Summer 2020

Machine Learning Approaches for Improving Prediction Performance of Structure-Activity Relationship Models

Gabriel Idakwo

MACHINE LEARNING APPROACHES FOR IMPROVING PREDICTION PERFORMANCE OF STRUCTURE-ACTIVITY RELATIONSHIP MODELS

by

Gabriel Anderson Idakwo

A Dissertation
Submitted to the Graduate School,
the College of Arts and Sciences
and the School of Computing Sciences and Computer Engineering
at The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved by:

Dr. Chaoyang Zhang, Committee Chair
Dr. Ping Gong
Dr. Zhaoxian Zhou
Dr. Dia Ali
Dr. Weihua Zhou

August 2020

Published by the Graduate School

THE UNIVERSITY OF
SOUTHERN
MISSISSIPPI®

ABSTRACT

In silico bioactivity prediction studies are designed to complement *in vivo* and *in vitro* efforts to assess the activity and properties of small molecules. *In silico* methods such as Quantitative Structure-Activity/Property Relationship (QSAR) are used to correlate the structure of a molecule to its biological property in drug design and toxicological studies. In this body of work, I started with two in-depth reviews into the application of machine learning based approaches and feature reduction methods to QSAR, and then investigated solutions to three common challenges faced in machine learning based QSAR studies.

First, to improve the prediction accuracy of learning from imbalanced data, Synthetic Minority Over-sampling Technique (SMOTE) and Edited Nearest Neighbor (ENN) algorithms combined with bagging as an ensemble strategy was evaluated. The Friedman's aligned ranks test and the subsequent Bergmann-Hommel post hoc test showed that this method significantly outperformed other conventional methods. It was also found that a strong negative correlation existed between the prediction accuracy and the imbalance ratio (IR), which is defined as the number of inactive compounds divided by the number of active compounds. SMOTEENN with bagging became less effective when IR exceeded a certain threshold (e.g., >40). The ability to separate the few active compounds from the vast amounts of inactive ones is of great importance in computational toxicology.

Deep neural networks (DNN) and random forest (RF), representing deep and shallow learning algorithms, respectively, were chosen to carry out structure-activity relationship-based chemical toxicity prediction. This is particularly important as picking the right algorithm that can best learn the underlying pattern in data is a major driver of success in QSAR studies. Results suggest that DNN significantly outperformed RF ($p < 0.001$, ANOVA) by 22-27% for four metrics (precision, recall, F-measure, and AUPRC) and by 11% for another (AUROC).

Lastly, current features used for QSAR based machine learning are often very sparse and limited by the logic and mathematical processes used to compute them. Transformer embedding features (TEF) were developed as new continuous vector descriptors/features using the latent space embedding from a multi-head self-attention often referred to as transformer architecture. The significance of TEF as new descriptors was evaluated by applying them to tasks such as predictive modeling, clustering, and similarity search. An accuracy of 84% on the Ames mutagenicity test indicates that these new features has a correlation to biological activity.

Overall, the findings in this study can be applied to improve the performance of machine learning based Quantitative Structure-Activity/Property Relationship (QSAR) efforts for enhanced drug discovery and toxicology assessments.

ACKNOWLEDGMENTS

I would like to thank Dr Chaoyang Zhang, my committee chair, and Dr Ping Gong, my advisor, for their relentless effort in guiding me through this program. Your direction and feedback have been immensely helpful. To the other members of my committee, Dr. Zhaoxian Zhou, Dr. Dia Ali and Dr. Weihua Zhou, I am deeply grateful for your insightful feedback.

To my family, you all have been my rock throughout this program. It really took a village. I am deeply appreciative of your support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

QSAR	Quantitative Structure-Activity/Property Relationship
SMOTE	Synthetic Minority Over-sampling Technique

ENN	Edited Nearest Neighbor
DNN	Deep Neural Networks
RF	Random Forest
AUPRC	Area Under the Precision Recall Curve
AUROC	Area Under the Receiver Operating Characteristic curve
ANOVA	Analysis of Variance
TEF	Transformer embedding features
MoRSE	Molecular Representation of Structures based on Electronic diffraction
WHIM	Weighted Holistic Invariant Molecular
GETAWAY	GEometry, Topology, and Atom Weights AssemblY
EVA	EigenVAlue descriptors
ECFP	Extended-Connectivity FingerPrints
MACCS	Molecular ACCess System
CDK	Chemistry Development Kit
SFS	Sequential Forward Selection
SBE	Sequential Backward Elimination
RFE	Recursive Feature Elimination
ML	Machine Learning
PCA	Principle Component Analysis
LDA	Linear Discriminant Analysis
ICA	Independent Component Analysis
SOM	Self Organizing Maps

SVM	Support Vector Machines
RBF	Radial Basis Function
KNN	*k*-Nearest Neighbors
RF	Random forest
DT	Decision Trees
CART	Classification and Regression Trees
ID3	Iterative Dichotomiser 3
DL	Deep Learning
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
GAN	Generative Adversarial Networks
SALI	Structure–Activity Landscape Index
MCC	Mathew’s Correlation Coefficient
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
R^2	Coefficient of Determination
RUS	Random Under-Sampling
AD	Applicability Domain
IR	Imbalance Ratio
NIH	National Institute of Health
EPA	Environmental Protection Agency
FDA	Food and Drug Administration
LBD	Ligand-Binding Domain

UAS	Upstream Activator Sequence
SID	Substance ID
CID	Compound ID
VAE	Variational AutoEncoder
LSTM	Long Short-Term Memory
NMT	Neural Machine Translation
SMILES	Simplified Molecular-Input Line-Entry System
SMART	SMILES Arbitrary Target Specification
IUPAC	International Union of Pure and Applied Chemistry
InChI	International Chemical Identifier

PREFACE

This dissertation is focused on developing innovative ways to improve the predictive performance of machine-leaning based structure-activity relationship (SAR) models. *In silico* toxicity and bioactivity studies such as SAR modeling, is designed to complement experimental efforts with a view toward improving the quality of bioactivity predictions for activity/safety assessment while decreasing the associated time, cost and ethical conflicts for generation of drug leads and assessment of toxicity.

Common challenges limiting the accuracy of SAR models include class imbalance which can be attribute to the high specificity of small molecules to target proteins. Other challenge includes the use of algorithms with the right level of complexity depending on the data available, and generation of highly informational low dimension feature vector that can be used to differentiate between molecules. The contributions of this body of work include:

- demonstration of SMOTEENN as a hybrid resampling technique coupled with bootstrap aggregation to improve SAR modeling on imbalanced dataset with imbalanced ratio less than 40. This study also confirms the inverse relationship between imbalance ratio and prediction performance. The significant of handling imbalance is very relevant as with cheminformatics data, it is almost guaranteed to be imbalanced due to the high specificity between small molecules and target proteins.
- deciding on the right complexity of algorithm to apply to an SAR problem is critical to finding active drug leads and filtering toxic compounds. This work demonstrates that the advantage provided by complex algorithms like deep learning comes with the next for extensive hyperparameter tuning. It also confirms that machine learning models have more difficulty discriminating between compounds with similar backbone structures but different bioactivity.

This finding can be applicable to drug repurposing tasks and provides an explanation for the sensitivity of SAR classification tasks.

- implementation of the theoretical principle that chemical string notations can be treated as human language text and the embedded vector space between two string representation translated using a multi-head self-attention network hold information rich feature. Considering that a model is only as good as the features it receives, this concept can vastly transform the use of embeddings in place of fingerprints and descriptors which are sparse and inconsistent.

Chapter I of this work provides an overview of the end-to-end machine learning process in SAR modeling. It is published as Idakwo, G., Luttrell, J., Chen, M., Hong, H., Zhou, Z., Gong, P., & Zhang, C. (2018). A review on machine learning methods for in silico toxicity prediction. Journal of Environmental Science and Health, Part C, 36(4), 169-191. In *Chapter II*, methods and importance of curating useful low dimension features (descriptors and fingerprint) that can adequately distinguish compounds are presented. *Chapter II* is published as Idakwo, G., Luttrell IV, J., Chen, M., Hong, H., Gong, P., & Zhang, C. (2019). A Review of Feature Reduction Methods for QSAR-Based Toxicity Prediction. In Advances in Computational Toxicology (pp. 119-139). Springer, Cham.

In the following chapters, I studied three existing challenges. In *Chapter III*, currently under review as Idakwo G, Thangapandian S, Luttrell J, Li Y, Wang N, Zhou Z, Hong H, Gong P, Zhang C. 2019. Structure-Activity Relationship-based Chemical

Classification of Highly Imbalanced Tox21 Datasets. Journal of Cheminformatics., synthetic minority over-sampling technique (SMOTE) and Edited Nearest Neighbor (ENN) are combined with bootstrap sampling to overcome the challenge of data imbalance.

Using the right machine learning algorithm with the appropriate level of complexity is critical to achieving a high performing SAR model. In *Chapter IV*, deep learning and random forest were employed as algorithms with varying complexities to evaluate the bioactivity of small molecules against the androgen receptor. This chapter is published as Idakwo, G., Thangapandian, S., Luttrell, J., Zhou, Z., Zhang, C., & Gong, P. (2019). Deep Learning-Based Structure-Activity Relationship Modeling for Multi-Category Toxicity Classification: A Case Study of 10K Tox21 Chemicals With High-Throughput Cell-Based Androgen Receptor Bioassay Data. Frontiers in physiology, 10, 1044.

