Journal of Computer-Aided Molecular Design (2018) 32:583–590
https://doi.org/10.1007/s10822-018-0116-z

Comparing the performance of meta-classifiers—a case study on selected imbalanced data sets relevant for prediction of liver toxicity

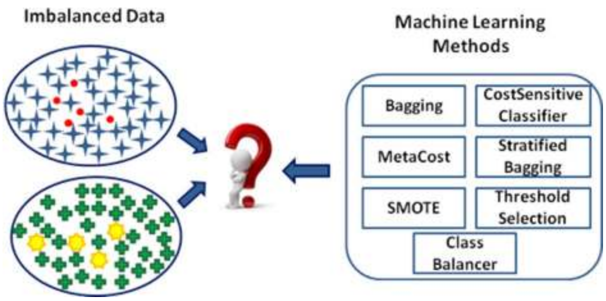
Sankalp Jain[1] · Eleni Kotsampasakou[1,2] · Gerhard F. Ecker[1]

Received: 23 December 2017 / Accepted: 29 March 2018 / Published online: 6 April 2018

Abstract

Cheminformatics datasets used in classification problems, especially those related to biological or physicochemical properties, are often imbalanced. This presents a major challenge in development of in silico prediction models, as the traditional machine learning algorithms are known to work best on balanced datasets. The class imbalance introduces a bias in the performance of these algorithms due to their preference towards the majority class. Here, we present a comparison of the performance of seven different meta-classifiers for their ability to handle imbalanced datasets, whereby Random Forest is used as base-classifier. Four different datasets that are directly (cholestasis) or indirectly (via inhibition of organic anion transporting polypeptide 1B1 and 1B3) related to liver toxicity were chosen for this purpose. The imbalance ratio in these datasets ranges between 4:1 and 20:1 for negative and positive classes, respectively. Three different sets of molecular descriptors for model development were used, and their performance was assessed in 10-fold cross-validation and on an independent validation set. Stratified bagging, MetaCost and CostSensitiveClassifier were found to be the best performing among all the methods. While MetaCost and CostSensitiveClassifier provided better sensitivity values, Stratified Bagging resulted in high balanced accuracies.

Graphical Abstract



Keywords Imbalanced datasets · Machine learning · Classification model · Meta-classifiers · Stratified bagging · Cost sensitive classifier

Sankalp Jain and Eleni Kotsampasakou have contributed equally to this manuscript.

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10822-018-0116-z) contains supplementary material, which is available to authorized users.

Extended author information available on the last page of the article

Abbreviations

AUC	Area under the ROC curve
HTS	High throughput screening
MCC	Matthews correlation coefficient
OATP1B1	Organic anion transporting polypeptide 1B1
OATP1B3	Anion transporting polypeptide 1B3
RF	Random Forest
sd	Standard deviation

SMOTE	Synthetic minority over-sampling technique
SVM	Support vector machines

Introduction

A wide range of classification and regression methods have been applied in QSAR studies. However, many classification methods assume that datasets are balanced in terms of the number of instances of each class and thus give equal importance to all classes, often resulting in classification models of poor accuracy [1, 2]. A major problem that arises in this context is class imbalance, i.e. the number of instances of one class substantially differ from those of the other classes. Especially in the field of drug discovery, imbalanced datasets [2–4] need to be frequently dealt with [2]. Characteristically, a classifier developed on an imbalanced data set shows a low error rate for the majority class and a high error rate for the minority class [5, 6]. Nevertheless, a few studies pointed out that the class imbalance is not a main obstacle in learning [7, 8], and several methods have been developed to address this issue. These methods can be broadly divided into (1) data-oriented/re-sampling techniques; (2) algorithm-oriented methods; and (3) combinatorial/ensemble/hybrid techniques [2, 3, 7, 9, 10].

Several studies compared classifiers that handle imbalanced datasets. Schierz et al. [11] compared four WEKA classifiers (Naïve Bayes, SVM, Random Forest and J48 tree) and reported SVM and J48 to be the best performing for bioassay datasets. Lin and Chen in 2013 found SVM threshold adjustment as the best performing classifier (among linear discriminant analysis, Random Forest, SVM and SVM-threshold adjustment) to deal with imbalanced HTS datasets [9]. Later, Zakarov et al. used under-sampling and threshold selection techniques on several imbalanced PubChem HTS assays to test and develop robust QSAR models in the program GUSAR [12]. In a recent study, Razzaghi et al. reported multilevel SVM-based algorithms to outperform conventional SVM, weighted SVM, neural networks, linear regression, Naïve Bayes and C4.5 tree using public benchmark datasets having imbalanced classes and missing values and real data in health applications [13].

