Support Vector Machines for Biodegradability Prediction of Chemicals

Registration No: 2300221165

***Abstract*—Chemicals with persistence, bioaccumulation, and toxicity (PBT) properties present a hazard to natural environments and thus predict the properties of new chemicals, such as biodegradability, which would allow for the creation of more environmentally friendly chemicals. This paper presents three versions of the SVM machine-learning algorithm to predict the biodegradability of chemicals from QSAR data from Mansouri et al. All three models showed high levels of accuracy, performing well in all metrics. Adding the RBF kernel to the SVM model causes a greater level of overfitting than the linear version. It was deemed that the L2-regularised model performed best from a holistic viewpoint.**

I. Introduction

CHEMICALS with persistence, bioaccumulation, and toxicity (PBT) properties do not decay over time, damaging the environment and the organisms that inhabit it [1]. Predicting the ability of a chemical to biodegrade aerobically is vital for developing safe chemicals due to its primary role in eliminating organic chemicals from the environment [2].

In developing predictive models for assessing the biodegradability of chemicals, prioritising precision and reliability is vital. Inaccuracies in these models may lead to adverse environmental consequences, including the contamination of ecosystems as seen in tragedies like Love Canal, New York [3] and the uses of pesticides and industrial chemicals like dichlorodiphenyltrichloroethane (DDT), chlordane, and dieldrin [1].

Quantitative Structure-Activity Relationship (QSAR) models are one of the two main ways of predicting the biodegradation properties of chemicals. QSAR models are a statistical approach where the model for biodegradation is created with experimental biodegradation data and structural infomation about the chemicals [4].

Mansouri et al. [5] created a QSAR model containing a data set of 1055 chemicals with the aim of developing a reliable classification QSAR models for chemical biodegradability. The study presented the classification performances of the three QSAR models (kNN, PLSDA, SVM), demonstrating good fitting and cross-validation statistics. High accuracy in predicting the test set was achieved, surpassing already published models on biodegradation.

II. Data Processing

A. Data-Set

The QSAR data used in this report comes from Mansouri et al. [5] and contains 41 predictor variables and binary classifications for individual chemicals. The data set comprises a total of 1055 samples.

B. Data Validation

To maintain result accuracy and code integrity, the data set is checked for missing/duplicate entries using built-in Matlab functions. This step found no faults in the data set.

C. Outlier detection

Outliers were removed from the data set to reduce noise and help create a more reliable model. An outlier was defined as any value outside of the mean $\pm$ 3x*standard deviation range [6]. This reduced the dataset from 1055 to 730 samples.

D. Data standardisation

The z-score was used on the dataset to standardise the variables, preventing features with larger magnitudes from dominating the model training process [7].

E. Principal Component Analysis

Principal Component Analysis (PCA) was used to reduce the dimensionality while retaining 90% of the variance. PCA extracts new features from a data set's original features, creating uncorrelated variables [7]. Two data sets were created at this stage, one with and one without PCA.

F. Cross validation

The data-set was divided into training and test sets using 5-fold cross-validation, creating five pairs. This approach maximises data utility and reduces bias by systematically partitioning the data-set into subsets for training and testing.

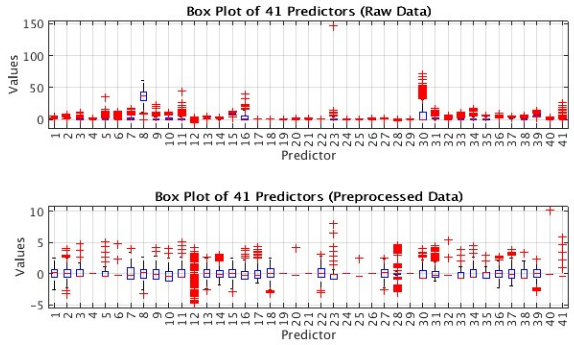

Fig. 1: Box plot of data set before and after pre-processing

III. Methodology

A. Support Vector Machines

In a binary classification problem, the dataset takes the form 1, where x represents a vector of predictors and y is a label of -1 or 1. Support vector machines (SVM) attempt to predict the label by finding a hyperplane that best separates data points of different classes.

