JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

pubs.acs.org/jcim

Prediction of Chemical Biodegradability Using Support Vector Classifier Optimized with Differential Evolution

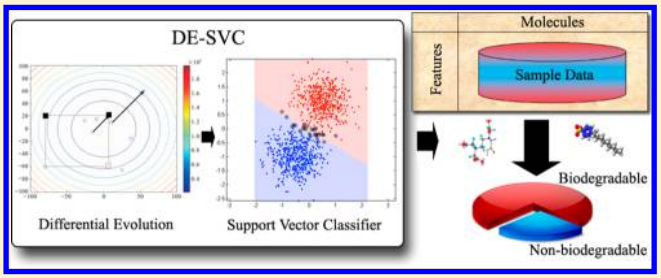
Qi Cao*[,†,‡] and K. M. Leung[‡]

[†]Department of Training, Logistical Engineering University, Chongqing 401311, China

[‡]Department of Computer Science and Engineering, Polytechnic School of Engineering, New York University, Brooklyn, New York 11201, United States

Supporting Information

ABSTRACT: Reliable computer models for the prediction of chemical biodegradability from molecular descriptors and fingerprints are very important for making health and environmental decisions. Coupling of the differential evolution (DE) algorithm with the support vector classifier (SVC) in order to optimize the main parameters of the classifier resulted in an improved classifier called the DE-SVC, which is introduced in this paper for use in chemical biodegradability studies. The DE-SVC was applied to predict the biodegradation of chemicals on the basis of extensive sample data sets and known structural features of molecules. Our optimization experiments showed that DE can efficiently find the proper parameters of the SVC. The resulting classifier possesses strong robustness and reliability compared with grid search, genetic algorithm, and particle swarm optimization methods. The classification experiments conducted here showed that the DE-SVC exhibits better classification performance than models previously used for such studies. It is a more effective and efficient prediction model for chemical biodegradability.



1. INTRODUCTION

Organic chemicals of diverse levels of toxicity exist in consumer products and often end up in our environment. As the primary environmental dissipation mechanism, chemical biodegradation is a main factor influencing the transformation and ultimate fate of these chemicals in our ecosystem. Chemicals that degrade slowly may have their toxicities concentrated and aggregated to harmful levels. Moreover, their harmfulness can be amplified immensely with prolonged exposure.

As a result, various health and environmental regulatory organizations around the world have included persistence in the evaluation of industrial chemicals. Under the Pollution Prevention Act of 1990, it is the policy of the United States that pollution should be prevented or reduced at the source whenever possible.[1] Europe put forward the Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH) program, which asks for information on ready biodegradation for industrial chemicals produced or imported in quantities of more than 1 ton per year.[2] One way to reduce health risks is to design safer compounds. However, the number of those chemicals is already over tens of thousands and is rapidly increasing year after year. Experimental testing of every single chemical is almost impossible given the high financial cost and the impact on animal welfare.

The fact that industry is producing these chemicals at a much higher rate than regulatory organizations can experimentally test their biodegradation necessitates in silico assessment using several statistical quantitative structure−biodegradability relationship (QSBR) models in order to predict the biodegradation of chemicals.[3−9] These models rely on the use of hundreds of physicochemical descriptors and fingerprints on the biodegradability of compounds and data sets of diverse chemical structures. In particular, artificial intelligence methods have recently become popular modeling approaches. Cheng et al.[10] used four different methods, namely, support vector machine (SVM), *k*-nearest neighbor (kNN), naïve Bayes, and C4.5 decision tree, to build the combinatorial classification probability model of ready biodegradability (RB) versus not ready biodegradability (NRB) using physicochemical descriptors and fingerprints separately. This combinatorial model could correctly predict in an external validation set. However, the external validation set contained only 27 molecules, and some new structural features were not included in this model. After improving the external validation set and structural features, Mansouri et al.[11] applied kNN, partial least-squares discriminant analysis (PLSDA), and SVM as well as their two consensus models to discriminate biodegradable and non-biodegradable chemicals. The first consensus model got results almost similar to those from the three original models, and the second one demonstrated good classification performance with respect to already published models. However, the superior model would not assign molecules that could not be classified into the same class by the three original models. It was

Received: June 1, 2014
Published: August 18, 2014

dx.doi.org/10.1021/ci500323t | J. Chem. Inf. Model. 2014, 54, 2515−2523

inevitable to affect the ranges and results of classification on biodegradation to some extent. In fact, we may arrive at the solution from a different perspective. Besides the combinatorial models, a good method with significant improvement can work well for this problem too.

SVM is a powerful machine learning method based on statistical learning theory and the structural risk minimization principle that has been successfully applied in classification and regression problems.[12] In recent years, it has been widely applied in the biology, medicine, environtology, and chemistry areas and has shown satisfactory efficiency and effectiveness.[13,14] Both Cheng et al.[10] and Mansouri et al.[11] also made use of SVM as a basic modeling method, but they did not pay much attention to the parameter optimization issue of SVM, which might affect the classification and prediction performances directly.[15] Compared with the traditional grid search method, soft computing methods, especially involving evolutionary algorithms, can often obtain better solutions in less time when being used in optimization.[16] Lessmann et al.[17] proposed a meta-strategy utilizing a genetic algorithm (GA) for model selection striving to determine all parameters of the SVM classifier. İlhan and Tezel[18] optimized the C and γ parameters of SVM by using a particle swarm optimization (PSO) algorithm to predict single nucleotide polymorphisms (SNPs) and applied a GA to select tag SNPs. Bai et al.[19] proposed a new algorithm, the parallel artificial fish swarm algorithm (PAFSA), to optimize the kernel parameter and penalty factor of SVM and applied the optimal parameters to a speech recognition system. Aydin et al.[20] presented a multiobjective artificial immune system (AIS) to optimize the kernel and penalize parameters of SVM and applied it to fault diagnosis of induction motors and anomaly detection problems. Ao et al.[21] proposed a SVM parameter optimization method based on an artificial chemical reaction optimization algorithm (ACROA) to diagnose roller bearing faults. Differential evolution (DE) is a popular evolutionary algorithm that is primarily suited for numerical optimization problems and is well-known for its simple and easy-to-understand concept, high convergence characteristics, and robustness. Bhadra et al.[22] devised a metamodel by using SVM as the underlying classifier and optimized its kernel parameters by DE. Kryś et al.[23] applied DE to tune the hyperparameters of a least-squares SVM classifier on a signal-averaged electrocardiography data set. However, the sample data and features of these applications were relatively few, preventing a complete demonstration and validation of the benefits of DE and SVM in complex or high-dimensional environments.

