Computer Methods and Programs in Biomedicine 230 (2023) 107345

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Application of machine learning to improve the efficiency of electrophysiological simulations used for the prediction of drug-induced ventricular arrhythmia

Pablo Rodríguez-Belenguer[a,b,1], Karolina Kopańska[a,1], Jordi Llopis-Lorente[c], Beatriz Trenor[c], Javier Saiz[c], Manuel Pastor[a,*]

[a] Research Programme on Biomedical Informatics (GRIB), Department of Medicine and Life Sciences, Universitat Pompeu Fabra, Hospital del Mar Medical Research Institute, Barcelona, Spain
[b] Department of Pharmacy and Pharmaceutical Technology and Parasitology, Universitat de València, Valencia, Spain
[c] Centro de Investigación e Innovación en Bioingeniería (Ci2B), Universitat Politècnica de València, Valencia, Spain

ARTICLE INFO

Article history:
Received 9 August 2022
Revised 16 December 2022
Accepted 7 January 2023

ABSTRACT

Background and Objective: *In silico* prediction of drug-induced ventricular arrhythmia often requires computationally intensive simulations, making its application tedious and non-interactive. This inconvenience can be mitigated using matrices of precomputed simulation results, allowing instantaneous computation of biomarkers such as action potential duration at 90% of the repolarisation (APD_{90}). However, preparing such matrices can be computationally intensive for the method developers, limiting the range of simulated conditions. In this work, we aim to optimise the generation of these matrices so that they can be obtained with less effort and for a broader range of input values.
Methods: Machine learning methods were applied, building models trained with only a small fraction of the originally simulated results. The predictive performances of the models were assessed by comparing their predicted values with the actual simulation results, using percentual mean absolute error and mean relative error, as well as the percentage of data with a relative error below 5%.
Results: Our method obtained highly accurate estimations of the original values, leading to a nearly one hundred-fold decrease in computation time. This method also allows precomputing more complex matrices, describing the effect of more ion channels on the $APD_{90.}$ The best results were obtained by applying Support Vector Machine models, which yielded errors below 1% in most cases. This approach was further validated by predicting the APD_{90} of a set of 12 CiPA compounds and exporting the optimal settings for predicting APD_{90} using a different set of ion channels, always with satisfactory results.
Conclusions: The proposed method effectively reduces the computational effort required to generate matrices of precomputed electrophysiological simulation values. The same approach can be applied in other fields where computationally costly simulations are applied repeatedly using slightly different input values.

1. Introduction

Assessing the arrhythmogenic risk of new drug candidates is an important step in safety studies. The mechanism by which drugs induce ventricular arrhythmias involves their binding to one or multiple ion channels, thereby altering the ionic conductance that controls cardiomyocyte membrane potential [1]. As a result, the form and duration of ventricular action potentials (APs) change, and the net effects can be observed at tissue and organ levels, such as the prolongation of the QT-interval on the surface ECG [2]. A significant prolongation of the QT-interval, is often linked to severe adversities such as early afterdepolarisations (EADs), which can quickly progress to one of the most severe effects of proarrhythmic drugs: the polymorphic ventricular tachycardia known as Torsade de Pointes (TdP) [3].

As the occurrence of TdP historically led to the withdrawal of several marketed drugs, the International Council for Harmonisation of Technical Requirements for Pharmaceuticals for Human Use (ICH) developed standardised guidelines for safety testing of novel

[*] Corresponding author.
E-mail address: manuel.pastor@upf.edu (M. Pastor).
[1] Both authors have contributed equally

https://doi.org/10.1016/j.cmpb.2023.107345

medicines [4]. Resting upon the preclinical ICH S7b guideline [5], the estimation of proarrhythmic risk is done through the integration of results from *in vitro* inhibition assay of the Rapid Delayed Rectifier Potassium Current (I_{Kr}) encoded by the Human Ether-a-go-go-related Gene (hERG) and an *in vivo* animal QT-prolongation study. Following the clinical guideline ICH E14 [6], the potential of a drug to delay ventricular repolarisation is assessed by measuring *in vivo* human QT/QTc interval prolongation.

Indeed, testing drugs in compliance with these regulatory requirements over the last two decades resulted in no further removal of marketed drugs due to ventricular arrhythmia. However, the consideration of *in vitro* effects of drugs on a single ion channel and the application of a conservative cut-off for QT-prolongation is the reason why several potentially useful drug candidates with low toxicity risk are also discarded during the development stages. To provide a more complete description of the cellular mechanisms of drug proarrhythmia, a novel testing paradigm was proposed by the Comprehensive In Vitro Proarrhythmia Assay (CiPA) initiative [7,8]. The CiPA points out that the consideration of drug interactions with other currents along with the hERG is also important for the analysis of ventricular arrhythmia. The main aim behind the CiPA project is to combine *in vitro* measured drug effects on multiples ion channels (I_{Na}, I_{NaL}, I_{Kr}, I_{to}, I_{CaL}, I_{K1}, and I_{Ks}) with computational simulations, such as *in silico* reconstructions of cardiac myocyte electrophysiology, and to compare these results with *in vitro* human stem cell results and human ECG phase 1 clinical trials [9].

Adding *in silico* elements to the cardiac safety testing pipeline has two main advantages, the first being the ability to fill data gaps when experimental results are not yet available at early stages of drug development and the second being an increased analytical accuracy due to the solid mechanistic foundation of the CiPA paradigm [10].

Several works have been published on the implementation of the CiPA based *in silico* simulations for the prediction of ventricular arrhythmia and TdP biomarkers using predicted or experimentally determined drug-induced ion channel inhibition data [11–19]. Computational models of human and animal electrophysiology operate at different biological levels, ranging from a single channel to whole tissue simulations and vary in terms of the degree of complexity and abstraction, the underlying mathematical approaches, and physiological parameters [11]. Although the predictions generated by such models are considered valuable and relevant, they also have limitations related to their usability. Usually, computational safety models are designed based on the subjective scientific interests of the developers and the required efficiency to run on high-performance-computing platforms is seldom reached. But most importantly, the simulation consists of multiple steps, making the prediction process rather tedious [20]. For example, Beattie et al. [21] presented a safety tool based on concentration-effect data for four cardiac ion channels (hERG, NaV1.5, CaV1.2, KCNQ1), in which drug-induced channel inhibition of selected compounds was predicted and used for the computation of QT interval changes in rabbit ventricles using computationally demanding one-dimensional tissue simulation.

To speed up the process, our group developed an *in silico* system that transforms multi-channel blockage into proarrhythmia biomarkers, such as action potential duration at 90% of the repolarisation (APD_{90}), in which the most computationally intensive steps are precomputed, allowing to produce results instantaneously [22,23]. In our system, input values are pre-processed by combining channel-specific half-maximal inhibitory concentration (IC_{50}) and the Hill coefficient for the currents I_{Kr}, I_{Ks}, and I_{CaL} with the concentration of the drug. The APD_{90} prolongation values are then predicted using isolated human ventricular myocyte models as a function of these three input values. Since the calculation can take a considerable time, the predictions are generated by making use of precomputed matrices comprising large sets of possible combinations of input values, each of which is associated with a particular value of the output biomarker. These technical features make the prediction system simple, practical, and rapid [22,23].