A comprehensive comparison of the performance of different meta-classifiers on datasets with different levels of class imbalance, which would provide guidance for choosing the appropriate method for an imbalanced dataset, has not been attempted so far. Herein, we evaluated the performance of seven distinct meta-classifiers from the three aforementioned categories on four datasets from the toxicology domain. The imbalance ratio of the datasets ranges from 1:4 to 1:20 for the positive and the negative class, respectively. The meta-classifiers were applied to build classification models based on three different sets of descriptors. Considering its wide applicability in modeling imbalanced datasets, Random Forest was used as the common base-classifier for all models [14–18]. Further, we discuss the reasons behind the superior performance of certain meta-classifiers in comparison to the others while explaining their intrinsic limitations.

Methods

Training datasets

Four different datasets from the biomedical sciences domain were used in this study. Two of these are the OATP1B1 and OATP1B3 inhibition datasets consisting of 1708 and 1725 compounds, respectively. Both were compiled and used in our previous study that reported classification models for OATP1B1 and 1B3 inhibition [19]. The other two datasets come from the toxicology domain and are related to drug-induced cholestasis for human data and animal data which comprise 1766 and 1578 compounds, respectively. Both datasets were published in a previous study that reported computational models for hepatotoxicity and other liver toxicity endpoints [20].

External test datasets

The external test sets for OATP1B1 and 1B3 inhibition from our previous study served as test datasets in this study [19]. The test set for human cholestasis was compiled in two stages from two previous studies [21]. The positives for human cholestasis were compiled from literature [22–25] and from the SIDER v2 database [26, 27]. As cholestasis is one of the three types of drug induced liver injury (DILI), and the compounds that are negative for DILI will also be negative for cholestasis, the negatives for drug-induced liver injury compiled in a previous study [21] were used as negatives for cholestasis. Overall, the external human cholestasis dataset consisted of 231 compounds. No data were available for animal cholestasis to be used as an external test dataset. The composition and degree of class imbalance of each training and test dataset is presented in Table 1.

The chemotypes in the datasets were curated using the following protocol:

- Removed all inorganic compounds according to chemical formula in MOE 2014.09 [28].
- Removed salts and compounds containing metals and/or rare or special atoms.
- Standardized chemical structures using Francis Atkinson Standardiser tool [29].
- Removed duplicates and permanently charged compounds using MOE 2014.09 [28].

Table 1 An overview of the training and test datasets used in this study

Dataset name	Total number of compounds	Number of positives	Number of negatives	Imbalance ratio (negatives: positives)	Source
OATP1B1 inhibition training	1708	190	1518	8:1	Kotsampasakou et al. [19]
OATP1B1 inhibition testing	201	64	137	2:1	Kotsampasakou et al. [19]
OATP1B3 inhibition training	1725	124	1601	13:1	Kotsampasakou et al. [19]
OATP1B3 inhibition testing	209	40	169	4:1	Kotsampasakou et al. [19]
Cholestasis human training	1766	347	1419	4:1	Mulliner et al. [20]
Cholestasis human testing	231	53	178	3:1	Kotsampasakou et al. [21]
Cholestasis animal training	1578	75	1503	20:1	Mulliner et al. [20]

– 3D structures were then generated using CORINA (version 3.4) [30], and energy minimized with MOE 2014.09 [28], using default settings (Forcefield MMF94x, gradient 0.05 RMS kcal/mol/A^2, preserving chirality).

Molecular descriptors

Three different sets of descriptors were calculated for each of the datasets:

1. All 2D MOE [28] descriptors (192 descriptors in total).
2. ECFP6 fingerprints (1024 bits) calculated with RDKit [31].
3. MACCS fingerprints (166 bits), calculated with PaDEL software [32].

Machine learning methods

Random Forest [33] implemented in the WEKA software suite [34, 35] was used as a base-classifier along with all the meta-learning methods evaluated in this study. The number of trees was arbitrarily set to 100 (default), since it has been shown that the optimal number of trees is usually 64–128, while further increasing the number of trees does not necessarily improve the model's performance [36]. The following meta-classifiers were investigated: (1) Bagging, (2) Under-sampled stratified bagging, (3) Cost-sensitive classifier, (4) MetaCost, (5) Threshold Selection, (6) SMOTE and (7) ClassBalancer.

