25th International Congress on Modelling and Simulation, Darwin, NT, Australia, 9 to 14 July 2023
mssanz.org.au/modsim2023

Surrogate model for CFD based on machine learning

Ashfaqur Rahman [a], Gerald Pereira [a], Phil Kilby [a] and Paulus Lahur [b]

[a] Data61, CSIRO, Australia
[b] IM&T, CSIRO, Australia
Email: Ashfaqur.rahman@data61.csiro.au

Abstract: The paper presents the results of a data-driven surrogate modelling effort to reproduce the results of a Computational Fluid Dynamics (CFD) model. The CFD model considered here simulates a scenario where air containing CO_2 flows through a tube. A metal scaffold (that are highly selective in adsorbing CO_2 from air) is placed inside the tube that absorbs CO_2 from the fluid as it flows through. The CFD model computes two important quantities: (i) the amount of transportation (a measure of CO_2 absorbed by the metal scaffold) and (ii) the amount of fluid mixing. Given the shape features of the metal scaffold as input, the CFD model (a) produces a 3D lattice filling part of it with the metal scaffold, (b) uses Lattice-Boltzmann equation to solve the CFD, and (c) compute amount of transportation and fluid mixing.

CFD models normally requires solving partial differential equations across the grid making it a highly time-consuming process. To reduce the computation time, we have investigated surrogate models for this CFD based on a machine learning (ML) method. The ML model takes as input the shape features of the metal scaffold and predicts amount of CO_2 absorption and fluid mixing. The objective is to come up with a ML model that produces the above quantities with reasonably low error. ML models can produce the results much faster than CFD model. We investigated several ML models to find their effectiveness to produce reasonably accurate results for predicting the two quantities above.

The data for training and testing the ML model was generated from the CFD based on different shape features of the metal scaffold. The underlying objective of the project was to produce an optimal design of the metal scaffold that maximises absorption (of CO_2) and fluid mixing. The optimization was done using an evolutionary algorithm (EA). EA starts with a set (population) of random shapes and modifies the shapes over generations based on feedback from CFD. In each generation the CFD computes the absorption and mixing for each of the shapes in the population. The best of the shapes from the population goes through some transformations (crossover and mutation) to generate the population for next generation. The process continues until the optimization converges. The shape features and their fitness (absorption and mixing) of the population from the several generations are combined into one large dataset where the shape features are input, and the fitness values are target. This constitutes the data used for training and testing the ML models. Given a total of n generations, we trained ML models on data up to k–th generation and tested the model on the data from last $n–k$ generations.

We have trained ML models separately for the two targets: amount of transportation and fluid mixing. The input features (i.e., properties of the scaffold) remain the same for both models. We trained and tested several machine learning models to produce this mapping between properties of the scaffold and the two targets. The ML models are: Multivariate Linear Regression, Support Vector Regression, Neural Network Regression, and Decision Tree Regression. The Decision Tree Regression performed best for our scenario. We present the CFD model, the data-driven ML based surrogate models, and some preliminary results in this paper.

Keywords: *Mixer modelling, CFD, surrogate models*

1. INTRODUCTION

Computational Fluid Dynamics (CFD) is an area of fluid dynamics where behaviour of fluid flow is modelled using numerical method and algorithms. Many real-world scenarios involving fluid flow requires testing under different physical phenomenon. CFD offers a cost-effective alternative to imitate the physical scenario in a mathematically simulated scenario and test the behaviour of fluid under different testing scenarios. However, due to computationally heavy, CFD models take large time to generate results even on high performance computing platforms. ML based surrogate models offer a faster alternative to CFD models if it can produce accurate predictions. In this paper, we explore some data-driven ML models to find their effectiveness in accurately emulating the CFD model.

The CFD model considered in this paper is aimed at simulating a scenario where fluid air containing CO_2 flows through a tube containing a metal scaffold. The metal scaffold is highly selective in adsorbing CO_2 from air. The underlying objective of the project is the produce a design of the metal scaffold that maximises absorption and fluid mixing. The overall process is an iterative process. An evolutionary optimization method based on Genetic Algorithm (Katoch et al. 2021) iterates over generations producing different shape parameters. The shape parameters are taken as input by a bespoke geometry generator to produce a solid body with certain shape. A grid generator then takes the geometry and produces computational grid.