In the present work, DE was introduced into a support vector classifier (SVC) to optimize its model parameters, which included a penalty parameter and other parameters related to the choice of the kernel function. First we adopted the popular radial basis function (RBF) as the kernel function and compared DE versus several other methods to optimize the parameters of the SVC using the available data sets for chemical biodegradability. Detailed comparisons demonstrated the usefulness of DE over other methods. We then considered the problem of whether the RBF is the best choice for the kernel function by using DE with the SVC for various choices of kernel function. Our results suggested that the RBF is indeed the best choice for our present problem. The resulting classifier, which we call the DE-SVC, was then applied to predict the biodegradation of chemicals on the basis of the relatively complete sample data and structural features of molecules. Training, test, and external validation data sets were used in the classification. The performance of the classifier was measured according to its accuracy, sensitivity, and specificity, and the results were compared with published results obtained using kNN, PLSDA, SVM (without optimization), and the consensus models of Mansouri et al.[11] We found that the DE-SVC performed better than those methods in terms of robustness, reliability, classification accuracy, and selectivity. It is an excellent classification method for the prediction of chemical biodegradability.

The paper is organized as follows. In the next section, the fundamental principles of SVC are given. In section 3, the DE algorithm is introduced into the SVC to optimize the main parameters of the classifier, affording the DE-SVC. In section 4, the optimization and classification experiments are presented to illustrate the efficiency and effectiveness of the proposed classifier. Finally, remarks and conclusions are given in section 5.

2. FUNDAMENTAL SUPPORT VECTOR CLASSIFIER PRINCIPLES

In the two-class case, a support vector classifier attempts to locate a hyperplane that maximizes the distance from the members of each class to the optimal hyperplane. Given a set of sample data $\{\mathbf{x}_k, y_k\}_{k=1}^{n}$, where $\mathbf{x}_k \in \mathbf{R}^n$ is an n-dimensional input vector and $y_k \in \{-1, +1\}$ is the binary class label, the classifier is capable of producing a corresponding label y for any given input vector $\mathbf{x}$ that is not in the sample set after the support vector classifier has been trained using the sample set. The training patterns are said to be linearly separable if a weight vector $\mathbf{w}^T$ (which determines the orientation of a discriminating plane) and a scalar b (which determines the offset of the discriminating plane from the origin) can be defined so that the following inequalities are satisfied for all of the sample data:[24]

$$\begin{cases} \mathbf{w}^T\varphi(\mathbf{x}_k) + b \geq +1 & \text{if } y_k = +1 \\ \mathbf{w}^T\varphi(\mathbf{x}_k) + b \leq -1 & \text{if } y_k = -1 \end{cases} \tag{1}$$

The aim is to find a hyperplane that divides the data in such a way that all of the points with the same label lie on the same side of the hyperplane. This problem is equivalent to finding the weight vector $\mathbf{w}^T$ and the bias b that obey the following inequality:

$$y_k(\mathbf{w}^T\varphi(\mathbf{x}_k) + b) \geq 1, \quad k = 1, 2, \cdots, n \tag{2}$$

In this expression, $\varphi(\bullet)$ is in general a nonlinear function from $\mathbf{R}^n \rightarrow \mathbf{R}^h$ that maps the input space into a high-dimensional feature space. It is important to note that the dimension h of this space is defined only in an implicit way and that the space can be infinite dimensional. In primal weight space, the classifier takes the form

$$y(x) = \text{sign}(\mathbf{w}^T\varphi(\mathbf{x}_k) + b) \tag{3}$$

However, it is never evaluated in this form. On the basis of structural risk minimization in statistical learning theory,[25] the classifier is equivalent to solving the following constrained optimization problem:

$$\begin{aligned} &\text{minimize}_{\mathbf{w},b}\, J = \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to } y_k(\mathbf{w}^T\varphi(\mathbf{x}_k) + b) \geq 1, \quad k = 1, 2, \cdots, n \end{aligned} \tag{4}$$

To ensure that the above problem is solvable, slack variables ξ_k are introduced. The optimization problem then takes on the following form:

$$\text{minimize}_{\mathbf{w},b,\xi}\, J = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{k=1}^{n}\xi_k$$
$$\text{subject to}\begin{cases} y_k(\mathbf{w}^T\varphi(\mathbf{x}_k) + b) \le 1 - \xi_k \\ \xi_k \ge 0 \end{cases}, \quad k = 1, 2, \cdots, n \tag{5}$$

where the constant C is the penalty parameter, which must be positive. The proportion of confidence intervals and empirical risk is controlled in feature space by C, which aims at finding the best generalization of the SVC for unseen data. With increasing C, the penalty of the experimental error gets larger. When the complexity of machine learning is high, its experience risk value is small, which leads to overlearning, otherwise known as less learning.

The minimization of $\|\mathbf{w}\|^2$ corresponds to a maximization of the margin between the two classes of training vectors. The Lagrangian is given as follows:[24]

$$L(\mathbf{w}, b, \xi; \alpha, \nu) = J(\mathbf{w}, \xi) - \sum_{k=1}^{n}\alpha_k(y_k(\mathbf{w}^T\varphi(\mathbf{x}_k) + b) - 1 + \xi_k) - \sum_{k=1}^{n}\nu_k\xi_k \tag{6}$$

where $\alpha_k \ge 0$ and $\nu_k \ge 0$ $(k = 1, 2, \cdots, n)$ are the Lagrangian multipliers. The solution is characterized by the saddle point of the Lagrangian:

$$\text{maximize}\, J = \max_{\alpha,\nu}\min_{\mathbf{w},b,\xi} L(\mathbf{w}, b, \xi; \alpha, \nu) \tag{7}$$

Further calculation gives the following results:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{k=1}^{n}\alpha_k y_k \varphi(\mathbf{x}_k)$$
$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{k=1}^{n}\alpha_k y_k = 0$$
$$\frac{\partial L}{\partial \xi_k} = 0 \Rightarrow 0 \le \alpha_k \le C, \quad k = 1, 2, \cdots, n \tag{8}$$

Replacing $\mathbf{w}$ in the Lagrangian by the first expression above yields the following quadratic programming problem:

$$\text{maximize}_{\alpha}\, J = -\frac{1}{2}\sum_{k,l=1}^{n} y_k y_l K(\mathbf{x}_k, \mathbf{x}_l)\alpha_k\alpha_l + \sum_{k=1}^{n}\alpha_k$$
$$\text{subject to}\begin{cases} \sum_{k=1}^{n}\alpha_k y_k = 0 \\ 0 \le \alpha_k \le C, \quad k = 1, 2, \cdots, n \end{cases} \tag{9}$$

where $\mathbf{w}$ and $\varphi(\mathbf{x}_k)$ are not calculated. On the basis of the Mercer condition, the kernel methods deal only with the kernel function (shown in eq 10) rather than the explicit form of φ, which efficiently avoids the complex inner product and the design of the machine itself.

$$K(\mathbf{x}_k, \mathbf{x}_l) = \varphi(\mathbf{x}_k)^T\varphi(\mathbf{x}_l) \tag{10}$$

Finally, in dual space the nonlinear SVC is replaced by the following:

$$y(\mathbf{x}) = \text{sign}\left(\sum_{k=1}^{n}\alpha_k y_k K(\mathbf{x}, \mathbf{x}_k) + b\right) \tag{11}$$

where the α_k are positive real constants and b is a real constant. The nonzero Lagrange multipliers α_k are called support values. The corresponding data points are called support vectors; they are located close to the decision boundary and contribute to the classifier model. The bias b can be computed using the Karush−Kuhn−Tucker (KKT) conditions.

