

REVIEW

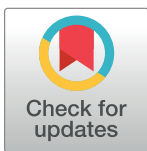
Bridging the gap between mechanistic biological models and machine learning surrogates

Ioana M. Gherman¹, Zahraa S. Abdallah¹, Wei Pang², Thomas E. Gorochowski³, Claire S. Grierson³, Lucia Marucci^{1*}

1 Department of Engineering Mathematics, University of Bristol, Bristol, United Kingdom, **2** School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, United Kingdom, **3** School of Biological Sciences, University of Bristol, Bristol, United Kingdom

☞ These authors contributed equally to this work.

* lucia.marucci@bristol.ac.uk



Abstract

Mechanistic models have been used for centuries to describe complex interconnected processes, including biological ones. As the scope of these models has widened, so have their computational demands. This complexity can limit their suitability when running many simulations or when real-time results are required. Surrogate machine learning (ML) models can be used to approximate the behaviour of complex mechanistic models, and once built, their computational demands are several orders of magnitude lower. This paper provides an overview of the relevant literature, both from an applicability and a theoretical perspective. For the latter, the paper focuses on the design and training of the underlying ML models. Application-wise, we show how ML surrogates have been used to approximate different mechanistic models. We present a perspective on how these approaches can be applied to models representing biological processes with potential industrial applications (e.g., metabolism and whole-cell modelling) and show why surrogate ML models may hold the key to making the simulation of complex biological systems possible using a typical desktop computer.

This is a *PLOS Computational Biology* Methods paper.

Introduction

Mathematical mechanistic models have been used for centuries to understand and represent the natural laws that shape the world around us. Initially, the focus was on modelling specific phenomena and the mechanics underpinning them, but in time, these models became more complex and are now able to represent interconnected processes and the interaction with their environment, bringing us closer to building digital twins [1–3]. This complexity brings a number of challenges, among which their computational demand is one of the most pressing ones. Simulations of complex mechanistic models can take hours or days to run, making them

OPEN ACCESS

Citation: Gherman IM, Abdallah ZS, Pang W, Gorochowski TE, Grierson CS, Marucci L (2023) Bridging the gap between mechanistic biological models and machine learning surrogates. *PLoS Comput Biol* 19(4): e1010988. <https://doi.org/10.1371/journal.pcbi.1010988>

Editor: Marcel Holger Schulz, Goethe University Frankfurt: Goethe-Universität Frankfurt am Main, GERMANY

Published: April 20, 2023

Copyright: © 2023 Gherman et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: LM was funded by the Engineering and Physical Sciences Research Council (EPSRC, EP/S01876X/1) and the Biotechnology and Biological Sciences Research Council (BBSRC, Bristol Center for Engineering Biology, BB/W013959/1; Breakthrough Award BB/W012235/1). TEG was funded by the Royal Society University Research Fellowships UF160357 and URF/R/221008, and a Turing Fellowship from The Alan Turing Institute (EPSRC grant EP/N510129/1). IMG was supported by an EPSRC Doctoral Training Studentship and two BBSRC grants (BB/W013770/1 and BB/X511997/1). The funders had no role in study

design, data collection and analysis, decision to publish or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

unfeasible for real-time decision-making or sensitivity analysis. This can make users reluctant to utilise them, despite their predictive power.

Here, we show how the high computational demand of some mechanistic models can be alleviated by using machine learning (ML) surrogates as a proxy. ML surrogates, also known as emulators or metamodels, are simpler models that approximate the behaviour of a mechanistic one. Usually, they take as input the initial conditions and/or parameters of the mechanistic model and they predict some or all of its outputs. Once the ML surrogate is trained and validated, it can replace the original mechanistic model in all future simulations, with the added advantage of making the simulations several orders of magnitude faster. The process of training and using an ML surrogate is shown in Fig 1. To create a surrogate, it is necessary to decide what output needs to be predicted and which inputs of the mechanistic model will be varied. Often it is not necessary to predict everything that the mechanistic model outputs, making the training process of the surrogate faster and more efficient. Once these choices are made, several simulations of the mechanistic model can be run by varying the chosen inputs to create input-output pairs for training, testing, and validating the ML model. The data is usually split such that 80% to 90% is used for training, and the remaining 10% to 20% is used for testing. To validate the surrogate model, one can either split the training data further such that a fixed percentage (usually 10% to 20%) is used only to validate the model [4–6] or implement cross-validation

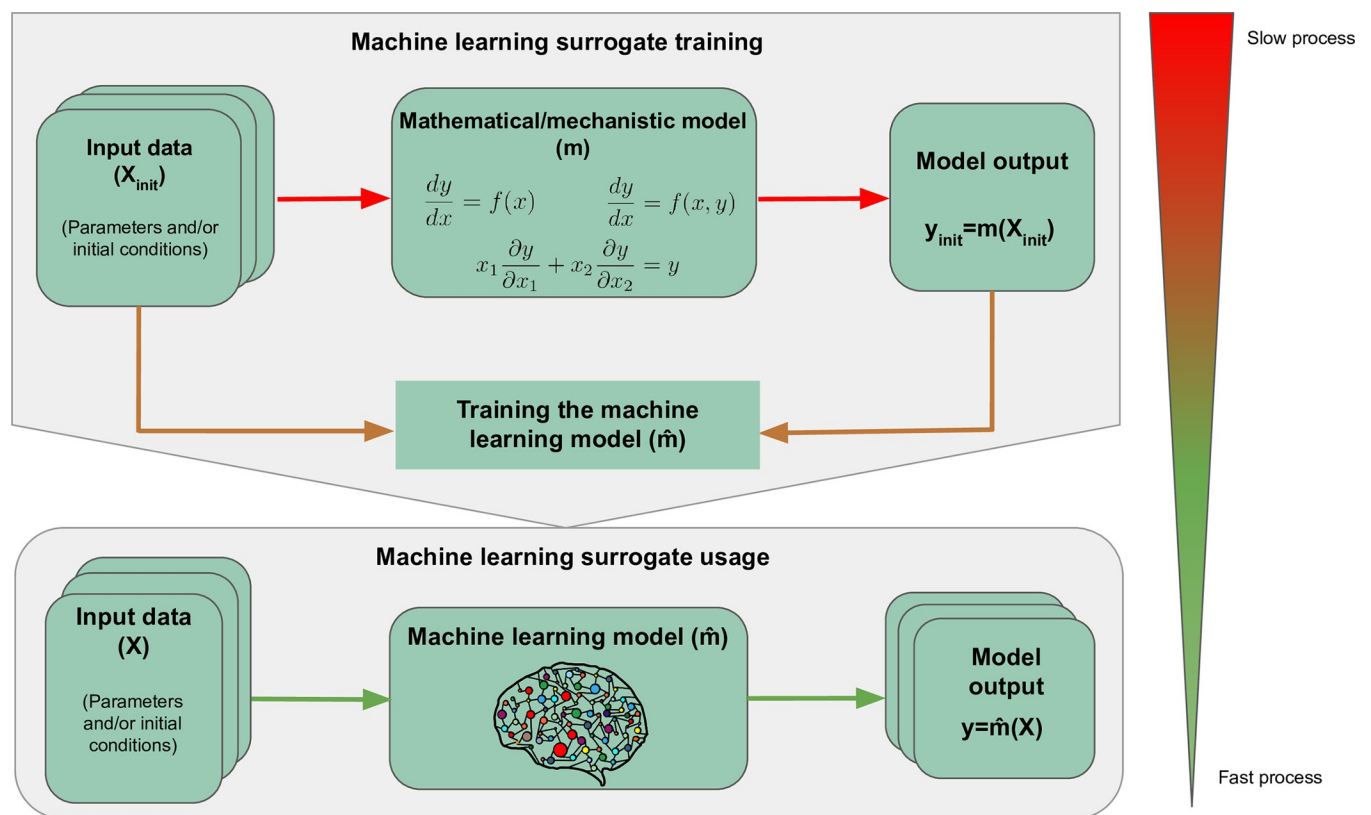


Fig 1. Schematic representation for training and using an ML-based surrogate model. The mechanistic model is simulated (the top process connected by red arrows) to obtain the input-output pairs that are used to train the ML surrogate. This training stage (the middle process connected by orange arrows) is an average process in terms of speed. Its complexity will depend on the ML algorithm used, the complexity of the data preprocessing steps, and the quantity of training iterations needed to obtain a satisfactory accuracy. Once this is achieved, the ML model can be used for all future predictions, effectively approximating the mechanistic model while running several orders of magnitude faster. The green arrows at the bottom of the figure represent this final (fast) process.