The CFD model takes the computational grid as input and then solves the Lattice Boltzmann equations (Benzi et al. 1992) and computes two quantities – amount of transportation (a measure of absorption of CO_2 by the metal substrate) and fluid mixing. Fluid mixing is required so we can get as uniform as possible deposition on substrate. These quantities act as a fitness function for the Genetic Algorithm (GA). GA starts with a population of random shapes and the CFD produces their fitness score. The next generations of shapes are modified based on the fitness values produced by the CFD and the process continues until the GA converges.

The whole process is very time consuming. The idea of the ML exercise is to investigate if the mapping between the shape parameters and fitness values (transportation and fluid mixing) can be learned from the data produced over generations by the evolutionary algorithm framework. We have explored different ML models to find the mapping between shape parameters of the metal scaffold and the fitness values. The details of the underlying method and results are presented in the following sections.

2. METHODS

2.1. CFD modelling

The CFD model considered in this study focuses on solving the steady-state velocity field for fluid flow within the long tube. This is done by solving the Lattice Boltzmann (LB) equations for fluid flow. The LB method is a microscopic model which can be shown to yield solutions equivalent to the well-known Navier-Stokes equations. The advantage here is that LB models treat complex shapes comparatively easily and is the method is highly parallelizable (both improving compute speed and accuracy). Once the velocity field has been determined, fictitious tracers are launched through the long tube. Their trajectory is determined by numerically solving a simple equation ($d\mathbf{r}/dt = \mathbf{v}(\mathbf{r})$), where $\mathbf{r}$ is the position of a tracer particle at time t and $\mathbf{v}$ is the corresponding velocity at this position). Many thousands of tracers are modelled to give a statistically meaningful sample, from which the quantities of interest (metal scaffold transport and fluid mixing) are determined.

2.2. Surrogate modelling

This section explains the method of developing a data-driven surrogate model for the CFD. As discussed in the previous section, an Evolutionary Algorithm (EA) optimizes the shape of the mixer metal scaffold over generations and the CFD model provides fitness values (mixing and transportation) for the shapes in that generation. Let the EA works on a population (number of alternative shapes in each generation) of size N. Each shape is described by a set of n features ω. The CFD produces two fitness values mixing (σ) and transportation (ρ) for each of the N shapes in that generation. Let, $\boldsymbol{\omega}_{ij}$, σ_{ij} and ρ_{ij} denotes the feature vector, mixing and transportation amounts respectively for the j-th shape in the i-th generation. If the EA converges on k–th generation, the EA and CFD generate the following:

Feature vector	Mixing	Transportation
$\boldsymbol{\omega}_{11}$	σ_{11}	ρ_{11}
$\boldsymbol{\omega}_{12}$	σ_{12}	ρ_{12}
…	…	…
$\boldsymbol{\omega}_{1N}$	σ_{1N}	ρ_{1N}
$\boldsymbol{\omega}_{21}$	σ_{21}	ρ_{21}
$\boldsymbol{\omega}_{22}$	σ_{22}	ρ_{22}
…	…	…
$\boldsymbol{\omega}_{2N}$	σ_{2N}	ρ_{2N}
…	…	…
…	…	…
$\boldsymbol{\omega}_{kN}$	σ_{kN}	ρ_{kN}

A total of $k \times N$ CFD runs generate mixing and transportation amount for $k \times N$ shape features as proposed by EA. This constitutes the data to train and test ML models. Given a set of feature vectors $\boldsymbol{\omega}_{11}, \dots, \boldsymbol{\omega}_{mN}$ up to m-th generation, and corresponding mixing and transportation amounts $\sigma_{11}, \dots, \sigma_{mN}$ and $\rho_{11}, \dots, \rho_{mN}$ (as generated by CFD), we train two different ML models f_m and f_a for mixing and transportation separately. Let, θ_m and θ_a be the parameters for the ML models f_m and f_a respectively. Given a set of inputs $\boldsymbol{\omega}_{11}, \dots, \boldsymbol{\omega}_{mN}$ and corresponding targets (mixing or transportations), an ML model aims to approximate a function to map the input to the corresponding targets such that cumulative error between the predictions and the targets are minimized and hence the optimal values of θ_m and θ_a are obtained as:

$$\underset{\theta_m}{argmin} \left(\sum_{\forall_{i,j}} \left| \sigma_{ij} - f_m(\boldsymbol{\omega}_{ij}) \right| \right) \tag{1}$$

$$\underset{\theta_a}{argmin} \left(\sum_{\forall_{i,j}} \left| \rho_{ij} - f_a(\boldsymbol{\omega}_{ij}) \right| \right) \tag{2}$$