Popular kernel functions in machine learning theory include the following:

the linear kernel function,

$$K(\mathbf{x}, \mathbf{x}_k) = \mathbf{x}_k^T\cdot\mathbf{x}$$

the polynomial kernel function,

$$K(\mathbf{x}, \mathbf{x}_k) = [\gamma(\mathbf{x}_k^T\cdot\mathbf{x}) + r]^d, \quad \gamma > 0$$

the RBF kernel function,

$$K(\mathbf{x}, \mathbf{x}_k) = \exp(-\gamma\|\mathbf{x} - \mathbf{x}_k\|^2), \quad \gamma > 0$$

and the sigmoid kernel function,

$$K(\mathbf{x}, \mathbf{x}_k) = \tanh[\gamma(\mathbf{x}_k^T\cdot\mathbf{x}) + r]$$

where γ, r, and d are kernel function parameters.

3. SUPPORT VECTOR CLASSIFIER OPTIMIZED WITH DIFFERENTIAL EVOLUTION

In the SVC, the penalty parameter C and the kernel function parameters do not result from solving the matrix equations. They are basic parameters of the SVC, and their values, which clearly depend on the sample data, must be specified. Bad choices can lead to drastic deterioration of classification performance. Therefore, in this study the DE algorithm was introduced into the SVC in order to determine these parameter values for a given data set by performing an optimization. We will refer to this procedure as DE-SVC. The basic principle of DE-SVC is to use the DE algorithm to evolve the parameters of the SVC iteratively and to train the SVC using the new parameters in each generation. The fitness value of the SVC is computed and then compared with current best value iteratively until the terminal conditions are satisfied, which results in the proper parameters for the classifier.

The fitness function of our DE-SVC is chosen to be the error estimator of a k-fold cross-validation:[26]

$$e_k(S_n, P, Z) = \frac{1}{n}\sum_{i=1}^{k}\sum_{(\mathbf{f},c)\in P_i} 1(c, \psi_i(\mathbf{f})) \tag{12}$$

where $S_n = \{(\mathbf{f}^{(1)}, c^{(1)}), (\mathbf{f}^{(2)},c^{(2)}), \cdots, (\mathbf{f}^{(n)},c^{(n)})\}$ is the training set, which is randomly partitioned into k folds of similar sizes, $P = \{P_1, P_2, \cdots, P_k\}$ for $i = 1, 2, \cdots, k$; $\mathbf{f} = (f_1, f_2, \cdots, f_d)$ is a features vector, where d is the number of features; c is the class label; $(\mathbf{f}, c)$ is a random vector with a joint feature−label probability distribution; ψ is the SVC that maps $\mathbf{f}$ into c; $1(l, m) = 1$ if $l \ne m$ and zero otherwise; and $Z = \{Z_1, Z_2, \cdots, Z_D\}$ is the set of parameters of the SVC, where D is the number of such parameters. Thus, the error estimator is the average of the errors committed by the classifiers ψ_i in the partitions P_i.

dx.doi.org/10.1021/ci500323t | J. Chem. Inf. Model. 2014, 54, 2515−2523

The basic process of the DE-SVC algorithm used in this study is shown as Figure 1. The individual steps of the algorithm are described in the following paragraphs.

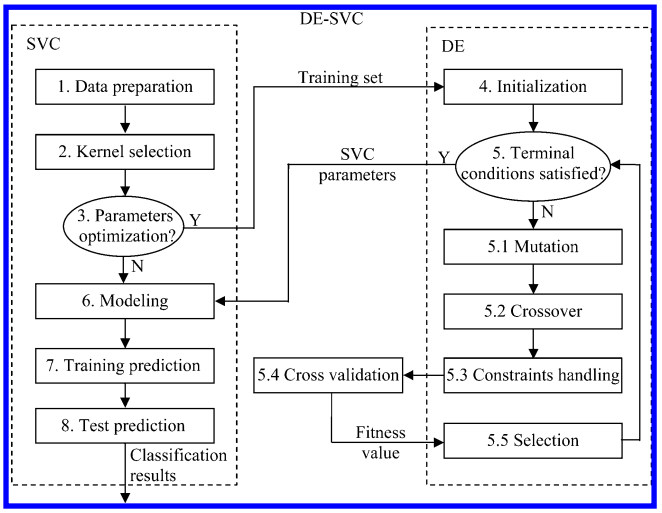


Figure 1. Basic process of the DE-SVC algorithm

Step 1: Data preparation. Essential data preprocessing, including removal of bad data and normalization, is first performed on the full data set S_{full}. Normalization converts all of the data in S_{full}, which includes the features f and the class labels c, into the interval [0, 1] according to the following prescription:

$$f' = \frac{f - f_{min}}{f_{max} - f_{min}}, \quad c' = \frac{c - c_{min}}{c_{max} - c_{min}} \tag{13}$$

S_{full} is then partitioned into the training set S_n and the test set S_t, so $S_{full} = S_n \cup S_t$.

Step 2: Kernel selection. Next, the proper kernel function is selected for the classifier. Such a choice not only influences the number of SVC parameters D but can also impact the performance of the classifier. The RBF was selected here as the kernel function in DE-SVC because of its simplicity. Other choices are certainly possible. A comparative analysis based on different kernel functions using the given data sets may be necessary in order to identify a suitable choice.

