https://doi.org/10.15255/KUI.2021.008
KUI-2/2022
Original scientific paper
Received February 1, 2021
Accepted July 17, 2021

Cumulative Drug Release Modelling of PCL-PVP Encapsulated Tramadol by DA-SVM, MLR, PLS, and OLS Regression Techniques

A. Chabane,[a] *F. Bouchal*,[a] *M. Hentabli*,[b*] *F. Rezgui*,[a] and *H. E. Slama*[c]

[a] Laboratoire des Matériaux Organiques (LMO), Département de Génie des Procédés, Faculté de Technologie, Université de Bejaia, 06000 Bejaia, Algeria
[b] Laboratory of Biomaterials and Transport Phenomena (LBMPT), Faculty of Technology, University Yahia Fares of Médéa 26 000, Médéa, Algeria
[c] Quality Control Laboratory, SAIDAL-Médéa 26 000, Médéa, Algeria

Abstract

This work aimed to model the kinetics of cumulative drug release from formulations based on encapsulation by biodegradable polycaprolactone and polyvinylpyrrolidone polymers. Different ratios of the polymerswere prepared by a solvent evaporation method using Span 20 and Span 80 as surfactants. The cumulative drug release was estimated depending on the formulation component and time. Four models: hybrid model of support vector machine and dragonfly algorithm (DA-SVM), partial least squares (PLS) model, multiple linear regression (MLR) model, and ordinary least squared (OLS) model, were developed and compared. The statistical analysis proved there were no issues in variable inputs. The results showed that the DA-SVM model gave a better result where a determination coefficient was close to one and RMSE error close to zero. A graphical interface was built to calculate the cumulative drug release.

Keywords

Dragonfly algorithm, support vector machine, Tramadol, cumulative drug releases, modelling, biopolymer, least squares

1 Introduction

Anyone who has ever felt a violent pain will agree that anything that could reduce it is more than welcome. In view of that fact, we tend to consider opiates as healthcare products, and their daily use in medicine confirms the correctness of this opinion.[1] Tramadol is a central synthetic opioid, categorized by World Health Organization as a step II analgesic. It is prescribed in the treatment of both acute and chronic pain.[2] It is an analogue of codeine action-wise, and an agonist of morphine receptors. The different therapeutic presentations of the molecule make its clinical use easy both entirely and parenterally, as well as locoregional.[3]

Bioencapsulation is among pharmaceutical processes used by large drug companies, which, according to a determined process, consists of enclosing, active molecules whether chemical or organic, within other inactive materials, in the minimum order to improve the properties of its preservation, presentation, and bioavailability.[4–6]

A multivariate linear model is a statistical model in which one seeks to express a random variable as a function of explanatory variables in the form of a linear operator on the unknown parameters of the model. Linear adjustment is the approximation operation allowing choosing the best possible hyperplane. In this work, several linear methods were used, like partial least squares (PLS) regression, multiple linear regression (MLR) model, and ordinary least squared (OLS) regression.

Wide margin support vector machines are a set of learning techniques designed to solve the discrimination and regression problems.[7–11] Support vector machines are widely utilised in many fields (bioinformatics, information research, computer vision, pharmaceuticals, *etc.*), mainly due to their fast training, better accuracy, and robustness. In this paper, we attempted to test how an support vector machine performs when applied in the bioencapsulation of Tramadol by polycaprolactone (PCL) and polyvinylpyrrolidone (PVP) polymers.

2 Methods

2.1 Support vector machines

Support vector machines (SVM) methodology is based on statistical learning theory that was first introduced by *Vapnik*.[12–14] The SVM model architecture has never been predetermined. The output estimation formula of the aforementioned model is given as follows:

$$f(x) = w \cdot \varnothing(x) + b \tag{1}$$

where w is a weight vector, b is a bias denoting the dot product, and $\varnothing$ is the nonlinear transfer function that maps the input vectors into a high-dimensional feature space in which, theoretically, a simple linear regression can cope with the complex nonlinear regression of the input space.

For the purpose of solving the Eqs. (2) and (3), *Vapnik* introduced the following convex optimization problem with a ε-insensitivity loss function:[15]

* Corresponding author: Mohamed Hentabli, PhD student
Email: hentm92@hotmail.fr

$$\text{minimize}: \frac{1}{2}w^2 + C\sum_{K=1}^{N}\left(\xi_k^+ + \xi_k^-\right) \tag{2}$$

$$\text{subject to}\begin{cases} y_k - \left(w\cdot\varnothing\left(x_k\right)+b\right) \le \varepsilon + \xi_k^+ \\ -y_k + \left(w\cdot\varnothing\left(x_k\right)+b\right) \le \varepsilon + \xi_k^- \quad k = 1,2,\ldots\ldots,N \\ \xi_k^+, \xi_k^- \ge 0 \end{cases} \tag{3}$$

ξ_k^+ and ξ_k^- are slack variables that penalize training errors by the loss function over the error tolerance (ε), while C is a positive trade-off parameter (*i.e.*, capacity parameter) that determines the degree of the empirical error in the optimisation problem, and also dictates the trade-off between the flatness of the function and the amount at which deviations larger than ε are tolerated.

Eqs. (4) and (5) are solved using Lagrangian multipliers and the Karush-Kuhn-Tucker (KKT) optimality condition as follows:[16]

$$f\left(x,\alpha_i,\alpha_i^*\right) = \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right)\cdot K\left(x,x_i\right) + b \tag{4}$$

where α_i and α_i^* are Lagrangian multipliers, K is a kernel function defined by an inner product of the nonlinear kernel functions[15] with σ as the Gaussian kernel function parameter.

Linear $$K\left(x_n,x\right) = x_n^{'}x \tag{5}$$

Gaussian $$K\left(x_n,x\right) = e^{\frac{-x_n^{'}-x^2}{\sigma}} \tag{6}$$

Polynomial $$K\left(x_n,x\right) = \left(1+x_n^{'}x\right)^n \tag{7}$$

2.2 Dragonfly algorithm

The dragonfly method belongs to metaheuristic methods. It is an optimisation algorithm aimed at solving difficult optimisation problems. Dragonfly approach was proposed by *Mirjalili et al.*[20] The latter summarised the principles leading any swarm behaviour as:

- *Separation principle,* which represents the constant collision avoidance between individuals in a population.
- *Cohesion principle,* which is based on a tendency of individuals for arrangement towards the mass midpoint of the neighbourhood.
- *Alignment principle,* which represents the identical velocity of dragonfly's members existing in the same group.

