



Subject Areas:

Biomedical engineering, Computer modelling and simulation, Artificial intelligence

Keywords:

Parameter estimation, Finite element method, Surrogate model, Machine learning method, Inverse problem

Author for correspondence:

Lei Ren

e-mail: renlei1@mail.nwpu.edu.cn

Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium

Li Cai^{1,2,3}, Lei Ren^{1,2,3,*}, Yongheng Wang^{1,2,3}, Wenxian Xie^{1,2,3}, Guangyu Zhu⁴ and Hao Gao⁵

¹Xi'an Key Laboratory of Scientific Computation and Applied Statistics, ²NPU-UoG International Cooperative Lab for Computation and Application in Cardiology and ³School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an, 710129, China

⁴School of Energy and Power Engineering, Xi'an Jiaotong University, Xi'an, 710049, China

⁵School of Mathematics and Statistics, University of Glasgow, Glasgow, G12 8QQ, UK

A long-standing problem at the frontier of biomechanical studies and applied mathematics is to develop fast methods capable of estimating material properties from clinical data. In this paper, we studied some surrogate models based on machine learning (ML) methods for fast parameter estimation of LV myocardium. We used three ML methods named K-nearest neighbour (KNN), XGBoost and Multi-layer perceptron (MLP) to emulate the relationships between pressure, volume and strain during diastolic filling of the LV. Firstly, to train the surrogate models, a forward finite element simulator of LV diastolic filling was used with Holzapfel-Ogden constitutive model. Then, the training data were obtained by numerical simulation in a projected parameters space. Then, three ML models were trained to learn the relationships of pressure-volume and pressure-strain. Finally, the parameters estimation problem was solved by those trained surrogate models. We select R^2 as the evaluation criterion and mean square error (MSE) as loss function for hyper-parameter optimization. The results show ML methods could learn the relationships of pressure-volume and pressure-strain very well and take a minutes to inversely estimate unknown parameters for the XGBoost and the MLP, which is much faster than using a gradient-based approach using the forward simulator.

1. Introduction

Heart disease, such as myocardial infarction, has been seriously affecting the quality of life of human beings [1]. Early diagnosis and treatment can effectively reduce the incidence and the mortality of heart disease [2]. In recent years, with the development of medical image technology, such as magnetic resonance imaging (MRI), it has provided paramount data for describing the structure and function of the human heart non-invasively [3]. While imaging data alone does not tell the whole story of heart function, i.e. the stress and myocardial stiffness, etc. Mathematical modeling and numerical simulation of cardiac function, broadly termed as *sillico medicine*, have been considered as the next generation of medicine for deciphering the mechanism of heart function physiologically and pathologically [4,5].

Finite element method (FEM) has been successfully applied to cardiac modeling [6–9] in the past several decades. Myocardial material was generally considered to be a hyperelastic material with a strong nonlinear anisotropic stress response [10]. Many constitutive models have been used to describe myocardial material behaviors, including isotropic models, transversely isotropic models and, more recently, orthotropic models [11]. In particular, the Holzapfel-Ogden model is a structure-based orthotropic constitutive model that not only accurately describe the mechanical behaviour of the myocardium well from various experimental data [12], but also has been successfully applied to subject-specific cardiac models purely based on *in vivo* routine imaging data [13].

In general, material parameter estimation of a FEM heart model is formulated as an inverse problem [14–18], which essentially requires solving a constrained optimization problem [19,20] by minimizing the mismatch between limited measured data and the FEM model predictions through finding potential material parameters. This constrained optimization problems can be solved by some gradient-based optimization methods [21] generally, such as the Newton method, the conjugate gradient method, and some intelligent methods [22,23], i.e. the genetic method [24], the particle swarm method [25] etc. The gradient-based optimization methods generally are easy to implement and may converge quickly, but it cannot guarantee global optimization. On the other hand, although the evolutionary methods converge slowly especially for high-dimensional problems, it can theoretically guarantee global optimization. When the actual forward problem (high-accuracy model) is computationally intensive and difficult to solve, and further lead to significant computational demand for inferring myocardial parameter. Thus, to reduce the significant computational resources in the constrained optimization problem, surrogate models, which are statistical approximations of the forward problems have been developed for fast parameter inference. In general, for surrogate models, rapid solutions can be quickly computed instead of the original models which will speed up the optimization process dramatically, even in real time. Surrogate models has been widely used in certain engineering problems, like aerospace systems including aerodynamics, structures, and propulsion, among other disciplines [26,27].

The latest research shows that heart disease can cause some changes in myocardial properties [28], recently, the estimation of myocardial material parameters using image-based models has attracted intensive interests ranging from gradient-based inverse problems to using ML-based surrogate approaches for fast parameter inferences. By using ML models, the behaviours of the LV in response to changes in material properties, loads and boundary conditions etc. can be predicted in real time, and can be further applied to the design of medical instruments and monitoring heart condition. For instance, Liang et al. [29] has been first put forward the deep learning technique, a multilayer neural network, as a surrogate of FEM for stress analysis and the trained model is capable of predicting the stress distributions with average errors less than 1%. Dabiri et al. [30] adopted XGBoost to predict the LV pressures, volumes as well as LV stresses by training with hundreds of forward FEM simulations, and results show XGBoost could capture the shape of LV pressure and pressure-volume loops very well. Noe and Davies et al. [31,32] have already presented a statistical emulation framework using Gaussian process for accelerating the parameter estimation of LV myocardium from *in vivo* data, and the computational costs can be

reduced by about three orders of magnitude. Paolo et al. [33] also used Gaussian process for inverse problems and parameter search for 3D geometric features of several classes of patients from some public databases [34].

In this paper we developed surrogate models based on three ML methods. Firstly, the original high dimensional material parameters space was projected to a low dimensional space, and the training data were obtained by forward numerical simulation with sampled parameters from the low dimensional space. Then the ML methods was used to learn the relationships between pressure-volume and pressure-strain respectively. Finally, the trained models were used as surrogate models to solve the inverse problem of parameter estimation of LV myocardium.

2. Methods

(a) Biomechanical Model of Left Ventricular Passive Dynamics

In this section, we will introduce the passive dynamics of the left ventricle in diastole. A human LV model from our previous studies are used here as shown in Figure 1 with 53548 nodes and 48050 hexagonal elements. A rule-based approach is used to generate the layered myofibre structures within the myocardium, they are the fibre direction (\mathbf{f}), the sheet direction (\mathbf{s}) and the sheet-normal (\mathbf{n}). In this work, the fibre angle α linearly rotates from -60° to 60° from endocardium to epicardium, and the sheet angle β linearly rotates from -45° to 45° in a similar way, and $\mathbf{n} = \mathbf{f} \times \mathbf{s}$. Details of the LV model reconstruction can be found in [7,8].

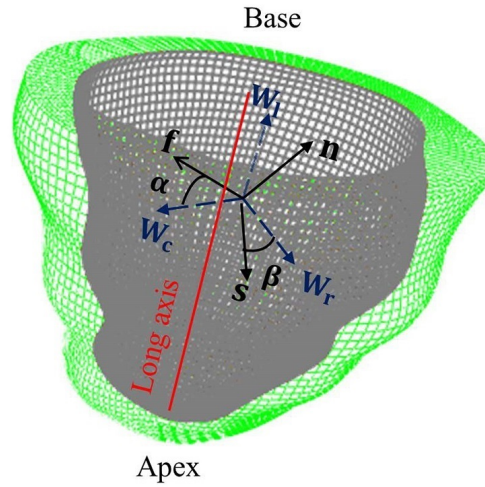


Figure 1. Visualize of the LV geometry. (\mathbf{f} , \mathbf{s} , \mathbf{n}) are the fibre, sheet and sheet-normal axes, as described in the text, and (\mathbf{W}_c , \mathbf{W}_l , \mathbf{W}_r) are coordinate axes that indicate the local circumferential, longitudinal and radial axes. The helix angle α is defined to be the angle between \mathbf{f} and \mathbf{W}_c in the plane spanned by \mathbf{W}_c and \mathbf{W}_l , and the sheet angle β is defined to be the angle between \mathbf{s} and \mathbf{W}_r in the plane spanned by \mathbf{W}_l and \mathbf{W}_r . The grey colour represents the epicardium and the blue colour represents endocardium.