Even if storing precomputed data matrices is a very convenient way to obtain predictions interactively, with minimal computational requirements for the end-user, the procedure has the drawback that the preparatory simulations that the method developer needs to run are extremely expensive in terms of computational power and time. This is because accurate predictions can only be produced when the input values cover a wide range of possibilities starting with safe and ending with very toxic representations of drug effects on each considered ion channel. The number of combinations is calculated as X^n, being X the number of possible values considered for each input value and n the count of the input values considered (number of ion channels). This fact imposes a practical upper limit to the number of currents that can be considered since incorporating one more channel multiplies by X the number of simulations to run. Since incorporating additional currents could have substantial benefits, we studied how to overcome these limitations. A potential solution would be to train a machine learning (ML) model with part of the data array and use it to predict the rest of the data array, thereby reducing the number of required simulations. The use of ML in the field of arrhythmia and electrophysiology-oriented research is not new, and the spectrum of published ML applications in this area is very broad [20,21]. For example, classification and regression algorithms can be applied to build models describing the association between the molecular structure and the inhibitory potential of drugs on ion channels [24] or to produce high-level arrhythmogenic risk indicators [25–27]. Another example of the application of ML in combination with *in silico* simulations to improve the predictive results of the arrhythmogenic risk in post-infarction patients was described by Maleckar and colleagues (2020), who simulated the data for the analysis only partially and predicted the rest using ML methods [28].

In this work, we describe an application of ML which aimed only to optimise the generation of precomputed matrices that link input ionic currents with output APD_{90} values. The basic idea was to train a model with a few of the array nodes and to use it to reconstruct the whole array. We show that even a tiny fraction of nodes (5% or less) can produce a very accurate estimation of the values obtained using simulations for the remaining part of the array (95% or more). Therefore, using an ML model can save up to 95% of the computation time and, more importantly, opens the possibility to precompute matrices with more currents that can provide better, more useful predictions. In this work, we compare different machine learning approaches, optimise their parameters, and evaluate the quality of the predictions obtained using different sample sizes to make the most optimal choices for future simulations. Then, we present the best methodological settings and validate our selected model by predicting the APD_{90} for a series of compounds from the CiPA dataset. Lastly, we evaluate the value of our method by simulating a real production scenario where it was applied to a new electrophysiological simulation.

2. Methods

2.1. Data collection for model building

In silico action potential (AP) modelling of the healthy human endocardial cardiomyocyte and APD_{90} measurements were done using a modified version of the widely known model published by O'Hara and colleagues [29]. The modifications were designed to better reproduce the experimental data of drug effects. Briefly, the

AP model modifications included: i) the scaling of the following conductances: I_{Kr} by 1.119, I_{NaL} by 2.274, I_{K1} by 1.414, I_{Ks} by 1.648, I_{CaL} by 1.018, and I_{Na} by 0.4; and ii) a reformulation of the activation and inactivation gates of I_{Na}. For further details about the electrophysiological model, see Llopis-Lorente et al. [16]. Simulations were run with a basic cycle length of 1,000 ms, a stimulus of 1.5-fold the diastolic threshold of amplitude and a duration of 0.5 ms, at physiological temperature (37 °C) and the following extracellular concentrations: $[Na^+]$=140 nM, $[Ca^{2+}]$=1.8 nM and $[K^+]$=5.4 nM. Measurements of APD_{90} under drug effects were done after 500 beats starting from control -no drug- initial conditions.

In this work, we considered the effects of drug action on two combinations of cardiac ion channels. Primarily, aiming to improve the *in silico* modelling tool described by Obiol-Pardo, we considered drug effects on I_{Kr}, I_{Ks} and I_{CaL} currents [22,23]. To evaluate the applicability of our new methodology to other combinations of ionic channels and validate the proposed machine learning methods, we selected the currents I_{Kr}, I_{CaL}, I_{NaL} that were recently described by Llopis-Lorente [16]. Drug effects on the AP were simulated using the simple pore block model [30]. Drug inhibition produced on each channel was simulated by scaling the channel's maximal conductance (g_i) using the standard Hill equation Eq. (1).

$$g_{i,\, drug} = g_i \left[1 + \left(\frac{D}{IC_{50,i}} \right)^h \right]^{-1} \tag{1}$$

where $g_{i,\,drug}$ is channel i's maximal conductance in the presence of the drug, D is the drug concentration, $IC_{50,\,i}$ is the half-maximal inhibitory concentration for that drug, and channel i and h is the Hill coefficient, which represents the number of molecules that are sufficient to block an ion channel.

A wide combination of input values representing the ratio $(\frac{D}{IC_{50}})^h$ for I_{Kr}, I_{Ks}, I_{CaL} was simulated and stored in an array. The array consisted of 3 channel input values: I_{Kr}, I_{Ks}, and I_{CaL}. Each of them represented the logarithm of the ratio $(\frac{D}{IC_{50}})^h$, as described in Eq. (2). For each channel (I_{Kr}, I_{Ks}, I_{CaL}), the input value ranged from −3 to 2.5, with a step increment of 0.1. These values were chosen to cover the properties found in real molecules, avoiding the need to extrapolate the models. Therefore, the simulated array comprised 175,616 instances (56 data points for each current).

$$\text{Input value } = \log_{10}\left(\left[\frac{D}{IC_{50}} \right]^h \right) \tag{2}$$

The output value of the array was the APD_{90}, simulated as described above for each of the input values combinations. For each set of input values, an additional binary variable was included to indicate whether early afterdepolarisations (EAD) occurred during the simulation of that drug (EAD=1) or not (EAD=0). An EAD was defined as any event with a positive voltage gradient ($dV/dt > 0$ mV/ms) after 100 ms from the beginning of the action potential or with a value of membrane voltage at the end of the beat being higher than resting membrane voltage ($Vm > -40$ mV).

The standard use of such array was as follows: for a given compound at a concentration D, Eq. 2 was applied for the three ionic channels (I_{Kr}, I_{Ks}, I_{CaL}). The results of Eq. (2) were rounded to the first decimal and bounded between −3 and 2.5, *i.e.*, if an input value was lower than −3 or higher than 2.5, the value was then transformed to −3 or 2.5. For each combination of the three calculated input values, the corresponding output (APD_{90}) was stored in a three-dimensional result array. For example, a drug with the following IC_{50}s: 1 nM for I_{Kr}, 1000 nM for I_{Ks} and 10 nM for I_{CaL} at a concentration of 1 nM yielded the data point [0, −3, −1], which led to an APD_{90} of 369.16 ms.

Electrophysiological simulations and generation of the APD_{90} array were carried out using MATLAB version R2021b. The table with the APD_{90} values for a wide combination of input values is available online, named “KrKsCaL.xlsx”, on the public repository of the Polytechnic University of Valencia (RIUNET, https://riunet.upv.es/handle/10251/183067).

Table 1
Percentages of data from the original array sampled using four different rates to generate the training, validation, and test series for model building.

