

# Artificial intelligence approaches to predict coronary stenosis severity using non-invasive fractional flow reserve

Proc IMechE Part H:  
J Engineering in Medicine  
1–14

© IMechE 2020



Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/0954411920946526

journals.sagepub.com/home/pih



Jason M Carson<sup>1,2,3</sup> , Neeraj Kavan Chakshu<sup>1</sup>, Igor Sazonov<sup>1</sup> and Perumal Nithiarasu<sup>1</sup>

## Abstract

Fractional flow reserve is the current reference standard in the assessment of the functional impact of a stenosis in coronary heart disease. In this study, three models of artificial intelligence of varying degrees of complexity were compared to fractional flow reserve measurements. The three models are the multivariate polynomial regression, which is a statistical method used primarily for correlation; the feed-forward neural network; and the long short-term memory, which is a type of recurrent neural network that is suited to modelling sequences. The models were initially trained using a virtual patient database that was generated from a validated one-dimensional physics-based model. The feed-forward neural network performed the best for all test cases considered, which were a single vessel case from a virtual patient database, a multi-vessel network from a virtual patient database, and 25 clinically invasive fractional flow reserve measurements from real patients. The feed-forward neural network model achieved around 99% diagnostic accuracy in both tests involving virtual patients, and a respectable 72% diagnostic accuracy when compared to the invasive fractional flow reserve measurements. The multivariate polynomial regression model performed well in the single vessel case, but struggled on network cases as the variation of input features was much larger. The long short-term memory performed well for the single vessel cases, but tended to have a bias towards a positive fractional flow reserve prediction for the virtual multi-vessel case, and for the patient cases. Overall, the feed-forward neural network shows promise in successfully predicting fractional flow reserve in real patients, and could be a viable option if trained using a large enough data set of real patients.

## Keywords

Artificial intelligence, computational mechanics, biomedical engineering, haemodynamic modelling, coronary heart disease, fractional flow reserve

Date received: 29 November 2019; accepted: 30 June 2020

## Introduction

Cardiovascular diseases (CVDs) are the world's major cause of mortality, being responsible for approximately 31% of deaths worldwide.<sup>1</sup> Coronary heart disease (CHD) is the biggest sub-category of CVD,<sup>2</sup> and is most commonly caused by atherosclerosis build-up on the inner layer of a coronary artery, which narrows the vessel lumen area. The current gold standard diagnostic tool for estimating CHD severity is the fractional flow reserve (FFR),<sup>3</sup> although other diagnosis measures such as the instantaneous wave-free ratio (iFR) have been proposed.<sup>4,5</sup> FFR is performed invasively during cardiac catheterisation. A pressure sensitive wire measures the pressure at the aorta and at a point distal to a

stenosis simultaneously. The pressure ratio of pressure distal to stenosis divided by aortic pressure is used to determine if the stenosis is flow limiting. If the pressure decrease is greater than 20%, which corresponds with a

<sup>1</sup>Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Swansea, UK

<sup>2</sup>Swansea University Medical School, Swansea University, Swansea, UK

<sup>3</sup>HDR UK Wales and Northern Ireland, Health Data Research UK, London, UK

### Corresponding author:

Perumal Nithiarasu, Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Bay Campus, Engineering Central, Swansea SA1 8EN, UK.

Email: p.nithiarasu@swansea.ac.uk

FFR value below 0.8, the patient will normally be required to undergo further surgical treatment such as an angioplasty. As the FFR procedure is invasive and expensive, there have recently been attempts at determining the FFR non-invasively through the use of coronary computed tomography angiography (CCTA) and physics-based computational fluid dynamics (CFD) models.<sup>6–8</sup> These models have been shown to give respectable agreement with the invasive clinical measures.

In recent years, there has been renewed interest in artificial intelligence (AI) with many areas now focusing on data science techniques to find correlations and predictions using data. This is partly due to two main developments: (1) the accessibility to much greater computational resources, and (2) the wider availability of large amounts of data that are required for training AI models. Some of these techniques are making significant inroads into medical research. Examples include AI algorithms being developed to detect cancer and determine a prognosis,<sup>9</sup> and to manage and support treatment of diabetes.<sup>10</sup> An AI model has even been shown to be more reliable at detecting brain tumours than the current techniques used in radiography.<sup>11</sup> Now, AI models are being applied to FFR, which ranges from replacing one-dimensional (1D) models with machine learning (ML) models,<sup>12</sup> to using the CCTA images and deep learning models to estimate the FFR directly from the CT images.<sup>13,14</sup> While AI offers an excellent opportunity to compute FFR faster than the majority of physics-based models, assessing the accuracy and robustness remains a real challenge.

In the present work, we compare three ML models of varying degrees of complexity to further understand the applicability of AI in the FFR calculations. They are (1) multivariate polynomial regression (MPR), (2) feed-forward neural network (FFNN), and (3) a long short-term memory (LSTM) model. The ML models are first trained using a virtual patient database created using a validated 1D physics-based model,<sup>15</sup> then the ML models are compared with clinically invasive FFR measurements on a small cohort of 25 patients.

### **ML and deep learning**

The foundation of current ML and deep learning algorithms originated from work in 1943 by McCulloch and Pitts.<sup>16</sup> They proposed the idea and theory of a neuron, what it is and how it works, and created an electrical circuit of the model, thus creating the first neural network. In 1950, Turing<sup>17</sup> published the seminal paper ‘Computing Machinery and Intelligence’ that discussed the theoretical and philosophical ideas of AI. From this work, the Turing test was born. The normal interpretation of the Turing test is to have an interrogator attempting to distinguish between two ‘players’, one of which is human and the other is a computer. The development of ML and deep learning algorithms merely use the human nervous system, and in particular

the brain, as a reference for inspiration in the development of these algorithms.

There has been a recent resurgence in the use of neural networks, primarily due to large amounts of data, and easier access to powerful computational resources. However, some of the fundamental mathematics used in AI are well established. For example, the gradient descent optimisation method, which forms the foundation of the back propagation step while training the model for many projects, was originally proposed by Cauchy<sup>18</sup> in 1847. Although at this stage gradient descent was used for minimisation problems on a system of simultaneous equations, it was not until the 1960s that gradient descent was used in the context of multi-stage, non-linear systems.<sup>19,20</sup> In 1982, PJ Werbos<sup>21</sup> described the use of gradient descent in a neural network. The original gradient descent method is rarely used nowadays, but the improved optimisation methods of momentum<sup>22</sup> and ADAM,<sup>23</sup> essentially extend the gradient descent method and are widely used.

## **Methodology**

### ***Virtual patient generation and feature extraction from CT data***

It is difficult to obtain a significant number of clinical patient data for training an ML model and hence we created virtual patients. There are two different types of virtual patient generation. The first considers a patient as a single vessel and randomises the vessel area profile, length, and flow rate through the vessel. These single vessel cases are used to train the ML models. The second case involves creating a network of the left coronary artery (LCA) branch that consists of nine main vessels.

**Single vessel patient generation.** The single vessel cases are generated by first randomising the proximal and distal area of the vessel independently in the range 0.0059 to 0.8131 cm<sup>2</sup>. The vessel length is also chosen at random with the range 0.0957 to 11.1475 cm. These ranges for the vessel area and length were chosen as they represent the spread of areas and lengths observed in the real patient cohort. Initially the area profile is constructed by linearly tapering between the proximal and distal areas. A mesh is then created with an element size of  $\Delta x = 0.01$  cm; however, this is adapted by adding more randomisation into the area profile after the creation of the stenosis. This forms the foundation of the vessel geometry that is then adapted to include a stenosis.

The characteristics of the stenosis are also partially randomised. The stenosis is assumed to be located in the middle of the vessel. The severity of the stenosis is randomised between a 0% decrease in diameter (no stenosis), to a 80% decrease in vessel diameter. The largest decrease in vessel radius is chosen as 80% as clinicians generally do not perform the invasive FFR

measurement for any blockage worse than this as the risk of perforating an artery would increase. In the clinic, it is assumed these patients would need surgical intervention. The length of the stenosis is assumed to vary as a ratio of between 0% (no stenosis) and 60% of the total length of that vessel. This will cover the cases where no stenosis is present through to more diffuse stenosis.