Step 3: Branch judgment. This step judges whether the parameters of the SVC must be optimized. If optimization is needed, then a DE optimization is performed on the SVC parameters using the training set S_n; otherwise, optimization is not performed, and the classical SVC is executed with the default set of SVC parameters:

IF *optimize* = TRUE THEN
  Jump to Step 4
ELSE
  Jump to Step 6
END IF

Step 4: Initialization. This step sets the generation number G to zero and randomly initializes a population $\mathbf{P}_G = \{\mathbf{X}_{1,G}, \mathbf{X}_{2,G}, ..., \mathbf{X}_{N_P,G}\}$ containing N_P individuals $\mathbf{X}_{i,G} = \{x^1_{i,G}, x^2_{i,G}, ..., x^D_{i,G}\}$, $i = 1, 2, ..., N_P$, that are uniformly distributed in the range $[\mathbf{X}_{min}, \mathbf{X}_{max}]$, where $\mathbf{X}_{min} = \{x^1_{min}, x^2_{min}, ..., x^D_{min}\}$, $\mathbf{X}_{max} = \{x^1_{max}, x^2_{max}, ..., x^D_{max}\}$, and $N_P = 5D$. These initial values are computed as follows:

$$x^j_{i,G} = \text{rand}_i[0, 1)\cdot(x^j_{max} - x^j_{min}) + x^j_{min}, \quad j = 1, 2, ..., D \tag{14}$$

Step 5: Main loop. The main loop is performed while the terminal conditions are not satisfied. The terminal conditions include two situations: (1) the total number of iterations reaches a prescribed maximum number; (2) the best value of the fitness function (i.e., the error estimator of the k-fold cross-validation computed by eq 12) remains unchanged consecutively for a certain prescribed number of iterations. When either of these two terminal conditions is satisfied, then the optimized set of SVC parameters Z as determined by the currently best solution is sent to the SVC, which is then used to perform modeling and prediction.

WHILE the terminal conditions are not satisfied DO

Step 5.1: Mutation. This step generates a mutated vector $\mathbf{V}_{i,G} = \{v^1_{i,G}, v^2_{i,G}, ..., v^D_{i,G}\}$ for each target vector $\mathbf{X}_{i,G}$:

FOR i = 1 to N_P

$$\mathbf{V}_{i,G} = \mathbf{X}_{r^i_1,G} + F\cdot(\mathbf{X}_{r^i_2,G} - \mathbf{X}_{r^i_3,G}) \tag{15}$$

END FOR

The most popular DE strategy (DE/rand/1/bin) is used here. The indices r^i_1, r^i_2, and r^i_3 are mutually exclusive integers that are randomly generated within the range $[1, N_P]$. These indices are also different from the index i and are randomly generated for each mutant vector. The scaling factor F is a fixed positive control parameter for scaling the difference vector and lies within the range [0, 2].

Step 5.2: Crossover. This step generates a trial vector $\mathbf{U}_{i,G} = \{u^1_{i,G}, u^2_{i,G}, ..., u^D_{i,G}\}$ for each target vector $\mathbf{X}_{i,G}$:

FOR i = 1 to N_P
  $j_{rand} = \lceil \text{rand}[0, 1)\cdot D \rceil$
  FOR j = 1 to D

$$u^j_{i,G} = \begin{cases} v^j_{i,G} & \text{if rand}[0, 1) \le \text{CR or } j = j_{rand} \\ x^j_{i,G} & \text{otherwise} \end{cases} \tag{16}$$

  END FOR
END FOR

The crossover probability CR is another user-defined value that controls the fraction of parameter values that are copied from the mutant. It has a fixed value that lies within the range [0, 1]. The variable j_{rand} is a randomly chosen integer in the range [1, D].

Step 5.3: Constraints handling. This step judges and handles boundary constraint violations in the trial vector $\mathbf{U}_{i,G}$:

FOR i = 1 to N_P
  FOR j = 1 to D
    IF $u^j_{i,G} > x^j_{max}$ THEN

$$u^j_{i,G} = x^j_{max} - \text{rand}[0, 1)\cdot(u^j_{i,G} - x^j_{max})$$

    END IF
    IF $u^j_{i,G} < x^j_{min}$ THEN

$$u^j_{i,G} = x_{min} + \text{rand}[0, 1)\cdot(x^j_{min} - u^j_{i,G})$$

    END IF
  END FOR
END FOR

The boundary constraints are handled via the bounce-back method, which randomly selects a parameter value that lies between the base parameter value and the bound being violated.

Step 5.4: Cross-validation. This step trains the SVC on the basis of the trial vector and computes the value of the fitness function $f(\mathbf{U}_{i,G})$:

FOR $i = 1$ to N_P

$f(\mathbf{U}_{i,G}) = e_k(S_n, P, \mathbf{U}_{i,G})$

END FOR

The elements of the trial vector $\mathbf{U}_{i,G}$ are used as the parameters of the SVC to train S_n with the k-fold cross-validation set P. The error estimator obtained from eq 12 is used as the fitness function value $f(\mathbf{U}_{i,G})$.

Step 5.5: Selection. Here a more appropriate vector is selected by comparing the fitness function values of the trial vector and the target vector:

FOR $i = 1$ to N_P
  IF $f(\mathbf{U}_{i,G}) < f(\mathbf{X}_{i,G})$ THEN
    $\mathbf{X}_{i,G+1} = \mathbf{U}_{i,G}$
    $f(\mathbf{X}_{i,G+1}) = f(\mathbf{U}_{i,G})$
    IF $f(\mathbf{U}_{i,G}) < f(\mathbf{X}_{\text{best},G})$ THEN
      $\mathbf{X}_{\text{best},G} = \mathbf{U}_{i,G}$
      $f(\mathbf{X}_{\text{best},G}) = f(\mathbf{U}_{i,G})$
    END IF
  END IF
END FOR

If the trial vector presents a lower value of the fitness function than the target vector, it will be selected to pass on to the next generation. Otherwise, the original target vector is retained.

(**Main loop iteration**) The generation number is increased by 1: $G = G + 1$.

END WHILE

Step 6: Modeling. In this step, the SVC prediction model is trained using the SVC parameters Z and the given training set S_n:

model = SVMTRAIN(S_n, Z)

Step 7: Training prediction. In this step, predictions of the training set S_n are made on the basis of the SVC model generated by step 6:

$\text{result}_{\text{training}}$ = SVMPREDICT(S_n, model)

Step 8: Test prediction. In this step, predictions of the test set S_t are made on the basis of the SVC model from step 6, and the classification results of both sets of data are returned as output:

$\text{result}_{\text{test}}$ = SVMPREDICT(S_t, model)
OUTPUT($\text{result}_{\text{training}}$, $\text{result}_{\text{test}}$)

4. OPTIMIZATION AND CLASSIFICATION EXPERIMENTS FOR PREDICTION OF CHEMICAL BIODEGRADABILITY

4.1. Experimental Data. As in ref 11, a set of 837 molecules was used for calibration purposes, while 218 molecules were used to test the calibrated classifier. The optimization experiments mainly made use of the combined total of 1055 molecules. In addition, the classifier was further evaluated using an external validation set consisting of 670 molecules and the former 1055 molecules, which constituted the data set for the classification experiments. The number of RB and NRB molecules contained separately in the training, test, and external validation sets are shown in Table 1.