1. *Bagging* (*Bootstrap AGGregatING*) [37] is a machine learning technique that is based on an ensemble of models developed using multiple training datasets sampled from the original training set. It calculates several models and averages them to produce a final ensemble model [37]. A traditional bagging method generates multiple copies of the training set by selecting the molecules with replacement from training set in a random fashion. Because of random sampling, about 37% of the molecules are not selected and left out in each run. These samples create the "out-of-the-bag" sets, which are used for testing the performance of the final model. A total of 64 models were used for our analysis, since it was shown in an earlier study by Tetko et al. [38] that larger numbers of models per ensemble (e.g. 128, 256, 512 and 1024) did not significantly increase the balanced accuracy of models.
2. *Under-sampled stratified bagging* [2, 8, 38] In this method, the total bagging training set size is double the number of the minority class molecules. Although a small set of samples was selected each time, the majority of molecules contributed to the overall bagging procedure, since the datasets were generated randomly. The performance of the developed models is tested with molecules from the "out-of-the-bag" set [38]. Since only one way of stratified learning, i.e., under-sampling stratified bagging, was used in the study, we refer to it as "Stratified Bagging".

   Bagging and Stratified Bagging were used as implemented in the Online Chemical Modeling Environment (OCHEM) [39, 40]. For other meta-classifiers, WEKA(v. 3-7-12) [34, 35] was used.
3. *Cost sensitive classifier* [2–4, 10, 11] is a meta-classifier that renders the base classifier cost-sensitive. Two methods can be used to introduce cost-sensitivity: (i) reweighting training instances according to the total cost assigned to each class, i.e. the weights are applied during learning, or; (ii) predicting the class with minimum expected misclassification cost (rather than the most likely class), i.e. the "cost-sensitive" is introduced in the test phase. In our case, the cost sensitivity was introduced according to method (i) using the CostSensitiveClassifier from the set of meta-classifiers of the WEKA software [34, 35].
4. *MetaCost* [41] is another application that provides the methodology to perform cost-sensitive training of a classifier in a generalized meta-learning manner independent of the underlying classifier. It is a combination of Cost-

sensitive meta-classifier and Bagging [37]. The algorithm uses class-relabeling, i.e. it modifies the original training set by changing the class labels to the so-called "optimal classes". The classifier is then trained on this modified training set, which results in having the error rate minimized according to the cost matrix provided to the MetaCost algorithm. This implementation uses all bagging iterations when reclassifying training data. MetaCost is advantageous as, unlike CostSensitiveClassifier, a single cost-sensitive classifier of the base learner is generated, thus giving the benefits of fast classification and interpretable output (if the base learner itself is interpretable). MetaCost further differs from traditional bagging by the fact that the number of examples in each resample may be smaller than the training set size. This variation improves the efficiency of the algorithm. More details about the method can be found in [41].

For both CostSensitiveClassifier and MetaCost, several trials of different cost matrices were applied, until a satisfactory outcome was retrieved.

5. *ThresholdSelector* [42] is a meta-classifier implemented in WEKA [34, 35] that sets a threshold on the probability output of a base-classifier. Threshold adjustment for the classifier's decision is one of the methods used for dealing with imbalanced datasets [2, 43]. By default, the WEKA probability threshold to assign a class is 0.5, i.e. if an instance is attributed with a probability of equal or less than 0.5, it is classified as negative for the respective class, while if it is greater than 0.5, the instance is classified as positive. For our study, the optimal threshold was selected automatically by the meta-classifier by applying internal fivefold cross validation to optimize the threshold according to FMeasure (Eq. 7), a measure of a model's accuracy which considers both precision and sensitivity [44].
6. *SMOTE* [45] (*Synthetic minority over-sampling technique*) increases the minority class by generating new "synthetic" instances based on its number of nearest neighbours. SMOTE, as implemented in WEKA, was used to generate synthetic examples. For our study, five nearest neighbours of a real existing instance (minority class) were used to compute a new synthetic one. For different datasets, different percentages of SMOTE instances were created, which can be found in the supplementary information (Table S1). The complete algorithm is explained in [45].
7. *ClassBalancer* [34, 35, 46] reweights the instances so that the sum of weights for all classes of instances in the data is the same, i.e. the total sum of weights across all instances is maintained. This is an additional way to treat class imbalance, unlike CostSensitiveClassifier or MetaCost, which try to minimize the total misclassification cost.