<https://doi.org/10.1371/journal.pcbi.1010988.g001>

[7–11]. In terms of computational demand, the speed of the training phase relies on the ML model used and the number of iterations needed to obtain a satisfactory accuracy.

The improvement in computational speed that ML surrogate models achieve is particularly useful when predictions are needed in real-time [9,10,12] or when large numbers of simulations have to be run, for example, to explore a model's parameter space [13–15]. It is important to acknowledge that simplified models can also lead to further improvements of the original ones. For example, different types of surrogate models have been used to analyse the uncertainty in the structure and the predictions of mechanistic models [16,17]. Also, while building the surrogate, it is possible to gain further understanding of the model's relationship between inputs, parameters, and outputs, discovering for example, insensitive parameters [18].

Biological processes have been modelled mechanistically for decades and can be split based on 2 paradigms. The first one classifies them by scale [19], considering the cell as the “unit” for measurement, it is possible to create models at the subcellular, cellular, or macroscopic level. Subcellular models describe the evolution of individual physical and biochemical states of a cell [20]. Cellular-level models describe the interactions among different molecules and processes within cells, and macroscopic-level models describe processes that involve groups of cells [19]. The second paradigm classifies biological models based on the mathematical formalism that they use [21]. Biological processes are commonly modelled using ordinary differential equations (ODEs) [22–24], partial differential equations (PDEs) [25–27], agent-based modelling [21,28], cellular automata [29,30], stoichiometric matrices [31,32], stochastic techniques (for example, stochastic differential equations, SDEs) [23], or rule-based methods [33]. Details of each modelling approach are addressed in [19,21].

The main aim of this review is to bridge the gap between computationally demanding mechanistic models that describe biological systems at different cellular levels and the potential use of ML surrogates. First, we will review the performance of different ML-based surrogate models, while analysing their advantages and disadvantages when applied to ODE, SDE, and PDE-based mechanistic models. Then, we will discuss the benefits of using surrogate ML models in general, their limitations, and the future avenues for improving these models and making them more usable by scientists from different fields and communities. Finally, we will present how ML surrogate models can be relevant to approximate mechanistic models in the context of bioengineering industrial applications.

ML as a surrogate in systems biology

ML-based surrogates were used to approximate mechanistic models of biological systems based on ODEs, SDEs [4,7], and PDEs [5,8–10,13]. These applications will be summarised below, focusing on the methodology and results of each study. Table 1 presents a summary of the surrogate ML models used for biological applications and their performance relative to the original mechanistic model they approximate. An overview of relevant methodological studies that apply ML-based surrogate modelling to mechanistic models from other engineering disciplines is presented in Table 2. The rest of this section will explain how these results were obtained.

ML surrogates of ODE and SDE systems

Dynamical systems that evolve only in 1 dimension are usually modelled using ODEs or SDEs. ML surrogates were successfully applied to approximate systems biology models based on both types of techniques. For example, Renardy and colleagues [7] built a surrogate based on an orthogonal polynomial basis from the generalised polynomial chaos (gPC) using the least square approximation for the heterotrimeric G-protein cycle of budding yeast. Once trained,

Table 1. Summary of the performance and methodologies of the ML surrogates of the systems biology models.

Original model description	Surrogate algorithm	Surrogate accuracy	Improvement in computational time	Reference
SDE model of the MYC/E2F pathway [22,23]	LSTM	R^2 0.925–0.998	-	[4]
Heterotrimetric G-protein of budding yeast [24]	Orthogonal polynomial basis from the generalised polynomial chaos (gPC)	MAE 2.5×10^{-2}	20% reduction in CPU time	[7]
Pattern formation in <i>E. coli</i> [25]	LSTM.	R^2 0.987–0.99	30,000 fold acceleration	[4]
Pheromone-induced cell polarisation in budding yeast [26]	Orthogonal polynomial basis from the generalised polynomial chaos	MAE 0.14	180-fold reduction	[7]
Statistical model for aorta shapes [27,34,35]	PCA, bidirectional neural network, feedforward neural network	Avg. MAE 0.533 mm	4 orders of magnitude	[8]
Risk for ascending aortic aneurysm [35]	PCA, bidirectional neural network, feedforward neural network	Avg. MAE: 1.366 KPa	4 orders of magnitude	[9]
Stress analysis of arterial walls under atherosclerosis [5]	Feedforward neural network	Test error 9.86%	-	[5]
Normal left ventricle [36–38]	XGBoost or Cubist	MAE for volume 1.495, MAE for pressure 1.544	2–3 orders of magnitude	[10]
Human left ventricle model [13,39,40]	Gaussian process	MSE 0.0001	3 orders of magnitude	[13]
Human left ventricle [11,39]	K-Nearest Neighbour, XGBoost, Multilayer Perceptron	R^2 0.999 (for the XGBoost and Multilayer Perceptron)	3–4 orders of magnitude	[41]
Physiology models: Small and HumMod [42]	SVM regression	Average error for Small: $0.05\hat{A} \pm 2.47$ and for HumMod: -0.3 ± 3.94	6 orders of magnitude	[43]

The accuracy of the models is reported in terms of mean average error (MAE), mean squared error (MSE), coefficient of determination (R^2), and average error.

<https://doi.org/10.1371/journal.pcbi.1010988.t001>

the surrogate was used to compare its outputs to experimental data, with results showing high consistency between the 2, a mean absolute error (MAE) of 2.5×10^{-2} , as well as a 20% reduction in CPU time. The authors noted that this speed-up in computational time might not be high enough to balance the time invested in building the surrogate, suggesting that it is important to approximate a priori the expected improvements in computational time that a surrogate might bring.

In [4], Wang and colleagues used a surrogate model based on a long short-term memory (LSTM) deep neural network to replicate the behaviour of an SDEs model describing the MYC transduction pathways with E2F regulator (MYC/E2F) in cell-cycle progression. The mechanistic model consisted of 10 SDEs and 24 trainable parameters. These parameters were varied using some prespecified ranges, and simulations were run to produce the training data necessary for the surrogate. The output to be predicted by the surrogate is a kernel distribution of the final values of each of the 10 variables. Apart from the high accuracy of the model and the improvement in computational time (Table 1), this analysis shows how surrogate ML models

Table 2. Summary of the performance and methodologies of the ML surrogate models that describe engineering processes with methodologies that can be extended to surrogates of biological models.

Original model description	Surrogate algorithm	Surrogate accuracy	Improvement in computational time	Reference
Simplified land model in the Energy Exascale Earth System Model [6]	Singular value decomposition and neural network	MSE loss: 0.02	3–4 orders of magnitude	[6]
Stochastic analysis of time-dependent PDEs solved using Monte Carlo method [44]	Convolutional autoencoder and feed forward neural network	MSE loss: 0.03	81 times	[44]

Accuracy is reported as mean squared error (MSE).

<https://doi.org/10.1371/journal.pcbi.1010988.t002>

can be used to replicate stochastic systems. Specifically, different runs of the mechanistic model using the same parameters will produce different concentration levels for each molecule, but the distribution of these concentrations is deterministic for a sufficiently large number of runs. This suggests that each combination of parameters leads to a distribution for each molecule, corresponding to the spatial pattern output of the SDE model, which can be predicted by the surrogate neural network.

