Ind. J. Pharm. Edu. Res., 2026; 60: 10.5530/ijper.20265275
https://www.ijper.org

Original Article

Machine Learning Assisted Simultaneous Estimation of Drugs in Multicomponent Formulations by Spectrophotometry

Mariya Thomas[1], Kavitha M Parameswaran[1], Anjaly P Johny[1], Kannan Kamarajan[1,2,*], Krishnakumar Kunnambath[1,2]

[1]St. James College of Pharmaceutical Sciences, River Bank, Chalakudy, Kerala, INDIA.
[2]St. James Hospital Trust Pharmaceutical Research Centre, St. James College of Pharmaceutical Sciences, River Bank, Chalakudy, Kerala, INDIA.

ABSTRACT

Analysis of multi-drug formulations by spectrophotometric method is a challenging task due to mutual interference. Mathematical models reported for such analysis are applicable only where there are well separated absorption maxima of component drugs. In the present work Machine learning model has been developed for the simultaneous estimation of drugs in multi-drug pharmaceutical formulations using a custom made, comprehensive, interactive and user friendly software package, by spectrophotometry. Performance of the model was assessed by estimating the drugs in tablet formulation containing amlodipine besylate and losartan potassium. The accuracy, by recovery, was more than 98%. Intraday precision studies exhibited relative standard deviation of 1.67% for amlodipine besylate and 0.93% for losartan potassium where as in inter day precision studies the method exhibited relative standard deviation of 0.86% for amlodipine besylate and 0.77% for losartan potassium. The results indicated that machine learning model could be a promising tool for simultaneous estimation of drugs in multicomponent formulations by simple spectrophotometry.

Keywords: Machine learning, Spectrophotometry, Simultaneous estimation, Amlodipine, Losartan.

Correspondence:

Dr. Kannan Kamarajan

[1]St. James College of Pharmaceutical Sciences, River Bank, Chalakudy-680307, Kerala, INDIA.

[2]St. James Hospital Trust Pharmaceutical Research Centre, St. James College of Pharmaceutical Sciences, River Bank, Chalakudy-680307, Kerala, INDIA.

Email: drkannan.k@gmail.com

Received: 28-02-2025;
Revised: 06-05-2025;
Accepted: 15-08-2025.

INTRODUCTION

Drug development, manufacture and quality control needs a variety of samples to be analyzed for different purposes. The nature of the sample may vary from simple active compound to complex biological samples. The process becomes more complex when the Pharmaceutical formulation contains two or more drugs, due to mutual interference. Sensitive analytical methods like HPLC and hyphenated techniques are available for such analysis. However, these techniques are time consuming and expensive making it cumbersome for routine quality control tests. Simple spectrophotometric methods using hard modelling techniques like simultaneous equation method, differential spectrophotometric methods[1] can be used only when the absorption maxima of the sample components are well separated, which is not the case in many instances.

Recent developments in the field of Artificial Intelligence (AI) has made it possible to solve many scientific problems and it might be one of the effective tools for the simultaneous estimation of compounds, in a mixture, by simple spectrophotometry. Artificial Neural Network (ANN) is one such AI tools for solving both linear and non-linear problems including spectrophotometric analysis of multicomponent samples.[2-15]

Machine Learning (ML), a subset of AI,[16,17] is being used for solving scientific problems including recognition of diseases,[18] radiotherapy and radiology,[19] epidemic prevention,[20] drug discovery,[21,22] and in clinical trials and clinical Research.[23] Unlike ANN, ML has specific algorithms for solving linear problems[24] and more suitable for simultaneous spectrophotometric estimation of components in a mixture as there is linear relation between absorption and concentration. Hence in the present study comprehensive software has been developed in Python language to design ML based models, using Scikit-learn tools,[25] for simultaneous estimation of drugs in multi-drug dosage forms. The performance of the model was validated using tablet dosage form containing two drugs, Amlodipine besylate (AML) and Losartan potassium (LS), as representative multicomponent dosage form for analysis. Many spectrophotometric methods

DOI: 10.5530/ijper.20265275

Publishing Partner : Manuscript Technomedia. [www.mstechnomedia.com]

have been developed for the simultaneous estimation of AML and LS.[26-29] All these methods are based on hard modelling techniques like simultaneous equation method, which are bi-variate method based on measurement of absorbance at two wavelengths and not suitable when there is no well-defined peak as well as when there is excessive overlap of spectrum. There is a report on estimation of AML and LS using multivariate chemometrics, a soft modelling tool.[30] Here the authors have done the computation using chemometric tools of MATLAB. Generating training data and integrating them with main programme is complex when MATLAB platform is used. AI based analytical methods are multivariate approach, where the absorbance values in the entire UV range can be used for the estimation. Dedicated software exclusively for spectrophotometric analysis of multicomponent formulation, using freely available open source platform will help analyst to apply the method in their analytical work.