In *Chapter V*, new continuous vector features were designed to be of variable dimension and to encode more useful information than conventional features using multi-head self-attention translation models. This chapter provides initial results and will be further developed prior to publishing. Lastly, a summary and my perspective of this work's contribution is detailed in *Chapter VI.*

CHAPTER I - MACHINE LEARNING METHODS FOR IN SILICO TOXICITY PREDICTION

1.1 Introduction

Computational approaches to understanding, predicting and preventing the adverse effect of chemicals on humans and other living organisms have gained prominence over the years (Greene & Pennie, 2015; Kruhlak, Benz, Zhou, & Colatsky, 2012; Perkins, Fang, Tong, & Welsh, 2003). Regulatory agencies and pharmaceutical companies are burdened with evaluating the toxicity profile of chemicals. Government regulatory agencies need to ensure public safety by mitigating contact with harmful chemicals in the environment that can be found in many places, ranging from food to household and industrial chemicals (R. Kavlock & Dix, 2010). In the pharmaceutical industry, compounds with a lower chance of eliciting toxicity must be prioritized early in the drug discovery process to avoid attrition and, consequently, a high development cost resulting in lower return on investment (Greene & Pennie, 2015; R. J. Kavlock et al., 2008; Segall & Barber, 2014).

Experimental toxicological approaches such as *in vivo* and *in vitro* methods can be used to assess the toxicity of new chemicals; however, these techniques alone are not considered to be the most efficient and humane. Consequently, in both pharmaceutical industry and regulatory decision-making process, there is a demand for more timely risk assessment, a reduction in the cost of evaluation, and methods that minimize the use of animal testing (Raies & Bajic, 2016).

Computational toxicology steps in to alleviate the stated challenges by applying interdisciplinary knowledge of advances in molecular biology, chemistry and computational science to increase the efficiency and the effectiveness by which the potential hazards and risks of chemicals are determined (R. J. Kavlock et al., 2008). Various methods have been adopted for the generation of models to predict toxicity endpoints. These methods include but are not limited to Read-Across and Trend Analysis (Patlewicz et al., n.d.), Dose and Time–Response Models (Raies & Bajic, 2016), toxicokinetic and toxicodynamic models, and Structure–Activity Relationship models (Greene & Pennie, 2015). In this review, emphasis is placed on Structure-Activity Relationship (SAR) models that use molecular descriptors and machine learning methods to predict toxicity endpoints.

An SAR model is a statistical/mathematical model used to establish an approximate relationship between a biological property of a compound and its structure-derived physicochemical and structural features (Cherkasov et al., 2014; Perkins et al., 2003; Alexander Tropsha, n.d.) in order to predict the activities of unknown molecules. The basic assumptions in SAR modeling are that molecules with similar structures exhibit similar biological activity, and that the physicochemical properties and/or structural properties of a molecule can be encoded as molecular descriptors to predict the biological activity of structurally related compounds. The independent variable is referred to as molecular descriptors generated from the structure of the molecule, while the dependent variable could be a numeric value of toxicity, such as LD_{50} in the case of quantitative SAR, or the classification of a compound as toxic versus nontoxic in a binary qualitative SAR model. Generally, the steps for developing a toxicity prediction model involve (see Figure 1.1): (1) data curation (gathering and cleaning data that relates chemicals to toxicity endpoints), (2) molecular descriptors generation, (3) prediction model development, and (4) model evaluation and validation.

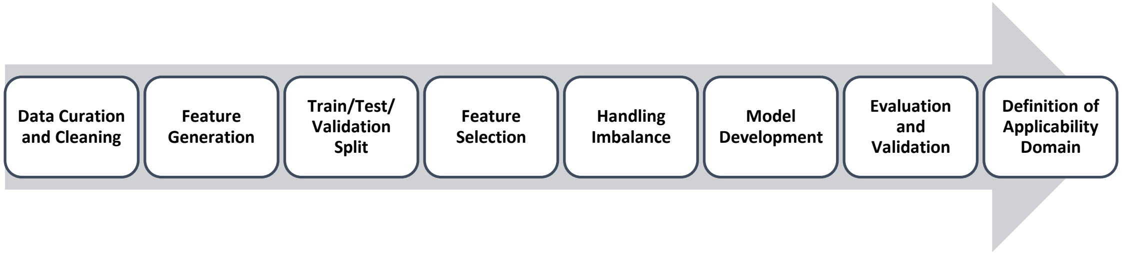

Data Curation and Cleaning
Feature Generation
Train/Test/ Validation Split
Feature Selection
Handling Imbalance
Model Development
Evaluation and Validation
Definition of Applicability Domain

Figure 1.1 A typical SAR-based modeling workflow

1.2 Data gathering and cleaning

1.2.1 Data curation

In machine learning, it is important to have large numbers of examples/instances (compounds) for a classifier to learn from while keeping an eye on quality. The goal is for the classifier to have enough examples to learn a pattern and approximate the statistical/mathematical relationship between the toxicity of a compound and its structure-derived features. The more diverse and less redundant the data set is, the more generalizable the model is likely to be. The chemical space of the data used to train a model affects the applicability domain as discussed in section 1.4.3. Table A.1 details some sources of data for *in-silco* toxicity prediction.

1.2.2 Preprocessing

It should be noted that, irrespective of data source, both *in vivo* and *in vitro* data are subject to numerous sources of errors and noise. As with any machine learning model, the predictive power of QSAR models is only as good as the chemical data on which they are trained. Having imperfections in the data used to train and evaluate models is often one of the reasons for the lack of predictive power observed with computational approaches.

In most cases, chemical structures are not used as inputs to machine learning models. Instead, descriptors calculated from chemical structures are used as numerical representations of the structures. Consequently, any error in the structure of a compound will be expressed in the descriptors that are serving as variables in the training data. Such erroneous descriptors could result in non-robust and weak models. The importance of paying attention to the quality of chemical structures in the data set has been reported in literature (Mansouri, Grulke, Richard, Judson, & Williams, 2016; Young, Martin, Venkatapathy, & Harten, 2008; Zhao, Wang, Sedykh, & Zhu, 2017). Tropsha (Alexander Tropsha, n.d.) demonstrated that the presence or absence of structural errors in a library and the choice of descriptors had a greater impact on performance than model optimization. Hence, a need to pay attention to systematic chemical curation protocols prior to modeling. Fourche *et al*. (Fourches, Muratov, & Tropsha, 2016) provide a reproducible workflow for cleaning up chemical data prior to developing a model.

The methods and steps involved in cleaning up a chemical library often vary depending on the data itself and the goal of the project. However, commonly required steps include removal of fragments, such as salts and inorganic or organometallic entities that may pose a challenge; normalization of specific chemotypes (for instance, tautomers whose difference includes a (1,3)-shift of H atoms between heteroatoms, movable charges, or ion-pair representations) to ensure that different ways of writing the same structure will result in the same representation of the compound (Martin, 2009; O’Boyle, 2012; Sitzmann, Ihlenfeldt, & Nicklaus, 2010); curation of tautomeric forms that may or may not result in redundancy, and the removal of duplicates. The removal of duplicates and compounds with ambiguous assay outcomes is vital but tricky. For example, descriptors calculated from 2D representations of any pair of enantiomers or diastereoisomers using chemical graphs will likely yield duplicates (Alexander Tropsha, 2010). In such cases, descriptors that take chirality into consideration should be employed (O'Boyle, 2012), or only one of the isomers should be included in the library. These protocols can be achieved with a number of different tools, including: (1) free-for-academic-use software such as JChem from ChemAxon ("ChemAxon," n.d.) and OpenEye ("OpenEye," n.d.); (2) publicly available standalone tools like OpenBabel (O'Boyle et al., 2011), RDKit (Greg, n.d.), Indigo ("Indigo Toolkit," n.d.), and Chemistry Development Kit (Willighagen et al., 2017); or (3) as modules in KNIME ("KNIME," n.d.) (a data mining platform with graphical user interface).

1.2.3 Feature generation

Features (descriptors and fingerprints) play a crucial role in the successful development of toxicity prediction models (Kruhlak et al., 2012). They may be referred to as the chemical characteristic of a compound encoded in numerical form, depending on the molecular representation and the algorithm used for calculation (Danishuddin & Khan, 2016). Broadly, descriptors are organized by their nature into the following groups: constitutional – molecular composition and general properties (atom/bond/ring count, molecular weight and atom type); topological – applies graph theory to the connections of atoms in the molecule (Zagreb and connectivity indices); geometric – a more computationally expensive set of descriptors requiring information that describes the relative positions/coordinates of the atoms in 3D space (3D-MoRSE, WHIM, GETAWAY, EVA) (“Molecular Descriptors,” 2007; Todeschini, Consonni, & Wiley InterScience (Online service), 2000), while also offering more discriminative power than topological descriptors; and physiochemical – the physical and chemical properties of the 2D structure of the molecule (partition coefficient, lipophilicity, solubility, and permeability). Other descriptor types include quantum mechanical/electronic descriptors (Danishuddin & Khan, 2016; Lo, Rensi, Torng, & Altman, 2018).

Fingerprints are a particularly complex form of descriptors containing a fixed number of bits, with each bit representing the presence (1) or the absence (0) of a feature, either on its own or in conjunction with other bits in the bit string (Lo et al., 2018). The fingerprints most widely used for toxicity prediction modeling and similarity searching include the Extended-Connectivity FingerPrints (Rogers & Hahn, 2010) (ECFP), MACCS (“OpenEye,” n.d.) and PubChem (Health, n.d.) fingerprints. ECFPs are circular topological fingerprints whose bits are not predefined, so they can represent an infinite number of structural variation. They have been successfully employed in a number of toxicity prediction studies. For example, ECFPs were wildly employed in both the DREAM (Eduati et al., 2015) and Tox21 (Mayr et al., 2016) challenges to predict the toxic effect of compounds. The MACCS fingerprint is a 166-bit structural key descriptor in which each bit is associated with a specific structural pattern. A structural key is a fixed-length bit string in which each bit is associated with a specific molecular pattern. The PubChem fingerprint encodes 881 bits for properties of substructures, such as type and count of rings, element count, and atom pairs. The PubChem database (Y. Wang et al., 2009) employs this fingerprint for similarity neighboring and searching. Danishuddin (Danishuddin & Khan, 2016) provides a detailed review of descriptors. The choice of descriptors often depends on the properties of the molecules in the library as well as the target of the prediction exercise. Duan et al. (Duan, Dixon, Lowrie, & Sherman, 2010) compared the performance of eight molecular descriptors and reported that most of the fingerprints resulted in similar retrieval rates. However, hybrid fingerprints averaged over all of the molecules led to higher performance. Open source tools like RDKit (Greg, n.d.), Chemistry Development Kit (CDK) (Willighagen et al., 2017) and PaDEL (Yap, 2011) (a graphical tool based on CDK) have been widely employed for feature generation.