The stenosis is constructed using the following equation

$$D = D_0 + \frac{(D_A - D_0)}{2} \left( 1 - \cos\left(\frac{x}{2\pi L_a}\right) \right) \quad (1)$$

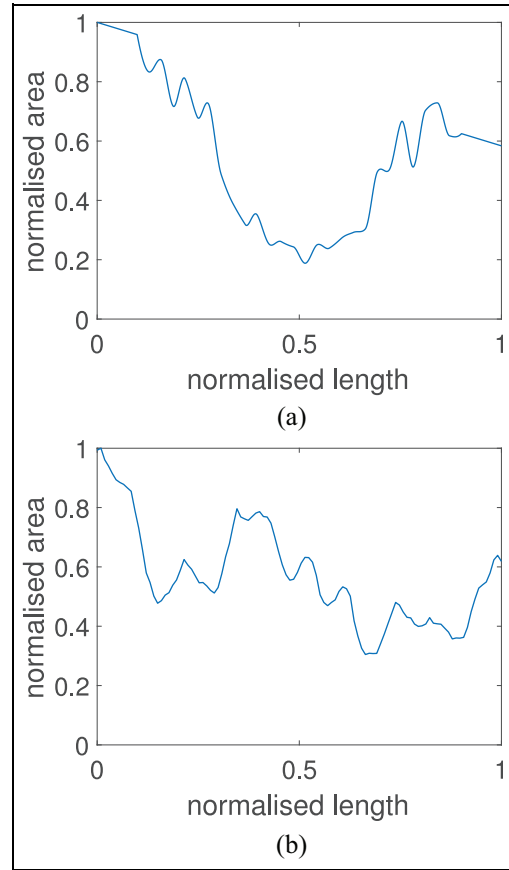
where  $D_0$  is the reference diameter across the entire stenosis,  $D_A$  is the stenosed area,  $x$  is the axial coordinate along the stenosis, and  $L_a$  is the length of the stenosis. The stenosis is then inserted into the middle of the vessel.

At this stage, the stenosed vessel undergoes an additional randomisation procedure to ensure that the vessel geometry is not too smooth. Across the length of the vessel, the first and last 10% will remain unchanged in the randomisation procedure, only the internal 80% is affected. This internal region is further split by choosing specific nodes using the equation

$$N_c = \max(5, \text{floor}(N/20)) \quad (2)$$

where  $N$  is the number of nodes in the stenosed region and  $N_c$  is the number of nodes that are to be adapted. The minimum number of area nodes chosen to perturb is 5. At these chosen  $N_c$  nodes, the areas are then randomly varied between a 20% decrease and a 40% increase. Then, a shape-preserving piecewise cubic Hermite interpolating polynomial is utilised to interpolate all areas at the nodal points located between the  $N_c$  nodes. This generates an area profile across the vessel with similar attributes to those seen in real patients as shown in Figure 1. This procedure will influence the severity of the stenosis and hence the stenosis area and mean vessel area are extracted from the new area profile. The final feature that is required for the ML models is the volumetric flow rate through the vessel. The volumetric flow was randomised in the range 0.3896 and 11.1475 cm<sup>3</sup>/s. All randomisation procedures for all parameters were performed using a uniform distribution within their respective ranges.

**Virtual patient network generation.** The virtual patient network uses the left coronary network branch consisting of nine vessels, presented in Mynard and Smolich,<sup>24</sup> as the foundation for which vessel area and length adaptations are performed. On top of this foundation, a similar randomisation procedure is performed to that of the single vessel case. First, in the procedure is that the length, and the area at the start and end of every vessel are all randomly adapted by increasing or decreasing



**Figure 1.** Normalised area profile of a generated patient and a normalised area profile of the left anterior descending artery extracted from a patient CT scan: (a) virtual patient area profile; (b) real patient area profile.

around their reference state by up to 70%. In the network, the vessel in which the stenosis is to be located is also chosen at random. At this stage, the stenosed vessel undergoes the same procedure as that described for the single vessel case. The remaining vessels do not have a stenosis added; however, the remaining parts of the area randomisation procedure are still performed.

**Feature extraction from CT images.** In order to extract the features from the patient CT images, the images first need to be segmented. The segmentation and extraction of centreline information is performed using the image segmentation software VMTKLab (Orobix, Italy). The area profile and vessel length information can then be extracted and can be used to find the input features for two of the ML models, which include the area at the start and end of each vessel, the mean vessel area, and the vessel length. The volumetric flow rate at the inlet of the LCA is assumed to be 425 mL/min, and the flow distribution in the network is calculated via a power law which uses Murray's law with an exponent of 2.78.<sup>8</sup>

## ML

Many of the concepts and theory used in ML were previously developed in other areas of mathematical

sciences, such as using statistics to find the likelihood of an event occurring or finding relationships and correlations in various types of data. One such method is MPR, which is a generalised form of linear regression.

**MPR.** MPR is the least complex model implemented in this work and utilises linear regression with a high-order polynomial feature space on multiple variables. Although polynomial regression is quite simple, it can still provide useful and powerful predictions. However, one must be careful when creating the training and test data for these models as polynomial regression tends to perform reliably well when used to interpolate data, but can produce erroneous predictions when attempting to extrapolate data. This issue is often exacerbated when higher-order polynomials are used for the feature space. In order to describe this method, it is advantageous to first describe univariate polynomial regression, and then discuss its extension to the multivariate case.

Univariate polynomial regression for a dependant variable  $y$  and an independent variable  $x$ , the  $i$ th example while using a  $n$ th – order polynomial feature space can be written in the form

$$y_i = b + w_1x_i + w_2x_i^2 + w_3x_i^3 + \dots + w_nx_i^n \quad (3)$$

where  $b$  is referred to as the bias, and  $w_j$  (for  $j = 1 : n$ ) are referred to as the weights. Using enough training cases of known outputs  $y_i$  for some given input features  $x_i$ , the system of equations can be solved to find the optimum values of the bias and weights to best fit the data. These bias and weights can then be used for future predictions. This forms the foundation of polynomial regression, and its extension to the multivariate case is relatively straightforward. The main change from the univariate to multivariate case is that there are now more possible combinations of polynomial terms for the feature space, for example, in a case with two independent variables ( $x_1$  and  $x_2$ ) up to a second – order polynomial feature space, the general form will be

$$y_i = b + w_1x_{1,i} + w_2x_{1,i}^2 + w_3x_{2,i} + w_4x_{2,i}^2 + w_5x_{1,i} \cdot x_{2,i} \quad (4)$$

which contains the multiplication of the two independent variables to create the new feature  $x_{1,i} \cdot x_{2,i}$ . The general matrix form of MPR can be expressed as

$$Y = XW \quad (5)$$

where  $Y$  is the column vector

$$Y = [Y_1 \ Y_2 \ \dots \ Y_m]^T \quad (6)$$

containing all outputs/predictions, where  $m$  is the number of examples,  $X$  is a matrix containing all input

features that includes different combinations of the independent variables and has the form

$$X = \begin{bmatrix} 1 & X_{1,1} & X_{1,2} & \dots & X_{1,n} \\ 1 & X_{2,1} & X_{2,2} & \dots & X_{2,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & X_{m,1} & X_{m,2} & \dots & X_{m,n} \end{bmatrix} \quad (7)$$

where  $n$  is the number of possible feature combinations, and  $W$  is a vector that contains all of the bias and weight terms

$$W = [b \ W_1 \ \dots \ W_n]^T \quad (8)$$

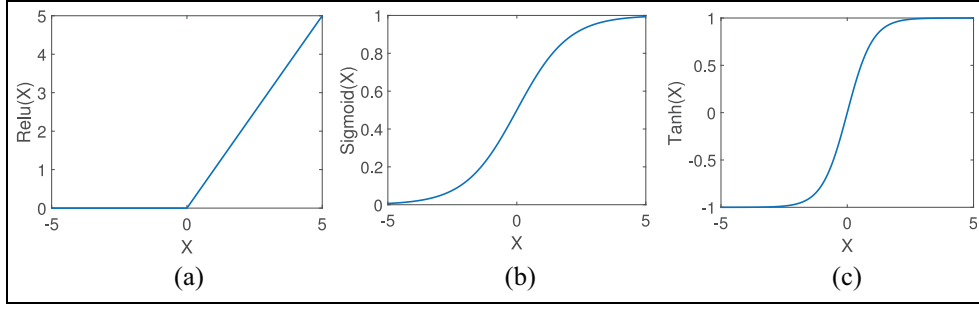
In the present article, the least-squares method is used to solve for the bias and weight terms for polynomial regression. Six input features are used which includes the area at the start and end of a vessel, the minimum area in the vessel, the mean area of the vessel, the estimated flow rate in the vessel, and the length of the vessel.