Sampling rate	Training series	Validation series	Test series
1/20	5%	5%	90%
1/50	2%	2%	96%
1/100	1%	1%	98%
1/200	0.5%	0.5%	99%

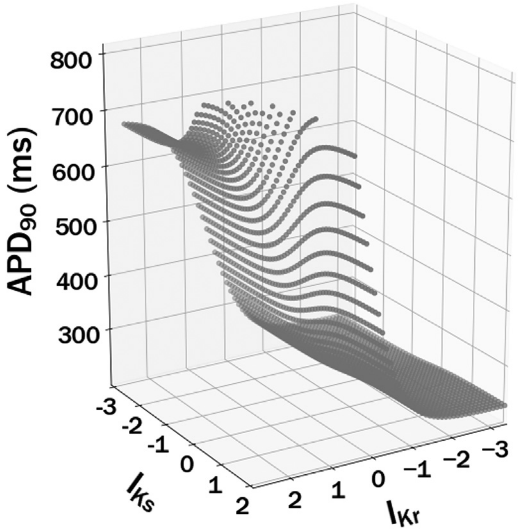

Fig. 1. 3D plot showing the non-linear relationship between the APD_{90} and the input values (I_{Kr} and I_{Ks}) for the simulated data. In this plot, a fixed value of 0.3 was used for I_{CaL}.

2.2. Data pre-processing

We removed from the analysis all data points for which EADs were detected. Also, we applied filters to remove simulation results yielding APDs greater than 1000 ms. These conditions represent repolarisation abnormalities, and the numerical result is considered unreliable. Additionally, data points with an APD_{90} larger than the 3rd quartile plus 1.5 times the interquartile range were considered outliers and removed. This filter removed 1.4% of the data points, with values ranging between 777.59 and 865.47 ms. After the pre-processing, the number of simulation results was reduced to 140,269.

The data array was divided into training, validation, and test sets using four different sampling rates that were used in the models (Table 1). In each case, the training and validation series were extracted by picking the results at regular and pre-defined intervals to guarantee an even distribution of values for fitting and validation along with the explored range of input values. All remaining data were used as test series to evaluate the predictive performance of the models.

2.3. Machine learning algorithms

Figure 1 shows a 3D representation of the APD_{90} values obtained for different combinations of two current pairs (I_{Kr} and I_{Ks}). The APD_{90} values are distributed on a non-linear 2D surface smoothly distributed. This observation suggests that by the application of ML algorithms suitable for processing non-linear data, we could obtain a good model fitting. In this work, we selected three different ML methods: Polynomial Transformation with Ridge regression (PR), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). For each one, we optimised their hyperparameters and validated the models using three partitions and an external test set with selected CiPA compounds.

2.4. Polynomial regression

The PR model was built using polynomial regression Eq. (3), a form of linear regression in which the relationship between the independent and dependent variables is modelled as a polynomial of the n^{th} degree. In this algorithm [31], the polynomial degree increases proportionally to the complexity of the data structure:

$$\hat{y} = b + w_1.x + w_2.x^2 \ldots + w_n.x^n \tag{3}$$

Where ŷ is the target variable, n is the degree of the polynomial, x is the independent variable, w represents the model coefficients, and b is the offset.

To reduce the chance of overfitting the model by selecting a too high polynomial degree, Ridge regression[32] Eq. (4) was applied:

$$J(w,b) = \sum_{i=1}^{M} \left(y_i - b - \sum_{j=1}^{p} w_j \cdot x_{ij} \right)^2 + \alpha \sum_{(j=1)}^{p} w_j^2 \tag{4}$$

Ridge regression, which operates by performing L2 regularisation, penalises the model coefficients by adding the factor (α). The greater the factor α, the greater the impact of the shrinkage penalty, resulting in a larger reduction of the magnitude of model coefficients. Therefore, finding an optimal value for α is particularly important to control model overfit.

2.5. Support vector machine

To build the SVM model, we used a non-linear support vector machine for regression (SVR) which can be explained by a line enclosed between two decision boundaries, where the width between is controlled by the parameter ε [33]. As the data points that lie within the boundaries get assigned a loss of 0, the best value of ε is the one that maximally increases the number of the data points included within. On the other hand, the error is computed using slack variables that quantify the distance from the decision bound aries Ɛ's to the points outside the margin. Support vector machine models strive towards a maximal error reduction as defined in Eq. (5).

$$\text{Minimise } \frac{w^T w}{2} + C \sum_{i=1}^{N} \left(\xi_i + \xi_i{}^* \right) \text{ subject to}$$
$$\begin{cases} y_i - w^T \phi(x_i) - b \le \varepsilon + \xi_i, \\ w^T \phi(x_i) + b - y_i \le \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \ge 0, \; i = 1, \ldots, n \end{cases} \tag{5}$$

ξ_i and $\xi_i{}^*$ are the slack variables, $\|w\|$ represents the Euclidian normalisation of the weight (w) vector. C is a regularisation parameter where the strength of the regularisation is inversely proportional to this parameter. φ(x) is the transformation from input space into feature space, and b is the bias term.

To process non-linear data, support vector regressors perform the kernel trick [34], a method that allows for a representation of the data only through a set of pairwise similarity comparisons between two instances in the input space. More precisely, a kernel function $K(x_i,x_j)$ takes as input the original low dimensional data points (x_i, x_j) and computes a dot product of these data in the transformed high dimensional space, without explicitly determining their coordinates in this feature space. In this work Radial Basis Function (RBF) [35] kernel Eq. (6) was used.

$$K\left(x_i, x_j\right) = e^{-\gamma x_i - x_j{}^2} \tag{6}$$

γ is the parameter of the gaussian kernel and (x_i, x_j) are two selected input instances. In this work, scale mode γ Eq. (7) was selected because it is invariant against the scale of the inputs.

$$Y_{scale\ mode} = \frac{1}{n\ .\ x_{variance}} \tag{7}$$

Where n is the number of features and $\mathrm{x_{variance}}$ corresponds to the variance in the input data.

2.6. Multilayer perceptron

Multilayer perceptron [36] is a feedforward artificial neural network class belonging to the family of supervised machine learning algorithms. The basic structure of an MLP consists of a dot product of the input data (x) with their weights (w) + the bias (b) and of an activation function which in most cases is non-linear Eq. (8). These inputs yield an output of a single neuron.

$$output = f(y) = f\left(\sum_{k=1}^{n} w_k.x_k + b \right) \tag{8}$$

The output obtained from the first neuron is transmitted to the next one through feedforward propagation. In order to reduce the error between the desired output and the predicted output, the weights are updated in a process of backpropagation [37]. The most important hyperparameters that impact the predictive performance of the neural network are hidden layers, activation function [38], learning rate (lr), which controls the step-size in updating the weights, the L2 regularisation parameter penalty alpha (ɑ), and the solver for weight optimisation.

2.7. Evaluation metrics

The three machine learning algorithms were applied to four training series generated with different sampling rates (as shown in Table 1) to build 12 models. The predictive performances of the models were compared using three evaluation metrics: Mean Absolute Error (MAE) Eq. (9), the Mean Relative Error in% (MRE) computed from Relative Error Eq. (10), and the percentage of data with Relative Error (RE) below 5% (non-large data-points error, NLDE). These metrics were used to quantify the differences between predicted and simulated $\mathrm{APD_{90}}$ values and to guarantee that the quantity of the sampled data from the original simulated data array is enough to build a robust ML model. We only consider acceptable the simulations with an RE below 5%.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \left| Y_i - \hat{Y}_i \right| \tag{9}$$

$\hat{Y}_i$ corresponds to the predicted value, Y_i is the real value, and n is the number of data points.

$$RE(\%) = \frac{\left| Y_i - \hat{Y}_i \right|}{Y_i}.100 \tag{10}$$

RE (%) values, computed as a function of $\mathrm{APD_{90,}}$ were plotted for a visual evaluation

2.8. Hyperparameters of all described ML algorithms

Algorithm-specific hyperparameters selected for the optimisation of the ML models are listed in Table 2.