1.2.4 Feature selection and extraction

In QSAR modeling, the relationship between molecules and their toxicity profile or other biological activity is established via molecular descriptors. With the large number of available descriptors (Danishuddin & Khan, 2016), datasets often suffer from the "curse of dimensionality" (problems caused by performing predictions in a very large feature space) and the so-called "large p, small n" problem (where *p* is the number of descriptors and *n* is the number of molecules). In other words, models trained on a very small set of molecules that are described with a very large set of descriptors tend to be prone to overfitting (Perez-Riverol, Kuhn, Vizcaíno, Hitz, & Audain, 2017). An overfitted model can mistake small fluctuations for important variance in the data, which can result in significant prediction errors. Identifying reliable descriptors for establishing this relationship can pose a serious challenge. Models with fewer descriptors are easier to interpret, less computationally expensive, higher performing for new molecules, and less prone to overfitting/overtraining (Eklund, Norinder, Boyer, & Carlsson, 2014; Perez-Riverol et al., 2017). The task of selecting relevant descriptors that encode the maximum amount of information about the molecules with minimal collinearity is crucial to obtaining a high performing model (Goodarzi, Dejaegher, & Heyden, 2012). Two techniques employed in reducing the number of features include feature selection and feature extraction.

Feature selection involves picking a subset of features by eliminating irrelevant and redundant descriptors, yielding the best possible performance based on a selection criterion. The process does not alter the original representation of the descriptors, thus maintaining the physical meanings and allowing for interpretability. Feature selection techniques can be classified into filter, wrapper, and embedded methods (Danishuddin & Khan, 2016; Kohavi & John, 1997; Tang, Alelyani, & Liu, n.d.). Filters work without taking the classifier into consideration. They rely on measures of the general characteristics of the training data, such as distance, consistency, dependency, information, and correlation. By doing so, the bias of a classifier does not interact with the bias of a feature selection technique (Tang et al., n.d.). Information gain (S. Lei, 2012), correlation coefficient (Kwasnicka, Michalak, Kwa´snicka, & Kwa´snicka, 2006), variance thresholding (S. Lei, 2012), and chi squared (Héberger & Rajkó, 2002) are among the most representative algorithms of the filter model. Filters are the least computationally intensive of the feature selection methods, but they may ignore the effects of the selected feature subset on the performance of classifier.

Wrapper methods use the performance (accuracy) of a learning algorithm to determine the relevance of a selected subset of features. The feature subset with the best predictive performance is selected to train the classifiers (Tang et al., n.d.). Unlike filters, this allows wrapper methods to detect feature dependencies. However, wrapper methods are computationally inefficient for very large feature sets, considering the search space for *p* features is $O(2^p)$ (Tang et al., n.d.). Wrapper methods could either be deterministic (Sequential Forward Selection [SFS] and Sequential Backward Elimination [SBE]) or randomized (genetic algorithms and simulated annealing) (Kohavi & John, 1997).

Embedded methods were designed to alleviate the challenges posed by filter and wrapper methods. Thus, the embedded model usually achieves both comparable accuracy to the wrapper model and comparable efficiency to the filter model. Recursive Feature Elimination (RFE), an embedded method, starts with all of the features, generates the importance of each feature, and then prunes the least important features. This process continues until the desired accuracy value or number of most relevant features is obtained. Decision trees and random forests are very common embedded methods with built-in ID3 and C4.5 algorithms for feature selection (Tang et al., n.d.). Hybrid algorithms that utilize a combination of feature selection techniques benefit from the various advantages of their constituent methods. These algorithms have also been applied to develop models for HIV1 protease inhibitors (Rao et al., 2009; Zeng, Zhang, Zhang, & Zhang, 2014) and for selection of relevant cancer genes (Y. X. Liu, Zhang, He, & Lun, 2015). Another example is the kNN model-based feature selection method (kNNMFS), which was introduced by Guo as a feature selection method for toxicity prediction (Y. X. Liu et al., 2015).

Feature extraction approaches project the initial feature set into a new feature space with lower dimensionality, and the new constructed features are, in many cases, a transformation of original features. Therefore, it is difficult to tie the new features to the original ones, and further analysis of the transformed features may be challenging.

Feature extraction methods could be linear, such as Principle Component Analysis (PCA) (Ringnér, 2008) and Linear Discriminant Analysis (LDA) (Dorfer, Kelz, & Widmer, 2015); or non-linear, such as kernel PCA (Reverter, Vegas, & Oller, 2014; Schölkopf & Smola, 2001), Independent Component Analysis (ICA) (Hyvärinen, Hyvärinen, & Oja, 2000), neural algorithms (like Self Organizing Maps [SOM]) (Kohonen, 1982), and autoencoders (Baldi, 2012). PCA has been employed in several attempts at selecting the best descriptors (Ling Xue, Jeff Godden, Hua Gao, & Bajorath, 1999; Xue & Bajorath, 2000), and for modeling the oral LD50 toxicity of chemicals on rats and mice (Bhhatarai & Gramatica, 2011).

1.3 Model development

In terms of developing Machine Learning (ML) techniques for toxicity prediction, the aim is to create models/functions that can extract the underlying patterns and information encoded in molecular descriptors in order to predict the toxicity profile of new compounds. Several ML algorithms have been used to infer the relationship between molecular descriptors and toxicity, including Logistic Regression, Multiple Linear Regression, Naïve Bayes, K-Nearest Neighbors, Decision Trees, Support Vector Machines, and Neural Networks. This section discusses some commonly used techniques that can be applied to both qualitative classification and quantitative regression tasks.

1.3.1 Support vector machines

Support Vector Machines (SVM) were introduced by Vapnik et al (Cortes & Vapnik, 1995) as a supervised machine-learning algorithm to handle datasets with high-dimensional variables. In the context of toxicity prediction, the algorithm uses kernel functions, such as linear, polynomial, sigmoid, and radial basis (RBF) to project molecules encoded by descriptor vectors into a space that maximizes the margin (or separation boundary) between different classes. The goal is to make the classes linearly separable. After the training process is complete, the features in the projected space are separated by a hyperplane that delineates the difference between active and inactive molecules. The choice of the kernel used to achieve this is mostly dependent on empirical and experimental analysis. Different optimization parameters are used to find a hyperplane that maximizes the margin between the classes, ensuring that molecules in each class are as far away from those of the other class as possible. The key assumption is that the larger the margin between the classes, the higher the probability of the model to correctly classify new molecules that it was not exposed to during training. Points that lie on (or relatively close to) the hyperplane are referred to as support vectors, as shown in Figure 1.2.

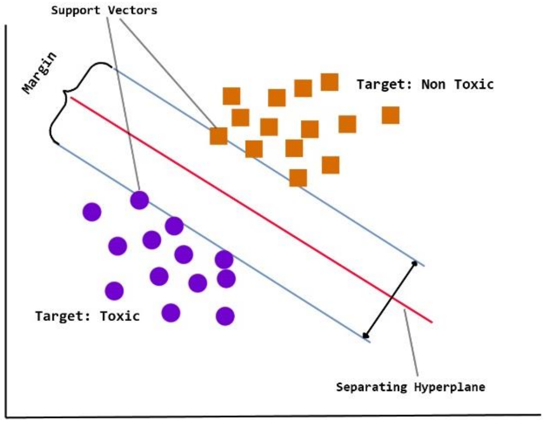


Figure 1.2 Support vector machine showing the optimal separating hyperplane and support vectors, i.e. points that are the closest to the separating hyperplane.

1.3.2 Random forests

Random forest (RF) (Breiman, 2001) is an ensemble modeling approach that operates by constructing multiple Decision Trees (DTs) as base learners. A DT is commonly depicted as a tree with its root at the top and its leaves at the bottom. Beginning from the root, the tree splits into two or more branches/edges. Each branch splits into two or more branches, and this continues until a leaf/decision is reached. The split of a branch is referred to as an internal node of the tree. The root and leaves are also referred to as nodes, and the link between nodes represents a decision (rule). For toxicity prediction, each leaf at the end of the tree is labeled by a class (Active or Inactive), while all internal nodes and the root are assigned a molecular descriptor. DTs tend to grow in an unrestrained manner and also tend to overfit. To handle such growth, pruning is employed. Pruning involves removing the branches that make use of molecular descriptors that have low importance. Thus, the complexity of tree as well as its ability to overfit is reduced. Some of the most commonly used decision tree algorithms include ID3, C4.5 and CART (Singh & Gupta, 2014). These algorithms use either information gain, gain ratio, or gini index respectively for deciding which variable to use for splitting a node.

RF is an ensemble classifier made up of many DTs. The fundamental idea behind training a random forest model to perform toxicity prediction is to combine many DTs developed using a subset of the molecular descriptors and data points (molecules) of the training set. This subset is randomly sampled with replacement. Usually, about two-thirds of the data is used for training the DT, and what is left is used for evaluating the tree. This random sampling lends the name "random forest", and it is also responsible for an increase in the diversity of the DTs that make up the forest. The result is more generalizable predictions. To predict the toxicity of a new molecule, the trained RF model takes an average/vote of all the DTs in the forest. RF models offer a number of advantages over individual DTs. They implicitly perform feature selection, and they are not affected by nonlinear relationships between variables. Furthermore, they are also less prone to overfitting and are better for handling the problem of imbalanced classes.