MATERIALS AND METHODS

Materials

Amlodipine besylate and Losartan potassium used were Indian Pharmacopoeial grade. Methanol and water used were HPLC grade. Tablet formulation was purchased in local market. The spectra were recorded in Shimadzu UV-1700 UV-visible spectrophotometer. The software package was developed in Python 3.8 platform using Scikit-learn 1.0.2 tools.

Methods

Software

Customized comprehensive software has been developed in python version 3.8 (Appendix: 1). Necessary additional libraries viz., tkinter, pandas, numpy, scipy, sklearn, pickle, datetime and random were installed, if it is not bundled in python, which are also available as free open source. Linear regression tool of Scikit-learn[25] is used to develop the model. The software can be used to generate training set containing required training pairs, design the user defined model, train the model using ML algorithm and predict the concentration of each drugs from the spectral data of samples, All the events during training and prediction, from file input and completion of training as well as prediction could be recorded and stored in text file automatically, for future reference.

Linearity

Standard stock solutions of AML and LS were prepared by dissolving 100 mg of AML or LS in 10 mL methanol and making up to hundred mL with water. Further dilutions were made in water to get concentrations ranging from, 2 to 50 mcg/mL. The absorbance of the solutions was measured at 360 nm for AML and at 247 nm for LS. From these data their linearity of response was determined.

Training

Standard solution of AML and solution of LS with concentration of 10mcg/mL were prepared from the standard stock solutions. The absorbance spectra of both the solutions were recorded between 200-400 nm using UV spectrophotometer. The spectral data were exported to ASCII format using export function of instrument software and saved as text file. When these data were fed into the programme, required number of UV data for various proportions of AML and LS would be generated and are divided into training set and test set. Using the training set, the model is trained and intermittently validated using test data set by iterations. The process continues till the prediction level comes to predefined accuracy. Six replicate trainings were conducted. Once training is over the model was saved as pickle file. Effect of number of training pairs (spectral data along with corresponding concentration of the drugs form one training pair) in training set and spectral range on the performance of the model were evaluated by changing one parameter at a time and keeping the other parameter constant. From the study optimum number of training pair required and suitable spectral range were determined.

Analysis of marketed formulation

The average weight of tablets was determined by weighing 5 tablets and powdered. Powder equivalent to 2.5 mg AML was weighed and dissolved in 10 mL methanol. The solution was sonicated for 15 min to dissolve the drug completely, the volume was made up to 100 mL with water and filtered. The filtrate was diluted suitably with water to keep the concentration of the drugs between 2 mcg/mL and 50 mcg/mL. The absorption spectra were recorded between 200 to 400 nm; the data was exported to ASCII format and saved as text file. Using the data in the text file, the concentrations were predicted using the trained model. Five replicate analyses were carried out.

The developed method was validated for accuracy and precision. Intraday precision was determined by replicate analysis of commercial tablet dosage form on the same day and inter-day precision was determined by repeating the analysis on six different days. Further ruggedness of analysis was determined by analyzing the tablet sample by two different analysts. Accuracy, by recovery studies, was carried out using standard addition method. The samples were prepared by spiking known amount of pure drug to the powdered marketed formulation and analyzed by the developed method, to determine the total drug content. The tablet powder equivalent to 2.5 mg AML and 25 mg LS was weighed accurately and were spiked with standard drug solutions at 50%, 100%, and 150% level in terms of drugs content in the formulation. The UV absorbance spectra were recorded from 200 to 400 nm. The concentrations were predicted using the trained model as described above. Three replicate analyses were carried out at each spike level.

RESULTS

Software has been developed in python version 3.8., using Scikit-learn tools to perform scientific functions which can perform all the functions required for designing the model, generating training set, training the model and predicting the concentration of drugs. The software can be used for the analysis of samples containing any number of drugs, by simple spectrophotometric method. However the performance of the model was evaluated by analysing tablet sample containing only two drugs, AML and LS, in the present study.

AML exhibited two peaks at about 237 nm and 360 nm where as LS exhibited no well defined peak in ultraviolet region (Figure 1). The absorbance values of both the drugs were linear between 2 mcg/mL and 50 mcg/mL at 237 nm.

The performance of the model was not affected by the number of training pairs, when kept between 25 to 250 pairs. The studies on the effect of spectral range on the performance of the model indicated that the model could perform better only when the data of entire UV range, that is 200 nm to 400 nm was used (Table 1). Under these conditions model could be trained within few seconds and the trained model could be stored. The trained model could be used to predict the concentration of AML and LS from the spectral data of sample. The marketed samples were analyzed using the trained model and found to have required precision (Table 2) and accuracy as determined by recovery study (Table 3).

