Machine Learning to Predict Toxicity of Compounds

Ingrid Grenet[1], Yonghua Yin[2], Jean-Paul Comet[1], and Erol Gelenbe[1,2]

[1] University Côte d'Azur, I3S laboratory, UMR CNRS 7271, CS 40121, 06903 Sophia Antipolis Cedex, France
[2] Intelligent Systems and Networks Group, Department of Electrical and Electronic Engineering, Imperial College, London, UK

Abstract. Toxicology studies are subject to several concerns, and they raise the importance of an early detection of the potential for toxicity of chemical compounds which is currently evaluated through *in vitro* assays assessing their bioactivity, or using costly and ethically questionable *in vivo* tests on animals. Thus we investigate the prediction of the bioactivity of chemical compounds from their physico-chemical structure, and propose that it be automated using machine learning (ML) techniques based on data from *in vitro* assessment of several hundred chemical compounds. We provide the results of tests with this approach using several ML techniques, using both a restricted dataset and a larger one. Since the available empirical data is unbalanced, we also use data augmentation techniques to improve the classification accuracy, and present the resulting improvements.

Keywords: Machine Learning, Toxicity, QSAR, Data Augmentation

1 Introduction

Highly regulated toxicology studies are mandatory for the marketing of chemical compounds to ensure their safety for living organisms and the environment. The most important studies are performed *in vivo* in laboratory animals during different times of exposure (from some days to the whole life-time of the animal). Also, in order to rapidly get some indication of a compound's effects, *in vitro* assays are performed using biological cell lines or molecules, to obtain hints about the bioactivity of chemicals, meaning their ability to affect biological processes. However, all of these studies raise ethical, economical and time concerns; indeed it would be ideal if the toxicity of chemical compounds could be assessed directly through physical, mathematical, computational and chemical means and processes.

Therefore, in order to predict as early as possible the potential toxic effect of a chemical compound, we propose to use machine learning (ML) methods. The ambitious objective is to predict long term effects that will be observed in *in vivo* studies, directly from chemical structure. Nonetheless, this long term prediction seems to be difficult [24] because of the high level of biological variability and because toxicity can result from a long chain of causality. Therefore, in this paper we investigate whether taking into consideration the *in vitro* data, can improve the quality of the prediction. In such a case the global objective of the long term toxicity prediction could be split into two parts: (i) first the prediction of *in vitro* bioactivity from chemical structure [27], and (ii) secondly the prediction of long term *in vivo* effects from *in vitro* bioactivity [23].

Here we focus on the first part (i) using ML approaches to determine a "quantitative structure-activity relationship" (QSAR) [17]. QSAR models aim at predicting any kind of compounds activity based on their physico-chemical properties and structural descriptors. Our purpose is to predict using an ML approach, whether a compound's physico-chemical properties, can be used to determine whether the compound will be biologically active during *in vitro* assays. If ML could be shown to be effective in this respect, then it would serve to screen compounds and prioritize them for further *in vivo* studies. Then, *in vivo* toxicity studies would only be pursued with the smaller set of compounds that ML has indicated as being less bioactive, and which must then be certified via *in vivo* assessment. Thereby a significant step forward would be achieved, since animal experimentation could be reduced significantly with the help of a relevant ML based computational approach.

This paper is organized as follows. Section 2 details the data, algorithms and performance metrics used in this work. Section 3 presents the first results obtained on a subset of data. Section 4 shows the performance of an algorithm on the global dataset. Finally, we conclude in Section 5.

2 Learning Procedure

In this section we first describe the data used, then the ML algorithms that are tested and finally the metrics used to evaluate performances of the models.

2.1 Data Description

Since the long term objective aims at predicting *in vivo* toxicity, we need publicly available data for both *in vivo* and *in vitro* experimental results. The US Environmental Protection Agency (EPA) released this type of data in two different databases: (i) ToxCast database contains bioactivity data obtained for around 10,000 of compounds tested in more than several hundreds *in vitro* assays [7], (ii) the Toxicity Reference database (ToxRefDB) gathers results from several types of *in vivo* toxicity studies performed for several hundreds of chemicals [20]. It is important to notice that not all the compounds have been tested in all the assays from ToxCast and in each type of *in vivo* studies present in ToxRefDB.

