Symbolic neural networks for automated covariate modeling in a mixed-effects framework

Sundell, Jesper; Wahlquist, Ylva; Soltesz, Kristian

Published in:
IFAC-PapersOnLine

2024

Document Version:
Peer reviewed version (aka post-print)

Symbolic neural networks for automated covariate modeling in a mixed-effects framework

Jesper Sundell [*,**] **Ylva Wahlquist** [*,**] **Kristian Soltesz** [**]

[*] *These authors contributed equally to this work.*
[**] *Lund University, Dept. Automatic Control, Sweden*
(e-mail: {first.last}@control.lth.se)

Abstract: Mixed-effects models are used to describe the inter-patient variability in drugs. Modeling of these variabilities include both fixed and random effects. Fixed effects relate covariates such as age and weight to compartment volumes and clearances, whereas random effects account for unexplained variability. Traditionally, the development of fixed effects models is an inefficient process where covariate relationships are evaluated in a step-wise manner. In this study, we implemented a symbolic neural network (SNN) to automate the development of a fixed effects model and used it to develop a population pharmacokinetic model for propofol. With the SNN, we can find covariate relationships that are traditionally not evaluated. Then, we apply random effects and estimate parameters in the standard mixed-effects modeling framework. Our final model shows comparable predictive performance to a published model for propofol, despite having fewer covariates and model parameters.

Keywords: Neural networks, mixed-effects modeling, pharmacokinetics, pharmacometrics, covariate models

1. INTRODUCTION

Modeling and simulation have become intrinsic to drug development. Non-linear mixed-effects (NLME) models are used to describe the pharmacokinetics (PK) and pharmacodynamics of drugs. The models are utilized for dose selection in clinical trials and to inform individual adaptations in dose regimens (Mould and Upton, 2012). Such models contain fixed effects (parameters describing the central tendencies of the population) and random effects accounting for individual differences. Variability between individuals can be described by models that map covariates, such as age or gender, to PK parameters. Such covariate models are typically formulated to depend on a set of optimizable parameters. The remaining inter-individual variability (IIV), not explained by the fixed effects, is ascribed to random effects.

For drug delivery systems, such as target-controlled infusion (TCI) or closed-loop control, the models are commonly used without random effects (Schnider et al., 1998; White et al., 2008). However, considering random effects in automated systems can increase safety for a broader population. We therefore see a need for modeling tool chains from data set to control system design that explicitly take random effects into account. In this paper, we present a crucial link in such a tool-chain: a methodology for automating the covariate modeling step.

⋆ This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. All authors are members of the ELLIIT Strategic Research Area at Lund University.

Covariate models are traditionally developed through an iterative process where simple functions mapping covariates to PK parameters are evaluated one by one (Wählby et al., 2002). The evaluation is focused on the identification of relevant covariates and any profound exploration of the functional relationship between covariates and parameters is seldom performed. As the number of covariates that are evaluated increases, the evaluation further becomes increasingly ineffective due to its combinatorial, iterative nature. It also increases the risk of overfitting.

We have developed a method for the automatic identification of influential covariates and their functional mapping to parameters of pharmacometric models in Wahlquist et al. (2023). The method is based on neural networks with customized activation functions. By iterative pruning, we can obtain a sparse symbolic neural network (SNN) that represents a simple covariate expression. Such methodology allows for exploring covariate relationships that are not commonly considered in the traditional framework. However, the method was developed without consideration of random effects.

In the present study, we evaluated the SNN modeling method in the context of a mixed-effects model. We use the covariate model structure produced by the SNN and apply a standard pharmacometric mixed-effect modeling framework (Wählby et al., 2002). This involves assigning parameterized prior distributions to model parameters, and then estimating the parameters by maximizing the likelihood of the training data.

The method is demonstrated by the development of a population PK model for the anesthetic drug propofol. Fur-

thermore, it was compared to a previously published model developed with the state-of-the-art modeling methodology on the same underlying data set (White et al., 2008). The main advantage of our method compared to the state-of-the-art is the automation in the model search, which enables consideration of a large number of low-complexity models.