DISCUSSION

Prediction of concentration of drugs in a multi-drug sample solution, from its UV absorption data, is a much easier task for powerful AI tools and ANN has been used for such analysis.[3-15] ANN can be used for solving both linear problems and nonlinear problems whereas ML algorithms specific for solving linear problems are available. Hence in the present work ML algorithm for multi input-multi output linear regression model of ML was used. The success of such method depends upon proper training of the model used, for which UV data of minimum of fifty sample solutions, containing both drugs in varying proportions, are needed. This involves preparation of fifty solutions containing standard drugs in varying proportions, recording their UV spectra, exporting the data to ASCII format to be saved in text format and using these data for training. Though it is tedious and time consuming, it is one time work and the model once developed can be used for any number of analyses. However such approach cannot take care of experimental variation at different situations and it could be more rugged and robust if both sample solutions and standard solutions are prepared simultaneously and analyzed. Hence in the present work training data were generated from standard solution, one for each drug, and the model was trained on daily basis along with sample analysis. This is possible as the training could be completed within few seconds, with presently available computer hardware and high speed processors, unlike two decades ago when it took hours to complete the training. Hence the developed method is as simple as spectrophotometric analysis of single drug formulations. In the present work comprehensive, interactive, customized and user friendly software has been developed in Python platform. Scikit-learn tools were used for ML modelling. All the required softwares are freely available as open source along with free tutorial videos and do not require a programming expert. Analyst with basic programming knowledge can develop the model and use it. The developed software package can be used to develop model for estimation of drugs in dosage form containing any number of drugs. In the present study tablet dosage form containing two drugs, AML and LS, has been chosen to validate the prediction

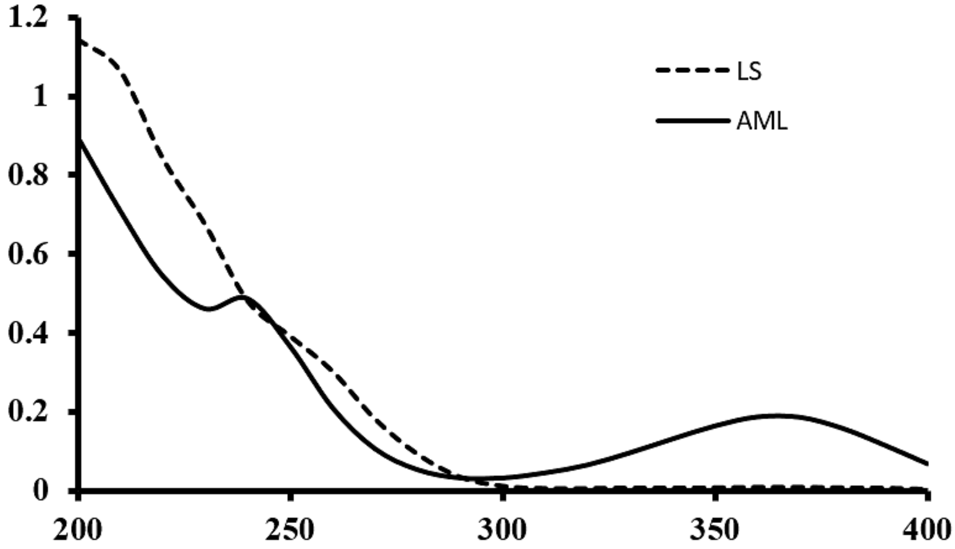

Figure 1: Overlain spectra of AML and LS.

Table 1: Effect of spectral range on the performance of the model.

Spectral range (nm)	Mean Concentration predicted (mcg/mL)		Mean Concentration Actual (mcg/mL)		Prediction error %	
	AML	LS	AML	LS	AML	LS
200-400	10.20	9.93	10	10	2.0	0.7
200-230	9.46	10.46	10	10	5.4	4.6
220-250	9.83	10.41	10	10	1.7	4.1
200-250	10.24	9.93	10	10	2.4	0.7
250-300	9.50	10.49	10	10	5.0	4.9

Table 2: Analysis of tablets: precision.

Inter-day precision			Intra-day precision		
Day	Predicted drug content (mg/tablet)		Replicate no.	Predicted drug content (mg/tablet)	
	AML	LS		AML	LS
1	4.95	50.43	1	4.95	49.5
2	4.925	50.55	2	4.93	50.65
3	5.025	49.83	3	5.03	49.65
4	4.92	50.45	4	4.83	50.25
5	4.97	49.72	5	5.03	49.9
Mean	4.96	50.20	Mean	4.95	49.99
SD	0.04	00.39	SD	0.08	0.47
%RSD	0.86	0.77	%RSD	1.67	0.93
Label Claim	5 mg/tablet	50 mg/tablet	Label Claim	5 mg/tablet	50 mg/tablet

ability of the developed model. AML has well defined peaks at about 237 nm and 360 nm where as LS has no well defined peak at Ultraviolet region (Figure 1) and these two drugs exist in 1:10 ratio in the tablet, making it difficult to analyse by usual multi-component spectrophotometric analytical methods.