$$(x_i, y_i)_{i=1}^{N} | y_i \in \{-1, 1\} \tag{1}$$

2 represents the prediction function of SVM, denoted as p(x). Where w is the weight vector, b is the bias term, and X is the feature vector. SVM aims to find the optimal values for w and b such that the hyperplane p(x)=0 effectively separates the data into two classes. During training, SVM aims to maximize the margin, the distance between the hyperplane and the nearest data points from each class (a support vector). 1

$$p(x) = wX + b = b + w_1X_1 + w_2X_2 + ... + w_NX_N \tag{2}$$

B. Optimization problem

The initial two models presented in this report use the linear SVM predictor (2). The corresponding objective functions for these models are as follows: The SVM-L1 model (Equation 2) is expressed as

$$P(w, b) = \lambda \|w\|_1 + C \sum_{i=1}^{n} l(w^T x_i + b, y_i) \tag{3}$$

where $\|w\|_1$ denotes the L1 norm of the weight vector. Additionally, the SVM-L2 model (Equation 3) is characterized by

$$P(w, b) = \frac{\lambda}{2} \|w\|_2^2 + C \sum_{i=1}^{n} l(w^T x_i + b, y_i) \tag{4}$$

where $\|w\|_2^2$ denotes the L2 norm of the weight vector. References [8], [9] , [10], and [11] correspond to sources supporting and detailing the development of these models.

$$\hat{\omega} = \underset{\omega}{argmin} P(w, b) \tag{5}$$

The optimization process involves finding the optimal weight vector $\hat{\omega}$ that minimizes the objective function P(w,b) among all possible weight vectors w, as expressed by (5).

The objective functions have two regularization terms. C influences the hinge loss function, where a larger value broadens the margin, potentially increasing errors, and a smaller value narrows it [8]. λ is the weight penalty; a larger λ increases the penalty for large weights, yielding a simpler model with better generalization but potentially sacrificing training set accuracy [12].

$$l(w^T x + b, y) = \max(0, 1 - y(w^T x + b)) \tag{6}$$
$$l(w^T x + b, y) = \max(0, 1 - y(w^T x + b))^2 \tag{7}$$

The loss function $l(p(x), y)$ estimates the cost of the prediction error where the value $l(p(x), y)$ is greater than zero if the predicted label is different from that of the actual value. This acts as a way to measure the quality of the classification made by a model. Two loss functions were used in the models presented: hinge loss (6) for L1 regularization and quadratically smoothed hinge loss (7) for L2 regularization [10].

C. Stochastic Gradient Descent

The Stochastic Gradient Descent (SGD) algorithm is a simplified version of gradient descent. Rather than calculating the gradient of —equation— precisely, it estimates the gradient based on a single randomly picked sample [13]. The update rules for w_t and b_t can be seem below:

$$\begin{aligned} w_t &= w_{t-1} - \eta_t g_t(w_t), \\ b_t &= b_{t-1} - \eta_t g_t(b_t). \end{aligned} \tag{8}$$

$$g_t(w_t) = \lambda w_t - C \sum_{i=1}^{n} \frac{\partial l(w^T x + b, y)}{\partial w} x \tag{9}$$

$$\sum_{t=1}^{\infty} \eta_t = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} \eta_t^2 = \infty \tag{10}$$

In determining the update step decay schedule in SGD, Robbins and Monro [14] showed that convergence is guaranteed if the conditions (10) are met. A commonly used formula (12) was proposed in Xu [15], meeting the

required criteria, and is used to update the learning rate after each iteration through the SGD algorithm.

$$g_t(b_t) = -C \sum_{i=1}^{n} \frac{\partial l(w^T x + b, y)}{\partial b} \tag{11}$$

$$\eta_t = \frac{\eta_0}{(1 + \lambda \eta_0 t)} \tag{12}$$

This formula ensures that the learning rate decreases over time, providing a balance between convergence and exploration during the SVM training process with SGD.

D. Kernel

The third model presented in this report is a SVM model using a Radial Basis Function (RBF) kernel expressed by $G(x_j, x_k) = \exp\left(-\|x_j - x_k\|^2\right)$. This model enhances the dataset by establishing a non-linear hyperplane for when the dataset is not linearly separable.

E. Hyper-parameter Selection

The regularisation terms λ and C were treated as hyperparameters in the model and were optimised using Bayesian Optimisation. This method is a probabilistic model-based approach that uses a probabilistic surrogate model to approximate the objective function and an observation model describing the data [16]. The model's implementation was accomplished by using MATLAB's built-in Bayesopt function.