Still guided by the long term objective, we consider a subset of these data including compounds for which both *in vitro* and *in vivo* results were available. The subset selection follows three steps. First, we look for the overlap of compounds present both in ToxCast and ToxRefDB and having results for *in vivo* studies performed in rats during two years. We obtain a matrix with 418 compounds and 821 assays, with a lot of missing values. Secondly, we look for a large complete sub-matrix and we obtain a matrix of 404 compounds and 60 *in vitro* assays. Finally, in order to be sure to get a minimum of active compounds in the datasets, i.e compounds for which an AC50 (half maximal activity concentration), could be measured, we remove assays with less than 5% of them and obtain a final matrix of 404 compounds and 37 assays.

For each of the 37 assays, we build a QSAR classification model to predict the bioactivity of a compound. These models use structural descriptors computed from the

compound's structure described in Structured Data Files. Two types of descriptors are used: (i) 74 physico-chemical properties (*e.g.* molecular weight, logP, *etc.*) which are continuous and normalized variables and (ii) 4870 fingerprints which are binary vectors representing the presence or absence of a chemical sub-structure in a compound [21]. Fingerprints being present in less than 5% of compounds are removed, leading to a final set of 731 fingerprints. Therefore, the obtained dataset is composed of 805 structural descriptors for the 404 compounds.

The property that we wish to predict, is the activity in each *in vitro* assay in a binarised form. It is generally measured as a AC50 value which is the dose of compound required to obtain 50% of activity in the assay. In the following, we consider that the binary version of the activity is 0 if AC50 value equals 0 and 1 otherwise.

2.2 Learning algorithms

- **The Random Neural Network (RNN)** is a mathematical model of the spiking (impulse-like) probabilistic behaviour of biological neural systems [11,9] and it has been shown to be a universal approximator for continuous and bounded functions [10]. It has a compact computationally efficient "product form solution", so that in steady-state the joint probability distribution of the states of the neurons in the network can be expressed as the product of the marginal probabilities for each neuron. The probability that any cell is excited satisfies a non-linear continuous function of the states of the other cells, and it depends on the firing rates of the other cells and the synaptic weights between cells. The RNN has been applied to many pattern analysis and classification tasks [6]. Gradient descent learning is often used for the RNN, but in this work we determine weights of the RNN using the cross-validation approach in [28].
- **The Multi Layer RNN (MLRNN)** uses the original simpler structure of the RNN and investigates the power of single cells for deep learning [25]. It achieves comparable or better classification at much lower computation cost than conventional deep learning methods in some applications. A cross-validation approach is used to determine the structure and the weights and 20 trials are conducted to average the results. The structure of the MLRNN used here is fixed as having 20 inputs and 100 intermediate nodes.
- **The Convolutional Neural Network (CNN)** is a deep-learning tool [18] widely used in computer vision. Its weight-sharing procedure improves training speed with the stochastic gradient descent algorithm recently applied to various types of data [26,15]. In this work, we use it with the following layers: "input-convolutional-convolutional-pooling-fully*connected-output" [5].
- **Boosted Trees** (called XGBoost in the sequel) is a popular tree ensemble method (such as Random Forest). The open-source software library XGBoost [4] provides an easy-to-use tool for implementing boosted trees with gradient boosting [8] and regression trees.

2.3 Classification Settings and Performance Metrics

For each of the 37 assays, we randomly subdivide the corresponding dataset D into a training set D_T and a testing set D_t. From D we randomly create 50 instances of D_T and

its complementary test set D_t so that for each instance, $D = D_T \cup D_t$. Each of the ML techniques listed above are first trained on each D_T and then tested on D_t. The results we present below are therefore averages over the 50 randomly selected training and testing sets. Since the output of the datasets is either 0 or 1, this is a binary classification problem.