2. METHODS

2.1 Population PK models

A pharmacokinetic (PK) model describes the evolution of drug concentration in communicating compartments that partition the patient's body. In the example of this paper, and likewise White et al. (2008), we consider a PK model for propofol with three compartments as depicted in Fig. 1. Letting $\boldsymbol{c} = [c_1, c_2, c_3]^\top$ denote the drug concentration of each compartment, the PK model can be formulated as a linear differential equation

$$\dot{\boldsymbol{c}}(t) = \underbrace{\begin{bmatrix} -\dfrac{CL+Q_2+Q_3}{V_1} & \dfrac{Q_2}{V_2} & \dfrac{Q_3}{V_3} \\ \dfrac{Q_2}{V_1} & -\dfrac{Q_2}{V_2} & 0 \\ \dfrac{Q_3}{V_1} & 0 & -\dfrac{Q_3}{V_3} \end{bmatrix}}_{A} \boldsymbol{c}(t) + \underbrace{\begin{bmatrix} \dfrac{1}{V_1} \\ 0 \\ 0 \end{bmatrix}}_{B} u(t), \tag{1}$$

where $u(t)$ is the per-time addition of the drug to the first compartment, modeling the blood plasma. The PK parameter vector $\boldsymbol{p} = [V_1, CL, V_2, V_3, Q_2, Q_3]^\top$ parameterizes A and B.

Due to inter-individual variability, the predictive performance of the PK model can be improved by assigning individual-specific values of $\boldsymbol{p}_i$ for an individual i. This problem can be approached by letting $\boldsymbol{p}_i$ depend on a set of known (observable) individual-specific covariates $\boldsymbol{\varphi}_i$. The covariate set used both by (White et al., 2008) and us is

$$\boldsymbol{\varphi} = [\text{age}, \text{weight}, \text{gender}]^\top. \tag{2}$$

The covariate relationship for p_{ij}, being the j^{th} PK parameter of the i^{th} individual, is modeled as

$$p_{ij} = f_j(\boldsymbol{\varphi}_i; \boldsymbol{\theta}) \exp(\eta_{ij}), \tag{3}$$

where f_j is an ordinary function parameterized in a vector $\boldsymbol{\theta}$, referred to as the fixed effects, and the random effects η_{ij} are assumed $\mathcal{N}(0, \omega_j^2)$.

For each individual in a PK modeling data set, there is a series of observations. In (White et al., 2008), these consist of drug concentration measurements in blood plasma. We assume normal measurement errors that are either additive (4a) or proportional (4b)

$$y_{ik} = c_1 + \epsilon_{ik}, \tag{4a}$$
$$y_{ik} = c_1(1 + \epsilon_{ik}), \tag{4b}$$

where k indexes measurements, i indexes individuals, and ϵ_{ik} is assumed $\mathcal{N}(0, \sigma^2)$.

In our context, pharmacometric mixed-effect modeling is the process of finding $\boldsymbol{\theta}$, $\boldsymbol{\omega} = [\omega_1, \dots \omega_{n_p}]^\top$ (where n_p is the number of PK parameters), that maximizes the likelihood of the parameters conditional on the measurement data.

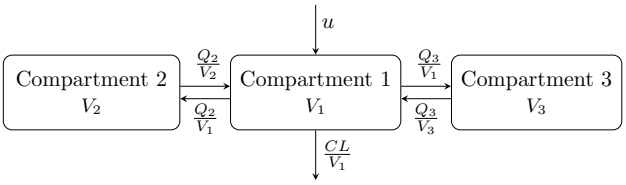

Fig. 1. Three-compartment mammillary PK model. The compartmental volumes are V_1, V_2, V_3 and the corresponding clearances are CL, Q_2, Q_3. Drug is administered at volumetric rate u and eliminated from the central compartment at rate CL/V_1.

In this work, we first developed a base model consisting of the structural model (1), residual error (4), and IIVs (3). Secondly, functions relating covariates to PK parameters $\boldsymbol{f} = [f_1, \dots, f_{n_p}]^\top$ were identified using symbolic neural networks (SNNs). Lastly, the two were combined into a final population PK model.

