*pharmaceutics*

MDPI

*Article*

# Adoption of Machine Learning in Pharmacometrics: An Overview of Recent Implementations and Their Considerations

**Alexander Janssen [1,*], Frank C. Bennis [2] and Ron A. A. Mathôt [1]**

[1] Department of Clinical Pharmacology, Hospital Pharmacy, Amsterdam University Medical Center, 1105 Amsterdam, The Netherlands
[2] Quantitative Data Analytics Group, Department of Computer Science, Vrije Universiteit Amsterdam, 1081 Amsterdam, The Netherlands
* Correspondence: a.janssen@amsterdamumc.nl

**Abstract:** Pharmacometrics is a multidisciplinary field utilizing mathematical models of physiology, pharmacology, and disease to describe and quantify the interactions between medication and patient. As these models become more and more advanced, the need for advanced data analysis tools grows. Recently, there has been much interest in the adoption of machine learning (ML) algorithms. These algorithms offer strong function approximation capabilities and might reduce the time spent on model development. However, ML tools are not yet an integral part of the pharmacometrics workflow. The goal of this work is to discuss how ML algorithms have been applied in four stages of the pharmacometrics pipeline: data preparation, hypothesis generation, predictive modelling, and model validation. We will also discuss considerations before the use of ML algorithms with respect to each topic. We conclude by summarizing applications that hold potential for adoption by pharmacometricians.

**Keywords:** machine learning; pharmacometrics; pharmacokinetics; pharmacodynamics



**Citation:** Janssen, A.; Bennis, F.C.; Mathôt, R.A.A. Adoption of Machine Learning in Pharmacometrics: An Overview of Recent Implementations and Their Considerations. *Pharmaceutics* **2022**, *14*, 1814. https://doi.org/10.3390/pharmaceutics14091814

Academic Editor: David Barlow

Received: 20 July 2022
Accepted: 22 August 2022
Published: 29 August 2022



## 1. Introduction

### 1.1. Background

Pharmacometrics is a multidisciplinary field utilizing mathematical models of physiology, pharmacology, and disease to describe and quantify the interactions between medication and patient. This involves models of drug pharmacokinetics (PK), pharmacodynamics (PD), exposure-response (PK/PD), and disease progression. One of the main themes of interest is the explanation of variability in drug response between patients. Various statistical techniques have been adopted to quantify such inter-individual variation (IIV) [1].

Non-linear mixed effect (NLME) modelling has been embraced as a statistical method for describing treatment effect on a population and individual level [2,3]. Population PK modelling makes efficient use of sparse data by pooling information of multiple individuals, and breaking down treatment response in shared and individual effects. Observations of the dependent variable (i.e., drug concentrations or treatment effect) can then be used to adapt the prediction to the individual patient, resulting in higher accuracy.

Recently, however, advances in hospital digitization, data collection, and inclusion of increasingly extensive laboratory testing in standard clinical care have resulted in the availability of richer datasets. This increased accessibility of complex data sources such as genomic or gene expression data stresses current modelling approaches as they can lack the flexibility to handle these data. As a response, more attention is being paid to the opportunity of using machine learning (ML) algorithms as an innovative strategy for pharmacometric modelling [4,5]. The field of ML has seen an explosive boost of promising applications for image analysis, text recognition, and other high-dimensional data. There are many examples of their successful application in the medical domain, for example for the diagnosis of breast cancer [6], identification of biomarkers from gene expression

data [7], and survival analysis [8]. As ML methods offer strong predictive performance there is no denying that its adoption in pharmacometrics brings with it exciting new modelling opportunities.

As the relatively young ML research field is maturing at a rapid pace, more advanced model architectures are frequently being proposed in order to further improve predictive accuracy. Consequently, understanding the differences and intricacies of distinct learning methods is becoming increasingly more difficult for non-experts. A proper understanding of the advantages and pitfalls of these methods is essential for their responsible and reliable use, especially for clinical applications. As most of the emphasis has been put on the supposed high predictive accuracy of ML methods, it is easy to become overconfident in their abilities. It is thus important to monitor and guide the adoption of ML in pharmacometrics.

In this review, we will discuss recent approaches for the use of ML algorithms in the context of pharmacometrics, while also providing important considerations for their use. For some examples, we will provide demonstrations based on simulation experiments. We also discuss the important concept of model validation and the importance of understanding what is actually learned by the algorithm.

In this work, we will be assuming a general understanding of ML and the most common algorithms. For those wanting to learn more about the basic concepts of ML, Badillo et al. offer an excellent tutorial on ML aimed at pharmacometricians [9].

### *1.2. Structure of This Review*

This review is structured as follows. First, we discuss applications of ML algorithms in three stages of the pharmacometrics pipeline: data preparation, hypothesis generation, and predictive modelling. We define these stages as follows: data preparation deals with the imputation of missing data and dimensionality reduction. Next, in the section on hypothesis generation we discuss methods for clustering data, and how ML can be used for the detection of influential covariates. In the predictive modelling section, we discuss ML-based alternatives to traditional modelling approaches. We will conclude our review of recent application with a discussion on model validation, focussing mainly on estimating model generalizability and the interpretation of ML models.

For each topic, we first discuss the current approach, its (possible) limitations, followed by what ML techniques have been proposed to address the issues. At the end of each topic, we will summarize the discussion with considerations for the use of ML for each issue.

### *1.3. Literature Search*

In order to support the initial framing of our discussion we performed a literature search. Our objective was to find recent articles discussing ML in the context of pharmacometrics. The following search query for PubMed was constructed:

> (“machine learning” [tiab] OR “artificial intelligence” [tiab] OR “random forest” [tiab] OR “gradient boosting” [tiab] OR “XGBoost” [tiab] OR “support vector” [tiab] OR “neural network” [tiab] OR “deep learning” [tiab]) AND
> (“pharmacometric*” OR “pharmacokinetic*” OR “pharmacodynamic*” [tiab] OR “pharmacogen*” [tiab] OR “drug concentration” [tiab] OR “dose estimation” [tiab] OR “dose optimization” [tiab]) AND
> (“2016/01/01” [Date - Publication] : “3000” [Date - Publication]) NOT (review[Publication Type]).

The search identified a total of 586 articles (as of 30 May 2022), of which 198 were included based on abstract screening. Additional articles were obtained by means of scanning the reference lists of included articles, or by specifically searching in the arXiv database (https://arxiv.org/; accessed from 30 May 2022 until 30 June 2022). Some ML papers are only indexed in pre-print servers, and thus can not be found in PubMed.

## 2. Data Preparation

### *2.1. Data Imputation*

Missing data are a frequent occurrence in the clinical setting. When encountering missing data one can drop all data entries or covariates with missing data, impute missing data, or employ maximum likelihood estimation techniques. As many clinical datasets are relatively small, the latter two options are often preferred. Missing data are often categorized in one of three categories; they are either missing completely at random (MCAR), missing at random (MAR; missingness depends on observed data), or missing not at random (MNAR; missingness depends on unobserved data). The source of the missing data can affect the choice of imputation method. In addition, the type of data (i.e., continuous or categorical) can also be a reason for choosing different methods. In the below sections, we will focus on the problem of data imputation, which requires us to choose an appropriate model for the prediction of missing data. How do we select such a model?

#### 2.1.1. Standard Methods for Data Imputation

Imputation can either be performed once (single imputation), or multiple times (multiple imputation). Commonly used methods for single imputation include imputation by mean or mode, grouping missing data in a separate category (in the case of categorical covariates), or regression-based imputation. In multiple imputation, multiple samples are taken from a predictive distribution allowing for the quantification of the resulting variance of model output. This provides a measure of uncertainty of the imputation. A Bayesian multiple imputation strategy has been proposed for NLME models, which presented lower bias of parameter estimates compared to mean value imputation for MCAR and MAR data [10]. A maximum likelihood procedure based on this strategy was also shown to lead to less biased PK parameter estimates compared to mode and logistic regression based approaches [11].

Prior studies have been mainly concerned with the imputation of categorical variables. Model-based (i.e., multiple imputation and likelihood-based) approaches seem to perform well for this kind of data, but do require one to make assumption about the distribution of the data. This can be more difficult for continuous variables. In these cases, it might be compelling to also evaluate regression-based techniques for imputation. Unfortunately, choosing an appropriate regression model when the assumed relationship is non-linear can be difficult. This is especially the case when covariates are correlated. For this reason, ML-based regression techniques have been suggested with the goal of improving the accuracy of regression-based imputation. An early study suggests that when covariates are simulated based on non-linear relationships, the bias of PK parameters after performing imputation can be reduced by using a random forest or neural network prediction model rather than mean imputation [12].

#### 2.1.2. Machine Learning Methods for Data Imputation

A paper by Batista and Mondard compared the accuracy of the *k*-nearest neighbour (*k*-NN) method to mode, decision tree, and rule-based methods for MCAR data imputation [13]. They found that *k*-NN generally was the most accurate method. In *k*-NN, individuals are grouped in *k* clusters based on similarity (for example based on Euclidean distance). Next, missing values can be imputed based on mean/median values from their respective cluster. The method is simple to implement, but might be less effective in small or very homogeneous datasets.

Although Batista and Mondard found that the decision tree-based method was not as accurate, random forest-based approaches have been more successful [14–16]. In the popular implementation missForest [14], a random forest is combined with multiple imputation by chained equations (MICE; [17]). MICE is an iterative procedure where each missing covariate is imputed based on the remaining covariates. Initially, missing data are imputed using an arbitrary method (e.g., by their mode) and a model is fit to predict missing data

for each covariate independently. This process is repeated with the assumption that each iteration, more accurate imputations of the covariates are used for the predictions. The overall process can be repeated for multiple initial datasets. This way, MICE allows for multiple imputation based on deterministic regression models. Using missForest outperformed *k*-NN and linear MICE for single imputation of MCAR data [14]. However, performing multiple imputation using linear MICE was more accurate than single imputation using missForest on MCAR and MAR data [15]. Performing multiple imputation using missForest led to the overall best accuracy. This implies that multiple imputation procedures might also generally be preferred for regression-based imputation.

Several probabilistic approaches have also been proposed for performing regression-based multiple imputation. Unsupervised deep latent variable models, such as generative adversarial networks (GANs) and variational auto-encoders (VAE), have recently been successfully applied to data imputation problems [18,19]. A GAN is a combination of two neural networks, a generator and a discriminator, which compete against each other. The discriminator learns to discern true from generated data, such that the generator becomes increasingly effective at reproducing real data. The generator learns to represent the generative distribution of the data. The GAIN approach, a GAN specifically developed for the imputation of missing data, was found to more accurately impute MCAR and MAR compared to missForest and MICE [18].

A VAE is a special neural network architecture that learns to encode its input into a distribution of latent variables by means of variational inference. A decoder neural network is then learned to reproduce the original input from samples of this distribution. Mattei and Frellsen describe the combination of an importance-weighted autoencoder with a maximum likelihood objective for imputation of MAR data. This method was found to be more accurate than *k*-NN and missForest [19].

Finally, Gaussian Processes (GPs) are stochastic processes which represent a prior distribution over latent functions. GPs are a non-parametric framework to fit models to data, while simultaneously providing a measure of variance. This allows one to sample from the distribution of latent functions and perform multiple imputation. A recent study has proposed a deep GP approach which suggested more accurate imputation of MCAR data compared to *k*-NN, MICE, and GAIN [20].

#### 2.1.3. Considerations

We have described several ML techniques for data imputation. These techniques might offer improved imputation of covariates that have non-linear relationships with the non-missing covariates. Several findings point to improved performance of regression models when using multiple imputation compared to single imputation. Since most standard regression methods are deterministic, strategies such as MICE are advisable for performing multiple imputation. Using missForest in this context might improve the prediction of more complex covariates. MICE is flexible in that a different model can be used for imputation of each covariate. We can thus use ML methods for the prediction of non-linear covariates while using likelihood-based approaches for categorical covariates. This might be a promising method to explore in the context of NLME modelling.

