KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
KUMASI
COLLEGE OF SCIENCE
DEPARTMENT OF MATHEMATICS

MACHINE LEARNING METHODS WITH APPLICATION IN PHARMACOKINETICS

By
EWUSI KEREN EFUA
SAFOA IRENE
GYAMFI BRIGHT OKYERE

A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, KWAME NKRUMAH UNIVERSITY OF SCIENCE AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF BACHELOR OF SCIENCE, MATHEMATICS

SEPTEMBER 2023

Declaration

We hereby declare that this dissertation is our own work under the able supervision of Dr. Reindorf Borkor, towards the award of the Bachelor of Science (BSc.) in Mathematics and that, to the best of our knowledge, it contains no material previously published by another person nor that which had been accepted for the award of any other degree of the university, except where due acknowledgement had been made in the text.

EWUSI KEREN EFUA	.....	08/09/2023 .....
Student	Signature	Date
GYAMFI BRIGHT OKYERE	.....	08/09/2023 .....
Student	Signature	Date
SAFOA IRENE	.....	08/09/2023 .....
Student	Signature	Date

Certified by:

DR. REINDORF NARTEY BORKOR	.....	08/09/2023 .....
Supervisor	Signature	Date

Certified by:

REV. PROF. OBENG DENTEH	.....	.....
Head of Department	Signature	Date

DEDICATION

This research paper is dedicated to our warm-hearted supervisor, Dr. Reindorf Borkor who provided assistance, guidance and encouragement throughout to the completion of our research work and not forgetting our families and friends who have supported us in the advancement of our academic career.

Abstract

Human pharmacokinetics and the evaluation of its parameters is of great importance in the drug study and development process. Traditional prediction methods involved in-vivo, in-vitro experimenting and analysis. These models are often used to describe the relationship between blood plasma or relevant tissue concentration of the drug and time and are built using compartments. Although the traditional methods have been successfully employed in drug discovery, increasing complexity of drug compounds in the drug discovery process has made drug absorption and disposition more important. Physiologically Based Pharmacokinetic (PBPK) models employing in silico estimation and analysis of pharmacokinetic parameters has proved to be a suitable alternative to the in-vivo method due to it's time and cost efficiency. This research seeks to integrate PK models with ML models to estimate several Pharmacokinetic (PK) parameters.

The limitations of the PK models were also discussed.

Acknowledgements

We thank the Almighty God for His grace all along through this work. We would also like to express our deepest gratitude to our supervisor, Dr. Reindorf Borkor as well as our parents for their support and prayers. We say God richly bless you all.

Contents

LIST OF ABBREVIATIONS

PK ..... Pharmacokinetics.

PBPK ..... Physiologically Based Pharmacokinetics.

MRT ..... Mean Residence Time.

Cl ..... Clearance.

c ..... Constant for calculating Clearance.

Fu ..... Fraction Unbound.

Vd ..... Volume of Distribution.

AUC ..... Area Under Curve.

FIH ..... First-In-Human.

DDI ..... Drug-Drug Interaction.

NCA ..... Non-Compartmental Model.

$T_{\frac{1}{2}}$ ..... Half Life.

X_0 ..... Amount of drug in the body at time, t = 0.

C_0 ..... Concentration of drug in the body at time, t = 0.

X_g ..... Amount of drug or compound in the gut.

$X_g(t)$ ..... Amount of drug or compound in the gut per time.

k_a ..... Absorption rate constant.

t ..... Time.

V_h ..... Volume of the liver.

Q_h ..... Hepatic blood flow rate.

C_h ..... Hepatic substrate concentration.

C_b ..... Blood substrate concentration.

$K_{p,h}$ ..... Liver-to-plasma concentration ratio.

R_b ..... Blood-to-plasma concentration ratio.

$f_{u,p}$ ..... Plasma unbound fraction.

V_1 ..... Volume of systemic circulation.

CL_h ..... Hepatic clearance.

$CL_{h,int}$ ..... Hepatic intrinsic clearance.

RF ..... Random Forest.

MLR ..... Multiple Linear Regression.

SVR ..... Support Vector Regression.

MSE ..... Mean Squared Error.

MAE ..... Mean Absolute Error.

$RMSE$ ..... Root Mean Squared Error.

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

1.1 Background of Study

Pharmacokinetics is the movement of a drug through the body's biological systems,these processes include absorption, distribution, metabolism, and elimination.

Pharmacokinetic studies evaluate;

- The rate chemical compounds are absorbed and distributed in the body.
- The rate and pathways for metabolism and elimination of compounds.
- The plasma concentration of a drug over time.

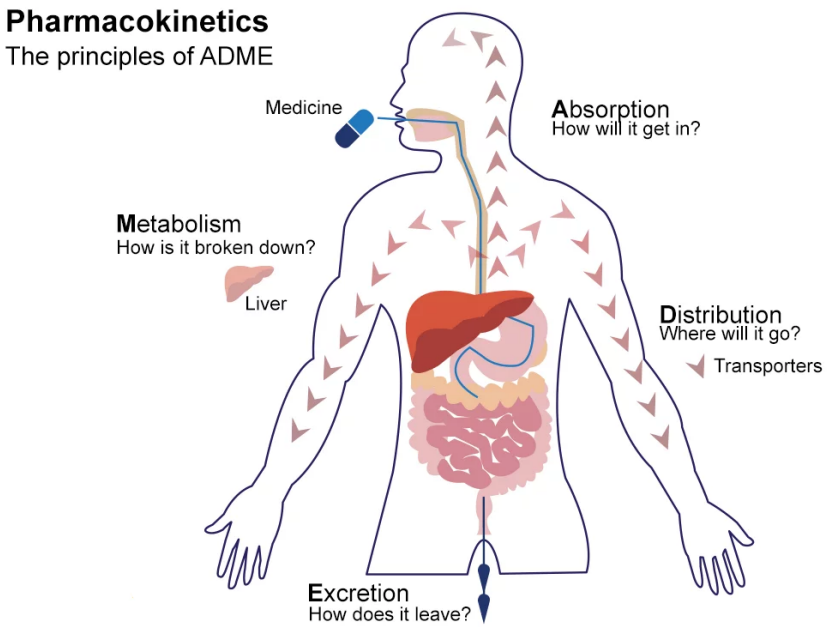

Figure 1.1: ADME Processes in Pharmacokinetics

Absorption.

Describes how a chemical enters the body. Absorption relates to the movement of a chemical from it's route of administration to the bloodstream.The route of administration can be:

- Intravenous infusion.
- Intravenous bolus.
- Extravascular.

Distribution.

Once a drug has been absorbed, it moves from the absorption site to tissues around the body. This distribution from one part of the body to another is typically accomplished via the bloodstream, but it can also occur from cell-to-cell.

Drug metabolism.

Is the bio-transformation of a drug by organs or tissues (primarily the liver, kidney, skin or digestive tract) so that the drug can be excreted. To facilitate removal via faeces or urine, the drug compound is altered to become more water-soluble.

Excretion.

Is the process by which the metabolized drug compound is eliminated from the body. Researchers want to know how rapidly the drug is excreted and what pathway it takes to exit the body. Most drug excretion occurs as faeces or urine. Other excretion methods include through the lungs or in sweat through the skin. Molecular size and charge influence the excretion pathway.

Not every drug compound is fully excreted. When the chemical or metabolic by-products bio-accumulate, adverse effects can occur. Lipid-soluble compounds are more prone to bio-accumulate compared to water-soluble compounds.

1.2 Problem Statement

- Pharmacokinetics (PK) models are useful tools in drug development and risk assessment of environmental chemicals.
- PK model development requires the collection of species-specific data to describe the processes of absorption, distribution, metabolism, and excretion (ADME) of a drug substance, which can be a complex, time-consuming and expensive process.
- This raises a need to create computational models capable of predicting input parameter values for PK models, especially for new compounds.

1.3 Objective

- Develop one and two compartmental models
- Develop Machine Learning models to predict ADME parameters.

1.4 Methodology

We incorporated three (3) Machine Learning algorithms which are Support Vector Machine (SVM), Linear Regression and Random Forest to describe the statistical relationship and patterns between PK parameters and further used these relationships to make predictions on new compounds.