Table 1. Numbers of Molecules Included in the Training, Test, and External Validation Sets[a]

data set	RB	NRB	total
training set	284	553	837
test set	72	146	218
external validation set	191	479	670

[a]Downloaded from http://pubs.acs.org/doi/suppl/10.1021/ci4000213/suppl_file/ci4000213_si_001.xlsx.[11]

4.2. Molecular Features. A set of 41 molecular features was selected to be applied in both the optimization and classification experiments. Symbols and brief descriptions for the molecular features, together with the feature blocks from DRAGON, are collected in Table 2.[11]

4.3. Experimental Environment. Our computational hardware consisted of a 2.4G Hz CPU with 1.92G RAM and a 32-bit OS. The software included GA, PSO, and DE models and the library LIBSVM 3.17, which was compiled and run in MATLAB R2012b.

4.4. Optimization Experiments and Results Analysis. First we performed optimization experiments to measure the performances of four different methods of determining the SVC parameters, namely, the penalty parameter C and the kernel parameter γ. The second parameter resulted because of our choice of the RBF kernel function for use in each of the optimization methods. The four methods that we used were 5-fold cross-validations using a grid search, GA, PSO, and DE. Each of these optimization methods had its own set of fundamental control parameters, and they are shown in Table 3. In the three evolution-based methods, the size of the population was chosen to be 10, and the maximum number of allowed iterations was 100. In addition, the optimization process was terminated if the fitness value of the cross-validation remained unchanged in 20 consecutive iterations. Each method was run 10 separate times using data from either the training set or the test set, as shown in Table 1. The performances of these four optimization methods were measured using the following four different evaluation criteria: best cross-validation accuracy, accuracy on the training set, accuracy on the test set, and total runtime. The detailed experimental results for each of the 10 runs using these four methods are shown in Figure 2. The figure enables us to sense the robustness, reliability, and reproducibility of the methods. The average results for the 10 runs are displayed in Table 3.

Since the traditional grid search method is deterministic, the results obtained for the accuracy for cross-validation, the training set, and the test set did not vary from one run to the next. However, the runtimes were almost 1.2 to 2.8 times more than those of the evolutionary methods, as can be seen in Figure 2d. The method is expected to face a bigger time complexity problem when solving high-dimensional optimization problems or when the data sets become huge.

Among the four optimization methods, GA achieved the highest mean accuracy in the training set but the lowest mean accuracy in the test set, as shown in Figure 2b,c. This may be an indication that it suffered from overfitting. Moreover, GA required the highest mean runtime among the three evolutionary methods.

Table 2. List of Molecular Features Selected for Use in the Classifier

symbol	description	DRAGON block
B01[C-Br]	presence/absence of C−Br at topological distance 1	2D atom pairs
B03[C-Cl]	presence/absence of C−Cl at topological distance 3	2D atom pairs
B04[C-Br]	presence/absence of C−Br at topological distance 4	2D atom pairs
C%	percentage of C atoms	constitutional indices
C-026	R−CX−R	atom-centered fragments
F01[N-N]	frequency of N−N at topological distance 1	2D atom pairs
F02[C-N]	frequency of C−N at topological distance 2	2D atom pairs
F03[C-N]	frequency of C−N at topological distance 3	2D atom pairs
F03[C-O]	frequency of C−O at topological distance 3	2D atom pairs
F04[C-N]	frequency of C−N at topological distance 4	2D atom pairs
HyWi_B(m)	hyper-Wiener-like index (log function) from Burden matrix weighted by mass	2D matrix-based
J_Dz(e)	Balaban-like index from Barysz matrix weighted by Sanderson electronegativity	2D matrix-based
LOC	lopping centric index	topological indices
Me	mean atomic Sanderson electronegativity (scaled on carbon atom)	constitutional indices
Mi	mean first ionization potential (scaled on carbon atom)	constitutional indices
N-073	$Ar_2NH/Ar_3N/Ar_2N-Al/R\cdots N\cdots R$	atom-centered fragments
nArCOOR	number of esters (aromatic)	functional group counts
nArNO2	number of nitro groups (aromatic)	functional group counts
nCb-	number of substituted benzene $C(sp^2)$	functional group counts
nCIR	number of circuits	ring descriptors
nCp	number of terminal primary $C(sp^3)$	functional group counts
nCrt	number of ring tertiary $C(sp^3)$	functional group counts
nCRX3	number of CRX_3	functional group counts
nHDon	number of donor atoms for H-bonds (N and O)	functional group counts
nHM	number of heavy atoms	constitutional indices
nN	number of nitrogen atoms	constitutional indices
nN-N	number of N hydrazines	functional group counts
nO	number of oxygen atoms	constitutional indices
NssssC	number of atoms of type ssssC	atom-type E-state indices
nX	number of halogen atoms	constitutional indices
Psi_i_1d	intrinsic state pseudoconnectivity index−type 1d	topological indices
Psi_i_A	intrinsic state pseudoconnectivity index−type S average	topological indices
SdO	sum of dO E-states	atom-type E-state indices
SdssC	sum of dssC E-states	atom-type E-state indices
SM6_B(m)	spectral moment of order 6 from Burden matrix weighted by mass	2D matrix-based
SM6_L	spectral moment of order 6 from Laplace matrix	2D matrix-based
SpMax_A	leading eigenvalue from adjacency matrix (Lovasz−Pelikan index)	2D matrix-based
SpMax_B(m)	leading eigenvalue from Burden matrix weighted by mass	2D matrix-based
SpMax_L	leading eigenvalue from Laplace matrix	2D matrix-based
SpPosA_B(p)	normalized spectral positive sum from Burden matrix weighted by polarizability	2D matrix-based
TI2_L	second Mohar index from Laplace matrix	2D matrix-based

PSO showed quality performances on the mean accuracy of the test set and on the mean runtime. However, we found that the results tended to vary much more from one run to the next, as can be seen in Figure 2. These fluctuations in the optimization performances would weaken its reliability and independence.

DE achieved the best mean accuracy in cross-validation and for the test set. It also had the lowest runtime. Moreover, the results obtained using DE had relatively much less variation from one run to the next compared with those for PSO and GA, as shown in Figure 2. This indicates the robustness and reliability of DE. Thus, compared with the three other methods that we tried, DE was found to be the most suitable one to optimize the parameters of the SVC.