The myocardium is considered to be an nearly-incompressible and orthotropic hyper-elastic material developed in [11], namely the H-O constitutive model,

$$\Psi = \frac{a}{2b} \exp[b(I_1 - 3) - 1] + \sum_{i=f,s} \frac{a_i}{2b_i} \exp[b_i(I_{4i} - 1)^2 - 1] + \frac{a_{fs}}{2b_{fs}} \exp[b_{fs}I_{8fs}^2 - 1] + \frac{1}{2}K(J - 1)^2, \quad (2.1)$$

where $a, b, a_f, b_f, a_s, b_s, a_{fs}, b_{fs}$ are material parameters, the term $\frac{1}{2}K(J-1)^2$ accounts for the incompressibility of myocardium, and K is a constant bulk modulus (10^6 Pa). I_1, I_{4i}, I_{8fs} ($i = f, s$) are the invariants along myofibre, sheet, and sheet-normal directions,

$$\begin{aligned} I_1 &= \text{tr}(\mathbf{C}), \\ I_{4f} &= \mathbf{f}_0 \cdot (\mathbf{C}\mathbf{f}_0), \\ I_{4s} &= \mathbf{s}_0 \cdot (\mathbf{C}\mathbf{s}_0), \\ I_{8fs} &= \mathbf{f}_0 \cdot (\mathbf{C}\mathbf{s}_0), \end{aligned} \quad (2.2)$$

$\mathbf{C} = \mathbf{F}^T \mathbf{F}$ is the right Cauchy-Green deformation tensor and \mathbf{F} is the deformation gradient. $\mathbf{f}_0, \mathbf{s}_0$ and \mathbf{n}_0 are the layered fibre structure in the reference configuration. In the current configuration, the fibre structure is defined as

$$\mathbf{f} = \mathbf{F} \mathbf{f}_0, \quad \mathbf{s} = \mathbf{F} \mathbf{s}_0, \quad \mathbf{n} = \mathbf{F} \mathbf{n}_0. \quad (2.3)$$

The passive response of the LV dynamics in diastole is implemented and solved using the finite-element approach in a general-purpose finite-elements package ABAQUS (Simulia, Providence, RI, USA). The basal plane is fixed in the long-axial direction (\mathbf{W}_l -axis) and the circumferential direction (\mathbf{W}_c -axis), but allowing radial expansion. A linearly ramped end-diastolic pressure with 25 equal steps is applied to the endocardial surface and simulation results are saved in each loading step. In this study, the LV cavity volume and principal strains at certain locations are chosen from the forward ABAQUS simulations: the maximum principal strain (ε_{\max}), which is related to myofibre strain and the minimum principal strain (ε_{\min}), which is related to wall thinning in diastole. In detail, to extract principal strains, 20 locations within the LV wall were randomly chosen using *random* function in Matlab [35], and then the maximum and minimum principal strains are spatially averaged at each loading step. Note that we only select 20 random positions once, and the same 20 positions are used to extract strain data for different simulations. The ventricular volume is the volume enclosed by the endocardial surface. Figure 2A-C shows the relationships between the pressure and the cavity volume, the mean maximum and minimum principal strains from one simulation till end diastole. Published studies have found that exponential functions can characterize the non-linear relationship between the pressure and the volume very well. For example, based on ex vivo human heart experiments, Klotz et al. [36] found that the relationship between the normalized volume (v_n) and the loaded pressure (p) can be approximated with $p = A_n v_n^{B_n}$, in which A_n and B_n are coefficients, and both coefficients are almost invariants between subjects and species. Thus, in this study, we assume the relationships between the pressure and the ventricular volume, the mean maximum and minimum principal strains also comply the exponential function as suggested by Klotz et al [36], they are

$$p = \alpha_0 v_n^{\beta_0}, \quad p = \alpha_1 (\bar{\varepsilon}_{\max})^{\beta_1}, \quad p = \alpha_2 |\bar{\varepsilon}_{\min}|^{\beta_2}, \quad (2.4)$$

in which $v_n = \frac{v-v_0}{v_0}$ is the normalized volume with respect to the initial value v_0 , $\bar{\varepsilon}_{\max}$ and $\bar{\varepsilon}_{\min}$ are mean maximum and minimum principal strains at the chosen 20 positions. α_0 and β_0 can be least-squared fitted to the p - v_n curve. α_1 and β_1 are derived from the p - $\bar{\varepsilon}_{\max}$ curve, and α_2 and β_2 are derived from the p - $\bar{\varepsilon}_{\min}$ curve. Here because the value of minimum principal strain is in negative, we take its absolute value. Figure 2D-F shows the fitting results from one simulation, the p - v_n , p - $\bar{\varepsilon}_{\max}$ and p - $\bar{\varepsilon}_{\min}$ are all fitted well with Equation (2.4). Therefore, the output features from a forward ABAQUS simulation are reduced to 3 pairs of data, rather than 3 different curves as show in Figure 2A-C.

It has been shown that there is a strong correlation among the eight parameters in Equation (2.1) [35], thus, it can be very challenging to uniquely determine all 8 parameters by using only end-diastolic strain and volume. Following two recent studies from Noe et al. [31]

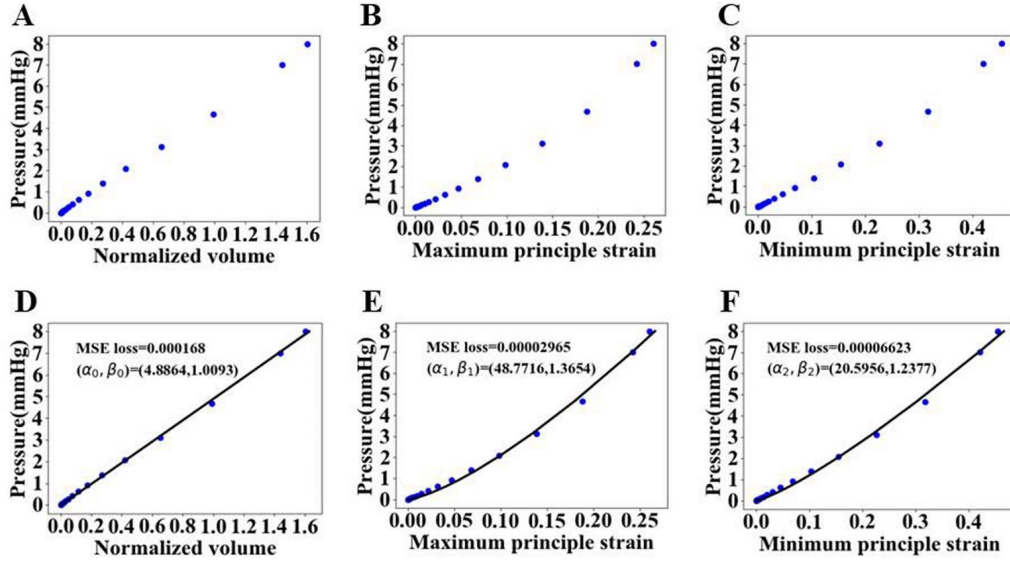


Figure 2. A simple example of end-diastolic pressure-normalized volume and pressure-mean principal strain relationships. (A-C) are relationships between the loaded pressure and the normalized volume, mean maximum principal strain, mean minimum principal strain from one simulation. And (D-F) shows the results of curve fitting, respectively. The discrete point are the original data from the round of simulation, the solid line are the fitted curve using the expression as Equation (2.4).

Davies et al. [32], the eight dimensional parameter space is projected into a 4-dimensional space

$$\begin{aligned}
 a &= q_1 a_0, & b &= q_1 b_0, \\
 a_f &= q_2 a_{f0}, & a_s &= q_2 a_{s0}, \\
 b_f &= q_3 b_{f0}, & b_s &= q_3 b_{s0}, \\
 a_{fs} &= q_4 a_{fs0}, & b_{fs} &= q_4 b_{fs0},
 \end{aligned} \tag{2.5}$$

where $\mathbf{q} = (q_1, q_2, q_3, q_4) \in [0.1, 5]^4$ are the reduced parameters, $a_0 = 0.22$ kPa, $b_0 = 1.62$, $a_{f0} = 2.43$ kPa, $b_{f0} = 1.83$, $a_{s0} = 0.39$ kPa, $b_{s0} = 0.77$, $a_{fs0} = 0.39$ kPa, $b_{fs0} = 1.70$ are the empirical reference values for a healthy left ventricular model [37].