### Deep learning

In biomedical engineering, the more basic statistical techniques such as polynomial regression have generally fallen out of favour. These have largely been replaced by artificial neural networks (ANNs), which are considered as more powerful alternatives when it comes to finding correlations in complex real-world data. The term ANN was used as they are inspired by, and try to resemble the human nervous system, and particularly the human brain. The human brain is composed of many interconnected neurons which transmit information in the form of electrical impulses. Analogously, ANNs are composed of layers of interconnected ‘neurons’ which pass information to and from other neurons.

The ‘depth’ of a neural network generally refers to the number of hidden layers that are present in an ANN, and in general increasing the number layers in an ANN allows a greater degree of non-linear features to be captured. There are many types of deep learning architectures, each with different strengths and weaknesses. In addition, there are three main paradigms of deep learning which are supervised learning algorithms, which seek to find a relationship between input features of training data to their known outputs; unsupervised deep learning algorithms, which seek to find patterns in data without knowing any result or outputs; and reinforcement learning, which is a goal-oriented algorithm that seeks to learn the best possible action to maximise its ‘rewards’ for a particular situation, and thus learns from its experience.

The two types of supervised deep learning models utilised in the present article are an FFNN, and an LSTM model which is a type of recurrent neural network (RNN).



**Figure 2.** Comparison of the most common non-linear activation functions for FFNN networks: (a) relu activation function; (b) logistic (sigmoid) activation function; and (c) tanh activation function.

*Non-linear activation functions.* The activation functions utilised in this article (including those used in the LSTM) are shown in Figure 2 and are the rectified linear unit (relu) function which is given by

$$A(z) = \max(0, z) \quad (9)$$

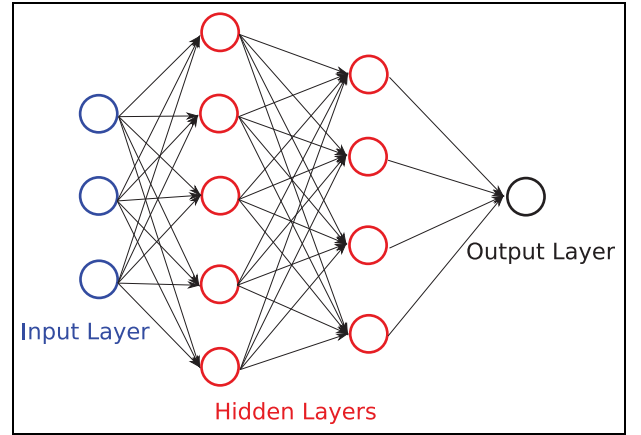
and is most often used for regression problems, has an output range of  $[0 \infty]$ , and is shown in Figure 2(a); the logistic (sigmoid) function has the output range of  $[0 1]$ , is shown in Figure 2(b) and can be expressed as

$$A(z) = \frac{1}{1 + \exp(-z)} \quad (10)$$

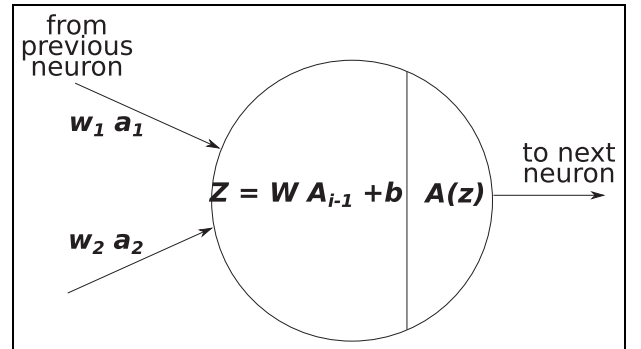
The logistic function is more often used in classification problems; the final activation function used in this article is tanh which also tends to be used in classification problems, has an output in the range  $[-1 1]$ , is shown in Figure 2(c) and can be written as

$$A(z) = \frac{\exp(z) - \exp(-z)}{\exp(z) + \exp(-z)} \quad (11)$$

*FFNN.* FFNNs consist of an input layer, and output layer, and at least one hidden layer. FFNN is a supervised learning technique that utilises non-linear activation functions at each layer (with the exception of the input layer), in order to capture non-linear relationships in the data. FFNNs are referred to as vanilla neural networks when they have only one hidden layer. A diagram of a FFNN network is shown in Figure 3, where each arrow shown in the diagram represents a linear mapping, followed by a non-linear activation function as indicated in Figure 4. Training an FFNN typically involves four main steps (steps are expanded upon in the next three paragraphs): (1) forward propagation which computes the predicted output value from an initial estimation of weights and biases. The direction of computation is from the input layer through to the output layer (hence it travels forward); (2) computing the cost function (error) of the predicted output to the expected output; (3) backward propagation which



**Figure 3.** Example of a FFNN with an input layer of three features, two hidden layers of five and four neurons, respectively, and an output layer with a single neuron. Bias terms are neglected in this diagram to reduce complexity.



**Figure 4.** Example of an input and output of a single neuron during forward propagation, where  $A_{i-1}$  is a matrix of all outputs  $a_j$  from the previous layer.

computes the derivatives of the cost function with respect to the weights and biases. The direction of computation is from the output layer through towards the input layer (hence it travels backward); (4) finally, the derivatives that were calculated from backward propagation are used to update the bias and weight values.

**FFNN training.** The FFNN training process begins with the initialisation stage which includes randomising the weights and biases, and giving the algorithm the training set of input features. The next step is forward propagation which involves a linear mapping for outputs from neurons in the previous layer to neurons in the next layer as

$$z_{i,j} = w_{i,k}A_{k,j} + b_i \quad (12)$$

where  $A$  is a matrix containing all values from the neurons in the previous layer,  $i = 1 : nl$  where  $nl$  is the number of neurons in the next layer,  $j = 1 : n$  where  $n$  is the number of training examples, and  $k = 1 : m$  where  $m$  is the number of features in the previous layer. The linear mapping is then followed by a non-linear activation function  $A(z)$  as shown in Figure 4. In an FFNN, the use of non-linear activation functions allows the network to capture more complex relationships in the data, in fact FFNNs are regarded as universal function approximators as, in theory, they can represent every possible computable function, although the number of neurons and layers required to do this are generally not known. The output of these activation functions is then used as the input for the next layer in the neural network. This essentially means that every additional hidden layer folds additional non-linearities into the prediction.

The general initialisation and forward propagation algorithm for the FFNN utilised in the present work is as follows:

1. define input features and select the training set,
2. initialise weights and biases for each layer,
3. linear mapping from input features to hidden layer 1,
4. non-linear activation function (relu),
5. linear mapping from hidden layer 1 to hidden layer 2,
6. non-linear activation function (tanh),
7. linear mapping from hidden layer 2 to hidden layer 3,
8. non-linear activation function (relu),
9. linear mapping from hidden layer 3 to the output layer,
10. non-linear activation function (relu) for FFR prediction.

After the forward propagation is complete, a cost function is used to calculate the error in the prediction compared to the actual values. The most common type of cost function for regression problems is the mean square error

$$\text{Cost}_{mse} = \frac{1}{n} \sum_{j=1}^n (\hat{ffr} - ffr)^2 \quad (13)$$

where  $\hat{ffr}$  is the FFR predicted by the FFNN algorithm and  $ffr$  is the real FFR value (physics-based model).

