Pharmaceutical Sciences 2026;32(3):394-404
doi:10.34172/ps.43408
https://ps.tbzmed.ac.ir

Original Article

Reverse-Engineering Drug-Release Kinetics from Dissolution Profiles with Machine Learning

Sergio Sánchez-Herrero[1,2*], Joaquin Herrerias-Lopez-De-Heredia[2], Laura Calvet[3#], Angel A. Juan[4#]

[1]Dept. of Computer Science, Multimedia and Telecommunication, Universitat Oberta de Catalunya, 08018 Barcelona, Spain
[2]Simulation Department, Empresarios Agrupados Internacional S.A., 28015 Madrid, Spain
[3]Telecommunications and Systems Engineering Department, Universitat Autònoma de Barcelona, 08202 Sabadell, Spain
[4]Research Center on Production Management and Engineering, Universitat Politècnica de València, 03801 Alcoy, Spain

ARTICLE INFO

Article History:
Received: October 27, 2025
Revised: January 26, 2026
Accepted: February 17, 2026
ePublished: July 31, 2026

Keywords:
Machine learning, Pharmacokinetics, Drug release, Modelling and simulation, Dissolution

Abstract

Introduction: Dissolution is commonly modeled forward from formulation inputs. We propose an inverse-learning framework where machine learning (ML) infers mechanistically interpretable release-kinetic parameters from dissolution profiles and relates them to formulation and pharmacokinetic (PK) descriptors within Quality by Design (QbD).

Methods: An in silico dataset of release profiles was generated using six kinetic models (zero-order, first-order, Higuchi, Hixson–Crowell, Korsmeyer–Peppas and Weibull). Each profile was fitted to all candidate models to select the best mechanism and estimate its parameters. ML regressors (tree-based ensembles and artificial neural networks) were trained using full time series and engineered summaries (e.g., lag time, fractional release at 2/6/12 h and AUC0–24); PK features (e.g., Tmax, Cmax, CL, Vc) were incorporated when applicable. Performance was evaluated by cross-validation and external tests using R^2 and error metrics, and interpretability was assessed with SHAP.

Results: Gradient boosting performed best for simpler kinetics (zero/first/Higuchi/Hixson–Crowell; $R^2 \approx 0.99$; AFE/AAFE≈1.00–1.01). For complex kinetics, neural networks were best for Korsmeyer–Peppas ($R^2 = 0.89$), and boosting remained strong for Weibull. SHAP highlighted AUC0–24, Tmax and Cmax as dominant predictors. External experimental profiles showed good agreement by visual comparison.

Conclusion: Inverse learning can recover mechanistically meaningful release parameters from dissolution data and connect them to formulation and PK descriptors, supporting faster and more transparent modified-release design under QbD.

Introduction

Recent advances in modelling and simulation (M&S) strategies have profoundly influenced the drug development process. These innovations reflect an improved capacity to manage the increasing complexity of drug release profiles and to integrate large, heterogeneous datasets.[1] As a result, quantitative predictive models have emerged that enhance the efficiency of model-informed drug discovery and development (MID3). The growing importance of M&S approaches is evident in regulatory guidelines,[2] industry reports, and position papers from major conferences,[3] all underscoring their pivotal role in modern pharmaceutical science.

A key element of M&S in formulation design is the establishment of in vitro–in vivo correlation (IVIVC). IVIVC provides a predictive mathematical link between an in vitro property (such as dissolution) and an in vivo response (such as absorption or plasma concentration). Regulatory agencies particularly encourage IVIVC for modified-release (MR) dosage forms, where in vivo pharmacokinetic (PK) profiles should align with in vitro release data.[4] Such correlations facilitate prediction of clinical performance from dissolution data and support comparability assessments during formulation changes.

Controlled-release systems—including delayed-release tablets, microspheres, liposomes, nanoparticles, micelles, and transdermal patches—aim to achieve target exposure by modulating drug release over time.[5] Mathematical models are routinely applied to describe dissolution and release kinetics, supporting mechanism identification and rational design.[1] However, empirical equations are often preferred in practice due to process complexity; while practical, these models can limit mechanistic interpretability and complicate comparisons across formulations.[6-7]

Data-science approaches, particularly artificial intelligence and machine learning (AI/ML), offer flexible tools to capture non-linear kinetics, extract informative features from dissolution profiles, and construct predictive models for drug development tasks.[7] Beyond forward

***Corresponding Author:** Sergio Sánchez-Herrero, Email: ssanchezherre@uoc.edu
#Contributing Authors: Laura Calvet, Angel A. Juan

modelling (i.e., predicting dissolution from formulation inputs), there is increasing interest in inverse tasks: learning from dissolution profiles to infer mechanistically meaningful parameters. These include model-dependent kinetic constants (e.g., zero-, first-order, Higuchi, Hixson–Crowell, Korsmeyer–Peppas and Weibull). Such parameters can be mapped to formulation attributes and, when required, translated to in vivo exposure within a model-informed workflow. Bridging in silico predictions with experimental validation is therefore a powerful means of accelerating the development of long-acting drug delivery systems.[8-10]

Hybrid strategies combining PK modelling with machine learning have also been explored. While offering detailed mechanistic insights, they demand extensive physiological data that may not always be available. In contrast, the present work focuses on combining classical kinetic models with ML to ensure interpretability and broader applicability during early-stage formulation development.

Here, we propose a model-dependent methodology based on well-established kinetic functions (zero-order, first-order, Higuchi, Hixson–Crowell, Korsmeyer–Peppas, Weibull). We validate the approach using three complementary data sources: (i) simulated datasets designed to capture typical and extreme drug release behaviors; (ii) experimental datasets from published studies, enabling robust external validation; and (iii) virtual datasets generated by integrating release models into a two-compartment pharmacokinetic system, subsequently used to train ML algorithms. This integrated strategy has the potential to substantially reduce the time and cost of developing long-acting drug delivery systems by enabling efficient exploration of the parameter space and minimizing experimental trials. As a consequence, work introduces a hybrid, model-dependent machine learning strategy that bridges the gap between purely empirical kinetic equations and data-intensive "black-box" AI/ML methods, ensuring mechanistic interpretability while maintaining broad applicability during early-stage formulation development. Ultimately, simulation-supported workflows combined with experimental validation provide a reliable framework for the rational design of advanced drug delivery systems.