ML surrogates of PDE systems

Complex dynamical systems that evolve in 2 or more dimensions are often modelled using PDEs. Traditionally, these models are solved numerically using finite element analysis (FEA) methods [45]. In this section, we will review the applicability of ML surrogates to mathematical models described by PDEs for molecular biology processes [4,7] and biomedical systems [5,8–10,13,41,43].

Applications in molecular biology. In molecular biology, surrogate models based on LSTM neural networks [4,46] were built to predict pattern formation in *E.coli* programmed by a synthetic gene circuit [25] represented as the spatial distribution of different molecules. The LSTM took as input the parameters of the mechanistic model and was trained to predict 2 outputs: the logarithm of the peak value of the profile of different molecules and their normalised profile. The authors reported a 30,000-fold computational acceleration [4], the LSTM being successfully used to identify new patterns by screening 10^8 parameter sets in 12 days (compared to thousands of years which is how long it would have taken for the PDE model to achieve this). To improve the robustness of the ML model, the authors also proposed a reliability metric based on a voting system across different neural networks trained in parallel. This was an important addition to surrogate modelling that can prove particularly useful in cases when the surrogate is uncertain about a prediction, since the mechanistic model can be run instead.

In [7], Renardy and colleagues presented a technique based on polynomial surrogates using a Legendre polynomial basis that was applied to a spatial model of pheromone-induced cell polarisation of budding yeast. Once the polynomial surrogate was fit, it was used to compute parameter sensitivities and perform rapid Bayesian parameter inference. Using the surrogate, it was possible to run simulations that would take approximately 200 years to run using the mechanistic model. Furthermore, the surrogate facilitated the convergence for the distribution of 15 parameters in only a few hours using Bayesian inference.

Applications in organ modelling and physiology. Biomedical engineering is a field where surrogate models have been built extensively over the past decade, with a particular focus on biophysical models of the heart. Modelling myocardial properties that can help in making real-time clinical decisions could contribute to understanding and treating heart diseases [47]. Several mathematical models of the myocardium could be used for these aims. However, most are restricted by their high computational demand. Several studies suggest that these limitations can be addressed by implementing surrogate models based on different ML algorithms [5,8–10,14,15,41,48,49].

For example, in [8], Liang and colleagues built an ML surrogate of the FEA method to estimate the zero-pressure geometry of the human thoracic aorta. The input (i.e., a pair of shapes) and output were first encoded as a set of scalars using principal component analysis (PCA). Then, the nonlinear mapping between the encoded input and output was performed using a feed-forward fully connected neural network. Lastly, the output was decoded again using PCA. It was shown that ML surrogates could enable real-time applications of the model, with prediction time under 1 s and an average mean absolute error of 0.533 mm. A similar approach

was used by Liang and colleagues [9], the main difference being that here the model takes 1 shape as input, while in [8] a pair of input shapes was used.

Two deep learning approaches were tested to build a surrogate that predicts the point-wise distribution of stress on the arterial walls under atherosclerosis in [5]. The inputs of the surrogate model were parameters describing the geometry and arterial pressure, and the outputs were point-wise stress distributions. The performance of a feed-forward neural network was compared against that of a convolutional neural network, with the first outperforming the second. Similarly to other studies [10,48], the authors performed a features' importance analysis by adjusting 1 input feature at a time and studying the impact of these changes on the accuracy of the deep learning model. This approach revealed expected correlations between arterial pressure and stress, but also less obvious ones such as the fact that lipid pool information had more impact on maximum stress compared to calcium deposits. This suggests that besides their predictive power, ML surrogates can also unravel some dynamics of the system that have not been studied previously.

Cai and colleagues [41] also used simulations of the LV diastolic filling with the aim of estimating model parameters. Features were first projected into a lower dimensional space, and 3 different ML models (K-nearest neighbour, XGBoost, and a multilayer perceptron) were tested to assess how well they learn the pressure-volume and pressure-strain relationships. The computational cost of simulations was reduced by 3 to 4 orders of magnitude when using the ML surrogate. Davies and colleagues [13] used 2 interpolation methods and 2 loss functions to estimate the material properties of a healthy volunteer's left ventricle using only non-invasive data. Minimising the loss between the biomechanical model's output and the emulator produced an estimate of the unknown parameters. Two loss functions were used: the Euclidean loss function that assumes that the outputs are independent and a Mahalanobis distance-based loss function that allows for correlation across outputs. The best results were achieved using local Gaussian process interpolation and the Euclidean loss function. The reported mean square error (MSE) was 0.0001, and the computational time was reduced by approximately 3 orders of magnitude, from weeks to a quarter of an hour.

Another proof-of-concept for the usability of surrogate modelling assessed the applicability of ML models to emulate 2 physiology mathematical models, Small and HumMod [42,43]. Support vector machine (SVM) regression models were used to map the parameter samples to the drop in mean arterial pressure. The accuracy of these surrogates was calculated with respect to the drop in mean arterial pressure observed after running the original mathematical model. Further error analysis showed that there was no significant difference between the performance of the ML model and the mechanistic one. The authors also compared the time complexity of the 2 approaches and showed that the ML model could make predictions approximately 6 orders of magnitude faster than both dynamical models.

Besides the improvements in computational demand, the studies presented in this section also address other important modelling aspects such as building surrogates of stochastic models, implementing reliability metrics [4], performing parameter sensitivity, inference [7], and feature importance analysis [5]. Furthermore, since biological systems are often high dimensional, dimensionality reduction is another important modelling aspect that has been combined with surrogate-based ML models in studies describing both biological [8,9,41] and other engineering systems [6,44]. To help with deciding whether such analysis can bring value to a study and understand what are the options when deciding on the algorithms and techniques to be used, we will next review some technical aspects that can help to design optimal models.

Building, training, and using ML surrogates

Each of the aforementioned studies has its own set of limitations. Some of these are domain-specific and rely heavily on the knowledge of domain experts. For example, some methods cannot be used in a clinical setting yet because they were only trained on myocardial models coming from a patient with specific characteristics, and hence, they do not include inter-patient variability. Other limitations and design matters are more general, being common across surrogate models regardless of their application area. These technical aspects addressing the design and training of ML surrogates are discussed below.

Active learning

Given that surrogates are built to avoid running expensive simulations many times, it is important to minimise the number of simulations needed to train the ML model while keeping them as informative as possible. In active learning, a model can choose the data that it will learn from next, based on the information it gained from previous training examples. A summary of the process combining surrogate modelling and active learning is shown in Fig 2. Active learning has been applied together with surrogate models in engineering studies where the original mechanistic models were based on PDEs [50,51]. For example, in [50], Pestourie and colleagues built a neural network-based surrogate for the PDE model representing the Maxwell equations for composite materials and used an active learning algorithm that selects new training points from the parameter space where the estimated model error was higher. This error