The next step is to perform backward propagation, the first part requires the calculation of the derivative of the cost function with respect to the predicted values  $\hat{ffr}$ . Differentiation and the chain rule is then used to find the derivatives of the cost function with respect to all weight and bias terms for each layer of the network. For example, the derivative of the cost function with respect to a weight in the last hidden layer can be calculated as

$$\frac{\partial \text{Cost}}{\partial w} = \frac{\partial \text{Cost}}{\partial A} \frac{\partial A}{\partial z} \frac{\partial z}{\partial w} \quad (14)$$

The general implementation of the backward propagation algorithm for the FFNN is as follows:

1. calculate the derivatives of the cost function w.r.t. the weights and biases for the output layer,
2. calculate the derivatives of the cost function w.r.t. the weights and biases for hidden layer 3,
3. calculate the derivatives of the cost function w.r.t. the weights and biases for hidden layer 2,
4. calculate the derivatives of the cost function w.r.t. the weights and biases for hidden layer 1.

The weights and biases can then be updated for all layers via an optimisation algorithm such as gradient descent or ADAMs optimisation. For the FFNN model, the ADAMs optimisation<sup>23</sup> was used to find a local minima.

In order to predict FFR values from new input features, forward propagation can be performed with the final converged weights and biases.

**FFNN model description.** The FFNN model implemented in the present article uses an input layer of six features that includes the area at the start and end of the vessel, the minimum area in a vessel, the mean area of a vessel, the estimated flow rate in the vessel, and the length of the vessel. There are three hidden layers, the first hidden layer contains 64 neurons and uses the relu function, the second hidden layer contains 32 neurons and uses the tanh function, and the third hidden layer also uses 32 neurons but uses the relu function; the output layer gives a single FFR output value for the end of the vessel and uses the relu function. The architecture was chosen from iteratively trialling different combinations of the number of neurons in a layer, the number of layers, and the type of activation function used, all using grid searching. It is known that for many problems the relu function increases the speed of convergence; however, it was observed that model training performed better with at least one layer of the tanh function. The model did not require any additional techniques to prevent over-fitting to the data as the training and test accuracy were both close to 99%. The ADAMs optimisation algorithm is used to update the model weights and biases. The learning rate used was

$\alpha = 1 \times 10^{-4}$  for the first 1000 iterations of ADAMs optimiser, and then the learning rate was reduced to  $\alpha = 1 \times 10^{-6}$ . The parameters of the ADAMs optimiser were the commonly used  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 1 \times 10^8$ .

**LSTM.** In the medical arena, there are many diagnostic and monitoring systems that take time dependant measurements and can be viewed as sequence data. RNNs are particularly convenient for these problems as they can exhibit dynamic temporal behaviour and can have sequence data as a model input and as an output. An RNN cell contains a closed-loop which allows the output of the current step to be influenced by the output of the previous step. In theory, RNNs are able to use previous values in the sequence to aid in predicting a future value (time dependencies) and they tend to perform quite well when the distance between a point and a dependent value is quite small; however, in practice, they often struggle with longer-term dependencies.<sup>25,26</sup> This shortcoming was resolved with the development of the LSTM algorithm.<sup>27</sup> LSTMs have a different structure for a module, which instead of having a single layer that contains a loop, the module now contains four structured layers with a loop, although several LSTM variations exist. Fundamental to the structure of an LSTM model is the idea of gates, which includes an input gate, a forget gate, and an output gate. An important characteristic of LSTM is in its ability to add or remove information from the sequential inputs given to it, which allows it to retain useful information and remove inessential parts automatically. Each LSTM ‘cell’ contains two states in parallel, an internal cell state  $C_i$  that is not seen, and an output cell state  $Out_i$  which is the output value seen. There are different types of LSTM architectures, thus the one described here is the version implemented in the present work. The general forward pass of an LSTM module is shown in Figure 5 and is as follows:

1. initialise weights, biases, cell states, and output states,

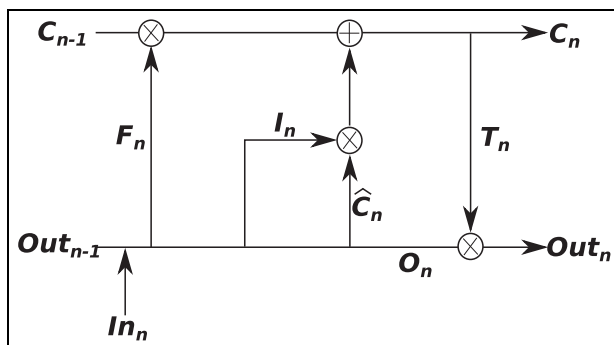


Figure 5. Overview of an LSTM module.

2. forget gate has inputs  $Out_{n-1}$  and  $In_n$ , and utilises a hard sigmoid function (output of  $\text{sigmoid}_h$  is either 0 or 1)

$$F_n = \text{sigmoid}_h(W_{f,O} \cdot Out_{n-1} + W_{f,I} \cdot In_n + b_f) \quad (15)$$

3. input gate and cell state update

$$\begin{aligned} I_n &= \text{sigmoid}(W_{i,O} \cdot Out_{n-1} + W_{i,I} \cdot In_n + b_i) \\ \hat{C}_n &= \tanh(W_{u,O} \cdot Out_{n-1} + W_{u,I} \cdot In_n + b_u) \\ C_n &= F_n \cdot C_{n-1} + I_n \cdot \hat{C}_n \end{aligned} \quad (16)$$

4. output gate and output vector

$$\begin{aligned} O_n &= \text{sigmoid}(W_{o,O} \cdot Out_{n-1} + W_{o,I} \cdot In_n + b_o) \\ T_n &= \tanh(C_n) \\ Out_n &= O_n \cdot T_n \end{aligned} \quad (17)$$

The output state and cell state at step  $n$  are then inputs to step  $n - 1$ . For a more exhaustive explanation of the LSTM, the reader is referred to Hochreiter and Schmidhuber.<sup>27</sup>

In this article, the stacked LSTM model is made up of five LSTM layers all with 32 modules, followed by five fully connected (dense) layers with 32, 32, 16, 8, and 1 neurons, respectively. The learning rate used was  $\alpha = 1 \times 10^{-4}$  and ADAMs optimiser was used with  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 1 \times 10^8$ . A dropout rate of 0.4 was utilised in the dense layers to prevent the model from over-fitting the data. As with the FFNN, the LSTM model used grid searching when deciding upon the type of architecture, and the hyper-parameters used. The input features for the LSTM model is the entire vessel area profile with a length step size of 0.01 cm between area positions. As coronary vessel lengths will vary, the input vector that describes the vessel area profile would also vary in size. Thus, in order to train the LSTM, which requires the model input to be of the same length, zero-padding was applied at the end of the input vector for model training.

### Computational mechanics

One of the main limitations that can hinder ML and deep learning model development is the ability to collect an adequate amount of real-world data for training the model. This is particularly the case in biomedical engineering where there are many issues that impact the ability to collect real patient data, such as the number of patients that have a particular health problem, the type of data that is collected in the medical clinic may not be exactly what is needed to correctly model the problem, some data may be missing (some patients may not have pressures recorded while others do), and

the quality of the data itself is affected by the accuracy of the measurement device or the quality of the imaging data. Thus, in the present article, we utilise a 1D haemodynamic model that has been previously validated against invasive FFR measurements<sup>8,15</sup> in order to generate virtual FFR patients. These virtual patients are then used to train the three AI models.

The 1D model of blood flow utilised in the present article is described in Carson and colleagues<sup>15,28</sup> and contains two tiers: (1) the closed-loop model that is described in Carson and colleagues<sup>28,29</sup> is used to generate the flow rate waveforms for the inlet of the left coronary and right coronary arteries; (2) an open-loop coronary network (the coronary network in Mynard and colleagues<sup>24,30</sup>) is then used, where the vessel length and areas are adapted at random to produce similar vessel area variations to what is seen in real patient data, in addition a stenosis is added to one of the vessels in the network using a random blockage percentage, stenosis length, and stenosis location.