Materials and Methods

Software Description

The PhysPK® (version 2.6.0) bio-simulation toolkit was utilised in the development of the drug release mathematical models.[11] PhysPK® version 2.6.0 has been developed using EcosimPro version 7.0.1 software, which functions similarly to other object-oriented programming languages, such as C/C++, Python, and R. To facilitate seamless integration, EcosimPro includes a feature that allows Python and R scripts to be embedded directly within the software. PhysPK® is built upon advanced symbolic and numerical methods, enabling it to manage complex systems of differential-algebraic equations (DAEs) and transform them into consistent mathematical models.[12] In addition, PhysPK® is equipped with libraries for complex PK/PD modelling and physiological systems. This ensures adherence to conservation laws for chemical compounds across spatial regions.

In this study, Python, a cross-platform, open-source programming language, was employed for machine learning estimation analysis. Specifically, Python version 3.11.00 [GCC 11.4.0] was utilised in conjunction with its extensive libraries for data management, statistical computation, and graphical visualisation. The analysis made use of several Python libraries, including NumPy (version 1.25.2),[13] pandas (version 1.5.3),[14] Statsmodels (version 0.14.1),[14] Matplotlib (version 3.7.1),[15] Seaborn (version 0.13.1),[16] scikit-learn (version 1.2.2) and scipy.stats (version 1.14.1).[17]

Development of the Drug Release Mathematical Models

The workflow strategy employed in developing the mechanistic drug release characterisation processes incorporated in PhysPK® is depicted in Figure 1. This workflow validation ensures the reliability and applicability of the results, providing a robust framework for the development of advanced drug delivery systems. The mathematical model implementation includes drug release models and parameter retrieval as key components in drug delivery systems, an internal validation, external validation and finally the ML estimation, where various ML algorithms were trained and evaluated using virtual data. This integrated approach aims to optimize release parameters and compound parameters based on pharmacokinetic and dissolution profiles, enabling the efficient exploration of a wide parameter space, reducing experimental costs, and accelerating the identification of optimal formulations. It is important to note that each mathematical release model incorporates unique considerations specific to its design.

Two output scales were used depending on the validation dataset. For internal validation, release profiles were analysed as the cumulative released concentration $C_{rel}(t)(mg{\cdot}L^{-1})$, i.e., not normalized. For external validation, release profiles were taken from the original literature sources and reported as percent release; for model computations we used the corresponding dimensionless fraction $F(t) = \%/100$ (figures are shown in % for readability). Consequently, the units of each rate constant follow directly from the model formulation and the dependent variable: concentration-based fits yield constants that include concentration terms (e.g., $mg{\cdot}L^{-1}{\cdot}h^{-1}$), whereas fraction-based fits yield time-based units (e.g., h^{-1}, $h^{-1/2}$, h^{-n}). The units reported in Tables 1–2 are rated constants compared only within the same data scale unless explicit normalization is performed. This choice was maintained because several external studies do not provide the information required to reliably convert % release into absolute concentration.

Zero-Order Kinetics

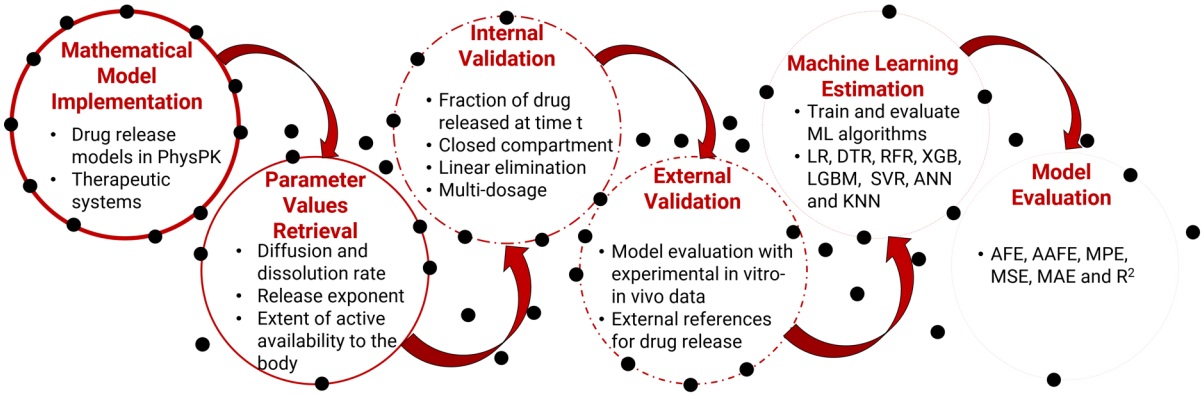

Figure 1. An overview of the strategy for developing and validating drug delivery systems in PhysPK®

Table 1. Drug- and system delivery related parameters for internal validation.

Drug Release	Parameter	Release Constant (K)		Dosage		Multi-Dosage	
		1.1.	1.2.	2.1.	2.2.	3.1.	3.2.
Zero	K_0 (mg·L⁻¹·h⁻¹)	[1e-10, 0.1, 1, 10 , 1e10][1]	[2] , Ke=1 l/h	1, 10 mg[1]	[3]	[1,6]	[7]
First	K_1 (mg·h⁻¹)	[1e-10, 0.1, 1, 10 , 1e10][1]	[2] , Ke=1 l/h	1, 10 mg[1]	[3]	[1,6]	[7]
Higuchi	K_H (mg·L⁻¹·h^(-1/2))	[1e-10, 0.1, 1, 10 , 1e10][1]	[2] , Ke=1 l/h	1, 10 mg[1]	[3]	[1,6]	[7]
Hixson-Crowell	K_B ((mg·L⁻¹)^(1/3)·h⁻¹)	[1e-10, 0.1, 1, 10 , 1e10][1]	[2] , Ke=1 l/h	1, 10 mg[1]	[3]	[1,6]	[7]
Korsmeyer-Peppas	K (mGh⁻ⁿ)	10	[4] , Ke=1 l/h	1, 10 mg[1]	[3]	[1,6]	[7]
	n (-)	[0.25, 0.5, 0.75, 1 , 1.25][1]					
Weibull	b (-)	[0.5, 1, 1.5][1]	[5] , Ke=1 l/h	1, 10 mg[1]	[3]	[1,6]	[7]

[1] Scenarios 1.1., 2.1 and 3.1. Ke=0 (l/h), [2] Same K values than scenario 1.1., [3] Same doses values than scenario 2.1. and Ke=1 (l/h), [4] Same K and n values than scenario 1.1., [5] Same b values than scenario 1.1., [6] Scenario 2.1. with administration in mg at time 0 and 12 hours, [7] Scenario 2.2. with administration in mg at time 0 and 12 hours.

Table 2. Common values and variability of model parameters for the simulated data.