AI models need to be trained sufficiently to get required accuracy in prediction. Overtraining should also be avoided as it leads to underperformance due to memorizing rather than learning. Number of training pairs in the training set is to be sufficient for proper training and at the same time should not lead to overtraining. Hence the performance of the model was evaluated by varying the numbers of training pairs between 25 to 200. The mean log square value was between 3.4759 e^{-31} to 2.5802 e^{-30}, indicating that there is no change in the performance of the model, when the number of training pair was kept between 25 to 250. Further studies were carried out using training set with 100 training pairs. Spectral range exhibited critical influence on the performance of the model and accurate results (<2% relative error) obtained only when the entire range that is 200 nm to 400 nm was used for training and prediction (Table 1).

The final model was based on Linear Regression algorithm of Scikit-learn, running on Python platform. The training set with 100 training pairs were used to train the model. The wavelength range used was 200 nm to 400 nm with 0.2 nm interval. For both the drugs, absorbance was linear between 2 to 50 mcg/mL and hence in all studies the concentration of the drugs in the final solution was kept within this range. Training set was generated from spectral data of standard solutions of AML and LS, keeping the concentration of 10 mcg/mL for both the solutions. The model was trained at above conditions using the generated training set, saved and used for predicting concentration of AML and LS in tablet sample solution.

Five replicate analyses were carried out on the same day to assess the method precision (Table 2). Inter day precision was assessed by analysing the tablet on five different days (Table 2). In the method precision studies the method exhibited percentage relative standard deviation of 1.67% for AML and 0.93% for LS. Where as in inter day precision the method exhibited percentage relative standard deviation of 0.86% for AML and 0.77% for LS. In both the studies the percentage relative standard deviation was less than the accepted limit of 2% indicating that the method has sufficient precision.

The accuracy of the method was evaluated by standard addition method at three different spike levels that is 50%, 100% and 150% of the expected drug content in the tablet sample. The percentage recovery was more than the accepted limit of 98% (Table 3) indicating that the method has required accuracy.

Table 3: Recovery studies to assess accuracy.

Drug	% added	% Recovery			Mean
		1	2	3	
AML	50	100.65	99.6	99.71	99.98
	100	100.83	101.1	99.85	100.59
	150	99.63	100.21	103.1	100.98
LS	50	101.2	99.96	100.71	100.62
	100	100.83	102.1	99.65	100.86
	150	99.1	100.21	100.12	99.81
Mean % recovery					100.47

CONCLUSION

The study indicated that ML is a powerful tool for the analysis of dosage forms containing more than one drug by simple spectrophotometric method without need for prior separation or a need for costly and time-consuming chromatographic methods. Availability of free open source software like Python and user friendly SciKit-Learn tools make it possible, for analysts with little programming skill, to use this technique for the analysis of multicomponent samples by spectrophotometry. Further, the study indicated that the method is simple and cost effective and has sufficient accuracy and precision to meet the regulatory requirements.

ACKNOWLEDGEMENT

Not applicable.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

ABBREVIATIONS

AML: Amlodipine besylate; **LS:** Losartan potassium; **AI:** Artificial intelligence; **ANN:** Artificial neural network; **ML:** Machine learning; **UV:** Ultra Violet; **mcg:** Microgram; **mL:** millilitre; **ASCII:** American Standard Code for Information Interchange.

SUMMARY

Comprehensive software has been developed in python programming language, using machine learning algorithm of Scikit-learn tools, for the analysis of multicomponent samples by spectrophotometry. The software can be used to design the model, generate training set from spectral data of standard solutions, train the model and predict the concentrations of analytes from the spectral data of sample solution. Performance of the model was evaluated by analyzing the tablet formulation containing amlodipine besylate and losartan potassium. The accuracy, by recovery, was more than 98% and the results of replicate analysis exhibited less than 2% relative standard deviation indicating the method has required accuracy and precision. The results indicated that machine learning model could be a promising tool for simultaneous estimation of drugs in multicomponent formulations by simple spectrophotometry.