1.5 Justification

Developing ML models and incorporating them with pharmacokinetic models to estimate pk parameters with improved accuracy.

1.6 Report organization

- The first chapter discusses the background of the study, statement of the

problem as well as the objective of the study.It also discusses the methodology and justification of the study.

- The second chapter reviews various methods used by other researchers related to this study.
- The third chapter discusses the methodology.It also explains the mathematical formulation of the problem in the study.
- The fourth chapter shows the result and interpretations.
- The fifth chapter concludes the study with some recommendations. It also acknowledges the reference used in the study.

Chapter 2

Literature Review

2.1 Introduction

Predicting the relationship between ADME parameters has been done by many researchers using Mathematical methods or algorithms that give nearly accurate predictions.

Articles on ADME parameters and their predictions differ by parameters used, objectives and as well as methodology. Over the last few years the overall objective of works published fall under the prediction of parameters.

In this chapter we review some articles where predictions are made with different methodology. And these are classical methods and Machine learning based methods.

- Classical methods - Under this method is the compartmental models.Typical compartmental models have between one and three compartments, but more compartments can be added to the model depending on the application.

  To apply this method, the body is divided up into hypothetical compartments(one compartment,two compartment etc). Often, the containers used in compartment modeling do not represent actual physiological tissues in the body but are used as a proxy so that PK parameters can be determined.

- Machine learning based methods - Machine Learning is a sub-domain of Computer Science involved in developing and training algorithms to learn and extract meaningful insights from data.

  Machine Learning algorithms can be classified based on the method of training and data used.Most researchers use supervised learning algorithms for the predictions of PK parameters.

2.2 Classical models

Kamiya(2018) employed two different models which one was a simple one-compartment model and the other was a simple PBPK model consisting of a chemical receptor compartment, a metabolizing compartment, and a central compartment. He then established sets of differential equations for each test compound, solved to determine the concentration in each compartment and used the results for predictive analysis.

The equations describing the process are;

$$\frac{dX_g(t)}{dt} = -k_a \cdot X_g(t), \qquad t = 0, X_g(0) = dose \tag{2.1}$$

$$V_h \frac{dC_h}{dt} = Q_h \cdot C_b - \frac{Q_h \cdot C_h \cdot R_b}{K_{p,h}} + k_a \cdot X_g - CL_{h,int} \cdot \frac{C_h}{K_{p,h}} \cdot f_{u,p} \tag{2.2}$$

$$V_1 \frac{dC_b}{dt} = -Q_h \cdot C_b + \frac{Q_h \cdot C_h \cdot R_b}{K_{p,h}} - CL_r \cdot C_b \tag{2.3}$$

Adachi(2021) pharmacokinetically analyzed a frozen plasma samples collected from a patient who intentionally took an overdose of duloxetine (780g) in combination with three other antipsychotic drugs in a suicide attempt. Based on the reported human blood concentrations in patients orally treated with the normal therapeutic doses of the four antipsychotic drugs, four simple PBPK model consisting of receptor (gut), metabolizing (liver) and central compartments were seperately set up to find the relationship between the effects of the antipsychotic drugs when there is an overdose administration.
Rate constants for the transfer of drugs from and to the central (first) compartment to and from the peripheral (second) compartment were adopted for the four antipsychotic drugs.

Equations for the compartments are;

$$\frac{dX_g(t)}{dt} = -k_a \cdot X_g(t), \qquad t = 0, X_g(0) = dose \tag{2.4}$$

$$V_h \frac{dC_h}{dt} = Q_h \cdot C_b - \frac{Q_h \cdot C_h \cdot R_b}{K_{p,h}} + k_a \cdot X_g - CL_{h,int} \cdot \frac{C_h}{K_{p,h}} \cdot f_{u,p} \tag{2.5}$$

$$V_1 \frac{dC_b}{dt} = -Q_h \cdot C_b + \frac{Q_h \cdot C_h \cdot R_b}{K_{p,h}} - k_{12} \cdot V_1 \cdot C_b + k_{21} \cdot X_{peripheral} - CL_r \cdot C_b \tag{2.6}$$

$$\frac{dX_{peripheral}}{dt} = K_{12} \cdot V_1 \cdot C_b - K_{12} \cdot X_{peripheral} \tag{2.7}$$

2.3 Machine learning Models

Wang(2019) built a quantitative structure-property relationship studies to predict four human pharmacokinetics parameters including volume of distribution at steady state (VD), Clearance (CL), Terminal half-life, and fraction unbound in plasma (fu), using a data set consisting of 1352 drugs. A series of regression models were built using most suitable features selected by four machine learning methods including Support Vector Machine (SVM), Random Forest (RF), Gradient boosting machine (GBM), and XGBoost (XGB). In his analysis, SVM showed the best performance with R-squared equal to 0.870 and RMSE equal to 0.208 even though the other models demonstrated excellent stability and predictive ability. As he compared with other published models for human pharmacokinetics parameter estimation , it was confirmed that the machine learning models he used obtained better predictive ability and could be used in the selection of pre-clinical candidates.

Watanabe(2018) researched and developed a machine learning model to predict the fu in plasma. Molecular drug descriptors for each compound was calculated, the descriptors also distinguished between compounds. A correlation matrix to determine the relationship between compound parameters and dimensional analysis for the compounds were developed. Regression, binary classification and

three-class classification models were constructed to predict fu using the random forest (RF), support vector machine (SVM), k-nearest neighbours (kNN), artificial neural network (ANN), AdaBoost and partial least squares algorithms (PLS). 10-fold cross-validation was implemented in training for model evaluation, accuracy and sensitivity was used also to evaluate the models on the test set. Statistical results of the binary and three-class classification models were recorded and the accuracy of training and test set compared. The random forest model was the best performing model with a training and testing accuracy of 0.809 and 0.801 for binary classification and 0.704 and 0.676 for the three-class classification. The AdaBoost model was the next best performing model for binary classification with 0.809 and 0.793 for training and testing and support vector machine model with accuracy values of 0.693 and 0.673 for the three-class classification.

Hou(2007) conducted an ADME evaluation on the prediction of human intestinal absorption by a Support Vector Machine. He studied the performance of a support vector machine to classify compounds with high or low fractional absorption on a data set consisting of 578 structural diverse drug-like molecules which have been divided into a 480-molecule training set and a 98-molecule testing set. He then generated Ten SVM classification to investigate the impact of different individual molecular properties. After further analysis on the influence of the size of the training set and unbalanced nature of the data set, the analysis demonstrated that large data set is necessary for the SVM model and then scaling the data set as well helps in accurate predictive analysis.

Chapter 3

Methodology

3.1 Introduction

In this chapter, we will focus on the formulation of mathematical equations for one and two compartment models.
We will also discuss the mathematical equations behind the supervised machine learning algorithms used.

3.2 Problem Description

Pharmacokinetic modelling is the task of predicting and describing the processes and pathways undertaken by chemical compounds from administration to disposition.
In our research, compartmental equations were developed, along with machine learning models to establish the relationship between compounds and predict drug parameters.

3.2.1 Some Terminology

- Volume of distribution (Vd).
  the apparent volumes of fluid in which the drug is distributed in the body, expressed in units of volume (L). Vd is proportional to the amount of drug in the body divided by the concentration in the plasma.

- Clearance (CL).
  the rate at which the drug is eliminated from the body, often expressed in units of volume/time (L/hour). Clearance is proportional to the rate constant for elimination from the central compartment.

- Fraction unbound in plasma (f_u).

  Proportion of the drug not bound to the plasma proteins in the blood stream.

- Mean Residence Time (MRT).

  The average amount of time the molecules of a dosed drug stays in the body, expressed in units of time (Hours).

- Terminal half-life ($t_{1/2}$).

  the time it takes for the drug concentration in the plasma to decrease by half, expressed in units of time (Hours). Half-life is related to the rate constants for elimination from the central compartment and can be calculated from Clearance and Volume of distribution.

3.2.2 Assumptions

Assumptions for one compartment models:

1. One compartment models assume the entire body as a single compartment and drugs administrated distribute quickly through the system.

2. Drug directly enters the bloodstream hence no absorption is involved.

3. Drug elimination follows first order linear kinetics. This means a constant proportion of the drug is eliminated from the compartment per unit time.