2.2 Base model

The base model was defined as (3) without covariates. For example, the clearance PK parameter CL is stochastically modeled as

$$CL = \theta_{CL} \exp(\eta_{CL}), \tag{5}$$

where θ_{CL} is the fixed effect associated with clearance and η_{CL} is the random effect, with $\eta_{CL} \sim \mathcal{N}(0, \omega_{CL}^2)$. The base model was used as a reference to determine if the addition of covariates improved model fit to observations.

The objective function value (OFV) was used for the comparison of models which included random effects. The OFV is defined as the $-2\log$-likelihood of the parameters (θ_{CL} and ω_{CL} in the above clearance example) conditional on the data. The likelihood ratio test may therefore be used to compare two nested models, where one is a special case of the other with some parameters fixed. The difference in OFV between the two nested models is approximately χ^2 distributed. A decrease in OFV by -3.84 is hence equivalent to a statistically significant model improvement by $p = 0.05$ for one degree of freedom (one parameter difference) (White et al., 1992).

Both an additive and proportional residual error (4) were evaluated (Mould and Upton, 2013). IIVs were evaluated on all six PK parameters. IIVs that did not statistically improve the model according to a decrease in the objective function value (OFV) were removed from the model.

2.3 Covariate model

Finding the covariate model, consists of identifying a function f_j that relates the covariate vector $\boldsymbol{\varphi}_i$ to the PK parameter p_{ij}(3). In this study, we identified the covariate expressions f_1 and f_2 of

$$p_{i1} = V_1 = f_1(\boldsymbol{\varphi}_i), \tag{6a}$$
$$p_{i2} = CL = f_2(\boldsymbol{\varphi}_i), \tag{6b}$$

where $\boldsymbol{\varphi}_i$ is the covariate vector for the i^{th} individual. We used the covariates of (2), which was also used by White et al. (2008). In the same study (White et al., 2008),

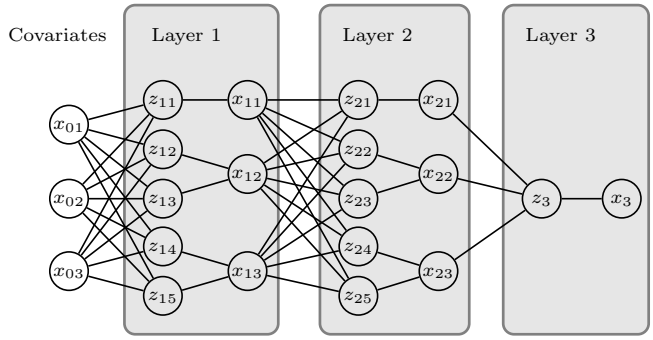

Fig. 2. Symbolic neural network (SNN) representing a covariate expression, relating covariates (here three) to PK parameter $p_{ij} = f_j(\boldsymbol{\varphi}_i)$. The figure represents the nominal structure of the SNN before pruning with three dense layers and following activation functions. $\boldsymbol{z}_l$ is the output from the dense layer (7) and $\boldsymbol{x}_l$ is the output from the activation function of layer l (8).

covariate relationships were identified using a traditional fixed effects modeling framework for the central compartment volume V_1 and clearance CL. While we aimed to find functional covariate expressions for the PK parameters in (6), the other PK parameters for volumes and clearances p_{ij}, where $j = 3, 4, 5, 6$, were estimated as constants during the development of the covariate model.

During model development, we scaled the input covariates. The continuous covariates (age and weight) were scaled between 0 and 1. Gender was modeled as a binary covariate 0.5 for male and -0.5 for female. Thus, all covariate values were roughly of the same magnitude to not favor any covariates based on their scale.