REFERENCES

1. Víctor C, Piyawan P, Sergio F. TrAC Trends Anal Chem. 2022; 157: 116772.
2. Joselin. J, Dinesh. T, Ashiq.M. A Review on Neural Networks. IJTSRD. 2018; 2: 565-9.
3. Ni Y, Liu C, Kokot S. Simultaneous kinetic spectrophotometric determination of acetaminophen and phenobarbital by artificial neural networks and partial least squares. Anal Chim Acta. 2000; 419(2): 185-96. doi: 10.1016/S0003-2670(00)00978-8.
4. Balamurugan C, Aravindan C, Kannan K, Sathyanarayana D, Valliappan K, Manavalan R. Artificial neural network for the simultaneous estimation of multicomponent sample by UV-spectrophotometry. Indian J Pharm Sci. 2003; 65(3): 274-8.
5. Satyanarayana D, Kannan K, Manavalan R. Genetic algorithm optimized neural network ensemble as calibration Model for simultaneous spectrophotometric estimation of atenolol and losartan potassium in tablets. S. Arf. J Chem. 2006; 59: 3-11.
6. Satyanarayana D, Kannan K, Manavalan R. Chemometric Model for Simultaneous spectrophotometric Estimation of phenobarbitone and phenytoin sodium in Tablet Dosage using Back-Propagation Neural Network. Indian J Pharm Sci. 2006; 68(5): 625-31.
7. Satyanarayana D, Kannan K, Manavalan R. Genetic algorithm optimized neural networks ensemble for estimation of mefenamic acid and paracetamol in tablets. Acta Chim Slov. 2005; 52: 440-9.
8. Satyanarayana D, Kamarajan K, Rajappan M. Principal component artificial neural network calibration models for simultaneous spectrophotometric estimation of phenobarbitone and phenytoin sodium in tablets. Acta Chim Slov. 2005; 52: 138-44.
9. Sathyanarayana D, Kannan K, Manavalan R. Back propagation Neural Network Model for simultaneous spectrophotometric estimation of losartan potassium and hydrochlorothiazide in tablet. Indian J Pharm Sci. 2004; 66 (Nov-Dec):745-52.
10. Hamid RAH, Mahmoud RS. Artificial neural networks (ANN) for the simultaneous spectrophotometric determination of fluoxetine and sertraline in pharmaceutical formulations and biological fluid. Iran J Pharm Sci. 2017; 16(2): 478-89.
11. Abasi N, Sohrabi MR, Motiee F, Davallo M. The application of artificial neural network and least square support vector machine methods based on spectrophotometry method for the rapid simultaneous estimation of triamcinolone, neomycin, and nystatin in skin ointment formulation. Optik. 2021; 241: 167210. doi: 10.1016/j.ijleo.2021.167210.
12. Elkady EF. Simultaneous spectrophotometric determination of diclofenac potassium and methocarbamol in binary mixture using chemometric techniques and artificial neural networks. Drug Test Anal. 2011; 3(4): 228-33. doi: 10.1002/dta.216, PMID 21500367.
13. Mohammadpoor M, Kakhki RM, Assadi H. A bayesian regularized artificial neural network for simultaneous determination of loratadine, naproxen and diclofenac in wastewaters. Curr Pharm Anal. 2020; 16(8): 1083-92. doi: 10.2174/1573412915666190618123154.
14. Hassaninejad-Darzi S, Es'haghi Z, Nikou S, Torkamanzadeh M. Rapid and simultaneous determination of montelukast, fexofenadine and cetirizine using partial least squares and artificial neural networks modeling. Iran J Chem Chem Eng. 2017; 36(3): 81-96.
15. Darwish HW. Application of smart spectrophotometric methods and artificial neural network for the simultaneous quantitation of olmesartan medoxamil, amlodipine besylate and hydrochlorothiazide in their combined pharmaceutical dosage form. Chem Cent J. 2013; 7(1): 22. doi: 10.1186/1752-153X-7-22, PMID 23374392.
16. Gladchuk V. A history of Machine Learning from. Vol. 2021; 1940s to present days. Labelyourdata.com. Available from: https://labelyourdata.com/articles/history-of-machine-learning-how-did-it-all-start.

17. ***17***. What is machine learning: definition, types, applications and examples. Potentia analytics; 2021. Available from: https://www.potentiaco.com/what-is-machine-learning-definition-types-applications-and-examples/.
18. Alsheref FK, Hassan W. Blood diseases detection using classical machine learning algorithms. IJACSA. 2019; 10(7). doi: 10.14569/IJACSA.2019.0100712.
19. James BA, Alexandre T, Amish PS, Derek WJ, Phillip WP, Jose CP, *et al.* Linear regression model for predicting patient-specific total skeletal spongiosa volume for use in molecular radiotherapy dosimetry. The Journal of Nuclear Medicine. 2006; 47(7): 1875-1883.
20. Shashank PST, Vaibhavi A, Vaishnavi J, Jabbar MA. Regression Model for prediction of epidemic outbreaks. IOP Conf S Mater Sci Eng. 2021; 1042(1): 012016. doi: 10.1088/1757-899X/1042/1/012016.
21. Patel L, Shukla T, Huang X, Ussery DW, Wang S. Machine learning methods in drug discovery. Molecules. 2020; 25(22): 5277. doi: 10.3390/molecules25225277, PMID 33198233.
22. Lo YC, Rensi SE, Torng W, Altman RB. Machine learning in chemoinformatics and drug discovery. Drug Discov Today. 2018; 23(8): 1538-46. doi: 10.1016/j.drudis.2018.05.010, PMID 29750902.
23. Zame WR, Bica I, Shen C, Curth A, Lee HS, Bailey S *et al.* Machine learning for clinical trials in the era of COVID-19. Stat Biopharm Res. 2020; 12(4): 506-17. doi: 10.1080/19466315.2020.1797867, PMID 34191983.
24. Brownlee J. Linear regression for machine learning. Machine learning mastery; 2021. Available from: https://machinelearningmastery.com/linear-regression-for-machine-learning/.
25. Fabian P, Gaël V, Alexandre G, Vincent M, Bertrand T, Olivier G, *et al.* Scikit-learn: machine learning in python, JMLR. 2011; 12(85): 2825-30.
26. S B Wankhede, K C Raka, S B Wadkar, S S Chitlange, Spectrophotometric and HPLC Methods for Simultaneous Estimation of Amlodipine Besilate, Losartan Potassium and Hydrochlorothiazide in Tablets, Indian J Pharm Sci. 2010; 72(1): 136-40.
27. Ramya Gavini, S. B. Puranik, G. V. S. Kumar, K. A. Sridhar and Ramya Gavini, Simultaneous estimation of amlodipine and losartan by UV-method in bulk drug and tablet dosage formulation, Arch. Appl. Sci. Res., 2012; 4(5): 2206-12
28. Priyanka, R. P., U. R. Sachin, P. N. Dhabale, and K. B. Burade. "Simultaneous UV Spectrophotometric method for estimation of losartan potssium and amlodipine besylate in tablet dosage form." Asian J. Research Chem, 2009; 2(1): 183-7.
29. Manisha Masih, Abhilasha Mittal, Nandy BC, Spectrophotometric simultaneous estimation of amlodipine besylate and losartan potassium in tablet dosage forms, Asian J Pharm Clin Res, 2014; 7(4): 61-3
30. D. Nagavalli, V. Vaidhyalingam, A. Santha, A. S. K.Sankar and O. Divya, Simultaneous spectrophotometric determination of losartan potassium, amlodipine besilate and hydrochlorothiazide in pharmaceuticals by chemometric methods, Acta Pharm. 2010; 60: 141-52.