With respect to parameters, not for all classifiers a parameter optimization was performed. For instance, no parameters were adjusted for ClassBalancer since it automatically reassigns weights to the instances in the dataset such that each class has the same total weight [46]. For Bagging and Stratified Bagging, the only parameter to optimize would be the number of bags. In our case, the number of bags was adjusted to 64 as a previous study [38] suggests that generation of 64 models provides satisfactory results without exponentially increasing the computational cost. In case of ThresholdSelector, an optimal threshold was selected automatically via fivefold cross-validation before selecting the final model on the basis of FMeasure. For both CostSensitiveClassifier and MetaCost, the cost for misclassification was initially applied in accordance with the imbalance ratio, which, in case it did not provide a sensitivity of at least 0.5, was further increased to arrive at the final model. In case of SMOTE, similar principles were applied: initially, the number of the synthetic instances created was set to a number that balances the two classes. If insufficient, it was further increased until no further improvement in sensitivity (with no reduction in specificity) was observed. The detailed parameter settings of the best performing models for each method are provided in the supplementary material (Table S1).

Validation

All models were evaluated in a 10-fold cross-validation followed by an external validation performed on independent test sets, except for Bagging and Stratified Bagging. For Bagging and Stratified Bagging, since multiple training datasets were generated by selecting the molecules with replacement from training set in a random fashion, this leaves out about 37% of the instances in each run. Therefore, these molecules that constitute the 'out-of-the-bag' sets are later used for testing the performance of the final model.

Model performance assessment: selection of the optimal method

Prior to identifying the best performing method, an optimal model for each meta-classifier was selected. The best parameters for the model were selected using linear search (as explained in the "Methods" section). For all models, different performance measures including sensitivity (Eq. 1), specificity (Eq. 2), accuracy (Eq. 3), balanced accuracy (Eq. 4), Matthews correlation coefficient (MCC, Eq. 5), area under the curve (AUC) and precision (Eq. 6) were calculated. A model was considered eligible for selection if the 10-fold cross-validation provided a sensitivity value of at least 0.5 and a specificity value not less than 0.5. As the datasets are relevant to different toxicological endpoints, sensitivity was considered more important. For a highly imbalanced dataset, accuracy may be misleading. Therefore we considered balanced accuracy (which considers both sensitivity and specificity) as a more appropriate performance measure to compare different classifiers for their ability to handle imbalanced datasets. If two models provided the same sensitivity, the model that demonstrated higher balanced accuracy was prioritized for selection. Furthermore, 20 iterations were performed by varying the seed for cross validation [by assigning values from 1 (default) to 20]. For Bagging and Stratified Bagging, the 20 iterations were performed by changing the random seed for the Random Forest generation by assigning values from 1 (default) to 20. After cross-validation, average values for different performance measures were calculated and compared. The best method was then evaluated by performing a statistical t-test in R [47], as well as on the basis of the performance on external test sets. The individual settings used in selecting the best model for each meta-classifier can be found in the supplementary information (Table S1).

$$Sensitivity = \frac{TP}{(TP + FN)} \tag{1}$$

$$Specificity = \frac{TN}{(TN + FP)} \tag{2}$$

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{3}$$

$$Balanced\ Accuracy = \frac{1}{2}\left(\frac{(TP)}{(TP + NP)} + \frac{(TN)}{(TN + FP)}\right) \tag{4}$$

$$MCC = \frac{\{(TP \times TN) - (FP \times FN)\}}{\{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)\}^{1/2}} \tag{5}$$

$$Precision = \frac{(TP)}{(TP + FP)} \tag{6}$$

$$FMeasure = \frac{2TP}{(2TP + FP + FN)} \tag{7}$$

TP: true positives; TN: true negatives; FP: false positives; FN: false negatives.

Results and discussion

Tables S2–S5 in the supplementary material report the performance measures for predictions on all datasets used in this study. The performance values of the base-classifier (Random Forest) are also reported to facilitate a comparison with the investigated methods. For each dataset, the mean and the standard deviation values of performance of the best performing models (based on 20 iterations) were calculated and are reported in Tables S6–S9 (supplementary material). Figure 1a–c, Figure S1(a–d) in the supplementary material provide a comparison of performances of different meta-classifiers on the three test datasets (no test set available for animal cholestasis) and four training sets respectively.