2.3 SVM optimisation with dragonfly algorithm

Dragonfly is the proposed method for the optimisation of the hyperplane parameters.[19] Schematic representation of methodology of the applied hybrid of support vector machine and dragonfly algorithm (DA-SVM) can be drawn in accordance with the flowchart in Fig. 1.[10,17–19]

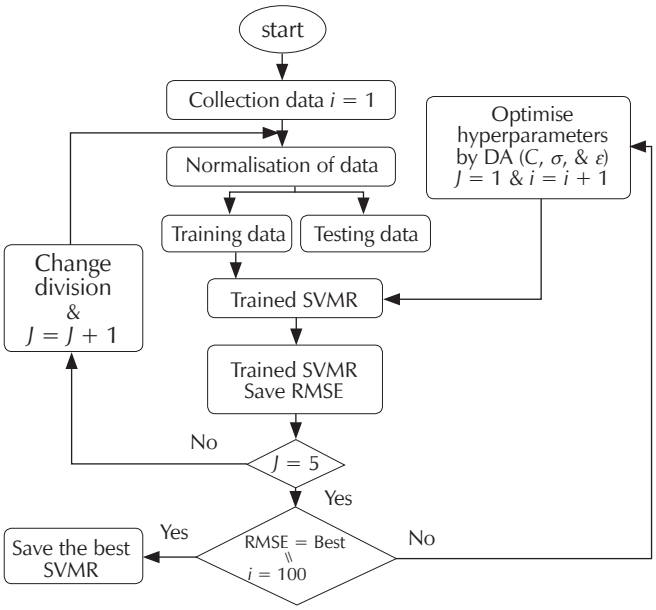

Fig. 1 – DA-SVM technique schematic[10,17–19]

Although the principle of SVM remains unchanged, the DA mainly feeds the SVM with a random set of hyper-parameters in their predefined ranges. In five iterations, the steps starting at the data division and moving up the model chart are repeated, and the minimum root mean square error (RMSE) value collected during the process is saved as the best value. The dragonfly algorithm then produces a new population of hyper-parameters that are meant to be fed into the SVM algorithm, and the same set of steps are run again for obtaining a new best RMSE. The whole operation is executed for 100 trials, among which the minimum RMSE corresponds to the resulting optimal DA-SVM model.

Numeric values in the input data matrix are normalised to improve optimisation and speed convergence. The normalisation function is expressed as follows:

$$X_{in} = {}^{1/0.1}\sqrt{X_i} \tag{8}$$

To assess the predictive power of the DA-SVM model, a root mean square error (RMSE) and a coefficient of determination (R^2) were used as evaluation criteria.[7,9,11,21]

2.4 Partial least squares

Partial least squares (PLS) regression is a statistical method that transforms matrix of independent variables X into dependent variable Y. In our case, X is an [*n_m*] matrix of reflectance,[8] where *n* is the number of inputs, *m* is the number of observations, and Y is the matrix containing cumulative drug release values. PLS regression decomposes X and Y by projecting them in new directions with the restriction that the decomposition describes how the variables change together as much as possible. After variable decomposition, a regression step is performed in which the

decomposed X and Y are used to calculate a regression model called complete model Eq. (9).

$$\begin{pmatrix} X_{1,1} & \dots & X_{1m} \\ \vdots & \ddots & \vdots \\ X_{n,1} & \dots & X_{nm} \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} Y_1 \\ \vdots \\ Y_n \end{pmatrix} + e \tag{9}$$

Eq. (3) can be written as Eq. (10):

$$Y = a_0 + a_1 X_1 + \dots + a_n X_n + e \tag{10}$$

where a_0 refers for *Y*-intercept, X_1 to X_n stand for the independent variables, a_1 to a_n are the coefficients of independent variables, *e* refers to the error term, and *Y* is the dependent variable.

2.5 Multiple linear regression

The regression analysis is frequently used for prediction. The objective of this model is to construct a mathematical model that can be utilised to predict the dependent variable based on the inputs of independent variables or the predictors.[8,22] Multiple linear regression (MLR)model has been used to obtain the significant relationship as well as correlation between the input variable and the output. Here, 80 % of data were used for training, and 20 % data applied for testing of this model. MLR model follows in Eq. (10).

2.6 Ordinary least squares

The ordinary least squares (OLS) method is a type of linear least squares method for estimating the unknown parameters in a linear regression model.

OLS chooses the parameters of a linear function of a set of explanatory variables by the principle of least squares: minimising the sum of the squares of the differences between the observed dependent variable (values of the variable being observed) in the given dataset and those predicted by the linear function.

OLS is seen as the sum of the squared distances, parallel to the axis of the dependent variable, between each data point in the set and the corresponding point on the regression surface – the smaller the differences, the better the model fits the data. The resulting estimator can be expressed by a simple formula, especially in the case of a simple linear regression, in which there is a single regressor on the right side of the regression equation.

The OLS was chosen because it generates the output feature class and optional tables with diagnostics and coefficient information which can be easily interpreted. The following phases were involved to carry out the study.

3 Data analysis

The modelling data were gathered from the literature[2] (*i.e.*, 12 formulation × 12 samples over time = 144 points). Table 1 represents the data composition, such as polycaprolactone (PCL) and polyvinylpyrrolidone (PVP) polymers, and span 20 and span 80 surfactants. Data are divided into four factors PVP (20, 15, 10.5) % (w/v), PCL (80, 85, 90, 95) % (w/v), Span 20 (1.2) % (w/v), and Span 80 (1.2) % (w/v), to obtain 12 kinetics.

Table 2 is a brief representation of dependent and independent variables, their ranges, unit of measure, standard deviation, variance, and Kurtosis.

Table 1 – Formulation conditions

Formulation	Number	1	2	3	4	5	6	7	8	9	10	11	12
surfactant / % (w/v)	Span 20	2	2	2	2	0	0	0	0	1	1	1	1
	Span 80	0	0	0	0	2	2	2	2	1	1	1	1
polymer / % (w/v)	PCL	80	85	90	95	80	85	90	95	80	85	90	95
	PVP	20	15	10	5	20	15	10	5	20	15	10	5

Table 2 – Statistical analysis

		Abs	Unit	Domain		STD	Variance	KURTOSIS
Input	Polycaprolactone	PCL	% (w/v)	80.000	95.000	5.608	31.452	1.640
	Polyvinylpyrrolidone	PVP		5.000	20.000	5.608	31.452	1.640
	Span-20	Span20		0	2.000	0.819	0.671	1.500
	Span-80	Span80		0	2.000	0.819	0.671	1.500
	Time	*t*	h	0	12.000	3.754	14.090	1.786
Output	Cumulative drug release	CD	%	0.000	90.066	23.289	542.362	2.505

3.1 Probability density

A kernel smoother estimate is a nonparametric representation of the probability density function (PDF) of a random variable. Kernel distribution used for describing data is defined by a smoothing function and a bandwidth value controlling the smoothing of the resulting density curve.