Cite this article: Mariya T, Kavitha MP, Anjali PJ, Kannan K, Krishnakumar K, Machine Learning Assisted Simultaneous Estimation of Drugs in Multicomponent Formulations by Spectrophotometry. Indian J of Pharmaceutical Education and Research. 10.5530/ijper.20265275.

Appendix 1

Source code of the developed software

```
#Import required libraries
import tkinter as tk
import pandas as pd
import numpy as np
import scipy as si
import sklearn
import pickle
import datetime
import random
# Import necessary modules
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn import metrics
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
from datetime import datetime
from tkinter import filedialog
# Decide whether this session is to train a new netwrok or to predict concentration using a trained network
mode=int(input("Enter 1 for Training a New REG""\n"" OR""\n""Enter 2 for Prediction Using a Trained REG :"))
if mode==1:
#Open a text file to record the details of training session for future reference and enter date and time
Filename=str(input("Name the File to Save REG Training Details: "))
f=open(Filename,"w+")
txt="REG TRAINING DETAILS"
xa=txt.center(30)
print(xa,file=f)
def main():
now=datetime.now()
today=now.strftime("%d/%m/%Y, %H:%M:%S")
print("\nDate and Time: ",today,file=f)
f.close()
main()
#Get the information on the spectral range to be used for analysis, number of srugs/analytes in the sample and sample size required for training the network.
f= open(Filename,"a+")
number=int(input("Enter the Number of Drugs in the Sample: "))
print("\nNumber of Drugs in the Sample: ",number,file=f)
Start_nm=float(input("Enter start wavelength (between 200nm to 400nm: "))
End_nm=float(input("Enter stop wavelength (between 200nm to 400nm: "))
print("\nWavelength Range_nm: ",Start_nm," To ", End_nm,file=f)
size=int(input("Enter sample size required in training set: "))
print("\nSample Size Required: ",size,file=f)
# Get information about the drug_1
name=str(input("Enter name of drug_1: "))
print("\nName of Drug_1: ",name,file=f)
conc1=float(input("concentarion of " + name + " std solution _ mcg per mL: "))
print("\nConcentarion of " + name + " Standard_mcg per mL: ", conc1,file=f)
min1=float(input("minimum expected concentration of " + name + " -mcg per mL: "))
print("\nMinimum Expected Concentration of" + name + "_mcg per mL: ",min1,file=f)
max1=float(input("maximum expected concentration of " + name + " -mcg per mL: "))
print("\nMaximum Expected Concentration of" + name + "_mcg per mL: ",max1,file=f)
# import the text file contaning UV data of drug_1 and make the programme to read the file and store the data as data frame
root= tk.Tk()
root.title("REG_Analyzer")
canvas1=tk.Canvas(root, width=300, height=300, bg='lightsteelblue')
canvas1.pack()
def getText ():
global df
```

```
import_file_path=filedialog.askopenfilename()
df=pd.read_table (import_file_path)
browseButton_Text=tk.Button(text='Import '+ name + ' UV data_file', command=getText, bg='green', fg='white', font=('helvetica', 12, 'bold'))
canvas1.create_window(150, 150, window=browseButton_Text)
root.mainloop()
#slice the required spectral range from the full range UV data of drug_1
df=df[df.iloc[:, 0].between(Start_nm, End_nm, inclusive=”both”)]
print("\n\nUV data of " +name, file=f)
print(df, file=f)
array=df.values
# slice abosrbance values omitting the wavelength from the data frame
x=array[ :, -1 ]
# Generate required number og uv spectral data of drug_1 at different concentration from the UV data of standard by generating random numbers as conc and calculating aborbance values for that concentration
input_array=[]
target_array=[]
a=1
while a <= size:
b=round(random.uniform(min1, max1), 1)
c=[i*b/conc1 for i in x]
input_array.append(c)
target_array.append(b)
a=a+1
input_data=np.array(input_array)
target_data=np.array(target_array)
print("\n\nInput_Data of"+name, file=f)
print(input_data, file=f)
print("array shape: ",input_data.shape, file=f)
print("\n\nTarget_Data of "+name, file=f)