(b) Surrogate model

One of the major obstacles to finite-element modelling in cardiac mechanics is the long computational time and requiring tremendous computational resources, in particular if an inverse problem is formulated with finite-element cardiac models. One way of alleviating this burden is to use ML-based surrogate models [26,38]. The basic idea of surrogate model is to construct a measurable functional $\hat{\mathbf{m}}(\mathbf{q})$ to statistically approximate the underlying dynamical model $\mathbf{m}(\mathbf{q})$, the ABAQUS FE model in this study, based on costly sampling to get the dataset \mathbb{D} . Whenever a new forward simulation is needed, the costly function evaluation of $\mathbf{m}(\mathbf{q})$ can be replaced by a fast prediction from the surrogate model $\hat{\mathbf{m}}(\mathbf{q})$. Figure 3 shows the concept of surrogate models compared to the forward simulator.

(i) Design of sampling

In order to fit the surrogate model to the simulator, the ABAQUS FE model, we need to generate the dataset \mathbb{D} . For each forward simulation, we consider the input features are the reduced 4

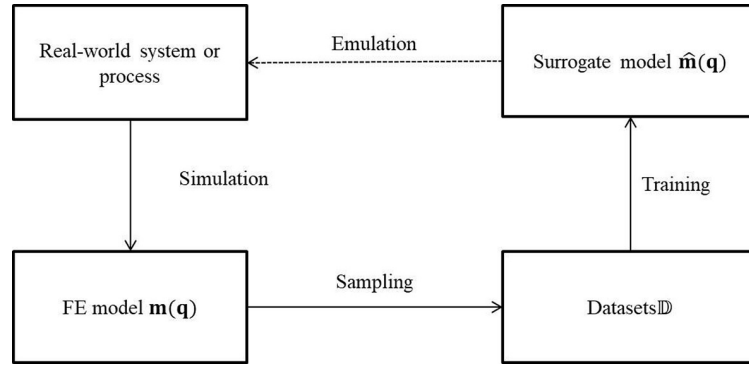


Figure 3. The main idea of surrogate model. First, the real-world system, that is the diastolic filling of the left ventricle, needs to be simulated using the forward model $\mathbf{m}(\mathbf{q})$. Then an reasonable sampling process is used to collect the datasets \mathbb{D} . Next the forward model is estimated by a surrogate model $\hat{\mathbf{m}}(\mathbf{q})$ according to the collected datasets. In the end, the well-established surrogate model is used to emulate real-world system (the diastolic filling of the left ventricle).

parameters defined in Equation (2.5), and the output features are three pair of positive values (α_0, β_0) , (α_1, β_1) and (α_2, β_2) as defined in Equation (2.4). Thus the datasets can be described as

$$\mathbb{D} = \{(\mathbf{q}^i, \mathbf{y}^i) | \mathbf{q}^i \in [0.1, 5]^4, \mathbf{y}^i \in (0, +\infty)^6, i = 1, 2, \dots, n\}, \quad (2.6)$$

where n is the total sample number. The input features q_j^i represents the j -th feature from i -th sample, and the output features are ordered as $\mathbf{y}^i = (y_1^i, \dots, y_6^i) = (\alpha_0^i, \beta_0^i, \alpha_1^i, \beta_1^i, \alpha_2^i, \beta_2^i)$.

Commonly sampling methods are the grid sampling, the uniform sampling, and the Latin hypercube sampling, etc. [26]. The most straightforward way of sampling is using a rectangular grid of points or from a uniform distribution. However, this can easily lead to points being clustered together, and causing an ineffective coverage of the parameter space. The Latin hypercube sampling [38] is a statistical method for generating a near-random sample of parameter values from a multidimensional distribution. When sampling a function of N variables, the range of each variable is divided into M equally probable intervals, M sample points are then placed to satisfy the Latin hypercube requirements, which will ensure from each placed hypercube we could travel the function space along any direction parallel with any of the axes without encountering any other placed hypercube. This is one of the main advantages of this sampling scheme. In this paper, we generated 10,000 samples, of which 90% are used for training and the remainders are for testing.

(ii) Strategy for output features

In this study, we will use three ML-based surrogate models, namely K-nearest neighbour (KNN) [39], XGBoost [30,40], and multi-layer perceptron (MLP) [41]. The three chosen ML models can be considered to be a supervised learning regression problem [42].

We find that the range of these 6 output features can be very different, specifically the ranges of α_0 , α_1 and α_2 are in hundreds and the ranges of β_0 , β_1 and β_2 are from 1 to 3, which can be seen in Figure 7. To balance the output features, the general methods contain feature scaling and normalization [43]. In this work, we first normalize the output feature, which can be expressed as $\frac{y - \bar{y}}{\text{std}(y)}$, where \bar{y} and $\text{std}(y)$ represent the mean value and standard deviation of y , then a multiple model is used to each output feature. Figure 4 shows the main idea of the multiple model, 6 sub-models are trained by hyper-parameters tuning, note these sub-models can have different hyper-parameters.

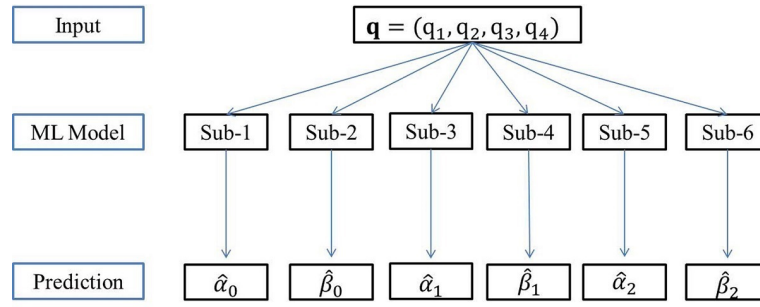


Figure 4. 6 sub models for output features. ML model contains six sub-models, and each sub-model predicts exactly one feature.

(iii) K-nearest neighbour

Nearest neighbour methods are based on a simple idea by treating the training set as the model and making prediction of new points based on how close they are to those in the training set. One natural way is to make prediction using the closest training data points, while most datasets contain some degree of noise, a more common method would be to take a weighted average of a set of K nearest neighbours [39,44]. KNN method is a basic classification and regression method, the choice of K value, distance metric and decision rule are three basic criterias [45]. For the training data $\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^n$, and the corresponding target values y^1, y^2, \dots, y^n . Here y^i is denoted as any feature in \mathbf{y}^i . The prediction in a new point \mathbf{q}^p can be realized by firstly searching a set of K nearest neighbours in the training set, and then \hat{y}^p is given by a weighted averaged of those K nearest neighbours, that is

$$\hat{y}^p = \sum_{i=1}^K \omega_i y^i \quad (2.7)$$

in which ω_i is the weight for the i -th nearest neighbour. Note the closer the point to the predicted point, the greater weight this point takes, and ω_i is defined as

$$\omega_i = \frac{1/L_{1,2}(\mathbf{q}^i, \mathbf{q}^j)}{\sum_{j=1}^K 1/L_{1,2}(\mathbf{q}^j, \mathbf{q}^p)} \quad (2.8)$$

in which $L_{1,2}$ is a distance metric function, which can be either as $L1$ or $L2$:

$$L_1(\mathbf{q}^i, \mathbf{q}^j) = \sum_{k=1}^4 |q_k^i - q_k^j| \quad \text{and} \quad L_2(\mathbf{q}^i, \mathbf{q}^j) = \sqrt{\sum_{k=1}^4 (q_k^i - q_k^j)^2}, \quad (2.9)$$

The primary reasons for using KNN method as a surrogate model in this study are its proven capabilities as function approximation tool from a given data set, and its simple structure to implement. The algorithm for KNN is

Algorithm 1 KNN method for each sub-model.