Let *TP*, *FP*, *TN* and *FN* denote the number of true positives, false positives, true negatives and false negatives, respectively. Then the performance metrics that we use to evaluate the results are the $Sensitivity$ $(TP/(TP+FN))$, the $Specificity$ $(TN/(TN+FP))$ and the $BalancedAccuracy$, denoted for short BA $((Sensitivity+Specificity)/2)$.

3 Classification Results

In the 37 datasets corresponding to the 37 assays, the ratio between positive and negative compounds varies between 5% and 30% with a mean around 12%. This highlights the unbalanced property of the data in the favor of negative compounds. Here we test the ML algorithms on these unbalanced data and after balancing using data augmentation.

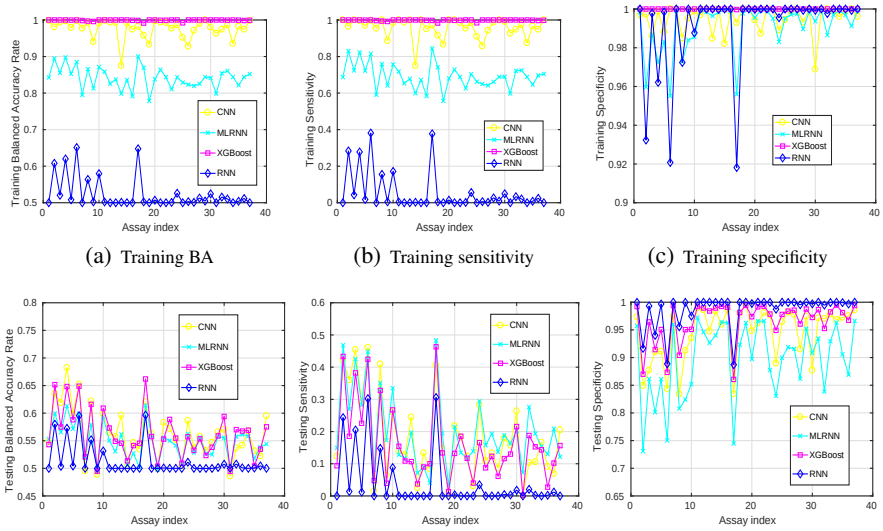
3.1 Results on Unbalanced Datasets

The MLRNN, RNN, CNN and XGBoost algorithms are exploited to classify the 50×37 pairs of training and testing datasets and results are summarized into Figure 1. Since these are unbalanced datasets, the BA may be a better metric to demonstrate the classification accuracy. In addition, the situation of misclassifying positive as negative may be less desirable than that of misclassifying negative as positive. Therefore, the metric of $Sensitivity$ is also important.

When looking at the BA obtained on the training data set (Figure 1(a)), we observe that the RNN method is not good at learning from these unbalanced datasets, while the CNN, MLRNN and XGBoost techniques learn much better.

Compared to the training accuracy, the performance on the testing dataset is more important since it demonstrates whether the model generalises accurately with regard to classifying previously unseen chemical compounds. The testing results are presented in Figures 1(d) to 1(f). Here, we see that RNN performs the worst in identifying true positives ($Sensitivity$) and tends to classify most unseen chemical compounds as inactive, except for some assays. It can be explained by the overall number of inactive compounds much larger than the number of active compounds in the training dataset. The CNN, MLRNN and XGBoost perform a bit better in identifying the TPs, and the MLRNN performs the best. But $Sensitivity$ is still low and really depends on the assays and probably on the balance between active and inactive compounds in the corresponding datasets.

Among all assays, the highest testing *BA* achieved by these classification tools is 68.50% attained by the CNN for assay number 4, with the corresponding $Sensitivity$ being 47.10%. Among all assays, the highest testing $Sensitivity$ is 47.75% (MLRNN for assay 17) with a corresponding *BA* of 60.80%.



(a) Training BA (b) Training sensitivity (c) Training specificity

(d) Testing BA (e) Testing sensitivity (f) Testing specificity

Fig. 1. Training and testing mean-value results (Y-axis) versus different assays (X-axis) when the CNN, MLRNN, XGBoost, RNN are used for classification.