A kernel density estimate is the assumed PDF of a random variable. For any real value of x, kernel density estimation formula can be expressed as Eq. (11). Fig. 2 represents the uniform data distribution.

$$\hat{f}_h(x) = \frac{1}{nh}\sum_{j=1}^{n} k\left(\frac{x - x_i}{h}\right) \tag{11}$$

where $x_1, x_2, \ldots, x_n$ are random samples of an unknown distribution, n is the sample size, k is the kernel smoothing function, and h is the bandwidth.

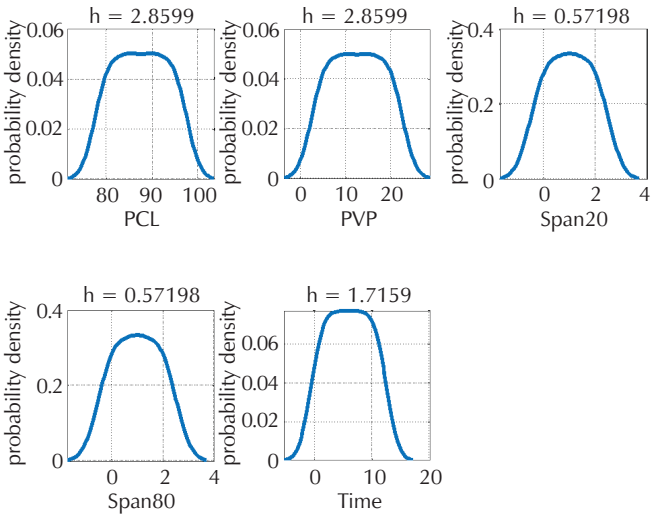

Fig. 2 – Probability density of input variables

3.2 Multicollinearity

Multicollinearity is an issue that occurs in a regression, and that is when certain predictor variables in the model measure the same phenomenon. Strong multicollinearity is problematic because it can increase the variance in the regression coefficients and make them unstable and difficult to construe. Fig. 3 represents the multicollinearity analysis.

4 Results and discussion

In the current study, the data set was divided into two sub data sets, one for learning and the other for validation. The validation part was employed by the cross-validation technique (Holdout with 20 %), and defined a random partition on a data-set. This partition is used to define training and test sets to validate a statistical model using cross-validation. This method was applied to all modelling techniques employed in this study (PLS, MLR, OLS, and DA-SVM).

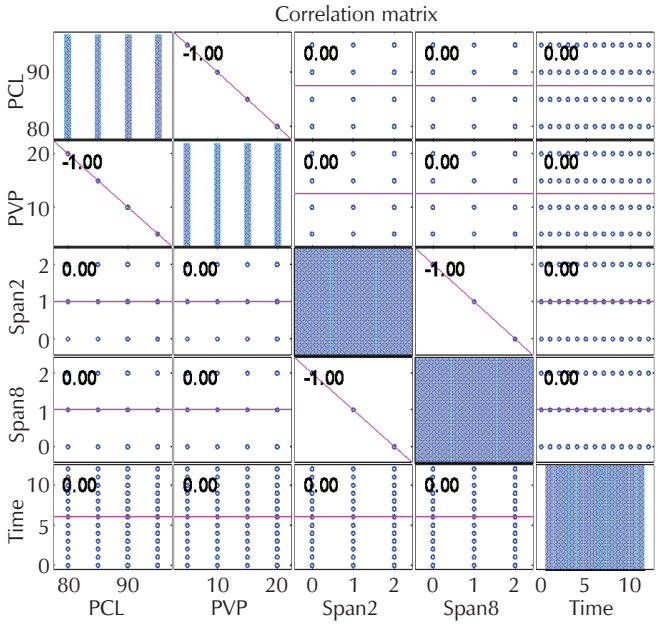

Fig. 3 – Multicollinearity of input variables

Partial least squares

The results of the PLS method showed that this model predicted the cumulative drug release (CD%) following the equation:

$$\begin{aligned} CD\% = {} & 6.108\cdot 10^{15} - 9.58\cdot 10^{14}PCL - \\ & - 9.58\cdot 10^{14}PVP - 2.58\cdot 10^{17}Span20 - \\ & - 2.58\cdot 10^{17}Span80 + 583.6\cdot time \end{aligned} \tag{12}$$

The model reproducibility is measured by the determination of several statistical parameters, such as correlation coefficient, determination coefficient, and root mean square error (RMSE). The results of all phases are summarised in Table 3, and presented graphically in Fig. (4).

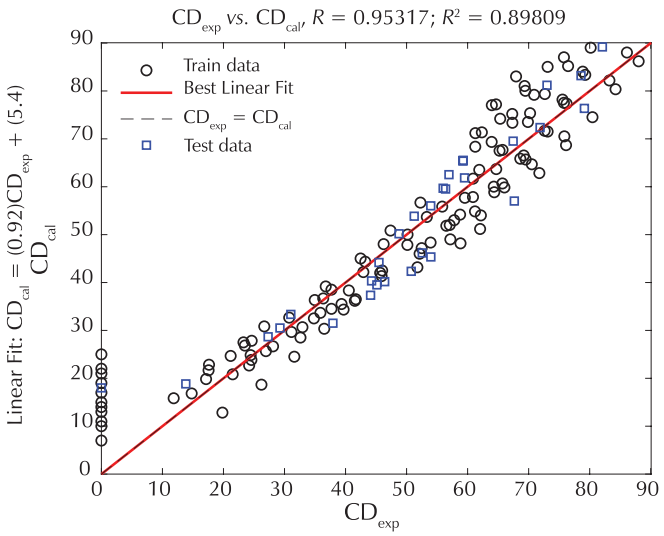

Fig. 4 – Linear correlation between the experimental and estimated cumulative drug release values