The hyperparameter tuning for the different models aimed to minimise the validation set MAE. We also tested whether the hyperparameters of the three selected algorithms can be optimised

Table 2
Selected hyperparameters for the optimisation of selected ML models.

Internal name	Algorithm	Hyperparameters
PR	Ridge regression with a polynomial transformation	Polynomial degree=[2–15], α=[1.10^{-6} – 10]
SVM	Support Vector Machine Regression	C=[0.1 – 30.10^{5}], kernel=RBF, Υ=scale, $\mathcal{E}$=0.1
MLP	Multilayer Perceptron	Hidden layers=[(50,50,50), (50,100,50), (100,)], learning rate=[constant, adaptative, sgd], solver= Adam, α=[0.05, 0.1,0.5], activation=ReLU

using only the training set or if it requires an additional validation set.

The scripts were developed using Python 3.8. Machine learning models were built and evaluated using standard libraries Scikit-learn [39], NumPy [40], Pandas [41] and Matplotlib [42]. The source code of the scripts used for building and validating the models, together with the datasets described and analysed in this manuscript, are available at GitHub (https://github.com/phi-grib/cardioML) and distributed as open source under GNU GLP-3.0 license.

2.9. Example case study using CiPA compounds

To obtain a more realistic evaluation, focused on the range of IC_{50} observed in commonly used drugs for the I_{Kr}, I_{Ks}, and I_{CaL} channels, as well as drug concentrations reached in their clinical use, we computed the input values as described in Eq. (2) for 12 CiPA drugs belonging to three different TdP risk classes (low, intermediate, high). For these compounds, we used the concentration corresponding to their Effective Free Therapeutic Plasma Concentration (EFTPC) values, the channel-specific half-maximal inhibitory concentrations (IC_{50}) and Hill coefficients (h) extracted from Llopis-Lorente et al. [16]. D, IC_{50}s, h values and the corresponding input values used for the simulation of the 12 CiPA drugs are available at the file “12_CiPA_Drugs.D-IC50-h.xlsx” at GitHub (https://github.com/phi-grib/cardioML/blob/main/12_CiPA_Drugs.D-IC50-h.xlsx). The PR, SVM, and MLP models, trained with data sampled 1/100, were applied to these compounds to predict their APD_{90}.

The predicted results were eventually compared with the simulated APD_{90} read-out from the data array, and the differences were expressed as Relative Error (%).

3. Results

3.1. Overview

The starting point for this work was to generate a large number of APD_{90} values using electrophysiological simulations, as described in the Methods section. For these simulations, the input values represent the relation between the IC_{50}, Hill coefficient and the drug concentration for three ion channels I_{Kr}, I_{Ks}, and I_{CaL}. The output values are the APD_{90} we expect to obtain for cardiomyocytes exposed to a drug with the given I_{Kr}, I_{Ks}, and I_{CaL} input values. These values were collected in an array containing the APD_{90} values produced by the simulations for a wide combination of input values.

The next step was generating small samples of the original data, which were used to train ML models that were used to predict the remaining data as accurately as possible. The results were compared to identify the best ML methods and the lowest training series size producing acceptable results. Finally, the quality of the models was further compared, and the method was validated using 12 CiPA compounds.

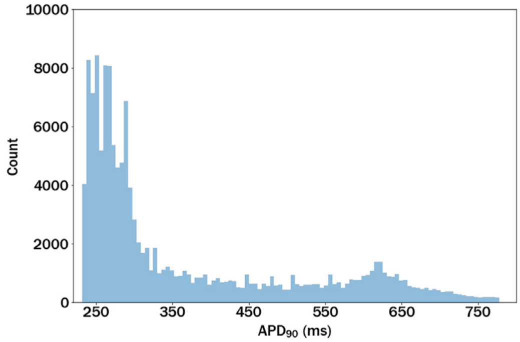

Fig. 2. Distributions of APD_{90} values after data pre-processing.

Our study showed that a simulation of only 1–5% of data is sufficient to build an ML model able to produce accurate estimations of the remaining 99–95% of the APD_{90} values. Such a large reduction in the computation automatically translates into a substantial improvement of both the time and computing power required for the preceding data collection step. Consequently, this reduction opens the possibility of considering drug effects on more than three channels, thereby improving the mechanistic description of the *in silico* tool. From the model settings evaluated, the best results were obtained using SVM. A sampling ratio 1/100 was considered a good trade-off between estimation quality and computation reduction, according to three quality evaluation metrics considered: MRE (%), MAE (%) and percentage of data points with RE below 5% computed for the training, validation, and test set. In the external validation using CiPA compounds, we showed that the maximum error obtained by the SVM model for the sampling ratio 1/100 barely exceeds 1.5% of RE, representing approximately 4 ms of deviation.

3.2. Compilation of the data array

As described above, a data array of APD_{90} obtained for different simulation input values (ratio of drug concentration over I_{Kr}, I_{Ks}, and I_{CaL} IC_{50}) was generated. This dataset consisted of simulated APD_{90} for 175,616 possible combinations of drug effects on channels I_{Kr}, I_{Ks}, and I_{CaL}. It covers a range of blockades from 0.1% to 99.7% for each channel. The pre-treatment applied removed values assigned the top cut-off value (1000 ms) and higher (see Methods section for details). Figure 2 shows the final distributions of the APD_{90} values, where most of the values are concentrated around the physiological biomarker values (264 ms).

This data array resulted from the systematic application of electrophysiological simulations using a range of input values that start from practically safe scenarios (−3 indicates the ratio of 1:1000 between the effective free therapeutic plasma concentration and the IC_{50}). Larger APD_{90} values can only be observed for a few combinations of input values, with a slight concentration of around 610 ms.

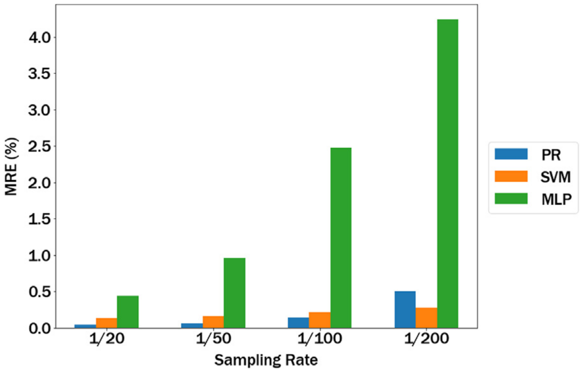

Fig. 3. Selected evaluation metric for different ML models and partitions of data, MRE (%).