Input: Datasets \mathbb{D} and one new point \mathbf{q}^p

Output: Prediction value \hat{y}^p for the new point \mathbf{q}^p

Hyper parameters: K , the distance metric function L_1 or L_2

- 1 Select the value of K , distance metric function as Equation (2.9) ;
 - 2 Compute the distance between points in \mathbb{D} and the point \mathbf{q}^p according the selected distance metric function in 1;
 - 3 Find the K nearest neighbour to point \mathbf{q}^p according the result of 2;
 - 4 Predict the value of \mathbf{q}^p according to Equations (2.7)-(2.8).
-

(iv) XGBoost

Tree boosting is a highly effective and widely used machine learning method, in particular XGBoost (eXtreme Gradient Boosting), a boosting tree model developed by Chen [40], an engineering implementation based on GBDT [46], which boosts many tree models to form a strong regression model. In XGBoost, the additive strategy is used to predict output variable [47], learning one CART tree [48] per iteration to fit the residual of the predicted results of the previous trees to the true value of the training sample. Note the prediction value in \mathbf{q}^p at step t as $\hat{y}^{p(t)}$. Then we can write each sub-model in the form:

$$\hat{y}^{p(t)} = \sum_{j=1}^t g_j(\mathbf{q}^p), \quad g_j \in \mathcal{G}, \quad (2.10)$$

where t is the number of CART trees, g_j is a function in the functional space \mathcal{G} , and \mathcal{G} is the set of all possible regression trees. The loss function to be optimized is given for each feature, which is defined as

$$\mathcal{L}_{XGB}^{(t)} = \sum_{i=1}^n l(y^i, \hat{y}^{i(t)}) + \sum_{j=1}^t \Omega(g_j) = \sum_{i=1}^n l(y^i, \hat{y}^{i(t-1)} + g_t(\mathbf{q}^i)) + \Omega(g_t) + \text{constant} \quad (2.11)$$

Here l is a distance metric function that measures the distance between the prediction \hat{y}^i and the target $y^{i(t)}$. The term Ω penalizes the complexity of the model. We take the Taylor expansion of the loss function up to the second order:

$$\mathcal{L}_{XGB}^{(t)} = \sum_{i=1}^n \left[l(y^i, \hat{y}^{i(t-1)}) + p_i g_t(\mathbf{q}^i) + \frac{1}{2} h_i g_t^2(\mathbf{q}^i) \right] + \Omega(g_t) + \text{constant}, \quad (2.12)$$

where the p_i and h_i are the first and second order derivatives of $l(y^i, \hat{y}^{i(t-1)})$. After we remove all the constants, the specific objective at step t becomes:

$$\mathcal{L}_{XGB}^{(t)} = \sum_{i=1}^n \left[p_i g_t(\mathbf{q}^i) + \frac{1}{2} h_i g_t^2(\mathbf{q}^i) \right] + \Omega(g_t). \quad (2.13)$$

For more algorithm details, please refer to references [40,46].

(v) Multi-layer perceptron

MLP is a class of feedforward artificial neural network [41], consisting of at least three layers of nodes: an input layer, a hidden layer and an output layer. Fully connected multiple hidden layers can be added to the hidden layer. Except for input nodes, each node in MLP is a neuron with a non-linear activation function. A feedforward neural network should have a linear output layer, and at least one hidden layer of an activation function. Universal approximation theorem [49] indicates that if we gives the network a sufficient number of hidden layers, it can approximate the Borel measurable function of any finite dimensional space to another finite dimensional space. In this sense, multilayer feedforward networks are a class of universal approximators. In this work, we adopted a MLP model with only single hidden layer [50]. The input layer had four nodes which will take q_1, q_2, q_3, q_4 in the reduced space, and the output layer was combined with three pairs of least square coefficients as shown in Figure 5.

Activation function here we fixed as rectified linear unit (ReLU) at the node of hidden layer [51], which can be describe as

$$\sigma(\mathbf{q}) = \max(\mathbf{0}, \boldsymbol{\omega}_h^T \cdot \mathbf{q} + \mathbf{b}_h), \quad (2.14)$$

in which $\boldsymbol{\omega}_h$ and \mathbf{b}_h represents weights and bias between input layer and output layer. Because the MLP model only contains one hidden layer, thus, the prediction for each feature y^p can be

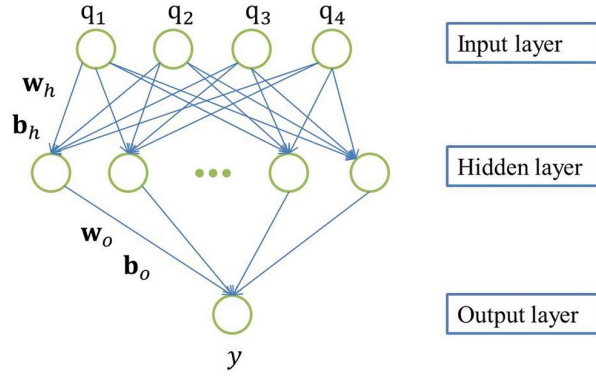


Figure 5. Multilayer perceptron model with one single hidden layer

represented as

$$\begin{aligned}\hat{y}^p &= \omega_o^T \cdot \sigma(\mathbf{q}^p) + \mathbf{b}_o \\ &= \omega_o^T \cdot \max(0, \omega_h^T \cdot \mathbf{q}^p + \mathbf{b}_h) + \mathbf{b}_o\end{aligned}\quad (2.15)$$

Here ω_o and \mathbf{b}_o represents weights and bias between hidden layer and output layer. Mean square error is adopted as loss function with L_2 penalty (regularization term) shrinks model parameters to prevent overfitting, which is defined as

$$\mathcal{L}_{MLP} = \frac{\sum_{i=1}^n (y^i - \hat{y}^i)^2}{n} + \lambda(\|\omega_o\|_2 + \|\omega_h\|_2), \quad (2.16)$$

where \mathbf{y} are the true values, and $\hat{\mathbf{y}}$ are the predictions from MLP. λ is penalty parameter and we fix as $1e-4$. $\|\cdot\|$ denote the L_2 norm in euclidean space.

(c) Training ML-based surrogate models

A good ML model not just fits the training data well, but also has a good predictability for the test data. The choice of hyper-parameters can significantly affect the performance of model, but determining good values can be complex [52]. To avoid over-fitting and determine optimal hyper-parameters, in this paper, a grid search method with a 5-fold cross validation is adopted [53]. The mean squared error (MSE) is used to as the loss function

$$\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_{i=1}^n (y^i - \hat{y}^i)^2}{n}, \quad (2.17)$$

where $\mathbf{y} = (y^1, y^2, \dots, y^n)$ are the true values derived from the simulator, and $\hat{\mathbf{y}} = (\hat{y}^1, \hat{y}^2, \dots, \hat{y}^n)$ are the predictions from ML models. We further introduce coefficient of determination R^2 , which is used to evaluate the predictability of a ML model:

$$\begin{aligned}R^2(\mathbf{y}, \hat{\mathbf{y}}) &= 1 - \frac{\sum_{i=1}^n (y^i - \hat{y}^i)^2}{\sum_{i=1}^n (y^i - \bar{y})^2}, \\ \bar{y} &= \frac{1}{n} \sum_{i=1}^n y^i.\end{aligned}\quad (2.18)$$

It represents the proportion of variance that has been explained by input variables in the model. Best possible score is 1.0 and it can be negative because the model can be arbitrarily worse.

Tuning hyper-parameters using grid search method in a multidimensional space always take a long time (usually several hours even a few days). To select the grid values reasonably and efficiently, we consider to optimize one hyper-parameter each time to acquire a small grid interval. Take the MLP method as an example. We first fix initial learning rate (*learning_rate_init*) as 0.01, then change the size of hidden layer (*hidden_layer_sizes*) in a big grid range to get the optimal value, next we decided the range of *hidden_layer_sizes* as interval nearby the optimal value. Using the same way, we fix *hidden_layer_sizes* and change the range of *learning_rate_init* to get a small interval range. Grid search values of other two ML methods are similar to MLP method. All grid values of each hyper-parameter are shown in Table 1.

Table 1. Grid values of hyper-parameters

ML methods	Hyperparameters	Grid search values
KNN	K value	1,2,4,6,8,10
	Distance metric	L_1, L_2
XGBoost	<i>max_depth</i>	4,6,8,16
	<i>learning_rate</i>	0.05,0.1,0.075
	<i>n_estimators</i>	500,1000, 2000
MLP	<i>hidden_layer_sizes</i>	256,512,1024
	<i>learning_rate_init</i>	$5e^{-3}, 1e^{-3}, 5e^{-2}, 1e^{-2}$

(d) Parameter estimation

Estimating myocardial property from measured volume and strain is very challenging because of sparse of data, co-relation among different parameters. In general, an inverse problem is formulate by solving a constrained optimization problem with thousands of forward simulations which may take days and weeks [54]. The objective function is usually built on the differences between computational model derived data and experimental measurements (e.g. volume and strain of diastolic geometry), constitutive parameters are iteratively adjusted until the objective function is minimized [19,35]. We further apply the three ML-based surrogate models for myocardial parameter estimation by using a synthetic LV model implemented in ABAQUS with known parameters to provide referential volume v^* and principal strains ε^* in diastole.