Zero	First	Higuchi	Hixson-Crowell	Korsmeyer-Peppas K (h⁻ⁿ)	n (-)	Weibull a (-)	Weibull b (-)	t (h)	Reference
K_0 (%h⁻¹)	K_1 (%h⁻¹)	K_H (%h^-1/2)	K_B (% (1/3)·h⁻¹)	K (%h⁻ⁿ)	n (-)	a (-)	b (-)	t (h)	
8.716	0.015	21.706	0.038	11.490	0.855	7.679	1.067	0.465	[26]
[4.1;5.98;10.92;25.35]	*	*	*	*	*	*	*	*	[27]
*	[0.516;4.248;3.006;2.046]	*	*	*	*	*	*	*	[28]
*	*	81.5	0.61	80.9	0.43	*	*	*	[29]
*	*	*	*	0.57	0.504	1.057	0.753	0	[30]

* Not model data in the reference.

The drug release process under zero-order kinetics is independent of the drug concentration within the delivery system. Typically, the active agent is administered over a short period, which can create challenges in maintaining effective drug concentrations and may impact patient adherence and dosing accuracy. In contrast, therapeutic delivery systems are designed to release the active agent at a constant rate over a predetermined period, helping to mitigate concentration fluctuations. This release rate is directly proportional to the amount of drug delivered at each interval.[18]

First-Order Release Kinetics

First-order release kinetics are characterized by the relationship between changes in drug concentration over time and the current concentration, as mathematically described by Siepmann et al.[19] This kinetic model is commonly observed in various therapeutic systems. For water-soluble active agents embedded within a porous matrix, the amount of drug released is directly proportional to the remaining amount of drug in the matrix. As a result, the amount of active agent released decreases progressively over time.

Higuchi Model

The Higuchi model was the first mathematical approach developed to describe drug release from matrix systems.[20] It has been found to be applicable for studying the release of both water-soluble and poorly soluble drugs when embedded in semisolid and solid matrices. Experimental data typically plot the cumulative percentage of drug release versus the square root of time, with the slope of this plot representing the Higuchi dissolution constant, K_H.

Hixson-Crowell Cube Root Law

The Hixson-Crowell cube root law provides a theoretical

framework for the drug release process from systems characterized by a reduction in surface area and particle or tablet diameter.[21] In the absence of shape change, as a suspended solid dissolves, the surface area decreases as the two-thirds power of its weight. This relationship was utilized by Hixson and Crowell in deriving the cube root law.

Korsmeyer - Peppas Model

Korsmeyer et al. derived a simple relationship to describe drug release from a polymeric system.[22] They developed an empirical equation to analyze Fickian and non-Fickian drug release from swelling and non-swelling polymeric delivery systems. The model was specifically designed for drug release from a polymeric matrix, such as a hydrogel.

Weibull Model

This model is a distribution function that effectively describes phenomena and processes occurring over a finite time period. It is based on a function originally proposed by Weibull to describe drug release curves.[23] The Weibull equation is commonly applied to analyse drug dissolution and release from dosage forms. Notably, this model is considered a more effective tool for comparing drug release profiles in matrix systems.

Internal Validation

Drug release models were internally validated through a simulation-based analysis, where a set of several theoretical scenarios, shown in Table 1, were considered and accounted for drug release parameters. The release constant (K) examines drug release under two conditions, without elimination (1.1) and with linear elimination (1.2), representing different pharmacokinetic environments. The dosage evaluates the release profiles for single-dose administration under conditions of no elimination (2.1) and linear elimination (2.2), providing insight into how elimination affects drug bioavailability and finally, multi-dosage: assesses drug release for multiple doses under both no elimination (3.1) and linear elimination (3.2) conditions, simulating repeated dosing scenarios in clinical settings.

Input parameters (drug- and system-related) for each scenario are reported in Table 1. The drug-related parameters do not correspond to any specific drug. The system was modelled as a finite-volume compartment without enforced sink boundary combined with either linear elimination or no elimination. This choice was made to preserve mass balance and to explore both sink-like and non-sink limits of dissolution behaviour, including extreme "stress-test" scenarios (e.g., no elimination) that are not intended to represent typical in vivo conditions. Only the release process and its theoretical behaviour were considered,[24] excluding solubility-related phases, which were addressed in Cuquerella-Gilabert et al.[25].Because some classical kinetic models (e.g., Higuchi and Hixson–Crowell) are derived under sink assumptions,[20,21] when applied in non-sink configurations their fitted constants should be interpreted as apparent/empirical (lumped) parameters rather than strictly mechanistic constants.[24] Although some scenarios may seem unrealistic, they were included to simulate extreme conditions. The release constants were chosen to cover a wide range of release rates, ensuring robustness across both slow and rapid drug release profiles. A linear elimination rate constant of 1 L/h was applied to progressively assess the influence of drug elimination on release profiles (rather than to mimic typical human pharmacokinetics). Lastly, drug doses of 1 mg and 10 mg were selected to illustrate how different concentrations influences the release and elimination processes while maintaining visual clarity in the figures. Therefore, when elimination is present, removal from the central compartment can maintain low dissolved concentrations, yielding sink-like behaviour when elimination is fast relative to release.

External Validation

The external evaluation of drug release profile models involves the utilisation of an independent dataset for the purpose of assessing the accuracy and bias of the overall model performance in subjects who exhibit characteristics similar to those of the subjects with whom the models were developed. This methodology is also useful in evaluating and selecting the most accurate and precise model for a different target population. External dissolution profiles extracted from the literature were digitized and overlaid with simulated curves for visual comparison; no additional kinetic fitting or parameter estimation was performed. Consequently, external evaluation is an appropriate approach for selecting ML models available for model-informed precision dosing. Table 2 shows a relation between references, drug release models and inputs for external validation.

The data from these references were extracted using the Plot Digitizer software (Version v3).[31] This is a free data-extraction program that invokes the external tool AutoTrace for automatic curve detection.

Machine Learning Estimation

In this study, an intravenous two-compartment model was used to simulate the controlled release of a drug-like polymeric nanoparticle. The drug is gradually released from the nanoparticles over time, and its immediate availability in the bloodstream upon intravenous administration enables rapid distribution to the central compartment, followed by distribution to peripheral tissues. The linear elimination of the drug from the central compartment was a key consideration. The two-compartment intravenous pharmacokinetic model is described by Sanchez et al.[11] This approach facilitates pharmacokinetic modelling because it eliminates the necessity to account for absorption and first-pass metabolism, which are inherent in the process of oral administration. This approach offers a more direct representation of the controlled release process. Furthermore, these absorption processes are given significant attention in the analysis of Cuquerella

et al. (2024).[25]