3.3. Machine learning: fitting and quality

Before model building, the original data array was split into training, validation and test sets using regular and equal-sized patterns with four different sampling rates: 1/20, 1/50, 1/100, and 1/200. A first sample of data points was assigned as a training set and a second one as a validation set, whereby the rest of the data was devoted to the test set. Then, we used the training set to build PR, SVM and MLP models. A detailed description of the applied sampling rates and ML algorithms is provided in the Methods section. In this work, we optimised the hyperparameters of each algorithm by minimising the loss function on both the training (data known for the model) and the validation (independent data) series and compared the results to evaluate whether a separate validation set is necessary or somewhat redundant in the process of model optimisation. Furthermore, this allows assessing if the best modelling settings (hyperparameters determined for a specific algorithm and sampling) can be re-used to obtain a suitable model for another data set of similar nature without needing a validation set.

After building 12 models, their quality was evaluated using the MAE, MRE and NLDE, computed as explained in the Methods section. Figure 3 summarises the results obtained in the calculation of MRE for each model and the four selected sampling ratios. In the general quality assessment of the models, the lowest MAE (results not shown) and MRE (%) were produced by the PR algorithm. Nevertheless, the differences between PR and SVM, considering both evaluation metrics are minimal, of approximately 0.2%. Compared with the SVM and PR models, the MAE and MRE computed for the MLP model are generally higher and increase for low sampling ratios.

The plots in Fig. 4 illustrate the RE (%) calculated for the predicted APD_{90} values from the test series. We show the differences between the three tested models: PR (blue), SVM (orange), and MLP (green) and how the different sampling ratios impacted the evaluation metrics from the smallest to the highest. In models PR and SVM, the RE (%) range is smaller than for MLP. All the models have in common that the RE (%) is larger for APD_{90} below 300 ms and above 600 ms. In the graphical distribution of RE (%) along the APD_{90} axis, it is noticeable that the initial and end regions of the APD_{90} value range are the ones with the largest RE (%) increase.

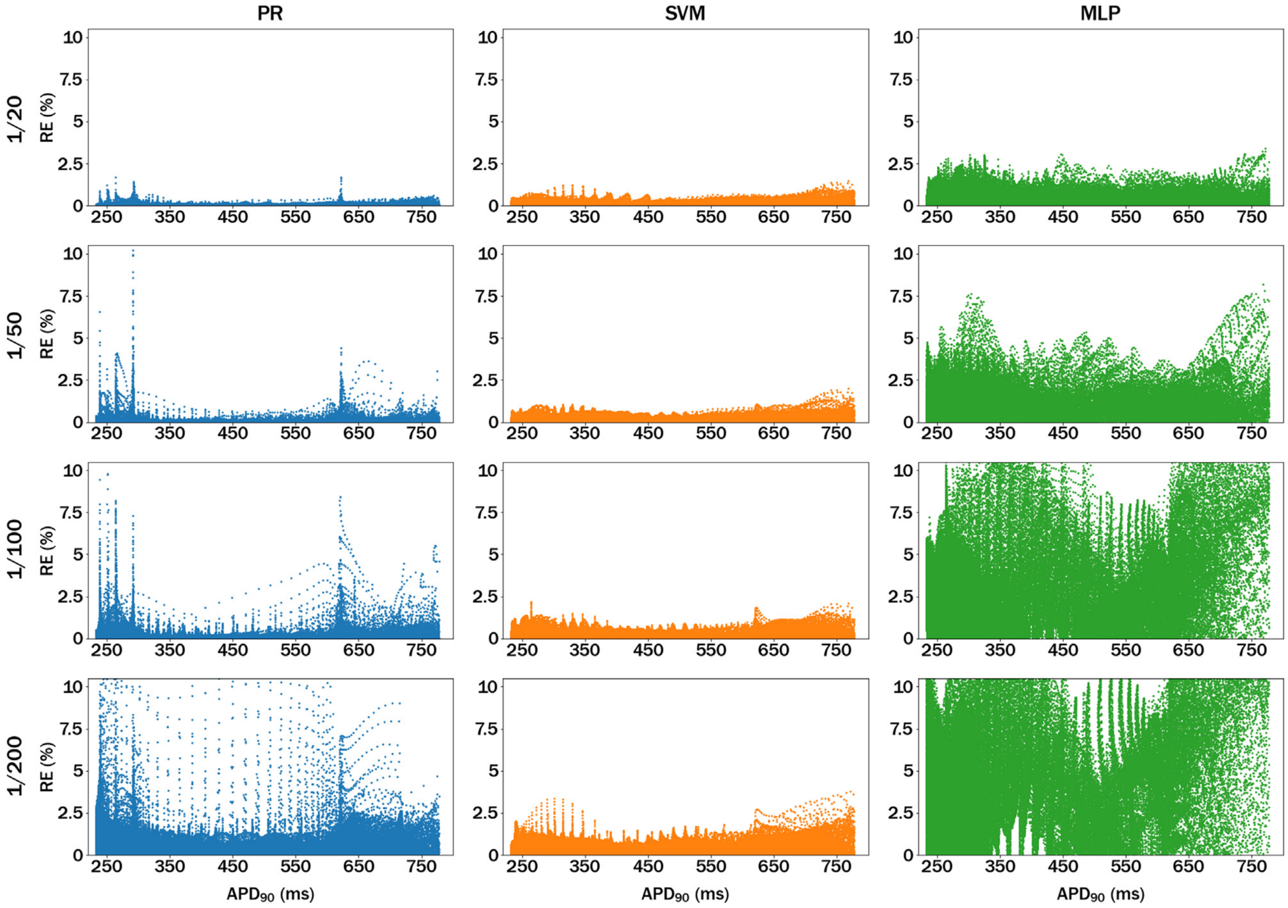

Fig. 4. Each plot shows the RE (%) as a function of the experimental values of APD_{90}. Columns represent three trained models PR, SVM, and MLP. Rows correspond to the sampling ratios applied to the input data starting from 1/20 to 1/200.