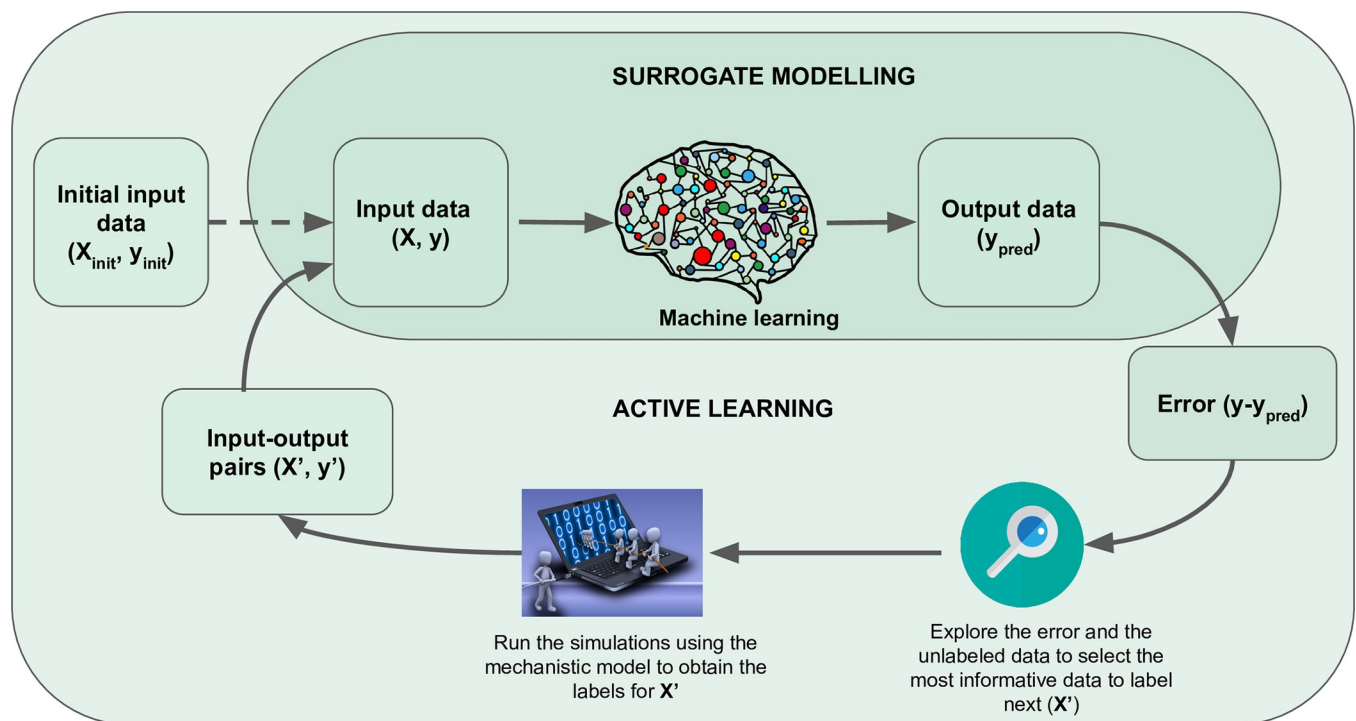


Fig 2. Schematic representation of how active learning and ML surrogates can work together. Initially, an ML model is trained on a set of data generated by some initial simulations of a mechanistic model (X_{init}, y_{init}), which are equivalent to (X, y) for this initial step. The ML model is used to make predictions (y_{pred}). The estimated error between the prediction of the mechanistic model (y) and that of the ML model (y_{pred}) is used to select a subset from all the possible input data that has not been used to make predictions using the mechanistic model in the past (X'). The mechanistic model is run using X' as input to obtain a new set of input-output pairs (X, y) , equivalent to the newly generated (X', y') , that when included in the ML pipeline are expected to reduce the estimated error ($y - y_{pred}$).

<https://doi.org/10.1371/journal.pcbi.1010988.g002>

was recalculated after the training set was updated with new data obtained by running the simulations using the mechanistic model. The active learning approach was compared to a baseline where the training set was randomly sampled from the mechanistic model's parameter space. The active learning surrogate matched the numerical integration result more closely, using 1 order of magnitude less training data compared to the surrogate trained on randomly sampled points.

Lye and colleagues [51] used deep learning surrogates and active learning to solve the constrained optimisation problem of 3 systems: optimal control problem for a nonlinear ODE, parameter identification for the heat equation, and shape optimisation of airfoils subject to the Euler equations. The algorithm presented is called iterative surrogate model optimization (ISMO), where a deep neural network queries a standard optimisation algorithm, quasi-Newton approximation, to provide training examples that will minimise its error. The ISMO algorithm outperformed the purely deep neural network surrogate in terms of error decay and robustness to parameter change and the standard optimisation algorithm for aerodynamic shape optimisation by more than an order of magnitude [51]. Other studies such as [52] presented strategies for building surrogate models with active learning using iterative parallel computations on single-core, multi-core, and multi-node architectures. These approaches can further speed up the modelling process.

Designing the surrogate ML model

Choosing the right ML model to build the surrogate is an essential step in the process and it can be approached in several ways. First, the choice depends on what is the output to be predicted. In most of the use cases presented in Tables 1 and 2, the surrogate was used to predict a continuous output. This is the main reason why the corresponding ML algorithms are regression models. However, the studies reviewed above show that there is no consensus regarding the best regression algorithms to use. Tables 1 and 2 show that neural network methods perform very well, with an R^2 of up to 0.99. However, they need a significant amount of data to be trained, meaning that more simulations of the mechanistic model need to be run, and are lacking explainability. On the other hand, decision tree-based methods have the advantage of being interpretable to some degree since they can output the features' ranking, and perform similarly to the deep learning models in [41]. Algorithms such as Gaussian processes are both interpretable and can estimate the uncertainty in the predictions, which is why they are preferred in some cases.

If the output to be predicted by the surrogate is a discrete or categorical value, classification models are more appropriate. As in the case of regression, choosing the best algorithm depends on several factors such as the dimensionality and the amount of data available, whether uncertainty quantification and interpretability are important for the problem being addressed, and also how much technical ML knowledge the person building the surrogate has. These last 2 aspects are addressed in more detail in the sections regarding model usability and interpretability.

Since most of the mechanistic models presented here describe the dynamics of different systems, it is expected that an increasing number of future models will be based on time-series data. Such data often needs to be treated differently compared to tabular data. With the rise of computer power and deep learning techniques, there has been a lot of progress in the field of time-series analysis and prediction. Several reviews outline the state of the art when it comes to time-series forecasting algorithms [53–55] and time-series classification algorithms [56–58], some with the aim to make them interpretable [59–61].

One aspect that was mentioned in many of the studies presented in the previous section is related to the dimensionality of the data. When the input or output data is high dimensional,

before training any ML model, it is good practice to apply some dimensionality reduction techniques [62,63] in order to address the curse of dimensionality issue [64] and overfitting. In fact, this was done in some of the studies presented in Tables 1 and 2 [6,8,9,41,44]. Another approach when dealing with high-dimensional data is to choose the inputs/outputs that the ML model will be trained on based on expert knowledge. This is less analytical, but in some cases, it can be more appropriate. When the output to be predicted by an ML model is multidimensional, it is also possible to build the model such that it predicts multiple outputs. The feasibility of this approach depends on the dimensionality of the mechanistic model. In [65], Xu and colleagues reviewed different methods and challenges regarding multi-output ML approaches, focusing on assessing the algorithms based on volume, velocity, variety, and veracity, all being important characteristics for models of biological and medical systems.

Another important aspect to be considered when it comes to the design of ML surrogates is the quantity of data used for training. Training data can be augmented by running more simulations (potentially using active learning to optimise this process), by using analytical data augmentation techniques, or when possible by adding matching experimental data to the synthetic data already used. To the best of our knowledge, the latter has not been done before. However, augmenting experimental data with synthetic one was done in [66], and in the training phase of their ML model, it had a positive impact on the performance. This data augmentation technique may not improve the accuracy of the surrogate when compared to the mechanistic model, but it may help the surrogate outperform the mechanistic model when tested against experimental observations, therefore, making the surrogate model more generalisable.

Different approaches may be considered when the mechanistic models to be emulated are not fully deterministic. Surrogate models have been used to approximate stochastic mechanistic models, and it was shown that if sufficient simulations are run, the distribution of the output of these models is approximately deterministic [4]. Another approach for building surrogates of stochastic models is to include the random seed that was used for the simulations as an input when training the ML model [67].