To ascertain the influence of using different kernel functions on the performance of the classifier for the given data sets, three other kernel functions, namely, the linear kernel, polynomial kernel, and sigmoid kernel, were also tested individually in the optimization experiments. It should be noted that the number of parameters in the kernel functions varies from 0 to 3. In each case, DE was used to optimize the SVC. The statistical results of these experiments are compared with those for the RBF kernel in Table 4.

First, it is not surprising to find that the mean runtimes for the four choices of kernel functions increase with the number of kernel parameters. The linear kernel function has no parameters, so its time complexity was the lowest, leading to the shortest mean runtime. However, the three criteria for classification accuracy were not great. The results were only better than the corresponding ones for the sigmoid kernel, which had the worst performance among four kernels. On the contrary, the polynomial kernel function yielded good classification accuracy but took 7−33 times longer mean runtime than the other kernel functions, which was attributed to its complicated structure with three kernel parameters. The RBF kernel was proved the most suitable kernel function for

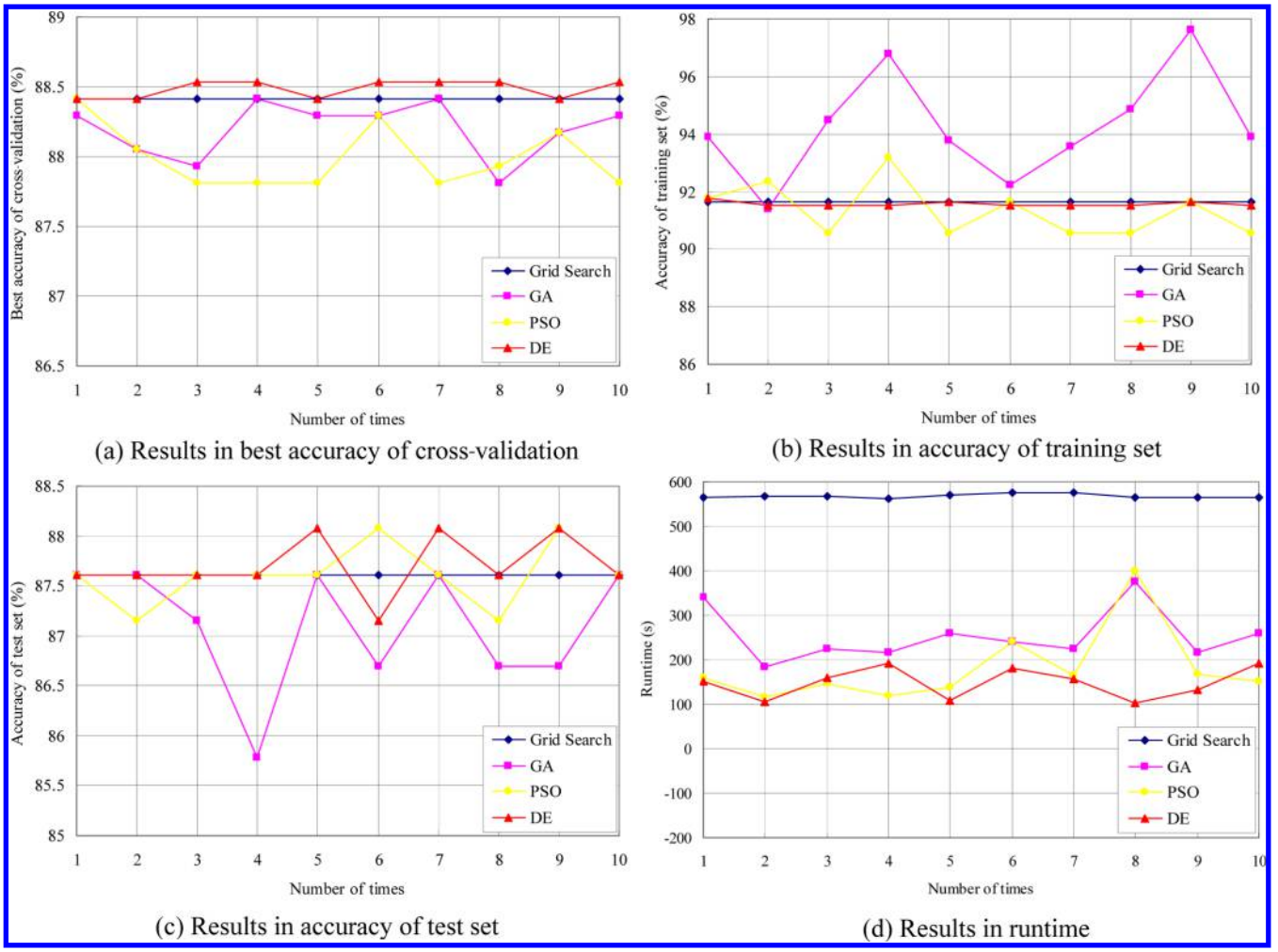


Figure 2. Optimization results of SVC with different parameter selection methods

Table 3. Statistical Results for SVC with Different Parameter Selection Methods

optimization method	main model parameters	mean best accuracy of cross-validation (%)	mean accuracy of training set (%)	mean accuracy of test set (%)	mean runtime (s)
grid search	$C \in [2^{-7}, 2^{7}]$, $\gamma \in [2^{-7}, 2^{7}]$, $C_{step} = 0.4$, $\gamma_{step} = 0.4$	88.41	91.64	87.61	567.46
GA	$C \in [0, 100]$, $\gamma \in [0, 100]$, GGAP = 0.9, $P_c = 0.7$, $P_m = 0.0175$	88.20	94.25	87.11	254.10
PSO	$C \in [0.1, 100]$, $\gamma \in [0.1, 100]$, $k = 0.6$, $c_1 = 1.5$, $c_2 = 1.7$, $w_V = 1$, $w_P = 1$	87.99	91.34	87.61	180.59
DE	$C \in [0, 100]$, $\gamma \in [0, 100]$, $F = 0.7$, CR = 0.9, strategy = DE/rand/1/bin	88.48	91.57	87.71	148.17

Table 4. Statistical Results for DE-SVC with Different Kernel Functions

kernel function	kernel parameters	mean best accuracy of cross-validation (%)	mean accuracy of training set (%)	mean accuracy of test set (%)	mean runtime (s)
linear	none	87.57	89.24	86.70	93.78
polynomial	γ, r, d	88.47	90.90	86.79	3131.76
sigmoid	γ, r	83.31	85.41	83.49	451.52
RBF	γ	88.48	91.57	87.71	148.17

this research. It obtained the best mean accuracy for cross-validation, the training set, and the test set, and the mean runtime was rather small. Thus, we were justified in using the RBF kernel function for the DE-SVC in this work to predict the biodegradation of chemicals in the following classification experiments.

