3_5.

37. Zhan Y, Taylor ME. Online transfer learning in reinforcement learning domains. *CoRR* 2015; **abs/1507.00436**. URL http://arxiv.org/abs/1507.00436.

38. Lal R, Nicoud F, Bars E, Deverdun J, Molino F, Costalat V, Mohammadi B. Non invasive blood flow features estimation in cerebral arteries from uncertain medical data. *Annals of Biomedical Engineering* 08 2017; **45**:1–18, doi:10.1007/s10439-017-1904-7.

39. Rapadamnaba R, Nicoud F, Mohammadi B. Backward sensitivity analysis and reduced-order covariance estimation in noninvasive parameter identification for cerebral arteries. *International Journal for Numerical Methods in Biomedical Engineering* 2019; **35**(4):e3170, doi:10.1002/cnm.3170. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cnm.3170, e3170 cnm.3170.

40. Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* Nov 1998; **86**(11):2278–2324, doi:10.1109/5.726791.

41. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature* 2015; **521**:436–444, doi:10.1038/nature14539. URL https://doi.org/10.1038/nature14539.

42. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press: Cambridge, MA (USA), 2016. URL http://www.deeplearningbook.org.

43. Dumoulin V, Visin F. A guide to convolution arithmetic for deep learning. *arXiv e-prints* Mar 2016; :arXiv:1603.07285URL https://ui.adsabs.harvard.edu/#abs/2016arXiv160307285D, provided by the SAO/NASA Astrophysics Data System.

44. Olufsen MS, Nadim A, *et al.*. On deriving lumped models for blood flow and pressure in the systemic arteries. *Math Biosci Eng* 2004; **1**(1):61–80.

45. Gul R. Mathematical modeling and sensitivity analysis of lumped-parameter model of the human cardiovascular system. *PhD Thesis in Freien Universität Berlin* 01 2016; URL https://www.researchgate.net/publication/301852867_Mathematical_Modeling_and_Sensitivity_Analysis_of_Lumped-Parameter_Model_of_the_Human_Cardiovascular_System.

46. Reymond P, Merenda F, Perren F, Rüfenacht D, Stergiopulos N. Validation of a one-dimensional model of the systemic arterial tree. *American Journal of Physiology-Heart and Circulatory Physiology* 2009; **297**(1):H208–H222.

47. Boileau E, Nithiarasu P, Blanco PJ, Müller LO, Fossan FE, Hellevik LR, Donders WP, Huberts W, Willemet M, Alastruey J. A benchmark study of numerical schemes for one-dimensional arterial blood flow modelling. *International journal for numerical methods in biomedical engineering* 2015; **31**(10):1–33.

48. Olufsen MS. Structured tree outflow condition for blood flow in larger systemic arteries. *American journal of physiology-Heart and circulatory physiology* 1999; **276**(1):H257–H268.

49. Williams N, Zander S, Armitage G. A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. *SIGCOMM Comput. Commun. Rev.* Oct 2006; **36**(5):5–16, doi:10.1145/1163593.1163596. URL http://doi.acm.org/10.1145/1163593.1163596.

50. Wu X, Kumar V, Ross Quinlan J, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Yu PS, *et al.*. Top 10 algorithms in data mining. *Knowledge and Information Systems* Jan 2008; **14**(1):1–37, doi:10.1007/s10115-007-0114-2. URL https://doi.org/10.1007/s10115-007-0114-2.

51. Tsai CF, Hsu YF, Lin CY, Lin WY. Intrusion detection by machine learning: A review. *Expert Systems with Applications* 2009; **36**(10):11 994 – 12 000, doi:https://doi.org/10.1016/j.eswa.2009.05.029. URL http://www.sciencedirect.com/science/article/pii/S0957417409004801.

52. Ayodele TO. Types of machine learning algorithms. *New Advances in Machine Learning*, Zhang Y (ed.). chap. 3, IntechOpen: Rijeka, 2010, doi:10.5772/9385. URL https://doi.org/10.5772/9385.

53. Valletta JJ, Torney C, Kings M, Thornton A, Madden J. Applications of machine learning in animal behaviour studies. *Animal Behaviour* 2017; **124**:203 – 220, doi:https://doi.org/10.1016/

j.anbehav.2016.12.005. URL http://www.sciencedirect.com/science/article/pii/S0003347216303360.

54. Shameer K, Johnson KW, Glicksberg BS, Dudley JT, Sengupta PP. Machine learning in cardiovascular medicine: are we there yet? *Heart* 01 2018; **104**(14):1156–1164, doi:10.1136/heartjnl-2017-311198. URL https://heart.bmj.com/content/104/14/1156.

55. Draper NR, Smith H. *Applied Regression Analysis*. John Wiley & Sons, Ltd: New York (USA), 2014, doi:10.1002/9781118625590. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118625590.

56. Hastie T, Tibshirani R, Firedman J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York, 2017. URL https://web.stanford.edu/~hastie/ElemStatLearn/printings/ESLII_print12.pdf.

57. Mclachlan G, Do KA, Ambroise C. *Discriminant Analysis*, chap. 6. John Wiley & Sons, Ltd: Hoboken, New Jersey, 2005; 185–220, doi:10.1002/047172842X.ch6. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/047172842X.ch6.

58. Kuhn M, Johnson K. *Applied Predictive Modeling*. SpringerLink : Bücher, Springer New York, 2013, doi:10.1007/978-1-4614-6849-3. URL https://books.google.fr/books?id=xYRDAAAAQBAJ.

59. Breiman L, Spector P. Submodel selection and evaluation in regression. the x-random case. *International Statistical Review / Revue Internationale de Statistique* 1992; **60**(3):291–319. URL http://www.jstor.org/stable/1403680.

60. Kohavi R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1995; 1137–1143. URL http://dl.acm.org/citation.cfm?id=1643031.1643047.

61. Byrne GD, Hindmarsh AC. Stiff ode solvers: A review of current and coming attractions. *Journal of Computational physics* 1987; **70(1)**:1–62.

62. SP Corwin ST, White S. Solving odes and ddes with impulses. *JNAIAM J. Numer. Anal .Indust. Appl. Math* 2008; **3**:139–149.