IV. Model Analysis

TABLE I: Model Evaluation Metrics

Model	Accuracy	Precision	Recall	F1Score
	Mean Average over Training Sets			
SVM L1	82.0	0.783	0.707	0.742
SVM L1 PCA	78.6	0.724	0.685	0.702
SVM L2	85.0	0.832	0.745	0.784
SVM L2 PCA	82.7	0.809	0.694	0.746
SVM RBF	92.4	0.911	0.882	0.896
SVM RBF PCA	90.1	0.878	0.851	0.864
	Mean Average over Test Sets			
SVM L1	81.8	0.766	0.742	0.749
SVM L1 PCA	76.7	0.697	0.643	0.666
SVM L2	83.6	0.798	0.739	0.764
SVM L2 PCA	81.1	0.777	0.668	0.718
SVM RBF	85.5	0.808	0.784	0.795
SVM RBF PCA	85.5	0.811	0.777	0.792

TABLE II: Confusion Matrix for models summed across all k-folds of training sets

SVM L1 regularisation

	Predicted - No PCA		Predicted - PCA	
	Positive	**Negative**	**Positive**	**Negative**
Actual Positive	200 (TP)	64 (FN)	174 (TP)	75 (FN)
Actual Negative	69 (FP)	397 (TN)	95 (FP)	389 (TN)

SVM L2 regularisation

	Predicted - No PCA		Predicted - PCA	
	Positive	**Negative**	**Positive**	**Negative**
Actual Positive	200 (TP)	51 (FN)	181 (TP)	50 (FN)
Actual Negative	69 (FP)	410 (TN)	88 (FP)	411 (TN)

SVM RBF Kernel

	Predicted - No PCA		Predicted - PCA	
	Positive	**Negative**	**Positive**	**Negative**
Actual Positive	212 (TP)	49 (FN)	210 (TP)	47 (FN)
Actual Negative	57 (FP)	412 (TN)	59 (FP)	414 (TN)

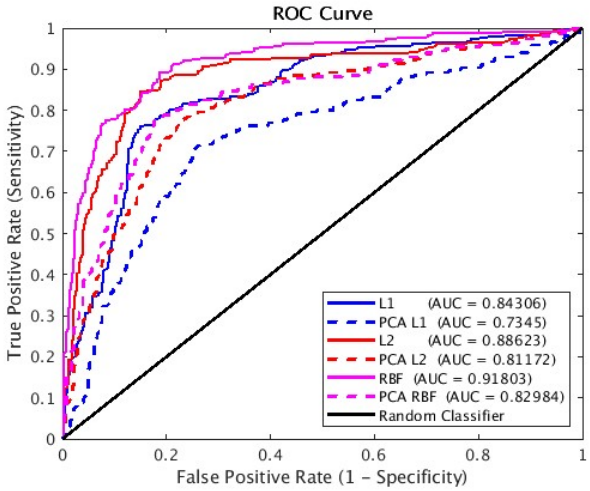

Fig. 2: ROC Curve comparing the models

V. Conclusion and Recommendations

A. Fitting

When comparing the performance metrics of the training set to those of the test sets, there is only a marginal decrease (a difference of between 0.2% and 1.9%) in value between the linear models, suggesting that none of the modes are overfitted as they are still performing well when presented with new data. However, the RBF kernel model has a difference between 4.6% and 6.9%, suggesting greater overfitting. This is a negative trait as it also shows lower precision on the test set, meaning it is more likely to classify a non-biodegradable chemical as bio-degradable.

B. Effect of PCA

When comparing the models, PCA generally shows a reduction in accuracy, suggesting the addition of PCA might sacrifice some discriminative power. This trend is seen in the other metrics where the non-dimensionality-reduced data outperforms PCA. However, this effect is inconsistent in scale across the board, especially considering the difference between SVM RBF and SVM RBF PCA.

C. Effect of regularization

The area under the ROC curve is a metric that shows the trade-off between true positive and false positive rates. SVM L2's auc is 0.88, and SVM L1's auc is 0.84, suggesting that SVM L2 is better at classifying the data points than L1. This idea is further supported by the Accuracy Precision Recall F1Score values, all supporting L2 as the better model.