The surrogate models do not directly predict volume and principal strains, but can be easily converted their predictions into volume and principal strains, as shown in Equation (2.4). Then we can define the objective function [35] as

$$f(\mathbf{q}) = \sum_{i=1}^T \left[(v_i - v_i^*)^2 + (\bar{\varepsilon}_{max(t_i)} - \bar{\varepsilon}_{max(t_i)}^*)^2 + (\bar{\varepsilon}_{min(t_i)} - \bar{\varepsilon}_{min(t_i)}^*)^2 \right], \quad (2.19)$$

where T represents total loading step which is equal to 25 as described in Section (a). The $v_i, \bar{\varepsilon}_{max(t_i)}, \bar{\varepsilon}_{min(t_i)}$ represents the normalized volume, mean maximum and minimum principal strains in each loading process of time step t_i . $v_i^*, \bar{\varepsilon}_{max(t_i)}^*, \bar{\varepsilon}_{min(t_i)}^*$ are corresponding to the referential normalized volume, mean maximum and minimum principal strains. Figure 6 shows the iterative process of the inverse problem for myocardial parameter estimation. An differential evolution (DE) method is implemented in optimization process based on a python library `scipy.optimize`.¹ Differential evolution is a stochastic population based method that is useful for global optimization problems [55]. At each pass through the population, the DE algorithm mutates each candidate solution by mixing with other candidate solutions to create a trial candidate. To compare the efficiency between using the ML-based surrogate models and the

¹<https://docs.scipy.org/doc/scipy-1.0.0/reference/tutorial/optimize.html>

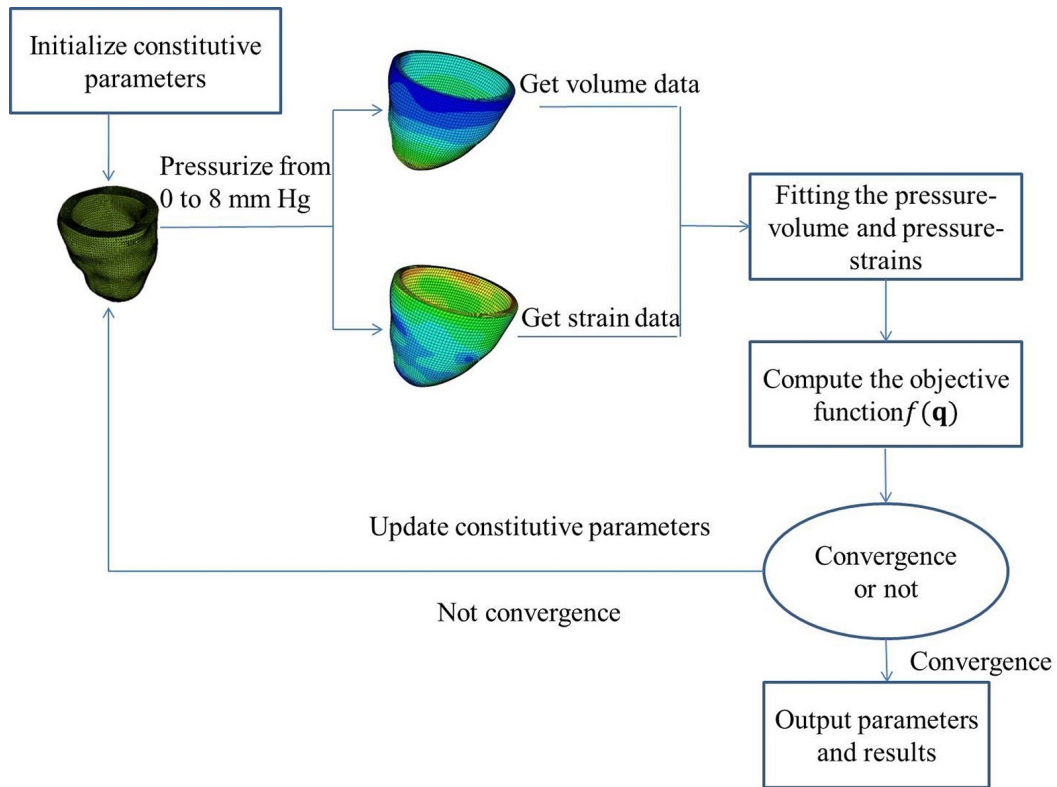


Figure 6. Iterative process of the inverse problem. (i) initialize the constitutive parameters; (ii) pressurize the LV endocardial surface to obtain the volume and strain data of each analysis step; (iii) compute the objective function to determine whether the algorithm converges; (iv) if the algorithm converges, output parameters and results otherwise a new round of iterative process is performed by updating the constitutive parameters.

forward simulator for parameter estimation, we further apply the inverse approach developed in [35], which requires running ABAQUS simulation during each iteration.

The system we use is a 64-bit Windows 7 Operating System with 1.7 GHz Intel Core E5-2609 CPU and 32 GB RAM. All methods are performed on sixteen threadings for parallelism.

3. Results

(a) Output feature reduction

Figure 7 shows the distributions of each feature from all 10,000 samples with box-plots. Overall, MSEs of fitting to the volume, the mean maximum and minimal principal strains are around $1e-4$, indicating LV diastolic dynamics from the ABAQUS simulator can be well characterized by the three pairs of features $\alpha_0, \beta_0, \alpha_1, \beta_1, \alpha_2$ and β_2 , a massive dimension reduction for describing LV dynamics.

(b) ML methods and hyper-parameters tuning

Note that the higher the R^2 , the better the model predictability. Firstly, we adopt 5 fold cross validation to train the ML models on training data based on the MSE loss function and R^2 metric function. Then, we test every model on test data and select the optimal hyper-parameters in the

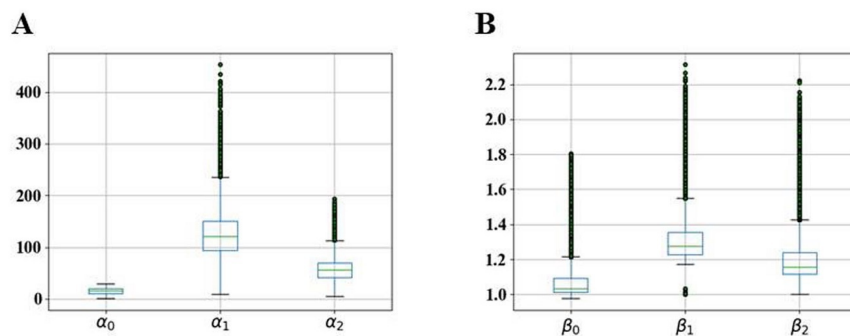


Figure 7. Distribution of six outputs of 10000 samples. (A) boxplots of the α_0 , α_1 , α_2 and (B) boxplots of the β_0 , β_1 , β_2

Table 2. Hyper parameters tuning results for 6 sub-models

ML methods	Hyper parameters	Best hyper parameters					
		α_0	β_0	α_1	β_1	α_2	β_2
KNN	K value	10	8	8	6	8	6
	Distance metric	L_2	L_1	L_1	L_1	L_1	L_1
XGBoost	<i>max_depth</i>	6	6	6	6	4	4
	<i>learning_rate</i>	0.05	0.05	0.1	0.05	0.05	0.1
	<i>n_estimators</i>	1000	1000	1000	1000	1000	1000
MLP	<i>hidden_layer_sizes</i>	1024	1024	1024	512	1024	1024
	<i>learning_rate_init</i>	1e-3	1e-3	1e-3	1e-3	1e-3	1e-3

grid values with which model obtains the highest R^2 score on test data. For each surrogate model, 6 identical ML sub-models are constructed, and each sub-model has different optimal hyper-parameters, which can be found in Table 2.

Table 3. Values of loss function on test data for 6 sub model.

Methods	MSE loss function					
	α_0	β_0	α_1	β_1	α_2	β_2
KNN	0.005492	0.045363	0.044863	0.073405	0.025613	0.054499
XGBoost	0.000231	0.000416	0.000416	0.000620	0.000716	0.000638
MLP	0.000235	0.000339	0.005232	0.017846	0.001483	0.009786

Table 4. Values of R^2 on test data for 6 sub model.