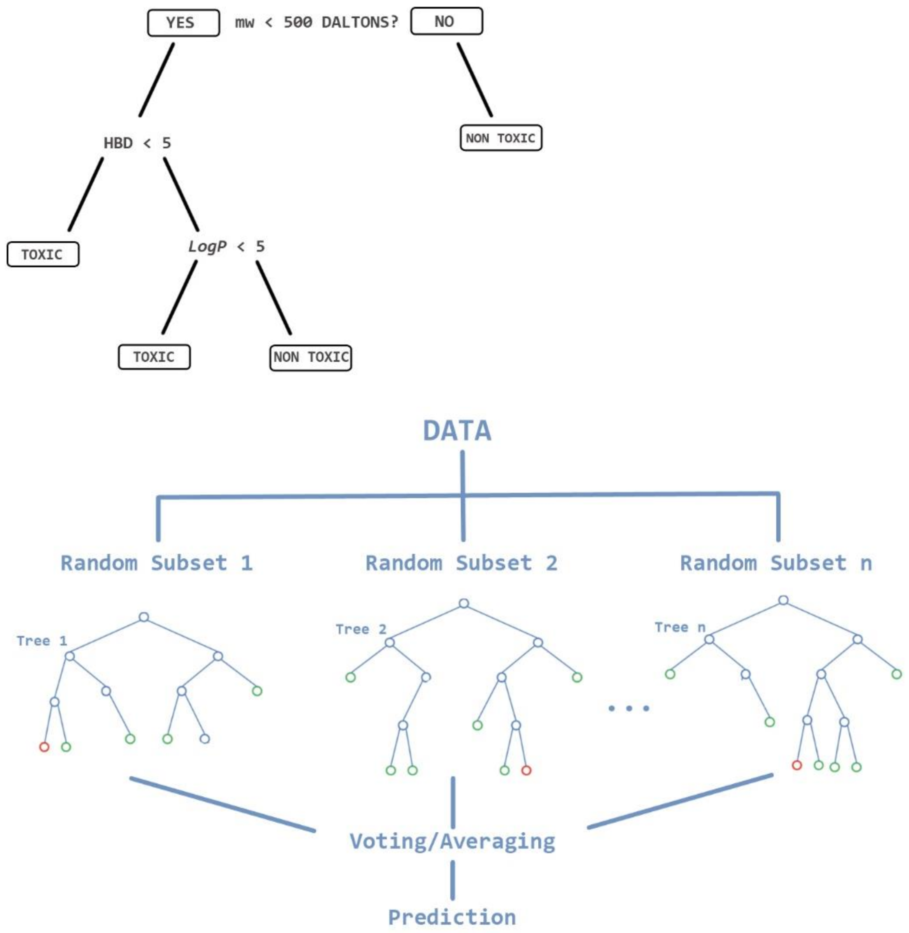


Figure 1.3 (a) A decision tree showing the path from the root to the leaves (b) An ensemble of decision trees that forms a random forest.

1.3.3 Neural networks and deep learning

Inspired by the structure of neurons, Neural Networks (NNs) were developed to emulate the learning ability of biological neural systems. The basic architecture of a NN consists of several processing units combined as layers, with consecutive layers being connected by means of weights (W). In NN models trained for toxicity prediction, this parallel computational structure maps molecular descriptors (input variables) in the input layer to the toxicity endpoints in the last/output layer via an intermediate set of hidden layers. Hidden layers each receive the data modified by the previous layer in the sequence.

Deep neural networks, or simply named as deep learning (DL), which describes a family of NNs with multiple hidden layers, has become very popular as it has demonstrated enormous success in different tasks from multiple fields. For example, DL has revolutionized the fields of computer vision, text and speech analysis (LeCun, Bengio, Hinton, et al., 2015). DL models may be considered artificial neural networks, i.e. directed acyclic graphs consisting of multiple hidden layer layers with neurons that can process multiple levels of data abstraction to learn structural patterns. It has also been shown to minimize the need for feature engineering even in high dimensional cases (LeCun, Bengio, Hinton, et al., 2015).

Neurons in different layers are connected by weights (W), and each neuron is associated with an activation function (σ). The input into any neuron is computed as a non-linear transformation of the weighted sum of the outputs from the previous layer. This is given as:

$$y = \sigma(\sum W^T x)$$

It employs the backpropagation algorithm to instruct the model on how to adjust the internal parameters (weights) between its hidden layers (Goodfellow, Ian; Bengio, Yoshua; Courville, 2016). To learn, a network computes and minimizes the error, i.e. the difference between the prediction in the output layer and the known endpoint, using an objective function. The error is propagated backwards using gradient descent by obtaining the derivative of error with respect to each weight and then adjusting the weights to minimize this error. For supervised learning, a trained model (or function) is one with a minimized difference between predicted output and known results.

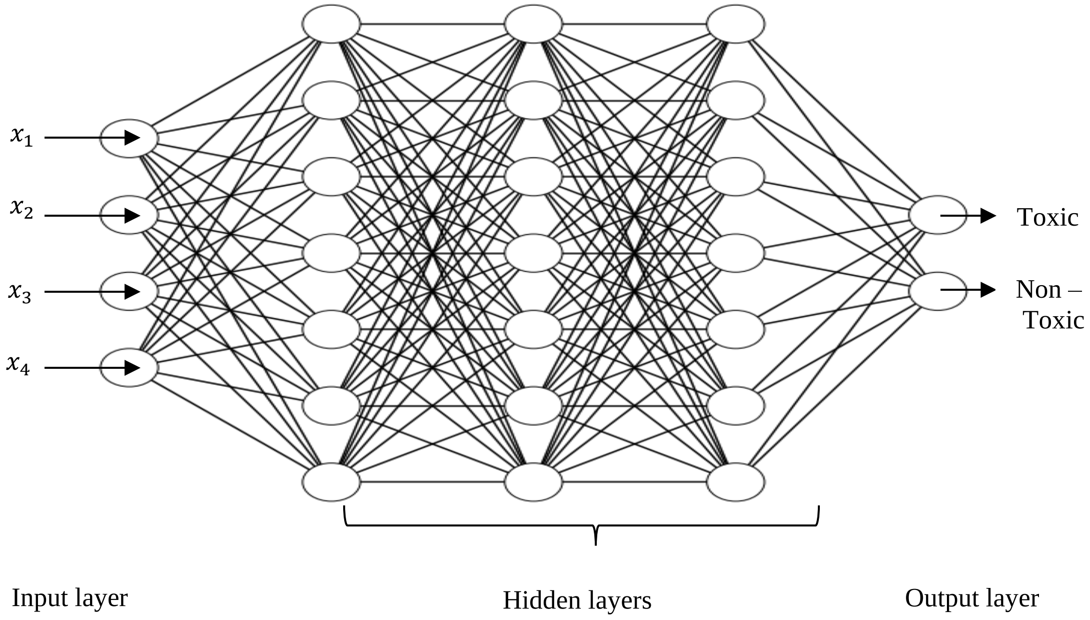


Figure 1.4 A fully connected deep neural network with three hidden layers, each with seven units. The input layer receives input data with features (*x_1, x_2, x_3, x_4*) and predicts outputs as two classes (either toxic or non-toxic).

Several variants of the DL algorithm have been employed for many tasks. Feedforward neural networks are among the most universal. Convolutional neural networks are best suited for data presented in multiple arrays, such as images and audio spectrograms. Recurrent Neural Networks (RNNs) have also been successful at tasks such as speech and language recognition as they require sequential inputs (LeCun, Bengio, Hinton, et al., 2015). The ability of RNNs to retain a state that can represent information from an arbitrarily long context window differentiates them from other DL architectures and makes them excellent candidates for tasks where the sequence of information is important. Self-Organizing Maps (SOMs) and Autoencoders are unsupervised DL algorithms. Autoencoders learn to create a representation of the input data by reproducing it in the output layer. They are useful for dimensionality reduction. Excellent reviews of deep learning in drug discovery (H. Chen, Engkvist, Wang, Olivecrona, & Blaschke, 2018; Gawehn, Hiss, & Schneider, 2016) and computational chemistry (Goh, Hodas, & Vishnu, 2017) have been published. Deep learning models have been employed in toxicity prediction (Mayr et al., 2016), multitask bioactivity prediction (Dahl, Jaitly, & Salakhutdinov, 2014; Ramsundar et al., 2015; Yuting Xu, Ma, Liaw, Sheridan, & Svetnik, 2017), and chemical reaction prediction (Fooshee et al., 2018).

More recently, Generative Adversarial Networks (GANs) (Goodfellow et al., n.d.), unlike the discriminative algorithms described above, have gained prominence. Given labels as y and features as x, discriminative algorithms will learn $p(y|x)$, whereas GANs are best for $p(x|y)$. In GANs, two differentiable functions represented by neural networks are pitted against each other to generate a data distribution similar to the input. Kadurin *et al* (Kadurin, Nikolenko, Khrabrov, Aliper, & Zhavoronkov, 2017)developed druGAN, a model to design new molecules de novo with desired properties. Similar successful efforts have been reported in literature (Barril, 2017; Putin et al., 2018; Schneider, 2018).

1.3.4 Common Pitfalls to developing SAR models with high predictive accuracy

1.3.4.1 Handling Imbalance

The problem of imbalanced datasets is particularly crucial in QSAR modeling, where the number of active compounds is far outweighed by the number of inactive compounds. The active class is often of more interest to the researcher; however, models tend to be more biased towards the majority inactive class. This challenge needs to be resolved before performing modeling that relies on data centric or algorithmic methods. Data centric methods involve resampling, either by oversampling the active minority class or by undersampling the inactive majority class (N. V. Chawla, Bowyer, Hall, & Kegelmeyer, 2002). While these methods have been used successfully in many cases, they have some drawbacks. Chawla (N. V. Chawla et al., 2002) reported that oversampling will easily cause overfitting, and undersampling may discard useful data that leads to information loss. Hence, the proposal of the oversampling technique called Synthetic Minority Oversampling Technique (SMOTE). Sun et al. (Bhhatarai & Gramatica, 2011) modeled the Cytochrome P450 profiles of environmental chemicals using undersampling and oversampling techniques. Sampling methods were also used by Chen et al. (J. Chen, Tang, Fang, & Guo, 2012) to predict the toxic action mechanism of phenols. It was reported that undersampling performed more consistently than oversampling.