The 1D model of blood flow in a compliant vessel is governed by the continuity equation

$$C \frac{\partial P}{\partial t} + \frac{\partial Q}{\partial x} = 0 \quad (18)$$

where  $C = (\partial A / \partial P)$  is the vessel compliance,  $P$  is the hydrostatic pressure,  $t$  is the time,  $Q$  is the volumetric flow rate, and  $x$  is the axial coordinate; and the conservation of linear momentum

$$\frac{\rho}{A} \frac{\partial Q}{\partial t} + \frac{\rho}{A} \frac{\partial \left( \frac{Q^2}{A} \right)}{\partial t} + \frac{\partial P}{\partial x} + \frac{\xi \mu \pi Q}{A^2} = 0 \quad (19)$$

where  $A$  is the area,  $\xi = -22$  is a viscous friction coefficient that corresponds to a relatively blunt velocity profile,  $\mu = 0.04P$  is the blood viscosity, and  $\rho = 1.06$  g/cm is the blood density. A visco-elastic constitutive law is chosen for the vessel lumen area to blood pressure relationship and consists of a power law model for the elastic term and a Voigt model for the viscous term<sup>8</sup> and has the form

$$P - P_0 = \frac{2\rho c_0^2}{b} \left( \left( \frac{A}{A_0} \right)^{b/2} - 1 \right) + \frac{\Gamma}{A_0 \sqrt{A_0}} \frac{\partial A}{\partial t} \quad (20)$$

$$b = \frac{2\rho c_0^2}{P_0 - P_{collapse}}$$

where  $P_0 = 80$  mmHg is the reference pressure,  $P_{collapse} = 10$  mmHg is a collapsing pressure,  $A_0$  is the cross-sectional area at the reference pressure, and  $c_0$  is the reference wave speed that is determined to be

$$c_0 = \sqrt{\frac{2}{3\rho} (k_1 \exp(k_2 r_0) + k_3)} \quad (21)$$

where  $k_1 = 2.00 \times 10^7$  g<sup>2</sup>/cm/s,  $k_2 = -22.53$  cm<sup>-1</sup>,  $k_3 = 8.65 \times 10^5$  g<sup>2</sup>/cm/s, and  $r_0$  is the radius of the vessel. The viscous wall coefficient can be expressed as

$$\Gamma = \frac{100}{2r_0} + 100 \quad (22)$$

To be consistent with the ML models, conservation of static pressure is assumed at the junction of vessels. The system of non-linear equations is solved using an implicit sub-domain collocation scheme.<sup>31</sup>

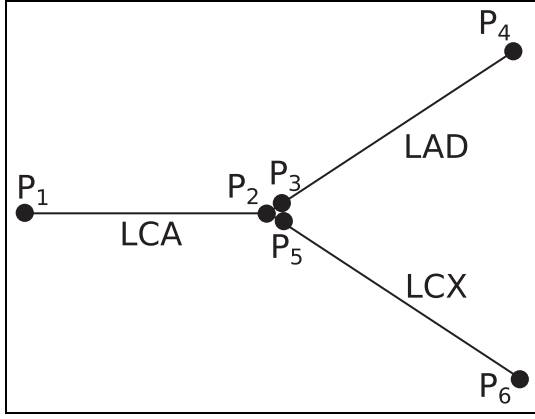
### Constructing an ML model of FFR on a coronary network

In the present work, two separate model constructions are proposed, respectively, using the pressure drop and the FFR value. The MPR and FFNN are trained using a single vessel model for the pressure drop (rather than FFR value), and the RNN (long-short term memory) model is trained on a single vessel model for the FFR value. This is performed for the following two reasons:

- There are large variations in patient coronary network geometry that includes different vessel sizes (lengths and areas), different vessel connectivities, and the inclusion or exclusion of certain vessels. The MPR and FFNN models only use six input features; thus, all variations in patient geometry need to be covered by these six features. Through tests on the 1D model,<sup>15</sup> the pressure drop across the stenosis is invariant to the aortic pressure, while the FFR value intimately depends on the aortic pressure. This means that the pressure drop is more reliable and easier to utilise for training these models, as the number of input features is low.
- the LSTM model is ideal for sequences and thus it is a natural choice for a network in which the solution in the next vessel will depend on the solution of the previous vessel. However, as all input sequences (vectors) must have the same length, post sequence zero-padding is performed for the area profile.

In order to describe the construction of the FFR solution in a network for the MPR and FFNN models, we first make the assumption (validated through numerical tests) that the pressure drop in a vessel is not influenced by the hydrostatic pressure at the start of the vessel. Second, we assume continuity of static pressure at vessel junctions (although an ML model could also be used to estimate the change in pressure between vessels). These two assumptions allow the use of a simple reconstruction technique to determine the FFR





**Figure 6.** Three-vessel configuration consisting of an LCA, LAD, and LCX. For the MPR and FFNN models, the pressure drop is calculated across each vessel separately and then the FFR values are reconstructed afterwards.

value from the aorta to the location of FFR measurements downstream of a stenosis. Thus, the MPR and FFNN models estimate the pressure drop over each vessel in a network which is reconstructed in the following way: consider a bifurcation of the LCA, left anterior descending (LAD) artery, and left circumflex (LCX) artery as shown in Figure 6.

There are six pressure ‘nodes’ representing the pressure at the start and end of the three vessels. For continuity of static pressure at vessel junctions, the pressure at the end of the LCA will be equal to the pressure at the start of both the LAD and LCX vessels. Thus, the FFR value at the end of the LAD and LCX can be reconstructed from the pressure drops ( $\Delta P$ ) of each of the three vessels (provided the mean aortic pressure is known, which can be measured and estimated non-invasively using the brachial artery cuff pressure) in the following way

$$\begin{aligned} \Delta P_{LCA} &= P_1 - P_2, & \Delta P_{LAD} &= P_3 - P_4 \\ \Delta P_{LCX} &= P_5 - P_6, & P_2 &= P_3 = P_5 \\ FFR_{LAD} &= \frac{P_1 - \Delta P_{LCA} - \Delta P_{LAD}}{P_1} \\ FFR_{LCX} &= \frac{P_1 - \Delta P_{LCA} - \Delta P_{LCX}}{P_1} \end{aligned} \quad (23)$$

This technique can be extended in a straightforward way to any size of vessel network, provided a good estimation of the mean aortic pressure is known. In the present work, we estimate that the mean aortic pressure is 93.333 mmHg, as we do not have pressure measurements from any patients. This is also the mean pressure of the computational model.

## Results

In order to test and compare the proposed methodologies, three test cases are constructed. The first test case

uses a virtual patient database of single vessels. In reality, the FFR measurement does not take place at points just either side of the stenosis, but instead the proximal pressure measurement occurs in the aorta near where the coronary arteries branch out from, while the measurement point distal occurs after a stenosis; thus, the full FFR ratio is affected by what happens over a vessel network and not over a single vessel. Thus, we test our reconstruction methodology on the second test case, which consists of a network of virtual patients where the FFR solution is determined by the 1D CFD model, and in the third test we evaluate the methodology on a small cohort of real patients where the FFR has been measured invasively during coronary catheterisation. In our results, the mean absolute error (MAE) that is calculated as

$$MAE = \frac{\sum_{i=1}^n |FFR_{pred,i} - FFR_{true,i}|}{n} \quad (24)$$

where  $n$  is the number of cases,  $i$  is the current sample item,  $FFR_{pred}$  is the model predictions, and  $FFR_{true}$  are the true values that are either computational model, or invasive FFR value, respectively. The standard deviation is calculated as

$$STD = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (FFR_i - \bar{FFR})^2} \quad (25)$$

where FFR is either from the ML predicted models, CFD model, or invasive FFR measurements, respectively, and  $\bar{FFR}$  is the mean of these samples. The bias is calculated as

$$Bias = \frac{\sum_{i=1}^n FFR_{true,i} - FFR_{pred,i}}{n} \quad (26)$$

### Results from single vessel

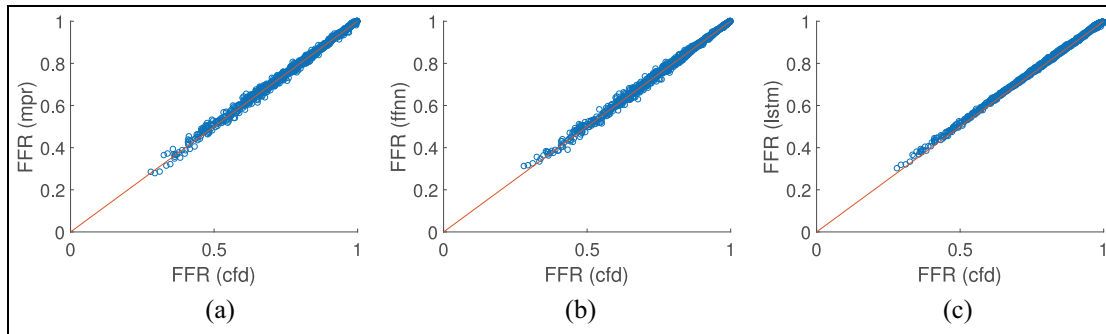
The first test case uses a virtual patient database of single vessels using the 1D blood flow model where the virtual patients were created by randomising the vessel area profile, stenosis location, stenosis length, stenosis severity, and the mean inflow rate. A total of 10,000 single vessel ‘virtual patients’ were generated with 70% used for model training and 30% used for testing. Figure 7 shows the results of the test set for all proposed methodologies, and Table 1 shows the overall performance of each methodology. The diagnostic accuracy is consistent between all of the models where they all achieve just under 99% accuracy. The sensitivity and specificity are closer in value for the MPR. For the FFNN model, the specificity is higher than the sensitivity, while for the LSTM model this is exacerbated further with the sensitivity 95.76% while the specificity

**Table 1.** Overview of the test set for each machine learning model compared to the 1D physics-based model on a single vessel model.