Irrespective of the dataset and the descriptor set used, Random Forest was found to be the weakest performing classifier as anticipated. Except on the test dataset for human cholestasis, Random Forest alone did not yield a sensitivity greater than 0.5, which indicates that assistance of a meta-classifier indeed consistently improves performance when handling imbalanced datasets. Among the Meta-Classifier based methods, bagging provided the lowest performance. A simple reason behind the failure of Bagging is that it only

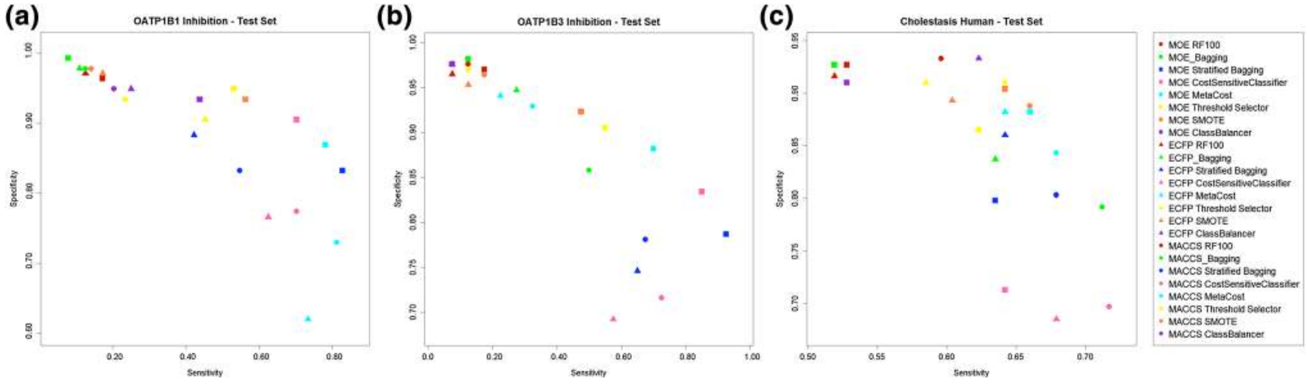


Fig. 1 Comparison of performances of different meta-classifiers on test sets **a** OATP1B1 inhibition **b** OATP1B3 inhibition **c** human cholestasis. *x-axis* corresponds to the sensitivity and on the *y-axis* is the specificity. The squares correspond to MOE descriptors, the triangles correspond to ECFP6 fingerprints and the circles correspond to MACCS fingerprints. Each classifier is depicted in a different color: red for RF standalone, green for Bagging, blue for Stratified Bagging, dark pink for CostSensitiveClassifier, cyan for MetaCost, yellow for ThresholdSelector, orange for SMOTE and dark violet for ClassBalancer. Please note that the scaling for the two axes are different

does resampling without any effort to balance or weight the two classes.

Threshold Selection was frequently found to be among the good performing methods. In many cases, this classifier could handle imbalance very well. However, the sensitivity measures were poor in comparison to other classifiers. This could be due to the fact that the thresholds were selected on the basis of FMeasure, as accuracy and specificity are not suitable due to the high impact of the majority class. If the selection of best models is done purely on the basis of sensitivity, this classifier yields very good sensitivity values (0.8–1.0), however with a radical decrease in specificity (0.2–0). Notably, Threshold Selection provided better results in combination with a second meta-classifier. But since the aim of the study was to compare the classifiers individually, this trend was not investigated further.

Stratified Bagging, CostSensitiveClassifier and MetaCost were consistently the best performing classifiers in both cross-validation and test set validation for all the datasets (see Fig. 1, Figure S1 in the supplementary material). Further, the t-test on the basis of 95% confidence interval (exact p-values not shown here) indicated a statistically significant difference in performance between the selected methods (meta-classifiers). The statistical test was performed pair-wise for all the obtained performance measures, with more stress on sensitivity and balanced accuracy. Both MetaCost and CostSensitiveClassifier tended to yield higher sensitivities while Stratified Bagging, on the other hand, was found to be superior in terms of MCC, balanced accuracy and AUC. An advantage of Stratified Bagging is that it is a straightforward method with only one parameter to optimize, i.e. the number of bags. On the other hand, cost-sensitive approaches tend to give more weight to sensitivity when needed, which is an advantage for toxicity prediction. Although both methods provided comparable performances, the cost that had to be applied was greater in case of CostSensitiveClassifier in comparison to MetaCost. This is due to the fact that the latter is a hybrid classifier which combines Bagging with the application of a cost, thus equilibrating the dataset more easily. It should further be noted that the computational cost for MetaCost is higher than that for CostSensitiveClassifier. On the other hand, Stratified Bagging is not computationally demanding (for the optimal parameter of 64 bags). Since each bag is double the size of the minority class, the calculation of models using Stratified Bagging requires less computational time, compared to the models built using Bagging (the bags are of the same size as the training set) and MetaCost (includes both bagging and weighting).