Methods	R^2 score					
	α_0	β_0	α_1	β_1	α_2	β_2
KNN	0.994557	0.952787	0.949117	0.924738	0.972407	0.945222
XGBoost	0.999771	0.999567	0.999567	0.999364	0.999229	0.999359
MLP	0.999754	0.998619	0.990997	0.938786	0.997589	0.966504

Figure 8 shows the predictions from the testing samples for each feature from the three ML-based surrogate models. The red line suggests exact predictions of the true values, blue points

are the predictions. The closer the blue points to the red line, the better predictability on the ML model. It can be found that the predictions from the XGBoost and MLP models can predict the features very well, all predictions almost lie in the red line. While the predictions of the KNN model seems performing worse for large true values, and MLP model performs not well for some features, such as β_1 and β_2 . Tables 3-4 summarizes corresponding MSE losses and R^2 scores for the testing samples. Similarly, the MSE losses of the KNN model are nearly ten orders higher than other two ML-based surrogate models. Meantime, the R^2 score of the XGBoost and MLP models are nearly over 0.99 and both higher than the R^2 score of KNN model which is about 0.95. Therefore, XGBoost and MLP methods are more suitable to learn our ABAQUS LV model in diastole because of high scores with both training and testing data, but not for the KNN.

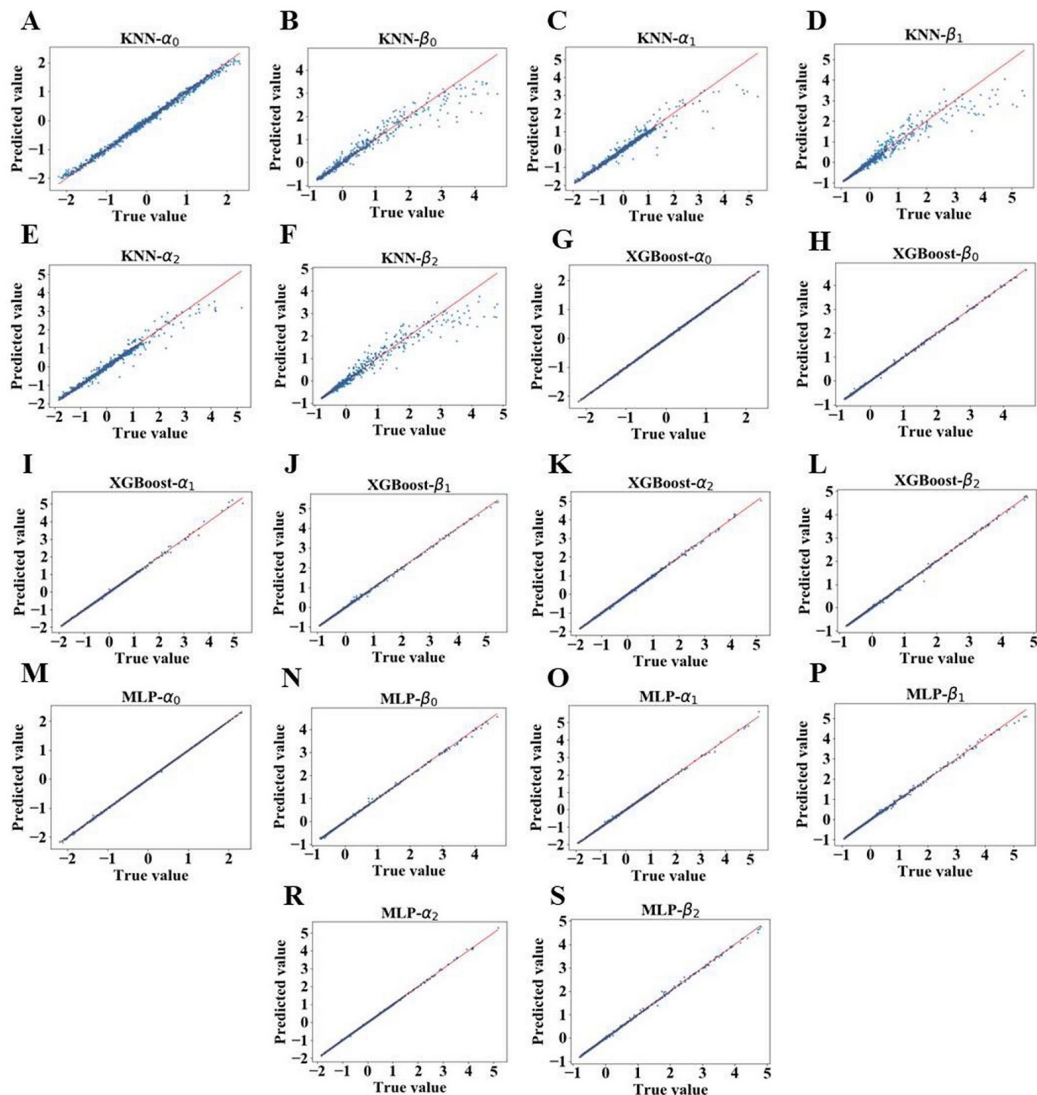


Figure 8. Distributions of true value and predictions of six output features for three ML models. (A-F), (G-L) and (M-S) are the results of KNN, XGBoost and MLP, respectively.

(c) Parameter estimation

To inversely infer the 8 unknown parameters using inverse approach developed in [35], 378 forward ABAQUS simulations are involved, which takes about 63 hours (almost 3 days). The objective function with respect to the iterations is shown in Figure 9A. While when using the ML-based surrogate models, it only took a few seconds, especially for the MLP model, which takes only 69 seconds (around 1 minute). Figure 9B shows the objective function with respect to the iterations for the 3 ML-based surrogate models. It can be found, the XGBoost model can reduce the objective function the most, followed by the MLP model, and poorest in the KNN model. Table 5 summarizes the estimated parameters, the final objective function and the total CPU time. It can be seen that the estimated parameters derived from all methods are fairly well, except for the value of q_3 and q_4 from the KNN and MLP methods. To check the possible multimodality of q_3 and q_4 , we plot the distributions of objective function by fixing q_1 and q_2 equal to the estimated value derived from three surrogate model. Figure 10 show the distributions of objective function on q_3 and q_4 for all three ML methods. Clearly, the blue area occupies a large proportion of the whole region, and changes slowly near the true value. Figure 9C-D shows the stretch-stress relationships with estimated parameters by the three ML methods and the ground truth data. For the stretch along fibre direction, the curves of three ML methods and the ground truth data are nearly identical. Slightly higher discrepancies can be found for the Cauchy stress along the sheet direction derived from the three ML methods. The poor estimations of the stretch-stress response was also reported in the studies from Neo et al. [31] and Davies et al. [32], which may be explained by the limitation of the reduced material parameter space.