Method	Diagnostic accuracy (%)	Sensitivity (%)	Specificity (%)	p-value	$\rho$	MAE	STD	Bias
MPR	98.71	98.06	98.95	0	0.9984	0.0048	0.1360	0.0001
FFNN	98.88	97.57	99.36	0	0.9986	0.0042	0.1361	0.0000
LSTM	98.88	95.76	100.00	0	0.9993	0.0055	0.1330	-0.0038

MPR: multivariate polynomial regression; FFNN: feed-forward neural network; LSTM: long short-term memory.

Diagnostic measures include the percentages of the diagnostic accuracy, sensitivity, and specificity; and the linear correlation  $\rho$ , p-value, the mean absolute error (MAE), standard deviation (STD), and the bias (4 dp).

**Figure 7.** Comparison of machine learning/deep learning methods against the CFD model results on a single vessel: (a) MPR results on test set; (b) FFNN results on test set; and (c) LSTM results on test set.**Table 2.** Comparison of the MPR, FFNN, LSTM models with the 1D physics-based model on a nine-vessel network.

Method	Diagnostic accuracy (%)	Sensitivity (%)	Specificity (%)	p-value	$\rho$	MAE	STD	Bias
MPR	97.57	96.86	97.76	0	0.2430	0.0312	0.5406	0.0057
FFNN	99.08	97.92	99.39	0	0.9959	0.0050	0.1402	0.0000
LSTM	96.48	92.61	97.52	0	0.9793	0.0213	0.1456	0.0072

MPR: multivariate polynomial regression; FFNN: feed-forward neural network; LSTM: long short-term memory.

Diagnostic measures include the percentages of the diagnostic accuracy, sensitivity, and specificity; and the linear correlation  $\rho$ , p-value, the mean absolute error (MAE), standard deviation (STD), and bias (4 dp).

is 100%. All methodologies show a very high linear correlation value and p-value of 0. Although the results are very consistent between the three models, the FFNN performs the best on this test case as it has the equal highest diagnostic accuracy with the lowest MAE when compared to the 1D physics-based model. The standard deviation of the test set used in this example was 0.1365, which is quite consistent with the standard deviations observed of the predictive models from Table 1.

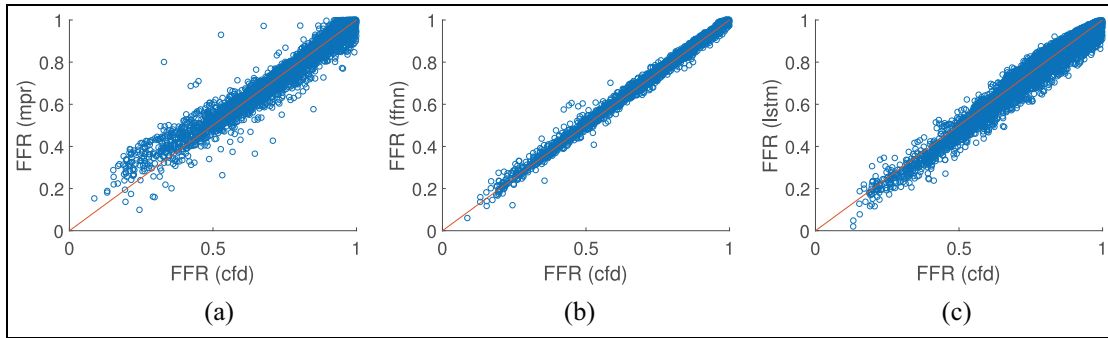
### Multi-vessel FFR

The second test case utilises a virtual patient database of LCA vessel networks. The left side of the coronary network proposed by Mynard and Smolich<sup>24</sup> is used as a basis, while the area profiles, vessel lengths, stenosis location, length, and severity, and flow rate distribution are varied randomly. A cohort of 10,000 virtual patient

coronary artery networks were generated, each network contained nine vessels, with 70% used for model training and 30% used for testing for all ML models. As shown in Table 2 and Figure 8, the FFNN model performs the best with a diagnostic accuracy of over 99% and also had the lowest mean absolute difference and highest linear correlation value. The LSTM, which is the most complex model implemented in this work, had the lowest diagnostic accuracy; however, it still had a lower mean absolute difference from the 1D physics-based model than the MPR model. The standard deviation of the underlying data in synthetic multi-vessel case was 0.1408, which is consistent with the values seen for the FFNN and LSTM in Table 2.

### Comparison with invasive FFR

The final test case compares the proposed methodologies and the 1D physics-based model to clinical invasive



**Figure 8.** Comparison of machine learning/deep learning methods against the CFD model results on coronary artery vessel networks: (a). MPR results on test set; (b) FFNN results on test set; and (c) LSTM results on test set.

**Table 3.** Comparison of the MPR, FFNN, LSTM, and ID physics-based model with invasive clinical measurements in the patient-specific networks.

Method	Diagnostic accuracy (%)	Sensitivity (%)	Specificity (%)	p-value	$\rho$	MAE	STD	Bias
MPR	72.00	90.00	60.00	$5.4 \times 10^{-3}$	0.5605	0.1363	0.1836	0.1026
FFNN	72.00	90.00	60.00	$3.7 \times 10^{-6}$	0.8048	0.0855	0.1165	0.0623
LSTM	64.00	100.00	40.00	$3.8 \times 10^{-2}$	0.4349	0.1132	0.1212	0.1020
CFD	72.00	90.00	60.00	$2.5 \times 10^{-4}$	0.6929	0.0677	0.1079	0.0559

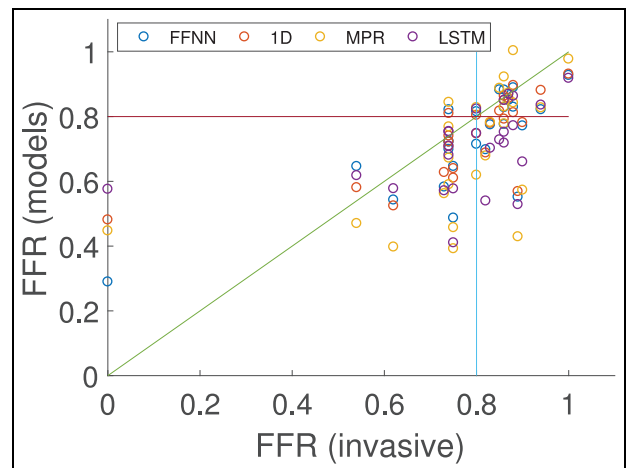
MPR: multivariate polynomial regression; FFNN: feed-forward neural network; LSTM: long short-term memory; CFD: computational fluid dynamics. Diagnostic measures include the percentages of the diagnostic accuracy, sensitivity, and specificity; and the linear correlation  $\rho$ , p-value, the mean absolute error (MAE), standard deviation (STD), and bias.

**Table 4.** Comparison of the MPR, FFNN, LSTM, with the ID physics-based model.

Method	MAE	STD	Bias
MPR	0.0863	0.0974	0.466
FFNN	0.0341	0.0484	0.0063
LSTM	0.0596	0.0640	0.4600

MPR: multivariate polynomial regression; FFNN: feed-forward neural network; LSTM: long short-term memory. The measures investigated are the mean absolute error (MAE), standard deviation (STD), and bias.