Algorithmic methods of handling imbalance involve building cost-sensitive learners that assign a higher cost to misclassification of the active minority class samples, and the use of ensemble classifiers (H. He & Ma, 2013). Ensemble learners coupled with resampling include UnderBagging(Barandela, Sánchez, & Valdovinos, 2003), SMOTEBagging (S. Wang & Yao, 2009), SMOTEBoost (Nitesh V. Chawla, Lazarevic, Hall, & Bowyer, 2003), and EUSBoost(Galar, Fernández, Barrenechea, & Herrera, 2013) (which is considered an improvement over RUSBoost) (Seiffert, Khoshgoftaar, Van Hulse, & Napolitano, 2010).

When learning from imbalanced data, it is important to use the right metric for evaluation as highlighted in section 4. Metrics such as Accuracy and even AUROC tend to be rather optimistic (Lobo, Jiménez-Valverde, & Real, 2008) when dealing with imbalanced data. AUPRC and other metrics such as Balanced Accuracy, Sensitivity, and Specificity appear to provide a better or at least complementary evaluation for imbalanced classifiers (Saito et al., 2015).

1.3.4.2 Activity Cliffs

Activity cliff is a term used for cases of structurally similar molecules that have markedly different activities against a particular target (Iyer, Stumpfe, Vogt, Bajorath, & Maggiora, 2013). Visually, they are the sharp spikes noticed on activity landscapes – a 2D projection of the chemical space with the activity of molecules as the third dimension (Bajorath, n.d.), resembling a topography map. Measures such as the Structure–Activity Landscape Index (SALI) (and & John H. Van Drie*, 2008) and the SAR Index (SARI) (Peltason & Bajorath, 2007) have been used to identify and estimate activity cliffs/data discontinuity.

The underlying assumption for SAR modeling is that molecules with similar structure will have similar activity. This lends to further assumption that the relationship between structure and activity is continuous; this is important for successful predictive modeling. Activity cliffs create discontinuous structure-activity relationship which may be detrimental to machine learning models, even those that are capable of learning nonlinear relationships (Cruz-Monteagudo et al., 2014). Activity cliffs often represent the contradictions in a dataset, hence can be detrimental to predictive modeling. Guha (Guha, 2011) reported that a model forced to learn from a dataset with a lot of activity cliff is prone to overfitting.

Maggiora (Maggiora, 2006) posits that for SAR models to be successful, the structure–activity landscape looks like gently rolling hills, whereas most landscapes are seen to be heterogenous, having spikes, gentle slopes and smooth regions. In dealing with such heterogeneity, Cruz-Monteagudo et al (Cruz-Monteagudo et al., 2014) suggested the development of a consensus/ensemble learner as each base learner should cover a different region of the chemical space. They also suggested the removal of activity cliff generators to ensure structure-activity relationship continuity but warned that a trade-off ensues as the activity domain of the new dataset will shrink (Cruz-Monteagudo et al., 2014). It is still unclear as to what extent, if any, a modeling process is affected because of the information lost due to the removal of activity cliff generators. Remediation of activity cliffs remain an active research area.

1.3.4.3 Generating Relevant Molecular features

Molecular descriptors, being numerical features extracted from molecular structures, are the most common variables used for SAR-based toxicity prediction modeling (H. Yang, Sun, Li, Liu, & Tang, 2018). The information encoded by descriptors depends on the molecular representation or “dimensionality” of the compound as well as the algorithm used to calculate the descriptors (Danishuddin & Khan, 2016). One dimensional (1D) descriptors are scalars encoding physiochemical properties (molecular weight, *logP*) and constitutional parameters, such as number of atoms, bond count, atom type, ring count, and fragment counts. 1D descriptors are insensitive to the topology of the molecule and tend to be similar for distinct compounds. As a result, they are often used in combination with other descriptors. Two-dimensional (2D) descriptors are more frequently used for chemical space description. 2D descriptors, including topological indices and structural fragments, are calculated from the connection table (chemical graph) representation of a molecule. They are not only independent of the conformation of the molecule but also graph invariant (not sensitive to altering the number of graph nodes). Three-dimensional (3D) descriptors provide a more complete characterization of molecular structures. 3D descriptors require conformational searching and can discriminate between isomers; this comes at the price of being computationally expensive. The ability to discriminate between isomers can translate to less redundant features. Examples of 3D descriptors include geometric, electrostatic, quantum-chemical, and WHIM & GETAWAY. Four-dimensional (4D) descriptors are much like 3D descriptors that evaluate multiple structural conformations simultaneously. Fingerprints are another form of molecular descriptors (Danishuddin & Khan, 2016; "Molecular Descriptors," 2007; Todeschini et al., 2000). Commonly used fingerprints include the Molecular ACCess System (MACCS) (Duan et al., 2010) substructure fingerprints, PubChem (Health, n.d.), and Extended Connectivity FingerPrints (ECFP) (Rogers & Hahn, 2010). These fingerprints and 2D descriptors were widely used in the Tox21 Data Challenge (R. Huang, Xia, Nguyen, et al., 2016) where the winning submissions used over 2500 predefined features covering a wide range of data from topological and physical properties to fingerprints (Mayr et al., 2016).

As shown above, the chemical structures used in QSAR modeling are characterized by many molecular descriptors. It is common to generate thousands of descriptors for a single molecule (Mayr et al., 2016). It is well known that the accuracy of predictive models is not positively correlated to the dimensionality of the data, as overfitting tends to become an issue (Clarke et al., 2008; Hinton & Salakhutdinov, 2006; Subramanian & Simon, 2013) but correlated to the amount of relevant information the descriptors encodes. High dimensional spaces are prone to include irrelevant and noisy features (Ang, Mirzal, Haron, & Hamed, 2016). SARs developed using such features tend to focus on the peculiarities of molecules and fail to be generalizable (Merkwirth et al., 2004). In the chemical space for a given library, each descriptor adds a dimension to the n-dimensional chemical space. Every molecule in the library is assigned a coordinate depending on its values for all the descriptors. A reduction in the dimensionality of the chemical space correlates with an increasing similarity between molecules. This is important because the underlying assumption in SAR modeling posits that molecules with similar structures should have similar activity (Bajorath, 2001; Venkatraman, Dalby, & Yang, 2004). Thus, one of the most important tasks prior to modeling is generation of features focused on encoding the most important and relevant information required for predicting the desired biological activity such as toxicity endpoint. Shen *et al*. (R. Huang, Xia, Nguyen, et al., 2016) demonstrated the usefulness of feature selection for toxicity prediction, particularly for interpreting the role of the features. In summary, a predictive model is only as good as the features it receives.

1.4 Estimation of model reliability

1.4.1 Model evaluation

The most common metrics for evaluating QSAR models, particularly binary models, are calculated based on the values of the confusion matrix. These values are true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The value of sensitivity reflects the model's ability to correctly predict positive samples (active), whereas the value of specificity represents the model's ability to correctly predict negative samples (inactive). Accuracy (ACC) estimates the overall predictive power of the model. However, this is only useful for models trained on data sets whose samples are relatively balanced across the classes. More often than not, QSAR models are very like to be highly imbalanced as result of the rarity of active compounds in comparison to inactive compounds reported from high throughput screenings.

Sensitivity = $\frac{TP}{TP+FN}$

Specificity = $\frac{TN}{TP+FN}$

Balanced Accuracy = $\frac{Sensitivity+Sensitivity}{2}$

Other metrics such as balanced accuracy and Mathew's Correlation Coefficient (MCC) become relevant. Model-wide evaluation metrics like Area Under the Receiver Operating Characteristic curve (AUROC) and Area Under the Precision Recall Curve (AUPRC) can also be applied to imbalanced cases of binary and multi-class/multi-target models. Saito (Saito et al., 2015) reported that AUROC is an overly optimistic metric for imbalanced binary learning, hence AUPRC is likely to present better a view of the model's performance. Frequently used metrics for regression models are root mean squared error (RMSE), coefficient of determination (R^2), and mean absolute error (MAE).

Where $\hat{y}$ is the predicted value, y is the observed value and $\bar{y}$ is the mean of observed values:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|$$

$$R^2 = 1 - \frac{\left(\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right)}{\left(\frac{1}{n}\sum_{i=1}^{n}(\bar{y}_i - y_i)^2\right)}$$

1.4.2 Model validation

Validation of in silico toxicity models is a very important step in the process of understanding the reliability of models when making predictions for new molecules that are not present in the training data set. The regulatory decisions and justification for using any toxicity prediction model are dependent on the model’s ability to make predictions for new molecules with some known degree of certainty (Raies & Bajic, 2016). Therefore, the validation of models is of utmost importance. While formal validation has been overlooked in the past, more emphasis is currently being placed on it as an important step that should involve statistical assessment, interpretability, and a definition of the model’s applicability domain. Model validation could be either internal (using the training set) or external (using a separate unseen data set). Internal validation methods include cross validation, Y-Randomization, and bootstrapping (Lavecchia, 2015). The external method involves using statistical assessments (metrics) to evaluate model performance on a separate test data set. There are varying schools of thought as to which validation technique is best, but external validation appears to be favored. When using an external test set for validation, care must be taken to ensure that the training and test set both exist within the same chemical space. Roy et al. (Roy, Kar, Das, et al., 2015) and

Tropsha et al. (Alexander Tropsha, 2010; Alexander Tropsha, Gramatica, & Gombar, 2003) suggested that a properly validated model is considered reliable and should have high predictive power if it is validated by making predictions on an external test set containing molecules that the model is blind to (assuming the domain of practical application for that model is defined).

1.4.3 Applicability domain

Regardless of how generalized a model may appear to be following validation, it is impractical to consider the model applicable to the entire chemical space. The predictions made by models on new compounds with descriptor values outside the training data descriptor (feature) space may not be reliable. It is therefore necessary to know the boundary within which the model can extrapolate reliably. The applicability domain (AD) defines the scope and limitations of a model. AD attempts to define the degree of generalization of the model by highlighting the range of chemical structures for which the model is considered to be reliably applicable (Netzeva et al., 2005; Sahigara et al., 2012). Predictions of compounds outside a model's AD cannot be considered reliable.