SMOTE and ClassBalancer were only in a few cases able to provide a sensitivity of at least 0.5 in both cross-validation and test set evaluation. Considering its reputation in handling such problems, the poor performance of SMOTE was quite surprising. We assume that the small size of the datasets could be the primary reason behind SMOTE's poor performance. The datasets used in this study are much smaller in size compared to the HTS datasets in which the minority class has enough instances for SMOTE to generate synthetic instances, although the overall imbalance ratio is typically in the range of 100:1 [12, 45, 48].

With respect to the different sets of descriptors used, the performance of the classifiers on different datasets remained almost the same. Of all the descriptors, 2D MOE descriptors and MACCS fingerprints provided the best performance across many of the datasets, while ECFP6 fingerprints consistently performed lower. Considering the amount of information encoded in ECFP6 (1024 bits) in comparison to MACCS fingerprints (166 bits) and the MOE descriptors, it might be assumed that the poor performance of ECFP6 is subject to the individual datasets in this study. This also highlights the fact that sometimes simple set of descriptors could provide better results than complex and highly populated descriptors. Moreover, in other recent studies [49–51] different descriptor and fingerprint combinations did not demonstrate significant differences in performance.

Overall, the best classifiers performed well regardless of the type of data (toxicity endpoint or a general or specific in vitro endpoint), the type and number of descriptor sets used, or the degree of class imbalance. However, there were instances where a dataset related to in vivo toxicity (animal cholestasis) could not be successfully handled by the best classifiers. Finally, highly sophisticated meta-classifiers such as Stratified Bagging and MetaCost, that combine re-sampling and a way to weight the two classes, performed in principle better than Bagging and ClassBalancer.

Conclusions

In this study, we compared the performance of seven different meta-classifiers for their ability to handle imbalanced datasets. We demonstrated that, for all datasets used in the study, Stratified Bagging performed at least as good as cost-sensitive approaches such as MetaCost and CostSensitiveClassifier and in most cases outperformed them. Random Forest (as a standalone classifier) and Bagging were unable to address the imbalance issue. Interestingly, the choice of descriptors did not play a substantial role in ranking the performance of different classifiers. Thus, considering that Stratified Bagging can be directly used in combination with any machine-learning method without parameter optimization, a general recommendation for handling imbalanced datasets is to wrap the modeling process in the stratified bagging loop. However, one should also consider the computational cost, as extensive re-sampling can be computationally expensive. Therefore, a method that balances between the complexity of the algorithm and computational cost would be an ideal choice to obtain optimal results.

Acknowledgements Open access funding provided by Austrian Science Fund (FWF). We gratefully acknowledge financial support provided by the Austrian Science Fund, Grants #F03502 (SFB35) and W1232 (MolTag).