Model usability

To train the ML surrogates, the user needs to be able to run the mechanistic model and train an ML model. Depending on the usability of the original model, its complexity and the user's knowledge regarding ML, this could take more time than running the computationally expensive mechanistic model [7]. This suggests that it would be highly beneficial to include a reproducibility metric as part of surrogate modelling studies. For the initial training of the surrogate model to be as efficient as possible, the mechanistic models should have clear instructions on how to simulate them under different conditions. Furthermore, the training of the ML model should be accessible to non-experts. Once the training data is available, this is already possible using tools such as AutoML [68] or TPOT [69–71]. However, despite the ability of these tools to optimise for the ML model with the highest accuracy, they can limit the freedom of the user when it comes to designing the training process, for example, by making it challenging to control overfitting.

The other important aspect that needs to be addressed when we discuss usability and reproducibility is how easy it is to deploy the already-built surrogate, to re-train it, or slightly modify its scope. We believe that as surrogate modelling will be used more widely and becomes part of experimental or clinical pipelines, it is important to think about its tunability. Therefore, the code used to build the surrogates should be publicly available, well structured, and the ML pipeline presented clearly.

Interpretability

Surrogate ML models can also help explain the behaviour of dynamical systems [5,10,48]. With the recent progress in the area of explainable artificial intelligence, once predictions are made, it becomes possible to interpret for example whether all input data impacts the prediction [72]. Furthermore, it is possible to understand which features influence the prediction the most and quantify this impact by investigating whether an increase in the value of one feature changes the prediction, and in the case of regression models whether the prediction is generally increased or decreased. Such methods could outline some behaviours of the system that were not previously known, especially when experimental data are used to train as well. In general, to make sure the results are robust, it can be helpful to apply different explainability methods and compare their results.

Above, we described the way surrogate machine learning models have been used in the literature and how the design and usability of such models can be enhanced. Using the information acquired from these sections, we propose future avenues for applying surrogate machine learning models to industrially relevant bioengineering.

Further applications of ML-based surrogates in bioengineering

Given the potential of metabolic and whole-cell models in designing novel renewable biofuels [73,74] and drugs [75], as well as their versatility for minimal genome design [76,77], we further present our vision regarding the applicability of surrogate modelling for these types of mechanistic models.

Metabolism is among the most complex processes taking place in a cell. Genome-scale metabolic models include all the known information about the metabolism of an organism, such as genes, enzymes, reactions, and metabolites [78]. These models can be used not only to predict metabolic fluxes but also to understand genotype–phenotype interactions. In addition, they can have a significant impact on understanding strain development for the production of bio-based materials and chemicals, drug targeting, predictions of enzyme function, and modelling interactions among different cells [79]. Given the system-level complexity of these techniques, the models often end up containing thousands of genes, metabolites, and reactions that interact with each other. Metabolic kinetic models are even more computationally expensive since they predict the temporal behaviour of the process and they combine multi-omics data sets with reaction network models [80]. Some models require up to 7 h for 1 simulation, especially when a protein expression network is included [81]. This computational problem is amplified when complex organisms are modelled or several simulations have to be run.

Often, metabolic models are used as part of a design–build–test–learn (DBTL) pipeline [82], corresponding to multiple combinations of inputs (Fig 3). This frequently involves a significant number of trial and error experiments, suggesting that ML surrogates of metabolic models would be particularly useful for such cases. For example, the ML surrogates can be trained on the initial state of the input variables of the mechanistic model and/or the parameters of the model, with the target variable being the desired phenotype to be predicted (a specific titer, rate, yield, or product). Once the training phase is completed, the surrogate can be used to approximate the original metabolic model. One of the challenges that may occur when implementing this framework is caused by the high dimensionality of metabolic models. This suggests that if the initial conditions are used as input for the ML model, it will be necessary to run several simulations to cover most of the variables' space. According to [83,84], at least 4 to 5 times as many simulations as the number of variables are needed to avoid the curse of dimensionality. In the cases when this is possible, it can also lead to the discovery of interesting dynamics of the system since an explainable ML model can show which parameters and

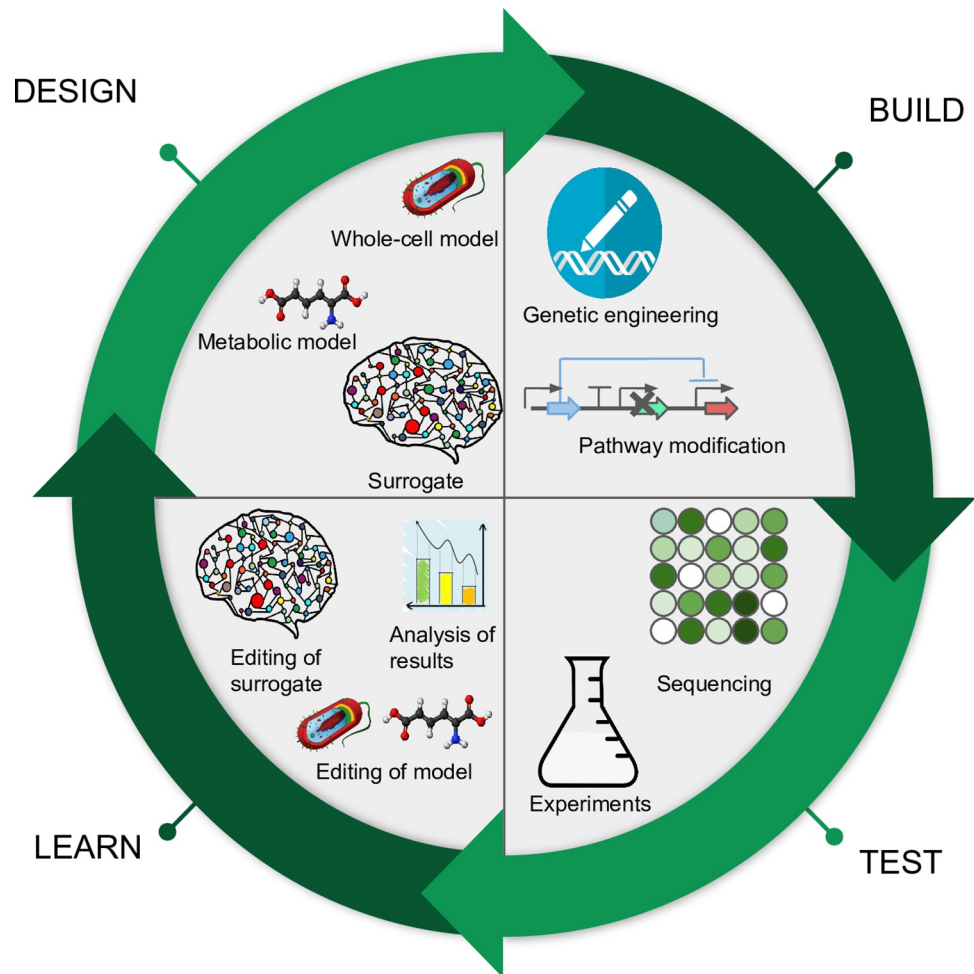


Fig 3. An example of the DBTL pipeline where the metabolic or whole-cell models can be replaced by surrogate models.

<https://doi.org/10.1371/journal.pcbi.1010988.g003>

variables are influencing the phenotype the most. However, in other cases, running a high number of mechanistic simulations might defeat the purpose of building a surrogate model. For such situations, it is possible to reduce the dimension of the input by applying different dimensionality reduction techniques [85] or by manually selecting only the variables that are known to influence the desired phenotype [12].

Whole-cell models are mathematical models that include and link all the well-annotated genes and processes of a cell. Two such models have been published to date, for *Mycoplasma genitalium* [2] and for *Escherichia coli* [3,86], and more are underway [87]. The completeness of these models makes them particularly powerful since, when used in a DBTL pipeline, they facilitate the study and design of interactions among different cellular processes, something that a metabolic model alone cannot achieve. Similarly to metabolic models, whole-cell models have already been used for in silico minimal genome design [77], and we anticipate that they will change the paradigm for metabolic engineering and development of microbial chassis [88]. These models add some extra levels of complexity to the genome-scale metabolic models and therefore are even more computationally expensive [89], with a simulation time of 15 min to 24 h per cell (on a desktop computer). This makes applications that involve multiple cells

growing over multiple generations prohibitively expensive due to their high computational time.