The SNN structure used in this work is illustrated in Fig. 2. We used an SNN with $n_l = 3$ dense layers, each followed by activation functions. We denote the input covariate vector $\boldsymbol{x}_0 = \boldsymbol{\varphi}$. The output from each dense layer is given by

$$\boldsymbol{z}_l = W_l \boldsymbol{x}_{l-1} + \boldsymbol{b}_l, \tag{7}$$

where the input vector of dense layer l is given by $\boldsymbol{x}_{l-1} = [x_{(l-1)1}, x_{(l-1)2}, ...]^\top$ and the output vector is $\boldsymbol{z}_l = [z_{l1}, z_{l2}, ...]^\top$. The weight matrix and bias vector of dense layer l are denoted W_l and $\boldsymbol{b}_l$. The activation function vector $\boldsymbol{h}$ is applied to the outputs of the dense layer, so that

$$\boldsymbol{x}_l = \boldsymbol{h}_l(\boldsymbol{z}_l) \tag{8}$$

where the activation functions of each layer $\boldsymbol{h}_l$ are given by

$$\boldsymbol{h}_1(\boldsymbol{z}_1) = \begin{bmatrix} z_{11} \\ z_{12} z_{13} \\ |z_{14}|^{z_{15}} \end{bmatrix} \tag{9a}$$

$$\boldsymbol{h}_2(\boldsymbol{z}_2) = \begin{bmatrix} z_{21} \\ z_{22} z_{23} \\ \frac{z_{24}}{z_{25}+1} \end{bmatrix} \tag{9b}$$

$$h_3(z_3) = |z_3|. \tag{9c}$$

The last activation function h_3 assures positive output of the final layer. The division in $\boldsymbol{h}_2$ has the term one in the denominator to ensure that the output does not blow up if $z_{25} \geq 0$ approaches zero. The SNN outputs the PK parameter $p_{ij} = x_3$. The activation functions were chosen based on previously published PK models for propofol (Eleveld et al., 2018; White et al., 2008). A detailed description of the SNN is provided in Wahlquist et al. (2023).

In the covariate modeling process, the PK parameter estimates $\boldsymbol{p}$ were iteratively evaluated through simulation of the corresponding infusion profile to obtain a predicted blood concentration profile. Training of the SNN means updating trainable network parameters $\boldsymbol{\gamma}$ (linear weights W_l and biases $\boldsymbol{b}_l$ in (7)) to minimize an error, expressed through the loss function. For the training of the SNN, we used the average of the median absolute logarithmic error (Mean(MdALE)). For an individual i, the loss is given by

$$\text{MdALE}_i = \text{Median}\left(|\ln(y_{ik}/(c_{\text{pred}})_{ik}|)\right), \ k = 1, ..., n_k \tag{10}$$

where y_{ik} are observed (measured) plasma concentrations and $(c_{\text{pred}})_{ik}$ are corresponding predictions produced by our model. The number entries of the measurement time series is denoted n_k. Over the population, the average error becomes

$$\text{Mean(MdALE)} = \frac{1}{n} \sum_{i=1}^{n} \text{MdALE}_i \tag{11}$$

where n is the number of individuals in the data set. We trained the SNN with backpropagation with the stochastic gradient-based optimization algorithm ADAM introduced by Kingma and Ba (2017).

To obtain simple readable covariate expressions, we pruned the SNN. Pruning is defined as removing parameters from $\boldsymbol{\gamma}$, i.e., setting them to zero and excluding them from further training. It is desirable to remove those that have relatively little influence on the loss (11). We alternated between parameter training and parameter pruning to successively remove parameters from $\boldsymbol{\gamma}$. Before each pruning iteration, we trained the network until convergence. This relates to finding a local minima of (11). In such local minima, the partial derivatives of the loss function with respect to $\boldsymbol{\gamma}$ are zero. Then, the second-order derivatives describe local parameter sensitivity, as further explained in LeCun et al. (1989). The second-order derivatives make up the Hessian matrix, and its diagonal elements represent the sensitivity in the individual parameters of $\boldsymbol{\gamma}$. The second-order terms take on the form

$$S(\gamma_k) = \gamma_k^2 H_k \tag{12}$$

where H_k is the k^{th} Hessian diagonal element of the loss function with respect to the parameter γ_k. $S(\gamma_k)$ denote the salience of a parameter γ_k.