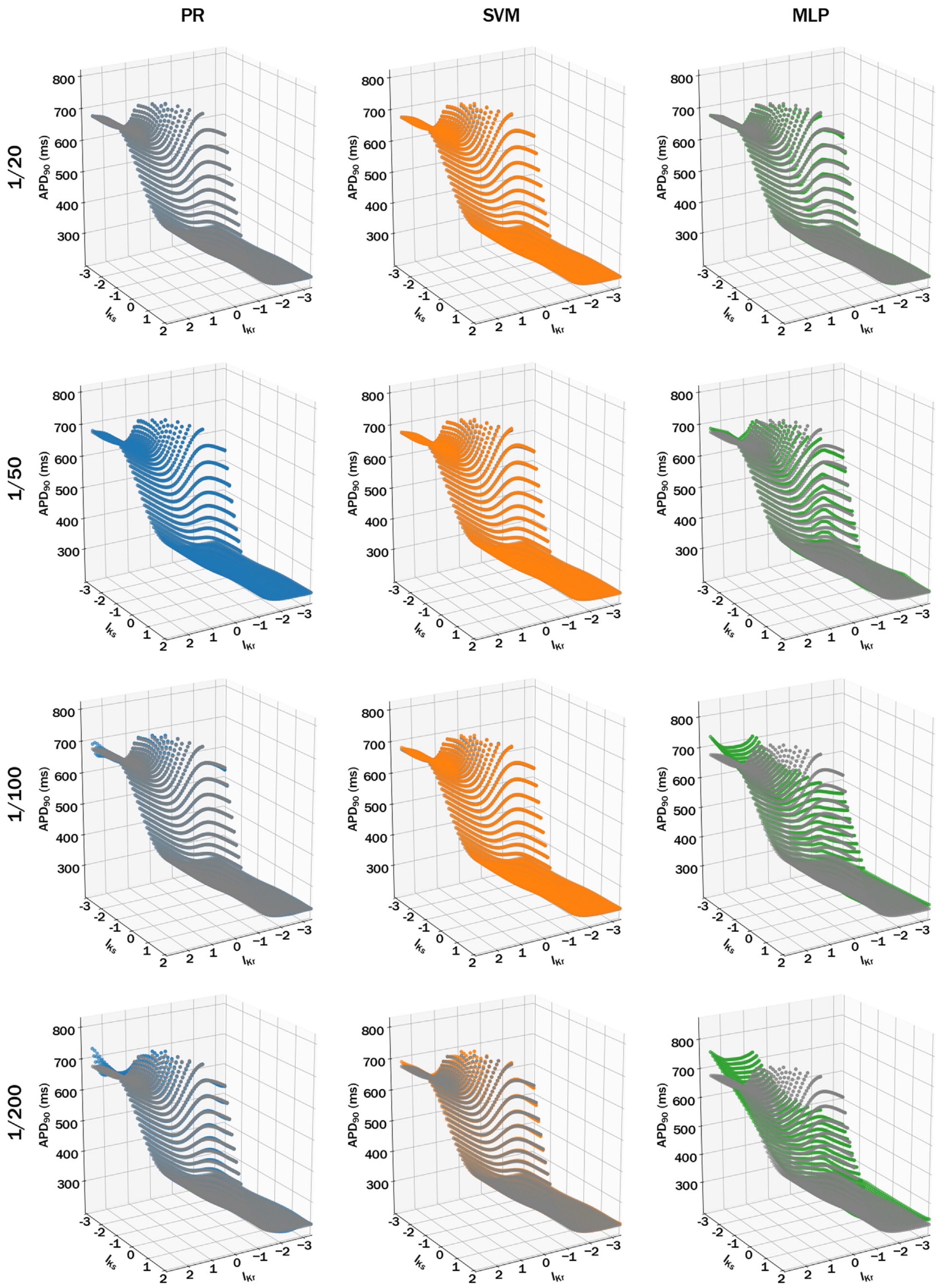

Fig. 5. 3D plots representing I_{Kr}, I_{Ks} and APD_{90} for a fixed value of I_{CaL} equal to 0.3 to give an example. Columns represent three trained models PR, SVM, and MLP. Rows correspond to the sampling ratios applied to the input data starting from 1/20 to 1/200.

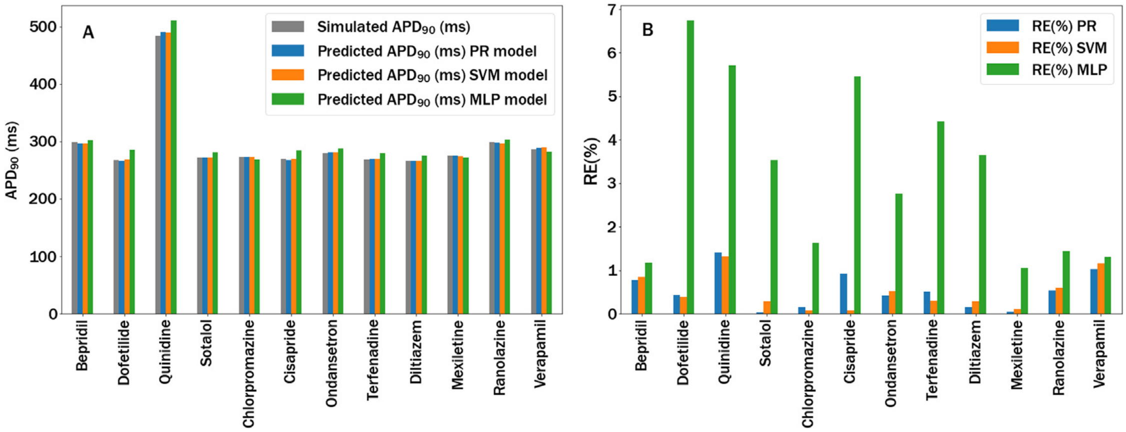

Fig. 6. External validation of the three ML models built using the training set sampled 1/100 performed using a set of 12 CiPA drugs selected from three TdP risk classes. A: Simulated and predicted APD_{90} values. B: RE (%).

Nonetheless, out of the three model types, SVM is the only algorithm that does not make any prediction above the considered threshold of 5% of RE.

A closer observation of the differences between the APD_{90} obtained from the simulation and the predicted, expressed as RE (%), shows that the largest errors have a periodic pattern. This can be observed, for example, in the region between 300 and 500 ms in the results of the PR with 1/100 sampling. These errors are produced by a border effect: the model does not fit well the data points located at the upper and lower limits of the input values. In these positions, there is an abrupt change of the surface, and some models struggle to fit the simulation results accurately. In particular, the use of equispaced sample points in PR can produce slight oscillations at the edges (Runge's phenomenon) [43].

Figure 5 represents a 3D plot, with APD_{90} in the Z (vertical axis) and I_{Kr} and I_{Ks} in the X and Y axes, respectively. A fixed value of 0.3 was used for I_{CaL} in all instances. For all models and sampling rates shown in the graphics, the predicted values correspond more precisely with the simulation results in the centre of the covered output ranges (APD_{90} between 300 and 600 ms). As described above, the values predicted by the three different models are plotted using the following colours PR (blue), SVM (orange), and MLP (green), while grey was used to depict simulated values on each plot. Still, some models exhibit minor deviations in the borders for the reasons explained above. However, even in these areas, we obtain errors well below 5% for all SVM models.

3.4. External validation using a set of CiPA compounds

A set of 12 CiPA compounds with well-defined cardiac electrophysiology, clinical response and known effective therapeutic concentration was used in our project to validate the predictive quality of the models.

Figure 6(A) illustrates the APD_{90} simulated and predicted using the three ML models and the sampling rate of 1/100 for a set of 12 CiPA drugs. For all selected CiPA drugs except Quinidine, which poses a high risk of inducing TdP, the duration of the experimental APD_{90} interval lies below 300 ms. This trend remains unchanged for the APD_{90} values predicted by all three models. Figure 6(B) illustrates the RE (%) for the CiPA dataset used for the external validation. The RE values are very low and below 1% in most cases. This external validation result confirms the results obtained in the validation and testing step of the model training, where the PR and SVM models perform comparatively well. In contrast, the predictions generated by the MLP model deviate more from the experimental values.

Table 3
Performance metrics assessed for the model APD_{90} – (I_{Kr}, I_{Ks}, I_{CaL}) using (A) training, validation, and test set and (B) using the double amount of data for training and the rest for test set.