Once the ML models f_m and f_a are trained on the first m generations, the models are tested on samples from generations $m+1$ to k. The errors computed on the test set (unseen to the training process), indicates the generalizability of the ML models. The method for generating the ML models from CFD is presented in Figure 1.

3. EXPERIMENTAL SETUP AND RESULTS

In this paper we present the results on a particular family of metal scaffolds called HEXRAIN (Figure 2). These substrates are placed inside the tube through which the air flows. The objective of the evolutionary algorithm is optimising the design of such shapes to maximise CO_2 transportation and fluid mixing. The shape of the of the HEXRAIN metal scaffold is controlled by 14 parameters (for instance, the distance between sub-elements, or the angle one element forms to a neighbouring one etc.). We use a shape generation process based on the L-system method. The details of L-Systems are beyond the scope of the paper but interested readers can find the details in (Lindenmayer 1968).

The genetic algorithm (the evolutionary algorithm) is a population-based method, using 16 individuals in each generation. An initial population of 16 individuals (representing substrate shape features), each represented by a vector of real numbers, is created randomly. The L-System is used to generate the shape represented by each parameter vector. The CFD system is then run to calculate the two fitness values for each shape – (a) CO2 transportation, and (b) fluid mixing. After this initial population, the GA proceeds in the same way each generation. First, a new population of 16 individuals is created by combining the parameter sets of the current population. This is done by crossover (combining parts of two solution) and mutation (altering some parameter values). The process of generating the shapes using the L-system and calculating the fitness values using CFD is then repeated for the new population. Finally, the 16 individuals with the best fitness values are selected from the 32 individuals in the combined new and old populations. These 16 go on to the next generation, and the procedure repeats until no improvement in fitness values is seen.