print (target_data, file=f)
print("array shape: ",target_data.shape, file=f)
d=2
while d <= number:
d_1=str(d)
# Get information about subsequent drugs one by one in cycle
name=input("Enter the name of drug_"+d_1+": ")
print("\n\nName of Drug_"+d_1+": ",name,file=f)
conc=float(input("concentarion of " + name + " std solution _mcg per mL: "))
print("Concentarion of " + name +" Standard_mcg per mL: ", conc,file=f)
min2=float(input("minimum expected concentration of " + name + " -mcg per mL: "))
print("Minimum Expected Concentration of "+name+"_mcg per mL: ",min2,file=f)
max2=float(input("maximum expected concentration of " + name + " -mcg per mL: "))
print("Maximum Expected Concentration of "+name+"_mcg per mL: ",max2,file=f)
root= tk.Tk()
root.title("REG_Analyzer")
canvas1=tk.Canvas(root, width=300, height=300, bg='lightsteelblue')
canvas1.pack()
# import the text file contaning UV data of subsequent drugs one by one in cycle and make thw programme to read the file and store the data as data frame
def getText ():
global df
import_file_path=filedialog.askopenfilename()
df=pd.read_table (import_file_path)
browseButton_Text=tk.Button(text='Import '+ name + ' UV data_file', command=getText, bg='green', fg='white', font=('helvetica', 12, 'bold'))
canvas1.create_window(150, 150, window=browseButton_Text)
root.mainloop()
#slice the required spectral range from the full range UV data of drug_1
df=df[df.iloc[:, 0].between(Start_nm, End_nm, inclusive=”both”)]
print("\n\nUV data of " + name, file=f)
print(df, file=f)
array=df.values
#print("UV data of " +name)
#np.savetxt(f, df, fmt="%2.3f", delimiter=',', newline='\n')
# slice abosrbance values omitting the wavelength from the data frame
x=array[ :, -1 ]
#print(x)
#UV_data_drug1=np.transpose(x)
#print(UV_data_drug1)
i_array=[ ]t_array=[ ]
# Generate required number of UV spectral data of drug_1 at different concentration from the UV data of standard by generating random numbers as conc and calculating aborbance values for that concentration
a=1
while a <= size:
b=round(random.uniform(min2, max2), 1)
c=[i*b/conc1 for i in x]
i_array.append(c)
t_array.append(b)
a=a+1
#Limit decimal point to 5
input_array=np.around(i_array,decimals=5)
target_array=np.around(t_array,decimals=5)
print("\n\nInput_Data of "+name, file=f)
print(input_array, file=f)
print("array shape: ",input_array.shape, file=f)
print("\n\nTarget_Data of "+name, file=f)
print(target_array, file=f)
print("array shape: ",target_array.shape, file=f)
e=np.array(input_array)
g=np.array(target_array)
#append target data of subsequent drugs one by one in each cycle to the target data of drug_1
target_data=np.vstack ((target_data,g) )
#Add input data of all the drugs one by one in each cycle to the input data of drug_1 to give training data
input_data=[[input_data[i][j] + e[i][j] for j in range
(len(input_data[0]))] for i in range(len(input_data))]
input_data=np.array(input_data)
d=d+1
print("\nInput_Data for Training \n", file=f)
print(input_data, file=f)
print("array shape: ",input_data.shape, file=f)
target_data=np.transpose(target_data)
print("\nTarget_Data for Training\n", file=f)
print(target_data, file=f)
print("array shape: ",target_data.shape, file=f)
# normalize the input data between 0-1
X=input_data
Y=target_data
scaler=MinMaxScaler(feature_range=(0, 1))
rescaledX=scaler.fit_transform(X)
np.set_printoptions(precision=3)
#print(rescaledX[0:5,:])
X_train, X_test, Y_train, Y_test=train_test_split(X, Y, test_size=0.2, random_state=42)
print(X_train.shape, X_test.shape, Y_train.shape, Y_test.shape,file=f)
#Train the model
model=LinearRegression()
reg=model.fit(X_train, Y_train)
#trained model is stored in the Python object reg
model.predict(X_test[ :, : ])
model.score(X_test, Y_test)
expected_Y =Y_test
predicted_Y=model.predict(X_test)
print ("\n\nmodel.score",model.score,file=f)
print("\n\nr2_score:",metrics.r2_score(expected_Y, predicted_Y),file=f)