The research flow chart, illustrated in Figure 2, is described below for model-informed drug discovery and development (MID3) ML estimation. The process begins with the generation of a virtual population (N = 100,000) using pharmacokinetic (PK) data. We adopted an 80% derivation / 20% validation split as a pragmatic and widely used hold-out strategy in machine-learning workflows, balancing the need for a large development set with a sufficiently large independent cohort for final performance assessment. Given the simulated population size (N = 100,000), reserving 20% for validation provides stable out-of-sample estimates while preserving statistical power for model development. This 80/20 split was chosen as a pragmatic and widely used compromise that preserves sufficient data for model development while retaining a large external subset for unbiased assessment of generalization. Within the derivation cohort, data were further split into training (80% of derivation) and test (20% of derivation), corresponding to 64% and 16% of the full dataset, respectively. A stratified split was used to preserve key-variable distributions. To ensure a uniform distribution of key variables across subsets, we performed stratified random sampling using a fixed random seed. Strata were defined by the cross-product of the main design factors (scenario × kinetic model × dose level). Within each stratum, observations were randomly assigned to derivation vs validation (80/20) and, within the derivation cohort, to training vs test (80/20), thereby preserving the marginal distributions of these categorical factors across subsets. We additionally assessed balance of key continuous covariates (dose, AUC0–24, Tmax, Cmax, half-life, Vd, and CL) by comparing summary statistics between subsets and applying standard two-sample comparisons (e.g., t-tests; optionally distributional checks such as Kolmogorov–Smirnov tests).. The derivation cohort is used to train machine learning (ML) models (including LR: Linear Regression; ANN: Artificial Neural Networks; RFR: Random Forest Regressor; DTR: Decision Tree Regressor; LGBM: LGBM Regressor; XGB: XGB Regressor; KNN: K Neighbors Regressor and SVR: Support Vector Regression)[32] based on key features such as drug release parameters, PK profiles, and metrics like AUC_{0-24}, T_{max}, C_{max}, half-life (HL), volume of distribution (V_d), and clearance (CL). The validation cohort is used to test the models and assess prediction accuracy. Performance measures, described before, are calculated, and the best-performing model is selected. The process includes 100 rounds of resampling for both training and validation to ensure robustness. This workflow facilitates a data-driven approach to drug discovery and development with ML.

The common values and variability of the model parameters are provided in Table 3 for a range of random doses from 10 mg to 100 mg. The model is described by log-normal distributions and an additive plus proportional error model. Concentrations are simulated for each subject at 1, 2, 3, 4, 6, 8, 10, 12, 16, 20 and 24 hours.

The predictions of two-compartment concentrations via Monte Carlo simulation are utilised for the training and testing of ML methods. A comprehensive consideration of other pharmacokinetics variables was undertaken, encompassing pharmacokinetic variables like the administered dose (Dose), clearance (CL), distribution, central volume (V_C), peripheral volume (V_p), release constant (K), maximum concentration (C_{max}), maximum time concentration (T_{max}), half-life (H1/2), and area under the curve at 24 hours AUC_{0-24}, as these variables are critical for understanding the drug's release, distribution, and elimination dynamics in the body.[33] Furthermore, the predictions of two-compartment parameters are evaluated and compared. The target variable was the release constant related to each kinetic model, with the objective being to identify the optimum regimens for different drug release scenarios.

Table 3. Common values and variability of model parameters for the simulated data.

	Simulated Data Two-compartment model
Dose	[10 mg - 100 mg]
Population	100 000 subjects
Population Parameters	V1 = 40 l, V2 = 40 l, K10 = 5 l/h and K12 = 10 l/h
Intersubject variability	20% on Constant Release, K10 , V1
Residual error	Var = (0.01 + 0.1 · IPRED)^2
Drug Release Parameters	Constant Release
Zero-order	20 (40% C.V.)
First-order	0.5 (40% C.V.)
Higuchi	20 (40% C.V.)
Hixson-Crowell	0.5 (40% C.V.)
Korsmeyer-Peppas	K = 20 , n = 0.75 (40% C.V.)
Weibull	b = 0.75, a = 5 , t = 0 (40% C.V.)

* C.V.: coefficient of variation

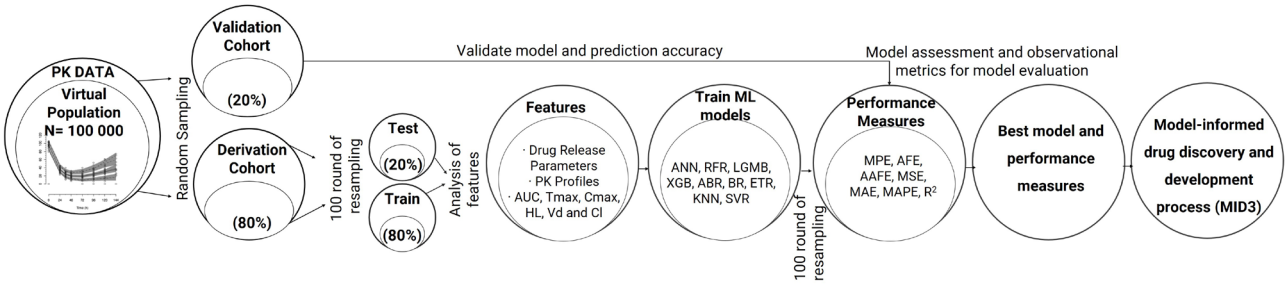

Figure 2. A flow chart outlining the steps followed in our research for model-informed drug discovery and development (MID3) with ML approach is presented

It is imperative to acknowledge that ML models are contingent upon pivotal parameters that cannot be directly estimated from the data. Consequently, hyperparameter tuning assumes paramount importance in optimising algorithm performance. These parameters, often referred to as 'tuning parameters', are not amenable to analytical determination of their optimal values. To address this, ML models were optimized through systematic hyperparameter tuning to enhance their performance, ensure better generalization, and achieve accurate predictions tailored to the specific characteristics of the dataset in Table 4.

As all datasets were synthetically generated, no missing values were present; residual variability was added only according to the combined additive-plus-proportional error model described in Table 3. Prior to model fitting, continuous predictors were standardized (zero mean, unit variance) for algorithms sensitive to feature scale (LR, ANN, SVR, and KNN). To mitigate overfitting and quantify robustness against sampling variability, we performed 100 resampling iterations using pandas.DataFrame.sample with a fixed pseudorandom seed to ensure reproducibility across all methods.[34] Model complexity and regularization were controlled

Table 4. Hyperparameter search spaces (distributions and bounds) and selected values.

