Machine Learning for the Classification of Toxicological Endpoints

Ruby Abrams, Nina Galanter, Denise Harness, and Channing Parker

May 20, 2017

1 Abstract

The Toxicology Reference Database (ToxRefDB), compiled by the Environmental Protection Agency (EPA) contains data on chemicals and their associated toxicological endpoints. However, this data does not contain all chemicals of interest, and further testing is resource intensive. Here we present machine learning methods used to predict whether substances will have toxic effects on rat test subjects in order to avoid further animal testing. Chemical features associated with each chemical are utilized to generate these predictions. Support Vector Machine and Decision Tree machine learning algorithms are applied to toxicology data sets provided by the Environmental Protection Agency. These methods are tested and improved through cross-validation, parameter optimization, and the committee of machines approach. Feature selection is employed to optimize the models and provide information on which chemical features are potentially relevant to toxicological effects. Feature selection methods implemented include PCA, ROC curves, and F-Scores for pre-processing, and sensitivity analysis for post-processing. Long term outcomes of this study are to support further research in reducing the amount of animal testing, as well as in developing mechanistic-based toxicological models.

2 Introduction

The Toxicology Reference Database (ToxRefDB), compiled by the Environmental Protection Agency (EPA) contains data on chemicals and their associated toxicological endpoints. Examples including organ weights, life observations, and pathologies[1]. However, the data is not exhaustive. Animal testing to generate further data is resource intensive. We aim to use machine learning methods in order to create Quantitative Structure–Activity Relationship (QSAR) classification models that predict if a chemical has an impact on categories of toxicological endpoints, given structural properties of the chemical [25]. The four endpoint categories we use are Developmental Reproductive, Nonproliferative Pathology, Proliferative Pathology, and Neoplastic Pathology. Standardized techniques are used on each of these four models. Each model is optimized through pre-processing and post-processing feature selection techniques, cross-validation, and the committee of machines approach.

3 Background

Machine learning is a field of study that involves pattern recognition and computational learning theory in artificial intelligence. Machine learning classification techniques are used to build models from “training sets” of input and output observations in order to make output predictions when new observations are introduced [14]. There are numerous machine learning methods that can be utilized in order to create these models. Machine learning models are used for facial recognition, sentiment analysis, and other real world situations [7, 27]. In this project, we present machine learning methods for the prediction of four categories of toxicological endpoints.

3.1 Machine Learning Algorithms

Five different machine learning algorithms are initially analyzed: k-Nearest Neighbors (kNN), Artificial Neural Networks (ANN), Linear Discriminant Analysis (LDA), Decision Trees, and Support Vector Machine (SVM).

3.1.1 k-Nearest Neighbors

The k-nearest neighbors method views each observation as a point in the coordinate space of the features. Using a distance metric, the algorithm will classify a new observation by finding its k closest neighbors, for some positive integer k, and then using the mode of their classes to as the predicted class [29].

[1] The data used in this project can be found at `https://catalog.data.gov/dataset/toxicity-reference-database`

3.1.2 Artificial Neural Networks

Artificial Neural Networks are much like neural networks found in the brain. ANN's work well with non-linear, dynamic relationships that are difficult to describe with conventional approaches. ANN's also work well with large data sets. ANN consists of an input layer of nodes, hidden layer(s) of nodes, and an output layer of nodes [31]. Each connection of two nodes, known as an edge, has a numerical weight associated with it.

Artificial Neural Networks produce continuous outputs. Thus, when predicting discrete outcomes, built-in functions such as sign or round must be used to convert the predictions. The greatest downfall of Artificial Neural Networks is the computational cost. Of the five methods, ANN consistently has the longest run time.

3.1.3 Linear Discriminant Analysis

Linear discriminant analysis is a binary classifier. When given traing data, LDA finds an optimal line through the feature space on which to project each observation [4]. This line maximizes the separation between class means divided by the in—class variance. LDA then calculates an optimal threshold to classify the projected values by minimizing the expected misclassification cost [17]. LDA uses the threshold to predict the class of new observations.

3.1.4 Decision Trees

A Decision Tree is a binary tree that, like the other methods presented, predicts an outcome given a set of input values. Each internal node in the tree represents a test that is applied to one of the input values, and the tree splits depending on the outcome of each test. The tree ultimately partitions the data into cells based on the outcome of these tests [1]. At the end of tree are the leaves, which represent the predictions [26]. The final predictions are the averages of all of the internal node values in the path that leads to that leaf, and each internal node value is the average of all values in that cell [21].

There are two methods that can be used when implementing Decision Trees, a classification tree or a regression tree. A classification tree predicts categorical, or discrete, outcomes, and a regression tree predicts continuous outcomes [28]. Regression trees can also be manipulated to predict categorical outcomes by rounding the predictions, much like ANN.

3.1.5 Support Vector Machines

Like LDA, Support Vector Machine (SVM) is a binary classifier, meaning that SVM classifies an unknown input into one of two output classes [23]. SVM reads in two sets of data, an $m \times 1$ training vector, and an $m \times n$ feature matrix that describes m chemicals with n features. For data that lives in an n-dimensional feature space, R^n, SVM makes a R^{n-1} hyper-plane that would separate the data into the two output categories.

In a feature set that has two distinct classes of data, there is more than one hyper-plane that could be used to separate the data. However, the best fit hyper-plane is one that separates the data points to best account for error. In practice, if a hyper-plane sits too close to a class of data points, a new data point of that class could be misclassified. This is a quadratic programming problem that requires nonlinear constraints [13].

When it is not possible to find a hyper-plane, the data is said to be non-separable. In cases of non-separable data, a Kernel function is used to transform the input data space into a feature space such that an optimal hyper-plane can be found. Options for kernel functions in MATLAB are linear, Gaussian, and polynomial. A custom kernel function can also be used if these methods do not fit the data. We chose to implement a Gaussian kernel function, which is a Radial-Basis function of the form $\phi = e^{-\gamma \|x_i - x_j\|}$, where γ is a parameter that aids in preventing overfitting and x_i, x_j are any two observations in our data set.

4 Data

The ToxRefDB data contains the results of studies on the observed impacts of chemicals on various species. Our research focuses on results from chronic adult rat studies. Specifically, this project uses data which contains the minimum dosage of a chemical that elicits an effect in one of the four endpoint categories of interest. This information was converted into binary data based on whether a chemical had an effect on a given endpoint. Chemicals with minimum dosages of one million were considered to have no effect, and were therefore listed as zeros. Chemicals with minimum dosages less than a million were consider to have an effect, and were therefore listed as ones. Of the chemicals studied, we analyze the 485 unique chemicals which were tested for all endpoint categories of interest. The PaDEL[2] chemical descriptor software generated 1,444 molecular features associated with each chemical. Examples of features generated include the number of hydrogen atoms, the number of bonds, and the pH.

The set of 1,444 chemical features contains features that have the same value across all of the tested chemicals. These features are therefore not informative to our models. Thus, these features were eliminated from the feature set, reducing the features from 1,444 to 1,222.

[2] The chemical features were extracted using PaDEL software found here http://www.yapcwsoft.com/dd/padeldescriptor/

A portion of the machine learning algorithms use distance measurements for classification [9,29]. In order to ensure the algorithms weigh all features equally, the input values need to be on the same scale for each feature. The chemical features for the data use different scales, and thus normalization ensures all features are considered equally and likely improves accuracy [18]. To normalize the data, for each feature the minimum value is subtracted from all values and all values are divided by the range of the feature.

5 Methodology

The 485 observations are split into sets of 435 training data and 50 testing data. The training data is used to create each model, and the testing data is inputted into each model to test the predictive accuracy. The predictions from the testing data are compared to the true testing data results, and the accuracy is obtained by dividing the number of correct predictions by the total number of predictions. All machine learning algorithms use the MATLAB machine learning toolbox except for SVM, which uses the LIBSVM library[3].

Method selection is conducted after initial testing shown in Table 1 of the Results section. In addition to the initial results, other considerations for selection of methods to utilize are the computational expense of Artificial Neural Networks coupled with the limited computation resources of this project, and low performance of kNN for large feature sets or irrelevant features [3, 20]. In light of the initial testing and these other considerations, the set of algorithms has been refined to Decision Trees and Support Vector Machines. Using these two methods, we then attempt to improve our accuracy using oversampling, cross validation, the committee of machines approach, and feature selection techniques.

5.1 Cross Validation

Cross validation is a method of assessing the performance of each predictive model. This method takes in the training observations, and splits this data into "known" data to train the model on and "unknown" data to test the model [19]. The process is then repeated with different splits. Cross validation can also be used for parameter optimization [30]. We used 10-fold cross validation with 90% known data and 10% unknown data repeated ten times.

[3] Documentation for the LBSVM library can be found at `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

5.2 Parameter Optimization

Decision trees and SVM both have parameters that can be optimized in order to improve accuracy.

Decision Trees can be optimized by manipulating the tolerance or cost parameters for making a split in the tree. Tolerance is defined for regression trees as the allowed quadratic error per node. The splitting of the tree halts "when the quadratic error per node drops below the tolerance multiplied by the quadratic error for the entire data." The default tolerance in MATLAB is 1e-6. Cost is defined for classification trees as the cost of misclassifying a point into the wrong class [16]. Both parameters described attempt to reduce the misclassification cost of making another split [6]. A line search algorithm is used to loop through various tolerance or cost levels. The tolerance or cost is then chosen based on the value that creates the highest cross validation accuracy.

SVM, using the Guassian kernel, can be optimized by manipulating two parameters, C and γ. The parameter C is known as the box constraint or the regularization parameter, which controls the maximum penalty imposed on margin violating observations and aids in preventing overfitting. Increasing the box constrain leads the SVM classifier to assign fewer support vectors. However, increasing the box constraint can lead to longer training times. The parameter γ is involved in the Kernel function. This parameter is a distance measure that defines the training points' influence on the hyper-plane [9]. The smaller the distance of a training point from the hyper-plane, the greater the influence of this point is in determining the optimal hyper-plane.In order to optimize these parameters, a grid search algorithm is utilized. A grid search takes a range of combinations of the two parameters and assesses the cross validation accuracy of each pair. The pair with the highest accuracy is then chosen for the model [9] (See Figure 1). Our SVM algorithm uses a search range of [0.01,1.01] with a step size of .2 for γ and a search range of [1,101] with a step size of 20 for C.

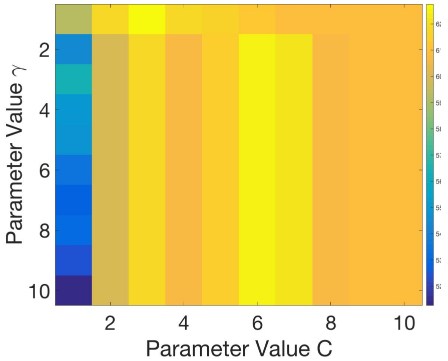

Figure 1: This heat map shows the accuracy of an SVM model trained with given parameters C and γ.

5.3 Committee of Machines Approach

We used the committee of machine approach as a way of reducing the variation across predictions. This approach involves training several different models [15], using the same machine learning method, on the set of 435 training data. Each model is then used to predict the set of 50 testing data. As the predictions for the four endpoint models are binary, the mode of the committee predictions is taken for the final prediction. These predictions are then used to calculate the accuracy.

In Decision Trees, using a random forest method is a way of implementing the committee of machines approach. A random forest method creates an ensemble of predictive trees and takes the mean of each of the predictions in order to get a consensus of multiple committee members' votes [12]. This method was implemented to create more accurate predictive models.

5.4 Oversampling

For each endpoint, the data is mildly to extremely unbalanced in terms of the effect or no effect outcome. Unbalanced data will make the models more inaccurate and difficult to analyze [11]. To balance the data, a random oversampling method is used. Different percentages for oversampling were chosen for each of the four models. Oversampling creates 1,305 total observations. Increasing the number of observations relative to the size of the feature set is expected to increase the predictive accuracy of the models [11].

Oversampling increases the cross validation accuracies of our different models. This occurs because when a 90-10% split is made of data that has repeated entries, it is more likely to correctly classify an entry that the model has already seen.

5.5 Feature Selection

A large number of input variables, as is the case with our data, can lead to poor performance due overfitting of the model to the training data as well as high dimensionality issues [10]. Thus, using feature selection methods to filter out irrelevant chemical features and determine the most significant chemical features can reduce the issue of overfitting and ultimately improve the accuracy of the predictive models.

5.5.1 Pre-Processing

Pre-processing techniques are ways of getting rid of unimportant features. Here we present three pre-processing techniques, Principal Component Analysis (PCA), F-scores, and Receiver Operating Characteristic (ROC) curves, that are utilized on this data.

Principal Component Analysis

Principal component analysis is a transformation procedure that changes correlated features into linearly uncorrelated features, known as principle components, which are eigenvectors of the covariance matrix of the variables. Principle components are linear combinations of the original features. The principle components with larger eigenvalues have larger variance and therefore explain the largest amount of the variance in the data. Considering only the principle components with large eigenvalues will reduce the number of features while preserving much of the information contained in the features [2].By choosing the principle components with associated eigenvalues above one, the feature set is reduced from 1,222 features to 220 features. The classification algorithms are tested using these 220 principal components.

F-Scores

An F-score is a measurement of the discrimination, or ability to classify data in a binary way, between two sets of numbers . Equation 1 calculates the F-score for the ith feature, where $x_{k,i}$ is the kth observation of feature i, $n^{(+)}$ ($n^{(-)}$) is the count of all positive (negative) observations, $\overline{x}_i^{(+)}$ ($\overline{x}_i^{(-)}$) is the mean of all observations in the positive (negative) class, and $\overline{x}_i$ is the overall mean of observations. A larger F-score is indicative of a more discriminative feature. Therefore, using features with only higher F-Scores reduce the dimensionality of the feature space while keeping informative features. We choose features with an F-score above any values stored as zero. One downside to F-Scores is that they do not reveal any mutual information amongst features. [8].

$$F(i) = \frac{\left(\overline{x}_i^{(+)} - \overline{x}_i\right)^2 + \left(\overline{x}_i^{(-)} - \overline{x}_i\right)^2}{\frac{1}{n^{(+)}-1}\sum_{k=1}^{n^{(+)}}\left(x_{k,i}^{(+)} - \overline{x}_i^{(+)}\right)^2 + \frac{1}{n^{(-)}-1}\sum_{k=1}^{n^{(-)}}\left(x_{k,i}^{(-)} - \overline{x}_i^{(-)}\right)^2} \tag{1}$$

ROC Curve

The Receiver Operating Characteristic Curve (ROC) measures the trade-off between sensitivity and specificity for a binary test. Given some variable, the ROC curve is plotted by varying the threshold to declare a positive result, and plotting the true positive rate by the false positive rate for each threshold. As the threshold increases, the rate of false positives will decrease, but so will the rate of true positives. Similarly, decreasing the threshold will result in an increase in true and false positives [24]. The ROC Curve can be used to evaluate a test through estimating the area under the curve (AUC) [5]. Treating each feature as the test variable, the AUC shows how well the feature classifies the data. In our research we estimate the AUC using the trapezoid method. We select features with an AUC value larger than .5.

5.5.2 Post-Processing

Post-processing techniques are methods which selecting important features, using the machine learning algorithms.

Sensitivity Analysis on Decision Trees

The OOBPermutedVarDeltaError output is a measure of feature importance generated by the MATLAB decision tree toolbox, which calculates the amount of error resulting from varying feature input values. Higher error is indicative of a more important feature. We generate OOBPermutedVarDeltaError values for all features using the oversampled full feature set, as well the three feature sets reduced by ROC curves, F-score, and PCA. We find the average error of each of the 1,222 features across ten treebags for each of the four models. The four models are then run on the set of features associated with the top 30% of errors. Figure 2 shows the distribution of errors found across each of the four endpoint models.

Sensitivity Analysis on Support Vector Machines

The method we use for sensitivity analysis on the Support Vector Machine models estimates the derivative of the the change in decision values with respect to the change in values of a feature. Decision values are scalar values SVM assigns to each observation before making a final classification. The derivative is estimated using the central difference method. For each feature f, the decision values for the $h = .001$ added to the original values and $h = .001$ subtracted from the original values using the p prediction function for each observation $x_{i,m,f}$, the ith observation predicted by the mth machine. The derivatives are averaged across the committee of machines, and the absolute value of the derivatives are averaged across observations. Finally, the sensitivity scores are compared in order to find the most significant features. The sensitivity for feature f is calculated using equation 2. Figure 3 shows the distribution of sensitivity scores across each of the four models.

$$S_f = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{1}{m}\sum_{j=1}^{m}\frac{p(x_{i,m,f}+h)-p(x_{i,m,f}-h)}{2h}\right| \tag{2}$$

Sensitivity Results

Below are the results of running sensitivity analysis across all chemical features. The values are skewed to the right, meaning that most of the chemical features had a corresponding sensitivity score of nearly zero, meaning that these features were not as important in predicting the toxicological endpoints of interest (See Figures 2 and 3).

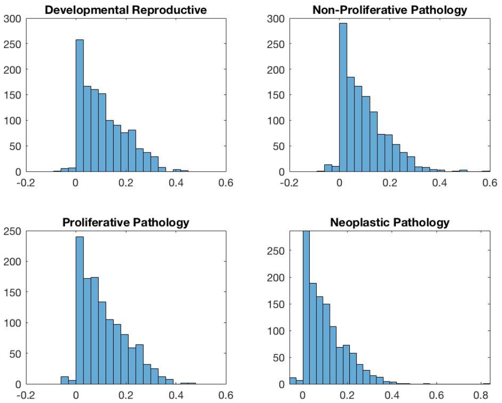

Figure 2: The distribution of the errors calculated from sensitivty analysis of the decision tree models for Developmental Reproductive, Non-Proliferative Pathology, Proliferative Pathology, and Neoplastic Pathology.

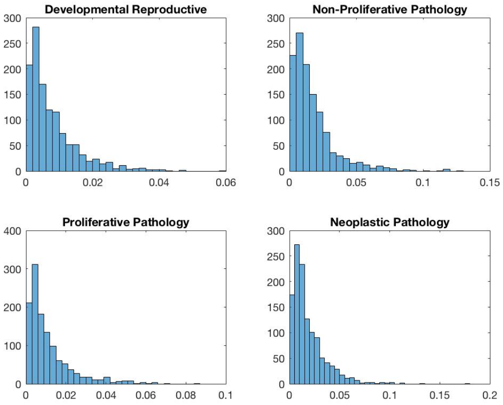

Figure 3: The distribution of sensitivity scores of the support vector machine models for Developmental Reproductive, Non-Proliferative Pathology, Proliferative Pathology, and Neoplastic Pathology.

5.6 Selected Features

Table 1 lists the amount of features which are selected by both pre-processing feature selection methods. Table 1 also contains the overlapping features selected by both the pre-processing and post-processing methods combined. For each comparison of methods, the size of the set of features selected by each post-processing technique is equal to the size of the set of features selected by each pre-processing techniques. An example of a feature selected by SVM and F-scores for all endpoints is number of nitrogen atoms. Examples of features given ROC curve and decision trees for all endpoints include Atomic logP and autocorrelation.

Table 1: Size of the Intersection between Feature Selection Methods

	Devlopmental	**Proliferative**	**Non-Proliferative**	**Neoplastic**
Roc ∩ Fsc	287	188	68	39
SVM ∩ Fsc	58	181	118	90
SVM ∩ Roc	874	452	193	71
DT ∩ Fsc	62	119	85	83
DT ∩ Roc	254	209	144	128

Intersections are listed as counts of features. Decision Tree Sensitivity abbreviated as DT, SVM Sensitivity abbreviated as SVM, F-score abbreviated as Fsc.

6 Results

The tables presented in this section show the predictive accuracies of each of the machine learning methods used for each in of the four endpoint models. The first percentage in each cell of the tables is the cross validation accuracy, and the second is the out of sample prediction accuracy. The best guess accuracy is the percent correct predictions obtained by simply guessing the effect that occurs most often. The bold numbers are the prediction scores of the models that met or exceeded the out of sample best guess accuracy. Each experiment was completed with 10-fold cross validation.

6.1 Initial Results

In order to asses the performance of each of the initial six machine learning methods, each of the endpoint models are evaluated using the algorithms without any pre-processing or post-processing.

Table 2 displays the accuracy results for each of the machine learning models predicting an effect or no effect for the Developmental Reproductive, Proliferative Pathology, Non-Proliferative Pathology, and Neoplastic Pathology endpoints categories. The top two methods, Decision Trees and Support Vector Machines, were then analyzed further based on highest accuracy and lowest run time.

Table 2: Initial Results Across all Six Machine Learning Methods

Endpoint	**Devlopmental**	**Proliferative**	**Non-Proliferative**	**Neoplastic**
Best Guess	60% / 72%	72% / 80%	50% / 60%	66% / 60%
R-Tree	58% / 64%	71% / **80%**	53% / 48%	68% / 54%
C-Tree	58% / **72%**	67% / 68%	54% / 46%	67% / 52%
SVM	62% / **72%**	72% / **80%**	58% / 46%	67% / 58%
ANN	80% / 68%	53% / 78%	51% / 52%	61% / **62%**
kNN	62% / 66%	72% / **80%**	56% / 48%	68% / 54%
LDA	52% / 70%	53% / 54%	53% / 50%	60% / 50%

6.2 Oversampling Results

Table 3 below contains the accuracy results for each of the chosen machine learning algorithms, Decision Trees and Support Vector Machines, predicting effect or no effect on the four toxicological endpoints. The models used to obtain these results are based on the oversampled data sets without feature selection.

Table 3: Oversampling Results Across Top Machine Learning Methods

Endpoint	**Devlopmental**	**Proliferative**	**Non-Proliferative**	**Neoplastic**
Best Guess	60% / 72%	50% / 80%	50% / 60%	50% / 60%
R-Tree	97% / 64%	97% / 69%	97% / 42%	98% / 52%
C-Tree	99% / 65%	98% / 66%	98% / 38%	99% / 55%
SVM	99% / **72%**	99% / 60%	99% / 42%	99% / 52%

6.3 Feature Selection Results

The tables presented display the top accuracy results for each of the chosen machine learning algorithms, Decision Trees and Support Vector Machines, predicting effect or no effect on the four toxicological endpoints. The models used to obtain these results are based on the oversampled data sets with feature selection.

6.3.1 Pre-Processing Results

Table 4 displays the top accuracy results chosen based on the best results for each of the impacts observed across all of the pre-processing techniques used. More detailed results are shown in Appendix A.

Table 4: Pre-Processing Results Across Top Machines Learning Methods

Endpoint	**Developmental**	**Proliferative**	**Non-Proliferative**	**Neoplastic**
Best Guess	60% / 72%	50% / 80%	50% / 60%	50% / 60%
Best R-Tree	97% / 67%**	99% / 78%***	97% / 45%*	99% / 56%***
Best C-Tree	99% / 70%**	99% / 78%***	98% / 45%*	99% / 58%***
Best SVM	99% / **74%***	98% / 76%***	98% / 54%*	99% / **60%***

Accuracies for each outcome using each pre-processing technique and each algorithm (Cross Validation/Out of Sample) rounded to the nearest whole number. Accuracies in red are at or above the accuracy obtained by always guessing the most frequent outcome. The percentages labeled with * are from ROC pre-processing, ** are from F-Score pre-processing, and *** are from PCA pre-processing.

6.3.2 Post-Processing Results

Table 5 displays the top accuracy results chosen based on the best results for each of the impacts observed across all of the pre-processing techniques used. More detailed results are shown in Appendix B.

Table 5: Post-Processing Results Across Top Machine Learning Methods

Endpoint	**Developmental**	**Proliferative**	**Non-Proliferative**	**Neoplastic**
Best Guess	60% / 72%	50% / 80%	50% / 60%	50% / 60%
Best R- Tree	97% / 69%**	98% / 77%***	98% / 47%***	99% / 57%***
Best C-Tree	99% / 67%*	97% / 77%***	96% / 46%***	97% / 57%***
Best SVM	99% / 66%*	99% / **80%*****	98% / **60%*****	99% / **60%*****

Accuracies for each outcome using each pre-processing technique and each algorithm (Cross Validation/Out of Sample) rounded to the nearest whole number. Accuracies in red are at or above the accuracy obtained by always guessing the most frequent outcome. The percentages labeled with * are from ROC pre-processing, ** are from F-Score pre-processing, and *** are from PCA pre-processing.

7 Conclusion

Our results show that using the chosen machine learning methods to create predictive models for impacts of chemicals on toxicological endpoints do not significantly improve upon the accuracy of simply guessing the most frequent outcome. Using Support Vector Machines and Decision Trees, high cross validation accuracies were obtained across all of the endpoints of interest. However, the out of sample accuracies remained low. The low out of sample prediction performance compared to the high cross validation performance is likely a result of artificial inflation of cross validation results due to oversampling, as well as the result of overfitting the data. The poor out of sample performance may also be due to the high dimensionality of the feature space, despite the use of feature selection methods.

Another potential source of error is flawed feature information, which may be caused either by errors in the chemical structures themselves or in the PaDEL software. This error could be reduced by processing the chemical structures for potential issues such as the presence of salts, problem causing metals, and charged structures. This could be implemented using a KNIME (Konstanz Information Miner) workflow as shown in Mansouri et al. [22].

The in vivo data used could also be an explanation for the poor results. While in vivo biological effects are clearly the result of bio-chemical interactions, the processes which connect chemical structure to in vivo toxicology are likely complex and difficult to predict using only chemical structure data. A solution may be to include in vitro testing results, which are less removed from in vivo results, as additional inputs into the models.

The in vivo data also used endpoint categories that contained a diverse variety of toxicological effects which were aggregated due to the low sizes of the set of chemicals tested for any specific effect. For example, the endpoint category of Developmental Reproductive contains data on cleft pallets along with data on fertility, among other effects. Thus, we may be predicting too broad of a category. If more observations are obtained, narrowing the endpoint categories to predict more specific outcomes will likely improve the models.

Another solution to connecting the chemical features to the in vivo data may be to create new features which are functions of the original chemical features. Certain chemical features may work together in order to elicit a toxicological effect. This relationship could be obtained by using biochemical knowledge of the interaction between structural properties.

An additional way to improve out of sample prediction accuracies is to combine different machine learning algorithms within one overall predictive model in order to capitalize on the strengths of each algorithm. This is known as consensus modeling, and could be implemented similarly to the consensus model in

Mansouri et al. [22].

A potential future direction for this research is creating continuous models to predict the minimum dosage of a chemical that would cause a toxicological effect on an given endpoint. A classification model could be created to predict a category specifying the range in which the minimum dose falls. In addition, a regression model could be created to predict the exact value of the minimum dose. For the categorized continuous model, accuracy would be evaluated by diving the correct predictions by the total predictions. For the continuous dosage model, accuracy would be calculated by looking at the mean squared error.

References

[1] Classification and regression trees. *Data Mining*, pages 36–350, 2009.

[2] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[3] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[4] Suresh Balakrishnama and Aravind Ganapathiraju. Linear discriminant analysis-a brief tutorial. *Institute for Signal and information Processing*, 18, 1998.

[5] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[6] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[7] Rama Chellappa, Charles L Wilson, and Saad Sirohey. Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):705–741, 1995.

[8] Yi-Wei Chen and Chih-Jen Lin. Combining svms with various feature selection strategies.

[9] Vladimir Cherkassky and Yunqian Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural networks*, 17(1):113–126, 2004.

[10] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[11] Ronaldo C. Prati Gustavo E. A. P. A. Batista and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. 6:20–29, 2004.

[12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer New York, 2009.

[13] Phuong Hoang. Supervised learning in baseball pitch prediction and hepatitis c diagnosis. 2015.

[14] William L. Hosch. Machine learning, 2016.

[15] Yu Hen Hu and Jenq-Neng Hwang. *Handbook for Neural Network Signal Processing*. CRC Press, 2001.

[16] The MathWorks Inc. classregtree. `http://www.mathworks.com/help/stats/classregtree.html`. Accessed: 2016-07-28.

[17] Alan Julian Izenman. Linear discriminant analysis. In *Modern Multivariate Statistical Techniques*, pages 237–280. Springer, 2013.

[18] P Juszczak, D Tax, and Robert PW Duin. Feature scaling in support vector data description. In *Proc. ASCI*, pages 95–102, 2002.

[19] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. 1995.

[20] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[21] Wei-Yin Loh. Fifty years of classification and regression trees. 2016.

[22] Kamel Mansouri, Ahmed Abdelaziz, Aleksandra Rybacka, Alessandra Roncaglioni, Alexander Tropsha, Alexandre Varnek, Alexey Zakharov, Andrew Worth, Ann M Richard, Christopher M Grulke, et al. Cerapp: Collaborative estrogen receptor activity prediction project. *Journal of Environmental Health Perspectives*, 2016.

[23] A Mathur and GM Foody. Multiclass and binary svm classification: Implications for training and classification users. *IEEE Geoscience and remote sensing letters*, 5(2):241–245, 2008.

[24] Charles E Metz. Basic principles of roc analysis. In *Seminars in nuclear medicine*, volume 8, pages 283–298. Elsevier, 1978.

[25] Tatiana I Netzeva, Andrew P Worth, Tom Aldenberg, Romualdo Benigni, Mark TD Cronin, Paola Gramatica, Joanna S Jaworska, Scott Kahn, Gilles Klopman, Carol A Marchant, et al. Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. *ATLA*, 33:155–173, 2005.

[26] Padraic G. Neville. Decision trees for predictive modeling. 1999.

[27] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[28] Roger Lirely Tammy W. Cowart and Sherry Avery. Two methodologies for predicting patent litigation outcomes: Logistic regression versus classification trees. 2014.

[29] Sergios Theodoridis, Aggelos Pikrakis, Konstantinos Koutroumbas, and Dionisis Cavouras. *Introduction to pattern recognition: a matlab approach*. Academic Press, 2010.

[30] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. 2006.

[31] Sun-Chong Wang. *Artificial Neural Network*. Springer US, 2003.

A Results: Pre-Processing

Results for the full, F-Score reduced, ROC curve reduced, and PCA reduced data sets were obtained for all endpoints. Results are listed in the form cross validation accuracy/out of sample accuracy.

Table 6: Developmental Reproductive

Method	**Best Guess**	**R-Tree**	**C-Tree**	**SVM**
Full	60.23% / 72%	97.21% / 63.6%	98.57% / 65.4%	99.08% / 70%
F-Score	60.23% / 72%	96.99% / 67.4%	98.55% / 69.6%	99.15% / 62%
ROC	60.23% / 72%	96.81% / 66%	97.68% / 67%	99.14% / **74%**
PCA	60.23% / 72%	98.23% / 58%	98.4% / 66%	98.70% / 60%

Table 7: Proliferative Pathology

Method	**Best Guess**	**R-Tree**	**C-Tree**	**SVM**
Full	50.04% / 80%	97.48% / 69%	98.23% / 66.4%	99.10% / 48%
F-Score	50.04% / 80%	97.18% / 62.2%	97.98% / 61.8%	98.89% / 66%
ROC	50.04% / 80%	97.24% / 70%	97.95% / 69%	99.12% / 32%
PCA	50.04% / 80%	98.5% / 78%	98.75% / 78%	98.19% / 76%

Table 8: Non-Proliferative Pathology

Method	**Best Guess**	**R-Tree**	**C-Tree**	**SVM**
Full	50.34% / 60%	97.09% / 42.2%	98.35% / 38.4%	98.64% / 46%
F-Score	50.34% / 60%	96.98% / 37.6%	98.34% / 32.8%	98.57% / 44%
ROC	50.34% / 60%	97% / 45%	97.68% / 45%	98.42%/ 54%
PCA	50.34% / 60%	98.18% / 43%	98.02% / 38%	98.44% / 54%

Table 9: Neoplastic Pathology

Method	**Best Guess**	**R-Tree**	**C-Tree**	**SVM**
Full	50.04% / 60%	97.5% / 52.2%	98.9% / 55.4%	99.69% / **60%**
F-Score	50.04% / 60%	96.86% / 51.6%	99.09% / 51%	99.27% / 46%
ROC	50.04% / 60%	97.25% / 51%	97.98% / 51%	99.2% / **60%**
PCA	50.04% / 60%	98.71%/ 56%	99.16% / 58%	98.97% / 40%

B Results: Post-Processing

Results for the full, F-Score reduced, ROC curve reduced, and PCA reduced data sets were obtained using the top 30% of features chosen by the sensitivity analysis. This cutoff was chosen based on the percentile that resulted in the highest accuracy for the full oversampled data sets. Results are listed in the form cross validation accuracy/out of sample accuracy.

Table 10: Developmental Reproductive

Method	**Best Guess**	**R-Tree**	**C-Tree**	**SVM**
Full	60.23% / 72%	97.63% / 65.4%	98.64% / 68.6%	99.14%/ 60%
F-Score	60.23% / 72%	96.98% / 69.4%	98.68% / 66.8%	99.90% / 62%
ROC	60.23% / 72%	97.96% / 62.8%	98.57% / 67.4%	99.14% / 66%
PCA	60.23% / 72%	98.25% / 54.8%	95.53% / 58.4%	99.04% / 56%

Table 11: Proliferative Pathology

Method	**Best Guess**	**R-Tree**	**C-Tree**	**SVM**
Full	50.04% / 80%	98.17% / 68.2%	98.25% / 65.6%	99.03% / 44%
F-Score	50.04% / 80%	96.98% / 67.6%	98.23% / 66.8%	98.57% / 64%
ROC	50.04% / 80%	97.71% / 68.2%	98.05% / 67.8%	98.52% / 68%
PCA	50.04% / 80%	98.48% / 76.8%	97.24% / 77%	98.62% / 80%

Table 12: Non-Proliferative Pathology

Method	**Best Guess**	**R-Tree**	**C-Tree**	**SVM**
Full	50.34% / 60%	97.8% / 40.4%	98.35% / 39.4%	98.51%/ 42%
F-Score	50.34% / 60%	96.99% / 40.2%	97.95% / 37.2%	99.57%/ 40%
ROC	50.34% / 60%	97.65% / 44%	98.07% / 44%	98.40%/ 56%
PCA	50.34% / 60%	97.91% / 47.2%	95.65% / 46.4%	98.25%/ 60%

Table 13: Neoplastic Pathology

Method	**Best Guess**	**R-Tree**	**C-Tree**	**SVM**
Full	50.04% / 60%	98.06% / 52%	98.92% / 54.4%	99.63% / 54%
F-Score	50.04% / 60%	96.8% / 51.6%	99.02% / 51.6%	99.57% / 60%
ROC	50.04% / 60%	97.32% / 51.8%	99.03% / 54.4%	90.57% / 56%
PCA	50.04% / 60%	98.68% / 57.2%	96.88% / 56.6%	99.27% / 60%