Recent probabilistic approaches, such as deep latent variable models and GPs, offer an interesting take on regression-based imputation. These methods use likelihood-based approach which might improve imputation accuracy [18–20]. However, it is not clear if more accurate methods of data imputation offer significant benefits in terms of reducing the bias of parameter estimates in NLME models. Studies will have to evaluate the benefit of using these more complex methods in the context of pharmacometric modelling.

### *2.2. Dimensionality Reduction*

Dimensionality reduction is a technique for detecting patterns in data and reducing this to a lower number of principal components. This can be useful when analysing very high-dimensional data (e.g., gene expression data). Such data are difficult to include in

pharmacometric models (and thus rarely are) as their effect might be dependent on a specific combination of patterns. One of the main linear techniques for dimensionality reduction is principal component analysis (PCA). It uses a linear mapping to project each data point to a lower-dimensional representation. The so called principal components are independent and aim to preserve as much of the original variance in the data. Such decompositions can be used to facilitate data visualization and can be used for hypothesis generation (see Section 3). This technique has for example been used to predict the impact of different factor VIII mutations on haemophilia A disease severity [21]. Here, 544 amino acid properties were collected, which they were able to reduce to 19 components using PCA. The researchers could thus drastically reduce data dimensionality while reportedly retaining 99% of the information in the dataset.

Non-linear methods have also been proposed which allow a more flexible mapping to lower-dimensional space. This way, these methods might be able to represent more complex patterns and thus increase the explained variance. One example is the VAE, where the input data are condensed into a set of latent variables. Other prominent examples include uniform manifold approximation and projection (UMAP) [22] and t-distributed stochastic neighbour embedding (t-SNE) [23]. Xiang et al., have recently performed a comparison of ten dimensionality reduction methods for predicting cell type from RNA sequencing data [24]. Although the study shows that no one-size-fits-all method exists, UMAP, t-SNE, and VAE were found to generally outperform other methods of dimensionality reduction. Accuracy of PCA was also high, but it suffered in terms of stability with respect to changes in the number of cells, cell types, and genes.

Another study by Becht et al., compared t-SNE to UMAP to discern cell populations based on patterns in single-cell RNA sequencing data [25]. In their case, UMAP led to more reproducible results and more meaningful visualizations. This is in contrast to the results of Xiang et al., which found t-SNE to be the best performing method [24].

Considerations

To our knowledge, the use of dimensionality reduction techniques in the context of pharmacometrics is still quite limited. One of its current principal uses might be as a pre-processing step for generating hypotheses. The lower dimensional representations are ideal for visualization and can be used to detect patterns in otherwise complex data. However, one downside can be that the meaning of the resulting lower dimensional components might be difficult to interpret. In a two dimensional *t*-SNE visualization for example, samples that are closer together and thus more similar would also have been more similar in high dimensional space. The actual features underpinning this similarity can not be discerned from the analysis. In such cases, further inspection of the data would be required to identify the explaining covariates. Another downside is that most of these methods require re-analysis of the data when including new samples. In addition, for each new sample we need to collect the same data to produce the lower dimensional representation.

An important finding is that there might not necessarily be a best method for performing dimensionality reduction [24]. This suggests that different methods should be compared based on the predictive performance of the resulting principal components. However, from our literature search we found that it was more common to use ML to directly learn to predict the outcome of interest from high-dimensional data and select influential covariates for downstream analysis [26–28]. We will further discuss such approaches in the context of covariate selection in Section 3. It is unclear how such an approach compares to the above-mentioned methods for dimensionality reduction. More studies are needed to explore the benefit of using these different techniques.

## 3. Hypothesis Generation

### *3.1. Discovery of Patient Sub-Populations*

Detecting patient subgroups can help to understand why groups of individuals respond differently to treatment. We found several studies that describe the use of clustering

techniques in the context of pharmacometrics. Most focus on grouping patients based on the similarity in terms of treatment response. Kapralos and Dokoumetzidis describe the use of *k*-means clustering for the detection of two patient sub-populations presenting distinctly different absorption patterns of Octreotide LAR [29]. Here, they used the Fréchet distance to define similarity between patients, which can be used to calculate the similarity of longitudinal data in terms of shape. These kinds of measures however do require that all patients are sampled at similar time points. This can be difficult to achieve in practice.

Another study describes the use of mixture models for dividing patients into different classes based on treatment response [30]. Next, the study attempted to predict subgroup based on patient characteristics. This allowed for the identification of clinical indicators that were associated with the different subgroups. The resulting seven treatment classes were validated in an external dataset. This study offers a nice representation how clustering can be used for hypothesis generation and subsequent analysis.

Clustering assumptions can also be implemented within predictive models. In one study, an expectation maximization (EM) approach is used to group individuals based on drug concentration data and a predefined compartment model [31]. Each cluster has its own distinct parameterization and estimate of (residual) variance. This allows us to categorize new patients to a cluster and treat them as were similar patients. Another study has taken an interesting approach where a mixture model-based algorithm is described for grouping individuals into different PK models [32]. These PK models are automatically constructed during the clustering procedure. This approach can thus also be used to generate hypotheses about the appropriate PK models to use for different patient groups. Requirement of a predefined model can also be avoided by combining clustering and supervised ML algorithms. Chapfuwa et al., describe a model for clustering patients in the context of time-to-event analysis [33]. A neural network is used to represent the covariates into a latent variable space, which is made to behave as a mixture of distributions. Each individual is assigned to a cluster in the latent space which contains a corresponding event-time distribution. This allows for the identification of heterogeneous treatment response groups based on covariate data.

Considerations

We have discussed examples of studies that have used *k*-means and mixture models to cluster patients in subgroups. Mixture models allow for probabilistic inference, and have been used in more complex model architectures [31–33]. These approaches are experimental, but may be of interest to apply to different problems for the purpose of hypothesis generation.

Both mixture models and *k*-means require the user to specify the number of clusters beforehand. This can be difficult when there is no prior information to choose the number of subgroups or when the data cannot be visualized due to high-dimensionality. In those cases, we can either reduce data dimensionality (see Section 2.2), or use some criterion to select the optimal number of clusters [30,34]. An additional downside of mixture models is that they are sensitive to local minima. One study found that prior initialization based on *k*-means++ (an adaptation of *k*-means) allows for a simple procedure to improve model convergence [35].

Clustering patients based on the dependent variable can pose issues in pharmacometric modelling. For example, difficulties arise when clustering patients based on drug concentration measurements when these are collected at different time points. Aside from increasing the speed of processing many samples, the benefit of clustering based solely on the dependent variable might be unclear when differences in drug exposure can be easily discerned from the concentration–time curve. Alternatively, clustering patients based on (individual) PK parameters, summary variables such as area under the concentration time curve (AUC), or independent variables might be more informative in practice.

### *3.2. Covariate Selection*

Considering the black-box nature of most ML algorithms, why would one consider using ML for covariate selection? Stepwise covariate modelling (SCM), which is perhaps the most commonly used covariate selection method in pharmacometrics, also has its limitations [36]. Stepwise approaches can lead to difficulties when the data contains many covariates, when there is high collinearity, or when covariate effects are highly non-linear and difficult to determine a priori. The potential of ML-algorithms in this context is that they can be used to learn the optimal implementation of covariates. By performing post-hoc analyses of the model, it can then be possible to determine influential covariates. In the below sections we first discuss limitations of stepwise methods in order to suggest a set of requirements for a successful covariate selection method. Next, we describe ML methods that have been used for this purpose and evaluate if they fit the derived requirements.

#### 3.2.1. Limitations of Stepwise Covariate Selection Methods

In SCM, covariates are included one by one (forward inclusion), and each time the covariate leading to the largest significant decrease in objective function value is included. After all covariates have been tested, the included covariates are removed from the full model one by one (backward elimination). The covariates that do not result in a significant increase in objective function value are removed. This approach leads to some issues. First, due to the potentially large number of statistical tests there is a risk of multiplicity. Second, for an honest implementation of stepwise methods, all hypotheses need to be defined beforehand. This includes all covariates to consider and their functional form. The latter can be quite difficult to determine without first extensively inspecting the data. Finally, the statistical tests are not independent, since the significance of the tests might depend on how and if other covariates have been included. This is especially a problem when there is high collinearity between covariates. Studies have indeed indicated that SCM has a relatively low power when covariates are correlated, have weak effects, or when the number of observations in the dataset is limited [37–39].

To reduce the effect of multiplicity and to ensure tests are independent, full model methods are preferred [36]. However, this does not resolve the issue of choosing a suitable functional form to implement each covariate a priori. We suggest the following definition of ideal covariate selection method: it (1) should perform a full model fit (i.e., test multiple hypotheses simultaneously); (2) should be able to learn covariate relationships from data; while (3) penalizing complex solutions (e.g., by regularization); and (4) should allow for the interpretation of resulting relationships. If the method is unable to learn optimal implementations of covariates, we risk making type II errors. If the method does not constrain model complexity it risks inflating the importance of covariates (by fitting arbitrarily complex relationships) resulting in higher type I error. Finally, if the method is not interpretable, we run into problems when actually implementing the selected covariates. If sub-optimal functions are used to implement the selected covariates, they might still result in insignificant effects.

#### 3.2.2. Linear Machine Learning Methods

The least absolute shrinkage and selection operator, or LASSO, is a regression-based method that performs covariate selection by regularization. The LASSO employs the $\ell^1$-norm, which penalizes the absolute size of the regression coefficients $\beta$. This causes the coefficients of unimportant covariates to be shrunk to exactly zero. All covariates of interest are tested simultaneously in the form of linear equations using a full model fit. Next, a hyperparameter $s$, which controls the size of $\beta$ such that $\sum_{j=1}^{N_{cov}} |\beta_j| \leq s$, can be selected using cross-validation procedures. The use of $s$ is a substitution for statistical testing as only the most important covariates will have coefficients greater than zero. The LASSO has seen applications for population PK and Cox hazard models where it outperformed stepwise methods in terms of speed and predictive accuracy [38,40]. Owing to its simplicity, direct integration into the non-linear mixed effects procedure is possible [38].

The LASSO performs a full model fit, penalizes complex solutions, and is interpretable. However, due to the assumption of linear relationships the LASSO fails to meet our second requirement. Since this assumption might not hold for all covariates, there is a risk of type I errors in covariate selection. Although the predictive performance of the LASSO holds up relatively well [41], its performance suffers when the relationship of some of the covariates are non-linear [42].

Multivariate adaptive regression splines (MARS) is a ML algorithm for the approximation of non-linear functions based on piecewise linear basis functions [43]. The method automatically learns the optimal number of splines and their location for single covariates and their combinations. Its classic implementation uses a stepwise approach to prune the number of basis functions to reduce model complexity. Alternatively, a LASSO-based implementation of MARS has been described which presented favourable performance compared to the classic approach [44]. This method has the potential of matching our requirements, but has not yet seen frequent use for the purpose of covariate selection. We have found one abstract mentioning its use, but it did not explore its benefit for approximating non-linear functions [45].

#### 3.2.3. Tree-Based Methods

Tree-based ML algorithms, such as the random forest and gradient boosting trees, have seen recent applications for the purpose of covariate selection. These methods offer a flexible approach to learning non-linear functions, while offering a large number of hyperparameters that can be tuned for regularization. Maximum tree depth, the change in minimum objective function change required for a split, or the minimal number of samples in each node can be empirically set (or automatically using cross validation) to reduce model complexity. The method fits our first three requirements, although the effects of regularization are more difficult to interpret compared to the $\ell^1$-norm.

In order to use tree-based methods for covariate selection, covariate importance scores based on "impurity" (also known as Gini importance) or permutation are often calculated. The covariates can be ranked based on these scores. Covariates can be included based on biological plausibility or if they meet a certain threshold [46]. Permutation-based methods are preferred over impurity based methods as the latter can be biased for differently scaled or high cardinal covariates [47]. Simulation studies seem to suggest relative accurate identification of true covariates [42,48].