The AD is defined using the training set, hence it is advised that the training set should cover the entire chemical space of the molecules in the total project library. Depending on the method used for interpolation space characterization, determination of AD using descriptors are generally achieved via range-based methods (Jaworska, Nikolova-Jeliazkova, & Aldenberg, 2005), geometric methods (Dimitrov et al., 2005), distance-based methods, probability density distribution-based methods (Netzeva et al., 2005), KNN and Decision Trees methods (Roy, Kar, & Ambure, 2015; Tong, Hong, Fang, Xie, & Perkins, 2003). A fifth class of methods called range of the response variable is based on the response space of the training set molecules (Roy, Kar, & Ambure, 2015). Hanser et al. (Hanser, Barber, Marchaland, & Werner, 2016) noted that results derived with different AD approaches may vary for the same dataset, and none of these approaches can be considered sufficient enough to be applied to all the cases. There are several ongoing attempts in the chemoinformatics community at developing new approaches for estimating an acceptable AD for models.

Distance-based methods use distance measures (e.g. Tanimoto or Euclidean) to calculate the distance between a new compound and its k-nearest neighbors (or the centroid of the training set). A threshold, based on distance, is used to determine if the new compound is within the AD or not. Predictions of any compound beyond the threshold are considered to be unreliable. The downside of this method is that the threshold value is often arbitrary (Sahigara et al., 2012). Using the Enalos module, KNIME provides a graphical user interface to generate the AD domain based on Euclidean Distance and Leverage. One other type of non-descriptor method is the structural fragment-based method, which requires that all structural fragments in the new molecule be present in the training set (Hewitt & Ellison, 2010).

In sum, the application of machine learning in predicting the toxicity profile of chemicals has been well documented. An increase in the access to data and computing power have contributed to the use of *in silico* methods for toxicity prediction. Large amounts of heterogeneous and high-dimensional data sets as shown in **Table 1** are available, in addition to easily accessible open source tools for data preprocessing and predictive modeling. ToxCast and Tox21 (R. Kavlock & Dix, 2010) are representative efforts by regulatory institutions at employing machine learning for toxicity prediction.

The success rate of these efforts has been shown to improve over time. Notwithstanding, several challenges limiting the toxicity prediction accuracy and reliability of SAR models remain.

Most machine learning models are ‘black boxes’ as rational interpretation of underlying mechanisms are difficult. Even models with high accuracy do not readily unearth the biological mechanisms behind such predictions (H. Chen et al., 2018; Gawehn et al., 2016). For instance, neural networks were the most successful algorithm in the Tox21 challenge (Mayr et al., 2016). This success did not provide an explanation on which substructures may have been responsible for specific toxicity predictions. Such information is useful to a toxicologist and medicinal chemist for lead optimization.

The quality of data used to train a model is considered more important than the choice of algorithm used. Fourches et al (Fourches et al., 2016) designed a workflow that can aid reproducibility of the data cleaning process. However, the process of cleaning and standardizing compounds prior to feature generation remain unclear and unreproducible in many published works. Details of the data curation process should be well documented. Molecular descriptors play an integral role in modeling the relationship between structure and activity. The choice of descriptors and the selection/extraction methods employed to keep only useful explanatory features for modeling was discussed. Although thousands of molecular descriptors exist, there is room to develop more informative and explanatory descriptors for molecules. Several methods, each with its advantages and disadvantages, have been proposed for dealing with imbalanced data. Such methods can prevent the development of biased or over-trained models. The definition and how to deal with activity landscapes remain an active research area, and no definitive work has been reported on the effect of removing activity cliff generators. One solution could be the use of ensemble learning methods to account for different regions of the chemical space being modeled (Cruz-Monteagudo et al., 2014).

Validation is a particularly important component of developing reliable SAR models. The validation of models and the definition of its applicability domain are overlooked too often. Tropsha et al (Alexander Tropsha et al., 2003) suggested that models are to be validated using external training sets and applicability domain will help define the chemical space within which the model may be considered reliable.

Overall, as more data from high throughput screening become available and new computational approaches and resources are made available, machine learning will continue to play a pivotal role in understanding the toxicity profile of many untested compounds.

1.5 Approaches to SAR modeling Pitfalls

In the following chapters, solutions are proffered to the challenges highlighted in the earlier sections. In *Chapter III*, synthetic minority over-sampling technique (SMOTE) and Edited Nearest Neighbor (ENN) are combined with bootstrap sampling to overcome the challenge of data imbalance. Using the right machine learning algorithm with the appropriate level of complexity is critical to achieving a high performing SAR model. In *Chapter IV*, deep learning and random forest were employed as algorithms with varying complexities to evaluate the bioactivity of small molecules against the androgen receptor. This is part of the model development stage in Figure 1.1. Lastly, new descriptors (features) were developed using multi-head self-attention translation models in *Chapter V*. These new features were designed to be of variable dimension and to encode more useful information than conventional features. These approached fit into the process for developing a SAR model as shown in Figure 1.1.

CHAPTER II - A REVIEW OF FEATURE REDUCTION METHODS FOR SAR-BASED TOXICITY PREDICTION

2.1 Introduction

The limitations of *in vivo* and *in vitro* approaches for determination of the biological activity of chemicals have fostered the development of *in silico* approaches (Lavecchia, 2015). *In silico* predictive toxicology is designed to complement experimental efforts with a view toward improving the quality of toxicity predictions for safety assessment while decreasing the associated time, cost and ethical conflicts (animal testing) (Greene & Pennie, 2015; Kruhlak et al., 2012; Raies & Bajic, 2016). Methodology for *in silico* predictive toxicology has been dominated by (Quantitative) Structure-Activity or Toxicity Relationship [(Q)SAR or (Q)STR] (hereafter called SAR). Traditional SAR models describe a relationship between the chemical structure of molecules (numerically encoded as molecular descriptors) and their activity against a specific biological target (Lavecchia, 2015). This is achieved by establishing a trend in the molecular descriptor space that links to a biological activity. Thus, all SAR models are developed on the assumption of a similarity principle. That is, molecules with similar structures (and descriptors, consequently) will have similar biological activity (Kruhlak et al., 2012; Alexander Tropsha, n.d.). A SAR model to predict toxicity (*T*) is given in equation:

$$T = g\left(D_f\right)$$

where $\left(D_f\right)$ represents the feature space of molecular descriptors as chemical properties, and g is a function that relates T to $\left(D_f\right)$ (Raies & Bajic, 2016). The accuracy of the model or function g has been shown to depend on the most representative set of molecular descriptors that will encode the useful properties of the molecules for prediction.

Molecular descriptors, being numerical features extracted from molecular structures, are the most common variables used for SAR-based toxicity prediction modeling (H. Yang et al., 2018). The information encoded by descriptors depends on the molecular representation or "dimensionality" of the compound as well as the algorithm used to calculate the descriptors (Danishuddin & Khan, 2016). One dimensional (1D) descriptors are scalars encoding physiochemical properties (molecular weight, *logP*) and constitutional parameters, such as number of atoms, bond count, atom type, ring count, and fragment counts. 1D descriptors are insensitive to the topology of the molecule and tend to be similar for distinct compounds. As a result, they are often used in combination with other descriptors. Two-dimensional (2D) descriptors are more frequently used for chemical space description. 2D descriptors, including topological indices and structural fragments, are calculated from the connection table (chemical graph) representation of a molecule. They are not only independent of the conformation of the molecule but also graph invariant (not sensitive to altering the number of graph nodes). Three-dimensional (3D) descriptors provide a more complete characterization of molecular structures. 3D descriptors require conformational searching and can discriminate between isomers; this comes at the price of being computationally expensive. The ability to discriminate between isomers can translate to less redundant features. Examples of 3D descriptors include geometric, electrostatic, quantum-chemical, and WHIM & GETAWAY. Four-dimensional (4D) descriptors are much like 3D descriptors that evaluate multiple structural conformations simultaneously. Fingerprints are another form of molecular descriptors (Danishuddin & Khan, 2016; “Molecular Descriptors,” 2007; Todeschini et al., 2000). Commonly used fingerprints include the Molecular ACCess System (MACCS) (Duan et al., 2010) substructure fingerprints, PubChem (Health, n.d.), and Extended Connectivity FingerPrints (ECFP) (Rogers & Hahn, 2010). These fingerprints and 2D descriptors were widely used in the Tox21 Data Challenge (R. Huang, Xia, Nguyen, et al., 2016) where the winning submissions used over 2500 predefined features covering a wide range of data from topological and physical properties to fingerprints (Mayr et al., 2016).

As shown above, the chemical structures used in SAR modeling are characterized by many molecular descriptors. It is common to generate thousands of descriptors for a single molecule (Mayr et al., 2016). It is well known that the accuracy of predictive models is not positively correlated to the dimensionality of the data, as overfitting tends to become an issue (Clarke et al., 2008; Hinton & Salakhutdinov, 2006; Subramanian & Simon, 2013). High dimensional spaces are prone to include irrelevant and noisy features (Ang et al., 2016). SARs developed using such features tend to focus on the peculiarities of molecules and fail to be generalizable (Merkwirth et al., 2004). In the chemical space for a given library, each descriptor adds a dimension to the n-dimensional chemical space. Every molecule in the library is assigned a coordinate depending on its values for all the descriptors. A reduction in the dimensionality of the chemical space correlates with an increasing similarity between molecules. This is important because the underlying assumption in SAR modeling posits that molecules with similar structures should have similar activity (Bajorath, 2001; Venkatraman et al., 2004). Thus, one of the most important tasks prior to modeling is dimension reduction focused on keeping the most important and relevant descriptors with the maximum amount of biologically meaningful information required for predicting the desired toxicity endpoint. Shen *et al*. (R. Huang, Xia, Nguyen, et al., 2016) demonstrated the usefulness of feature selection for toxicity prediction, particularly for interpreting the role of the features. By reducing the feature space, they were able to pinpoint *MolRef* and *AlogP* as the most important descriptors for predicting the toxicity of aromatic compounds.