References

1. Kotsiantis SB (2008) Handling imbalanced data sets with a modification of Decorate algorithm. Int J Comput Appl Technol 33:91–98. https://doi.org/10.1504/IJCAT.2008.021931
2. Kotsiantis S, Kanellopoulos D, Pintelas P (2006) Handling imbalanced datasets: a review. GESTS Int Trans Comput Sci Eng 30(1):25–36
3. Ali A, Shamsuddin SM, Ralescu AL (2015) Classification with class imbalance problem: a review. Int J Adv Soft Comput Appl 7:176–204
4. López V, Fernández A, Moreno-Torres JG, Herrera F (2012) Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification: open problems on intrinsic data characteristics. Expert Syst Appl 39:6585–6608. https://doi.org/10.1016/j.eswa.2011.12.043
5. Qiao X, Liu Y (2009) Adaptive weighted learning for unbalanced multicategory classification. Biometrics 65:159–168. https://doi.org/10.1111/j.1541-0420.2008.01017.x
6. Fernández A, Jesus MJ, del Herrera F (2010) Multi-class imbalanced data-sets with Linguistic fuzzy rule based classification systems based on pairwise learning. In: Hüllermeier E, Kruse R, Hoffmann F (eds) Computational intelligence for knowledge-based systems design. Springer, Berlin, pp 89–98
7. Galar M, Fernández A, Barrenechea E et al (2012) A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans Syst Man Cybern Part C 42:463–484. https://doi.org/10.1109/TSMCC.2011.2161285
8. He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21:1263–1284. https://doi.org/10.1109/TKDE.2008.239
9. Lin W-J, Chen JJ (2013) Class-imbalanced classifiers for high-dimensional data. Brief Bioinform 14:13–26. https://doi.org/10.1093/bib/bbs006
10. Garcia V, Sanchez JS, Mollineda RA, Alejo R, Sotoca JM (2007) The class imbalance problem in pattern classification and learning. In: II Congreso Español de Informática, Tamida, Saragossa, Spain, pp 283–291
11. Schierz AC (2009) Virtual screening of bioassay data. J Cheminform 1:21. https://doi.org/10.1186/1758-2946-1-21
12. Zakharov AV, Peach ML, Sitzmann M, Nicklaus MC (2014) QSAR modeling of imbalanced high-throughput screening data in PubChem. J Chem Inf Model 54:705–712. https://doi.org/10.1021/ci400737s
13. Razzaghi T, Roderick O, Safro I, Marko N (2016) Multilevel weighted support vector machine for classification on healthcare data with missing values. PLoS ONE 11:e0155119. https://doi.org/10.1371/journal.pone.0155119
14. Schlieker L, Telaar A, Lueking A et al (2017) Multivariate binary classification of imbalanced datasets-A case study based on high-dimensional multiplex autoimmune assay data. Biom J Biom Z 59:948–966. https://doi.org/10.1002/bimj.201600207
15. Chen J, Tang YY, Fang B, Guo C (2012) In silico prediction of toxic action mechanisms of phenols for imbalanced data with Random Forest learner. J Mol Graph Model 35:21–27. https://doi.org/10.1016/j.jmgm.2012.01.002
16. Khalilia M, Chakraborty S, Popescu M (2011) Predicting disease risks from highly imbalanced data using Random Forest. BMC Med Inform Decis Mak 11:51. https://doi.org/10.1186/1472-6947-11-51
17. Barta G (2016) Identifying biological pathway interrupting toxins using multi-tree ensembles. Front Environ Sci. https://doi.org/10.3389/fenvs.2016.00052
18. Koutsoukas A, St. Amand J, Mishra M, Huan J (2016) Predictive toxicology: modeling chemical induced toxicological response combining circular fingerprints with Random Forest and support vector machine. Front Environ Sci. https://doi.org/10.3389/fenvs.2016.00011
19. Kotsampasakou E, Brenner S, Jäger W, Ecker GF (2015) Identification of novel inhibitors of organic anion transporting polypeptides 1B1 and 1B3 (OATP1B1 and OATP1B3) using a consensus vote of six classification models. Mol Pharm 12:4395–4404. https://doi.org/10.1021/acs.molpharmaceut.5b00583
20. Mulliner D, Schmidt F, Stolte M et al (2016) Computational models for human and animal hepatotoxicity with a global application scope. Chem Res Toxicol 29:757–767. https://doi.org/10.1021/acs.chemrestox.5b00465
21. Kotsampasakou E, Ecker GF (2017) Predicting drug-induced cholestasis with the help of hepatic transporters—an in silico modeling approach. J Chem Inf Model 57:608–615. https://doi.org/10.1021/acs.jcim.6b00518
22. Kullak-Ublick G (2003) Drug-induced cholestatic liver disease. In: Trauner M, Jansen P, (eds) Mol Pathog Cholestasis. Springer, New York, pp 271–280
23. Mita S, Suzuki H, Akita H et al (2006) Inhibition of bile acid transport across Na+/taurocholate co transporting polypeptide (SLC10A1) and bile salt export pump (ABCB 11)-coexpressing LLC-PK1 cells by cholestasis-inducing drugs. Drug Metab Dispos Biol Fate Chem 34:1575–1581. https://doi.org/10.1124/dmd.105.008748
24. Padda MS, Sanchez M, Akhtar AJ, Boyer JL (2011) Drug induced cholestasis. Hepatol Baltim Md 53:1377–1387. https://doi.org/10.1002/hep.24229
25. Van den Hof WFPM., Coonen MLJ, van Herwijnen M et al (2014) Classification of hepatotoxicants using HepG2 cells: a proof of principle study. Chem Res Toxicol 27:433–442. https://doi.org/10.1021/tx4004165
26. Kuhn M, Campillos M, Letunic I et al (2010) A side effect resource to capture phenotypic effects of drugs. Mol Syst Biol 6:343. https://doi.org/10.1038/msb.2009.98
27. Kuhn M, Letunic I, Jensen LJ, Bork P (2016) The SIDER database of drugs and side effects. Nucleic Acids Res 44:D1075-1079. https://doi.org/10.1093/nar/gkv1075
28. Molecular Operating Environment (MOE), 2013.08. Chemical Computing Group Inc., 1010 Sherbooke St. West, Suite #910. Montreal, QC
29. Atkinson F (2014) Standardiser
30. Sadowski J, Gasteiger J, Klebe G (1994) Comparison of automatic three-dimensional model builders using 639 X-ray structures. J Chem Inf Comput Sci 34:1000–1008. https://doi.org/10.1021/ci00020a039