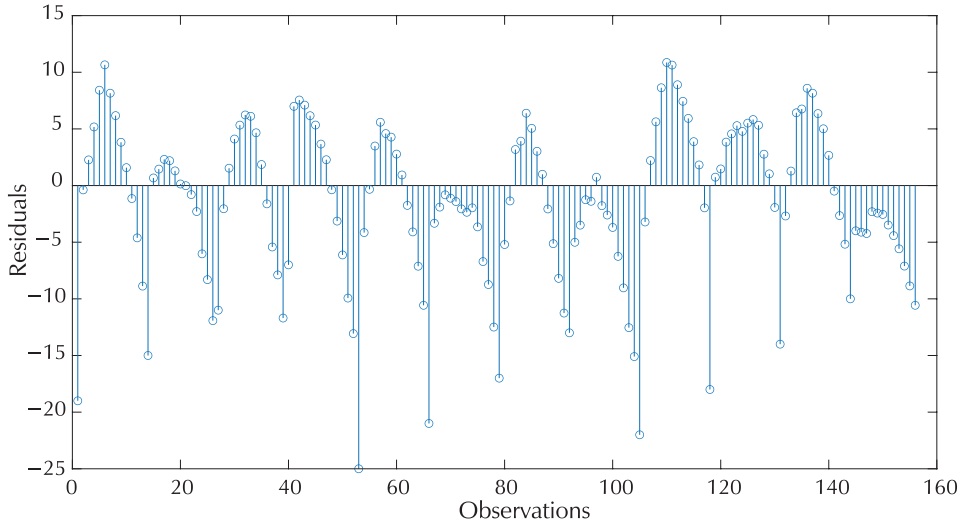

Fig. 5 – Plotted residual errors of the PLS model

Table 3 – Statistical criteria of the three phases (training, validation, and all data points)

	Training	Validation	ALL
RMSE %	7.34	6.44	7.17
r	0.9549	0.9444	0.9532
R^2	0.8993	0.8896	0.8981

Fig. (5) is a graphical representation of residual errors in estimated compared to the experimental cumulative drug release data.

Multiple linear regression

The results of the MLR method exhibited a prediction of the cumulative drug release (CD%) in accordance with the equation:

$$\text{CD\%} = 0.0319 \cdot \text{PCL} + 1.0435 \cdot \text{PVP} - \\ - 1.23 \cdot \text{Span20} + 5.83 \cdot \text{time} \qquad (13)$$

Efficiency of this model was assessed through determination of various statistical parameters including correlation coefficient, coefficient of determination, and root of the mean square error (RMSE). The results of different phases are presented in Table 4 and Fig. (6).

Table 4 – Plotted residual errors of the MLR model

	Training	Validation	ALL
RMSE %	6.95	6.12	6.79
r	0.9527	0.9571	0.9535
R^2	0.9077	0.9160	0.9092

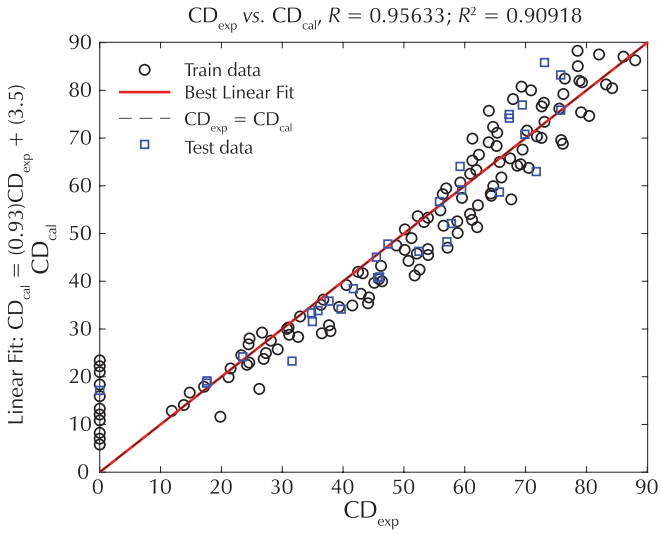

Fig. 6 – Linear correlation between the experimental and estimated cumulative drug release values

Fig. (7) is a graphical representation of the residual errors in estimated compared to the experimental cumulative drug release data.

Ordinary least squares

The results of the OLS method displayed that this model forecasted the cumulative drug release (CD%) following the equation:

$$\text{CD\%} = 0.037 \cdot \text{PCL} + 1.07 \cdot \text{PVP} - \\ - 1.97 \cdot \text{Span20} + 5.77 \cdot \text{time} \qquad (14)$$

The model reliability was measured by the determination of several statistical criteria, such as correlation coefficient *R*, coefficient of determination R^2, and root mean square error (RMSE). The results of all phases are summarised in Table 5, and presented graphically in Fig. (8).

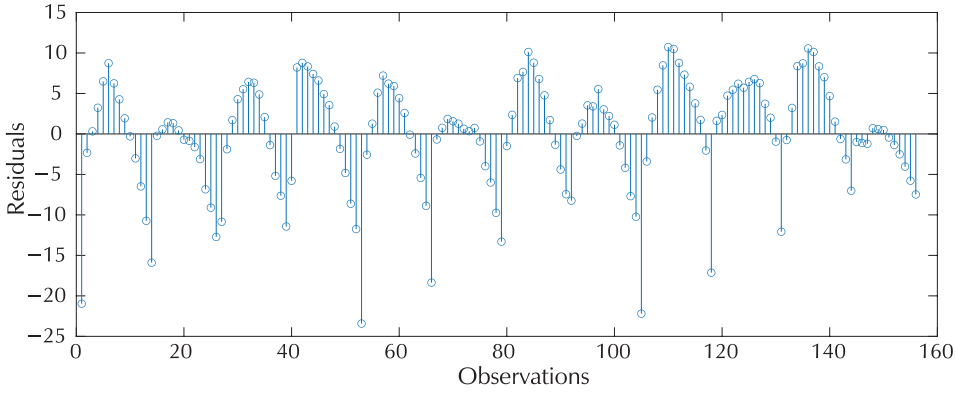

Fig. 7 – Plotted residual errors of the MLR model

Table 5 – Plotted residual errors of the OLS model

	Training	Validation	ALL
RMSE %	6.87	6.53	6.80
r	0.9545	0.9645	0.9562
R^2	0.9025	0.9253	0.9077

Fig. (9) is a graphical representation of the residual errors in estimated compared to the experimental cumulative drug release data.

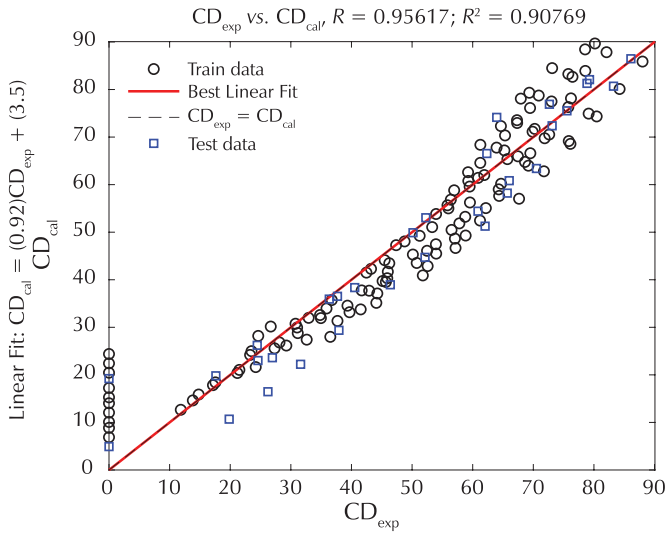

Fig. 8 – Linear correlation between the experimental and estimated cumulative drug release values

DA-SVM modelling

The modelling was performed using MATLAB® software; SVM parameters optimisation, namely C, γ and ε, was carried out by varying them in the range of [10^{-3}, 10^{3}], [10^{-3}, 10^{3}] and [0, 10^{-1}], respectively, with the use of three kernel functions: 'Gaussian', 'linear', and 'polynomial'. The database was randomly divided by the Holdout Partitions method into two sets: one for learning and another for validation, consisting of 80 % and 20 %, respectively.