Model*	Hyperparameter	Distribution	Range	Value
LR	fit_intercept	categorical (uniform)	{True, False}	True
	positive	categorical (uniform)	{True, False}	True
ANN	hidden_layer_sizes	categorical (uniform)	{(50,), (100,), (100,50), (200,100), (100,100,50)}	(100,50)
	activation	categorical (uniform)	{relu, tanh}	relu
	alpha	log-uniform	[1e-6, 1e-2]	1e-4
	learning_rate_init	log-uniform	[1e-4, 1e-2]	0.001
	batch_size	categorical (uniform)	{auto, 16, 32, 64}	auto
	learning_rate	categorical (uniform)	{constant, adaptive}	adaptive
DTR	max_depth	discrete uniform	integers in [2, 20]	10
	min_samples_split	discrete uniform	integers in [2, 30]	10
	min_samples_leaf	discrete uniform	integers in [1, 10]	5
	max_features	categorical (uniform)	{None, sqrt, log2}	None
RFR	n_estimators	discrete uniform	integers in [50, 500]	100
	max_depth	discrete uniform	integers in [2, 20]	10
	min_samples_split	discrete uniform	integers in [2, 30]	10
	min_samples_leaf	discrete uniform	integers in [1, 10]	5
	max_features	categorical (uniform)	{sqrt, log2, 1.0}	1.0
	bootstrap	categorical (uniform)	{True, False}	True
LGBM	learning_rate	log-uniform	[0.01, 0.3]	0.1
	n_estimators	discrete uniform	integers in [50, 500]	100
	max_depth	discrete uniform	integers in [3, 15]	10
	num_leaves	discrete uniform	integers in [15, 255]	31
	subsample	uniform	[0.6, 1.0]	0.8
	colsample_bytree	uniform	[0.6, 1.0]	0.8
	min_child_samples	discrete uniform	integers in [5, 50]	20
XGB	learning_rate	log-uniform	[0.01, 0.3]	0.1
	n_estimators	discrete uniform	integers in [50, 500]	100
	max_depth	discrete uniform	integers in [3, 15]	10
	subsample	uniform	[0.6, 1.0]	0.8
	colsample_bytree	uniform	[0.6, 1.0]	0.8
KNN	n_neighbors	discrete uniform	integers in [1, 30]	5
	weights	categorical (uniform)	{uniform, distance}	uniform
	p	categorical (uniform)	{1, 2}	2
SVR	kernel	categorical (uniform)	{rbf, linear}	rbf
	C	log-uniform	[1e-2, 1e3]	1.0
	epsilon	log-uniform	[1e-3, 1]	0.1

**LR: Linear Regression; ANN: Artificial Neural Networks; RFR: Random Forest Regressor; DTR: Decision Tree Regressor; LGBM: LGBM Regressor; XGB: XGB Regressor; KNN: K Neighbors Regressor and SVR: Support Vector Regression*

through algorithm-specific hyperparameters (e.g., n_neighbors, max_depth, min_samples_split; see Table 4). Hyperparameters were tuned with RandomizedSearchCV (5-fold cross-validation). Continuous positive scale parameters were sampled from log-uniform distributions; parameters bounded in [0,1] were sampled from uniform distributions; integer-valued hyperparameters were sampled from discrete uniform distributions over the specified ranges; and categorical options were sampled uniformly from the candidate sets. Selected values correspond to the best mean cross-validation performance; all other parameters were kept fixed at defaults or as specified. The test subset was kept strictly held out during tuning to prevent information leakage. After identifying the best hyperparameter configuration, each model was refit on the full training subset and evaluated once on the held-out test subset; final generalization performance was then assessed on the independent validation cohort. For RandomizedSearchCV, we tuned the main complexity/regularization hyperparameters per algorithm: ANN (hidden_layer_sizes, alpha, learning_rate_init), DTR/RFR (max_depth, min_samples_split, min_samples_leaf, plus n_estimators for RFR), LGBM/XGB (n_estimators, learning_rate, max_depth, subsample, colsample_bytree, plus num_leaves/min_child_samples for LGBM), KNN (n_neighbors, weights, p), and SVR (C, epsilon, RBF kernel). Remaining parameters were kept at library defaults. The final selected hyperparameters are reported in Table 4.

All drug release models were run maintaining consistent experimental ML settings. The dataset distribution ensured uniform data distribution across all models. Identical pre-processing steps were applied to the data to eliminate any biases introduced by variations in data preparation. Additionally, the same validation techniques were employed to assess model performance using consistent metrics. By standardizing these conditions, the evaluation focuses solely on the intrinsic differences between models, ensuring a fair and reliable comparison.

The primary objective is to explain the prediction of the target variable by quantifying the contribution of each feature to that prediction. A t-test was applied to show homogeneity between training, test and validation datasets.[35]

The generation of a residual data was undertaken in order to assess the fit of the models and to examine any patterns in the residuals that could indicate model misfit or violations of assumptions. This information is essential for evaluating the homoscedasticity and normality of residuals, which are key assumptions in regression analyses.[36]

Finally, the importance of individual features to the model's outcome was quantified using the SHAP (SHapley Additive exPlanations) package. SHAP values, grounded in cooperative game theory, enhance the transparency and interpretability of machine learning models by attributing the contribution of each feature to the model's output. This approach conceptualizes each feature as a 'player' in a cooperative game, thereby clarifying its unique contribution to the prediction for each observation or instance.[37]

Model Evaluation

The prediction performance of ML models were calculated using the percentage prediction error (PE), and mean percentage prediction error (MPE). The overall predictability of the model is evaluated in terms of bias and precision using the conventional metrics of average-fold error (AFE) and absolute average-fold error (AAFE). If the AFE and AAFE values are between 0.8 and 1.25-fold, then the predictive performance of the model is considered to be reasonably satisfactory. In addition to the aforementioned metrics, the following traditional ones were implemented as well mean squared error (MSE), root mean squared error (RMSE) and mean absolute error (MAE). For ease of interpretation, the standard deviation (SD) of the observed target variable in the evaluation set is also reported alongside RMSE (both in the same scale), enabling direct comparison between the typical prediction error and the natural variability of the data. It were also calculated, R^2 score and the adjusted coefficient of determination (adjusted R^2) to account for the number of predictors used by each model.[38]

Results

Development of the Drug Release Mathematical Models

The graphical representation of the drug release models in PhysPK® can be found from Figure S1 to Figure S6 in supplementary material and they are explained in the next subsection. These results validate the mathematical model in drug release behaviour across different dosing and elimination scenarios, emphasizing its applicability in pharmacokinetic modelling.

Internal Validation

Drug release processes were successfully implemented in PhysPK®, as confirmed by the internal validation against theoretical scenarios (Figures S1–S6 and Table 1). Across all tested conditions, the simulated profiles reproduced the expected kinetic behaviours for Zero-order, First-order, Higuchi, Hixson–Crowell, Korsmeyer–Peppas, and Weibull models.

In general, variations of release constants (K), doses, and elimination processes yielded concentration–time curves consistent with theoretical predictions: (i) linear or exponential release in the absence of clearance, (ii) dose-proportional increases without elimination, and (iii) dynamic equilibria between accumulation and clearance under repeated dosing. These patterns confirmed the correct mathematical implementation of each model.