4. Infusion rate, for drugs administered by infusion, follows zero order kinetics meaning a constant amount of the drug is introduced into the compartment per unit time.

5. The rate of elimination is proportional to the amount of drug in the body.

Assumptions for two compartment models:

1. Two compartment models assume the body as being made up of two compartments, a central and peripheral compartment. The central compartment represents highly perfused tissues such as liver, kidney, whilst the peripheral compartment represents lowly perfused tissues such as the bones and ligaments.

2. Drug directly enters the bloodstream hence no absorption is involved.

3. Elimination takes place in the central compartment and follows first order linear kinetics.

3.2.3 Parameters

- C - Concentration.
- C_0 - Concentration at $t = 0$.
- X - Amount of drug.
- X_0 - Amount of drug at $t = 0$.
- D_0 - Volume of Distribution, at $t = 0$.
- R_0 - Infusion rate.
- Cl - Clearance.
- Vd - Volume of Distribution.
- t - Time.
- k_a - Rate of absorption constant.
- k_e - Rate of elimination constant.
- dt - Change in time.
- dX - Change in drug.

- C_c - Concentration in central compartment.
- C_p - Concentration in peripheral compartment.
- X_c - Drug amount in central compartment.
- X_p - Drug amount in peripheral compartment.
- V_c - Volume in central compartment.
- V_p - Volume in peripheral compartment.
- K_{12} - Rate constant for central to peripheral compartment.
- K_{21} - Rate constant for peripheral to peripheral compartment.
- K_E - Elimination constant.
- $\lambda_{1,2}$ - Equation roots.
- α - Root constant.
- β - Root constant.
- $\omega_{1,2}$ - Eigenvectors.

3.2.4 One Compartment Intraveneous Bolus Equation

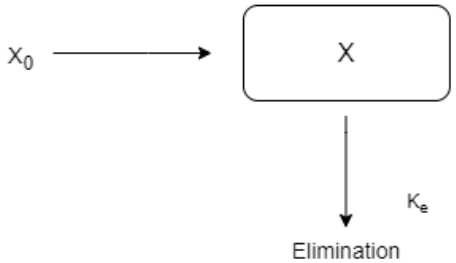

Figure 3.1: One Compartment Iv Bolus Model

Letting ;

k_a = rate of absorption.

X = amount of drug present.

k_e = rate of elimination.

Change in the amount of drug in the body, at any time = rate in - rate out.

$$\frac{dX}{dt} = k_a - k_e X \tag{3.1}$$

but $k_a = 0$.

$$\frac{dX}{dt} = -k_e X \tag{3.2}$$

Using separation of variables.

$$dX = -k_e X dt \tag{3.3}$$

$$\frac{dX}{X} = -k_e dt \tag{3.4}$$

Integrating both sides.

$$\int \frac{dX}{X} = \int -k_e dt \tag{3.5}$$

$$\ln |X| = -k_e t + c \tag{3.6}$$

$$X = e^{-k_e t + c} \tag{3.7}$$

$$X = e^{-k_e t} e^{c} \tag{3.8}$$

$$X = ce^{-k_e t}, \qquad e^{c} = c \tag{3.9}$$

$$\text{Let } c = X_0. \tag{3.10}$$

Substituting Eqn.(3.10) into Eqn.(3.9).

$$X = X_0 e^{-k_e t} \tag{3.11}$$

Concentration is proportional to amount of drug, from the first assumption.

$$\therefore C = C_0 e^{-k_e t} \tag{3.12}$$

Expressing Bolus injection in terms of Volume of distribution and Clearance.

$$Vd = \frac{X}{C} \tag{3.13}$$

Let $X = D_o, \quad C = C_0$.

$$Vd = \frac{D_0}{C_0} \tag{3.14}$$

$$C_0 = \frac{D_0}{Vd} \tag{3.15}$$

Clearance = rate of elimination / concentration.

$$Cl = \frac{k_e X}{C} \tag{3.16}$$

$$Cl = k_e(\frac{X}{C}) \tag{3.17}$$

$$Vd = \frac{X}{C} \tag{3.18}$$

$$Cl = k_e Vd \tag{3.19}$$

$$k_e = \frac{Cl}{Vd} \tag{3.20}$$

Substituting in Eqn.(3.15) and Eqn.(3.20) into Eqn.(3.12).

$$C = \frac{D_0}{Vd} e^{-(\frac{Cl}{Vd})t} \tag{3.21}$$

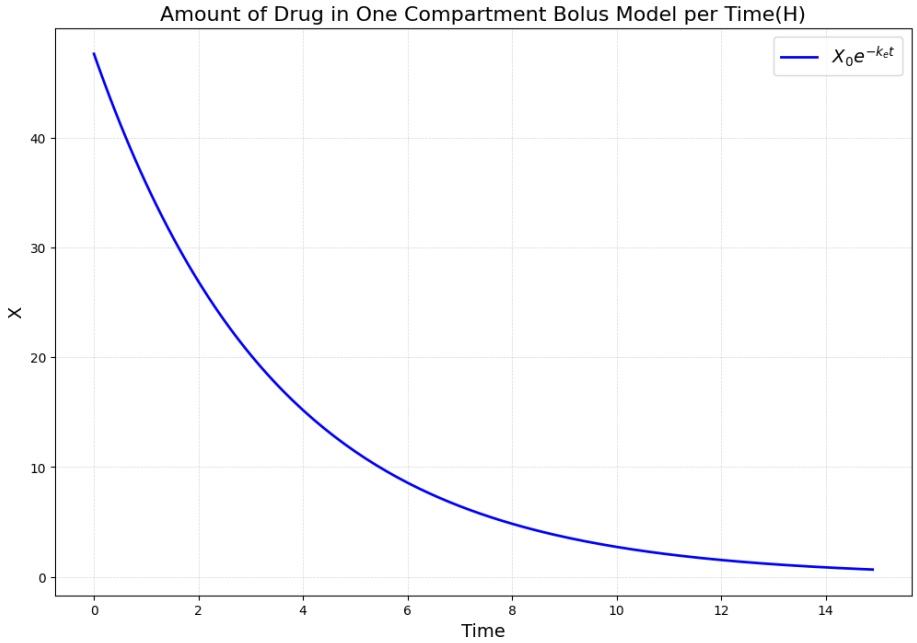

Figure 3.2: One Compartment IV Bolus Model Graph

3.2.5 One Compartment Intraveneous Infusion Equation

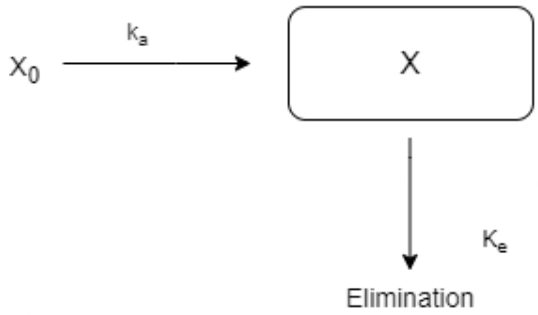

Figure 3.3: One Compartment Iv Bolus Model

For Infusion, drug input and output is considered.