A summary of the training and pruning steps is:

(1) Choose a nominal SNN architecture and the corresponding activation functions
(2) Train the network (until convergence).
(3) Compute the salience $S(\gamma_k)$ of each (trainable/ remaining) parameter of (12). Sort the parameters by salience and remove the N parameters with the smallest salience.
(4) Repeat steps 2 and 3 with $N = 1$ until the expression has reached a desired size (i.e., a desired number of parameters left).
(5) Train the network (until convergence).

(6) Convert the resulting network to a readable function representing our covariate expression.
(7) If possible, simplify the covariate expression.

During the first pruning step, several parameters were removed to accelerate training. During training, we used the learning rate of 0.005 and trained for 5000 epochs. At the initial pruning iteration, we removed $N = 10$ network parameters, and then $N = 1$ at a time until 10 parameters remained in each of the two covariate expressions f_1 and f_2. The initialization of the parameters of the SNN affects the resulting final expression. Thus, we ran the SNN training and pruning several times (here eight). If we obtained several models with similar fit to data (i.e lowest value of the loss (11)), we chose the covariate model with the simplest expression. After obtaining the expressions from the SNN, we investigated if the final model could be simplified further without worsening the fit to data. If so, the model was replaced with its simpler form.

2.4 Covariate model with random effects

In the development of the base model, the type of residual error, as well as significant IIVs, were identified based on reduction in the OFV according to Section 2.2. Then, the significant IIVs were applied to the PK covariate expressions for V_1 and CL (f_1 and f_2) that were obtained in the SNN training. For the other PK parameters (V_2, V_3, Q_2, and Q_3), the PK expressions were the same as for the base model (5), i.e., not including any covariates.

When the significant IIVs were combined with the covariate model according to (3), the fixed effects parameters $\boldsymbol{\theta}$ of the covariate expressions were re-optimized with Pumas. After the estimation of random effects and re-estimation of fixed effects $\boldsymbol{\theta}$, we obtained the final population PK model.

2.5 Software

This work was carried out in the Julia language (Bezanson et al., 2017). The NLME package Pumas (version 2.4) (Rackauckas et al., 2020) was used for the development of the base model and re-estimation of the model parameters of the final PK population model with random effects. The first order conditional estimation with an interaction algorithm was used for model fitting.

The full code implementation is enclosed in Sundell and Wahlquist (2024).

2.6 Data set

The proposed method was applied to a data set for the anesthetic drug propofol from White et al. (2008), hereby referred to as the White data set. The data set was chosen due to that propofol has well-studied pharmacokinetics and that the data set was openly disclosed by Eleveld et al. (2018). Ethical approval of the underlying study is declared in the original publication in White et al. (2008).

The data set is composed of propofol plasma concentration observations and infusion profiles from 107 individuals. The data set contains 1505 plasma concentration observations. Out of the 107 individuals, 54 are males and 53 are females with ages ranging from 17 to 88 years, and weights ranging from 42 to 100 kg. The data was pre-processed so that data points corresponding to subsequent infusion changes shorter than 1 s or smaller than 0.5 $\mu g\,s^{-1}$ were merged.

3. RESULTS

The final base model included a proportional error to describe the residual error. The proportional error was selected based on reduction in the OFV for the base model, which was 301.1 compared to the corresponding model with an additive error. Similarly, statistically significant IIVs were identified for V_1, V_2, CL and Q_2. Inclusion of IIVs on V_3 and Q_3 did not significantly improve model fit.

Table 1. Estimated parameter values for the final population PK model in (14).

Parameter	Estimated value	Relative standard error %
θ_1	0.13	12.5
θ_2	0.1	101.4
θ_3	0.45	321.1
θ_4	-0.0095	59.5
θ_5	0.02	29.3
θ_6	7.78	81.3
θ_7	1.08	52.6
θ_8	126.7	408.9
θ_9	0.89	79.4
σ	0.15	38.2

Table 2. Inter-patient variabilities (IIVs) and their variances of the final population PK model in (14).