(A)				
		SVM		
Sampling	Partition	MAE	MRE (%)	NLDE
1/100	Train	0.56	0.18	100
	Val	1	0.27	100
	Test	0.93	0.25	100
(B)				
		SVM		
Sampling	Partition	MAE	MRE (%)	NLDE
1/100	Train	0.56	0.18	100
	Test	0.93	0.25	100

3.5. Simulation of future use by applying the developed methodology to another data array

Once a suitable sampling rate and algorithm were selected, and its hyperparameters were optimised, could these settings be used to fit biomarkers obtained from a different electrophysiological simulation? Should the hyperparameters be optimised again using a validation set? To answer these questions, a second pre-simulated data array was used. The simulations were carried out following the *in silico* action potential (AP) modelling protocol described in the Methods section but now using input values which reflect the degree of inhibition of three different ion currents (I_{Kr}, I_{NaL}, I_{CaL}).

Data simulation and sampling were done using methods equivalent to those described above. Further on, the assessment of the SVM using a sampling ratio of 1/100 was performed following two different approaches. The first option was identical to the methodology described for the array APD_{90} – (I_{Kr}, I_{Ks}, I_{CaL}), in which we used 1/100 data points for model training, 1/100 for validation, and 98/100 for testing. The hyperparameters for this model were determined based on the validation set. In the second scenario, we built an SVM model and optimised its hyperparameters as a function of the training set only, which was compiled by combining the training and validation sets (summing to 2 data points per 100).

For the selected model and sampling rate, the results obtained using either two (Table 3 (A)) or three (Table 3 (B)) partitions are rather similar. Therefore, we found that in comparable situations, the same hyperparameters can be applied to train other models, making it unnecessary to include the validation partition.

4. Discussion

The methodology presented here allows the replacement of computationally costly simulations with estimations generated by a machine learning model. For the method to be profitable, the reduction must significantly impact the number of necessary simulations. In the Results section, it was shown that the number of data points available for training the model largely impacts the errors the model commits on average but selecting 1 of every 100 data points results in an excellent balance between the reduction of the calculations and the robustness and predictive accuracy of the simulation fitting.

Deciding on the necessary number of points required to capture the data structure is a problem-specific decision. In the current application, simulating 1/100 points would practically produce a one hundred-fold decrease in the number of required simulations and computation time, fulfilling our original objectives.

All in all, the described methodology led to the development of high-quality models able to produce APD_{90} values, which are a relatively accurate estimation of those produced by computationally intensive simulations. In this research, we obtained slight differences in the quality of the SVM as compared to the PR model. The errors produced by PR at the borders can be justified by the use of regularly spaced sample points, and could be mitigated by the use of Chebyshev nodes [43]. However, for this particular work we considered that the use of an ad-hoc sampling for PR will not allow a fair comparison with other models. The advantage of applying polynomial transformation is the simplicity of the underlying mathematics, especially in contrast to the Neural Network or SVM models when large regularisation values are used for training. Therefore, PR would be the preferred algorithm if taking the lowest computational complexity as the criteria for choosing the model. But very often, fitting complex data requires the application of a high polynomial degree which goes in hand with a high probability of overfitting, which is the downside of PR. This issue can be resolved through the application of regularisation. The most common regularisation methods are Lasso (L1) and Ridge (L2). While Ridge regression introduces a penalty factor to shrink the magnitude of the model coefficients, Lasso eliminates some of the insignificant coefficients of the model. This difference was extremely important since all features in our input data were essential to model the biological problem correctly and therefore, L2 regularisation was selected instead of the more rigorous L1.

For this reason, if increasing the number of ion channels is the objective of future works, Polynomial Regression would not be the best choice. This is because incrementing the number of input values could yield less smooth surfaces, requiring an increase of the polynomial degree and more rigorous regularisation. On the other hand, the Support Vector Machine algorithm is characterised by a very high generalisation ability, even when the number of instances is less than the number of variables [42]. However, one of the downsides of SVMs for regression is its sensitivity to outliers, which highlights the importance of both data pre-processing and model optimisation. The robustness of the SVM algorithm was confirmed in this work by obtaining high-quality models and precise predictions.

The third and last tested model, the MLP, did not generalise as well as the other two models. A possible explanation for this result may be the insufficient amount of data since Artificial Neural Networks generally require a lot of information to learn from and to predict well. Additionally, since the tuning of hyperparameters of MLP is comparatively expensive in terms of content and time, improving the performance of the neural network model would require testing a wider range of hyperparameters. Nevertheless, the scope of application of Multi-Layer Perceptron is wide and covers several modelling areas. To give a more related example, MLP algorithms were used with high accuracy in Arrhythmia Classification problems where the data was richer in specific information and valuable characteristics [44].

With respect to the method limitations, the models described here were developed and optimised for a combination of three ion channels. When re-using this methodology for a different combination of channels or ventricular arrhythmia biomarkers, the model building and validation would need to be repeated to ensure high-quality results.

We used a specific model (a modified version of O'Hara and colleagues) to generate the APD_{90} array. There are many available models in the field for which the methodology is expected to work well. This, however, would need to be confirmed.

5. Conclusion

In this work, we have shown that it is possible to significantly reduce the number of simulations required to make accurate predictions of ventricular-arrhythmia biomarkers through the application of ML models. We demonstrated that the total amount of the originally simulated data points can be reduced to just 1%. Such data reduction goes in hand with a significant reduction of the time necessary to produce an *in silico* prediction tool based on large pre-simulated datasets. The simple approach developed here opens up the possibility of modelling more complex biological processes, such as the alteration of ventricular-arrhythmia safety biomarkers as a response to an interaction of four and more ionic channels. Additionally, the methods described here are likely to be applicable to model other biomarkers than APD_{90} and even be applied to predict other computational simulation results in different fields of biomedical research. Lastly, the development of effective early-stage screening systems is aligned with the interests of pharmaceutical companies.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors received funding from the eTRANSAFE project, Innovative Medicines Initiative 2 Joint Undertaking under grant agreement No 777365, supported from European Union's Horizon 2020 and the EFPIA. We also received funding from the SimCardioTest supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 101016496. J.L.L. is being funded by the Ministerio de Ciencia, Innovacion y Universidades for the "Formacion de Profesorado Universitario" (Grant Reference: FPU18/01659). The work was also partially support by the Dirección General de Política Científica de la Generalitat Valenciana (PROMETEO/ 2020/043).

We wish to thank Prof. Emilio Soria-Olivas for the valuable discussion.

References

[1] D.C. Bartos, E. Grandi, C.M Ripplinger, Ion channels in the heart, Compr. Physiol. 5 (2015) 1423–1464.
[2] D.M. Roden, Drug-induced prolongation of the QT interval, N. Engl. J. Med. 350 (2004) 1013–1022.
[3] Y.G. Yap, A.J. Camm, Drug induced QT prolongation and torsades de pointes, Heart 89 (2003) 1363.
[4] N. Stockbridge, J. Morganroth, R.R. Shah, C. Garnett, Dealing with global safety issues: was the response to QT-liability of non-cardiac drugs well coordinated? Drug Saf. 36 (2013) 167–182.

[5] ICH S7B Non-clinical evaluation of the Potential For Delayed Ventricular Repolarization (QT Interval prolongation) By Human Pharmaceuticals, European Medicines Agency, 2020 |Available at https://www.ema.europa.eu/en/ich-s7b-non-clinical-evaluation-potential-delayed-ventricular-repolarization-qt-interval (Accessed: 25th May.