ML surrogate models can represent a strategy to address this challenge. For example, the input to the ML model can be defined as a subset of the initial conditions and parameters of the model, and the output as the phenotype to be predicted. This can be a continuous variable such as the growth rate of a cell, the production, titer or yield of different metabolites, or binary indicating, for example, whether a cell divides or not. Similarly to metabolic models, whole-cell models can have thousands of candidate variables that can be used as input to the ML model. As mentioned before, these can be reduced using dimensionality reduction techniques [85] or prior knowledge about the process under investigation [12].

Conclusion

There is a growing number of studies in the literature showing how ML surrogates can be used to emulate mechanistic models of biological processes, both at the molecular and macroscopic levels. These show that, besides the performance of the surrogate models in terms of accuracy compared to numerical integration, and improvement in computational speed, it is also beneficial to consider other design aspects. First, it is important to assess whether the mechanistic model is complex enough to invest the time in building a surrogate [7]. The design of the protocol for obtaining the training data of the ML surrogate should consider aspects such as stochasticity and whether active learning could bring any value [4,50,51,67]. Furthermore, dimensionality reduction of the inputs and/or outputs of the ML surrogate [6,8,9,41,44] and parameter sensitivity analysis [7] not only can help to optimise the performance of the model, but also to unravel some information about the dynamics of the system.

Author Contributions

Conceptualization: Ioana M. Gherman, Zahraa S. Abdallah, Wei Pang, Thomas E. Gorochofski, Claire S. Grierson, Lucia Marucci.

Funding acquisition: Ioana M. Gherman, Thomas E. Gorochofski, Lucia Marucci.

Investigation: Ioana M. Gherman.

Project administration: Lucia Marucci.

Supervision: Zahraa S. Abdallah, Wei Pang, Thomas E. Gorochofski, Claire S. Grierson, Lucia Marucci.

Visualization: Ioana M. Gherman.

Writing – original draft: Ioana M. Gherman.

Writing – review & editing: Ioana M. Gherman, Zahraa S. Abdallah, Wei Pang, Thomas E. Gorochofski, Claire S. Grierson, Lucia Marucci.

References

1. Fuller A, Fan Z, Day C, Barlow C. Digital twin: Enabling technologies, challenges and open research. *IEEE Access*. 2020; 8:108952–108971.
2. Karr JR, Sanghvi JC, Macklin DN, Gutschow MV, Jacobs JM, Bolival B Jr, et al. A whole-cell computational model predicts phenotype from genotype. *Cell*. 2012; 150(2):389–401. <https://doi.org/10.1016/j.cell.2012.05.044> PMID: 22817898
3. Macklin DN, Ahn-Horst TA, Choi H, Ruggero NA, Carrera J, Mason JC, et al. Simultaneous cross-evaluation of heterogeneous *E. coli* datasets via mechanistic simulation. *Science*. 2020; 369(6502). <https://doi.org/10.1126/science.aav3751> PMID: 32703847

4. Wang S, Fan K, Luo N, Cao Y, Wu F, Zhang C, et al. Massive computational acceleration by using neural networks to emulate mechanism-based biological models. *Nat Commun.* 2019; 10(1):1–9.
5. Madani A, Bakhty A, Kim J, Mubarak Y, Mofrad MR. Bridging finite element and machine learning modeling: stress prediction of arterial walls in atherosclerosis. *J Biomech Eng.* 2019; 141 (8). <https://doi.org/10.1115/1.4043290> PMID: 30912802
6. Lu D, Ricciuto D. Efficient surrogate modeling methods for large-scale Earth system models based on machine-learning techniques. *Geosci Model Dev.* 2019; 12(5):1791–1807.
7. Renardy M, Yi TM, Xiu D, Chou CS. Parameter uncertainty quantification using surrogate models applied to a spatial model of yeast mating polarization. *PLoS Comput Biol.* 2018; 14(5):e1006181. <https://doi.org/10.1371/journal.pcbi.1006181> PMID: 29813055
8. Liang L, Liu M, Martin C, Sun W. A machine learning approach as a surrogate of finite element analysis-based inverse method to estimate the zero-pressure geometry of human thoracic aorta. *Int J Numer Methods Biomed Eng.* 2018; 34(8):e3103. <https://doi.org/10.1002/cnm.3103> PMID: 29740974
9. Liang L, Liu M, Martin C, Sun W. A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis. *J R Soc Interface.* 2018; 15(138):20170844. <https://doi.org/10.1098/rsif.2017.0844> PMID: 29367242
10. Dabiri Y, Van der Velden A, Sack KL, Choy JS, Kassab GS, Guccione JM. Prediction of left ventricular mechanics using machine learning. *Front Phys.* 2019; 7:117. <https://doi.org/10.3389/fphys.2019.00117> PMID: 31903394
11. Gao H, Wang H, Berry C, Luo X, Griffith BE. Quasi-static image-based immersed boundary-finite element model of left ventricle under diastolic loading. *Int J Numer Methods Biomed Eng.* 2014; 30 (11):1199–1222. <https://doi.org/10.1002/cnm.2652> PMID: 24799090
12. Stolfi P, Castiglione F. Emulating complex simulations by machine learning methods. *BMC Bioinform.* 2021; 22(14):1–14. <https://doi.org/10.1186/s12859-021-04354-7> PMID: 34772335
13. Davies V, Noè U, Lazarus A, Gao H, Macdonald B, Berry C, et al. Fast parameter inference in a bio-mechanical model of the left ventricle by using statistical emulation. *J R Stat Soc Ser C Appl Stat.* 2019; 68(5):1555–1576. <https://doi.org/10.1111/rssc.12374> PMID: 31762497
14. Noè U, Lazarus A, Gao H, Davies V, Macdonald B, Mangion K, et al. Gaussian process emulation to accelerate parameter estimation in a mechanical model of the left ventricle: a critical step towards clinical end-user relevance. *J R Soc Interface.* 2019; 16(156):20190114. <https://doi.org/10.1098/rsif.2019.0114> PMID: 31266415
15. Di Achille P, Harouni A, Khamzin S, Solovyova O, Rice JJ, Gurev V. Gaussian process regressions for inverse problems and parameter searches in models of ventricular mechanics. *Front Physiol.* 2018; 9:1002. <https://doi.org/10.3389/fphys.2018.01002> PMID: 30154725
16. Doherty J, Christensen S. Use of paired simple and complex models to reduce predictive bias and quantify uncertainty. *Water Resour Res.* 2011; 47(12).
17. Matott LS, Rabideau AJ. Calibration of complex subsurface reaction models using a surrogate-model approach. *Adv Water Resour.* 2008; 31(12):1697–1707.
18. Young PC, Ratto M. Statistical emulation of large linear dynamic models. *Technometrics.* 2011; 53 (1):29–43.
19. Motta S, Pappalardo F. Mathematical modeling of biological systems. *Brief Bioinform.* 2013; 14(4):411–422. <https://doi.org/10.1093/bib/bbs061> PMID: 23063928
20. Helms V. Principles of computational cell biology: from protein complexes to cellular networks. John Wiley & Sons; 2018.
21. Soheilypour M, Mofrad MR. Agent-based modeling in molecular systems biology. *Bioessays.* 2018; 40 (7):1800020. <https://doi.org/10.1002/bies.201800020> PMID: 29882969
22. Wong JV, Yao G, Nevins JR, You L. Viral-mediated noisy gene expression reveals biphasic E2f1 response to MYC. *Mol Cell.* 2011; 41(3):275–285. <https://doi.org/10.1016/j.molcel.2011.01.014> PMID: 21292160
23. Lee TJ, Yao G, Bennett DC, Nevins JR, You L. Stochastic E2F activation and reconciliation of phenomenological cell-cycle models. *PLoS Biol.* 2010; 8(9):e1000488. <https://doi.org/10.1371/journal.pbio.1000488> PMID: 20877711
24. Yi TM, Kitano H, Simon MI. A quantitative characterization of the yeast heterotrimeric G protein cycle. *Proc Natl Acad Sci U S A.* 2003; 100(19):10764–10769. <https://doi.org/10.1073/pnas.1834247100> PMID: 12960402
25. Cao Y, Ryser MD, Payne S, Li B, Rao CV, You L. Collective space-sensing coordinates pattern scaling in engineered bacteria. *Cell.* 2016; 165(3):620–630. <https://doi.org/10.1016/j.cell.2016.03.006> PMID: 27104979