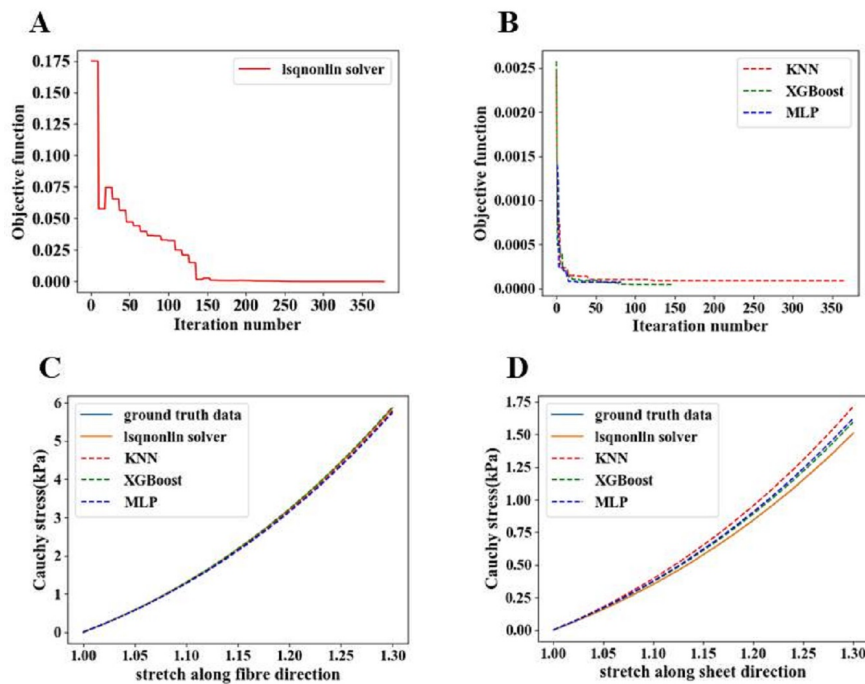
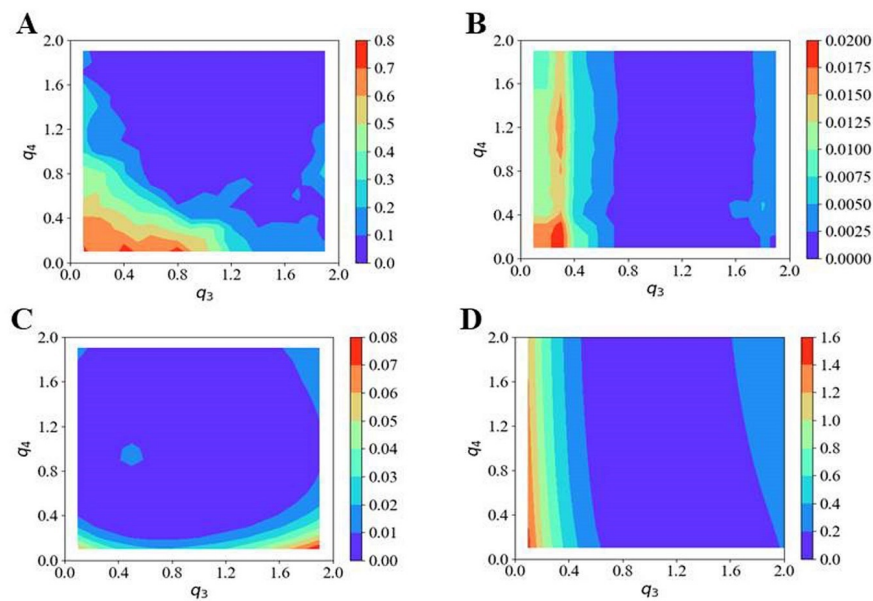


Figure 9. The optimization iteration curve by lsqnonlin solver (A) and three ML solvers (B). And relationships between stretch and Cauchy stress under uni-axial tension with estimated parameters by three ML methods along fibre (C) and sheet (D) directions.

Table 5. The estimated parameters by lsqnonlin and ML solvers.

Methods	q_1	q_2	q_3	q_4	$f(q)$	cost time
Referenced parameters [35]	1	1	1	1	-	-
lsqnonlin solver	1.0000	0.9999	1.0001	1.0002	5.8537e-08	63h
KNN	0.9634	0.9896	1.1625	1.8253	3.1227e-05	5930s
XGBoost	1.0228	1.0040	1.06299	0.9689	1.6431e-05	587s
MLP	1.0066	0.9788	1.0869	1.2164	1.7441e-05	69s

**Figure 10.** The distribution of objective function changes by q_3 and q_4 , where the range of q_3 and q_4 are from 0.1 to 2. (A-D) respectively represent the mesh of objective function by the KNN, XGBoost, MLP model and former ABAQUS simulator.

4. Discussion

In this study, we have developed three surrogate models based on three ML methods, namely the KNN, XGBoost and MLP. These models are used to estimate the material parameters of LV myocardium using the cavity volume and the maximum and minimum principal strains. The developed ML models are evaluated through cross validation, and the average scores of R^2 are very close to one. Comparing with the conventional parameter estimation based on numerical evaluations of the forward simulation model, the ML-based surrogate models provide a significant improvement in computational time with an acceleration by tens or even hundreds of times.

The dimension of output features in a general regression model is one, for many method it can easily expand to multiple outputs. In this study, there are 6 output features, and the range of these features are different with each other, as shows in Figure 7. Thus a multiple regression model is reconstructed with multi-output features, which means we need to train multiple models in order to fit each output feature to one regression model, and then predict each output feature by its corresponding model. This strategy can balance features and reduce the error at the end. The primary reason to use these three ML methods are the samples in this work are limited and the correspond input and output features are very few and these three ML model are easily

to implement. For instance, the neural network framework was proposed for the parameter estimation of aortic wall [56] and XGBoost was also used to predict the LV mechanics [30]. Other ML methods can also be used for material parameter estimation in soft tissue biomechanics, such as the Gaussian Process [31,32].

A limitation of the current approach is the fact that the data we used are from one single human LV, while the function of LV is individual. Even in the same pressure situation, the volume and strain may exist massive discrepancy. Therefore the estimated parameters of constitutive model might be different for different people. To validate the surrogate models we proposed is feasible or not for different human LV subjects, we test two sample cases based on the different LV subjects from the study of Gao et al [35]. Considering the discrepancy of myocardium parameters estimated from experimental data for different human LV, we assume the parameters of this two case are already known. Then we simulate again from the ABAQUS FE model to get two out sample data. Test results are listed in Table 6. Comparing with results of Table 3, the MSE between test data and predictions in this paper are far less than other two different cases, therefore the ML methods could not accurately predict the relationships of pressure-volume and pressure-strain for different LV subjects. The main reason of this discrepancy is the prediction of other two cases also use the models trained from this human subject. To analysis the difference of different subjects, we must study the relationships of different subjects firstly. These works may be included in future developments of our ML methodology.

Table 6. Loss of out sample test.

Methods	MSE loss function					
	α_0	β_0	α_1	β_1	α_2	β_2
KNN	0.364341	1.133695	0.185119	2.393936	0.035711	2.12244
XGBoost	0.376170	1.138444	0.166671	2.383279	0.033399	2.117243
MLP	0.3870	1.131395	0.168273	2.368105	0.028425	2.120199

5. Conclusions

In real-world optimization problems, the function evaluations usually require a large amount of computational time. Surrogate based analysis and optimization plays a valuable role in solving the large scale computational problem. In this paper, we have developed three surrogate models and applied these methods to solve the parameter estimation problem of the LV myocardium. Three surrogates based on different ML methods including the KNN, XGBoost and MLP, and they are using 10000 samples which are simulated from the original finite element model by a Latin hypercube sampling method. By comparing with a traditional gradient-based optimization method, our results show the ML methods can learn the relationships of pressure-volume and pressure-strain very well and the parameter inference using the surrogate model can be done in minutes, in particular the XGBoost and MLP based surrogate models. Our results further show that the XGBoost based surrogate model is the best one for predicting the LV diastolic dynamics and estimating parameters than the other two surrogate models. Further studies are warranted to investigate how the XGBoost based surrogate model can be used for fast emulating cardiac pump function in a multi-physics and multi-scale framework.

Acknowledgment

This research is supported by the National Natural Science Foundation of China (Grant Nos. 11871399, 11471261). G.Y. Zhu acknowledge the funding from the National Natural Science Foundation of China (Grant No. 11802227). HG also acknowledge the funding from