print("\n\nmetrics.mean_squared_log_error:",metrics.mean_squared_log_error(expected_Y, predicted_Y),file=f)
#save the model"Called pickle the model" to disk as pickle file
Pkl_Filename=str(input("Name the file to save trained model: "))
```

```
with open(Pkl_Filename, 'wb') as f:
pickle.dump(reg, f)
print ("model saved_as:"+ Pkl_Filename)
f.close()
f= open(Filename,"a+")
print ("\n\n model saved_as:"+ Pkl_Filename, file=f)
txt = " _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ END__________________________"
d=txt.center(40)
print(d,file=f)
f.close()
#prediction mode
else:
f= open("result.txt","a+")
def main():
now=datetime.now()
now=now.strftime("%d/%m/%Y, %H:%M:%S")
print("\nDate and Time: ",now,file=f)
main()
f.close()
# import saved model pickle file as read only mode and in binary format
f= open("result.txt","a+")
root= tk.Tk()
root.title("REG_Analyzer")
canvas1=tk.Canvas(root, width=500, height=300, bg='lightsteelblue')
canvas1.pack()
welcome=tk.Label(text="REG_Analyzer WELCOMES YOU", bg='green', fg='white', font=('helvetica', 14, 'bold'))
welcome.place(relx=0.5, rely=0.1, anchor='center')
def getModel ():
global reg_model
model_file_path=filedialog.askopenfilename()
print("\nModel File : " + model_file_path, file=f)
with open(model_file_path, 'rb') as model:
reg_model=pickle.load(model)
```

```
browseButton_model=tk.Button(text='Import saved model_file', command=getModel, bg='green', fg='white', font=('helvetica', 14, 'bold'))
browseButton_model.place(relx=0.5, rely=0.5, anchor="center")
root.mainloop()
window= tk.Tk()
window.title("REG_Analyzer")
canvas1=tk.Canvas(window, width=1000, height=300, bg='lightsteelblue')
canvas1.pack()
warning=tk.Label(text="Important:\n\n" + "The spectral wave length Range should be same for Training the network and for prediction.\n\n" + "For Spectral Range Used in Training: Refer 'REG Training Detail file' created during training this REG", bg='green', fg='white', font=('helvetica', 14, 'bold'))
warning.place(relx=0.5,rely=0.4, anchor='center')
accept=tk.Button(window,text="OK",bg='green', fg='white', font=('helvetica', 14, 'bold'), command=window.destroy)
accept.place (relx=0.5,rely=0.9, anchor='center')
window.mainloop()
Start_nm=float(input("Enter start wavelength in nm (Same as the one used to train this REG): "))
End_nm=float(input("Enter stop wavelength in nm(Same as the one used to train this REG): "))
print("\nSpectral Range Selected: ",Start_nm,"nm to ", End_nm,"nm", file=f)
repeat="y"
while repeat == "y":
f= open("result.txt","a+")
#main()
#import text file containing UV data of sample and make the programme to read the file and store the data as data frame
root= tk.Tk()
root.title("REG_Analyzer")
canvas1=tk.Canvas(root, width=300, height=300, bg='lightsteelblue')
canvas1.pack()
def getText ():
global df1import_file_path=filedialog.askopenfilename()
print("\n\nSample UV Data File: ",import_file_path, file=f)
```

```
df1=pd.read_table (import_file_path)
browseButton_Text=tk.Button(text='Import Sample UV data_file', command=getText, bg='green', fg='white', font=('helvetica', 12, 'bold'))
canvas1.create_window(150, 150, window=browseButton_Text)
root.mainloop()
# get the spectral range for analysis which should be same as that used for training this network
df1=df1[df1.iloc[:, 0].between(Start_nm, End_nm, inclusive=”both”)]
array_sample=df1.values
array_sample=np.transpose(array_sample)
#Predict concentration
reg_model.predict(array_sample)
result=np.delete(reg_model.predict(array_sample),0,0)
#main()
print("\nDRUG1_CONC DRUG2_CONC mcg per mL")
print (result)
#print('\n"DRUG1_CONC DRUG2_CONC mcg per mL"\n',file=f)
#print (result, file=f)
# write the result in to file named "result"
f.write('\n"DRUG1_CONC_mcg per mL DRUG2_CONC mcg per mL"\n\n')
np.savetxt(f,result, fmt="%2.3f", delimiter=' ', newline='\n')
txt="\n\n________________________END____________________________"
d=txt.center(40)
print(d,file=f)
f.close()
req=str(input("\nDo you have data of another sample for prediction(y/n): "))
repeat=(req)
```