$\frac{dX}{dt}$ = Infusion rate - rate of elimination.

$$\frac{dX}{dt} = R_0 - k_e X \tag{3.22}$$

Solving for X.

$$u = R_0 - k_e X \tag{3.23}$$

$$\frac{du}{dX} = -k_e \tag{3.24}$$

$$dX = \frac{-1}{k_e} du \tag{3.25}$$

Substituting Eqn.(3.23) into Eqn.(3.22).

$$\frac{dX}{dt} = u \tag{3.26}$$

$$\frac{dX}{u} = dt \tag{3.27}$$

Substituting Eqn.(3.25) into Eqn.(3.27).

$$\frac{-1}{k_e u} du = dt \tag{3.28}$$

$$\int \frac{-1}{k_e u} du = \int dt \tag{3.29}$$

$$\frac{-1}{k_e} \ln |u| = t + c \tag{3.30}$$

$$-\ln|u| = k_e(t+c) \tag{3.31}$$

$$\ln u^{-1} = k_e t + k_e c \tag{3.32}$$

$$u^{-1} = e^{k_e t + k_e c} \tag{3.33}$$

$$u = e^{-k_e t - k_e c} \tag{3.34}$$

$$u = e^{-k_e t} \cdot e^{-k_e c} \tag{3.35}$$

Let $R_0 = e^{-k_e c}$.

$$u = R_0 \cdot e^{-k_e t} \tag{3.36}$$

Substituting Eqn.(23) into Eqn.(36).

$$R_0 - k_e X = R_0 e^{-k_e t} \tag{3.37}$$

$$k_e X = R_0 - R_0 e^{-k_e t} \tag{3.38}$$

$$k_e X = R_0(1 - e^{-k_e t}) \tag{3.39}$$

$$X = \frac{R_0}{k_e}(1 - e^{-k_e t}) \tag{3.40}$$

Volume of distribution, $Vd = \frac{X}{C}$.

$$X = Vd \cdot C \tag{3.41}$$

$$Vd \cdot C = \frac{R_0}{k_e}(1 - e^{-k_e t}) \tag{3.42}$$

$$C = \frac{R_0}{k_e Vd}(1 - e^{-k_e t}) \tag{3.43}$$

Clearance, $\quad Cl = k_e Vd.$

$$C = \frac{R_0}{Cl}(1 - e^{-k_e t}) \tag{3.44}$$

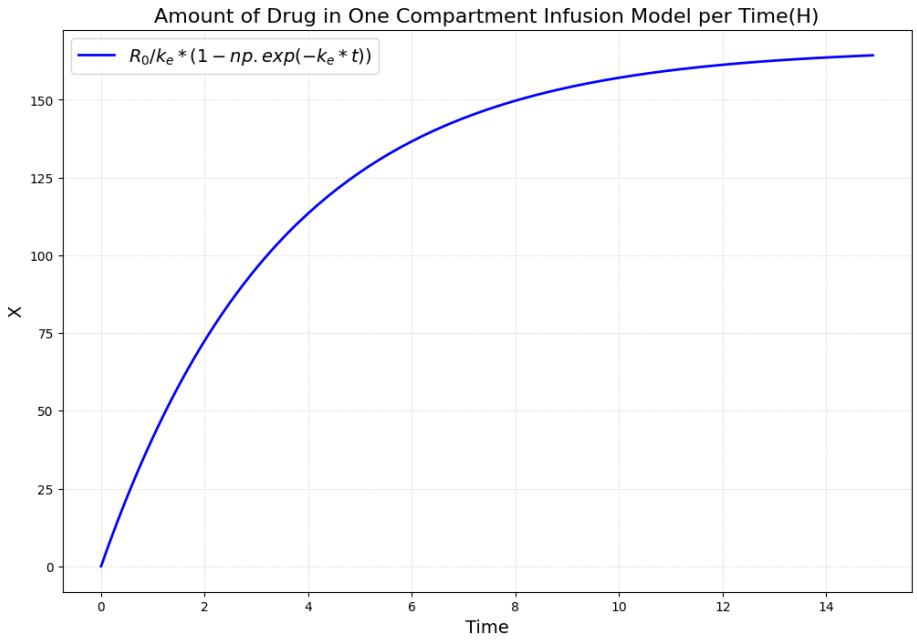

Figure 3.4: One Compartment IV Infusion Model Graph

3.2.6 Two Compartment Intraveneous Bolus Equation

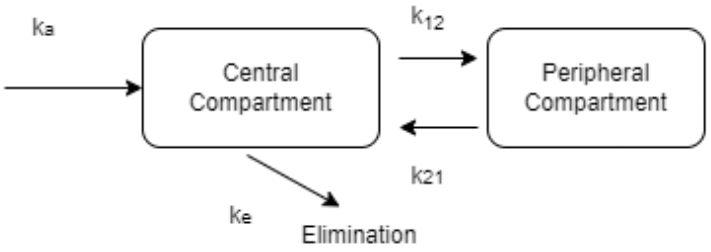

Figure 3.5: Two Compartment IV Bolus Model

Rate of change of concentration of drug in the body = rate in - rate out. For the Central Compartment.

$$\frac{dC_c}{dt} = K_{21}C_p - (K_{12}C_c + K_E C_c) \tag{3.45}$$

Expressing in terms of amount of drug.

$$\frac{dX_c}{V_c dt} = \frac{K_{21}X_p}{V_p} - (K_{12} + K_E)\frac{X_c}{V_c} \tag{3.46}$$

$$\frac{dX_c}{dt} = K_{21}X_p\frac{V_c}{V_p} - (K_{12} + K_E)X_c \tag{3.47}$$

$$X_c{}' = K_{21}X_p\frac{V_c}{V_p} - (K_{12} + K_E)X_c \tag{3.48}$$

For the Peripheral Compartment.

Rate of change of concentration of drug in the body = rate in - rate out.

$$\frac{dC_p}{dt} = K_{12}C_c - K_{21}C_p \tag{3.49}$$

Expressing in terms of amount of X.

$$\frac{dX_p}{V_p dt} = \frac{K_{12}X_c}{V_c} - K_{21}\frac{X_p}{V_p} \tag{3.50}$$

$$\frac{dX_p}{dt} = K_{12}X_c\frac{V_p}{V_c} - K_{21}X_p \tag{3.51}$$

$$X_p{}' = -K_{21}X_p + K_{12}(\frac{V_p}{V_c})X_c \tag{3.52}$$

Putting Eqn.(3.44) into Eqn.(3.52) into a matrix.

$$\frac{dX}{dt} = \begin{bmatrix} \frac{dX_p}{dt} \\ \frac{dX_c}{dt} \end{bmatrix} = \begin{bmatrix} -K_{21} & K_{12}\frac{V_p}{V_c} \\ K_{21}\frac{V_c}{V_p} & -(K_{12} + K_E) \end{bmatrix} \begin{bmatrix} X_p \\ X_c \end{bmatrix} \tag{3.53}$$

$$\det\,(A - \lambda I) = 0$$

$$(A - \lambda I) = \begin{bmatrix} -K_{21} & K_{12}\frac{V_p}{V_c} \\ K_{21}\frac{V_c}{V_p} & -(K_{12} + K_E) \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \tag{3.54}$$

$$\begin{vmatrix} -K_{21} - \lambda & K_{12}\frac{V_p}{V_c} \\ K_{21}\frac{V_c}{V_p} & -(K_{12} + K_E) - \lambda \end{vmatrix} = 0 \tag{3.55}$$

$$(-K_{21} - \lambda)(-(K_{12} + K_E) - \lambda) - (K_{21}\frac{V_c}{V_p} \cdot K_{12}\frac{V_p}{V_c}) = 0 \tag{3.56}$$

$$(-K_{21} - \lambda)(-K_{12} - K_E - \lambda) - (K_{21} \cdot K_{12}) = 0 \tag{3.57}$$

$$K_{21}K_{12} + K_{21}K_E + K_{21}\lambda + K_{12}\lambda + K_E\lambda + \lambda^2 - K_{21}K_{12} = 0 \tag{3.58}$$

$$K_{21}K_E + K_{21}\lambda + K_{12}\lambda + K_E\lambda + \lambda^2 = 0 \tag{3.59}$$

$$\lambda^2 + (K_{21} + K_{12} + K_E)\lambda + K_{21}K_E = 0 \tag{3.60}$$

Checking for the roots using $b^2 - 4ac$.

$$(K_{21} + K_{12} + K_E)^2 - 4K_{21}K_E > 0 \tag{3.61}$$

Eqn.(3.61) has two real and distinct roots, λ_1 & λ_2.