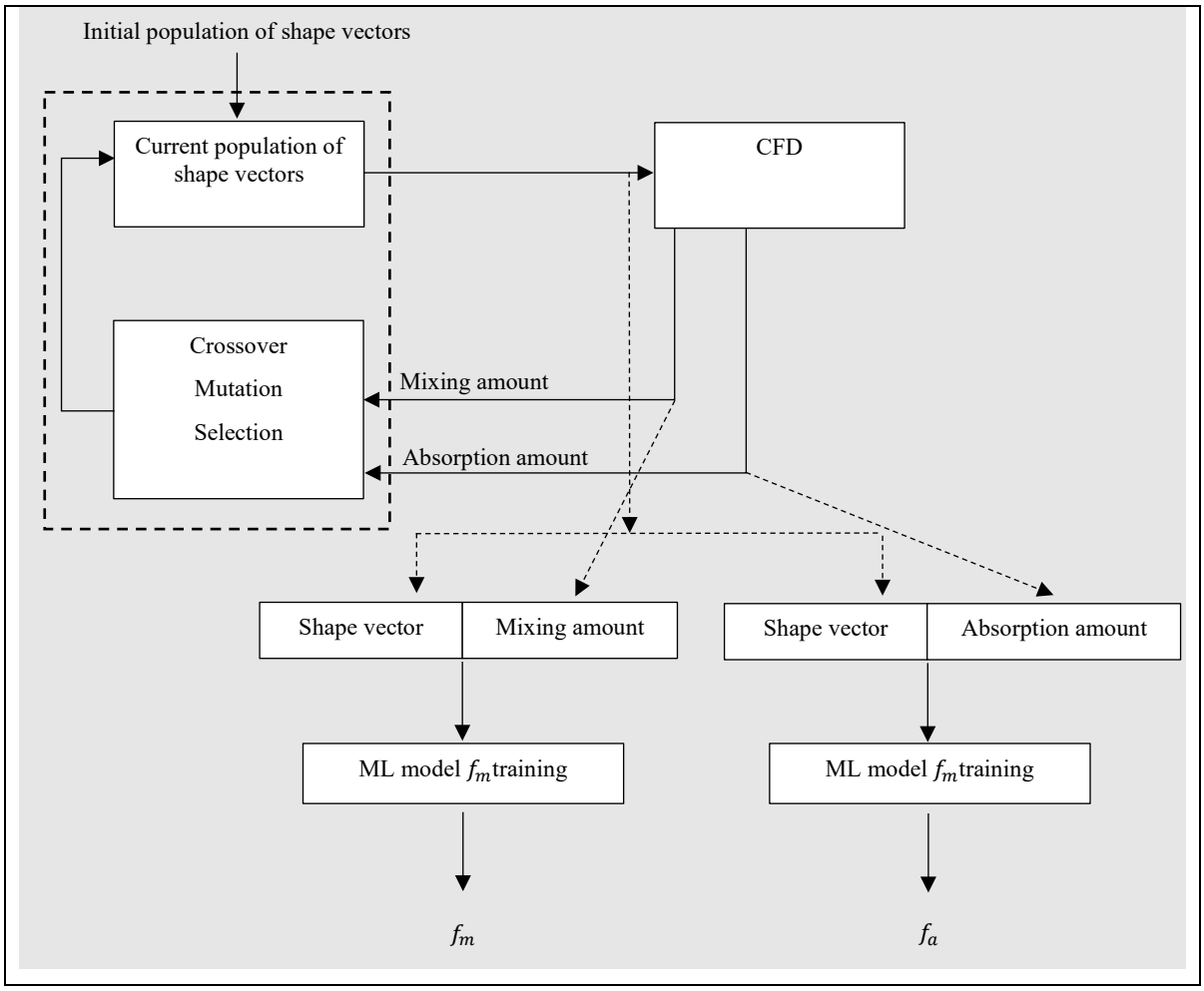


Figure 1. Method for generating data-driven surrogate ML models for the CFD

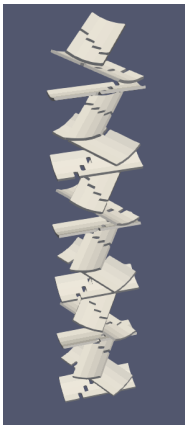


Figure 2. HEXRAIN shape metal scaffold/mixer (shown without hexagonal walls)

A common technique to explore the parameter space effectively is to restart this procedure from a new random initial population. We performed seven restarts from random initial populations. We used data from all these seven runs of GA. For each run, we used the shapes generated over the successive generations of the evolutionary algorithm for training and testing of the ML based surrogate models. We combined the data from first 40 or so generations from each of the seven restarts of HEXRAIN shapes to train the model. We then combined the data from the following 36 or so generations from four restarts of HEXRAIN shapes to test the model. A good number of samples from both training and test set were unusable as those scenarios lead to blockages in the tube – a situation that needs to be avoided. Hence such shapes are excluded from training and

testing. We ended up using 3712 samples for training and 2216 samples for testing. Each sample in the data has 14 input features (shape parameters of the metal scaffold) and the target for each sample is (a) CO_2 transportation, and (b) fluid mixing.

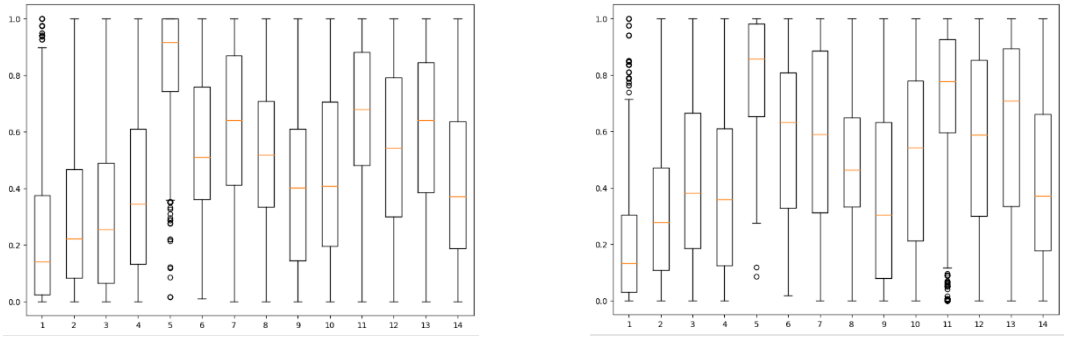
For training and testing, we have investigated several supervised machine learning models including Multivariate Linear Regression (Alexopoulos 2010), Support Vector Regression (Awad 2015), Neural Network Regression (Specht 1991), and Decision Tree Regression (Sammut et al. 2011). The methods are briefly explained bellow:

- Multivariate Linear Regression (MLR): Given a set of n input features x_i for a sample, and corresponding target y (where $1 \leq i \leq n$), MLR expresses y as a linear combination of the x_i as $y = \sum_{i=1}^{n} a_i x_i$. The a_is are solved for based on an optimisation method to reduce the difference between prediction $\sum_{i=1}^{n} a_i x_i$ and the actual target y for all the samples in the training set.
- Support Vector Regression (SVR): SVR uses a kernel function to map the input vector $\boldsymbol{x}$ to a set of features $\boldsymbol{x}'$ and aims to find a hyperplane (y') that contains majority of the points within a margin of the hyperplane. The optimal hyperplane is sought for during the training process from the given data. Some commonly used kernel functions are: Linear, Non-Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid function.
- Neural Network Regression (NNR): NNR is a directed graph structure organised in layers and the edges are weighted. The first layer of the NNR is the input vector. Each node in the following layer computes a weighted combination of the inputs from the previous layer, passes through a non-linear function (sigmoid, tanh, Relu etc. (Sharma et al. 2020)) to produce the output of that node. The last layer contains one node that computes a weighted summation of the outputs from the second last layer and to produce the output of the neural network. The weights of the NNR are optimized such that the sum of error between predictions and actual targets from the training data are minimized.
- Decision Tree Regression (DTR): Decision tree is a tree (a type of graph) where root node and other interior nodes represents one of the input features and the leaf nodes represents decision (regressed value). The building of the decision tree is an iterative process. First, the feature (/attribute) with highest variability (i.e., maximum entropy) among the samples is selected as the root node. The whole data set is then split into multiple smaller datasets depending on the range in which the attribute in the root node belongs. Each smaller dataset corresponds to an internal node. The variability/entropy calculation process continues the smaller data set to find a representative for the corresponding internal node. The process continues until certain criteria is satisfied. The average of the target values in each leaf node represents the decision of that leaf node.

The ML experiments were conducted in Python. The following Python packages were used in the experiments: pandas, numpy, sklearn, and GaussianMixture. We computed two error metrices: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Given an ordered set of targets $y_1, \ldots, y_n$ and the corresponding ordered set of predictions $y'_1, \ldots, y'_n$, MAE and RMSE are computed as:

$$MAE = \sum_{i=1}^{n} |y_i - y'_i| \text{ and } RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y'_i)^2} \quad (3)$$

ML models operate on the principal that training and test data are independent but identically distributed. Non-identical distribution will lead to poor performance on test set (Rahman et al. 2018). Figure 3 shows the distribution of features in the training and test set. The feature distribution is almost identical between two sets.