In simple terms, dimensionality reduction is considered desirable for activity prediction modeling for the following reasons (Goodarzi et al., 2012):

i. Employing fewer descriptors means that the model can focus on important information for establishing a relationship, thus improving prediction accuracy and reducing overfitting (Models with many features enjoy more discriminating power during training but are often not generalizable).

ii. As the number of features decreases, interpretability of certain models increases.

iii. Computational costs reduce significantly as the complexity of many learning algorithms is greater than linear (Merkwirth et al., 2004; Shahlaei, 2013).

iv. Elimination of irrelevant descriptors can help remove activity cliffs (Danishuddin & Khan, 2016).

v. Machine learning algorithms are statistical in nature; hence, they suffer from the “curse of dimensionality”, which is common with optimization problems as described by Bellman (Bellman, n.d.).

As the dimensionality increases, the amount of data needed to develop generalizable models increases exponentially (Chandrashekar & Sahin, 2014; Van Der Maaten, Postma, & Van Den Herik, 2009). SAR data rarely have an abundance of labeled molecules and, as such, the final model and resulting toxicity prediction will benefit from a reduction in dimension as a smaller dimension means fewer samples will be required during training. The optimal subset of a feature space is one which has the least number of dimensions yet offers the best learning accuracy (Van Der Maaten et al., 2009). Two techniques used to alleviate the challenges of high dimension in SAR data sets include feature selection and feature extraction.

This chapter discusses different methods for both feature selection and feature extraction techniques, as well as their applications in SAR modeling. In the next two sections, feature selection and feature extraction methods are discussed consecutively. In the last section, important aspects that must be considered are highlighted while attempting feature space reduction, such as the stability and validation of the methods.

2.2 Feature Selection

Feature selection works by selecting a subset of features from the original feature set and removing irrelevant features without altering the original representation of the data, on the basis of certain relevance criteria [18, 26–28]. The physical meanings of the features are retained.

Mathematically, considering a descriptor space $X = \{x_i,\ i = 1 \dots n\}$, find a subset Y_k (with $k < n$) that maximizes an objective function $J(X)$ for the probability P that a compound is correctly predicted as active or inactive using equation below.

$$Y_k = \{x_{(1),}\, x_{(2),} \dots, x_{(k)}\} = argmax_{Y_k \subseteq X} \quad J(Y_k)$$

Thus, the ultimate goal of feature selection is to define a subset of Y_k relevant descriptors (obtained from an initial set of X descriptors) which holds the most useful molecular structure information for learning the underlying pattern present in the data.

One pronounced benefit of feature selection is that it can be used to avoid overfitting. Models with high dimension offer many degrees of freedom and tend to learn random patterns and noise instead of important underlying patterns between descriptors and the target endpoint (Johnstone & Titterington, 2009; X. Zhu & Wu, 2004). Many feature selection algorithms have been documented. Broadly, these algorithms can be grouped into the following three categories depending on the availability of class labels for the training set: supervised (Chandrashekar & Sahin, 2014; Goodarzi et al., 2012; Kohonen, 1982; Tang et al., n.d.), semi-supervised (Ang et al., 2016; Sheikhpour, Sarram, Gharaghani, & Chahooki, 2017) and unsupervised (Ang et al., 2016; Dy & Brodley, 2004). The choice of an appropriate method is dependent on the learning algorithm to be employed and the data to be used (Guyon & Elisseeff, 2003). The focus of this review is on supervised feature selection methods. Supervised feature selection requires that the entire training dataset be labeled. Feature selection is achieved by eliminating descriptors that have a low correlation with the toxicity endpoint to be predicted (Tang et al., n.d.). Feature selection methods applied to supervised tasks can be classified into filter, wrapper and embedded methods (Tang et al., n.d.). This section discusses each of these methods and further describes Hybrid (Hsu, Hsieh, & Lu, 2011; Solorio-Fernandez, Martinez-Trinidad, Carrasco-Ochoa, & Yan-Qing Zhang, 2012) and Ensemble (Ben Brahim & Limam, 2017; Guan, Yuan, Lee, Najeebullah, & Rasel, 2014;

Seijo-Pardo, Porto-Díaz, Bolón-Canedo, & Alonso-Betanzos, 2017) methods, which are a blend of the earlier listed methods. These methods are illustrated in Figure 2.1.

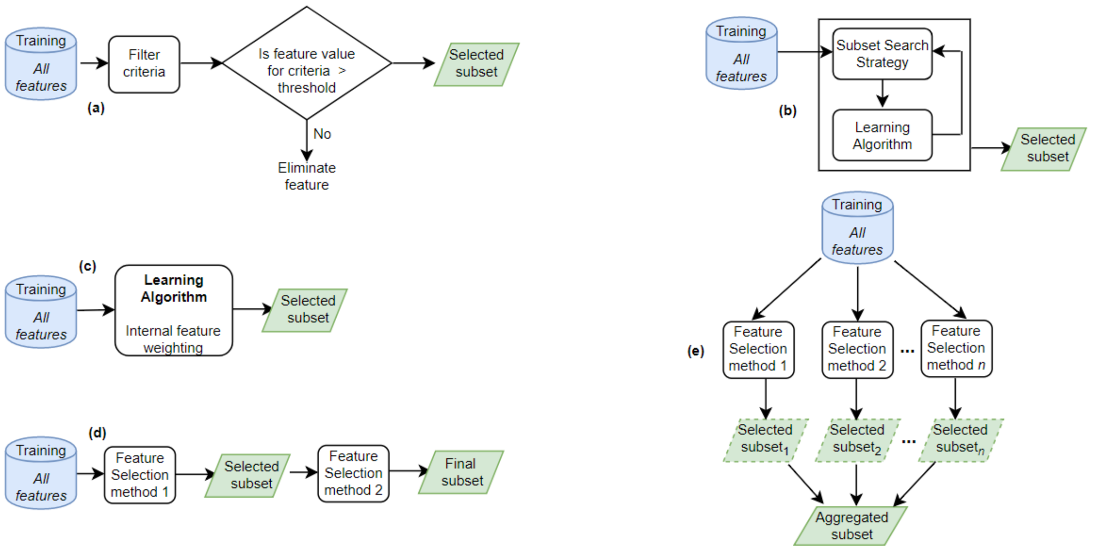


Figure 2.1 An illustration of different feature selection methods: (a) Filter (b) Wrapper (c) Embedded (d) Hybrid (e) Ensemble.

2.2.2 Filter

Filter methods evaluate the relevance of a feature based on its intrinsic properties and are completely independent of the learning algorithm (Ang et al., 2016; Cai et al., 2018; Janecek, Gansterer, Demel, & Ecker, n.d.; Tang et al., n.d.). The majority of filter methods are univariate, where each feature is considered independently of the feature space. Multivariate methods, such as correlation-based scores and paired *t-scores*, have also been used to assess the relevance of feature pairs and how well they synergize to enhance prediction of the desired endpoint (Hira & Gillies, 2015). Filter methods are computationally efficient and fast in comparison to wrapper methods. Their lack of dependence on any learning algorithm means that the features they select can be used with almost any learning algorithm. However, this independence often results in varied performance from these different learning algorithms (Tang et al., n.d.). Statistical methods make the assumption that the data they are applied on are normally distributed (Janecek et al., n.d.). By not taking the learning algorithm into consideration, filter methods also turn a blind eye to the heuristics and biases of these algorithms, which may impair their predictive abilities (Chandrashekar & Sahin, 2014).

Filter methods use feature ranking and filtering techniques as the basis for selection. Features are first evaluated and ranked based on a criterion. Then, a threshold is used to select all features above the mark that are considered to be relevant for predicting the endpoint (Ang et al., 2016; Hira & Gillies, 2015; Tang et al., n.d.), as shown in Fig. 2.1(a). The elimination of low-variance and highly correlated descriptors is a common filtering technique applied to SAR data sets (Mayr et al., 2016; Rajarshi & Jurs, 2004; Shahlaei, 2013). Several criteria have been employed for filtering descriptors, including variance score (Sheikhpour et al., 2017), correlation coefficient (Chandrashekar & Sahin, 2014; Guyon & Elisseeff, 2003), fisher (Guo, Neagu, & Cronin, 2005; Tang et al., n.d.), and information gain (Newby, Freitas, & Ghafourian, 2013).

2.2.3 Wrapper

Wrapper methods use learning algorithms to evaluate the relevance of a feature, where the learning algorithm's error rate or accuracy is treated as the objective function/criterion for evaluating a feature. A wrapper method begins by selecting a subset of the features heuristically or sequentially, and then a learning algorithm of choice is used to evaluate this subset. This process of subset generation and testing is repeated until the desired objective function is achieved (Cai et al., 2018; Tang et al., n.d.) (Fig. 7.1(b)). Wrappers tend to perform better than filters in selecting features since they consider feature dependencies and directly incorporate the specific biases and heuristics of the learning algorithm into the selection process. However, this implies that the selected features are unlikely to be optimal for any other classifiers (Ang et al., 2016).

The size of search space for *m* features is O(2^m) (Tang et al., n.d.). Since evaluating the subsets of such a search space is considered an NP-hard problem, the computational inefficiency of wrappers becomes evident when using larger datasets. However, search algorithms have been proposed for selecting optimal subsets of the feature space. Broadly, two groups of search strategies for wrappers are considered: Sequential and Heuristic Selection Algorithms (Chandrashekar & Sahin, 2014).

2.2.3.1 Sequential selection algorithms

Sequential selection can be achieved in two ways: forward selection and backward elimination. Sequential Forward Selection (SFS) begins with an empty set of features, and features are progressively incorporated into larger and larger subsets (one at a time) until no further improvement is recorded in the evaluation criterion. A backward elimination algorithm begins with the full set of features and iteratively eliminates the least relevant features (Tang et al., n.d.).