31. Landrum G (2006) RDKit: Open-source cheminformatics
32. Yap CW (2011) PaDEL-descriptor: an open source software to calculate molecular descriptors and fingerprints. J Comput Chem 32:1466–1474. https://doi.org/10.1002/jcc.21707
33. Breiman L (2001) Random Forests. Mach Learn 45:5–32
34. Hall M, Frank E, Holmes G et al (2009) The WEKA data mining software: an update. SIGKDD Explor Newsl 11:10–18. https://doi.org/10.1145/1656274.1656278
35. University of Waikato: Waikato, New Zeland Weka: Waikato Environment for Knowledge Analysis. http://www.cs.waikato.ac.nz/ml/weka/. Accessed 2 Nov 2010
36. Oshiro TM, Perez PS, Baranauskas JA (2012) How many trees in a Random Forest? In: Machine learning and data mining in pattern recognition. Springer, Berlin, pp 154–168
37. Breiman L (1996) Bagging predictors. Mach Learn 24:123–140. https://doi.org/10.1023/A:1018054314350
38. Tetko IV, Novotarskyi S, Sushko I et al (2013) Development of dimethyl sulfoxide solubility models using 163,000 molecules: using a domain applicability metric to select more reliable predictions. J Chem Inf Model. https://doi.org/10.1021/ci400213d
39. Sushko I, Novotarskyi S, Körner R et al (2011) Online chemical modeling environment (OCHEM): web platform for data storage, model development and publishing of chemical information. J Comput Aided Mol Des 25:533–554. https://doi.org/10.1007/s10822-011-9440-2
40. On-line CHEmical database and Modelling environment (OCHEM). https://www.ochem.eu. Accessed 7 Apr 2013
41. Domingos P (1999) MetaCost: a general method for making classifiers cost-sensitive. In: Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining. ACM Press, pp 155–164
42. ThresholdSelector. http://weka.sourceforge.net/doc.packages/thresholdSelector/weka/classifiers/meta/ThresholdSelector.html. Accessed 16 Jul 2017
43. Chawla NV, Japkowicz N, Kotcz A (2004) Editorial: special issue on learning from imbalanced data sets. SIGKDD Explor Newsl 6:1–6. https://doi.org/10.1145/1007730.1007733
44. Powers D (2011) Evaluation: from precision, recall and f-measure to roc., informedness, markedness & correlation. J Mach Learn Technol 2:37–63
45. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. J Artif Int Res 16:321–357
46. ClassBalancer. http://weka.sourceforge.net/doc.dev/weka/filters/supervised/instance/ClassBalancer.html. Accessed 16 Jul 2017
47. R Core Team (2013). R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna
48. Su B-H, Tu Y-S, Lin OA et al (2015) Rule-based classification models of molecular autofluorescence. J Chem Inf Model 55:434–445. https://doi.org/10.1021/ci5007432
49. Duan J, Dixon SL, Lowrie JF, Sherman W (2010) Analysis and comparison of 2D fingerprints: insights into database screening performance using eight fingerprint methods. J Mol Graph Model 29:157–170. https://doi.org/10.1016/j.jmgm.2010.05.008
50. Drwal MN, Siramshetty VB, Banerjee P et al (2015) Molecular similarity-based predictions of the Tox21 screening outcome. Front Environ Sci. https://doi.org/10.3389/fenvs.2015.00054
51. Drwal MN, Banerjee P, Dunkel M et al (2014) ProTox: a web server for the in silico prediction of rodent oral toxicity. Nucleic Acids Res 42:W53–W58. https://doi.org/10.1093/nar/gku401

Affiliations

Sankalp Jain[1] · Eleni Kotsampasakou[1,2] · Gerhard F. Ecker[1]

✉ Gerhard F. Ecker
gerhard.f.ecker@univie.ac.at

[1] Department of Pharmaceutical Chemistry, University of Vienna, Althanstrasse 14, 1090 Vienna, Austria

[2] Present Address: Computational Toxicology Group, CMS, R&D Platform Technology & Science, GSK, Park Road, Ware, Hertfordshire SG12 0DP, UK