[6] ICH E14 Clinical Evaluation of QT/QTc Interval Prolongation and Proarrhythmic Potential For Non-Antiarrhythmic Drugs, European Medicines Agency, 2020 |Available at https://www.ema.europa.eu/en/ich-e14-clinical-evaluation-qtqtc-interval-prolongation-proarrhythmic-potential-non-antiarrhythmic Accessed: 25th May.

[7] B. Fermini, et al., A new perspective in the field of cardiac safety testing through the comprehensive *in vitro* proarrhythmia assay paradigm, J. Biomol. Screen. 21 (2016) 1–11.

[8] P.T. Sager, G. Gintant, J.R. Turner, S. Pettit, N. Stockbridge, Rechanneling the cardiac proarrhythmia safety paradigm: a meeting report from the Cardiac Safety Research Consortium, Am. Heart J. 167 (2014) 292–300.

[9] G. Gintant, P.T. Sager, N. Stockbridge, Evolution of strategies to improve preclinical cardiac safety testing, Nat. Rev. Drug Discov. 15 (7) (2016) 457–471.

[10] Z. Li, et al., Improving the in silico assessment of proarrhythmia risk by combining hERG (Human Ether-à-go-go-Related Gene) channel-drug binding kinetics and multichannel pharmacology, Circ. Arrhythmia Electrophysiol. 10 (2017).

[11] M. Hwang, C.H. Lim, C.H. Leem, E.B Shim, In silico models for evaluating proarrhythmic risk of drugs, APL Bioeng. 4 (2020) 021502.

[12] Z. Li, et al., Assessment of an in silico mechanistic model for proarrhythmia risk prediction under the CiPA initiative, Clin. Pharmacol. Ther. 105 (2019) 466–475.

[13] E. Passini, et al., Human in silico drug trials demonstrate higher accuracy than animal models in predicting clinical pro-arrhythmic cardiotoxicity, Front. Physiol. 8 (2017).

[14] X. Zhou, et al., Blinded in silico drug trial reveals the minimum set of ion channels for torsades de pointes risk assessment, Front. Pharmacol. 10 (2020) 1643.

[15] M.C. Lancaster, E.A Sobie, Improved prediction of drug-induced torsades de pointes through simulations of dynamics and machine learning algorithms, Clin. Pharmacol. Ther. 100 (2016) 371–379.

[16] J. Llopis-Lorente, et al., In silico classifiers for the assessment of drug proarrhythmicity, J. Chem. Inf. Model. 60 (2020) 5172–5187.

[17] B. Christophe, Occurrence of early afterdepolarization under healthy or hypertrophic cardiomyopathy conditions in the human ventricular endocardial myocyte: in silico study using 109 torsadogenic or non-torsadogenic compounds, Toxicol. Appl. Pharmacol. 438 (2022) 115914.

[18] Y. Yoo, A. Marcellinus, D.U. Jeong, K.S. Kim, K.M Lim, Assessment of drug proarrhythmicity using artificial neural networks with in silico deterministic model outputs, Front. Physiol. 12 (2021) 2289.

[19] J. Llopis-Lorente, B. Trenor, J Saiz, Considering population variability of electrophysiological models improves the in silico assessment of drug-induced torsadogenic risk, Comput. Methods Programs Biomed. 221 (2022) 106934.

[20] F.R. Cooper, et al., Chaste: cancer, heart and soft tissue environment, J. Open Source Softw. 5 (2020) 1848.

[21] K.A. Beattie, et al., Evaluation of an in silico cardiac safety assay: using ion channel screening data to predict QT interval changes in the rabbit ventricular wedge, J. Pharmacol. Toxicol. Methods 68 (2013) 88–96.

[22] L. Romero, et al., In silico QT and APD prolongation assay for early screening of drug-induced proarrhythmic risk, J. Chem. Inf. Model. 58 (2018) 867–878.

[23] C. Obiol-Pardo, J. Gomis-Tena, F. Sanz, J. Saiz, M Pastor, A multiscale simulation system for the prediction of drug-induced cardiotoxicity, J. Chem. Inf. Model. 51 (2011) 483–492.

[24] N. Khalifa, L.S. Kumar Konda, R. Kristam, Machine learning-based QSAR models to predict sodium ion channel (Na v 1.5) blockers, Future Med. Chem. 12 (2020) 1829–1843.

[25] M. Varshneya, X. Mei, E.A Sobie, Prediction of arrhythmia susceptibility through mathematical modeling and machine learning, Proc. Natl. Acad. Sci. U. S. A. 118 (2021).

[26] P. Aghasafari, et al., A deep learning algorithm to translate and classify cardiac electrophysiology, Elife 10 (2021).

[27] C. Cai, et al., Deep learning-based prediction of drug-induced cardiotoxicity, J. Chem. Inf. Model. 59 (2019) 1073–1084.

[28] M.M. Maleckar, et al., Combined in-silico and machine learning approaches toward predicting arrhythmic risk in post-infarction patients, Front. Physiol. 12 (2021) 1903.

[29] T. O'Hara, L. Virág, A. Varró, Y. Rudy, Simulation of the undiseased human cardiac ventricular action potential: model formulation and experimental validation, PLOS Comput. Biol. 7 (2011) e1002061.

[30] G.R. Mirams, et al., Simulation of multiple ion channel block provides improved early prediction of compounds' clinical torsadogenic risk, Cardiovasc. Res. 91 (2011) 53–61.

[31] S.M. Stigler, Gergonne's 1815 paper on the design and analysis of polynomial regression experiments, Hist. Math. 1 (1974) 431–439.

[32] A.E. Hoerl, R.W. Kennard, Ridge regression: biased estimation for nonorthogonal problems, Technometrics 12 (1970) 55.

[33] H. Drucker, C.J. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, Adv. Neural Inf. Process. Syst. 9 (1996).

[34] M.A. Aizerman, È.M. Braverman, L. I R, Theoretical foundation of potential functions method in pattern recognition, Avtomat. i Telemekh 25 (1964) 917–936.

[35] Broomnhead, D.S., Lowe D.T.I.C. SELECTE, D., Broomhead, D. & Lowe, D. Radial basis functions, multi-variable functional interpolation and adaptive networks. (1988).

[36] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (1958) 386–408.

[37] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nat 323 (1986) 533–536.

[38] Nair, V. & Hinton, G.E. Rectified linear units improve restricted Boltzmann machines. (2010).

[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[40] C.R. Harris, et al., Array programming with NumPy, Nature (585) (2020) 357–362.

[41] W. McKinney, Data structures for statistical computing in python, in: Proc. 9th Python Sci. Conf., Vol. 445, 2010, pp. 51–56.

[42] J.D. Hunter, Matplotlib: a 2D graphics environment, Comput. Sci. Eng. 9 (2007) 90–95.

[43] L.N. Trefethen, Approximation Theory and Approximation Practice, Extended Edition, SIAM-Society for Industrial and Applied Mathematics, 2019.

[44] S. Savalia, V. Emamian, Cardiac Arrhythmia classification by multi-layer perceptron and convolution neural networks, Bioengineering 5 (2018).