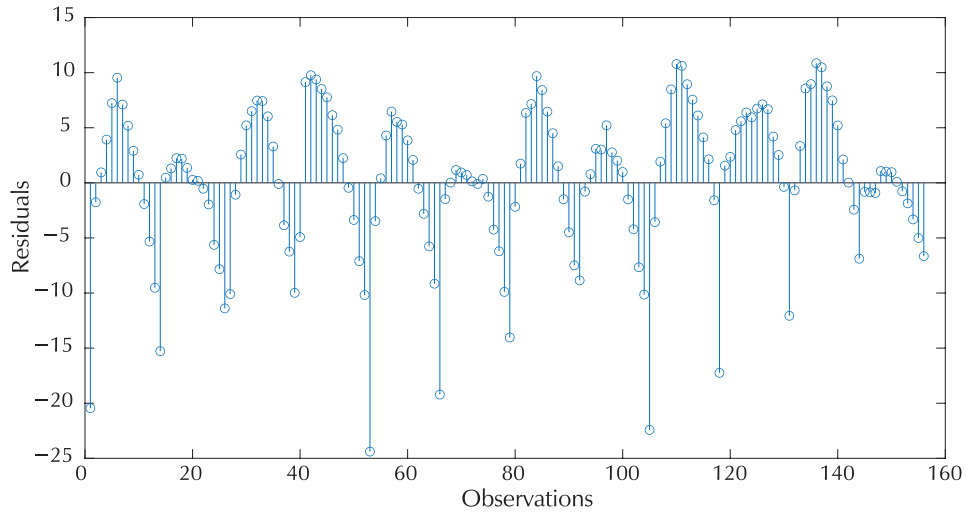

Fig. 9 – Plotted residual errors of the OLS model

Table 6 – Model parameters 'Polynomial kernel function'

C	Polynomial order	Epsilon ε	Kernel function	Quantity of support vectors	RMSE Training %	RMSE Validation %	RMSE ALL %
1000	3	0.6053	Polynomial	114	3.72	3.90	3.77

Polynomial kernel function results

Fig. 10 displays the linear regression curve of the cumulative drug releases ($\%CD_{cal}$) estimated by the polynomial kernel DA-SVM model optimised with the experimental cumulative drug release ($\%CD_{exp}$) for the two phases: learning (*i.e. training*) and validation, when using the *Polynomial* kernel function with a regression vector [α (slope), β (y-intercept), R (correlation coefficient), R^2 (determination coefficient)] = [0.96, 1.40, 0.98, 0.97]. Table 6 summarises the parameters of the model.

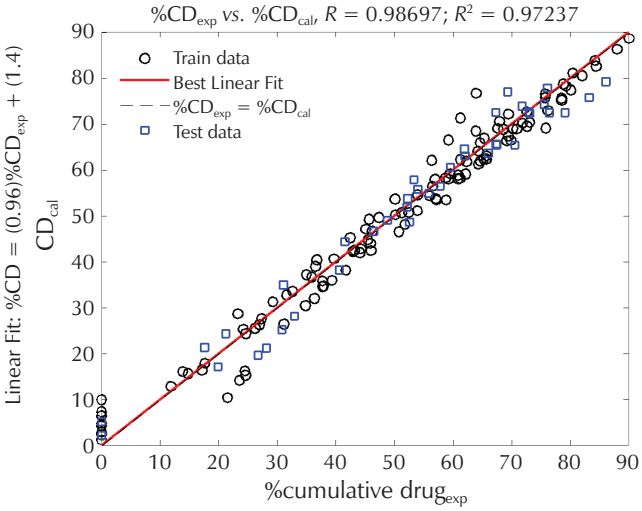

Fig. 10 – Linear correlation between the experimental values and the values predicted by the Polynomial kernel function model

Linear kernel results

With the *Linear* kernel function being used (under the same circumstances as the previous model), the resulting regression linear curve has a correlation vector comprising the following values: [α (slope), β (y-intercept), R (correlation coefficient), R^2 (determination coefficient)] = [1.000, 0.120, 0.989, 0.978], as presented in Table 7 and Fig. 11.

Table 7 – Model parameters linear kernel function

C	Kernel function	Quantity of support vectors	RMSE Training %	RMSE Validation %	RMSE ALL %
2000	Linear	125	3.52	3.13	3.45

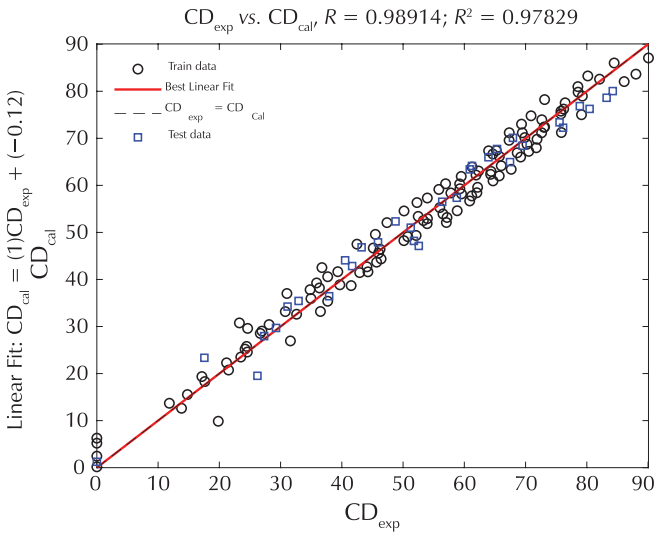

Fig. 11 – Linear correlation between the experimental values and the values predicted by the Linear kernel function model

Gaussian kernel function results

The SVM model performance results obtained during the learning and the validation phases (Fig. 12), clearly show that all of the dots are located on the first bisector with a regression coefficient that is near ideal [α (slope), β (intercept), R (correlation coefficient), R^2 (determination coeffi-