Solving for λ_1 & λ_2

$$\lambda_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = 0 \tag{3.62}$$

$$\frac{-(K_{21} + K_{12} + K_E) \pm \sqrt{(K_{21} + K_{12} + K_E)^2 - 4\lambda^2 K_{21}K_E}}{2} = 0 \tag{3.63}$$

Comparing Eqn.(3.61) to $ax^2 + bx + c = 0$.

$$a = 1 \tag{3.64}$$

$$b = (K_{21} + K_{12} + K_E) \tag{3.65}$$

$$c = K_{21}K_E \tag{3.66}$$

Let:

$$\lambda_1 = -\alpha \tag{3.67}$$

$$\lambda_2 = -\beta \tag{3.68}$$

$$\text{sum of roots} = \lambda_1 + \lambda_2 = \frac{-b}{a} \tag{3.69}$$

$$-\alpha - \beta = -(K_{21} + K_{12} + K_E) \tag{3.70}$$

$$\text{product of roots} = \lambda_1 \cdot \lambda_2 = \frac{c}{a} \tag{3.71}$$

$$\alpha\beta = (K_{21}K_E) \tag{3.72}$$

The general solution of Eqn.(3.48) and Eqn.(3.52).

$$X = Cw_1e^{\lambda_1 t} + Cw_2e^{\lambda_2 t} \tag{3.73}$$

$$X = C(w_1e^{\lambda_1 t} + w_2e^{\lambda_2 t}) \tag{3.74}$$

Let $C = X_0$

$$X = X_0(w_1e^{\lambda_1 t} + w_2e^{\lambda_2 t}) \tag{3.75}$$

w_1 & w_2 are the corresponding eigenvectors for λ_1 & λ_2

$$X = X_0\Bigg(\begin{bmatrix} w_{1p} \\ w_{1c} \end{bmatrix} e^{\lambda_1 t} + \begin{bmatrix} w_{2p} \\ w_{2c} \end{bmatrix} e^{\lambda_2 t} \Bigg) \tag{3.76}$$

Solving for $\lambda_1 = -\alpha$

$$(A - \lambda I) = \Bigg(\begin{bmatrix} K_{21} & -K_{12}\frac{V_p}{V_c} \\ -K_{21}\frac{V_c}{V_p} & (K_{12} + K_E) \end{bmatrix} - \begin{bmatrix} -\alpha & 0 \\ 0 & -\alpha \end{bmatrix} \Bigg) \begin{bmatrix} w_{1p} \\ w_{1c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.77}$$

$$(A - \lambda I) = \begin{bmatrix} K_{21} + \alpha & -K_{12}\frac{V_p}{V_c} \\ -K_{21}\frac{V_c}{V_p} & (K_{12} + K_E) + \alpha \end{bmatrix} \begin{bmatrix} w_{1p} \\ w_{1c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.78}$$

$$\begin{bmatrix} K_{21} - \alpha & -K_{12}\frac{V_p}{V_c} \\ -K_{21}\frac{V_c}{V_p} & (K_{12} + K_E) - \alpha \end{bmatrix} \begin{bmatrix} w_{1p} \\ w_{1c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.79}$$

$$w_{1p}(K_{21} + \alpha) - K_{12}(\frac{V_p}{V_c})w_{1c} = 0 \tag{3.80}$$

$$-K_{21}(\frac{V_c}{V_p})w_{1p} + (K_{12} + K_E + \alpha)w_{1c} \tag{3.81}$$

Solving for the eigenvectors.

$$\begin{bmatrix} w_{1p} \\ w_{1c} \end{bmatrix} = \begin{bmatrix} \frac{K_{12}}{\beta - \alpha} \\ \frac{K_{21} - \alpha}{\beta - \alpha} \end{bmatrix} \tag{3.82}$$

Solving for $\lambda_2 = -\beta$

$$(A - \lambda I) = \Bigg(\begin{bmatrix} K_{21} & -K_{12}\frac{V_p}{V_c} \\ -K_{21}\frac{V_c}{V_p} & (K_{12} + K_E) \end{bmatrix} - \begin{bmatrix} -\beta & 0 \\ 0 & -\beta \end{bmatrix} \Bigg) \begin{bmatrix} w_{1p} \\ w_{1c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.83}$$

$$(A - \lambda I) = \begin{bmatrix} K_{21} + \beta & -K_{12}\frac{V_p}{V_c} \\ -K_{21}\frac{V_c}{V_p} & (K_{12} + K_E) + \beta \end{bmatrix} \begin{bmatrix} w_{1p} \\ w_{1c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.84}$$

$$\begin{bmatrix} K_{21} - \beta & -K_{12}\frac{V_p}{V_c} \\ -K_{21}\frac{V_c}{V_p} & (K_{12} + K_E) - \beta \end{bmatrix} \begin{bmatrix} w_{1p} \\ w_{1c} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{3.85}$$

$$w_{1p}(K_{21} + \beta) - K_{12}(\frac{V_p}{V_c})w_{1c} = 0 \tag{3.86}$$

$$-K_{21}(\frac{V_c}{V_p})w_{1p} + (K_{12} + K_E + \beta)w_{1c} \tag{3.87}$$

Solving for the eigenvectors.

$$\begin{bmatrix} w_{2p} \\ w_{2c} \end{bmatrix} = \begin{bmatrix} \frac{K_{12}}{\alpha-\beta} \\ \frac{K_{21}-\alpha}{\alpha-\beta} \end{bmatrix} \tag{3.88}$$

Substituting Eqn.(3.67), Eqn.(3.68), Eqn.(3.82) and Eqn.(3.88) into Eqn.(3.76).

$$X = X_0\left(\begin{bmatrix} \frac{K_{12}}{\beta-\alpha} \\ \frac{K_{21}-\alpha}{\beta-\alpha} \end{bmatrix} e^{-\alpha t} + \begin{bmatrix} \frac{K_{12}}{\alpha-\beta} \\ \frac{K_{21}-\alpha}{\alpha-\beta} \end{bmatrix} e^{-\beta t} \right) \tag{3.89}$$

$$X_p = X_0\left[\frac{K_{12}}{\beta-\alpha}e^{-\alpha t} + \frac{K_{12}}{\alpha-\beta}e^{-\beta t} \right] \tag{3.90}$$

$$X_c = X_0\left[\frac{K_{21}-\alpha}{\beta-\alpha}e^{-\alpha t} + \frac{K_{21}-\alpha}{\alpha-\beta} \right] \tag{3.91}$$

Expressing in terms of concentration.

$$X = Vd \cdot C \tag{3.92}$$

Substitute Eqn.(3.92) into Eqn.(3.91) & Eqn.(3.90).

$$C_c = \frac{X_0}{V_c}\left[\left(\frac{K_{21}-\alpha}{\beta-\alpha} \right) e^{-\alpha t} + \left(\frac{K_{21}-\beta}{\alpha-\beta} \right) e^{-\beta t} \right] \tag{3.93}$$

$$C_p = \frac{X_0}{V_p}\left[\left(\frac{K_{12}}{\beta-\alpha} \right) e^{-\alpha t} + \left(\frac{K_{12}}{\alpha-\beta} \right) e^{-\beta t} \right] \tag{3.94}$$

From Eqn.(3.63).

$$\beta = \frac{-(K_{21} + K_{12} + K_E) + \sqrt{(K_{21} + K_{12} + K_E)^2 - 4\lambda^2 K_{21} K_E}}{2} \tag{3.95}$$

$$\alpha = \frac{K_{21} K_E}{\beta} \tag{3.96}$$

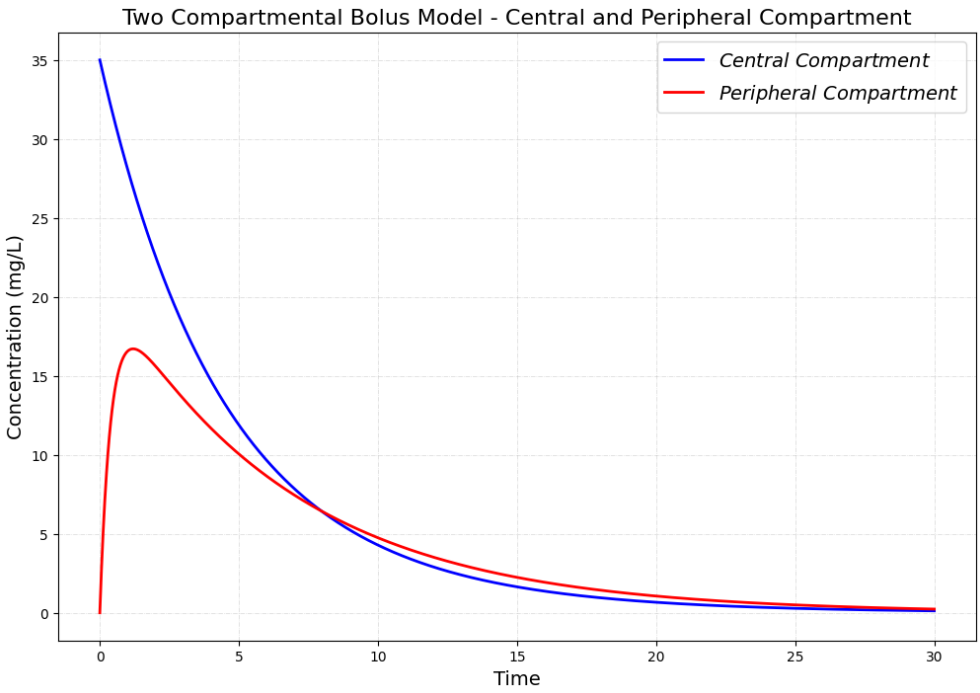

Figure 3.6: Two Compartment IV Bolus Model Graph

3.2.7 Simple PBPK Model Equation

$$\frac{dX_g(t)}{dt} = -k_a \cdot X_g(t), \qquad t = 0, X_g(0) = dose \tag{3.97}$$