Parameter	IIV	Variance (ω^2)	Relative standard error %
η_1	V_1	0.66	13.8
η_2	CL	0.27	153.4
η_3	V_2	0.77	56.3
η_4	Q_2	0.28	2893.4

Table 3. Prediction errors (mean(MdALE), (11)) for three population PK models without random effects ($\boldsymbol{\eta} = \mathbf{0}$) and objective function values (OFV) for the same models with random effects.

Model	mean(MdALE) ($\boldsymbol{\eta} = \mathbf{0}$)	OFV
Base model	0.196	1504.5
White model	0.174	1466.6
SNN model	0.172	1467.3

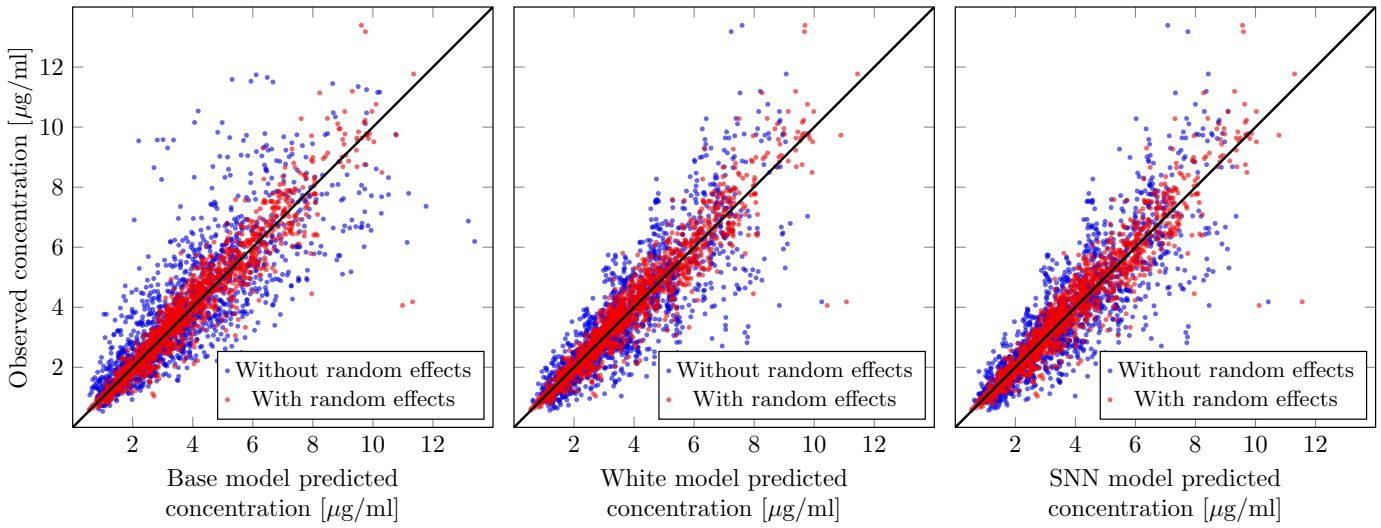

Fig. 3. Model predictions vs observations for the base model, the White model, and our SNN model, with and without random effects.

Allowing for a maximum of 10 parameters for each expression of V_1 and CL in the final covariate model, the SNN search for covariate functional relationships resulted in the following covariate expressions for V_1 and CL:

$$V'_{1,\text{male}} = \frac{8668.9 + 36.1\text{WGT} - 0.030\text{WGT}^2}{1165.9 - \text{WGT}} \quad (13a)$$

$$V'_{1,\text{female}} = \frac{8668.9 - 36.1\text{WGT} - 0.030\text{WGT}^2}{1165.9 + \text{WGT}} \quad (13b)$$

$$CL'_{\text{male}} = 0.023 - 0.00015\text{AGE} + 0.00032\text{WGT} \quad (13c)$$

$$CL'_{\text{female}} = 0.021 - 0.00015\text{AGE} + 0.00032\text{WGT}. \quad (13d)$$