3.2 Results on Balanced Datasets

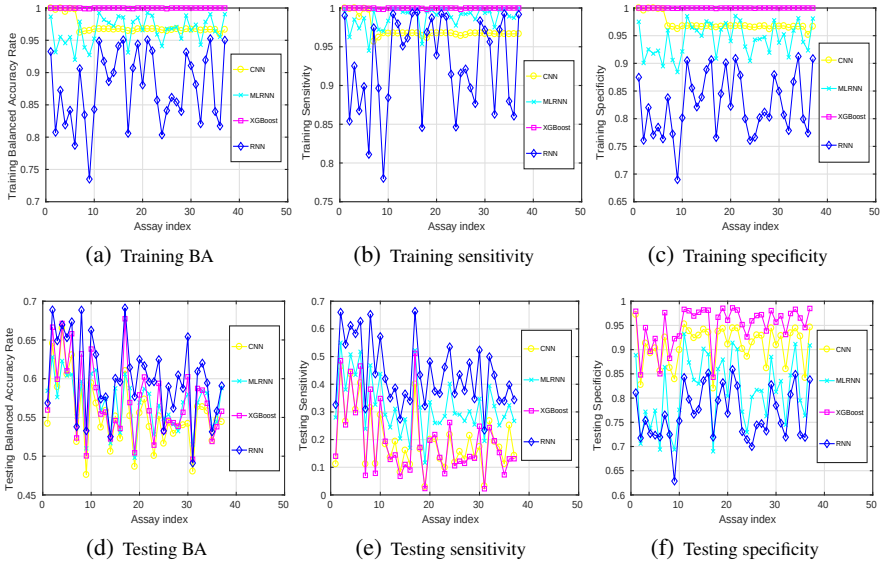
From the previous results, it appears that most of the classification techniques used are not good at learning unbalanced datasets. Therefore, we try balancing the 50×37 training datasets with data augmentation, while the corresponding testing datasets remain unchanged.

Here, the CNN, MLRNN, RNN and XGBoost are used to learn from the 50×37 datasets which are augmented for balanced training using the SMOTE method [3] as implemented in the Python toolbox *unbalanced_learn* [19]. The resulting $Sensitivity$, $Specificity$ and BA are summarised in Figure 2.

Compared to the training balanced accuracies given in Figure 1(a), Figure 2(a) shows that it is now evident that all the classification techniques we have discussed are capable of learning the training datasets after data augmentation. The training BA of the RNN method is still the lowest, but its testing BA is the highest for most of the assays.

Among all assays, the highest testing BA is 68.88% which is obtained with the RNN for the assay 17, with the corresponding testing $Sensitivity$ being 66.% and which is also the highest testing $Sensitivity$ observed. Note that these values are higher than those reported in Figure 1.

Finally, for a better illustration, Figure 3 compares the highest testing results obtained among all classification tools for classifying the datasets before and after data augmentation. This figure highlights the clear improvement of $Sensitivity$ for all assays, which also leads to a better BA for most of them. Not surprisingly, $Specificity$ is decreased after data augmentation since the proportion of negatives in the balanced

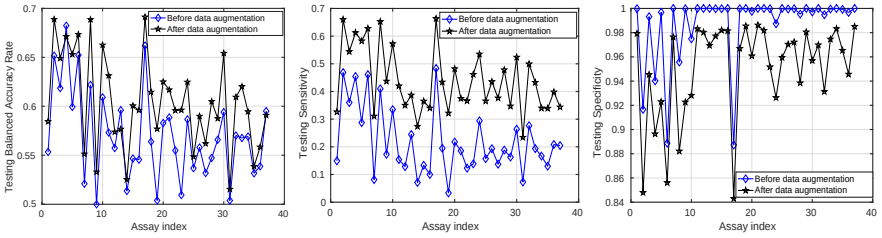


(a) Training BA (b) Training sensitivity (c) Training specificity

(d) Testing BA (e) Testing sensitivity (f) Testing specificity

Fig. 2. Training and testing mean-value results (Y-axis) versus different assays (X-axis) on balanced datasets.

training sets is much lower compared to the original ones. Therefore, the models do not predict almost everything as negative as they did before data augmentation.