D. Complexity

The complexity of implementing RBF models in the associated code is far simpler than that of the other model. However, this is primarily due to the use of an inbuilt function. However, this simplicity in the coding of the model does not transfer over to quicker training times, with SVM L2 taking 16.45 seconds (averaged over five runs) to train all five folds. In contrast, the RBF model took, on average, 32.4 seconds, a 97.0% increase. This model would not be needed to perform in real-time, so training time is less critical. However, when not implemented with inbuilt functions, the RBF model becomes more complicated owing to the extra steps involved with the kernel.

E. Recommondation

This report recommends that the SVM L2 model variant presented is the optimal model as it provides an accurate result without experiencing much overfitting while remaining a reasonably simplistic model with low training times.

F. Further Improvements

To further improve the SVM L2 model, the pre-processing could be further optimised, namely making normalising after generating the training and test sets, as this would prevent any information leakage from the testing set into the training set, which is currently a problem with the current model. Another area of exploration could be other kernels that may be more effective at creating a non-linear classification boundary without causing overfitting within the model.

REFERENCES

[1] F. Cheng *et al.*, "In silico assessment of chemical biodegradability," *Journal of Chemical Information and Modeling*, vol. 52, no. 3, pp. 655–669, 2012.

[2] K. Huang and H. Zhang, "Classification and regression machine learning models for predicting aerobic ready and inherent biodegradation of organic chemicals in water," *Environmental Science & Technology*, vol. 56, no. 17, pp. 12 755–12 764, 2022.

[3] M. R. Fowlkes and P. Y. Miller, "Chemicals and community at love canal," in *The Social and Cultural Construction of Risk: Essays on Risk Selection and Perception*, 1987, pp. 55–78.

[4] B. Philipp, M. Hoff, F. Germa, B. Schink, D. Beimborn, and V. Mersch-Sundermann, "Biochemical interpretation of quantitative structureactivity relationships (qsar) for biodegradation of n-heterocycles: A complementary approach to predict biodegradability," *Environmental Science & Technology*, vol. 41, no. 4, pp. 1390–1398, 2007.

[5] K. Mansouri *et al.*, "Quantitative structure–activity relationship models for ready biodegradability of chemicals," *Journal of Chemical Information and Modeling*, vol. 53, no. 4, pp. 867–878, 2013.

[6] J. Yang *et al.*, "Outlier detection: how to threshold outlier scores?" in *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, December 2019, pp. 1–6.

[7] H. S. Obaid *et al.*, "The impact of data pre-processing techniques and dimensionality reduction on the accuracy of machine learning," in *2019 9th Annual Information Technology, Electromechanical Engineering and Microelectronics Conference (IEMECON)*, Jaipur, India, 2019, pp. 279–283.

[8] D. Hoiem. (2023) Svms and sgd applied machine learning. Accessed: Dec. 11, 2023. [Online]. Available: https://courses.grainger.illinois.edu/CS441/sp2023/lectures/Lecture%2008%20-%20SVMs%20and%20SGD.pdf

[9] V. Srikumar. (unknown) Support vector machines: Training with stochastic gradient descent. Accessed: [insert date here]. [Online]. Available: https://users.cs.utah.edu/~zhe/pdf/lec-19-2-svm-sgd-upload.pdf

[10] K. Sopya and P. Drozda, "Stochastic gradient descent with barzilai–borwein update step for svm," *Information Sciences*, vol. 316, pp. 218–233, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025515002467

[11] A. Dedieu *et al.*, "Solving l1-regularized svms and related linear programs: Revisiting the effectiveness of column and constraint generation," *Journal of Machine Learning Research*, vol. 23, pp. 1–41, 2022, accessed: Dec. 11, 2023. [Online]. Available: https://www.jmlr.org/papers/volume23/19-104/19-104.pdf

[12] N. Halder. (2023) Decoding the regularization parameter lambda in machine learning: An in-depth exploration of its.... Medium. Accessed: Dec. 11, 2023. [Online]. Available: https://shorturl.at/zOYZ4

[13] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*. Physica-Verlag HD, 2010, pp. 177–186.

[14] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, 1951.

[15] W. Xu, "Towards optimal one pass large scale learning with averaged stochastic gradient descent," 2011.

[16] J. Snoek, "Practical bayesian optimization of machine learning algorithms," 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf