World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:12, No:3, 2018

Optimized Preprocessing for Accurate and Efficient Bioassay Prediction with Machine Learning Algorithms

Jeff Clarine, Chang-Shyh Peng, Daisy Sang

***Abstract*—**Bioassay is the measurement of the potency of a chemical substance by its effect on a living animal or plant tissue. Bioassay data and chemical structures from pharmacokinetic and drug metabolism screening are mined from and housed in multiple databases. Bioassay prediction is calculated accordingly to determine further advancement. This paper proposes a four-step preprocessing of datasets for improving the bioassay predictions. The first step is instance selection in which dataset is categorized into training, testing, and validation sets. The second step is discretization that partitions the data in consideration of accuracy vs. precision. The third step is normalization where data are normalized between 0 and 1 for subsequent machine learning processing. The fourth step is feature selection where key chemical properties and attributes are generated. The streamlined results are then analyzed for the prediction of effectiveness by various machine learning algorithms including Pipeline Pilot, R, Weka, and Excel. Experiments and evaluations reveal the effectiveness of various combination of preprocessing steps and machine learning algorithms in more consistent and accurate prediction.

***Keywords*—**Bioassay, machine learning, preprocessing, virtual screen.

I. Introduction

MODERN drug discovery encapsulates a myriad of sciences and technology. The application of in silico modeling has become popular to guide researchers to develop structure activity relationships (SAR) between new chemical entities (NCE) and bioassays [1]. One application of in silico modeling is virtual screening, which predicts a bioassay endpoint from the substance data that describes the NCE. Bioassay data measures the activity of an enzyme or receptor endogenous to the human body. A substance can have none to a measurable amount of potency on the enzyme or receptor. The endpoint is a non-calculable, continuous datum that can only be obtained from the bioassay. The virtual screening models are developed from machine learning algorithms (MLA) and training sets that include substance and bioassay data. Accurate SAR from virtual screening depends on the quality and meaningfulness of the data. Effective data preprocessing of substance and bioassay data is critical to the development of effective virtual screening models. Besides, having an optimized MLA is important to performance and accuracy of the results.

J. Clarine and C. S. Peng are with the Computer Science Department, California Lutheran University, Thousand Oaks, CA 91360 USA (phone: 805-493-3819; e-mail: peng@CalLutheran.edu).

D. Sang is with the Computer Science Department, California State Polytechnic University, Pomona, CA 91768 USA.

Bioassay data and chemical structures are mostly mined from multiple databases. Instance results from the bioassays and substance data for the NCEs are warehoused in multiple SQL databases. Instance selection is focused on how to handle multiple instances of the same NCE within a dataset, bioassay endpoints outside the expected range (also known as outliers) and splitting the dataset into training, testing and validation sets. NCEs can be referenced multiple times through the same assay, either as a control for the bioassay or a recheck of the final value. This can create multiple measured endpoints for the same NCE. The outcome from the bioassay can vary significantly. It is therefore critical to efficiently identify multiple instances of the same compound ID and examine their respective endpoints. Unfortunately, there are not many, if any at all, corresponding studies.

This paper studies and recommends pertinent methodology that can optimize the preprocessing of the input dataset, improve the reliability of the predicted values, and enhance the robustness of the models. Selected MLA are compared and refined for better accuracy in result prediction. Analysis uses development tools Pipeline Pilot (PP) [2], R [3], Weka [4], and Excel [5]. Pipeline Pilot is a high-level language that can retrieve data from multiple databases. Each retrieval (also known as observation) can be processed individually and recompiled into a dataset of multiple optional file formats. Graphical user interface is available to visualize the data processing. Pipeline Pilot contains many default and licensed components for handling chemical structures and processing scientific data [6]. The R environment is a statistical computing and graphics environment designed to quickly process datasets in vector and matrix formats. It is an ideal choice to preprocess and summarize datasets prior to import into Weka, which supports flexible implementation of selected learning algorithms. Section II discusses methods of preprocessing. Section III presents experiments and results. The paper concludes with the discussion.

II. Methods

The datasets used in this project are results from in vitro bioassay screenings for drug-drug interactions (DDI) and pharmacokinetic (PK). The DDI screenings screen for inhibitory potential of CYP2D6 and CYP3A4 human liver enzymes. The PK screenings determine the intrinsic clearance of the NCEs in human liver microsomes (HLM) and rat liver microsomes (RLM). Datasets are processed through instance

selection, discretization, normalization, feature selection, and machine learning.

NCEs can be assayed multiple times to detect duplicates or to determine the maximum and minimum cutoffs according to varied criteria. The complete datasets are separated into training sets and validation sets; 80% of the dataset is used for training and testing the MLA, and 20% of the dataset is reserved for validation [1]. Two types of sampling are compared in order to learn how sampling affects the robustness of the classification by MLA. The first is undersampling, where observations are decreased based on their bioassay endpoint values. The other is oversampling in which observations are increased also based on their bioassay endpoint values.

In the undersampled datasets, the numbers of active or inactive observations are reduced to one half or one third. In the oversampled datasets, the number of active or inactive observations can be doubled, tripled, or quadrupled. If the undersampling or oversampling is implemented on the active observations, the number of inactive observations remains fixed. On the other hand, if the undersampling or oversampling is implemented on the inactive observations then the number of active observations remains fixed. Fig. 1 shows the variations of the active and inactive observations for the CYP2D6 dataset. The Active:Inactive column represents the change in the number of active or inactive observations compared to the original datasets, which is denoted by 1:1.

Number of Active Samples	Number of Inactive Samples	Total Number of Samples	Active:Inactive
5689	20285	25974	0.33:1
8533	20285	28818	0.5:1
17065	6762	23827	1:0.33
17065	10143	27208	1:0.5
17065	20285	37350	1:1
17065	40570	57635	1:2
17065	60855	77920	1:3
34130	20285	54415	2:1
51195	20285	71480	3:1
68260	20285	88545	4:1

Fig. 1 Number of Active vs. Inactive Samples

Discretization [7] classifies a continuous dataset into a discrete list of values. The process consists of binning the datasets into a predetermined number of outcomes. This reduces the number of possible outcomes and enables the developer to control the balance between accuracy and precision. Reducing the numbers of possible outcomes to a discrete list reduces the overall precision of the predictions; but increases the accuracy. Balancing accuracy over precision is dependent on the goals of the end users [8]. The discretization process first determines the maximum and minimum values for the each of the datasets, and a cutoff value to identify if an NCE is active or inactive in the bioassay. Two (binary) bins are used to classify if an outcome is considered active or inactive. Three bins are used to identify if the outcome would be a maximum value, minimum value, or somewhere in between. Five bins and 10 bins are used to approximate the continuous bioassay data endpoints. In Fig. 2, bin values are set based on the data range, maximum and minimum values and active/inactive cutoff. Using R, the outcomes from the bioassay endpoints were discretized into 2 bins (binary), 3 bins, 5 bins or 10 bins. Fig. 3 shows how the original HLM bioassay outcomes can be transformed into 2 bins, 3 bins, 5 bins and 10 bins.

In normalization [7], data attributes and outcomes are normalized between the minimum value 0 and maximum value 1 according to the following equation, in which $x_{ij}^{normalization}$ is the normalized value, x_{ij} is the value of interest, x_j^{min} is the minimum value, and x_j^{max} is the maximum value.

$$x_{ij}^{normalization} = \left(x_{ij} - x_j^{min}\right)/\left(x_j^{max} - x_j^{min}\right)$$

Feature selection has three essential elements; feature reduction, feature construction, and feature set selection. Feature reduction, or attribute reduction, is implemented with the raw datasets, which includes the compound IDs, bioassay endpoints, chemical structures, and various attributes. Feature construction generates the basic chemical structure properties and chemical fingerprints that breakdown a chemical structure into fragments. Calculations are carried out with the Chemical Property Calculator script in Pipeline Pilot.

10 Bins	5 Bins	3 Bins	Binary
14	14	14	Inactive
38	78	207	Active
86			
134	207		
182			
231			
279	335		
327			
375			
399	399	399	

Fig. 2 Bin Values for Discretization of HLM Dataset

Compound ID	Outcome Value	10 Bin Value	5 Bin Value	3 Bin Value	Binary Value
1	139	134	78	207	active
2	24.5	38	78	207	active
3	14.5	38	78	207	active
4	399	399	399	399	active
5	14	14	14	14	inactive
6	399	399	399	399	active
7	14	14	14	14	inactive
8	14	14	14	14	inactive
9	14	14	14	14	inactive
10	14	14	14	14	inactive
11	162	182	207	207	active
12	210	231	207	207	active
13	399	399	399	399	active
14	18	38	78	207	active
15	71	86	78	207	active
16	62	38	78	207	active
17	399	399	399	399	active

Fig. 3 Sample Discretization Results for HLM Dataset

Chemical fragment attributes are created in Pipeline Pilot. If

the fragment exists in a chemical structure, the value is set to 1; otherwise, it is set to 0. There are various processes to generate chemical fragments (also known as fingerprints), e.g. Daylight [9], BCI [10], UNITY 2D [11], and MDL key fingerprinting [12]. MDL is selected for its restricted number of fragments and its integration within PP. MDL fingerprinting uses a general molecule perception algorithm to identify and count atoms and bonds and to recognize chemical's structural properties. Keybits are created for chemical structures. Every chemical structure has one or more keybits. The total number of keybits depends on the number and types of atoms and bonds. Keybits are then compared to the MDL keyset. The MDL keyset contains 960 unique chemical fragments. If the keybit from the input structure matches a keybit in the chemical fragments, the structure is said to contain that fragment. Accordingly, chemical fingerprint is generated which describes the chemical structure as a set of chemical fragments.

The normalized datasets are then processed by Decision Tree (DT), Logistic Regression (LogR), and Neural Networks (NN) machines learning algorithms. Decision Tree machine learning uses Weka's REPTree algorithm and REPTree is a Classification and Regression Tree algorithm that predicts both classification and continuous outcomes [4]. REPTree takes a top down approach to identify the variable and splitting criteria and form a tree of nodes and leaves. Nodes are the splitting criteria and leaves are the groups that result from splitting. The splitting criterion is based on outcomes' homogeneity which is measured by impurity function. Impurity functions quantify how many outcomes are the same within each leaf. REPTree uses entropy as the impurity measure [13]. Entropy impurity functions calculate the probability of the occurrences of an event; the lower the probability the higher the entropy value. Calculation first tests and identifies variable and splitting criteria that can grow a leaf with the least amount of impurity. Outcomes are then divided into two groups based upon the findings. These two steps iterates until the number of outcomes in the leaves are too small or has met the purity criteria.

Logistic Regression machine learning adopts Weka's Logistics Algorithm. LogR predicts a binary outcome using one or more variables. The outcomes are 1 (i.e. true, active, not observed) and 0 (i.e. false, not active, observed). LogR calculates a log curve for the data. A threshold value of 0.5 is assigned to the dependent outcome. If the result from the log curve is greater than 0.5 then the outcome is 1, otherwise the outcome is 0 [14].

Neural Network machine learning uses Weka's Multilayer Perceptron algorithm. NN is a black box, non-linear environment. For each attribute of the dataset, an input unit is initialized with the attribute's value. Depending on the machine learning implementation, the Multilayer Perceptron can result in multiple hidden layers. The last layer is the output layer, which provides the prediction data. The number of possible outcomes determines the number of output layers. When predicting a continuous output, the output units become unthresholded linear units [15]. All units are connected via weights each of which has a numeric value. The weights retain the knowledge learned in the training phase. As the model is trained, the weights are the only variable that are updated and retrained from one training cycle (also known as epoch) to another. Initial weights are assigned at random by the algorithm. If the Neural Network accurately predicts the outcomes for the observation, no changes are made to the weights. If the model predicts a wrong outcome, weights are updated. The degree of change in the weights is set by the learning rate. In the Multilayer Perceptron algorithm, the number of epochs can be set by the user. Higher epoch value translates to more training which typically increase the robustness and accuracy of the predictions. However, if the epoch value is too high, computation power can be wasted without corresponding gains in accuracy. Therefore, a proper epoch value is critical to the balance of efficiency and accuracy. Optimization of the Multilayer Perceptron is focused on varying the learning rate, number of hidden layers, and number of epochs. As shown in Fig. 4, the number of hidden layers are set to *a*, *i* or *o*.

DT, LogR, and NN are further trained through validation in which the training data is divided into three equal portions; two for training and one for testing. Error measurements for continuous outcomes are evaluated with Root Mean Squared Error and Correlation Coefficients. For binary outcomes, Confusion Matrix is generated for the outcomes, sensitivity, and specificity.

Number of Hidden Layer Symbol	Number of Hidden Layers
a	(Number of Attributes + Number of Outcomes)/2
i	Number of Attributes
o	Number of Classes

Fig. 4 Number of Hidden Layers in Neural Network Algorithm

Root Mean Squared Error (RMSE), $\sqrt{\sum_{i=1}^{n}(p_i - a_i)^2/n}$, calculates the error difference between the actual outcome *a* and predicted outcome *p*.RMSE gives an error estimation that predicts the dimensions of the outcomes. RMSE-based comparisons are used when comparing models on the same scale. For datasets of different scales, Percent RMSE (%RMSE), $\sqrt{\sum_{i=1}^{n}(p_i - a_i)^2/n}/(a_{max} - a_{min})$, is applied by dividing RMSE into the range of the scale.

Correlation Coefficient (CC) [16] equals $S_{PA}/\sqrt{S_P S_A}$, where;

$$S_{PA} = \sum_i (p_i - \bar{p})(a_i - \bar{a})/(n-1)$$

$$S_P = \sum_i (p_i - \bar{p})^2/(n-1)$$

$$S_A = \sum_i (a_i - \bar{a})^2/(n-1)$$

CC calculates the statistical correlation between the actual values *a* and predicted values *p*. The value of 1 indicates the model perfectly predicts the outcomes 100% of the time. The

value of 0 indicates there is no correlation between the predictions and the actual outcomes. The value of -1 indicates the model predicts the opposite outcome 100% of the time. CC can be used to assess the performance of the MLAs across the datasets since CC's value is independent of the scale of the outcome.

Confusion Matrix (CM) is a table that tabulates the number of true positives, true negatives, false positives, and false negatives. Sensitivity is the ratio of true positives vs. the sum of true positives and false negatives, and specificity is the ratio of true negatives vs. the sum of false positives and true negatives. The sensitivity value of 1 means the model correctly predicts the true positive values 100% of the time. The specificity value of 1 means the model correctly predicts the true negatives values 100% of the time. The sensitivity or specificity value of 0 indicates the model never predicts the correct outcome for either positive or negative values. The sensitivity and specificity values can be used in combination to evaluate the performance of a prediction model. Active compounds are considered a negative finding. Structures that are predicted to be active in CYP2D6, CYP3A4, HLM or RLM may not be synthesized. If the virtual screens predict the compounds as active but in practice otherwise, then potentially good drug candidates could be eliminated before they are ever synthesized and tested. Therefore, specificity is sought to be maximized.

Lastly, the Correctly Classified Observations (CCO) [16] provides estimation for the accuracy of the prediction model. Following is the formula:

$$\frac{(\sum TruePositive+\sum FalsePositive)}{(\sum TruePositive+\sum TrueNegative+\sum FalsePositive+\sum FalseNegavie)}$$

III. Experiments and Results

Dataset endpoints are categorized as inhibition data and intrinsic clearance data. Inhibition data, IC50, is the concentration of an inhibitor to inhibit 50% of the activity of specific enzymes. Intrinsic clearance data, CLint, is a substrate in pool of enzymes. IC50 is measured in micromoles (uM). A known substrate at a single concentration for the enzymes CYP2D6 and CYP3A4 is tested with a test compound at multiple concentrations to evaluate the inhibition of the test compound on the substrate. If the test compound, the inhibitor, inhibits enough turnover of the substrate by the enzyme, an IC50 can be calculated specific to the test inhibitor and can be used to evaluate the overall inhibitory potential of the test compound on the enzyme. The enzymes CYP2D6 and CYP3A4 are endogenous to the human body and are responsible for the metabolism of the majority of drugs. Therefore, new drugs should not be inhibitors of these enzymes.

CLint is measured in micrograms of test compound per milligram of protein per milliliter of the test mixture volume (ug/mg/mL). The test compound, substrate, is tested in human liver microsomes (HLM) and/or rat liver microsomes (RLM). The substrate is metabolized by the enzymes in the liver microsomes and time points are taken over 45 minutes. The amount of substrate remaining is measured at each of the time points, and a CLint value is calculated for the test compound in either HLM or RLM. CLint value smaller than 14 ug/mg/mL is often desired as it indicates lower overall metabolism in liver microsomes. A potential new drug with lower CLint will be available at therapeutic concentration in the human body for a prolonged period of time.

Datasets first undergo the instance selection. Fig. 5 shows the comparison before and after the removal of duplicates.

Number of Active Samples	Number of Inactive Samples	Total Number of Samples	Active:Inactive
5689	20285	25974	0.33:1
8533	20285	28818	0.5:1
17065	6762	23827	1:0.33
17065	10143	27208	1:0.5
17065	20285	37350	1:1
17065	40570	57635	1:2
17065	60855	77920	1:3
34130	20285	54415	2:1
51195	20285	71480	3:1
68260	20285	88545	4:1

Fig. 5 Instance Selection

%RMSE and CC are similar across datasets and MLAs with the exception of NN predictions on CYP3A4. The mean %RMSE in DT's predicted outcomes is 36 ±3% and mean CC in DT's predicted outcomes is 0.48 ±0.06. The mean %RMSE in NN's predicted outcomes is 90 ±106% and mean CC in NN's predicted outcomes is 0.30 ±0.24. Excluding CYP3A4 from the NN predictions gives a mean %RMSE of 37 ±3% and mean CC of 0.40 ±0.14. With the exception of CYP3A4 in the NN predictions, the %RMSE indicates both DT and NN can predict 36 to 37% of the actual value. CC between 0.40 and 0.48 indicates a weak correlation between actual outcomes and predicted outcomes. The high %RMSE and low CC for NN's CYP3A4 predictions (248% and -0.01, respectively) implies no correlation between the predicted outcomes and actual outcomes. Figs. 6 and 7 are the charts for %RMSE and CC, respectively.

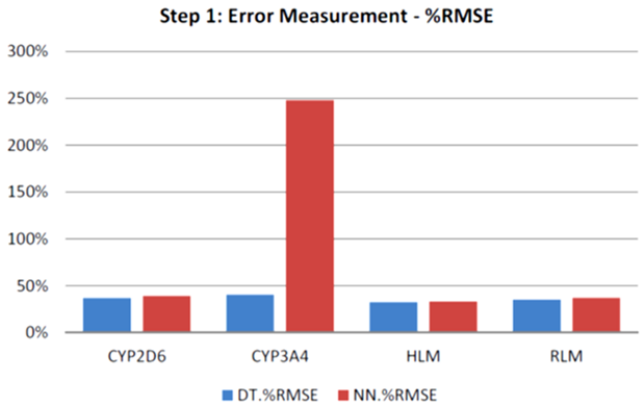

Fig. 6 Instance Selection %RMSE

Results of the second step, discretization, are shown in Fig. 8 to Fig. 12. The non-binary bins and continuous datasets have high %RMSE and low CC, which indicate a weak correlation

between actual and predicted outcomes. However, the binary datasets have moderate to high CCO values. The binary datasets thus can provide the most accurate predictions. As the result, the binary (2 bins) datasets are selected into the following normalization step.

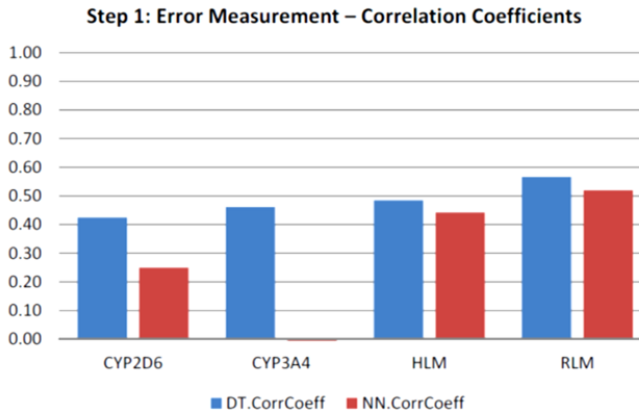

Fig. 7 Instance Selection CC

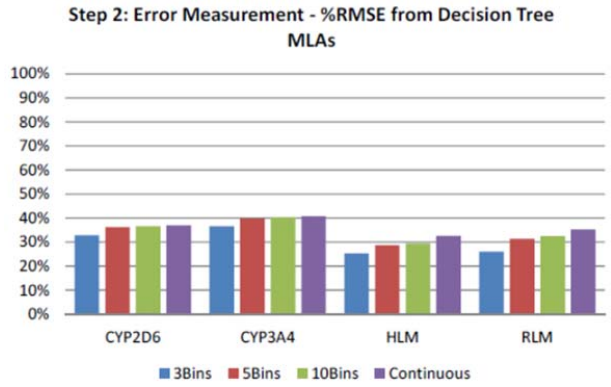

Fig. 8 Discretization %RSMC DT

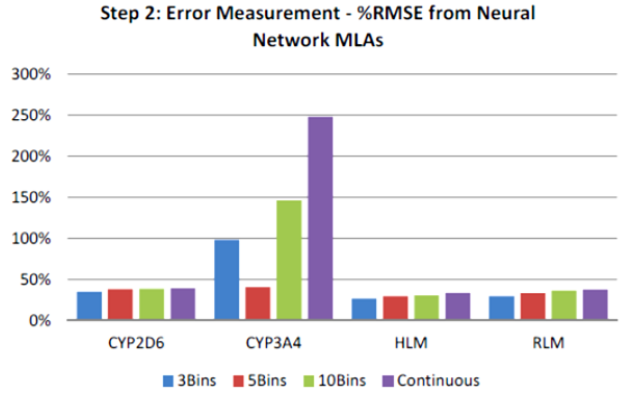

Fig. 9 Discretization %RSMC NN

Fig. 13 shows the result of normalization. Fig. 14 depicts the relation between sensitivity and specificity among the MLAs with and without normalization. Since the objective is to minimize the number of false positives, the non-normalized datasets best meet the criteria. Therefore, normalization is not implemented in the following steps.

Sensitivity and specificity are further evaluated with oversampling and undersampling. The result, in Fig. 15, shows that the 1:1 datasets had the highest specificity. Therefore, datasets with 1:1 sampling are selected for next step feature selection.

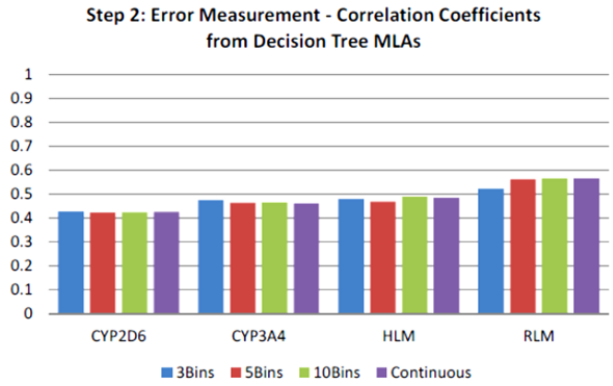

Fig. 10 Discretization CC DT

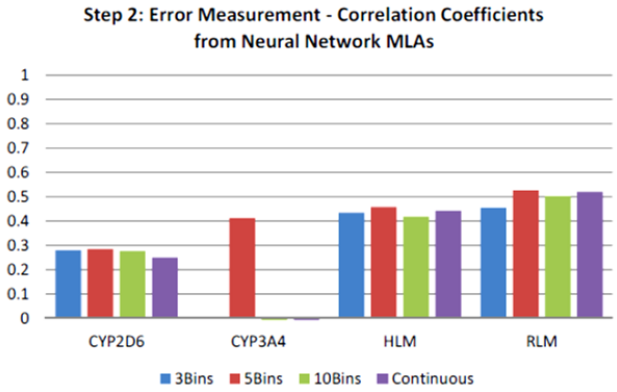

Fig. 11 Discretization CC NN

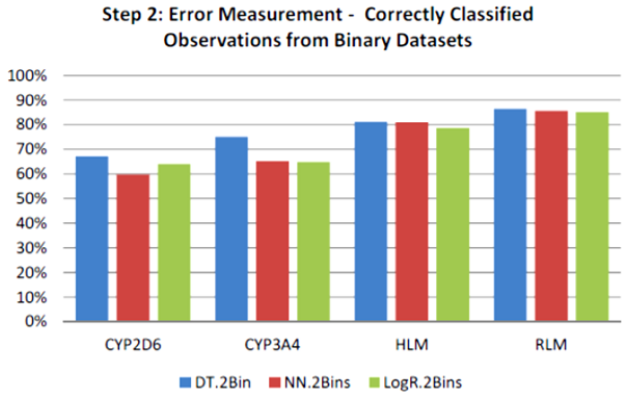

Fig. 12 Discretization CC Binary Bin

Feature selection results are shown in Fig. 16, in which ChemProp denotes basic chemical property, FingerOnly denotes MDL fingerprints, and All denotes the combination of both. For CYP2D6, the DT algorithm with All attributes is selected for its high specificity and acceptable sensitivity. For CYP3A4, the DT algorithm with ChemProp attributes is

selected for its high specificity. For HLM, the NN algorithm with All attributes is selected because of high specificity and good sensitivity. For RLM, the NN algorithm with All and FingerOnly attributes provides the same highest specificity and sensitivity results. Since the dataset with All attributes contains more descriptive information, corresponding NN is selected for the final machine learning optimization.

For the CYP2D6 and CYP3A4 datasets that were predicted with DT, the only optimization step required is pruning. No gains were observed. Thus using DT without pruning provides a more accurate prediction. For both HLM and RLM datasets, a learning rate of 0.03, *i* hidden layers, and 100 epochs provides the most accurate results with the highest specificity. Figs. 17 and 18 depict the results, respectively.

Dataset	Algorithm	Condition	CCO	Sensitivity	Specificity
CYP2D6	DT	Non-Normalized	0.67	0.61	0.72
		Normalized	0.54	0.04	0.95
	NN	Non-Normalized	0.60	0.87	0.37
		Normalized	0.56	0.06	0.97
	LogR	Non-Normalized	0.64	0.53	0.73
		Normalized	0.54	0.02	0.98
CYP3A4	DT	Non-Normalized	0.75	0.74	0.76
		Normalized	0.69	0.71	0.67
	NN	Non-Normalized	0.65	0.69	0.61
		Normalized	0.68	0.62	0.75
	LogR	Non-Normalized	0.65	0.7	0.59
		Normalized	0.65	0.74	0.54
HLM	DT	Non-Normalized	0.81	0.92	0.42
		Normalized	0.78	0.95	0.17
	NN	Non-Normalized	0.81	0.95	0.3
		Normalized	0.78	1	0.02
	LogR	Non-Normalized	0.79	0.96	0.19
		Normalized	0.78	1	0.03
RLM	DT	Non-Normalized	0.86	0.96	0.31
		Normalized	0.79	0.88	0.28
	NN	Non-Normalized	0.86	0.94	0.38
		Normalized	0.85	0.99	0.09
	LogR	Non-Normalized	0.85	0.98	0.1
		Normalized	0.85	0.98	0.09

Fig. 13 Normalization Results

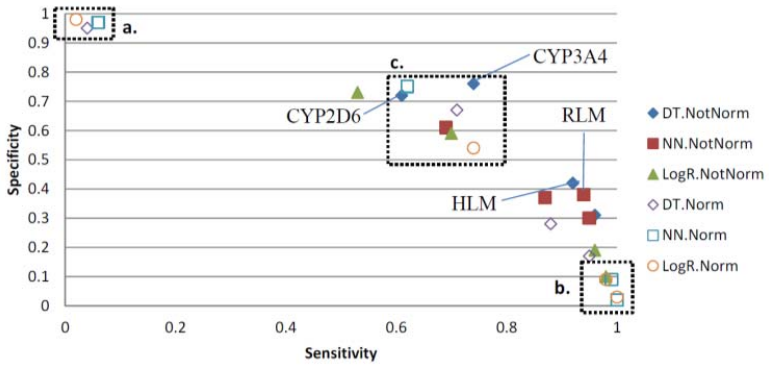

Fig. 14 Normalization Sensitivity vs. Specificity

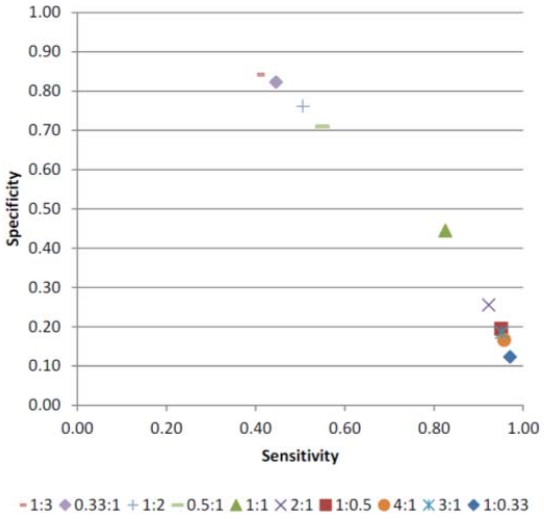

Fig. 15 Over-/Under-sampling Sensitivity vs. Specificity

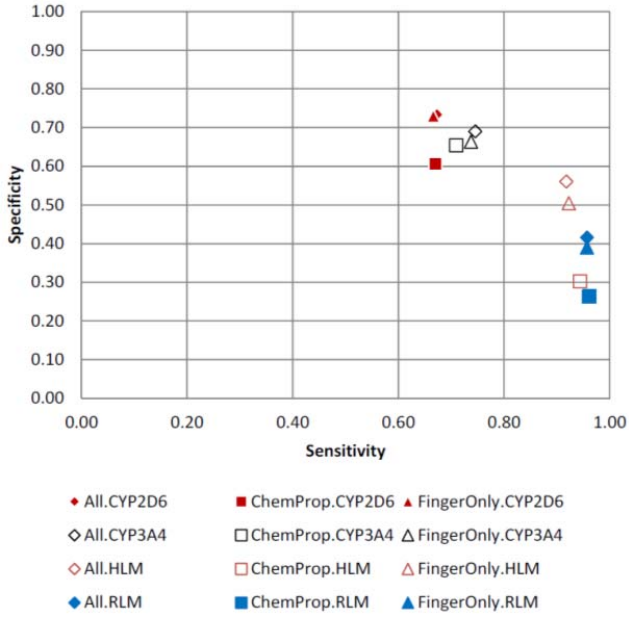

Fig. 16 Feature Selection Results

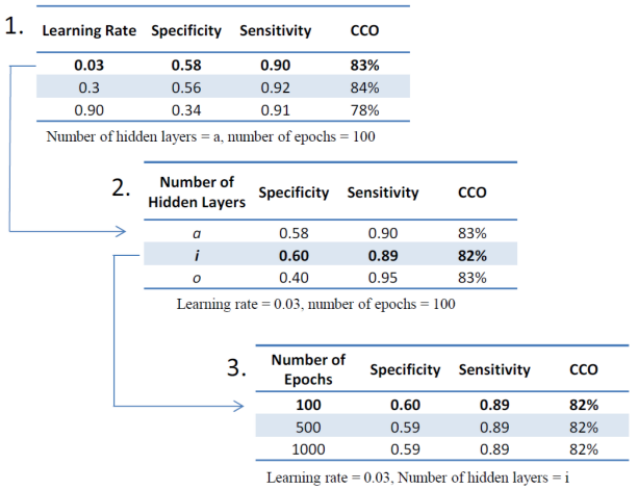

Fig. 17 HLM NN Optimization

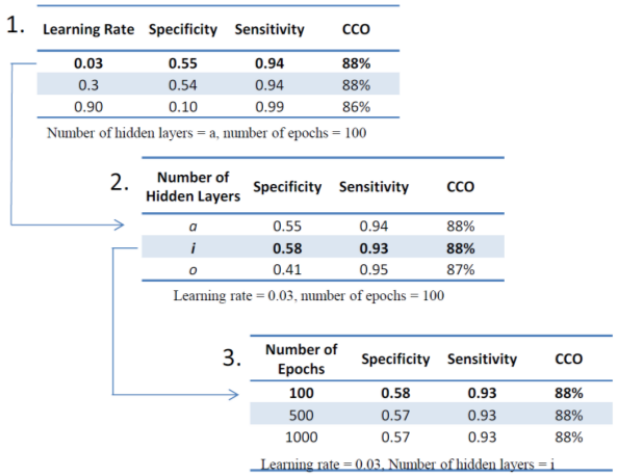

Fig. 18 RLM NN Optimization

IV. Conclusion

Prediction with raw data can be very unreliable. Discretization is an important tool for more useful results. Feature selection is demonstrated to help optimize machine learning. Once the MLA is optimized, users can add features from fingerprinting to finalize the model. Effective preprocessing not only makes better use the valuable computation resources but also help ensure more accurate and consistent prediction.

References

[1] A. C. Schierz, "Virtual screening of bioassay data," Journal of Cheminformatics 1(21), 2009.
[2] Pipeline Pilot, BIOVIA' Graphical Scientific Workflow Authoring Application, http://accelrys.com/, last access 2017.
[3] R, Big Data Statistical Computing and Graphics Software Environment, http://www.rdatamining.com/, last access 2017.
[4] Weka, Waikato Environment for Knowledge Analysis, http://www.cs.waikato.ac.nz/~ml/, last access 2017.
[5] Excel, Microsoft Excel Office Tool, https://products.office.com/en-us/excel, last access 2017.
[6] M. Hassan, R. D. Brown, S. Varma-O'brien, and D. Rogers, "Cheminformatics analysis and learning in a data pipelining environment," Mol Divers 10(3), pp. 283-99, 2006.
[7] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data Preprocessing for Supervised Learning," International Journal of Computer Science 1(2), pp. 111-117, 2006.
[8] J. L. Lustgarten, V. Gopalakrishnan, H. Grover, and S. Visweswaran, "Improving Classification Performance with Discretization on Biomedical Datasets," Proc. AMIA AnnuSymp, pp. 445–449, 2008.
[9] Daylight, Chemical Information Processing Software, http://www.daylight.com/, last access 2017.
[10] BCI, Cheminformatics Software, http://www.digitalchemistry.co.uk/, last access 2017.
[11] UNITY 2D, Biosimulation Software, http://tripos.com/, last access 2015.
[12] MDL, http://accelrys.com/, last access 2017.
[13] P. Ozer, "Data Mining Algorithms for Classification," BSc Thesis Artificial Intelligence, 2008.
[14] Bewick, L. Cheek, and J. Ball, "Statistic review 14: Logistic Regression," Crit Care 9(1), pp. 112-118, 2005.
[15] Multilayer Perceptron, Artificial Neural Network Modeling http://www.cs.waikato.ac.nz/ml/weka/documentation.html, last access 2017.
[16] Witten, E. Frank, and M. A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2011.