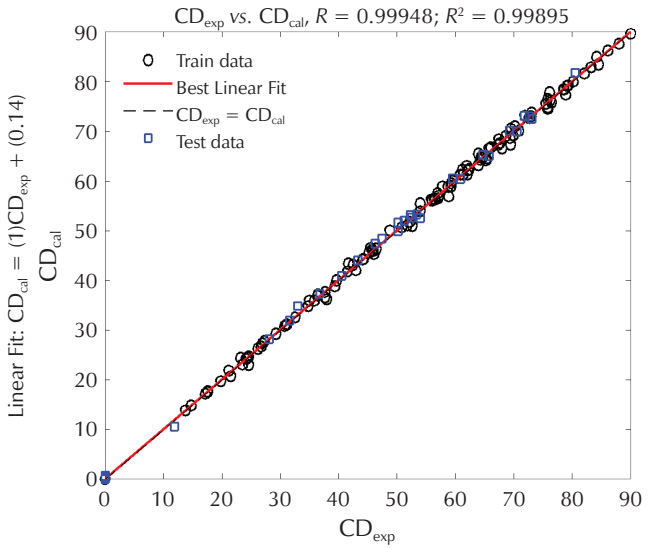

Fig. 12 – Linear correlation between the experimental values and the values predicted by the Gaussian kernel function model

Table 8 – Model parameters of Gaussian kernel function

C	Sigma σ	Epsilon ε	Kernel function	Quantity of support vectors	RMSE Training %	RMSE Validation %	RMSE ALL %
2.0995	1.3473	0.6053	Gaussian	125	0.72	0.86	0.75

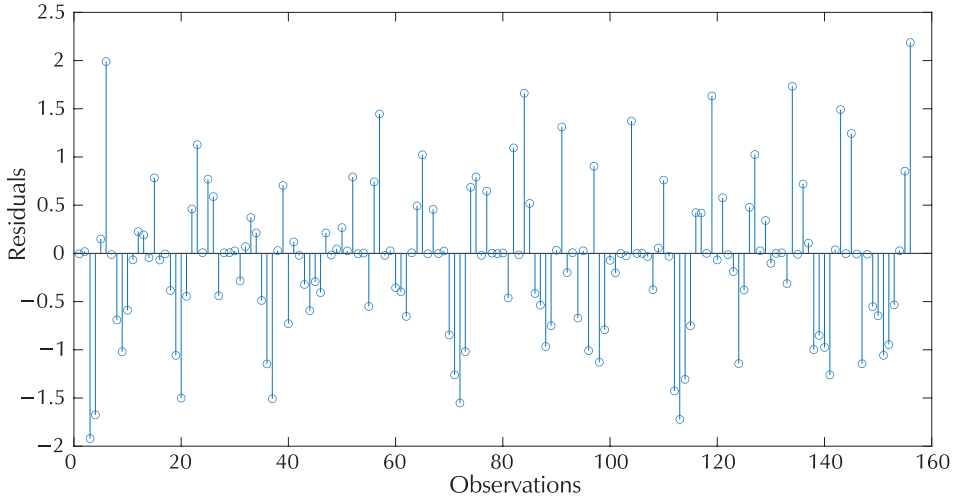

Fig. 13 – Plotted residual errors of the optimal DA-SVM model

cient)] = [1.000, 0.140, 0.999,0.999], which proves how robust is the SVM model. Results related to the model parameters from the Gaussian RBF kernel function are shown in Table 8.

Fig. (13) is a graphical representation of the residual errors in estimated compared to the experimental cumulative drug release data.

PLS, MLR, OLS, and DA-SVM comparison

In DA-SVM modelling, after using kernels (polynomial, linear, Gaussian), it became clear that the Gaussian kernel had strength and was better than were the others. That is why it was chosen for comparison with other models (PLS, MLR, OLS). This comparison in terms of RMSE is illustrated in Fig. (14). According to the results of each model, the comparative study shows the Model DA-SVM gives a good result of the RMSE errors.

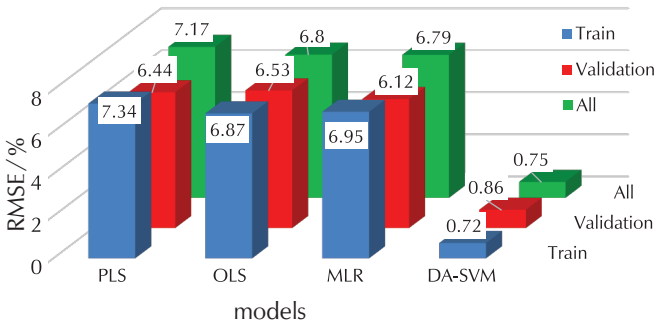

Fig. 14 – Graphical representation of RMSE errors of the four models

Graphic representation of the ideal model (Gaussian kernel function)

After modelling, the results were displayed in Figs. 7 to 10, where the error was almost non-existent, so that a good correspondence between the values of the experimental[2] and the predicted values appears.

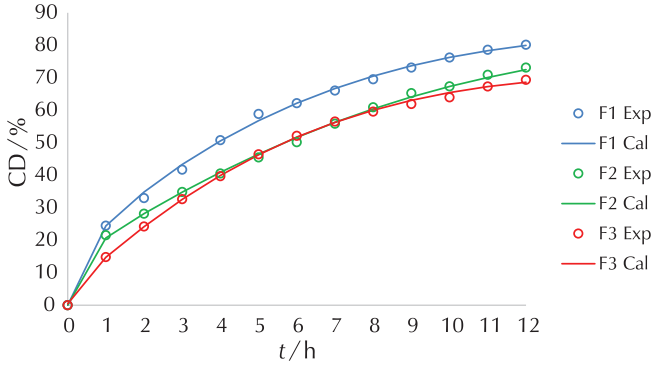

Fig. 15 – Drug release kinetics from three experimental[2] formulations and calculations (F1–F3)

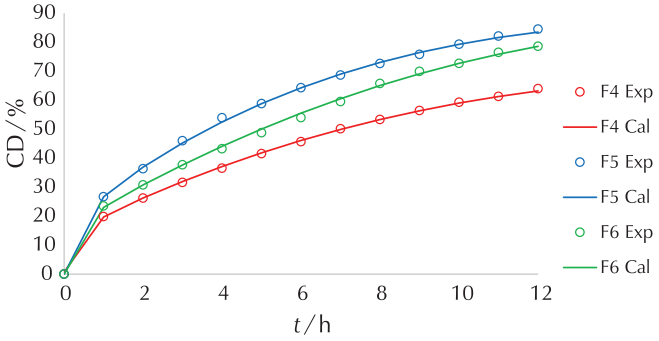

Fig. 16 – Drug release kinetics from three experimental² formulations and calculations (F4–F6)

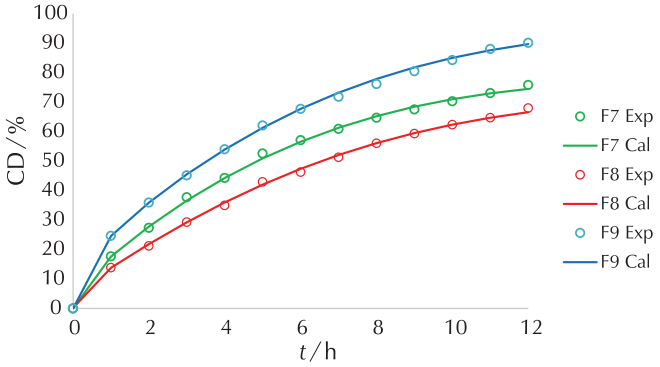

Fig. 17 – Drug release kinetics from three experimental² formulations and calculations (F7–F9)

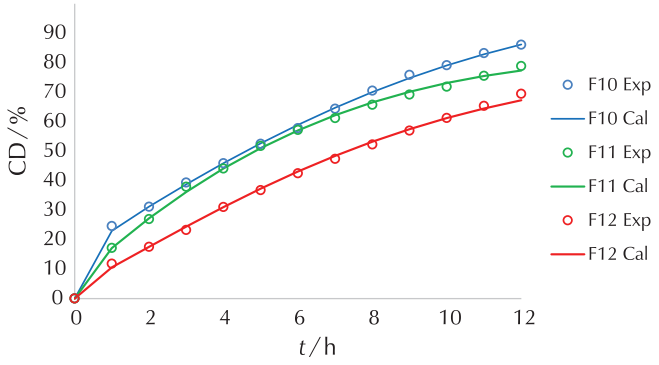

Fig. 18 – Drug release kinetics from three experimental² formulations and calculations (F10–F12)

Computer application

To facilitate the use of model DA-SVM developed, a graphical interface (Fig. 19) was built to calculate the cumulative drug release (CD%).

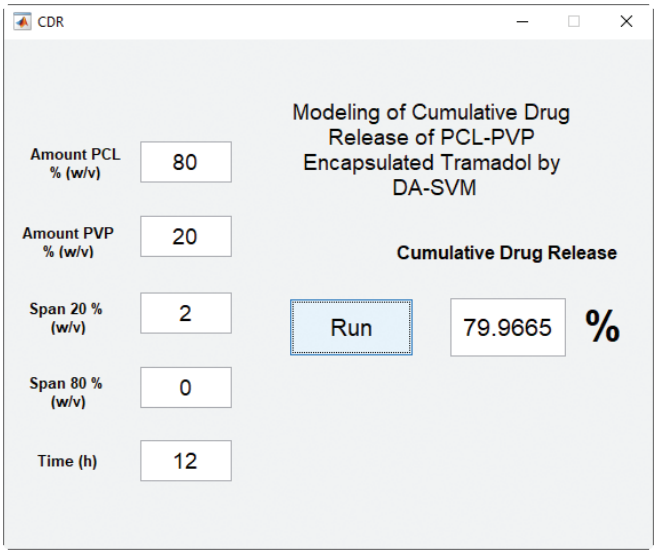

Fig. 19 – Graphical interface for calculating the CD%

5. Conclusion

In this paper, a numerical and statistical study was conducted using experimental data of the prolonged-release formulas of Tramadol, prepared by the complex variation of two biopolymers, PCL and PVP, taken from literature. DA-SVM, PLS, MLR, and OLS modelling techniques were used to develop a mathematical estimate of Tramadol release in twelve kinetics accurately. Best results obtained in terms of overall root mean square error and the determination coefficient were 0.753 %, 0.999 respectively, which show evident robustness of DA-optimised-SVM over other models tested. In addition, a graphical interface was built to determine CD% easily and to make better use of the established model.

References
Literatura

1. *J. H. Hoofnagle, J. Serrano, J. E. Knoben, V. J. Navarro*, LiverTox: A Website on Drug-Induced Liver Injury, Hepatology **57** (3) (2013) 873–874, doi: https://doi.org/10.1002/hep.26175.
2. *B. Bhanupriya, A. Shering, M. Sivabalan*, Preparation and Characterisation of Tramadol Microspheres for Post-Operative Pain, Int. J. Pharm. Biol. Arch. **2** (3) (2011).
3. *R. Cairns, A. L. Schaffer, J. A. Brown, S. Pearson, N. A. Buckley*, Codeine Use and Harms in Australia: Evaluating the Effects of Re-scheduling, Addiction **115** (3) (2020) 451–459, doi: https://doi.org/10.1111/add.14798.
4. *M. Przyklenk, M. Vemmer, M. Hanitzsch, A. Patel*, A Bioencapsulation and Drying Method Increases Shelf Life and Efficacy of Metarhizium Brunneum Conidia, J. Microencapsul. **34** (5) (2017) 498–512, doi: https://doi.org/10.1080/02652048.2017.1354941.
5. *T. Rudtanatip, B. Boonsri, J. Praiboon, K. Wongprasert*, Bioencapsulation Efficacy of Sulfated Galactans in Adult Artemia Salina for Enhancing Immunity in Shrimp *Litopenaeus Vannamei*, Fish Shellfish Immun. **94** (2019) 90–98, doi: https://doi.org/10.1016/j.fsi.2019.08.065.
6. *C. Nikapitiya, S. H. S. Dananjaya, S. L. Edirisinghe, H. P. S. U. Chandrarathna, J. Lee, M. De Zoysa*, Development of

Phage Delivery by Bioencapsulation of Artemia Nauplii with Edwardsiella Tarda Phage (ETP-1), Braz. J. Microbiol. **51** (4) (2020) 2153–2162, doi: https://doi.org/10.1007/s42770-020-00324-y.