FFR measurements that were performed under coronary angiography for a cohort of 25 patients.<sup>15</sup> The results are shown in Figure 9. It is important to highlight that although 25 FFR values are compared, only 23 cases had the exact FFR value from the clinic. For two of the cases, the FFR was only recorded as positive ( $FFR < 0.8$ ) or negative ( $FFR \geq 0.8$ ), respectively. These two cases are shown in Figure 9 as  $FFR = 0$  for the positive case, and  $FFR = 1$  for the negative case to distinguish these two cases from those that have the FFR values recorded. The MPR, FFNN, and ID physics-based methods have the same diagnostic performance, sensitivity, and specificity as shown in Table 3. All 25 cases were used to determine the diagnostic accuracy, sensitivity, and specificity, but only the 23 cases that had the exact clinical FFR recorded were used in



**Figure 9.** Comparison of the ID physics-based model and the three machine learning/deep learning methods against the invasive clinical measurements on the coronary artery vessel network of 25 patients.

the calculation of the remaining performance indices. However, the 1D physics-based model has the lowest MAE, while the MPR which is the least complex model implemented in this work has the largest MAE. The most complex AI model implemented, the LSTM, shows the lowest diagnostic accuracy as it has the lowest sensitivity and tends to overestimate the pressure drop, thus leading to predict a positive FFR value

(FFR < 0.8) for cases where the invasive FFR is just above 0.8. Table 4 compares the MAE, standard deviation, and bias of the three ML models with that of the 1D physics-based model. The FFNN model consistently gave solutions closer to the physics-based model and had a smaller bias than the other two models. The LSTM performed significantly better than the MPR; however, the bias values were similar. The standard deviation of the invasive FFR measurements was 0.0940.

## Discussion

For the single vessel case in Figure 7, the LSTM showed the least spread of FFR prediction and highest linear correlation, although there is no significant difference between the diagnostic performance of the methods employed. Generally, the LSTM model showed bias towards the specificity implying that it is biased towards a positive FFR prediction (FFR > 0.8). Interestingly, the LSTM method gave the largest MAE which is mainly due to the bias as discussed above. All three of the ML-based methods were shown to give respectable FFR predictions when compared to the 1D physics-based model on single vessels, although the FFNN was shown to have the lowest MAE. However, in the clinic, the FFR measurement takes place between the aortic pressure to a point distal to a stenosis, and therefore it is very rare for only a single vessel to be considered.

In the multi-vessel case in Figure 8, there is a more noticeable difference in the quality of the results between the methods. This case is more challenging than the single vessel case as there is significantly more variation in the flow rates, vessel lengths, and area profiles. The MPR struggles to account for this variation. The training set and test set come from the same distribution of input features as required to create a well-defined ML model. However, some of the test set contained input feature variations that were not observed in the training set and thus the models needed to extrapolate the data to give predictions. As a result, the MPR model has the largest MAE and by far the lowest linear correlation value. MPR gives several erroneous predictions (not seen in Figure 9 as these values are extreme), for example, the maximum FFR predicted by the MPR is FFR = 1500, and the lowest value is FFR = -500, which are both physiologically impossible. This issue is a well-known problem in polynomial fitting<sup>32</sup>; however, other deep learning models can also give erroneous solutions, but in this case the FFNN and LSTM generally handle these extrapolated cases without difficulty. The use of regularisation for the MPR case was attempted, but this did not lead to an increase in performance on these extrapolated values. This is potentially an issue for AI models, as they can give non-physical predictions. However, there has been work to address such issues by adding constraints to the AI model,<sup>33</sup> which could be used to impose

conservation of mass and/or momentum. The FFNN model achieves a very high level of diagnostic accuracy and by far the lowest MAE. It is not too surprising that the FFNN works well as it is a universal approximator, so it can be used to model any continuous function.<sup>34</sup>

When comparing the ML models and the 1D physics-based model from which they were originally trained with the invasive clinical measurements, the FFNN again shows the best predictive power of the AI models. Again, the MPR has the highest MAE; however, the diagnostic accuracy, sensitivity, and specificity are the same for the FFNN, MPR, and 1D model. The LSTM performs worse than the other methods in terms of diagnostic accuracy, and is generally biased to predict a positive FFR. This may be due to the LSTM using zero-padding, which although required, can influence the accuracy of the model,<sup>35</sup> and there is more network variation in the patient data compared with the training cases (patient cases range from 3 vessel networks to 12 vessels in the network). The LSTM model is the most complex of the two deep learning approaches and may have been expected to outperform the FFNN model. However, the input for the LSTM model is the entire area profile of each vessel in the coronary network, while the FFNN only considers six input features as it is not considered effective at handling sequence data. This essentially means that the LSTM had to learn more complex relationships. In addition, gives a more informative output prediction for other regions in the coronary vessel as the output of the LSTM is the FFR values across the entire length of the network, while the FFNN model only gives a single FFR value at the end of a vessel.

It is observed that the standard deviation of the synthetic data is significantly higher than that in the real patient data. Due to the small patient cohort in this work, and that the majority of the cohort had an FFR measurement in the range 0.75–0.9, the standard deviation of the real patient cohort was relatively small. In order to successfully train an ML model that can generalise to predict FFR, it was deemed necessary to have a greater spread, and thus larger standard deviation, in the synthetic patient cases. This lowers the likelihood of over-fitting the ML models, and can help generalise it for any future patient cases tested that are outside the range observed in the current real patient cohort utilised in this study.

Although ML models can effectively be used to replace aspects of physics-based models, there are several aspects that need to be addressed before AI can be used in medicine for direct diagnostic and treatment decision-making. There have been several AI-based models for FFR that have been proposed, ranging from replacing the physics-based models,<sup>12</sup> predicting FFR from the segmented vessels,<sup>36</sup> to making the prediction directly via the CT scans.<sup>13,14</sup> However, utilisation of these AI predictions in the clinic raises ethical and validity issues, which still need to be appropriately addressed.<sup>37</sup> Another issue involved in non-invasive

FFR is the extraction of the coronary geometry from the CT data. This is a particular stumbling block for both physics-based models and ML techniques. However, deep learning techniques and computer vision can be effective at identifying objects in images. It is not beyond the power of a deep learning algorithm to be used to extract the required coronary vessel features directly from the CT images. Although this would also require a significant number of CT images to train any model.

Although AI models could be utilised to replace the well-established physics-based models in areas such as medical research, there is an argument that AI could supplement existing physics-based modelling, and rather be focused on replacing or improving the bottlenecks in this area of medical research, which is segmentation of the CT scans. This is in part due to the fact that the well-established reduced-order models are already very fast, and have been shown to give good accuracy; while segmentation is the slowest part of the non-invasive FFR prediction process,<sup>15</sup> and is often the most variable aspect of the prediction process with significant differences in segmented geometry observed between experienced users,<sup>38</sup> and even between different CT scanners.<sup>39</sup> AI could be used to reduce the amount of segmentation required, or even replace it entirely, which has been achieved for good quality CT images.<sup>40</sup>

## Limitations

In this work, only a small cohort of real patient measurements was suitable for FFR prediction using ML. This needs to be increased significantly in order to train the ML models using real patient data, rather than on a validated 1D physics-based model that is used as a surrogate.

## Conclusion

In this study, three AI models of varying degrees of complexity were compared to invasive FFR measurements. The AI models were initially trained using a 1D physics-based model on a virtual patient database. The AI models, in order of least complex to most complex, are the MPR, the FFNN, and the LSTM model. The models were compared to single vessel, and multi-vessel network cases from the virtual patient database, and also on clinically invasive FFR measurements. The least complex model, the MPR, struggled with the significant variation of area profiles, lengths, and flow rate estimations in the data, and produced some erroneous predictions. The most complex model, the LSTM performed well for the single vessel cases, but did not perform as well for the multi-vessel network and patient cases. The FFNN performed well for all cases.

## Acknowledgements

The authors gratefully acknowledge the contribution of Prof. Carl Roobottom and Dr Robin Alcock in supplying the anonymised patient data from Derriford Hospital and Peninsula Medical School, Plymouth Hospitals NHS Trust, used in this work.


## Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by Health Data Research UK (MR/S004076/1), which is funded by the UK Medical Research Council, Engineering and Physical Sciences Research Council, Economic and Social Research Council, Department of Health and Social Care (England), Chief Scientist Office of the Scottish Government Health and Social Care Directorates, Health and Social Care Research and Development Division (Welsh Government), Public Health Agency (Northern Ireland), British Heart Foundation and the Wellcome Trust.

## ORCID iD

Jason M Carson  <https://orcid.org/0000-0001-6634-9123>

## References

1. World Health Organization. Cardiovascular diseases (CVDs), 2017, [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))
2. Roth GA, Johnson C, Abajobir A, et al. Global, regional, and national burden of cardiovascular diseases for 10 causes, 1990 to 2015. *J Am Coll Cardiol* 2017; 70(1): 1–25.
3. Pijls N, De Bruyne B, Peels K, et al. Measurement of fractional flow reserve to assess the functional severity of coronary-artery stenoses. *N Engl J Med* 1996; 334: 1703–1708.
4. Sen S, Escaned J, Malik I, et al. Development and validation of a new Adenosine-Independent Index of stenosis severity from coronary wave-intensity analysis: results of the ADVISE (ADenosine Vasodilator Independent Stenosis Evaluation) Study. *J Am Coll Cardiol* 2012; 59: 1392–1402.
5. Sen S, Asrress KN, Nijjer S, et al. Diagnostic classification of the instantaneous wave-free ratio is equivalent to fractional flow reserve and is not improved with adenosine administration. *J Am Coll Cardiol* 2013; 61(13): 1409–1420.
6. Koo B, Erglis A, Doh J, et al. Diagnosis of ischemia-causing coronary stenoses by noninvasive fractional flow

- reserve computed from coronary computed tomographic angiograms: results from the prospective multicenter DISCOVER-FLOW (Diagnosis of Ischemia-Causing Stenoses Obtained Via Noninvasive Fractional Flow Reserve) Study. *J Am Coll Cardiol* 2011; 58: 1989–1997.
7. Boileau E, Pant S, Roobottom C, et al. Estimating the accuracy of a reduced-order model for the calculation of fractional flow reserve (FFR). *Int J Numer Method Biomed Eng* 2017; 34(1): e2908.
  8. Carson JM, Pant S, Roobottom C, et al. Non-invasive coronary CT angiography-derived fractional flow reserve: a benchmark study comparing the diagnostic performance of four different computational methodologies. *Int J Numer Method Biomed Eng* 2019; 35(10): e3235.
  9. Kourou K, Exarchos TP, Exarchos KP, et al. Machine learning applications in cancer prognosis and prediction. *Comput Struct Biotechnol J* 2015; 13: 8–17.
  10. Contreras I and Vehi J. Artificial intelligence for diabetes management and decision support: literature review. *J Med Internet Res* 2018; 20(5): e10775.
  11. Kickingereder P, Isensee F, Tursunova I, et al. Automated quantitative tumour response assessment of MRI in neuro-oncology with artificial neural networks: a multicentre, retrospective study. *Lancet Oncol* 2019; 20(5): 728–740.
  12. Itu L, Rapaka S, Passerini T, et al. A machine-learning approach for computation of fractional flow reserve from coronary computed tomography. *J Appl Physiol* 2016; 121: 42–52.
  13. Wang ZQ, Zhou YJ, Zhao YX, et al. Diagnostic accuracy of a deep learning approach to calculate FFR from coronary CT angiography. *J Geriatr Cardiol* 2019; 16: 42–48.
  14. Kumamaru KK, Fujimoto S, Otsuka Y, et al. Diagnostic accuracy of 3D deep-learning-based fully automated estimation of patient-level minimum fractional flow reserve from coronary computed tomography angiography. *Eur Heart J Cardiovasc Imaging* 2019; 21: 437–445.
  15. Carson J, Roobottom C, Alcock R, et al. Computational instantaneous wave-free ratio (IFR) for patient-specific coronary artery stenoses using 1D network models. *Int J Numer Method Biomed Eng* 2019; 35: e3255.
  16. McCulloch WS and Pitts W. A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 1943; 5(4): 115–133.
  17. Turing AM. Computing machinery and intelligence. *Mind* 1950; LIX(236): 433–460.
  18. Cauchy A. methode gnrale pour la rsolution des systmes d'equations simultanenes. *C R Hebd Seance Acad Sci* 1847; 25: 536–538.
  19. Kelley HJ. Gradient theory of optimal flight paths. *ARS J* 1960; 30(10): 947–954.
  20. Kelley HJ. Method of gradients. *Math Sci Eng* 1962; 5: 205–254.
  21. Werbos PJ. Applications of advances in nonlinear sensitivity analysis. In: Drenick RF and Kozin F (eds) *System modeling and optimization*. Berlin: Springer, 1982, pp.762–770.
  22. Rumelhart DE, Hinton GE and Williams RJ. Learning representations by back-propagating errors. *Nature* 1986; 323(6088): 533–536.
  23. Kingma DP and Ba J. Adam: a method for stochastic optimization. In: *International conference on learning representations (ICLR)*, San Diego, CA, 7–9 May 2015. Ithaca, NY: arXiv.org.
  24. Mynard JP and Smolich JJ. One-dimensional haemodynamic modeling and wave dynamics in the entire adult circulation. *Ann Biomed Eng* 2015; 43(6): 1443–1460.
  25. Hochreiter S. *Untersuchungen zu dynamischen neuronalen Netzen*. Master's Thesis, Institut fur Informatik, Technische Universitat, Munchen, 1991.
  26. Bengio Y, Simard P and Frasconi P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans Neural Netw* 1994; 5(2): 157–166.
  27. Hochreiter S and Schmidhuber J. Long short-term memory. *Neural Comput* 1997; 9(8): 1735–1780.
  28. Carson J. *Development of a cardiovascular and lymphatic network model during human pregnancy*. PhD Thesis, Swansea University, Swansea, 2018.
  29. Carson J, Lewis M, Rassi D, et al. A data-driven model to study utero-ovarian blood flow physiology during pregnancy. *Biomech Model Mechanobiol* 2019; 18: 1155–1176.
  30. Mynard J, Penny D and Smolich J. Scalability and in vivo validation of a multiscale numerical model of the left coronary circulation. *Am J Physiol Heart Circ Physiol* 2014; 306: H517–H528.
  31. Carson J and Loon RV. An implicit solver for 1D arterial network models. *Int J Numer Method Biomed Eng* 2016; 33(7): e2837.
  32. Royston P. Model selection for univariable fractional polynomials. *Stata J* 2017; 17: 619–629.
  33. Zhu Y, Zabarás N, Koutsourelakis PS, et al. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J Comput Phys* 2019; 394: 56–81.
  34. Leshno M, Lin VY, Pinkus A, et al. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw* 1993; 6(6): 861–867.
  35. Dwarampudi M and Reddy NVS. Effects of padding on LSTMs and CNNs, 2019, <https://arxiv.org/abs/1903.07288>
  36. Cho H, Lee JG, Kang SJ, et al. Angiography-based machine learning for predicting fractional flow reserve in intermediate coronary artery lesions. *J Am Heart Assoc* 2019; 8(4): e011685.
  37. Vayena E, Blasimme A and Cohen IG. Machine learning in medicine: addressing ethical challenges. *PLoS Med* 2018; 15(11): e1002689.
  38. De Knecht MC, Haugen M, Linde JJ, et al. Reproducibility of quantitative coronary computed tomography angiography in asymptomatic individuals and patients with acute chest pain. *PLoS ONE* 2018; 13(12): e0207980.
  39. Symons R, Morris JZ, Wu CO, et al. Coronary CT angiography: variability of CT scanners and readers in measurement of plaque volume. *Radiology* 2016; 281(3): 737–748.
  40. Huang W, Huang L, Lin Z, et al. Coronary artery segmentation by deep learning neural networks on computed tomographic coronary angiographic images. In: *2018 40th Annual international conference of the IEEE engineering in medicine and biology society (EMBC)*, Honolulu, HI, 18–21 July 2018. New York: IEEE.