Minor numerical deviations observed at peak points (particularly for Hixson–Crowell) were attributed to integration artefacts and did not affect overall validity. Taken together, the results demonstrate that PhysPK® reliably reproduces classical release kinetics, supporting its use for further external validation and advanced

applications.

External Validation

An external validation of drug release models was conducted using five references from scientific papers exposed in Table 2. The results indicate remarkable consistency across the evaluated models, suggesting a high level of accuracy and reliability. The MPE values were below 2.5%, reflecting excellent predictive performance with minimal deviations from the experimental data. Furthermore, the AFE AAFE metrics exhibited average values of 1.00 ± 0.02, suggesting an almost perfect correlation between predicted and observed values, with no material differences observed". The external validation confirms the model's accuracy in predicting the release under specific conditions.

Furthermore, a close visual overlap was observed between the prediction curves of the models and the experimental curves reported in the literature, thereby further reinforcing the coherence of the results shown from Figure S7 to Figure S11.

These findings underscore the robustness of the selected models, thereby demonstrating the reliability of PhysPK® as a tool for describing and predicting drug release processes under various experimental conditions.

Machine Learning

The characteristics of the virtual datasets generated with the two-compartment model are summarized in Table S1-S3 in supplementary material. Three disjoint sets were used for ML analysis—training (64%), test (16%), and validation (20%)—using a fixed random seed and stratified sampling to preserve the distribution of key inputs (dose, AUC0–24, Tmax, Cmax, HL, Vd, and CL) across subsets. As a consequence, the final data partition corresponded to 64% training, 16% test, and 20% validation, generated via stratified sampling to preserve the distributions of scenario/kinetic model/dose level; key continuous covariates showed comparable distributions across subsets.

Table S4 compares ML performance across kinetic models. Gradient-boosting methods (XGBoost, LightGBM) achieve near-perfect fits in classical settings (Zero, First, Higuchi, Hixson–Crowell), with R^2 approx. 0.99, minimal MSE/MAE, and AFE/AAFE approx. 1.00–1.01. In the more complex KP profiles, artificial neural networks (ANN) attain the best accuracy ($R^2 = 0.89$), whereas boosting and Random Forests show a noticeable drop (R^2 approx. 0.63–0.64 and 0.56, respectively). Decision Trees and k-Nearest Neighbors perform competitively in simple kinetics but degrade in K_p and, to a lesser extent, Weibull. Linear Regression underperforms overall (typically $R2 < 0.5$). Support Vector Regression (SVR) yielded unstable solutions and did not converge satisfactorily; therefore, it was excluded from subsequent comparative summaries. Table S4 reports R^2 and adjusted R^2 side-by-side, and RMSE alongside SD(target), to contextualize error relative to target variability. Adjusted R^2 closely tracks R^2, indicating limited inflation of fit due to model complexity

SHAP analyses (Table S5 consistently identify $AUC_{0\text{-}24}$, T_{max}, and C_{max} as leading contributors across models, with additional importance for CL, V_c, and Dose depending on the mechanism. (Only the top contributors are displayed; minor effects are truncated to 0 at this precision.) This pattern aligns with the pharmacological relevance of exposure, timing, and peak metrics in shaping release/PK behaviour.

Residual ranges (Table S6) corroborate these trends: boosting and ANN models exhibit the narrowest residual bands in classical kinetics, with broader—but still controlled—dispersion in KP for all methods. Overall, the combination of mechanistic release models with modern ML (particularly boosting and ANN) yields accurate, stable predictions across a range of kinetic behaviours.

Note: All errors are computed on log K. Because the target variance differs by kinetic model, MSE values are not directly comparable across models; AFE/AAFE values near 1.0 indicate minimal bias.

Discussion

In comparison with hybrid PK–ML strategies, the inverse learning framework presented here offers a simpler and more interpretable solution, as it requires only dissolution data as input. This makes it particularly attractive for early-stage formulation screening, while PK integration remains a promising direction for future work.

Mathematical models have long served as essential tools for characterizing drug release kinetics. Classical approaches such as zero-order, first-order, Higuchi, Hixson–Crowell, Korsmeyer–Peppas, and Weibull remain widely used to predict release behaviour in systems ranging from transdermal patches and matrix tablets to osmotic pumps.[39] Their ability to simulate release mechanisms reduces experimental burden and accelerates formulation development. Our internal validation with PhysPK® confirmed that these models reproduce expected behaviours across different dosing and elimination scenarios, while external validation showed close alignment between simulations and published dissolution data. In the latter case, experimental profiles were digitized and overlaid with simulations for visual comparison; no additional parameter fitting was performed.

A limitation of this work is that perfect sink conditions were not imposed in the release module and some scenarios intentionally explored non-sink extremes (e.g., no elimination); therefore, for sink-derived models the fitted rate constants should be interpreted as apparent parameters and extrapolated to in vivo settings with caution. Accordingly, sink-derived model constants are used here primarily as descriptive features within the modelling/ML workflow rather than as direct mechanistic surrogates of in vivo release.

Machine learning substantially enhanced predictive performance beyond mechanistic models alone. Gradient boosting methods (XGBoost, LightGBM) consistently

achieved low residual errors and near-perfect R^2 values across most release scenarios. Artificial neural networks (ANN) achieved the best fit for Korsmeyer–Peppas, reflecting their ability to capture non-linear dynamics, though with some risk of overfitting.

Linear Regression showed poor predictive accuracy because the mapping between PK descriptors and the release parameters is strongly non-linear and involves interactions that cannot be captured by an additive linear model. In addition, SVR performed poorly and showed instability at this dataset scale; despite testing alternative kernels and regularization settings, performance remained low, likely due to the combination of non-linear interactions and the sensitivity of SVR to hyperparameter choices and computational scaling in large datasets. Ensemble approaches such as Random Forest also performed robustly, while individual Decision Trees were more sensitive to local variations. Linear Regression was adequate for simple kinetics (zero-order, Hixson–Crowell) but inadequate for non-linear profiles. In contrast, Support Vector Regression (SVR) failed to converge reliably, suggesting that kernel-based methods may be less suitable for high-dimensional pharmacokinetic datasets. These results support the use of flexible non-linear learners (tree-based ensembles and ANN), which are better suited to capture complex relationships in this problem setting.

Residual and SHAP analyses confirmed the reliability and interpretability of the best-performing models. Pharmacokinetic variables such as AUC_{0-24}, Cmax, and clearance emerged as the most influential predictors, consistent with their established role in drug exposure and elimination.