7. *A. Ibrir, Y. Kerchich, N. Hadidi, H. Merabet, M. Hentabli*, Prediction of the Concentrations of PM_1, $PM_{2.5}$, PM_4, and PM_{10} by Using the Hybrid Dragonfly-SVM Algorithm, Air Qual. Atmos. Heal. **14** (2020) 313–323, doi: https://doi.org/10.1007/s11869-020-00936-1.
8. *M. Laidi, A. Abdallah el Hadj, C. Si-Moussa, O. Benkortebi, M. Hentabli, S. Hanini*, CMC of Diverse Gemini Surfactants Modelling Using a Hybrid Approach Combining SVR-DA, Chem. Ind. Chem. Eng. Q. **27** (3) (2021) 299–312, doi: https://doi.org/10.2298/CICEQ200907048L.
9. *S. Keskes, S. Hanini, M. Hentabli, M. Laidi*, Artificial Intelligence and Mathematical Modelling of the Drying Kinetics of Pharmaceutical Powders, Kem. Ind. **69** (3–4) (2020) 137–152, doi: https://doi.org/10.15255/KUI.2019.038.
10. *H. Benimam, C. S. Moussa, M. Hentabli, S. Hanini, M. Laidi*, Dragonfly-Support Vector Machine for Regression Modeling of the Activity Coefficient at Infinite Dilution of Solutes in Imidazolium Ionic Liquids Using σ-Profile Descriptors, J. Chem. Eng. Data **65** (6) (2020) 3161–3172, doi: https://doi.org/10.1021/acs.jced.0c00168.
11. *M. Moussaoui, M. Laidi, S. Hanini, M. Hentabli*, Artificial Neural Network and Support Vector Regression Applied in Quantitative Structure-Property Relationship Modelling of Solubility of Solid Solutes in Supercritical CO_2, Kem. Ind. **69** (11–12) (2020) 611–630, doi: https://doi.org/10.15255/KUI.2020.004.
12. *C. Cortes, V. Vapnik*, Support-Vector Networks, Mach. Learn. **20** (3) (1995) 273–297, doi: https://doi.org/10.1007/BF00994018.
13. *H. Drucker, C. J. C. Burges, L. Kaufman, A. J. Smola, V. Vapnik*, Support Vector Regression Machines, in: Advances in neural information processing systems, Vol. 9 of NIPS, MIT Press, Cambridge, MA 1997, pp. 155–161.
14. *V. Vapnik, S. E. Golowich, A. J. Smola*, Support Vector Method for Function Approximation, Regression Estimation and Signal Processing, in: Advances in neural information processing systems 9, MIT Press, 1997, pp. 281–287, url: https://proceedings.neurips.cc/paper/1996/file/4f284803bd0966cc24fa8683a34afc6e-Paper.pdf.
15. *T. M. Bafitlhile, Z. Li*, Applicability of ε-Support Vector Machine and Artificial Neural Network for Flood Forecasting in Humid, Semi-Humid and Semi-Arid Basins in China, Water **11** (1) (2019) 85, doi: https://doi.org/10.3390/w11010085.
16. *J. Platt*, Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, Technical Report MSR-TR-98-14, 1998.
17. *M. Abdelkader, M Laidi, S. Hanini, M. Hentabli, A. Amrane*, A Grey Wolf Optimizer-Based Fractional Calculus in Studies on Solar Drying. Kem. Ind. **70** (1-2) (2021) 39–47, doi: https://doi.org/10.15255/KUI.2020.035.
18. *M. Hentabli, A.-E. Belhadj, H. Benimam, F. Dahmoune, S. Keskes*, Vacuum Drying of the Terbinafine HCl Powder: A Kinetics Study and Mathematical Modeling, Powder Technol. **383** (2021) 220–232, doi: https://doi.org/10.1016/j.powtec.2021.01.038.
19. *Y. Mesllem, A. E. H. Abdallah, M. Laidi, S. Hanini, M. Hentabli,* Artificial Neural Network Modelling of Multi-System Dynamic Adsorption of Organic Pollutants on Activated Carbon, Kem. Ind. **70** (1-2) (2021) 1–12, doi: https://doi.org/10.15255/KUI.2020.011.
20. *S. Mirjalili,* Dragonfly Algorithm: A New Meta-Heuristic Optimization Technique for Solving Single-Objective, Discrete, and Multi-Objective Problems, Neural Comput. Appl. **27** (4) (2016) 1053–1073, doi: https://doi.org/10.1007/s00521-015-1920-1.
21. *S. Keskes, M. Hentabli, M. Laidi, S. Hanini,* Modelling Drying Time of Candesartan Cilexetil Powder Using Computational Intelligence Technique, Kem. Ind. **70** (3-4) (2021) 137–144, doi: https://doi.org/10.15255/KUI.2020.048.
22. *Y. Mesllem, A. E. H. Abdallah, M. Laidi, S. Hanini, M. Hentabli,* Artificial Neural Network Modelling of Multi-System Dynamic Adsorption of Organic Pollutants on Activated Carbon, Neural Comput. Appl. (2021) 1–12, doi: https://doi.org/10.1007/s00521-021-05890-2.

SAŽETAK

DA-SVM, MLR, PLS i OLS modeliranje kumulativnog otpuštanja Tramadola iz formulacija inkapsuliranih s PCL i PVP

Ahmed Chabane,[a] *Fatiha Bouchal*,[a] *Mohamed Hentabli*,[b*] *Farouk Rezgui*[a] i *Houssam Eddine Slama*[c]

Cilj ovog rada bio je modeliranje kinetike kumulativnog otpuštanja lijeka iz formulacija inkapsuliranih biorazgradivim polikaprolaktonom i polivinilpirolidonom. Različiti omjeri polimera pripremljeni su isparavanjem otapala uz upotrebu Span 20 i Span 80 kao površinski aktivnih tvari. U modeliranju kinetike primijenjena su četiri pristupa: hibridni pristup kombiniranjem metode potpornih vektora i *Dragonfly* algoritma (DA-SVM), metoda parcijalnih najmanjih kvadrata (PLS), višestruka linearna regresija (MLR) te metoda najmanjih kvadrata (OLS). Provedena je usporedba kvalitete predviđanja kumulativnog otpuštanja lijeka, ovisno o primijenjenom polimeru i vremenu. Statistička analiza nije ukazala na probleme s odabranim ulaznim varijablama. Rezultati su pokazali superiornost predviđanja DA-SVM modelom uz koeficijent determinacije blizu jedinice te RMSE pogrešku blizu nule. Za izračun kumulativnog otpuštanja lijeka konstruirano je grafičko sučelje.

Ključne riječi
Dragonfly algoritam, metoda potpornih vektora, Tramadol, kumulativno otpuštanje lijekova, modeliranje, biopolimeri, metoda najmanjih kvadrata

[a] *Laboratoire des Matériaux Organiques (LMO), Département de Génie des Procédés, Faculté de Technologie, Université de Bejaia, 06000 Bejaia, Alžir*
[b] *Laboratory of Biomaterials and Transport Phenomena (LBMPT), Faculty of Technology, University Yahia Fares of Médéa 26 000, Médéa, Alžir*
[c] *Quality Control Laboratory, SAIDAL-Médéa 26 000, Médéa, Alžir*

Izvorni znanstveni rad
Prispjelo 1. veljače 2021.
Prihvaćeno 17. srpnja 2021.