26. Yi TM, Chen S, Chou CS, Nie Q. Modeling yeast cell polarization induced by pheromone gradients. *J Stat Phys*. 2007; 128(1):193–207.
27. Cootes TF, Taylor CJ, Cooper DH, Graham J. Active shape models-their training and application. *Comput Vis Image Underst*. 1995; 61(1):38–59.
28. An G, Fitzpatrick B, Christley S, Federico P, Kanarek A, Neilan RM, et al. Optimization and control of agent-based models in biology: a perspective. *Bull Math Biol*. 2017; 79(1):63–87. <https://doi.org/10.1007/s11538-016-0225-6> PMID: 27826879
29. Ermentrout GB, Edelstein-Keshet L. Cellular automata approaches to biological modeling. *J Theor Biol*. 1993; 160(1):97–133. <https://doi.org/10.1006/jtbi.1993.1007> PMID: 8474249
30. Xu X, Chen L, He P. A novel ant clustering algorithm based on cellular automata. *Web Intell Agent Syst*. 2007; 5(1):1–14.
31. Smolders G, Van der Meij J, Van Loosdrecht M, Heijnen J. Model of the anaerobic metabolism of the biological phosphorus removal process: stoichiometry and pH influence. *Biotechnol Bioeng*. 1994; 43(6):461–470. <https://doi.org/10.1002/bit.260430605> PMID: 18615742
32. Taymaz-Nikerel H, Borujeni AE, Verheijen PJ, Heijnen JJ, van Gulik WM. Genome-derived minimal metabolic models for *Escherichia coli* MG1655 with estimated in vivo respiratory ATP stoichiometry. *Biotechnol Bioeng*. 2010; 107(2):369–381. <https://doi.org/10.1002/bit.22802> PMID: 20506321
33. Hwang M, Garbey M, Berceci SA, Tran-Son-Tay R. Rule-based simulation of multi-cellular biological systems—a review of modeling techniques. *Cell Mol Bioeng*. 2009; 2(3):285–294. <https://doi.org/10.1007/s12195-009-0078-2> PMID: 21369345
34. Heimann T, Meinzer HP. Statistical shape models for 3D medical image segmentation: a review. *Med Image Anal*. 2009; 13(4):543–563. <https://doi.org/10.1016/j.media.2009.05.004> PMID: 19525140
35. Liang L, Liu M, Martin C, Elefteriades JA, Sun W. A machine learning approach to investigate the relationship between shape features and numerically predicted risk of ascending aortic aneurysm. *Biomech Model Mechanobiol*. 2017; 16(5):1519–1533. <https://doi.org/10.1007/s10237-017-0903-9> PMID: 28386685
36. Dabiri Y, Sack KL, Shaul S, Sengupta PP, Guccione JM. Relationship of transmural variations in myofiber contractility to left ventricular ejection fraction: implications for modeling heart failure phenotype with preserved ejection fraction. *Front Physiol*. 2018; 9:1003. <https://doi.org/10.3389/fphys.2018.01003> PMID: 30197595
37. Baillargeon B, Costa I, Leach JR, Lee LC, Genet M, Toutain A, et al. Human cardiac function simulator for the optimal design of a novel annuloplasty ring with a sub-valvular element for correction of ischemic mitral regurgitation. *Cardiovasc Eng Technol*. 2015; 6(2):105–116. <https://doi.org/10.1007/s13239-015-0216-z> PMID: 25984248
38. Sack KL, Aliotta E, Ennis DB, Choy JS, Kassab GS, Guccione JM, et al. Construction and validation of subject-specific biventricular finite-element models of healthy and failing swine hearts from high-resolution DT-MRI. *Front Physiol*. 2018; 9:539. <https://doi.org/10.3389/fphys.2018.00539> PMID: 29896107
39. Wang H, Gao H, Luo X, Berry C, Griffith B, Ogden R, et al. Structure-based finite strain modelling of the human left ventricle in diastole. *Int J Numer Method Biomed Eng*. 2013; 29(1):83–103. <https://doi.org/10.1002/cnm.2497> PMID: 23293070
40. Gao H, Li W, Cai L, Berry C, Luo X. Parameter estimation in a Holzapfel–Ogden law for healthy myocardium. *J Eng Math*. 2015; 95(1):231–248. <https://doi.org/10.1007/s10665-014-9740-3> PMID: 26663931
41. Cai L, Ren L, Wang Y, Xie W, Zhu G, Gao H. Surrogate models based on machine learning methods for parameter estimation of left ventricular myocardium. *R Soc Open Sci*. 2021; 8(1):201121. <https://doi.org/10.1098/rsos.201121> PMID: 33614068
42. Hester R, Brown A, Husband L, Ilescu R, Pruett WA, Summers RL, et al. HumMod: a modeling environment for the simulation of integrative human physiology. *Front Physiol*. 2011; 2:12. <https://doi.org/10.3389/fphys.2011.00012> PMID: 21647209
43. Pruett WA, Hester RL. The creation of surrogate models for fast estimation of complex model outcomes. *PLoS ONE*. 2016; 11(6):e0156574. <https://doi.org/10.1371/journal.pone.0156574> PMID: 27258010
44. Nikolopoulos S, Kalogeris I, Papadopoulos V. Non-intrusive surrogate modeling for parametrized time-dependent partial differential equations using convolutional autoencoders. *Eng Appl Artif Intel*. 2022; 109:104652.
45. Segerlind LJ. Applied finite element analysis. John Wiley & Sons; 1991.
46. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997; 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735> PMID: 9377276
47. Gao H, Mangion K, Carrick D, Husmeier D, Luo X, Berry C. Estimating prognosis in patients with acute myocardial infarction using personalized computational heart models. *Sci Rep*. 2017; 7(1):1–14.