Beyond methodological contributions, the integration of mechanistic release models with ML offers tangible benefits for pharmaceutical development. Within a Quality by Design (QbD) framework, the proposed strategy can reduce experimental burden by efficiently exploring design spaces, optimizing critical variables such as polymer ratios or release times. The approach can also facilitate comparability studies for modified-release formulations, potentially reducing the need for additional clinical trials during scale-up or reformulation. Furthermore, during preclinical development, the framework can help prioritize prototype formulations with the highest likelihood of meeting target release profiles, thereby reducing costs and accelerating development timelines.[8,9,40]

This work has several important limitations that must be acknowledged. First, most analyses relied on synthetic datasets. While this approach allowed systematic exploration of a wide parameter space, it inevitably lacks the biological variability and experimental noise that characterize real dissolution and pharmacokinetic studies. As a result, the predictive performance reported here may overestimate that observed in practice. Future work will therefore expand external validation using controlled in vitro dissolution experiments and, ultimately, in vivo clinical datasets to confirm translational relevance. Second, the present framework was evaluated within a predefined design space. More complex scenarios, such as multiphasic or pH-dependent kinetics, remain to be explored. Finally, hyperparameter optimization was applied uniformly across all ML models; model-specific tuning strategies could further enhance performance.

Despite these limitations, the proposed framework offers practical advantages for formulation development. By enabling rapid exploration of formulation parameters, it supports Quality by Design practices and facilitates regulatory comparability assessments.[4] For example, in modified-release tablets, the framework could guide optimization of polymer ratios to achieve desired dissolution times. Importantly, the recovery of mechanistically interpretable parameters enhances transparency and facilitates communication with regulatory authorities. Overall, these findings illustrate how coupling mechanistic and data-driven methods can provide both predictive accuracy and interpretability, thereby bridging the gap between in silico design and experimental validation, and ultimately accelerating the development of more precise, patient-specific therapies.

Conclusion

The integration of mathematical modelling, simulation, and machine learning (ML) provides a powerful strategy to improve the precision and efficiency of drug release optimization. Using PhysPK®, we validated classical models (Zero-order, First-order, Higuchi, Hixson–Crowell, Korsmeyer–Peppas, Weibull), confirming their robustness across diverse release conditions.

Incorporating ML substantially enhanced predictive performance. Gradient boosting methods (XGBoost, LightGBM) achieved near-perfect accuracy for linear and exponential kinetics, while artificial neural networks provided the best results for non-linear Korsmeyer–Peppas profiles. SHAP analysis identified AUC_{0-24}, Cmax, and clearance as the most influential predictors, ensuring interpretability and regulatory relevance. External validation further confirmed high predictive accuracy, with a mean percentage error below 2.5% and AAFE between 0.8 and 1.25.

This framework has clear practical applications:

- Rapid optimization of early-stage formulations.
- Comparability assessments for modified-release products.
- Support for regulatory decision-making under Quality by Design principles.

Future work will expand validation using systematic in vitro and clinical datasets, as well as explore complex release scenarios (e.g., multiphasic or pH-dependent kinetics). Overall, the proposed strategy bridges mechanistic modelling and data-driven learning, accelerating the development of more efficient and clinically relevant drug delivery systems.

Authors' Contribution

Conceptualization: Sergio Sánchez-Herrero, Joaquin Herrerias-Lopez-De-Heredia

Data curation: Laura Calvet.
Formal analysis: Sergio Sánchez-Herrero, Laura Calvet.
Investigation: Sergio Sánchez-Herrero.
Methodology: Sergio Sánchez-Herrero.
Project administration: Angel A. Juan.
Resources: Sergio Sánchez-Herrero, Joaquin Herrerias-Lopez-De-Heredia.
Software: Sergio Sánchez-Herrero;
Supervision: Laura Calvet and Angel A. Juan.
Validation: Sergio Sánchez-Herrero, Joaquin Herrerias-Lopez-De-Heredia.
Visualization:Laura Calvet, Angel A. Juan.
Writing–original draft: Sergio Sánchez-Herrero
Writing–review & editing: Laura Calvet, Angel A. Juan.

Competing Interests
The authors declare no conflict of interest.

Ethical Approval
Not applicable.

Funding
This research received no external funding.

Supplementary File
Supplementary file contains Figure S1-S6.

References
1. Peppas NA, Narasimhan B. Mathematical models in drug delivery: how modeling has shaped the way we design new drug delivery systems. J Control Release 2014;190:75-81. doi:10.1016/j.jconrel.2014.06.041
2. Branch SK. Guidelines from the International Conference on Harmonisation (ICH). J Pharm Biomed Anal 2005;38(5):798-805. doi:10.1016/j.jpba.2005.02.037
3. Aarons L, Karlsson MO, Mentré F, Rombout F, Steimer JL, van Peer A. Role of modelling and simulation in phase I drug development. Eur J Pharm Sci 2001;13(2):115-22. doi:10.1016/s0928-0987(01)00096-3
4. Wang YL, Chang YT, Yang SY, Chang YW, Kuan MH, Tu CL, et al. Approval of modified-release products by FDA without clinical efficacy/safety studies: a retrospective survey from 2008 to 2017. Regul Toxicol Pharmacol 2019;103:174-80. doi:10.1016/j.yrtph.2019.01.037
5. Li W, Wu J, Zhang J, Wang J, Xiang D, Luo S, et al. Puerarin-loaded PEG-PE micelles with enhanced anti-apoptotic effect and better pharmacokinetic profile. Drug Deliv 2018;25(1):827-37. doi:10.1080/10717544.2018.1455763
6. Lawless JF. Statistical Models and Methods for Lifetime Data. John Wiley & Sons; 2011. doi:10.1002/9781118033005
7. Wang Y, Li D, Lv Z, Feng B, Li T, Weng X. Efficacy and safety of Gutong Patch compared with NSAIDs for knee osteoarthritis: a real-world multicenter, prospective cohort study in China. Pharmacol Res 2023;197:106954. doi:10.1016/j.phrs.2023.106954
8. Gormley AJ. Machine learning in drug delivery. J Control Release 2024;373:23-30. doi:10.1016/j.jconrel.2024.06.045
9. Pillai N, Abos A, Teutonico D, Mavroudis PD. Machine learning framework to predict pharmacokinetic profile of small molecule drugs based on chemical structure. Clin Transl Sci 2024;17(5):e13824. doi:10.1111/cts.13824
10. Gruber A, Führer F, Menz S, Diedam H, Göller AH, Schneckener S. Prediction of human pharmacokinetics from chemical structure: combining mechanistic modeling with machine learning. J Pharm Sci 2023. doi:10.1016/j.xphs.2023.10.035
11. Sánchez-Herrero S, Martínez FC, Serna J, Cuquerella-Gilabert M, Rueda-Ferreiro A, Juan AA, Calvet L. Embedding R inside the PhysPK bio-simulation software for pharmacokinetics population analysis. BIO Integr 2023;4(3):97-113. doi:10.15212/bioi-2023-0008
12. Reig-Lopez J, Merino-Sanjuan M, Mangas-Sanjuan V, Prado-Velasco M. A multilevel object-oriented modelling methodology for physiologically-based pharmacokinetics (PBPK): Evaluation with a semi-mechanistic pharmacokinetic model. Comput Methods Programs Biomed 2020;189:105322. doi:10.1016/j.cmpb.2020.105322
13. van der Walt S, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. Comput Sci Eng 2011;13(2):22-30. doi:10.1109/MCSE.2011.37
14. McKinney W. Data Structures for Statistical Computing in Python. SciPy; 2010. doi:10.25080/Majora-92bf1922-00a
15. Tosi S. Matplotlib for Python Developers. Birmingham, UK: Packt Publishing; 2009. doi:10.5555/1822435
16. Waskom ML. Seaborn: statistical data visualization. J Open Source Softw 2021;6(60):3021. doi:10.21105/joss.03021
17. Kramer O. Scikit-learn. In: Machine Learning for Evolution Strategies. Heidelberg: Springer; 2016. p. 45-53. doi:10.1007/978-3-319-33383-0_5
18. Talevi A, Ruiz ME. Zero-order drug release. In: The ADME Encyclopedia: A Comprehensive Guide on Biopharmacy and Pharmacokinetics. Cham: Springer International Publishing; 2021. p. 1-6. doi:10.1007/978-3-030-51519-5_33-1
19. Siepmann J, Siepmann F. Mathematical modeling of drug delivery. Int J Pharm 2008;364(2):328-43. doi:10.1016/j.ijpharm.2008.09.004
20. Higuchi T. Mechanism of sustained-action medication. Theoretical analysis of rate of release of solid drugs dispersed in solid matrices. J Pharm Sci 1963;52:1145-9. doi:10.1002/jps.2600521210
21. Hixson AW, Crowell JH. Dependence of reaction velocity upon surface and agitation. Ind Eng Chem 1931;23(8):923-31. doi:10.1021/ie50260a018
22. Korsmeyer RW, Gurny R, Doelker E, Buri P, Peppas NA. Mechanisms of solute release from porous hydrophilic polymers. Int J Pharm 1983;15(1):25-35. doi:10.1016/0378-5173(83)90064-9
23. Weibull W. A statistical distribution function of wide applicability. J Appl Mech 1951;18:293-7. doi:10.1115/1.4010337
24. Silva JS, Marques-da-Silva D, Lagoa R. Towards the development of delivery systems of bioactive compounds with eyes set on pharmacokinetics. In: Azar AT, ed. Modeling and Control of Drug Delivery Systems. Academic Press; 2021. p. 125-44. doi:10.1016/B978-0-12-821185-4.00006-3
25. Cuquerella-Gilabert M, Reig-López J, Serna J, Rueda-Ferreiro A, Merino-Sanjuan M, Mangas-Sanjuan V, et al. PhysDAT: a physiologically-based pharmacokinetic model for unraveling the dissolution, transit and absorption processes using PhysPK®. Comput Methods Programs Biomed 2024;243:107929. doi:10.1016/j.cmpb.2023.107929
26. Permanadewi I, Kumoro AC, Wardhani DH, Aryanti N. Modelling of controlled drug release in gastrointestinal tract simulation. J Phys Conf Ser 2019;1295(1):012063. doi:10.1088/1742-6596/1295/1/012063
27. Ekenna IC, Abali SO. Comparison of the use of kinetic model plots and DD solver software to evaluate the drug release from griseofulvin tablets. J Drug Deliv Ther 2022;12(2-S):5-13. doi:http://dx.doi.org/10.22270/jddt.v12i2-s.5402
28. Zhao YN, Xu X, Wen N, Song R, Meng Q, Guan Y, et al. A drug carrier for sustained zero-order release of peptide therapeutics. Sci Rep 2017;7(1):5524. doi:10.1038/s41598-017-05898-6
29. Mohammadian M, Jafarzadeh Kashi TS, Erfan M, Pashaei Soorbaghi F. In-vitro study of ketoprofen release from synthesized silica aerogels (as drug carriers) and evaluation of mathematical kinetic release models. Iran J Pharm Res 2018;17(3):818-29.

30. Yeop A, Sandanasamy J, Pang SF, Gimbun J. Stability and controlled release enhancement of *Labisia pumila*'s polyphenols. Food Biosci 2021;41:101025. doi:10.1016/j.fbio.2021.101025
31. Aydin O, Yassikaya MY. Validity and reliability analysis of the PlotDigitizer software program for data extraction from single-case graphs. Perspect Behav Sci 2022;45(1):239-57. doi:10.1007/s40614-021-00284-0
32. Hastie T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer; 2009. doi:10.1007/978-0-387-84858-7
33. Mangoni AA, Jackson SH. Age-related changes in pharmacokinetics and pharmacodynamics: basic principles and practical applications. Br J Clin Pharmacol 2004;57(1):6-14. doi:10.1046/j.1365-2125.2003.02007.x
34. Yu CH. Resampling methods: concepts, applications, and justification. Pract Assess Res Eval 2002;8(1):19. doi:10.7275/9cms-my97
35. Petrelli M. Introduction to Python in Earth Science Data Analysis: From Descriptive Statistics to Machine Learning. Springer Nature; 2021. doi:10.1007/978-3-030-78055-5
36. Montgomery DC, Peck EA, Vining GG. Introduction to Linear Regression Analysis. John Wiley & Sons; 2021.
37. Antwarg L, Miller RM, Shapira B, Rokach L. Explaining anomalies detected by autoencoders using Shapley Additive Explanations. Expert Syst Appl 2021;186:115736. doi:10.1016/j.eswa.2021.115736
38. Sánchez-Herrero S, Calvet L, Juan AA. Machine learning models for predicting personalized tacrolimus stable dosages in pediatric renal transplant patients. BioMedInformatics 2023;3(4):926-47. doi:10.3390/biomedinformatics3040057
39. Paarakh MP, Jose PA, Setty CM, Peterchristoper GV. Release kinetics–concepts and applications. Int J Pharm Res Technol 2018;8(1):12-20. doi:10.31838/ijprt/08.01.02
40. Bannigan P, Aldeghi M, Bao Z, Häse F, Aspuru-Guzik A, Allen C. Machine learning directed drug formulation development. Adv Drug Deliv Rev 2021;175:113806. doi:10.1016/j.addr.2021.05.016