The output covariate expressions of the SNN for CL in (13) were simplified without affecting model predictive performance, i.e., no change in OFV. When the covariate model parameters were re-estimated and random effects added in Pumas, we saw that the intercept variable in the linear relationship of V_1 could be removed without change in OFV. The final population PK model with random effects became:

$$V_{1,\text{male}} = \theta_1 \text{WGT} \exp(\eta_1) \quad (14a)$$

$$V_{1,\text{female}} = \theta_2 \text{WGT} \exp(\eta_1) \quad (14b)$$

$$CL = (\theta_3 + \theta_4 \text{AGE} + \theta_5 \text{WGT}) \exp(\eta_2) \quad (14c)$$

$$V_2 = \theta_6 \exp(\eta_3) \quad (14d)$$

$$Q_2 = \theta_7 \exp(\eta_4) \quad (14e)$$

$$V_3 = \theta_8 \quad (14f)$$

$$Q_3 = \theta_9, \quad (14g)$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ are the parameter estimates. Parameter estimates for $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ with the corresponding standard errors are summarized in Table 1 and Table 2, respectively. The standard deviation of the proportional residual error estimate (4b) is presented in Table 1. All compartment volumes have units of L and all clearances have units of $\text{L}\,\text{min}^{-1}$.

In Table 3, the prediction errors (11) of the base model, the White model, and our SNN model with and without random effects (i.e., $\boldsymbol{\eta} = \mathbf{0}$) are presented. Mean(MDALE) was the measure for model fit in the covariate model development, the training loss for the SNN. OFV ($-2\log$-likelihood) was the measure of model fit in Pumas for the modeling of random effects. Note that the model parameters of the White model were re-estimated when we applied random effects to the model, as random effects were not available in the original publication of White et al. (2008).

Based on the evaluated prediction errors in Table 3, both the White model and the SNN model more adequately described data than the base model. The OFV for the SNN model was lower than for the base model equivalent to a statistically significant model improvement by $p < 0.01$ for three degrees of freedom, corresponding to a difference in three parameters. The White model and SNN model predictions were equivalent both with and without random effects. However, the SNN model includes one covariate less for each of V_1 and CL and three parameters less in total in the covariate model, as seen in (14a) to (14c). Predicted versus observed plasma concentrations for all models, with and without random effects, are illustrated in Fig. 3.

4. DISCUSSION

In the present paper, we demonstrate how traditional NLME modeling may be used in combination with SNNs for the development of population PK models with covariates. Unlike standard approaches, SNNs offer a flexible approach for identifying functions mapping covariates to PK parameters and do not require assumptions of the function structure. The lack of assumptions further makes the identification of complex functions supported by the data possible. For validation, the final model was compared to a previously published model developed on the same data set. Although the comparison was not the main objective of this work, the validation demonstrated that our method shows comparable performance to the standard state-of-the-art modeling methodology.

There is a trade-off between accuracy in fitting data and the model complexity, which is determined by the number of final parameters of the SNN. We chose to limit the number of network parameters to a maximum of 10 for SNN, to obtain simple covariate expressions. Consequently, our

final population PK model is a less complex model than the White model regarding the number of covariates and parameters. A less complex model results in greater readability and generalizability. While this reduces the risk of overfitting, it does not eliminate it. A sound practice is therefore to perform cross-validation as we have demonstrated in Wahlquist et al. (2023). Although standard artificial neural networks may greatly outperform pruned models in the prediction of the training data, such models would exhibit poor generalizability and thus result in poor applicability. Optimization of the pruning strategy with regards to the number of parameters to prune dependent on the complexity of the data set may therefore be of interest to investigate further.

Interestingly, we found linear functions to adequately map covariates to PK parameters similar to White et al. For smaller data sets such as the data set used in this study, the data may only support the identification of simple covariate expressions. However, for larger PK data sets, SNNs offer the possibility to identify complex covariate expressions that are unlikely to be identified using the current standard methodology for covariate model development, as was demonstrated in Wahlquist et al. (2023).

The precision of estimated parameters as quantified by the standard errors, were overall poor for both the SNN model and the White model (White model values not presented here). Poor parameter precision is indicative of poor identifiability of parameters. Furthermore, the poor parameter precision may indicate a weak relationship between covariates and PK parameters. However, since the aim of this work was to develop the SNN methodology, we did not perform any further analysis of which parameters that caused the poor precision.