48. Longobardi S, Lewalle A, Coveney S, Sjaastad I, Espe EK, Louch WE, et al. Predicting left ventricular contractile function via Gaussian process emulation in aortic-banded rats. *Philos Trans R Soc A*. 2020; 378(2173):20190334. <https://doi.org/10.1098/rsta.2019.0334> PMID: 32448071
49. Noè U, Chen W, Filippone M, Hill N, Husmeier D. Inference in a partial differential equations model of pulmonary arterial and venous blood circulation using statistical emulation. In: *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*. Springer; 2016. p. 184–198.
50. Pestourie R, Mroueh Y, Nguyen TV, Das P, Johnson SG. Active learning of deep surrogates for PDEs: application to metasurface design. *npj Comput Mater*. 2020; 6(1):1–7.
51. Lye KO, Mishra S, Ray D, Chandrashekar P. Iterative surrogate model optimization (ISMO): An active learning algorithm for PDE constrained optimization with deep neural networks. *Comput Methods Appl Mech Eng*. 2021; 374:113575.
52. Balaprakash P, Gramacy RB, Wild SM. Active-learning-based surrogate models for empirical performance tuning. In: *2013 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE; 2013. p. 1–8.
53. Tealab A. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Comput Inform J*. 2018; 3(2):334–340.
54. Torres JF, Hadjout D, Sebaa A, Martínez-Álvarez F, Troncoso A. Deep Learning for Time Series Forecasting: A Survey. *Big Data*. 2021; 9(1):3–21. <https://doi.org/10.1089/big.2020.0159> PMID: 33275484
55. Deb C, Zhang F, Yang J, Lee SE, Shah KW. A review on time series forecasting techniques for building energy consumption. *Renew Sustain Energy Rev*. 2017; 74:902–924.
56. Bagnall A, Lines J, Bostrom A, Large J, Keogh E. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min Knowl Discov*. 2017; 31(3):606–660. <https://doi.org/10.1007/s10618-016-0483-9> PMID: 30930678
57. Ruiz AP, Flynn M, Large J, Middlehurst M, Bagnall A. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min Knowl Discov*. 2021; 35(2):401–449. <https://doi.org/10.1007/s10618-020-00727-3> PMID: 33679210
58. Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller PA. Deep learning for time series classification: a review. *Data Min Knowl Discov*. 2019; 33(4):917–963.
59. Assaf R, Schumann A. Explainable Deep Neural Networks for Multivariate Time Series Predictions. In: *IJCAI*; 2019. p. 6488–6490.
60. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*; 2017. p. 618–626.
61. Nguyen TT, Le Nguyen T, Ifrim G. A Model-Agnostic Approach to Quantifying the Informativeness of Explanation Methods for Time Series Classification. In: *International Workshop on Advanced Analytics and Learning on Temporal Data*. Springer; 2020. p. 77–94.
62. Sorzano CO, Vargas J, Montano AP. A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*. 2014.
63. Reddy GT, Reddy MP, Lakshmana K, Kaluri R, Rajput DS, Srivastava G, et al. Analysis of dimensionality reduction techniques on big data. *IEEE Access*. 2020; 8:54776–54788.
64. Kppen M. The curse of dimensionality. In *5th online world conference on soft computing in industrial applications (WSC5) 2000* (vol. 1, p. 4–8).
65. Xu D, Shi Y, Tsang IW, Ong YS, Gong C, Shen X. Survey on multi-output learning. *IEEE Trans Neural Netw Learn Syst*. 2019; 31(7):2409–2429. <https://doi.org/10.1109/TNNLS.2019.2945133> PMID: 31714241
66. Costello Z, Martin HG. A machine learning approach to predict metabolic pathway dynamics from time-series multiomics data. *NPJ Syst Biol Appl*. 2018; 4(1):1–14. <https://doi.org/10.1038/s41540-018-0054-3> PMID: 29872542
67. Angione C, Silverman E, Yaneske E. Using machine learning as a surrogate model for agent-based simulations. *PLoS ONE*. 2022; 17(2):e0263150. <https://doi.org/10.1371/journal.pone.0263150> PMID: 35143521
68. Guyon I, Sun-Hosoya L, Boullé M, Escalante HJ, Escalera S, Liu Z, et al. Analysis of the AutoML Challenge series 2015–2018. In: *AutoML. Springer series on Challenges in Machine Learning*; 2019. Available from: <https://www.automl.org/wp-content/uploads/2018/09/chapter10-challenge.pdf>.
69. Olson RS, Bartley N, Urbanowicz RJ, Moore JH. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science. In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016. GECCO '16*. New York, NY, USA: ACM; 2016. p. 485–492. Available from: <http://doi.acm.org/10.1145/2908812.2908918>.

70. Olson RS, Urbanowicz RJ, Andrews PC, Lavender NA, Kidd LC, Moore JH. Automating Biomedical Data Science Through Tree-Based Pipeline Optimization. In: Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30–April 1, 2016, Proceedings, Part I. Springer International Publishing; 2016. p. 123–137.
71. Le TT, Fu W, Moore JH. Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics*. 2020; 36(1):250–256. <https://doi.org/10.1093/bioinformatics/btz470> PMID: 31165141
72. Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf Fusion*. 2020; 58:82–115.
73. Beller HR, Lee TS, Katz L. Natural products as biofuels and bio-based chemicals: fatty acids and isoprenoids. *Nat Prod Rep*. 2015; 32(10):1508–1526. <https://doi.org/10.1039/c5np00068h> PMID: 26216573
74. Chubukov V, Mukhopadhyay A, Petzold CJ, Keasling JD, Martín HG. Synthetic and systems biology for microbial production of commodity chemicals. *NPJ Syst Biol Appl*. 2016; 2(1):1–11. <https://doi.org/10.1038/npsba.2016.9> PMID: 28725470
75. Ajikumar PK, Xiao WH, Tyo KE, Wang Y, Simeon F, Leonard E, et al. Isoprenoid pathway optimization for Taxol precursor overproduction in *Escherichia coli*. *Science*. 2010; 330(6000):70–74. <https://doi.org/10.1126/science.1191652> PMID: 20929806
76. Wang L, Maranas CD. MinGenome: an in silico top-down approach for the synthesis of minimized genomes. *ACS Synth Biol*. 2018; 7(2):462–473. <https://doi.org/10.1021/acssynbio.7b00296> PMID: 29254336
77. Rees-Garbutt J, Chalkley O, Landon S, Purcell O, Marucci L, Grierson C. Designing minimal genomes using whole-cell models. *Nat Commun*. 2020; 11(1):1–12.
78. Passi A, Tibocha-Bonilla JD, Kumar M, Tec-Campos D, Zengler K, Zuniga C. Genome-Scale Metabolic Modeling Enables In-Depth Understanding of Big Data. *Metabolites*. 2022; 12(1):14.
79. Gu C, Kim GB, Kim WJ, Kim HU, Lee SY. Current status and applications of genome-scale metabolic models. *Genome Biol*. 2019; 20(1):1–8.
80. Islam MM, Schroeder WL, Saha R. Kinetic modeling of metabolism: Present and future. *Curr Opin Syst Biol*. 2021; 26:72–78.
81. Yang L, Ebrahim A, Lloyd CJ, Saunders MA, Palsson BO. DynamicME: dynamic simulation and refinement of integrated models of metabolism and protein expression. *BMC Syst Biol*. 2019; 13(1):1–15.
82. Nielsen J, Keasling JD. Engineering cellular metabolism. *Cell*. 2016; 164(6):1185–1197. <https://doi.org/10.1016/j.cell.2016.02.004> PMID: 26967285
83. Kuo FY, Sloan IH. Lifting the curse of dimensionality. *Not Am Math Soc*. 2005; 52(11):1320–1328.
84. Lawson CE, Martí JM, Radivojevic T, Jonnalagadda SVR, Gentz R, Hillson NJ, et al. Machine learning for metabolic engineering: A review. *Metab Eng*. 2021; 63:34–60. <https://doi.org/10.1016/j.ymben.2020.10.005> PMID: 33221420
85. Espadoto M, Martins RM, Kerren A, Hirata NS, Telea AC. Toward a quantitative survey of dimension reduction techniques. *IEEE Trans Vis Comput Graph*. 2019; 27(3):2153–2173.
86. Ahn-Horst TA, Mille LS, Sun G, Morrison JH, Covert MW. An expanded whole-cell model of *E. coli* links cellular physiology with mechanisms of growth rate control. *NPJ Syst Biol Appl*. 2022 Aug 19; 8(1):30. <https://doi.org/10.1038/s41540-022-00242-9> PMID: 35986058
87. Karr J. Models: Comprehensive computational models of individual cells; 2019. Available from: <https://www.wholecell.org/models/>.
88. Marucci L, Barberis M, Karr J, Ray O, Race PR, de Souza AM, et al. Computer-aided whole-cell design: taking a holistic approach by integrating synthetic with systems biology. *Front Bioeng Biotechnol*. 2020; 8:942. <https://doi.org/10.3389/fbioe.2020.00942> PMID: 32850764
89. Macklin DN, Ruggero NA, Covert MW. The future of whole-cell modeling. *Curr Opin Biotechnol*. 2014; 28:111–115. <https://doi.org/10.1016/j.copbio.2014.01.012> PMID: 24556244