(a) Highest testing BA (b) Highest testing sensitivity (c) Highest testing specificity

Fig. 3. Comparison between the highest testing results (Y-axis) versus different assay index (X-axis) on both unbalanced and balanced datasets.

4 Classification Results on Extended Datasets

4.1 New datasets and learning procedure

In this section we use a bigger dataset of 8318 compounds to classify the same 37 assays. This 8318×37 matrix is not complete since not all the compounds were tested

in all the assays. Thus, for each of the 37 assays, we build a classification model based on the compounds which were actually tested in the assay, leading to different datasets for each assay. Note that, as previously, the instance numbers of the two classes are very unbalanced.

Compared to the previous datasets, all the generated fingerprints are included in the global dataset which corresponds to 4870 fingerprints in total (added to the 74 molecular descriptors previously described). Nonetheless, for each of the 37 assays and before the learning, a descriptor selection is performed based on two steps: (i) descriptors having a variance close to 0 (in such case, they are not sufficiently informative) are removed, (ii) Fisher test is computed between each descriptor and the output assay and descriptors are ranked according to the obtained p-value; we keep the 20% best descriptors.

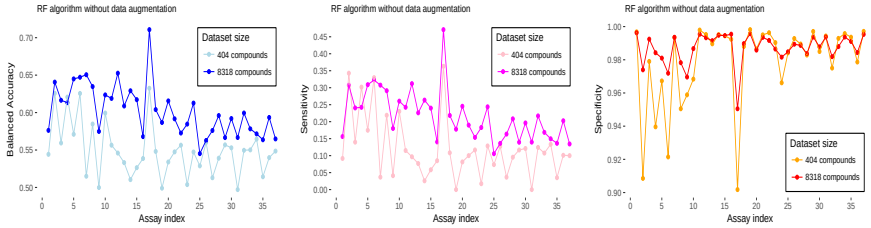
Random Forest (RF) classifier, an ensemble technique that combines many decision trees built using random subsets of training examples and features [2], is used for the learning because is has the advantage to deal with a large number of features without overfitting. A 10-fold cross-validation is performed 10 times and the average $Sensistivity$, $Specificity$ and BA are computed to evaluate the internal performance of the classifiers. As previously, we test the RF classifier on both unbalanced and balanced datasets.

4.2 Results on Unbalanced Datasets

Figure 4 presents the results obtained with the method described above applied to the datasets used in Section 3 as well as to the extended ones described in Section 4.1. We observe that, for both ensembles of datasets, the RF method is not good at identifying TPs ($Sensitivity < 50\%$) and is predicting almost all compounds as negatives ($Specificity > 90\%$). However, we see that the extended datasets lead to higher performance for most of the assays. Among all, the highest BA achieved by the RF is 71.08% for the assay 17 with corresponding $Sensitivity$ and $Specificity$ of 47.10% and 95.05% respectively. When looking at the distribution between active and inactive compounds in all assays, we see that the assay 17 is the one which has the less unbalanced dataset with 30% of actives in the initial dataset and 22% in the extended one. This could explain that this assay always lead to the best performances. Also, the percentage of active compounds for each assay in the extended dataset is always lower compared to the initial dataset (data not shown). Nevertheless, since the results are better with the extended dataset, it seems that the total number of observations has an impact on the results and not only the ratio between actives and inactives.