4.5. Classification Experiments and Results Analysis. Our next study used DE-SVC as our classifier to measure its performance in the prediction of chemical biodegradability. The results obtained here were compared with those already published in ref 11. First, the DE-SVC was used to predict the chemical biodegradability of the training set and the test set in Table 1. The classifier was run 10 times as described before. The mean results obtained here are compared with those from previous studies in Table 5. Next, we used the DE-SVC with the former data for training as well as the external validation set in Table 1 to further test the predictive power of our classifier. The mean results for 10 separate runs are shown in Table 6. The predictive performance was measured according to three

Table 5. Classification Results for DE-SVC Compared with Those for Other Classifiers on the Training and Test Sets

	training set			test set		
classification model	Acc	Sn	Sp	Acc	Sn	Sp
kNN	0.86	0.84	0.89	0.85	0.81	0.90
PLSDA	0.86	0.88	0.83	0.85	0.83	0.87
SVM	0.86	0.81	0.92	0.86	0.82	0.91
consensus 1[a]	0.89	0.86	0.91	0.87	0.82	0.92
consensus 2[b]	0.93	0.91	0.95	0.91	0.88	0.94
	(19% not assigned)			(15% not assigned)		
DE-SVC	0.92	0.86	0.94	0.88	0.77	0.93

[a]Each molecule was assigned to the most frequent class out of the three predictions obtained from previously published results using the kNN, PLSDA, and SVM methods. [b]A molecule was assigned only if all the three models (kNN, PLSDA, and SVM) classified it in the same class; otherwise, it was not assigned.

Table 6. Classification Results for DE-SVC Compared with Those for Other Classifiers on the External Validation Set

classification model	Acc	Sn	Sp
kNN	0.83	0.75	0.91
PLSDA	0.83	0.80	0.86
SVM	0.82	0.74	0.91
consensus 1	0.83	0.76	0.91
consensus 2	0.87	0.81	0.94
		(13% not assigned)	
DE-SVC	0.88	0.74	0.93

distinct criteria: classification accuracy (Acc), sensitivity (Sn), and specificity (Sp), as defined in the following way:

$$\mathrm{Acc} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

$$\mathrm{Sp} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \quad (17)$$

In these definitions, TP, TN, FP, and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively. It should be mentioned that the classification accuracy is the most important one among the three evaluation criteria. The results for different data sets are shown in Figure 3.

First we discuss the results obtained from the training and test data sets. The overall classification performance of the DE-SVC for the training and test data sets was found to be the best compared with the other computation methods reported in ref 11. The classification accuracy of the DE-SVC was better than those of the three single models kNN, PLSDA, and SVM (without optimization) as well as consensus model 1, where each molecule was assigned to the most frequently chosen class obtained from the three single models. For the training and test data sets, it was only slightly worse than the result of consensus model 2, where each molecule was assigned to the class given by the three single methods if they all agreed and otherwise no assignment was made. In fact, consensus model 2 failed to make any classification for 19% of the training data and 15% of the test data.

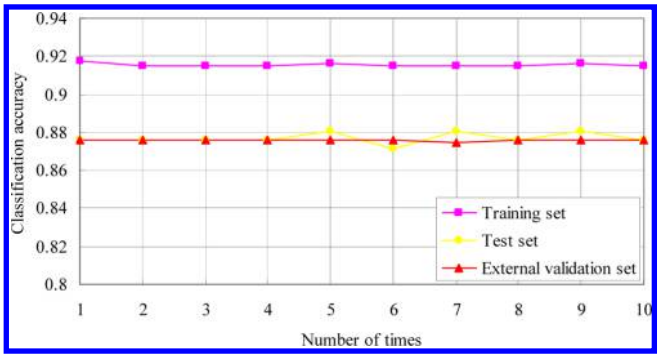


Figure 3. Classification accuracy results for DE-SVC on different data sets.

The sensitivity of the DE-SVC was comparable to those of the other models for the training data set but trailed all of the others for the test data set. It should be pointed out that sensitivity is not an important issue for the problem of biodegradability at all.

The specificity of the DE-SVC was better than those of the other models, with the exception of consensus model 2 (which failed to provide any classification at all for quite a fraction of the molecules), for both the training and test data sets. Even then, it trailed by only 1% for either data set.

Next we consider the external validation data set. The DE-SVC was also found to have better accuracy than all of the other models in the literature, including consensus model 1 and consensus model 2, which failed to make any classification at all for 13% of the cases.

Another very important feature that we have found to favor the DE-SVC is the robustness in the results. No matter whether we were dealing with the training, the test set, or the validation data set, the results for the accuracy, sensitivity, and specificity varied little from one run to the next. This can be seen from Figure 3 and Table S3 in the Supporting Information. This suggests that the results obtained from the DE-SVC are more reliable.

The DE-SVC performed very well with respect to both accuracy and specificity, but it did not perform so well for the sensitivity. However, the specificity is probably more important than the sensitivity in the prediction of chemical biodegradability of molecules. Mistakes of classifying biodegradable molecules as nonbiodegradable can be more tolerated than mistakes in which nonbiodegradable molecules are classified as biodegradable, since that would introduce more harmful chemicals into our environment.

In general, our proposed DE-SVC had similar or better classification performances with respect to the models already published in ref 11. In particular, the DE-SVC in this study showed balanced results on the training, test, and external validation sets, which proved it to be an effective and efficient prediction model for chemical biodegradability. The high and stable classification accuracy in the experiments further demonstrated the strong reliability and robustness of the DE-SVC.

5. CONCLUSIONS

In this study, the DE algorithm was introduced into the SVC to optimize the parameters of the classifier in order to produce an improved classifier called DE-SVC. Extensive experimentation using this new classifier was carried out on different data sets to see how it would perform in chemical biodegradability classification. Detailed comparisons with previous results obtained using a variety of other classifiers were performed.

The DE-SVC was found to outperform all of the other classifiers with the exception of consensus model 2 with respect to both accuracy and specificity. Even so, the DE-SVC performances were only 1% lower than those of consensus model 2. This is very impressive since consensus model 2 was unable to make any classification of more than 10% of the samples in the data sets. Moreover, the results obtained from the DE-SVC varied little from one run to the next. Thus, it is a reliable, robust model suitable for use in chemical biodegradability classification. Future research will pay attention to the improvement of the classification performance with respect to the sensitivity of the classifier.