$$V_h \frac{dC_h}{dt} = Q_h \cdot C_b - \frac{Q_h \cdot C_h \cdot R_b}{K_{p,h}} + k_a \cdot X_g - CL_{h,int} \cdot \frac{C_h}{K_{p,h}} \cdot f_{u,p} \tag{3.98}$$

$$V_1 \frac{dC_b}{dt} = -Q_h \cdot C_b + \frac{Q_h \cdot C_h \cdot R_b}{K_{p,h}} - CL_r \cdot C_b \tag{3.99}$$

3.2.8 Machine Learning Model

Linear Regression Model Equation

The equation for Linear Regression model is of the form :

$$y_i = A + bx_i$$

Y_i is the Dependent variable.

X_i is the independent variable.

A : intercept, the point at which the best-fit line touches the vertical axis.

b : slope, the measure of the steepness of the best-fit line.

The residual ϵ is the difference between the predicted dependent variable $y_{predicted}$ and the actual dependent variable y_i.

$$\epsilon_i = y_i - \hat{y}_i \tag{3.100}$$

The Cost function evaluates the performance of the regression model.

The Mean Squared Error Cost function often used, is of the form;

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.101}$$

n refers to the number of terms.

$$(y_i - \hat{y}_i) \text{ computes the error term.} \tag{3.102}$$

The Gradient Descent algorithm is used to optimise the algorithm by updating the coefficients and obtain the best values for the parameters A and b and reach the minimum cost function.

The Cost function optimised with Gradient Descent becomes;

$$A = A - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{3.103}$$

$$b = b - \alpha \cdot \frac{2}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \cdot x_i \tag{3.104}$$

The error evaluation metrics commonly used with Linear Regression are;

Coefficient of Determination or R-Squared (R2).

This measures the proportion of variance captured by the model. The value ranges between 0 and 1, with a higher R-squared model meaning the model fits the data very well.

$$R^2 = 1 - \left[\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y_i})^2}\right] \tag{3.105}$$

Root Mean Squared Error.

This is the square root of the variance of the residuals and specifies the absolute

fit of the model to the data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{3.106}$$

Support Vector Regression Model Equation

To classify a point X, we assume the point as a vector, and introduce a vector $\vec{w}$ perpendicular to the hyperplane.

The dot product of the two vectors are taken and the distance of $\vec{w}$ from the origin to the decision boundary is denoted as c.

$\vec{X} \cdot \vec{w} = c$ the point lies on the decision boundary.

$\vec{X} \cdot \vec{w} \geq c$ the point is classified as a positive sample.

$\vec{X} \cdot \vec{w} \leq c$ the point is classified as a negative sample.

The Decision rule is then:

$$\vec{X} \cdot \vec{w} + b \geq 0 \quad \text{for} \quad b \in \mathbb{R} \tag{3.107}$$

To maximise the margin, the constraints taken into consideration are;

For negative classes:

$$\vec{X_i} \cdot \vec{w} + b \leq -1 \tag{3.108}$$

For positive classes:

$$\vec{X} \cdot \vec{w} + b \geq 1 \tag{3.109}$$

This should always be true for a point to be correctly classified.

$$y_i(\vec{X_i} \cdot \vec{w} + b) \geq 1 \tag{3.110}$$

The dot-product of the two vectors give the projection of both vectors and the objective function.

Assuming x_2 and x_1 are support vectors.

$$(x_+ - x_-) \cdot \frac{\vec{w}}{||w||} \tag{3.111}$$

$$\frac{x_+ \cdot \vec{w} - x_+ \cdot \vec{w}}{||w||} \tag{3.112}$$

For positive point y = 1.

$$1 \times (\vec{w} \cdot x_+ + b) = 1. \tag{3.113}$$

$$\vec{w} \cdot x_+ = 1 - b \tag{3.114}$$

Similarly for negative point y = -1.

$$-1 \times (\vec{w} \cdot x_- + b) = 1 \tag{3.115}$$

$$\vec{w} \cdot x_- = -b - 1 \tag{3.116}$$

Putting Eqn.(3.114) & Eqn.(3.116) in Eqn.(3.107)

$$\frac{(1-b) - (-b-1)}{||w||} \tag{3.117}$$

$$\frac{1 - b + b + 1}{||w||} = \frac{2}{||w||} = d \tag{3.118}$$

The objective function to be maximised is:

$$J(w^*, b^*) = \frac{2}{||w||} \tag{3.119}$$

$$y_i(\vec{w} \cdot \vec{X}_i + b) \geq 1 \tag{3.120}$$

Random Forest Regression Model Equation

For a regression model, the random forest algorithm calculates variance reduction using the Mean Square Error and Mean Absolute Error.

$$\frac{1}{N} \sum_{i=1}^{N} (y_i - \mu)^2 \tag{3.121}$$

$$\frac{1}{N} \sum_{i=1}^{N} |y_i - \mu| \tag{3.122}$$

y_i is the label for an instance, N is the number of instances and μ is the mean.

$$\frac{1}{N} \sum_{i=1}^{N} y_i$$

3.2.9 Machine Learning Model Development

The pharmacokinetic data used in this study was obtained from Lombardo dataset(15). The dataset comprises of 1352 compounds and is an expansion of the Obach et al. (2008) dataset. Compounds with molecular weight of approximately 900 daltons and below easily diffuse across cell membranes, hence compounds beyond this threshold were removed.

Data analysis and cleaning were performed with the pandas library. This process involved removal of null values for corresponding parameters VD_{ss}, CL, $t_{1/2}$, f_u and MRT, and log transformation for the values of the parameters, to normalize the data and reduce the impact of outliers.

Data visualisation was performed in Python using the matplotlib and plotly libraries. The scikitlearn library was used to build the prediction models.
Three models to predict the numerical values of each of the parameters from the remaining parameters were developed using three machine learning methods; random forest, support vector regression and multiple linear regression.
Model performance is calculated by the MAE and MSE values.

MLR Model

The multiple linear regression model uses four of the five parameters as the predictor value, the remaining parameter serves as the predicted value. The dataset is randomly split into two parts, a training and testing set which comprises 70% and 30% of the dataset respectively. The random state hyperparameter is also included to increase randomness in the data and prevent overfitting.

SVR Model

The support vector regression model uses a linear approach and four of the five parameters as the predictor value, the remaining parameter serves as the predicted value. The dataset is randomly split into two sets of 70% and 30% representing the training and testing sets respectively. The random forest hyperparameter is used to increase randomness and decrease overfitting. The regularisation parameter C is included to regulate the margin size to fit as much data points as

possible, a larger value however makes the model prone to overfitting hence a C value of 1 is used.

RF Model

The random forest regressor model similarly uses four of the five parameters as the predictor value with the remaining parameter serving as the predicted value. The dataset is randomly split into training and test sets comprising of 70% and 30% of the dataset respectively. The algorithm has 100 trees and 10 nodes for depth and variation in model results and a learning rate value of 0.025 to aid in updating equation parameter values to optimal values. The hyperparameters also aid in preventing overfitting.

Chapter 4

Results And Analysis

4.1 Introduction

We shall discuss in this chapter the data used in our model, the implementation and performance of the selected algorithms on the data and the interpretation of the results obtained.

4.2 Test Specification

The drug parameter values were transformed to the logarithmic scale for a more evenly distributed dataset. Regression models were constructed with three machine learninng methods. Model performance was evaluated by the Mean Absolute Error (MAE) and Mean Square Error (MSE) evaluation metrics, results are shown in Table 4.1.

The MSE is the average of the squared difference between the actual values and the predicted values and the MAE is average of the absolute difference between the actual values and the predicted values. Small MSE and MAE values signify a good performing model.

The average of the MSE and MAE values for each of the models for all drug parameters were taken. All the models had small average error values, the random forest model, however stood out, having on average the smallest average error for all the parameters but Terminal halflife. This is due to the robust nature of the ensemble tree method and it's resistance to biases and variance in data.

Statistical Results of the Regression Models

	Evaluation	Linear Regression	SVR	RF-regression
Volume of distribution	MSE	0.638	0.651	0.142
	MAE	0.107	0.488	0.261
	Average	0.325	0.569	0.202
Clearance	MSE	0.420	0.724	0.102
	MAE	0.742	0.656	0.215
	Average	0.581	0.690	0.159
Fraction unbound in plasma	MSE	0.279	0.302	0.249
	MAE	0.123	0.127	0.112
	Average	0.201	0.214	0.181
Mean residence time	MSE	0.567	0.605	0.282
	MAE	0.808	0.582	0.358
	Average	0.687	0.594	0.160
Terminal halflife	MSE	0.263	0.571	0.294
	MAE	0.333	0.577	0.352
	Average	0.298	0.574	0.323

Table 4.1: Table of Statistical Results of the Regression. Models

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

Predicting the PBPK parameters of chemical compounds in humans for new chemical compounds is essential for predicting drug dosage, as this describes the patterns and relationships between the parameters.
In this report;

1. We formulated one and two compartment models using systems of differential equations. The systems of differential equations were solved using the method of separation of variables.

2. The equations for three machine learning algorithms were formulated and then implemented using the Python programming language and the scikitlearn, pandas, seaborn and matplotlib libraries.

3. Machine learning models developed were able to predict the relationship between the PBPK parameters and our work provides evidence that PBPK parameters can be well predicted by machine learning algorithms.

5.2 Recommendation

Machine Learning algorithms are widely used for predictive analysis and the choice of the methods depends on factors such as problem to be addressed, availability of data and level of interpretability required. Our model was feasible in predicting the relationship between the parameters and thus we recommend future researchers in this field to use Machine Learning algorithms to predict PBPK parameters such as Area-under-Curve and Half-life, since;

1. It saves time: Total duration of machine learning model development for in-silico PBPK parameter prediction is shorter, as compared to the Compartment and PK models which take days.

2. It is cost-effective: PBPK model development requires less resources to develop, the models are applicable to a diverse range of drugs and are also easily reusable.

3. Easily interpretable: Visualisations and numeric values help establish the relationship between parameters. Machine learning algorithms are also applicable in handling compounds with non-linear relationships by dimension reduction algorithms such as Principal Component Analysis (PCA).

REFERENCES

[1] A. Ahmad, K. Graesboll, L. Christiansen, N. Toft, L. Matthews, and S. Nielsen, *Pharmacokinetic-pharmacodynamic model to evaluate intramuscular tetracycline treatment protocols to prevent antimicrobial resistance in pigs*, Antimicrobial agents and chemotherapy, 59 (2014).

[2] L. A. Bauer, *Chapter 2. Clinical Pharmacokinetic Equations and Calculations*, The McGraw-Hill Companies, New York, NY, 2008.

[3] L. Z. Benet, J. K. Sodhi, G. Makrygiorgos, and A. Mesbah, *Critical examination of clearance concepts reveals the potential for errors in clinical drug dosing decisions.*, The AAPS journal, AAPS J. 2021;23(3) (2021 May 10).

[4] G. M. Brenner and C. W. Stevens, *Pharmacology*, Elsevier Health Sciences, 2012.

[5] C. R. Cederroth, U. Albrecht, J. Bass, S. A. Brown, J. Dyhrfjeld-Johnsen, F. Gachon, C. B. Green, M. H. Hastings, C. Helfrich-Förster, J. B. Hogenesch, et al., *Medicine in the fourth dimension*, Cell metabolism, 30 (2019), pp. 238–250.

[6] W. C. Chou and Z. Lin, *Machine learning and artificial intelligence in physiologically based pharmacokinetic modelling*, Toxicological Sciences, (2022).

[7] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, O'Reilly Media Inc., Sebastopol, California, 3 ed., 2022.

[8] N. Gleichmann and M. Campbell, *What is adme?* https://www.technologynetworks.com/drug-discovery/articles/what-is-adme-336683, 2020.

[9] C. Hull, *Symbols for compartmental models*, BJA: British Journal of Anaesthesia, 51 (1979), pp. 815–817.

[10] S. Iwasaki, Y. Kosugi, A. Z. Zhu, S. Nakagawa, N. Sano, M. Funami, M. Kosaka, A. Furuta, H. Hirabayashi, and N. Amano, *Application of unbound liver-to-plasma concentration ratio to quantitative projection of cytochrome p450-mediated drug-drug interactions using physiologically based pharmacokinetic modelling approach*, Xenobiotica, 49 (2019), pp. 1251–1259.

[11] H. Jones and K. Rowland-Yeo, *Basic concpets in physiologically based pharmacokinetics modeling in drug discovery and development*, CPT: Pharmacometrics and Systems Pharmacology, 2 (2013), pp. 1–12.

[12] A. Krüger, V. Goncalves Maltarollo, C. Wrenger, and T. Kronenberger, *Adme profiling in drug discovery and a new path paved on silica*, Drug Discovery and Development - New Advances, (2019), pp. 1–30.

[13] S. P. Leelananda and S. Lindert, *Computational methods in drug discovery*, Beilstein journal of organic chemistry, 12 (2016), pp. 2694–2718.

[14] J. Lin, D. C. Sahakian, S. De Morais, J. J. Xu, R. J. Polzer, and S. M. Winter, *The role of absorption, distribution, metabolism, excretion and toxicity in drug discovery*, Current topics in medicinal chemistry, 3 (2003), pp. 1125–1154.

[15] F. Lombardo, G. Berellini, and R. S. Obach, *Trend analysis of a database of intravenous pharmacokinetic parameters in humans for 1352 drug compounds*, Drug Metabolism and Disposition, 46 (2018), pp. 1466–1477.

[16] A. R. Maharaj and A. N. Edginton, *Physiologically based pharmacokinetic modeling and simulation in pediatric drug development.*, pharmacometrics systems pharmacology, 3 (2014).

[17] H. Mamada, K. Iwamoto, Y. Nomura, and et al., *Predicting blood-to-plasma concentration ratios of drugs from chemical structures and volumes of distribution in humans.*, Molecular Diversity, 25 (2021), p. 1261â1270.

[18] L. Marcu and W. Harriss-Phillips, *In silico modelling of treatment-induced tumour cell kill: Developments and advances*, Computational and mathematical methods in medicine, 2012 (2012), p. 960256.

[19] S. Maxwell, *What is adme?*, 2020.

[20] R. Meesters, *Bioanalytical evaluation of dried plasma spot microsampling methodologies in pharmacokinetic studies applying acetaminophen as model drug*, Journal of Applied Bioanalysis, 2 (2016), pp. 25–37.

[21] J. E. Murphy, *Clinical pharmacokinetics*, Clinical Pharmacokinetics, 7 (2021).

[22] K. O. Pharma, *Instantaneous distribution model*, 2019.

[23] A. Pharmaceuticals, *Compartmental modeling in pharmacokinetics*. https://www.allucent.com/resources/blog/compartmental-modeling-pharmacokinetics, 2022.

[24] S. Prasad, *Compartment model*. https://youtu.be/bkkwXiLqtnM, 2020.

[25] S. Raschka and V. Mirjalili, *Python Machine Learning*, Packt Publishing Ltd., Livery Place, Birmingham, 3 ed., 2019.

[26] J. E. . Sager, J. Yu, I. Ragueneau-Majlessi, and N. Isoherranen, *Physiologically based pharmacokinetic (pbpk) modeling and simulation approaches: A systematic review of published models, applications, and model verification*, Drug Metabolism and Disposition, 43 (2015).

[27] M. K. Saini Anshul, *Analytis vidhya*, 2021.

[28] J. Tibbitts, D. Canter, R. Graff, A. Smith, and L. A. Khawli, *Key factors influencing adme properties of therapeutic proteins: A need for adme characterization in drug discovery and development*, in MAbs, vol. 8, Taylor and Francis, 2016, pp. 229–245.

[29] K. Wolff and I. P. Stolerman, *First-order elimination kinetics*, Encyclopedia of Psychopharmacology, (2013), pp. 536–536.

[30] Y. Yang, Y. Zhao, J. Duan, P. Zhao, L. Zhao, and X. Zhang, *Predictive biopharmaceutics and pharmacokinetics: Modeling and simulation*, Developing Solid Oral Dosage Forms (Second Edition), (2017), pp. 399–413.

Appendix

```
# import needed libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import plotly.express as px


# import excel file and display first two rows
pj = pd.read_excel("project1.xlsx")
pj.head(2)


# Drop compounds with molecular weight less than 900
pj = pj.loc[pj['MW'] <= 900]


# assigning our DataFrame a new name and dropping all but the
    needed columns
data = pj
data.drop(data.iloc[:, 8:], inplace = True, axis = 1)
data.drop(data.iloc[:, 1:3], inplace = True, axis = 1)
data.head()


# renaming our columns for easy analysis
data.rename(columns={
                    "human VDss (L/kg)" : "human_VDss_L/kg",
                    "human CL (mL/min/kg)" : "human_CL_mL/min/kg"
    ,
                    "fraction unbound \nin plasma (fu)" : "
    fraction_unbound_in_plasma_fu",
                    "MRT (h)" : "MRT_h",
                    "terminal  t1/2 (h)" : "terminal_halflife_h"
                    }, inplace = True)
data.head()
```

```
34
35
36 '''Exploratory Data Analysis: getting statistical insights and
       analysis on the data'''
37
38
39 # shape and info on our dataframe
40 print('Shape of DataFrame:', data.shape)
41 print('\n')
42 data.info()
43
44
45 # Compute the median for each column
46 df = data.drop(['Name'], axis=1)
47 medians = df.median()
48 print("Median for the Dataset:")
49 print(medians, '\n')
50 df.describe().T
51
52
53 # see total null values by column
54 data.isnull().sum()
55
56
57 ''' Data cleaning: handling null values, by removal or
       replacement.
58 The null are replaced with the median because they're a more
       accurate representation of
59 our data and aren't easily influenced by outliers, outliers may
       influence the mean values.'''
60
61
62 # Handling null values
63 human_vdss = data['human_VDss_L/kg'].median()
64 human_cl = data['human_CL_mL/min/kg'].median()
65 fraction_unbound_in_plasma_fu = data['
       fraction_unbound_in_plasma_fu'].median()
66 MRT_h = data['MRT_h'].median()
67 terminal_halflife = data ['terminal_halflife_h'].median()
```

```
68
69 # assigning the new values to our old columns
70 data['human_VDss_L/kg'].fillna(human_vdss, inplace= True)
71 data['human_CL_mL/min/kg'].fillna(human_cl, inplace= True)
72 data['fraction_unbound_in_plasma_fu'].fillna(
       fraction_unbound_in_plasma_fu, inplace= True)
73 data['MRT_h'].fillna(MRT_h, inplace= True)
74 data['terminal_halflife_h'].fillna(terminal_halflife, inplace=
       True)
75
76 # Checking to confirm null values replacement
77 data.isna().sum()
78
79
80 # Applying log tranformation to dataframe
81 data[["human_VDss_L/kg", "human_CL_mL/min/kg", "
       fraction_unbound_in_plasma_fu", "MRT_h", "terminal_halflife_h"
       ]] = data[["human_VDss_L/kg", "human_CL_mL/min/kg", "
       fraction_unbound_in_plasma_fu", "MRT_h", "terminal_halflife_h"
       ]].apply(np.log10)
82 data.describe()
83
84
85 # Plot histograms for each variable to show new distribution of
       data
86 data.hist(figsize = (20, 15), bins = 20)
87 plt.show()
88
89
90 # Create a scatter matrix using Plotly Express
91 attributes = ["human_VDss_L/kg", "human_CL_mL/min/kg", "
       fraction_unbound_in_plasma_fu", "MRT_h", "terminal_halflife_h"
       ]
92 fig = px.scatter_matrix(data, dimensions=attributes)
93 fig.update_layout(width=1200, height=1100)
94
95
96 # Correlation values for the data
97 data = data[["human_VDss_L/kg", "human_CL_mL/min/kg", "
```

```
    fraction_unbound_in_plasma_fu", "MRT_h", "terminal_halflife_h"
    ]]
data.corr()

# Correlation plot
corrmat = data.corr()
plt.figure(figsize=(8, 7))
hm = sns.heatmap(corrmat, annot=True, fmt=".4f", annot_kws={'size
    ': 9}, cmap="Blues")
# Add legend
legend = plt.legend(loc="upper right", fancybox=True, shadow=True
    , ncol=3)
legend.set_title("Correlation Coefficients")
for label in legend.get_texts():
    label.set_fontsize(11)
    label.set_ha("center")


# scaling the data so it ranges between 0 and 1
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(data[['human_VDss_L/kg', '
    human_CL_mL/min/kg', 'fraction_unbound_in_plasma_fu', 'MRT_h',
     'terminal_halflife_h']])
scaled_df = pd.DataFrame(scaled_data, columns=['human_VDss_L/kg',
     'human_CL_mL/min/kg', 'fraction_unbound_in_plasma_fu', 'MRT_h
    ', 'terminal_halflife_h'])
scaled_df.head()


# Importing libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.svm import LinearSVR
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error,
    mean_absolute_error

```

```
# Assigning independent and dependent variables

# Volume of Distribution
X = scaled_df[['human_CL_mL/min/kg', '
    fraction_unbound_in_plasma_fu', 'MRT_h', 'terminal_halflife_h'
    ]]
y = scaled_df[['human_VDss_L/kg']]

# Clearance
X = scaled_df[['human_VDss_L/kg', 'fraction_unbound_in_plasma_fu'
    , 'MRT_h', 'terminal_halflife_h']]
y = scaled_df[["human_CL_mL/min/kg"]]

# Fraction unbound in plasma
X = scaled_df[['human_CL_mL/min/kg', 'human_VDss_L/kg', 'MRT_h',
    'terminal_halflife_h']]
y = scaled_df[['fraction_unbound_in_plasma_fu']]

# Mean residence time
X = scaled_df[['human_CL_mL/min/kg', '
    fraction_unbound_in_plasma_fu', 'human_VDss_L/kg', '
    terminal_halflife_h']]
y = scaled_df[['MRT_h']]

# Terminal halflife
X = scaled_df[['human_CL_mL/min/kg', '
    fraction_unbound_in_plasma_fu', 'MRT_h', 'human_VDss_L/kg']]
y = scaled_df[['terminal_halflife_h']]


"""
Model pipeline for each machine learning algorithm is developed.
X_train represents the values of the independent columns to be
    used for model training
X_test represents the values of the independent columns to be
    used for model evaluation
y_train represents the labels of the training data.
y_test represents the labels of the testing data.
y_pred is the column for the new predicted values to be evaluated
```

```
"""


# Linear Regression model
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size = 0.3, random_state = 42)

lin_reg = LinearRegression()

lin_reg.fit(X_train, y_train)
y_pred = lin_reg.predict(X_test)


# Support Vector Regressor model
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size = 0.3, random_state = 42)

svm_reg = LinearSVR(C = 0.01, epsilon = 0.1)

svm_reg.fit(X_train, y_train)
y_pred = svm_reg.predict(X_test)


# Random Forest Regressor model
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size = 0.3)

rf = RandomForestRegressor(n_estimators = 100, max_depth = 5)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)


# Error evaluation
Mean_Absolute_Error = mean_absolute_error(y_test, y_pred)
Mean_Squared_Error = mean_squared_error(y_test, y_pred)
print("Mean Absolute Error:", Mean_Absolute_Error)
print("Mean Squared Error:", Mean_Squared_Error)

#the code works best in notebook environments.
```