References

1. Hellermann JP, Goraya TY, Jacobsen SJ, Weston SA, Reeder GS, Gersh BJ, Redfield MM, Rodeheffer RJ, Yawn BP, Roger VL. 2003 Incidence of heart failure after myocardial infarction: is it changing over time?. *American journal of epidemiology* **157**, 1101–1107.
2. De Couto G, Ouzounian M, Liu PP. 2010 Early detection of myocardial dysfunction and heart failure. *Nature reviews Cardiology* **7**, 334.
3. Sermesant M, Moireau P, Camara O, Sainte-Marie J, Andriantsimiavona R, Cimirman R, Hill DL, Chapelle D, Razavi R. 2006 Cardiac function estimation from MRI using a heart model and data assimilation: advances and difficulties. *Medical Image Analysis* **10**, 642–656.
4. Sugiura S, Washio T, Hatano A, Okada J, Watanabe H, Hisada T. 2012 Multi-scale simulations of cardiac electrophysiology and mechanics using the University of Tokyo heart simulator. *Progress in biophysics and molecular biology* **110**, 380–389.
5. Sachse FB. 2004 *Computational cardiology: modeling of anatomy, electrophysiology, and mechanics* vol. 2966. Springer Science & Business Media.
6. Gao H, Carrick D, Berry C, Griffith BE, Luo XY. 2014 Dynamic finite-strain modelling of the human left ventricle in health and disease using an immersed boundary-finite element method. *IMA journal of applied mathematics* **79**, 978–1010.
7. Wang H, Gao H, Luo XY, Berry C, Griffith B, Ogden R, Wang T. 2013 Structure-based finite strain modelling of the human left ventricle in diastole. *International journal for numerical methods in biomedical engineering* **29**, 83–103.
8. Gao H, Wang H, Berry C, Luo XY, Griffith BE. 2014 Quasi-static image-based immersed boundary-finite element model of left ventricle under diastolic loading. *International journal for numerical methods in biomedical engineering* **30**, 1199–1222.
9. Wittek A, Derwich W, Karatolios K, Fritzen CP, Vogt S, Schmitz-Rixen T, Blase C. 2016 A finite element updating approach for identification of the anisotropic hyperelastic properties of normal and diseased aortic walls from 4D ultrasound strain imaging. *Journal of the mechanical behavior of biomedical materials* **58**, 122–138.
10. de Souza Neto EA, Peric D, Owen DR. 2011 *Computational methods for plasticity: theory and applications*. John Wiley & Sons.
11. Holzapfel GA, Ogden RW. 2009 Constitutive modelling of passive myocardium: a structurally based framework for material characterization. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **367**, 3445–3475.
12. Wang H, Luo XY, Gao H, Ogden R, Griffith B, Berry C, Wang T. 2014 A modified Holzapfel-Ogden law for a residually stressed finite strain model of the human left ventricle in diastole. *Biomechanics and modeling in mechanobiology* **13**, 99–113.
13. Guan D, Ahmad F, Theobald P, Soe S, Luo XY, Gao H. 2019 On the AIC-based model reduction for the general Holzapfel–Ogden myocardial constitutive law. *Biomechanics and modeling in mechanobiology* pp. 1–20.
14. Gao H, Mangion K, Carrick D, Husmeier D, Luo XY, Berry C. 2017 Estimating prognosis in patients with acute myocardial infarction using personalized computational heart models. *Scientific reports* **7**, 13527.
15. Cai L, Gao H, Luo XY, Nie YF. 2015 Multi-scale modelling of the human left ventricle. *Scientia Sinica Physica, Mechanica and Astronomica* **45**.
16. Moles CG, Mendes P, Banga JR. 2003 Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome research* **13**, 2467–2474.
17. Xi J, Lamata P, Niederer S, Land S, Shi W, Zhuang X, Ourselin S, Duckett SG, Shetty AK, Rinaldi CA et al. 2013 The estimation of patient-specific cardiac diastolic functions from clinical measurements. *Medical image analysis* **17**, 133–146.
18. Liu M, Liang L, Sun W. 2017 A new inverse method for estimation of in vivo mechanical properties of the aortic wall. *Journal of the mechanical behavior of biomedical materials* **72**, 148–158.

19. Huang ZQ, Li YQ. 2018 A Novel Method for Mechanical Constitutive Parameters Estimation of Human Left Ventricle. In *IOP Conference Series: Materials Science and Engineering* vol. 398 p. 012003. IOP Publishing.
20. Finsberg H, Xi C, Tan JL, Zhong L, Genet M, Sundnes J, Lee LC, Wall ST. 2018 Efficient estimation of personalized biventricular mechanical function employing gradient-based optimization. *International journal for numerical methods in biomedical engineering* **34**, e2982.
21. Ruder S. 2016 An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
22. Locatelli M, Schoen F. 2013 *Global optimization: theory, algorithms, and applications* vol. 15. Siam.
23. Xie W, Xu W, Cai L. 2008 Optimized transfer trajectories in the earth-moon system. *Chinese Physics B* **17**, 350.
24. Goldberg D. 1989 Genetic algorithms in search. *Optimization & Machinelearning*.
25. Poli R, Kennedy J, Blackwell T. 2007 Particle swarm optimization. In *IEEE Swarm Intelligence Symposium*.
26. Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK. 2005 Surrogate-based analysis and optimization. *Progress in aerospace sciences* **41**, 1–28.
27. Dasari SK, Cheddad A, Andersson P. 2019 Random Forest Surrogate Models to Support Design Space Exploration in Aerospace Use-Case. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* pp. 532–544. Springer.
28. Jie M, Hua-feng L, Mao-xiong Y, Hong-jie H. 2012 Biomechanical model guided dual estimation of myocardial motion and material parameters. *Journal of Zhejiang University (Engineering Science)* **46**, 912–917.
29. Liang L, Liu M, Martin C, Sun W. 2018 A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. *Journal of The Royal Society Interface* **15**, 20170844.
30. Dabiri Y, Velden AVd, Sack KL, Choy JS, Kassab GS, Guccione J. 2019 Prediction of Left Ventricular Mechanics using Machine Learning. *Frontiers in Physics* **7**, 117.
31. Noe U, Lazarus A, Gao H, Davies V, Macdonald B, Mangion K, Berry C, Luo X, Husmeier D. 2019 Gaussian process emulation to accelerate parameter estimation in a mechanical model of the left ventricle: a critical step towards clinical end-user relevance. *Journal of The Royal Society Interface* **16**, 20190114–17.
32. Davies V, Noè U, Lazarus A, Gao H, Macdonald B, Berry C, Luo XY, Husmeier D. 2019 Fast Parameter Inference in a Biomechanical Model of the Left Ventricle using Statistical Emulation. *arXiv preprint arXiv:1905.06310*.
33. Di Achille P, Harouni A, Khamzin S, Solovyova O, Rice JJ, Gurev V. 2018 Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics. *Frontiers in physiology* **9**, 1002.
34. Radau P, Lu Y, Connelly K, Paul G, Dick A, Wright G. 2009 Evaluation Framework for Algorithms Segmenting Short Axis Cardiac MRI.. *Sunnybrook Hospital*.
35. Gao H, Li W, Cai L, Berry C, Luo XY. 2015 Parameter estimation in a Holzapfel–Ogden law for healthy myocardium. *Journal of engineering mathematics* **95**, 231–248.
36. Ten Brinke E, Klautz R, Verwey H, Van Der Wall E, Dion R, Steendijk P. 2010 Single-beat estimation of the left ventricular end-systolic pressure–volume relationship in patients with heart failure. *Acta physiologica* **198**, 37–46.
37. Gao H, Aderhold A, Mangion K, Luo XY, Husmeier D, Berry C. 2017 Changes and classification in myocardial contractile function in the left ventricle following acute myocardial infarction. *Journal of the Royal Society Interface* **14**, 20170203.
38. Forrester A, Sobester A, Keane A. 2008 *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons.
39. Liu Y, Sun F. 2011 A fast differential evolution algorithm using k-Nearest Neighbour predictor. *Expert Systems with Applications* **38**, 4254–4258.
40. Chen T, Guestrin C. 2016 Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* pp. 785–794. ACM.

41. Widrow B, Lehr MA. 1990 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. *Proceedings of the IEEE* **78**, 1415–1442.
42. Bishop CM. 2006 *Pattern recognition and machine learning*. springer.
43. Zheng A, Casari A. 2018 *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc.
44. Altman NS. 1992 An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* **46**, 175–185.
45. Cover T, Hart P. 1967 Nearest neighbor pattern classification. *IEEE transactions on information theory* **13**, 21–27.
46. Friedman JH. 2001 Greedy function approximation: a gradient boosting machine. *Annals of statistics* pp. 1189–1232.
47. Friedman J, Hastie T, Tibshirani R et al.. 2000 Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics* **28**, 337–407.
48. Breiman L, etc. 1984 *Classification and Regression Trees*. Wadsworth International Group,.
49. Hornik K, Stinchcombe M, White H. 1989 Multilayer feedforward networks are universal approximators. *Neural networks* **2**, 359–366.
50. Auer P, Burgsteiner H, Maass W. 2008 A learning rule for very simple universal approximators consisting of a single layer of perceptrons. *Neural networks* **21**, 786–795.
51. Maas AL, Hannun AY, Ng AY. 2013 Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml* vol. 30 p. 3.
52. Claesen M, De Moor B. 2015 Hyperparameter search in machine learning. *arXiv preprint arXiv:1502.02127*.
53. Kohavi R et al.. 1995 A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* vol. 14 pp. 1137–1145. Montreal, Canada.
54. Jin C, Jijun L, Bo Z. 2018 Inverse problems for PDEs: Models, computations and applications. *SCIENTIA SINICA Mathematica* **49**, 643.
55. Storn R, Price K. 1997 Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* **11**, 341–359.
56. Liu M, Liang L, Sun W. 2019 Estimation of in vivo constitutive parameters of the aortic wall using a machine learning approach. *Computer methods in applied mechanics and engineering* **347**, 201–217.