4.3 Results on Balanced Datasets

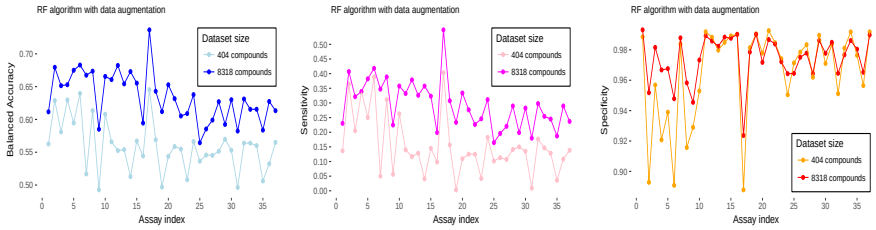
Figure 5 presents the results obtained with the same protocol but with the data augmentation method SMOTE applied to each training dataset of the cross-validation. As in Section 3, we observe that for extended datasets, all the results are improved after data augmentation ($Sensitivity$ is increased by 8% in average and BA by 3%). But still, the $Sensivity$ is low compared to the $Specificity$. Among all assays, the highest BA achieved by the RF on the extended dataset is 73.64% with corresponding $Sensitivity$ and $Specificity$ of 54.93% and 92.36% respectively, still for the assay 17. These results



(a) BA (b) Sensitivity (c) Specificity

Fig. 4. Results of RF algorithm (Y-axis) versus different assays (X-axis).

highlight that both the total number of compounds in the dataset and the ratio between active and inactive compounds have an impact on the performance of the models. Indeed, having a bigger dataset which is balanced allows increasing performances.



(a) BA (b) Sensitivity (c) Specificity

Fig. 5. Results of the RF algorithm (Y-axis) versus different assays (X-axis) on balanced datasets.

5 Conclusion and Perspectives

From the results presented here, we can draw several conclusions. First, the methods we have proposed can correctly predict bioactivity from the physico-chemical descriptors of compounds. However, some methods appear to be significantly better than others. Also, this appears to depend strongly on the assays themselves and their corresponding datasets. Moreover, we showed that the use of a larger dataset improves the classfication performance, even if the data is unbalanced. Furthermore, we see that data augmentation techniques can play an important role in classification performance for the unbalanced datasets.

This work on ML applied to toxicology data raises further interesting issues. Since there is no absolute winner among the classification techniques that we have used, we may need to test other methods such as Support Vector Machines (SVM) [1] or Dense Random Neural Networks (DenseRNN) [14]. Also, it would be interesting to apply the algorithms used on the small dataset to the extended one and compare against the RF

method. We may also test other data augmentation techniques to seek the most appropriate ones [16]. Futhermore, in order to assess the prediction accuracy of bioactivity for a new compound, it is important to know if this compound has a chemical structure that is similar to the ones used in the training set. For this, we could use the "applicability domain" approach [22] as a tool to define the chemical space of a ML model. Finally, if we refer to the long term objective of this work which is to link the molecular structure to *in vivo* toxicity, we could think about using the approach we have used as an intermediate step, and also train ML techniques to go from *in vitro* data to the prediction of *in vivo* effects. However, some preliminary tests that we have carried out (and not yet reported), reveal a poor correlation between *in vitro* and *in vivo* results, so that other data that is more directly correlated to toxicity, could be considered in future ML predictive models of toxicity. In addition, we could consider combining the results obtained with several ML methods, similar to a Genetic Algorithm based combination [13,12], to enhance the prediction accuracy.

References

1. Akbani, R., Kwek, S., Japkowicz, N.: LNAI 3201 - Applying Support Vector Machines to Imbalanced Datasets. LNAI 3201, 39–50 (2004)
2. Breiman, L.: Random Forests. Machine Learning 45, 5–32 (2001)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority oversampling technique. Journal of artificial intelligence research 16, 321–357 (2002)
4. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785–794. ACM (2016)
5. Chollet, F., et al.: Keras. `https://github.com/fchollet/keras` (2015)
6. Cramer, C.E., Gelenbe, E.: Video quality and traffic qos in learning-based subsampled and receiver-interpolated video sequences. IEEE Journal on Selected Areas in Communications 18(2), 150–167 (2000)
7. Dix, D.J., Houck, K.A., Martin, M.T., Richard, A.M., Setzer, R.W., Kavlock, R.J.: The ToxCast Program for Prioritizing Toxicity Testing of Environmental Chemicals. Toxicological Sciences 95(1), 5–12 (2007)
8. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Annals of statistics pp. 1189–1232 (2001)
9. Gelenbe, E.: Learning in the recurrent random neural network. Neural Computation 5(1), 154–164 (1993)
10. Gelenbe, E., Mao, Z.H., Li, Y.D.: Function approximation with spiked random networks. IEEE Transactions on Neural Networks 10(1), 3–9 (1999)
11. Gelenbe, E.: Réseaux neuronaux aléatoires stables. Comptes rendus de l'Académie des Sciences. Série 2, Mécanique, Physique, Chimie, Sciences de l'Univers, Sciences de la Terre 310(3), 177–180 (1990)
12. Gelenbe, E.: A class of genetic algorithms with analytical solution. Robotics and Autonomous Systems 22, 59–64 (1997)
13. Gelenbe, E.: Learning in genetic algorithms. In: International Conference on Evolvable Systems ICES 1998: Evolvable Systems: From Biology to Hardware. pp. 268–279. Springer Lecture Notes in Computer Science book series, LNCS, volume 1478 (1998)
14. Gelenbe, E., Yin, Y.: Deep learning with dense random neural networks. In: International Conference on Man–Machine Interactions 5, Proc. of the 5th International Conference on

Man-Machine Interactions, ICMMI 2017, Krakw, Poland, October 3-6, 2017. pp. 3–18. Advances in Intelligent Systems and Computing book series, volume 659 (2017)
15. Goh, G.B., Hodas, N.O., Vishnu, A.: Deep learning for computational chemistry. Journal of Computational Chemistry 38(16), 1291–1307 (2017)
16. Haibo He, Garcia, E.: Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)
17. Hansch, C.: Quantitative structure-activity relationships and the unnamed science. Accounts of chemical research 26(4), 147–153 (1993)
18. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
19. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research 18(17), 1–5 (2017)
20. Martin, M.T., Judson, R.S., Reif, D.M., Kavlock, R.J., Dix, D.J.: Profiling Chemicals Based on Chronic Toxicity Results from the U.S. EPA ToxRef Database. Environmental Health Perspectives 117(3), 392–399 (2009)
21. Rogers, D., Hahn, M.: Extended-Connectivity Fingerprints. Journal of Chemical Information and Modeling 50(5), 742–754 (2010)
22. Schultz, T.W., Hewitt, M., Netzeva, T.I., Cronin, M.T.D.: Assessing Applicability Domains of Toxicological QSARs: Definition, Confidence in Predicted Values, and the Role of Mechanisms of Action. QSAR & Combinatorial Science 26(2), 238–254 (2007)
23. Sipes, N.S., Martin, M.T., Reif, D.M., Kleinstreuer, N.C., Judson, R.S., Singh, A.V., Chandler, K.J., Dix, D.J., Kavlock, R.J., Knudsen, T.B.: Predictive Models of Prenatal Developmental Toxicity from ToxCast High-Throughput Screening Data. Toxicological Sciences 124(1), 109–127 (2011)
24. Thomas, R.S., Black, M.B., Li, L., Healy, E., Chu, T.M., Bao, W., Andersen, M.E., Wolfinger, R.D.: A Comprehensive Statistical Analysis of Predicting In Vivo Hazard Using High-Throughput In Vitro Screening. Toxicological Sciences 128(2), 398–417 (2012)
25. Yin, Y., Gelenbe, E.: Single-cell based random neural network for deep learning. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 86–93 (2017)
26. Yin, Y., Wang, L., Gelenbe, E.: Multi-layer neural networks for quality of service oriented server-state classification in cloud servers. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 1623–1627 (2017)
27. Zang, Q., Rotroff, D.M., Judson, R.S.: Binary Classification of a Large Collection of Environmental Chemicals from Estrogen Receptor Assays by Quantitative StructureActivity Relationship and Machine Learning Methods. Journal of Chemical Information and Modeling 53(12), 3244–3261 (2013)
28. Zhang, Y., Yin, Y., Guo, D., Yu, X., Xiao, L.: Cross-validation based weights and structure determination of chebyshev-polynomial neural networks for pattern classification. Pattern Recognition 47(10), 3414 – 3428 (2014)