It is important to note that there is no underlying theory that supports the use of these scores as selection criteria. In addition, another problem is that these scores do not provide information on what functional form to use in order to implement the covariate. It is possible that the relationship underlying the importance has a complicated functional form, and is less important when approximated using basic functions in the final model. As is, this approach does not meet our requirement of interpretability. Novel approaches such as explainable gradient boosting [49,50], might improve the interpretability of tree-based models.

#### 3.2.4. Genetic Algorithms

Genetic algorithms are a special form of search space optimization techniques that rely on evolutionary concepts such as natural selection, cross-over, and random mutation for selecting the most optimal model. They have long been suggested as an alternative approach to model selection for pharmacometric applications [51]. Genetic algorithms allow for testing many opposing hypotheses with respect to model structure simultaneously. In this way, it matches our first requirement. Its direct output is an optimal model (according to the survival function) and matches our fourth requirement.

The general procedure is as follows: first, the full search space is defined, containing all model features to be considered. Next, an objective function is chosen that describes model fitness. Usually this is a combination of the log likelihood of the model and additional penalties for model complexity. Then an initial population is formed containing random

combinations of the selected features. For each model, the fitness function is evaluated and the 'fittest' models are selected to produce the next generation. This process is repeated for several iterations or when a stopping criteria is met. Since many models have to be fit and evaluated the computational cost of fitting genetic algorithms can be relatively high.

A recent study describes the development of a software-based resource for automating model selection using genetic algorithms, improving their accessibility [52]. This application was compared to stepwise methods and seemed to more accurately recover the true model based on simulated data. Such comparisons are however difficult to make, since the penalty for model complexity was more conservative in the case of the stepwise methods versus the genetic algorithm. The reverse was found in another study, where a stricter fitness function resulted in overly simplified models [53]. Choosing an appropriate fitness function by balancing model accuracy and complexity is not straightforward. It is possible to use heuristic methods such as the Akaike or Bayesian information criterion, but it is likely that there is no one-size-fits-all solution. In addition, the method cannot be used to learn more complex representations of the covariates than were originally included in the search space. Genetic algorithms thus do not meet our second requirement.

### *3.3. Considerations*

We have discussed several ML algorithms that can be used for covariate selection. We also proposed four requirements that underlie an ideal covariate selection tool. All discussed methods test all hypotheses simultaneously and match our first requirement. The LASSO offers the most comprehensible approach to regularization but might risk higher type I error due to its assumption of linear relationships. More complex ML algorithms, such as tree-based methods, are more flexible with respect to the representation of non-linear relationships. Perhaps not surprisingly, these methods also suffer the greatest in terms of interpretability (with the exclusion of decision trees). This makes it difficult to translate the results of covariate importance to an appropriate model. The current principal use of tree-based methods might thus be for selecting covariates for subsequent analysis.

The MARS and explainable gradient boosting algorithms come closest to meeting all four requirements. By using piecewise linear functions, MARS approximates non-linear functions and is interpretable. In explainable gradient boosting, a large number of simple models (e.g., small depth decision trees) are fit to each covariate, and relationships can be visualized by summarizing over these models. The visualizations obtained from both methods could be useful in providing an initial intuition about the appropriate functional form to use when implementing covariates. Alternatively, model interpretation methods might be of interest to infer covariate relationships from ML models. We have previously performed an investigation into how one such explanation method can be used to visualize the relationships between covariates and estimated PK parameters [54]. We found that these relationships matched implementations in previous PK models and biological concepts. It might be of interest to further investigate the application of such tools in the context of pharmacometrics. Model explanation methods will be further discussed in Section 5.2.

We have performed a simple simulation study (see Appendix A for implementation details) to showcase the use of some of the previously mentioned methods for covariate selection. Each method was fit to predict individual clearance estimates based on covariate data containing two true covariates and 48 noise covariates. In Figure 1, we depict the measures of covariate importance as determined by means of LASSO, MARS, random forest, or explainable gradient boosting. Each method has correctly identified the two true covariates as important. In addition, we have depicted the approximation of the covariate effect by MARS and explainable gradient boosting (see Figure 1E,F).

We have also discussed the use of genetic algorithms for automation of model selection. Compared to local search or stepwise methods, genetic algorithms offer an intuitive procedure based on evolutionary concepts for simultaneously testing multiple hypotheses with respect to model selection. Software-based resources such as presented by Ismail et al., could help improve accessibility for performing experiments based on genetic

algorithms [52]. Although they might be an improvement compared to stepwise methods, genetic algorithms do not meet the suggested requirements for a comprehensive covariate selection method. The main issue lies with selecting an appropriate fitness function. There is no consensus on a generally applicable fitness function. This is worrisome, as choosing an inappropriate fitness function can negatively affect the result.

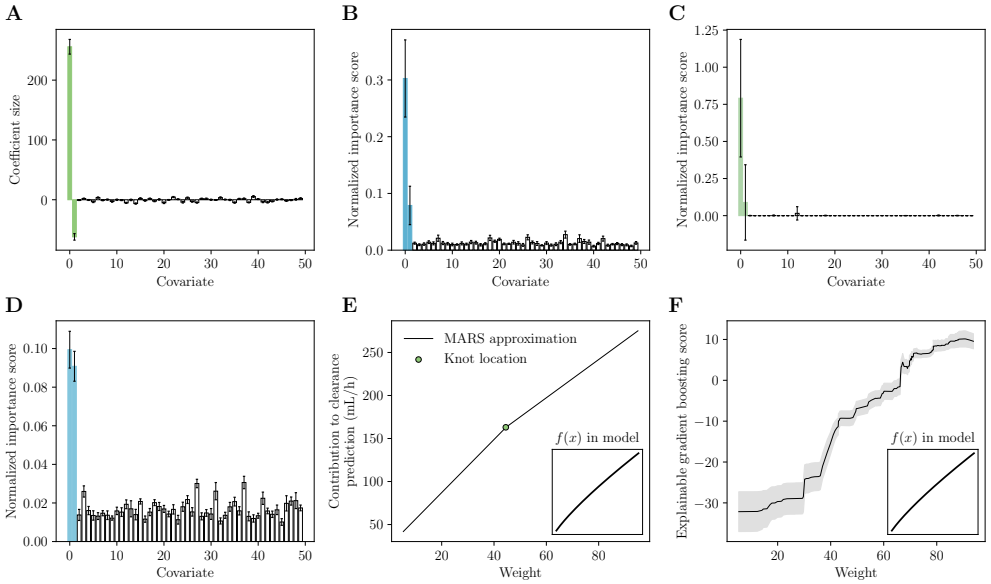


**Figure 1.** Examples of machine learning-based covariate importance scores. LASSO coefficients (**A**), random forest importance scores (**B**), MARS covariate importance (**C**), explainable gradient boosting scores (**D**), MARS (**E**) and explainable gradient boosting (**F**) approximation of the effect of covariate 1 are shown. Coloured bars indicate true covariates, whereas white bars represent noise covariates. Bar height represents the importance of each covariate. Importance should be larger for true covariates than for noise covariates. The resulting scores can for example be used to select covariates eligible for inclusion in a NLME model. Error bars indicate standard deviation of each score following a ten-fold cross validation. In (**E**), the point indicates the piecewise split location (i.e., a knot). In (**F**), shaded area represents the standard deviation of model predictions in the explainable gradient boosting model. Figure inset represents the function used for covariate 1 in the simulations.

In summary, none of the presented approaches can meet all our requirements for an ideal covariate selection method. Their purpose might thus mainly be for providing a more informed set of covariates to test. We have mentioned some methods such as MARS and explainable gradient boosting which might also provide intuition about appropriate functional forms to use. Next, genetic algorithms can be used as a full model based approach to testing the hypotheses. More research is however required to optimize this procedure.

## 4. Predictive Models

### *4.1. Machine Learning for Pharmacokinetic Modelling*

The development of NLME models is a time-consuming process and requires extensive domain knowledge. Recently, ML algorithms have seen applications as efficient alternatives to NLME modelling [55–58]. Aside from reducing the time spend on the model building process, ML algorithms can be used as a flexible approach to handle complex and high-dimensional data sources. For example, ML algorithms have been used to directly estimate PK parameters from dynamic contrast enhanced MRI images [59], or to screen almost 2000 genetic markers to find variants affecting tacrolimus exposure [60].

Our search identified many different ML algorithms used for pharmacokinetic modelling. However, we observe that most of these models suffer from problems impeding their reliable use. For example, most models take the current time and drug dose as direct inputs. Aside from leading to issues when multiple drug doses are given, it is uncertain how these

inputs will be interpreted by the model. In addition, some models are trained to predict drug concentrations at specific time points, making them unreliable when extrapolating to unseen time points. Finally, since the translation from covariates to drug concentrations can be quite non-linear, these models are prone to overfitting and might require larger datasets in order to generalize well. Based on these issues, we again suggest a set of requirements: (1) the model should be able to produce a continuous solution (i.e., extrapolate to unseen time points); (2) it should be able to adapt to complex treatment schedules (e.g., frequent dosing, mixing different types of administration); (3) it should be able to handle differences in the timing and the number of measurements per patient; and (4) should be reasonably interpretable. Below, we discuss several of the ML algorithms obtained from our literature search and identify two reliable methods for predicting drug concentrations.

#### 4.1.1. Evaluation of Different Approaches

A basic strategy has been to directly predict the concentration-time response based on patient characteristics (e.g., covariates), the dose, and the current time point of interest [55,57]. By predicting a single concentration, we can make independent predictions at each time point per patient. This way, we can meet requirement one and three. In order to satisfy requirement two, we must treat each dosing event as independent and add the remaining concentration from the previous dosing event to the current prediction. A problem with this approach is that the prediction does not represent the total concentration of drug in the body (usually only blood levels) so we lose information about drug accumulation in peripheral tissue. In addition, we assume that the model will learn to predict drug exposure based on the covariates, make adjustments based on the dose, and use the supplied time point to obtain the concentration along the time dimension. It is impossible to completely validate that the model uses these quantities as assumed.

We can however easily show that this approach is unreliable. We performed a simulation study using a neural network to predict real-life warfarin concentrations based on patient age, sex, the dose given at $t_0$, and the time point for which to evaluate (see Appendix B for implementation details). The neural network can provide a continuous solution (Figure 2A), and is reasonably able to represent the kinetics of warfarin (e.g., it seems to recognize its absorption and early distribution behaviour). However, when we extend the time frame beyond what was seen during training, we found that the model incorrectly predicts an increase in the exposure (data not shown). In addition, when artificially setting the dose to zero, the model still predicts a response. Admittedly, we can use data augmentation to learn the neural network to predict no exposure when the dose is zero, or when the time point is long after dose administration. We cannot however augment the data with counterfactual cases (specifically with respect to the given dose) and thus the method is inherently unreliable.

Other approaches have used ML to learn optimal dose or AUC instead of a full concentration-time response [61–63]. These approaches have their own issues. They will likely be more accurate when measurements are provided as input, resulting in problems relating to requirement three. In addition, it is more difficult to interpret the credibility of the current prediction. In our previous example we could identify problems with the concentration-time curve, but in the case of direct AUC or optimal dose predictions it is more difficult to for example validate the prediction based on visualizations. Interpretation of the model using covariate importance scores can also be difficult [63]. Determination of the AUC or optimal dose based on a prediction of a full concentration-time curve, which can be verified by the observed measurements, will likely still be a more reliable approach.