■ ASSOCIATED CONTENT

Supporting Information

Detailed results of the optimization (Table S1), kernel function (Table S2), and classification (Table S3) experiments. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: roy1976@163.com. Mailing address: Department of Training, Logistical Engineering University, College Town, Shapingba District, Chongqing 401311, China. Telephone: +86-18602351201. Fax: +86-23-86730370.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The research work was supported by the State Scholarship Fund of China under Grant 201203170027, the National High Technology Research and Development Program of China (863 Program) under Grant 2012AA063501, and the Fundamental and Advanced Research Project of Chongqing under Grant cstc2014jcyjA40008.

■ ABBREVIATIONS

SVC, support vector classifier; DE, differential evolution; GA, genetic algorithm; PSO, particle swarm optimization; kNN, *k*-nearest neighbor; PLSDA, partial least-squares discriminant analysis; SVM, support vector machine; RBF, radial basis function; Acc, accuracy; Sn, sensitivity; Sp, specificity.

■ REFERENCES

(1) U.S. Environmental Protection Agency. Pollution Prevention Act of 1990. http://www.epa.gov/p2/pubs/p2policy/act1990.htm (accessed June 15, 2011).

(2) European Commission. Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH), 2007. http://ec.europa.eu/enterprise/sectors/chemicals/reach/index_en.htm (accessed Sept 17, 2013).

(3) Boethling, R. S.; Sommer, E.; DiFiore, D. Designing small molecules for biodegradability. *Chem. Rev.* **2007**, *107*, 2207−2227.

(4) Howard, P. H.; Boethling, R. S.; Stiteler, W. M.; Meylan, W. M.; Hueber, A. E.; Beauman, J. A.; Larosche, M. E. Predictive model for aerobic biodegradability developed from a file of evaluated biodegradation data. *Environ. Toxicol. Chem.* **1992**, *11*, 593−603.

(5) Hiromatsu, K.; Yakabe, Y.; Katagiri, K.; Nishihara, T. Prediction for biodegradability of chemicals by an empirical flowchart. *Chemosphere* **2000**, *41*, 1749−1754.

(6) Hou, B. K.; Wackett, L. P.; Ellis, L. B. Microbial pathway prediction: A functional group approach. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1051−1057.

(7) DeLisle, R. K.; Dixon, S. L. Induction of decision trees via evolutionary programming. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 862−870.

(8) Philipp, B.; Hoff, M.; Germa, F.; Schink, B.; Beimborn, D.; Mersch-Sundermann, V. Biochemical interpretation of quantitative structure−activity relationships (QSAR) for biodegradation of N-heterocycles: A complementary approach to predict biodegradability. *Environ. Sci. Technol.* **2007**, *41*, 1390−1398.

(9) Andreini, C.; Bertini, I.; Cavallaro, G.; Decaria, L.; Rosato, A. A simple protocol for the comparative analysis of the structure and occurrence of biochemical pathways across superkingdoms. *J. Chem. Inf. Model.* **2011**, *51*, 730−738.

(10) Cheng, F.; Ikenaga, Y.; Zhou, Y.; Yu, Y.; Li, W.; Shen, J.; Du, Z.; Chen, L.; Xu, C.; Liu, G.; Lee, P. W.; Tang, Y. In silico assessment of chemical biodegradability. *J. Chem. Inf. Model.* **2012**, *52*, 655−669.

(11) Mansouri, K.; Ringsted, T.; Ballabio, D.; Todeschini, R.; Consonni, V. Quantitative structure−activity relationship models for ready biodegradability of chemicals. *J. Chem. Inf. Model.* **2013**, *53*, 867−878.

(12) Vapnik, V. N. *Statistical Learning Theory*; Wiley: New York, 1998.

(13) Zięba, M.; Tomczak, J. M.; Lubicz, M.; Świątek, J. Boosted SVM for extracting rules from imbalanced data in application to prediction of the post-operative life expectancy in the lung cancer patients. *Appl. Soft Comput.* **2014**, *14*, 99−108.

(14) Lucas, D. D.; Klein, R.; Tannahill, J.; Ivanova, D.; Brandon, S.; Domyancic, D.; Zhang, Y. Failure analysis of parameter-induced simulation crashes in climate models. *Geosci. Model Dev. Discuss.* **2013**, *6*, 585−623.

(15) Li, J.; Ding, L.; Xing, Y. Differential evolution based parameters selection for support vector machine. In *Proceedings of the 9th International Conference on Computational Intelligence and Security, Chengdu, China, Dec. 14−15, 2013*; pp 284−288.

(16) Hong, W. C. *Intelligent Energy Demand Forecasting*; Springer: New York, 2013.

(17) Lessmann, S.; Stahlbock, R.; Crone, S. F. Genetic algorithms for support vector machine model selection. In *Proceedings of the 2006 International Joint Conference on Neural Networks, Vancouver, Canada, July 16−21, 2006*; pp 3063−3069.

(18) İlhan, İ.; Tezel, G. A genetic algorithm−support vector machine method with parameter optimization for selecting the tag SNPs. *J. Biomed. Inf.* **2013**, *46*, 328−340.

(19) Bai, J.; Yang, L.; Zhang, X. Parameter optimization and application of support vector machine based on parallel artificial fish swarm algorithm. *J. Software* **2013**, *8*, 673−679.

(20) Aydin, I.; Karakose, M.; Akin, E. A multi-objective artificial immune algorithm for parameter optimization in support vector machine. *Appl. Soft Comput.* **2011**, *11*, 120−129.

(21) Ao, H.; Cheng, J.; Yang, Y.; Truong, T. K. The support vector machine parameter optimization method based on artificial chemical reaction optimization algorithm and its application to roller bearing fault diagnosis. *J. Vib. Control* **2013**, DOI: 10.1177/1077546313511841.

(22) Bhadra, T.; Bandyopadhyay, S.; Maulik, U. Differential evolution based optimization of SVM parameters for meta classifier design. *Procedia Technol.* **2012**, *4*, 50−57.

(23) Kryś, S.; Jankowski, S.; Piątkowska-Janko, E. Application of differential evolution for optimization of least-square support vector machine classifier of signal-averaged electrocardiograms. *Proc. SPIE* **2009**, *7502*, 75022T.

(24) Suykens, J. A. K. Nonlinear modelling and support vector machines. In *Proceedings of the 18th IEEE International Conference on Instrumentation and Measurement Technology, Budapest, Hungary, May 21−23, 2001*; pp 287−294.

(25) Vapnik, V. N. *The Nature of Statistical Learning Theory*, 2nd ed.; Springer: New York, 2000.

(26) Rodríguez, J. D.; Pérez, A.; Lozano, J. A. Sensitivity analysis of *k*-fold cross validation in prediction error estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 569−575.