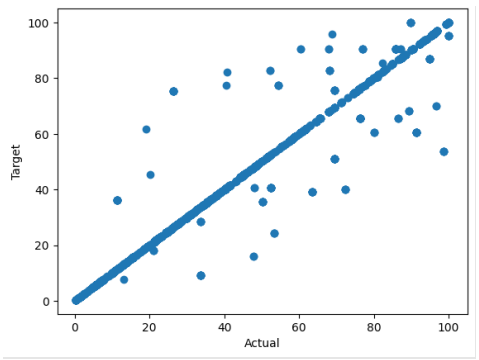
(a) Training data (b) Test data

Figure 3. Feature distribution in the data set in terms of box plot

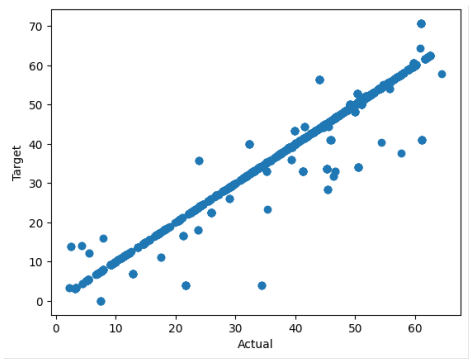
We implemented several data-driven ML models to obtain mapping between the input shape features of the metal scaffold and the quantities produced from the CFD model i.e., CO_2 transportation and fluid mixing. We trained and tested four models MLR, SVR, NNR, and DTR as explained above. We used linear_model, SVR, MLPRegressor, and DecisionTreeRegressor from scikitlearn package in python to train and test MLR, SVR, NNR, and DTR models respectively. Table 1 presents the errors in predicting transportation and mixing quantities by these ML models on the test set. Note that DTR produced the smallest error compared to all the other methods. The correlation plot between actual and predicted values (by DTR) for the two quantities in the test set are presented in Figure 4. As evidenced from the trend in both graphs, the predicted values aligned very well to the actual values in majority of cases.

Table 1. Comparison of performance of different ML models to predict target variables in the test set

	Transportation		**Mixing**	
	MAE	**RMSE**	**MAE**	**RMSE**
MLR	18.61	115.05	13.23	58.80
SVR	9.550	87.770	5.380	31.67
NNR	9.580	59.030	6.560	28.66
DTR	1.410	3.3600	0.710	3.050



(a) Transportation



(b) Mixing

Figure 4. Prediction performance of decision tree on test data

4. CONCLUSIONS

In this paper we have presented a data-driven surrogate ML modelling approach to a CFD to accurately predict two quantities: transportation (of CO_2) and fluid mixing. We trained and tested four different ML models including multivariate linear regression, support vector regression, neural network regression, and decision tree regression. Decision tree regression produced best results for both targets. These results are preliminary and more needs to be done in the future. The parameters of the learning models were not optimised, and we need to investigate this in future. The results presented in this paper are based on only one family of shapes HEXRAIN and we will investigate other family of shapes in future. We also need to investigate if we can transfer learn from models trained on one family of shapes and apply on another. Finally, the current approach is purely data driven and does not consider the underlying physics. In future we aim to investigate Physics Informed Neural Network (PINN) to incorporate underlying physics into the learning process.

REFERENCES

Alexopoulos, E.C., 2010. Introduction to multivariate regression analysis, Hippokratia. (Suppl 1):23–8. PMID: 21487487; PMCID: PMC3049417.

Awad M., Khanna, R., 2015. Support Vector Regression. In: Efficient Learning Machines, Apress, Berkeley, CA, https://doi.org/10.1007/978-1-4302-5990-9_4

Benzi, R., Succi, S., Ergassola, M., 1992. The lattice Boltzmann equation: theory and applications, Physics Reports, 222(3), 145–197, https://doi.org/10.1016/0370-1573(92)90090-M.

Katoch, S., Chauhan, S.S., Kumar, V., 2021. A review on genetic algorithm: past, present, and future, Multimed Tools Appl 80, 8091–8126. https://doi.org/10.1007/s11042-020-10139-6.

Lindenmayer, A., 1968. Mathematical models for cellular interactions in development II. Simple and branching filaments with two-sided inputs, Journal of Theoretical Biology, 18(3), 300–315.

Rahman, A., Smith, D.V., Little, B., Ingham, A.B., Greenwood, P.L., Bishop-Hurley, G.J., 2018. Cattle behaviour classification from collar, halter, and ear tag sensors, Information Processing in Agriculture, 5(1), 124–133, https://doi.org/10.1016/j.inpa.2017.10.001.

Sammut, C., Webb, G.I., 2011. Decision Trees For Regression, In: Sammut, C., Webb, G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-30164-8_205

Sharma, S., Sharma, S., Athaiya, A., 2020. Activation Functions in Neural Networks, International Journal of Engineering Applied Sciences and Technology, 4(12), 310–316.

Specht, D. F., 1991. A general regression neural network, IEEE transactions on neural networks, 2(6), 568–576.