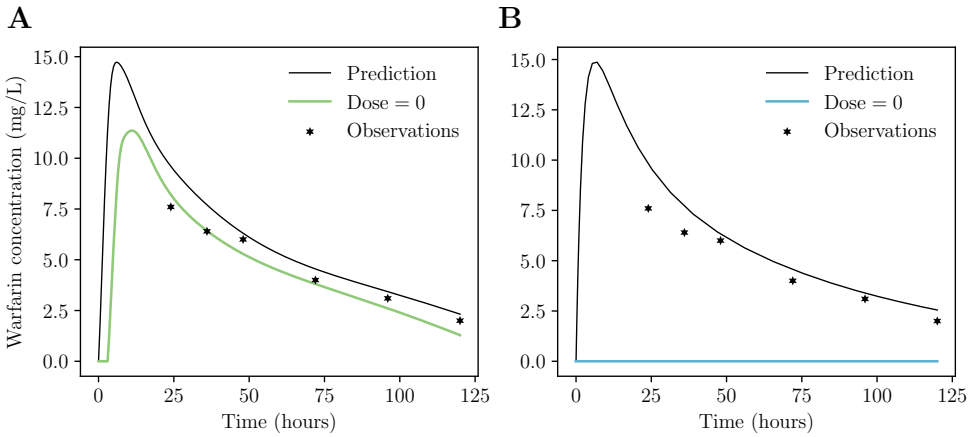


**Figure 2.** Examples of predicting drug concentrations using neural networks. Concentration-time curves for a single test set patient are shown as predicted using naive (**A**) and ODE-based (**B**) neural networks. Model prediction when artificially setting the dose to zero is depicted by the colored lines. Stars represent the measured warfarin concentrations for the patient.

The next strategy might thus be to use more complex ML algorithms better suited to time-series predictions, such as recurrent neural networks [56,58,64,65]. These studies suggest that these methods can indeed accurately predict the changing drug concentration over time in the context of multiple dosing events [56,58,65]. However, Lu et al., found that such methods did not extrapolate well to unseen dosing schedules [58]. Alternatively, the authors suggest a neural-ODE based approach. Here, a neural network is used to encode the covariates into a latent variable space $z$, which serves as the input to an ordinary differential equation (ODE) solver. This solver is a neural-ODE, a special form of recurrent neural network that learns to represent continuous dynamics similar to an ODE [66]. The neural-ODE is used to explicitly integrate the dosing and timing information. The resulting latent variables adjusted for the current time point are then fed into a decoder network, which produces concentration predictions. They show that this approach does correctly extrapolate to unseen dosing schedules, while also identifying no exposure when the dose is artificially set to zero [58]. This model fits our first three requirements. However, the model architecture is quite complex, and can be difficult to interpret.

We have also recently proposed a similar approach where we directly combine neural networks and ODEs [67]. Here, we also use an encoder network to transform the covariates into latent space variables. In contrast to the neural-ODE approach, we explicitly formulate an ODE system based on compartment models. Dosing events are then used to perturb this ODE system, which directly outputs concentration predictions at the desired time points. This offers several benefits over a neural-ODE: first, by explicitly defining drug kinetics we might reduce the data required to fit the model by imposing explicit constraints. Second, the latent variables now represent PK parameters (e.g., clearance or volume of distribution), which can be compared to previous results. Finally, since we are using a known ODE system, model predictions are credible and interpretable. This way, the method is a better match to requirement four compared to neural-ODE based methods.

#### 4.1.2. Considerations

The last two approaches indicate that ODE-based ML methods are more reliable for the prediction of drug concentrations. In Figure 2, we present a comparison between a naive and ODE-based architecture. We see that only the latter correctly identifies the absence of concentration response when the dose is set to zero (Figure 2B). A neural-ODE can be used to learn the kinetics underlying drug exposure, whereas an explicit ODE system can be used when prior knowledge is available. It is of interest to compare the performance of both these methods.

One remaining opportunity lies with the characterization of prediction uncertainty. In NLME models, the estimation of residual IIV allows for MAP estimation of the PK parameters to correct the prediction based on concentration measurements. Adding this functionality to the above ODE-based models might be of interest for encouraging their adoption in general practice [68]. One approach for obtaining an estimate of predictive and parameter uncertainty are deep ensembles [69]. Here, the predictions of multiple randomly initialized neural networks are combined, and the mean and variance of predictions is presented. This approach is simple to implement and might outperform methods that explicitly estimate parameter uncertainty, such as Bayesian neural networks [70].

Finally, one important aspect of the pharmacometrics pipeline is simulation. As we have shown by removing the dosing event for the neural networks in our example, actively searching for errors in our model is essential for its evaluation. Learning if new patients are different from the data that the model was trained on might help to provide intuition about cases where we trust model output and cases we do not. Simulation is an important approach for facilitating such analyses. We further discuss the topic of model validation in Section 5.2.

### *4.2. Machine Learning for Predicting Treatment Effects*

In the wake of -omics research, the interest to personalize patient treatment based on gene or protein expression profiles has increased greatly. High-dimensional data sources stress classical statistical modelling approaches, and many have turned to ML-based approaches [7,26,27]. In addition, some conventional methods such as the cox proportional hazard model, assume linear relationships with covariates. This has prompted the development of tree-based survival models [8], which might be better suited to problems where non-linear interactions can be expected [48]. In the following sections we will focus on the application of ML algorithms for exposure–response modelling (PK/PD models) and survival analysis (time-to-event models).

#### 4.2.1. Exposure-Response Modelling

Exposure-response modelling involves the prediction of treatment effect in relation to the current dose or concentration of an administered drug. It is similar to PK modelling in that it often involves the use of differential equations for describing the dynamics of drug action (e.g., target-site distribution or target binding). Likewise, it is possible to use ODE-based neural network architectures to learn the effects of covariates on model parameters. However, the assumptions underlying the chosen ODE system are often weaker than in the case of PK models [71].

One can discern two types of model components: drug-specific and biological system-specific properties [72]. Drug-specific properties, such as receptor affinity, can be estimated from in vitro data. Biological system-specific properties, such as protein or receptor expression, can only be measured in vivo and can be highly variable between individuals. The latter properties are thus especially sensitive to errors in modelling assumptions. In addition, these effects are often governed by highly non-linear relationships [72]. One approach can be to use neural-ODE based models to learn the relationship between exposure and response from data [73]. Novel interpretable methods have also been described to infer such physical relationships from data [74]. However, in many cases such a direct relationship offers an overly simplistic representation of the biological situation. Alternatively, we can explicitly define part of the ODE system and use neural-ODE to estimate unknown components [75]. This allows the user to explicitly include reasonably certain model components (e.g., drug-specific properties), while neural-ODEs estimates more the more variable and complicated biological system-specific properties from data. It might be of interest to compare such approaches to classical PK/PD approaches.

Novel approaches have also aimed at improving the estimation of biological system-based effects, for example by extrapolating from cell-line data or animal models [76–78]. These approaches allow for more frequent measurement of treatment endpoints, and can

be used to estimate otherwise difficult to obtain quantities (e.g., spatial distribution of drug in the target tissue [77]). As an example, patient-derived cancer xenograft models can be used to characterize the concentration-dependent effect of drugs on their target based on tumour growth data [76]. Obtaining such results from in vivo patient data would not only be complicated but also undesirable due to the high frequency by with which tumour biopsies would have to be performed.

Adoption of ML algorithms in the context of exposure-response modelling hold exciting opportunities. For example, a recent study describes a method for predicting the drug response of thousands of cancer cell lines based on mutations and expression profiles [79]. Another study describes a method for quantifying the individual variability in tumour dose-response while also identifying important biomarkers [80]. ML techniques can also be used to learn PK or PD (e.g., drug absorption or receptor binding) parameters based on quantitative structure-activity relationships [81]. In short, there have already been many diverse applications of ML in this field, and we expect them to further increase in the future.

#### 4.2.2. Survival Analysis

Time-to-event analysis refers to a set of methods that aim to describe the probability of a specified outcome occurring over time. In the case where only a single event is possible per individual (i.e., survival analysis), non-parametric methods such as the Kaplan-Meijer estimator are used to estimate the distribution describing the proportion of individuals who have 'survived' over time. These methods allow for the statistical comparison of the efficacy of two competing treatment modalities. Often, we are also interested how covariates affect this efficacy. The standard method for estimating of such effects is the Cox proportional hazard model. Here, the covariates are assumed to affect the hazard in a proportional manner. However, this assumption might be too limiting for a flexible analysis of high-dimensional data, complex time-dependent effects, or multi-state survival models. One might instead turn to ML for learning the effect of covariates or the underlying model structure.

As we have mentioned, the random survival forest model has been proposed for performing non-linear analysis of covariates. The random survival forest was suggested to obtain either similar or lower error compared to Cox proportional hazard models [8]. Recently, deep learning approaches have also been proposed for survival analysis [82,83]. These were generally suggested to obtained higher accuracy (in terms of concordance index) when compared to Cox models and survival forests on several clinical datasets. In addition, these approaches allow for the calculation of the individual risk of prescribing a certain treatment [82]. In Cox models, this risk is constant unless treatment interaction effects are explicitly included, which can be complicated. The neural network-based approach produced treatment recommendations that led to a higher rate of survival compared to random survival forests [82].

Recurrent neural networks have been suggested as a method for estimation of the effect of time-dependent covariates. These methods were again found to outperform previous methods (including neural networks) in terms of concordance index [84,85]. By predicting the individual risk based on current and previous information at discrete time intervals, these methods might improve learning of time-dependent effects.

Multi-state models step away from the usual alive-death dichotomy and instead specify disease progression into intermediary (non-fatal) or competing states [86]. In oncology for example, this allows for the categorization into induction, relapse, remission, and deceased states. This allows for prediction of the risk of relapse following complete remission or survival in the case of relapse [87]. Specification of the dynamics between different states and the influence of covariates might require strong assumptions. A generalizable approach used neural-ODE to learn the likelihood of being in each state over time [88]. This approach obtained improved performance over multi-state Cox models in a competing risk setting.

#### 4.2.3. Considerations

There are many interesting avenues exploring novel applications of ML in exposure-response modelling. The onset of Big Data has resulted in many opportunities for using more advanced and computationally efficient methods for analysing these data. However, some of these tools might still remain at the fringe due to their complexity. Domain-specific reviews providing an overview of recently developed algorithms might help to provide guidelines for optimal strategies for analysis and validation [81]. Without the availability of model code or comprehensive tutorials on the use of complex ML models, adoption of these methods will likely remain limited.

An important consideration for the use of ML for survival analysis is whether the current dataset supports such analyses. Small datasets or those without frequent measurements of the covariates over time might lack the power to correctly describe non-linear effects. As a result, we would recommend evaluating multiple different models for the task at hand. For example, for some datasets, Cox models either performed equal to or better than neural network-based approaches [85]. This could be the case in smaller datasets, or when the data does not support more complex models. In such scenarios, Cox models might be preferred due to their improved interpretability. One might also prefer Cox models when model interpretation is of the highest importance.

## 5. Model Validation

### *5.1. Choosing a Validation Strategy*

An essential component of any analysis using ML is a model validation strategy. Arguably, performing model validation is also more generally advisable in the context of pharmacometrics. In contrast to conventional statistical methods however, ML algorithms such as neural networks are extremely flexible. Even neural networks with a single hidden layer are considered to be universal function approximators, meaning that they can fit any data arbitrarily well [89,90]. This flexibility results in a high risk of "overfitting", a phenomenon where the resulting model is completely tailored to the current dataset such that it generalizes poorly. It is thus important to validate the generalizability of a ML model before it can be used in practice. Arguably the best validation method is to determine the predictive accuracy on independent datasets. Unfortunately, data are often limited. In this section, we report on alternatives for performing model validation.

#### 5.1.1. Options for Estimating Model Generalizability

In the most simple case the dataset is divided in a "train" and "test" set. The train set is used to fit the model, whereas the test set is used to estimate the accuracy of the model. In ML, usually a split using roughly 70–80% of data for training and 20–30% as test data is advised. This is however largely dependent on the size of the test set as it should contain a representative number of samples. Some ML models have additional parameters (i.e., hyperparameters) that can be tuned in order to affect performance. When performing such optimization, the dataset should be split in three parts: a train set (for fitting the model), a "validation" set (for determination of the performance of the current hyperparameters), and a test set (for determination of the accuracy of the final model). A similar approach is be advisable when performing covariate selection.

Performing a single random split of the dataset can be a poor estimate of model generalizability. For this reason, the accuracy is often evaluated on multiple train/test splits and their results are pooled. We will discuss three such techniques for estimating the generalization error: random subsampling without replacement, bootstrapping (subsampling with replacement), and *k*-fold cross validation. A schematic overview of the three methods is provided in Figure 3.