In the development of the covariate model, we used a loss function that had previously been applied to develop a state-of-the-art population PK model for propofol (Eleveld et al., 2018). However, other loss functions may be equally suitable for finding covariate functions with SNNs. Therefore, the modeler may tailor the loss function depending on the data and model.

In conclusion, we demonstrated an automated methodology for the development of covariate models without any assumptions on the structure of the covariate functions in an NLME framework. The final model developed by the method performed equivalently to a previously published model with fewer parameters. The presented methodology could become a powerful tool in precision dosing including drug delivery systems.

REFERENCES

Bezanson, J., Edelman, A., Karpinski, S., and Shah, V.B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98. doi:10.1137/141000671.

Eleveld, D.J., Colin, P., Absalom, A.R., and Struys, M.M.R.F. (2018). Pharmacokinetic–pharmacodynamic model for propofol for broad application in anaesthesia and sedation. *Br J Anaesth*, 120(5), 942–959. doi:10.1016/j.bja.2018.01.018.

Kingma, D.P. and Ba, J. (2017). Adam: A method for stochastic optimization. doi:10.48550/arXiv.1412.6980.

LeCun, Y., Denker, J., and Solla, S. (1989). Optimal brain damage. In D. Touretzky (ed.), *Advances in Neural Information Processing Systems*, volume 2.

Mould, D. and Upton, R. (2012). Basic concepts in population modeling, simulation, and model-based drug development. *CPT Pharmacometrics Syst Pharmacol*, 1(9), 6. doi:10.1038/psp.2012.4.

Mould, D. and Upton, R. (2013). Basic concepts in population modeling, simulation, and model-based drug development—part 2: Introduction to pharmacokinetic modeling methods. *CPT Pharmacometrics Syst Pharmacol*, 2(4), 38. doi:10.1038/psp.2013.14.

Rackauckas, C., Ma, Y., Noack, A., Dixit, V., Mogensen, P.K., Elrod, C., Tarek, M., Byrne, S., Maddhashiya, S., Calderón, J.B.S., Hatherly, M., Nyberg, J., Gobburu, J.V., and Ivaturi, V. (2020). Accelerated predictive healthcare analytics with pumas, a high performance pharmaceutical modeling and simulation platform. Preprint, Pharmacology and Toxicology. doi:10.1101/2020.11.28.402297.

Schnider, T.W., Minto, C.F., Gambus, P.L., Andresen, C., Goodale, D.B., Shafer, S.L., and Youngs, E.J. (1998). The Influence of Method of Administration and Covariates on the Pharmacokinetics of Propofol in Adult Volunteers. *Anesthesiology*, 88(5), 1170–1182. doi:10.1097/00000542-199805000-00006.

Sundell, J. and Wahlquist, Y. (2024). URL `https://github.com/jespersundell/symbolic-neural-networks-NLME`. Commit: a415d91.

Wahlquist, Y., Carlson, F.B., and Soltesz, K. (2023). Fast simulation of pharmacokinetics. *IFAC-PapersOnLine*, 56(2), 2995–3000. doi:10.1016/j.ifacol.2023.10.1425.

White, D.B., Walawander, C.A., Liu, D.Y., and Grasela, T.H. (1992). Evaluation of hypothesis testing for comparing two populations using NONMEM analysis. *J Pharmacokinet Biopharm*, 20(3), 295–313. doi:10.1007/BF01062529.

White, M., Kenny, G.N.C., and Schraag, S. (2008). Use of target controlled infusion to derive age and gender covariates for propofol clearance. *Clin Pharmacokinet*, 47(2), 119–127. doi:10.2165/00003088-200847020-00005.

Wählby, U., Jonsson, E.N., and Karlsson, M.O. (2002). Comparison of stepwise covariate model building strategies in population pharmacokinetic-pharmacodynamic analysis. *AAPS PharmSci*, 4(4), 68–79. doi:10.1208/ps040427.