In random subsampling without replacement (also known as Monte Carlo cross validation), the model is fit to a random split of the dataset multiple times, model accuracy is evaluated on the corresponding test sets, and the results are pooled. Since we are sampling without replacement, each sample occurs only once in either the training or

test set. Crucially, one should understand that this leads to biased estimates of the true population mean and its standard error. This is because the samples in each split are not independent, and thus violate the Central Limit Theorem. Optionally, one can use the finite population correction factor to improve estimates. However, since the choice of validation strategy is independent of experimental design, possible problems can simply be avoided by performing a bootstrap instead.

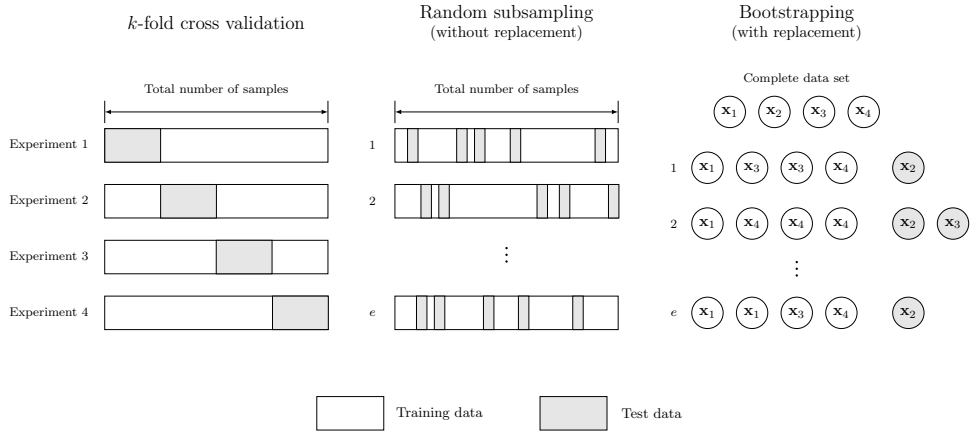


**Figure 3.** Examples of methods for estimation of model generalization accuracy. Schematic overview of three common validation strategies: *k*-fold cross validation, random subsampling, and bootstrapping (with replacement). The white shapes denote the training data, whereas grey shapes denote testing data. Here, *e* represents the total number of experiments to run.

In bootstrapping, samples are taken with replacement, resulting in independent samples. Usually $n$ (size of the original dataset) samples are taken, and the samples that are not in the training set are used as the test set. Again the results are pooled for a large number of replicates. This estimate can be reliable when all the models converge, which is often not a problem in ML.

Finally, in *k*-fold cross validation, the data are partitioned into $k$ "folds", or subsets, and the model is trained on $k - 1$ folds. The remaining fold is used to estimate test accuracy. Models are fit iteratively so that each fold is used for testing once. *k*-fold cross validation is also frequently performed in the context of hyperparameter optimization.

#### 5.1.2. Considerations

In general, drawing a consensus on the best method for estimating model generalizability is difficult. A downside of random subsampling and bootstrapping is the large computational cost associated with fitting a large number of models. In addition, when sampling with replacement, the number of unique samples in the training set is reduced, which may lead to higher bias of predictions in smaller datasets [91,92]. A similar issue can occur when performing *k*-fold cross validation with a low number of folds [92]. The other extreme, known as leave-one-out cross validation (LOOCV; where $k = n$), has been suggested to have the best bias-variance trade-off when compared to the other methods [91,92]. Performing ten-fold cross validation leads to similar results compared to leave-one-out cross validation at a lower computational cost [91,92]. The latter is especially relevant as dataset size increases in size (as models have to be fit). Papers have also reported on the inconsistency of LOOCV, specifically that selection of the true data generating model does not actually improve as dataset size increases [93].

Another important consideration when estimating model generalizability is to prevent data leakage. When multiple observations are available per patient, a simple random split might result in different observations of a single individual appearing in both the train and test set. These observations should be grouped to prevent information leakage. Care should also be taken when optimizing hyperparameters. The data that is used to test the current set of hyperparameters should not include samples from the final test set. This

means that the dataset should first be divided in a train and test set. The hyperparameters can then be optimized by performing *k*-fold cross validation on data from the train set only. The accuracy of the best model from the cross validation (containing the optimal set of hyperparameters) is then evaluated on the test set. This entire process can also be repeated for multiple test sets, essentially performing an additional (outer) cross-validation. This approach also estimates sensitivity of the hyperparameters to random sub-sets of the data.

Another point to consider is that creating random subsets of the data can exacerbate class imbalances. For example, an algorithm trained to diagnose disease (classifying samples in 'no disease' and 'disease') can present an inflated accuracy if the model always predicts no disease while the test set does not contain many samples from the disease group. This can often be the case for rare diseases. Alternatively, in case-control studies, train sets can be saturated with control patients, resulting in a model that is unable to make accurate predictions for the case group. In such situations, the data can be stratified so that class proportions are roughly similar in each fold. Care should again be taken to prevent data leakage; data should not be stratified based on the independent variables as this results in more similar train and test sets.

There likely is no single best method of estimating model generalizability. In many cases, *k*-fold cross validation might be preferred when bootstrapping encompasses a too high computational cost. When choosing cross validation, the most important aspect is to choose a suitable value of *k*. Arlot and Celisse provide an excellent survey on model selection using cross validation [94]. Their findings may aid in choosing the appropriate cross validation procedure on a per-problem basis.

### *5.2. Model Interpretation*

Explainable artificial intelligence has emerged as an important subfield of ML research. Especially with respect to the adoption of ML for medical applications, understanding why a certain prediction is made is crucial for instilling trust. As an example, model interpretation methods can be used to indicate regions-of-interest underlying predictions for medical image classification [95,96]. There are two types of explanation methods: model-specific and model-agnostic. Model-specific explanation methods for example involve using the regression coefficients from linear models to explain the proportional relationship between covariates and the dependent variable. Although these coefficients have a straightforward meaning, they are not necessarily true; correlated covariates can complicate the estimation of true covariate effects. In more complex models, such as neural networks, the meaning of the model parameters is not immediately obvious. Although neural network specific explanation methods exist [97], generally model-agnostic explanation methods are used. These methods themselves can be considered black-box as they aim to replace the complex model by a more simple, interpretable model.

There have been numerous suggestions for interpretation frameworks aimed at explaining ML model output, including Local Interpretable Model agnostic Explanations (LIME), Deep Learning Important FeaTures (DeepLIFT), and SHapley Additive exPlanations (SHAP) [97–99]. An overview of popular methods is provided by Holzinger et al. [100]. This reference also provides short discussions of each method which may provide intuition on what method to use for what goal. It can be difficult to understand how the function of each of these methods affects model interpretation. It is possible that using different frameworks on the same model results in different explanations. It has thus been of interest to find theoretical support for these methods. One method that aims to offer theoretical guarantees is SHAP [99].

SHAP has already seen use in pharmacokinetic modelling. One study used SHAP for the identification of important covariates when using neural networks to predict cyclosporin A clearance [101]. Aside from covariate importance, which only provides a limited interpretation of the model, SHAP can also be used to visualize covariate relationships [54]. To present an example, we performed a SHAP analysis on the prediction of warfarin absorption rate ($k_a$) by the previous discussed ODE-based neural network

(implementation details in Appendix B.3). In Figure 4, we depict the relationship between age and $k_a$, stratified by sex, as represented by SHAP values. Since we only have a single continuous and categorical variable as input to our neural network we can also obtain their exact functional relationships. In cases when more covariates are included, this is generally not possible. The model predicts a different effect of age on $k_a$ for males and females (see Figure 4). The SHAP values allow for the evaluation if the relationships adhere to biological expectations of covariate effects. However, since there are only a few female patients, we should take caution when performing such evaluations. Although the SHAP values seem to be able to represent the effect of the covariates well, extrapolating to unseen samples might be unreliable. Other approaches have been developed in order to estimate the uncertainty of out-of-distribution samples [102]. The use of only a single explanation method might thus not be enough for the complete evaluation of ML models.

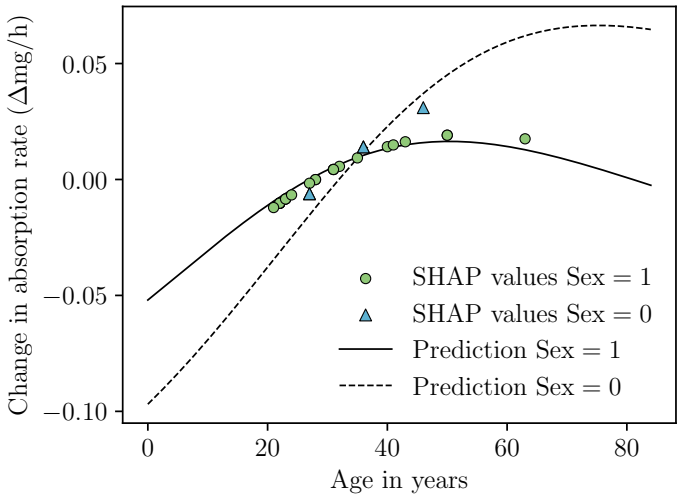


**Figure 4.** Examples of using SHAP for model interpretation. Change in warfarin absorption rate (Δmg/h) prediction by the neural network as estimated by SHAP values. Here, circles represent the SHAP values calculated for men, whereas triangles represent SHAP values calculated for female patients. Lines represent the neural network predicting when fixing patient sex (solid for male, and dashed for female) and predicting absorption rate based on different values for age.

Considerations

There are many available explanation frameworks for ML models. Here, we have chosen to discuss only one technique to illustrate how these methods can be used. In the case of ODE-based methods, model-agnostic methods are useful to visualize the effect of covariates. However, it is possible that such methods alone are not sufficient for use in clinical practice. It might also be of interest to know if each individual prediction can be trusted, especially when predicting for samples that are different from training data.

Due to the large number of available explanation methods and difficulties with representing their accuracy, choosing the correct method can be a daunting task. In most cases, we can only lean on some theoretical guarantees or expected behaviour on simple examples [103]. Molnar et al., present an excellent overview of pitfalls of these methods and how to resolve them [104]. Since most of the model-agnostic interpretation methods operate in a similar fashion (i.e., they perturb data and evaluate the effect on predictions), they share similar pitfalls. This study also makes the important suggestion that there again is no one-size-fits-all method.

Another great reference is the work by Yang et al. [105]. Here, the authors first outline frequently used concepts in model interpretation, after which they provide two showcases demonstrating how these concepts can be applied for explaining ML model output. This study shows how these techniques can provide insight into the strengths and weaknesses of ML models.

## 6. Main Points

We have discussed several recent applications of ML algorithms in the context of pharmacometrics. More specifically, we have presented how ML techniques can and have been used for data imputation, dimensionality reduction, unsupervised clustering, covariate selection, drug concentration prediction, and treatment response prediction. In general, tree-based models and neural networks were the most frequently used algorithms for these purposes. Most papers report an improvement in performance when comparing the use of these methods to classical approaches. In addition, more complex architectures, which were most frequently based on neural networks, were suggested to be the most accurate. More research is however needed to compare these methods to classical approaches, such as NLME models.

We have started our discussion with the application of ML methods for data preparation. With respect to missing data imputation, the literature suggests lower bias for estimated model parameters when using multiple imputation compared to single imputation. Several ML-based methods have been suggested for imputation of missing data based on regression of observed covariates. It is still unclear however if the added complexity of these methods actually leads to improved estimation of missing data. Evaluation of such methods in the context of the smaller datasets often seen in pharmacometrics is thus required. We have briefly discussed methods for dimensionality reduction. Such approaches might be interesting for facilitating the analysis of complex genetic or proteomics data. However, their benefit compared to using ML to detect influential covariates is not obvious. Although the latter approach can result in the loss of information on covariate dependencies, its results are more easily applicable.

Next, we discussed the application of ML techniques for hypothesis generation. Studies have used unsupervised clustering techniques for the detection of patient subgroups from data. These subgroups can for example be used to generate hypotheses regarding differences in treatment response between patients. ML methods have also been used for the detection of influential covariates. A study has suggested that covariate importance scores produced by the random forest can be used to obtain a better selection of covariates compared to stepwise methods [42]. This and other methods do not yet offer a complete alternative to stepwise methods, but are useful for producing an initial set of hypotheses regarding important covariates to consider for inclusion. Methods for search space optimization such as genetic algorithms are a promising approach for improving the selection of model components that lead to the best performance. This approach requires the selection of a fitness function to control model complexity, which can be difficult to choose. More research is needed for an empirical method for selecting an appropriate fitness function.

ML models have also been used as predictive models in the context of pharmacometrics. We make the point that ODE-based methods outperform other methods in reliability regarding the prediction of drug concentrations. We have showcased this point by using a simple example of how naive methods can misinterpret drug doses when they are passed directly as input. Next, we discuss several ML-based methods for predicting treatment response and efficacy. Again, ODE-based methods show potential for improving prediction reliability, especially in the case of PK/PD modelling. There have also been ML-based approaches for survival analysis. It is the question however if these are appropriate for every analysis, as more complex models might not always result in improved performance.

Finally, we have discussed model validation. Due to the flexibility of many of the discussed methods, deciding on a suitable model validation strategy should be an integral part of the modelling process. The generalization performance of the model is an important metric for judging its appropriateness. Validation of accuracy on a high quality external dataset is often regarded as one of the best options. It is however not clear what is the next best alternative when such data are not available. We would like to urge pharmacologists that are interested in using ML to first consider whether their use case supports the use of these tools. In our experience, we found that imposing constraints on these models

(for example based on prior knowledge using ODEs) can help in improving performance when data are sparse. We also want to stress the importance of evaluating what the ML model has learned. Examples include the analysis of the most important covariates, or performing sensitivity analysis (e.g., using methods such as SHAP) with respect to the model parameters. Understanding how the model makes its predictions allows for the removal of any biases, and adapting model regularization to prevent it from making 'mistakes'. Examination of undersampled regions of the input space can provide insight into the extend to which model predictions can be trusted. Specifically training the model on new data from undersampled patients will help improve generalizability in the long run.

In the coming years, our expectation is that the number of studies exploring the use of ML in pharmacometrics will keep increasing. Perhaps some of the methods mentioned in this review have already become a standard part of the pharmacometrician's tool kit in the near future. This could be the time for researchers interested in ML to educate themselves in ML concepts and, perhaps, to develop new model architectures better suited to problems in the field of pharmacometrics.

**Author Contributions:** Conceptualization, A.J.; Data curation, A.J.; Formal analysis, A.J.; Methodology, A.J.; Supervision, R.A.A.M.; Writing—original draft, A.J.; Writing—review and editing, A.J., F.C.B., and R.A.A.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Dutch Organization for Scientific Research (NWO) in the framework of the NWA-ORC grant number NWA.1160.18.038.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** All data, including model code, simulated datasets, and the warfarin dataset, are available at https://github.com/Janssena/SI-AIEP-paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| AUC | Area under the concentration time curve |
| DeepLIFT | Deep learning important features |
| EM | Expectation maximization |
| GAN | Generative adverserial network |
| GP | Gaussian Process |
| IIV | inter-individual variation |
| k-NN | k-nearest neighbour |
| LASSO | Least absolute shrinkage and selection operator |
| LIME | local interpretable model agnostic explanations |
| LOOCV | Leave-one-out cross validation |
| MAR | Missing at random |
| MARS | Multivariate adaptive regression splines |
| MCAR | Missing completely at random |
| MICE | Multiple imputation by chained equations |
| ML | Machine learning |
| MNAR | Missing not at random |
| NLME | Non-linear mixed effect |
| ODE | Ordinary differential equation |
| PCA | Principal component analysis |
| PD | Pharmacodynamic |

| | |
|---|---|
| PK | Pharmacokinetic |
| SCM | Stepwise covariate modelling |
| SHAP | Shapley additive explanations |
| t-SNE | t-distributed stochastic neighbour embedding |
| UMAP | Uniform manifold approximation and projection |
| VAE | Variational autoencoder |

## Appendix A. Machine Learning for Covariate Selection

### *Appendix A.1. Data*

A dataset of severe haemophilia A patients receiving a single dose of 50 IU $kg^{-1}$ blood clotting factor VIII (FVIII) concentrate was simulated based on a previous pharmacokinetic (PK) model [106]. Pharmacokinetic time profiles were based on a two compartment model with inter-individual variation on the clearance and central volume parameters. Typical clearance was estimated based on patient weight (power function) and age (linear function), whereas central volume was based on weight only (power function). Next, we sampled values to produce individual estimates of the clearance parameter. Patient age was sampled from a $Uniform(1.1, 66.0)$ distribution. We fit a Gaussian Process to predict patient weight based on age. The simulated age values were subsequently used to sample corresponding weight estimates from the Gaussian Process. Finally, we augmented the dataset with 48 noise covariates from a standard normal distribution.

### *Appendix A.2. Models*

We fit a LASSO (sklearn python package, v1.0.2 [107]), MARS (py-earth python package, v0.1.0 [107]), random forest model (sklearn python package, v1.0.2 [107]), and explainable gradient boosting model (interpret python package, v0.2.7 [50]) to predict the individual clearance estimates based on the augmented set of covariates. A ten-fold cross validation was performed. The test sets were used the calculate the accuracy of model predictions using the root mean squared error (RMSE). MARS was the most accurate model with a mean RMSE of 65.5 ± 10.3 mL/h, compared to 67.3 ± 9.74 for LASSO, 68.3 ± 8.45 for random forest, and 75.4 ± 16.3 for the explainable gradient boosting model. Next, we visualized LASSO coefficients and normalized importance scores for the random forest, MARS and explainable gradient boosting models. All four methods correctly identified the true covariates as most important, while noise covariates were less influential (see Figure 1A–D). For the MARS and explainable gradient boosting model, the learned function which approximates the effect of weight on clearance was also visualized (see Figure 1E,F).

## Appendix B. Neural Network for Drug Concentration Prediction

### *Appendix B.1. Data*

A publicly available dataset of 32 patients who received warfarin was used to depict how neural networks can be used to predict drug concentrations. The dataset contained a total of 251 warfarin concentration measurements, with a median of six measurements per patient. Every patient was given a single dose of warfarin at $t = 0$ and measurements were performed at $t \in \{0.25, 0.5, 1.0, 2.0, 4.0, 6.0, 12.0, 24.0, 48.0, 72.0, 96.0, 120.0\}$. Available covariates were patient weight, age, and sex.

### *Appendix B.2. Prediction of Warfarin Concentrations*

The patients from the real-world warfarin dataset were split into a training ($n = 22$) and testing ($n = 10$) set. A neural network with two hidden layers (with, 16, and 4 neurons, respectively) was fit to predict single warfarin concentrations based on patient age, sex, dose, and time point. The swish activation function was used for the hidden layers. Final layer used the softplus activation function in order to constrain output to be positive. Next, we trained an ODE-based neural network [67], which predicts the parameters of a set

of differential equations representing a compartment model. The same neural network architecture was used (hidden layer of 16 neurons and output layer of 4 neurons). Real-world warfarin concentrations were predicted based on patient age and sex. This model was trained on the same train set and accuracy was evaluated on the test set. Naive neural network achieved slightly lower RMSE 1.41 $IUmL^{-1}$ compared to the ODE-based neural network (1.60 $IUmL^{-1}$).

The results for the same test set patient was plotted for the naive neural network model and ODE-based model (Figure 2). We also artificially set the dose to zero for both models and plotted the results. Only the ODE-based model correctly recognized that no concentration response should be expected in this case (Figure 2B).

*Appendix B.3. SHAP Analysis*

We performed a SHAP analysis (ShapML Julia package, v0.3.2, Nick Redell (Chicago, IL, USA)) on the ODE-based model to visualize covariate relationships with absorption rate predictions. Since our neural network has only a single continuous covariate we could directly visualize the predicted absorption rate for a range of age values for male and female patients. These are visualized alongside the SHAP values (Figure 4). We can see that the SHAP values match the prediction by the neural network. However, visualizing the complete functional relationship of patient age allows us to infer the prediction for out-of-distribution data. In this case, the effect of age on the absorption rate can be quite different from the observed data. This is especially true for female patients, of whom the number of samples is low. Here we see a sharp decrease in absorption rate for female patients aged below 25 years, which might lead to poor generalization. However, as we do not have any samples in this age range, we are unable to reliably represent this effect using SHAP values.

## References


1. Beal, S.L.; Sheiner, L.B. Estimating population kinetics. *Crit. Rev. Biomed. Eng.* **1982**, *8*, 195–222. [PubMed]
2. Lindstrom, M.J.; Bates, D.M. Nonlinear mixed effects models for repeated measures data. *Biometrics* **1990**, *46*, 673–687. [CrossRef] [PubMed]
3. Racine-Poon, A.; Smith, A.F. Population models. *Stat. Methodol. Pharm. Sci.* **1990**, *1*, 139 –162.
4. Chaturvedula, A.; Calad-Thomson, S.; Liu, C.; Sale, M.; Gattu, N.; Goyal, N. Artificial intelligence and pharmacometrics: Time to embrace, capitalize, and advance? *CPT: Pharmacometrics Syst. Pharmacol.* **2019**, *8*, 440. [CrossRef]
5. McComb, M.; Bies, R.; Ramanathan, M. Machine learning in pharmacometrics: Opportunities and challenges. *Br. J. Clin. Pharmacol.* **2021**, *88*, 1482–1499. [CrossRef]
6. Osareh, A.; Shadgar, B. Machine learning techniques to diagnose breast cancer. In Proceedings of the IEEE 2010 5th International Symposium on Health Informatics and Bioinformatics, Antalya, Turkey, 20–22 April 2010; pp. 114–120.
7. van IJzendoorn, D.G.; Szuhai, K.; Briaire-de Bruijn, I.H.; Kostine, M.; Kuijjer, M.L.; Bovée, J.V. Machine learning analysis of gene expression data reveals novel diagnostic and prognostic biomarkers and identifies therapeutic targets for soft tissue sarcomas. *PLoS Comput. Biol.* **2019**, *15*, e1006826. [CrossRef]
8. Ishwaran, H.; Kogalur, U.B.; Blackstone, E.H.; Lauer, M.S. Random survival forests. *Ann. Appl. Stat.* **2008**, *2*, 841–860. [CrossRef]
9. Badillo, S.; Banfai, B.; Birzele, F.; Davydov, I.I.; Hutchinson, L.; Kam-Thong, T.; Siebourg-Polster, J.; Steiert, B.; Zhang, J.D. An introduction to machine learning. *Clin. Pharmacol. Ther.* **2020**, *107*, 871–885. [CrossRef]
10. Wu, H.; Wu, L. A multiple imputation method for missing covariates in non-linear mixed-effects models with application to HIV dynamics. *Stat. Med.* **2001**, *20*, 1755–1769. [CrossRef]
11. Johansson, Å.M.; Karlsson, M.O. Comparison of methods for handling missing covariate data. *AAPS J.* **2013**, *15*, 1232–1241. [CrossRef]
12. Bräm, D.S.; Nahum, U.; Atkinson, A.; Koch, G.; Pfister, M. Opportunities of Covariate Data Imputation with Machine Learning for Pharmacometricians in R. In Proceedings of the 30th Annual Meeting of the Population Approach Group in Europe. Abstract 9982. 2022. Available online: www.page-meeting.org/?abstract=9982 (accessed on 15 July 2022).
13. Batista, G.E.; Monard, M.C. An analysis of four missing data treatment methods for supervised learning. *Appl. Artif. Intell.* **2003**, *17*, 519–533. [CrossRef]
14. Stekhoven, D.J.; Bühlmann, P. MissForest—Non-parametric missing value imputation for mixed-type data. *Bioinformatics* **2012**, *28*, 112–118. [CrossRef] [PubMed]
15. Shah, A.D.; Bartlett, J.W.; Carpenter, J.; Nicholas, O.; Hemingway, H. Comparison of random forest and parametric imputation models for imputing missing data using MICE: A CALIBER study. *Am. J. Epidemiol.* **2014**, *179*, 764–774. [CrossRef]

16. Jin, L.; Bi, Y.; Hu, C.; Qu, J.; Shen, S.; Wang, X.; Tian, Y. A comparative study of evaluating missing value imputation methods in label-free proteomics. *Sci. Rep.* **2021**, *11*, 1–11. [CrossRef] [PubMed]
17. van Buuren, S.; Oudshoorn, K. *Flexible Multivariate Imputation by MICE*; TNO Public Health Institution: Leiden, The Netherlands, 1999.
18. Yoon, J.; Jordon, J.; Schaar, M. Gain: Missing data imputation using generative adversarial nets. In Proceedings of the International Conference on Machine Learning (PMLR), Stockholm, Sweden, 10–15 July 2018; pp. 5689–5698.
19. Mattei, P.A.; Frellsen, J. MIWAE: Deep generative modelling and imputation of incomplete datasets. In Proceedings of the International Conference on Machine Learning (PMLR), Long Beach, CA, USA, 9–15 June 2019; pp. 4413–4423.
20. Jafrasteh, B.; Hernández-Lobato, D.; Lubián-López, S.P.; Benavente-Fernández, I. Gaussian Processes for Missing Value Imputation. *arXiv* **2022**, arXiv:2204.04648.
21. Lopes, T.J.; Rios, R.; Nogueira, T.; Mello, R.F. Prediction of hemophilia A severity using a small-input machine-learning framework. *NPJ Syst. Biol. Appl.* **2021**, *7*, 1–8. [CrossRef]
22. McInnes, L.; Healy, J.; Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv* **2018**, arXiv:1802.03426.
23. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2565–2579.
24. Xiang, R.; Wang, W.; Yang, L.; Wang, S.; Xu, C.; Chen, X. A comparison for dimensionality reduction methods of single-cell RNA-seq data. *Front. Genet.* **2021**, *12*, 6936. [CrossRef]
25. Becht, E.; McInnes, L.; Healy, J.; Dutertre, C.A.; Kwok, I.W.; Ng, L.G.; Ginhoux, F.; Newell, E.W. Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.* **2019**, *37*, 38–44. [CrossRef]
26. Ciuculete, D.M.; Bandstein, M.; Benedict, C.; Waeber, G.; Vollenweider, P.; Lind, L.; Schiöth, H.B.; Mwinyi, J. A genetic risk score is significantly associated with statin therapy response in the elderly population. *Clin. Genet.* **2017**, *91*, 379–385. [CrossRef] [PubMed]
27. Kanders, S.H.; Pisanu, C.; Bandstein, M.; Jonsson, J.; Castelao, E.; Pistis, G.; Gholam-Rezaee, M.; Eap, C.B.; Preisig, M.; Schiöth, H.B.; et al. A pharmacogenetic risk score for the evaluation of major depression severity under treatment with antidepressants. *Drug Dev. Res.* **2020**, *81*, 102–113. [CrossRef] [PubMed]
28. Zwep, L.B.; Duisters, K.L.; Jansen, M.; Guo, T.; Meulman, J.J.; Upadhyay, P.J.; van Hasselt, J.C. Identification of high-dimensional omics-derived predictors for tumor growth dynamics using machine learning and pharmacometric modeling. *CPT Pharmacometrics Syst. Pharmacol.* **2021**, *10*, 350–361. [CrossRef] [PubMed]
29. Kapralos, I.; Dokoumetzidis, A. Population Pharmacokinetic Modelling of the Complex Release Kinetics of Octreotide LAR: Defining Sub-Populations by Cluster Analysis. *Pharmaceutics* **2021**, *13*, 1578. [CrossRef] [PubMed]
30. Paul, R.; Andlauer, T.F.; Czamara, D.; Hoehn, D.; Lucae, S.; Pütz, B.; Lewis, C.M.; Uher, R.; Müller-Myhsok, B.; Ising, M.; et al. Treatment response classes in major depressive disorder identified by model-based clustering and validated by clinical prediction models. *Transl. Psychiatry* **2019**, *9*, 1–15. [CrossRef] [PubMed]
31. Tomás, E.; Vinga, S.; Carvalho, A.M. Unsupervised learning of pharmacokinetic responses. *Comput. Stat.* **2017**, *32*, 409–428. [CrossRef]
32. Bunte, K.; Smith, D.J.; Chappell, M.J.; Hassan-Smith, Z.K.; Tomlinson, J.W.; Arlt, W.; Tiňo, P. Learning pharmacokinetic models for in vivo glucocorticoid activation. *J. Theor. Biol.* **2018**, *455*, 222–231. [CrossRef]
33. Chapfuwa, P.; Li, C.; Mehta, N.; Carin, L.; Henao, R. Survival cluster analysis. In Proceedings of the ACM Conference on Health, Inference, and Learning, Toronto, ON, Canada, 2–4 April 2020; pp. 60–68.
34. Guerra, R.P.; Carvalho, A.M.; Mateus, P. Model selection for clustering of pharmacokinetic responses. *Comput. Methods Programs Biomed.* **2018**, *162*, 11–18. [CrossRef]
35. Blömer, J.; Bujna, K. Adaptive seeding for Gaussian mixture models. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Auckland, New Zealand, 19–22 April 2016; pp. 296–308.
36. Harrell, F.E. Regression modeling strategies. *Bios* **2017**, *330*, 14.
37. Ribbing, J.; Jonsson, E.N. Power, selection bias and predictive performance of the Population Pharmacokinetic Covariate Model. *J. Pharmacokinet. Pharmacodyn.* **2004**, *31*, 109–134. [CrossRef]
38. Ribbing, J.; Nyberg, J.; Caster, O.; Jonsson, E.N. The lasso—A novel method for predictive covariate model building in nonlinear mixed effects models. *J. Pharmacokinet. Pharmacodyn.* **2007**, *34*, 485–517. [CrossRef]
39. Ahamadi, M.; Largajolli, A.; Diderichsen, P.M.; de Greef, R.; Kerbusch, T.; Witjes, H.; Chawla, A.; Davis, C.B.; Gheyas, F. Operating characteristics of stepwise covariate selection in pharmacometric modeling. *J. Pharmacokinet. Pharmacodyn.* **2019**, *46*, 273–285. [CrossRef]
40. Tibshirani, R. The lasso method for variable selection in the Cox model. *Stat. Med.* **1997**, *16*, 385–395. [CrossRef]
41. Chan, P.; Zhou, X.; Wang, N.; Liu, Q.; Bruno, R.; Jin, J.Y. Application of Machine Learning for Tumor Growth Inhibition–Overall Survival Modeling Platform. *CPT Pharmacometr. Syst. Pharmacol.* **2021**, *10*, 59–66. [CrossRef]
42. Sibieude, E.; Khandelwal, A.; Hesthaven, J.S.; Girard, P.; Terranova, N. Fast screening of covariates in population models empowered by machine learning. *J. Pharmacokinet. Pharmacodyn.* **2021**, *48*, 597–609. [CrossRef]
43. Friedman, J.H. Multivariate adaptive regression splines. *Ann. Stat.* **1991**, *19*, 1–67. [CrossRef]
44. Ki, D.; Fang, B.; Guntuboyina, A. MARS via LASSO. *arXiv* **2021**, arXiv:2111.11694.

45. Mitov, V.; Kuemmel, A.; Gobeau, N.; Cherkaoui, M.; Bouillon, T. Dose selection by covariate assessment on the optimal dose for efficacy—Application of machine learning in the context of PKPD. In Proceedings of the 30th Annual Meeting of the Population Approach Group in Europe. Abstract 10066. 2022. Available online: www.page-meeting.org/?abstract=10066 (accessed on 15 July 2022).
46. Wang, R.; Shao, X.; Zheng, J.; Saci, A.; Qian, X.; Pak, I.; Roy, A.; Bello, A.; Rizzo, J.I.; Hosein, F.; et al. A machine-learning approach to identify a prognostic cytokine signature that is associated with nivolumab clearance in patients with advanced melanoma. *Clin. Pharmacol. Ther.* **2020**, *107*, 978–987. [CrossRef] [PubMed]
47. Strobl, C.; Boulesteix, A.L.; Zeileis, A.; Hothorn, T. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinform.* **2007**, *8*, 1–21. [CrossRef] [PubMed]
48. Gong, X.; Hu, M.; Zhao, L. Big data toolsets to pharmacometrics: Application of machine learning for time-to-event analysis. *Clin. Transl. Sci.* **2018**, *11*, 305–311. [CrossRef] [PubMed]
49. Lou, Y.; Caruana, R.; Gehrke, J.; Hooker, G. Accurate intelligible models with pairwise interactions. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013; pp. 623–631.
50. Nori, H.; Jenkins, S.; Koch, P.; Caruana, R. InterpretML: A Unified Framework for Machine Learning Interpretability. *arXiv* **2019**, arXiv:1909.09223.
51. Bies, R.R.; Muldoon, M.F.; Pollock, B.G.; Manuck, S.; Smith, G.; Sale, M.E. A genetic algorithm-based, hybrid machine learning approach to model selection. *J. Pharmacokinet. Pharmacodyn.* **2006**, *33*, 195–221. [CrossRef] [PubMed]
52. Ismail, M.; Sale, M.; Yu, Y.; Pillai, N.; Liu, S.; Pflug, B.; Bies, R. Development of a genetic algorithm and NONMEM workbench for automating and improving population pharmacokinetic/pharmacodynamic model selection. *J. Pharmacokinet. Pharmacodyn.* **2021**, *49*, 243–256. [CrossRef]
53. Sibieude, E.; Khandelwal, A.; Girard, P.; Hesthaven, J.S.; Terranova, N. Population pharmacokinetic model selection assisted by machine learning. *J. Pharmacokinet. Pharmacodyn.* **2021**, *49*, 257–270. [CrossRef] [PubMed]
54. Janssen, A.; Hoogendoorn, M.; Cnossen, M.H.; Mathôt, R.A.; Group, O.C.S.; Consortium, S.; Cnossen, M.; Reitsma, S.; Leebeek, F.; Mathôt, R.; et al. Application of SHAP values for inferring the optimal functional form of covariates in pharmacokinetic modeling. *CPT Pharmacometr. Syst. Pharmacol.* **2022**, *11*, 1100–1110. [CrossRef]
55. Xu, Y.; Lou, H.; Chen, J.; Jiang, B.; Yang, D.; Hu, Y.; Ruan, Z. Application of a backpropagation artificial neural network in predicting plasma concentration and pharmacokinetic parameters of oral single-dose rosuvastatin in healthy subjects. *Clin. Pharmacol. Drug Dev.* **2020**, *9*, 867–875. [CrossRef]
56. Pellicer-Valero, O.J.; Cattinelli, I.; Neri, L.; Mari, F.; Martín-Guerrero, J.D.; Barbieri, C. Enhanced prediction of hemoglobin concentration in a very large cohort of hemodialysis patients by means of deep recurrent neural networks. *Artif. Intell. Med.* **2020**, *107*, 101898. [CrossRef]
57. Huang, X.; Yu, Z.; Bu, S.; Lin, Z.; Hao, X.; He, W.; Yu, P.; Wang, Z.; Gao, F.; Zhang, J.; et al. An Ensemble Model for Prediction of Vancomycin Trough Concentrations in Pediatric Patients. *Drug Des. Dev. Ther.* **2021**, *15*, 1549. [CrossRef]
58. Lu, J.; Deng, K.; Zhang, X.; Liu, G.; Guan, Y. Neural-ODE for pharmacokinetics modeling and its advantage to alternative machine learning models in predicting new dosing regimens. *Iscience* **2021**, *24*, 102804. [CrossRef]
59. Ulas, C.; Das, D.; Thrippleton, M.J.; Valdes Hernandez, M.d.C.; Armitage, P.A.; Makin, S.D.; Wardlaw, J.M.; Menze, B.H. Convolutional neural networks for direct inference of pharmacokinetic parameters: Application to stroke dynamic contrast-enhanced MRI. *Front. Neurol.* **2019**, *9*, 1147. [CrossRef]
60. Gim, J.A.; Kwon, Y.; Lee, H.A.; Lee, K.R.; Kim, S.; Choi, Y.; Kim, Y.K.; Lee, H. A Machine Learning-Based Identification of Genes Affecting the Pharmacokinetics of Tacrolimus Using the DMETTM Plus Platform. *Int. J. Mol. Sci.* **2020**, *21*, 2517. [CrossRef]
61. Tao, Y.; Chen, Y.J.; Xue, L.; Xie, C.; Jiang, B.; Zhang, Y. An ensemble model with clustering assumption for warfarin dose prediction in Chinese patients. *IEEE J. Biomed. Health Inform.* **2019**, *23*, 2642–2654. [CrossRef]
62. Woillard, J.B.; Labriffe, M.; Debord, J.; Marquet, P. Tacrolimus exposure prediction using machine learning. *Clin. Pharmacol. Ther.* **2021**, *110*, 361–369. [CrossRef] [PubMed]
63. Huang, X.; Yu, Z.; Wei, X.; Shi, J.; Wang, Y.; Wang, Z.; Chen, J.; Bu, S.; Li, L.; Gao, F.; et al. Prediction of vancomycin dose on high-dimensional data using machine learning techniques. *Expert Rev. Clin. Pharmacol.* **2021**, *14*, 761–771. [CrossRef] [PubMed]
64. Liu, X.; Liu, C.; Huang, R.; Zhu, H.; Liu, Q.; Mitra, S.; Wang, Y. Long short-term memory recurrent neural network for pharmacokinetic-pharmacodynamic modeling. *Int. J. Clin. Pharmacol. Ther.* **2020**, *59*, 138. [CrossRef] [PubMed]
65. Bräm, D.S.; Parrott, N.; Hutchinson, L.; Steiert, B. Introduction of an artificial neural network–based method for concentration-time predictions. *CPT Pharmacometr. Syst. Pharmacol.* **2022**, *11*, 745–754. [CrossRef]
66. Chen, R.T.; Rubanova, Y.; Bettencourt, J.; Duvenaud, D. Neural ordinary differential equations. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 2–8 December 2018; p. 31.
67. Janssen, A.; Leebeek, F.W.; Cnossen, M.H.; Mathôt, R.A.A.; For the OPTI- CLOT study group and SYMPHONY consortium. Deep compartment models: A deep learning approach for the reliable prediction of time-series data in pharmacokinetic modeling. *CPT Pharmacometr. Syst. Pharmacol.* **2022**, *11*, 934–945. [CrossRef]
68. Janssen, A.; Leebeek, F.W.G.; Cnossen, M.H.; Mathôt, R.A.A. The Neural Mixed Effects algorithm: Leveraging machine learning for pharmacokinetic modelling. In Proceedings of the 29th Annual Meeting of the Population Approach Group in Europe. Abstract 9826. 2021. Available online: www.page-meeting.org/?abstract=9826 (accessed on 19 July 2022).

69. Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–12.
70. Fort, S.; Hu, H.; Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv* **2019**, arXiv:1912.02757.
71. Zou, H.; Banerjee, P.; Leung, S.S.Y.; Yan, X. Application of pharmacokinetic-pharmacodynamic modeling in drug delivery: Development and challenges. *Front. Pharmacol.* **2020**, *11*, 997. [CrossRef]
72. Danhof, M.; de Lange, E.C.; Della Pasqua, O.E.; Ploeger, B.A.; Voskuyl, R.A. Mechanism-based pharmacokinetic-pharmacodynamic (PK-PD) modeling in translational drug research. *Trends Pharmacol. Sci.* **2008**, *29*, 186–191. [CrossRef] [PubMed]
73. Lu, J.; Bender, B.; Jin, J.Y.; Guan, Y. Deep learning prediction of patient response time course from early data via neural-pharmacokinetic/pharmacodynamic modelling. *Nat. Mach. Intell.* **2021**, *3*, 696–704. [CrossRef]
74. Kurz, D.; Sánchez, C.S.; Axenie, C. Data-driven Discovery of Mathematical and Physical Relations in Oncology Data using Human-understandable Machine Learning. *Front. Artif. Intell.* **2021**, *4*, 713690. [CrossRef] [PubMed]
75. Qian, Z.; Zame, W.R.; van der Schaar, M.; Fleuren, L.M.; Elbers, P. Integrating Expert ODEs into Neural ODEs: Pharmacology and Disease Progression. *arXiv* **2021**, arXiv:2106.02875.
76. Wong, H.; Alicke, B.; West, K.A.; Pacheco, P.; La, H.; Januario, T.; Yauch, R.L.; de Sauvage, F.J.; Gould, S.E. Pharmacokinetic–pharmacodynamic analysis of vismodegib in preclinical models of mutational and ligand-dependent Hedgehog pathway activation. *Clin. Cancer Res.* **2011**, *17*, 4682–4692. [CrossRef]
77. Randall, E.C.; Emdal, K.B.; Laramy, J.K.; Kim, M.; Roos, A.; Calligaris, D.; Regan, M.S.; Gupta, S.K.; Mladek, A.C.; Carlson, B.L.; et al. Integrated mapping of pharmacokinetics and pharmacodynamics in a patient-derived xenograft model of glioblastoma. *Nat. Commun.* **2018**, *9*, 1–13. [CrossRef]
78. Kong, J.; Lee, H.; Kim, D.; Han, S.K.; Ha, D.; Shin, K.; Kim, S. Network-based machine learning in colorectal and bladder organoid models predicts anti-cancer drug efficacy in patients. *Nat. Commun.* **2020**, *11*, 1–13. [CrossRef]
79. Chiu, Y.C.; Chen, H.I.H.; Zhang, T.; Zhang, S.; Gorthi, A.; Wang, L.J.; Huang, Y.; Chen, Y. Predicting drug response of tumors from integrated genomic profiles by deep neural networks. *BMC Med. Genom.* **2019**, *12*, 143–155.
80. Wang, D.; Hensman, J.; Kutkaite, G.; Toh, T.S.; Galhoz, A.; Dry, J.R.; Saez-Rodriguez, J.; Garnett, M.J.; Menden, M.P.; Dondelinger, F.; et al. A statistical framework for assessing pharmacological responses and biomarkers using uncertainty estimates. *Elife* **2020**, *9*, e60352. [CrossRef]
81. Keyvanpour, M.R.; Shirzad, M.B. An analysis of QSAR research based on machine learning concepts. *Curr. Drug Discov. Technol.* **2021**, *18*, 17–30. [CrossRef]
82. Katzman, J.L.; Shaham, U.; Cloninger, A.; Bates, J.; Jiang, T.; Kluger, Y. DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Med. Res. Methodol.* **2018**, *18*, 1–12. [CrossRef] [PubMed]
83. Lee, C.; Zame, W.R.; Yoon, J.; van der Schaar, M. Deephit: A deep learning approach to survival analysis with competing risks. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
84. Ren, K.; Qin, J.; Zheng, L.; Yang, Z.; Zhang, W.; Qiu, L.; Yu, Y. Deep recurrent survival analysis. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4798–4805.
85. Giunchiglia, E.; Nemchenko, A.; van der Schaar, M. RNN-SURV: A deep recurrent model for survival analysis. In Proceedings of the International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; pp. 23–32.
86. Meira-Machado, L.; de Uña-Álvarez, J.; Cadarso-Suárez, C.; Andersen, P.K. Multi-state models for the analysis of time-to-event data. *Stat. Methods Med. Res.* **2009**, *18*, 195–222. [CrossRef] [PubMed]
87. Gerstung, M.; Papaemmanuil, E.; Martincorena, I.; Bullinger, L.; Gaidzik, V.I.; Paschka, P.; Heuser, M.; Thol, F.; Bolli, N.; Ganly, P.; et al. Precision oncology for acute myeloid leukemia using a knowledge bank approach. *Nat. Genet.* **2017**, *49*, 332–340. [CrossRef] [PubMed]
88. Groha, S.; Schmon, S.M.; Gusev, A. A General Framework for Survival Analysis and Multi-State Modelling. *arXiv* **2020**, arXiv:2006.04893.
89. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [CrossRef]
90. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning requires rethinking generalization (2016). *arXiv* **2017**, arXiv:1611.03530.
91. Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the IJCAI, Montreal, ON, Canada, 20–25 August 1995; Volume 14, pp. 1137–1145.
92. Molinaro, A.M.; Simon, R.; Pfeiffer, R.M. Prediction error estimation: A comparison of resampling methods. *Bioinformatics* **2005**, *21*, 3301–3307. [CrossRef]
93. Gronau, Q.F.; Wagenmakers, E.J. Limitations of Bayesian leave-one-out cross-validation for model selection. *Comput. Brain Behav.* **2019**, *2*, 1–11. [CrossRef]
94. Arlot, S.; Celisse, A. A survey of cross-validation procedures for model selection. *Stat. Surv.* **2010**, *4*, 40–79. [CrossRef]
95. Zhang, Z.; Xie, Y.; Xing, F.; McGough, M.; Yang, L. Mdnet: A semantically and visually interpretable medical image diagnosis network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6428–6436.

96. Singh, A.; Mohammed, A.R.; Zelek, J.; Lakshminarayanan, V. Interpretation of deep learning using attributions: Application to ophthalmic diagnosis. In *Proceedings of the Applications of Machine Learning 2020*; SPIE: London, UK, 2020; Volume 11511, pp. 39–49.
97. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning important features through propagating activation differences. In Proceedings of the International Conference on Machine Learning (PMLR), Sydney, Australia, 6–11 August 2017; pp. 3145–3153.
98. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
99. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *31*, 4768–4777.
100. Holzinger, A.; Saranti, A.; Molnar, C.; Biecek, P.; Samek, W. Explainable AI methods-a brief overview. In Proceedings of the International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers, Vienna, Austria, 17 July 2020; pp. 13–38.
101. Ogami, C.; Tsuji, Y.; Seki, H.; Kawano, H.; To, H.; Matsumoto, Y.; Hosono, H. An artificial neural network- pharmacokinetic model and its interpretation using Shapley additive explanations. *CPT Pharmacometr. Syst. Pharmacol.* **2021**, *10*, 760–768. [CrossRef] [PubMed]
102. Hafner, D.; Tran, D.; Lillicrap, T.; Irpan, A.; Davidson, J. Noise contrastive priors for functional uncertainty. In Proceedings of the Uncertainty in Artificial Intelligence (PMLR), Online, 3–6 August 2020; pp. 905–914.
103. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [CrossRef] [PubMed]
104. Molnar, C.; König, G.; Herbinger, J.; Freiesleben, T.; Dandl, S.; Scholbeck, C.A.; Casalicchio, G.; Grosse-Wentrup, M.; Bischl, B. General pitfalls of model-agnostic interpretation methods for machine learning models. In Proceedings of the International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers, Vienna, Austria, 17 July 2020; pp. 39–68.
105. Yang, G.; Ye, Q.; Xia, J. Unbox the black-box for the medical explainable ai via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond. *Inform. Fus.* **2022**, *77*, 29–52. [CrossRef] [PubMed]
106. Björkman, S.; Oh, M.; Spotts, G.; Schroth, P.; Fritsch, S.; Ewenstein, B.M.; Casey, K.; Fischer, K.; Blanchette, V.S.; Collins, P.W. Population pharmacokinetics of recombinant factor VIII: The relationships of pharmacokinetics to age and body weight. *Blood, J. Am. Soc. Hematol.* **2012**, *119*, 612–618. [CrossRef] [PubMed]
107. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.