The Sequential Floating Forward Selection (SFFS) (Brendel, Zaccarelli, & Devillers, n.d.; Pudil, Novovičová, & Kittler, 1994) algorithm has been suggested as an improvement over SFS because it includes flexible backtracking capabilities. Similar to SFS, SFFS adds one feature at a time as determined by the objective function. Meanwhile, it backtracks by eliminating one feature at a time from the initial subset, followed by an evaluation. If an improvement is noticed in the objective function, it leaves that feature out and moves on to add a new feature. This process goes on iteratively until the desired goal is met with the fewest number of features.

2.2.3.2 Heuristic selection algorithms

Heuristic search algorithms evaluate different subsets to optimize the objective function. Subsets can be generated by evaluating a search space or by generating solutions to the optimization problem, with the learning algorithm's performance being the objective function (Chandrashekar & Sahin, 2014). Simulated Annealing (SA) (Kennedy, Eberhart, & gov, n.d.) and Genetic Algorithms (GA) (Goldberg & E., 1989), two widely used heuristic algorithms, find a subset of features for wrappers. A hybrid of these methods has also been suggested (Revathy, Revathy, & Balasubramanian, n.d.). In GA, the chromosome bits indicate if a feature should be included or not. SA, a stochastic algorithm, solves for the global minimum of a function by improving the initial solution repeatedly using small local perturbations until no such perturbations yield an improvement in the objective function. This process is randomized such that there are occasional and intentional deviations from the solution to lessen the probability of becoming stuck in a local optima. The use of GA to preselect descriptor subsets for SAR modeling of artificial and real data was shown to be successful in (R. Huang, Xia, Nguyen, et al., 2016) where 2D descriptors were employed to discriminate between active and inactive compounds. Particle Swarm Optimization (PSO) (Kennedy et al., n.d.) and Ant Colony Optimization (ACO) (Q. Shen, Jiang, Tao, Guo-li Shen, & Ru-Qin Yu, 2005) algorithms may also be employed for heuristic subset search. For instance, it has been shown that the ACO algorithm is a useful method for selecting descriptors for predicting Cyclooxygenase inhibitors (Q. Shen et al., 2005).

2.2.4 Embedded

Embedded feature selection methods incorporate feature selection into the model training process. Embedded feature learning, much like wrapper methods, takes the potential dependencies among features into consideration while being more computationally efficient and less prone to overfitting as compared to wrappers (Ang et al., 2016; Cai et al., 2018; Hira & Gillies, 2015; Tang et al., n.d.). A common embedded feature selection algorithm is random forest. A random forest is an ensemble of learners with a built-in mechanism for feature selection, such as ID3 and C4.5 (Jain & Singh, 2018; Tang et al., n.d.). Base learners, i.e. decision trees, look at each feature in the feature space individually and assign importance to them based on how well they contribute to the model attaining an optimal fit. Features with the lowest importance are discarded, and the forest with the least number of features and highest predictive performance is selected (Tang et al., n.d.) (Fig. 7.1(c)). Using the top 20 molecular descriptors from the random forest predictor importance method, Newby *et al*. (Newby et al., 2013) obtained more accurate decision tree classification models in most cases, compared to the use of filter methods such as information gain, chi-square and greedy search.

Pruning is another embedded feature selection approach that has been applied to neural networks as well as classical learning algorithms, specifically support vector machines (SVMs) (Chandrashekar & Sahin, 2014). For instance, SVM-recursive feature elimination (SVM-RFE) begins with all the features and recursively removes features that do not contribute positively to the model's predictive accuracy. To determine the optimal number of features for an RFE based model, cross-validation is used to evaluate and select the subset with the best performance. Hence, RFE can select the best features for a specific learning algorithm. RFE is considered to be computationally expensive as it traverses through all the features one after the other (Hira & Gillies, 2015). Weighted Kernels (Revathy et al., n.d.) and regularization methods (Osman, Ghafari, & Nierstrasz, 2017), like Lasso, Ridge and Elastic net, have also gained prominence.

2.2.5 Hybrid and Ensemble Feature Selection

Hybrid methods for feature selection involve combining at least two different methods and applying them, usually in succession. Hybrid methods attempt to take advantage of the benefits of the constituent methods while leveraging their strengths. In literature, the most reported is the combination of filter and wrapper methods. Their use has been widely reported for biomedical data (Solorio-Fernandez et al., 2012). Hsu *et al*. (Revathy et al., n.d.) separately filtered two sets of features using F-score or information gain as the filtering criterion. The resulting features were combined and further treated with wrappers (Figure. 2.1(d)). They reported improved predictions in comparison to using filters alone and a decreased computational time compared to using wrappers only. Reddy *et al*. (Reddy, Kumar, & Garg, 2010) applied a Hybrid-GA based descriptor optimization technique for consistently selecting descriptor subsets that represented the whole initial descriptor space. The weights of the selected subsets were analyzed to understand the contribution of each feature to the prediction of HIV protease inhibitors, revealing the role of hydrophobic interactions. This implies the interpretability of the method.

Ensemble methods represent the application of a feature selection method on different subsets of features obtained by using subsampling strategies like bootstrapping.

The resulting features from each of the subsets are aggregated using mean, weights or simple linear aggregation (Ben Brahim & Limam, 2017; Seijo-Pardo et al., 2017) (Fig. 7.1(e)). This method is often used to deal with the challenges of perturbation and instability experienced by most feature selection methods. Seijo-Pardo *et al*. (Seijo-Pardo et al., 2017) provided an in-depth discussion of ensemble methods of feature selection. Dutta *el al*. (Debojyoti Dutta, Rajarshi Guha, David Wild, & Chen, 2007) proposed an ensemble descriptor selection that searches for descriptor subsets using a genetic algorithm whose objective function is a linear combination of the root-mean-square deviation (RMSE) of all the models in the ensemble. They reported an improvement and found that the resulting model had good performance on the PDGFR and COX-2 data sets. A 96% reduction in noise and an improvement in performance was reported by (X.-W. Zhu, Xin, & Ge, 2015), using a recursive random forest to rule out a quarter of the least important descriptors at each iteration. This performed better than the least absolute shrinkage and selection operator (LASSO). The authors highlighted that the difference between the prediction performance of Random Forest and LASSO mainly resulted from the use of variables selected by different strategies, rather than from differences between the learning algorithms.

A summary of the characteristics, strengths and weaknesses of the five classes of feature selection methods are described in Table 2.2 in order to assist a user in choosing the appropriate tool based on user-specific requirements and/or goals.

Table 2.1 A summary of feature selection techniques

Methods	Description	Strengths	Weaknesses	Examples
Filter	• Rank features using a criterion calculated based on the data properties	• Fast, computationally inexpensive, and as such, can be applied to higher dimensions of data • Multivariate methods take the relationship between features into consideration	• Univariate methods ignore feature dependencies • Insensitive to the learner's heuristics • Deciding on the best threshold when selecting from ranked features is not deterministic	• Information gain • Chi-square test • Fisher score • Correlation coefficient • Variance threshold
Wrapper	• Use search strategies to generate feature subsets which are then evaluated by a learner	• Dependencies between features in a subset are considered • Interaction with the learner results in better performance than filter	• Features are learner specific • Interaction with the learner increases the likelihood of overfitting • Computationally expensive	• Sequential feature selection or elimination (e.g. RFE) • Genetic algorithm • Simulated annealing
Embedded	• Are learning algorithms that can weigh the contribution of each feature to its performance	• Interacts with the learner but is less prone to overfitting • Computationally less expensive than wrapper and has better performance than filter • Dependencies between features are inherently considered	• Features selected are learning algorithm specific	• LASSO • Ridge Regression • Elastic Net • Decision Trees
Hybrid	• Combines other methods to achieve the accuracy of wrappers and the efficiency of filters	• Better performance than filters and less computationally demanding than wrappers	• The setbacks of the filter and wrapper methods are not eliminated, they are reduced. The features remain specific to the learning algorithm	• Filter followed by embedded methods • Hybrid genetic algorithms
Ensemble	• Aggregates the output of different feature selection methods or subsets	• Ensures stable and robust feature selection	• Depending on the constituent methods, it could be computationally expensive and difficult to understand	• Could be made up of multiple feature selection methods

2.3 Feature Extraction

The algorithms employed for mathematical representation of molecular descriptors and fingerprints are independent of the size of molecules, allowing the generation of a fixed length set of descriptors for every molecule regardless of size (Danishuddin & Khan, 2016). The generation of fixed length vectors can introduce redundant descriptors for certain molecules within a library. An optimized feature set achieved by feature extraction can minimize redundancy, noise, correlation between descriptors, and consequently generate classifiers with improved prediction accuracy (Venkatraman et al., 2004).

A mathematical description of feature extraction is as follows: Considering a descriptor space, $x \in R^n$, find a mapping $y = f(x)$ to obtain transformed feature vector y , where $y \in R^k$ and $k < n$. The vector y should preserve the majority of molecular information in R^n. The goal is to achieve a reduction in dimension without negatively impacting the prediction performance. An optimal mapping, $y = f(x)$, is one that minimizes the prediction error.

Feature extraction transforms the initial feature space to a new, lower dimension feature space by combining the features in the original space. As a result, it is difficult to associate the new features with the old. Further analysis, such as feature importance explanation, becomes very difficult as there is no physical meaning for the newly mapped features that are obtained from feature extraction. Next, some commonly used feature extraction techniques are discussed.

2.3.1 Principal Component Analysis

Principal component analysis (PCA) is a multivariate, non-parametric method employed for dimensionality reduction (Lauria, Ippolito, & Almerico, 2009; Yoo & Shahlaei, 2018). It works by performing a linear combination of the features, also referred to as the principal components, to achieve the maximum variance. At its core, PCA is centered on determining the eigenvectors of the input data's covariance matrix. This linear transformation can minimize redundancy and reduce the number of features, which increases the information in the resulting features. Each of the resulting features, called principal components, is a combination of several original features. These principal components are also highly uncorrelated because the first principal component accounts for as much of the variability in the data as possible, and each succeeding component accounts for as much of the remaining variability as possible (Van Der Maaten et al., 2009). A detailed discussion on the different applications of PCA in SAR modeling was provided in (Yoo & Shahlaei, 2018). Klepsch *et al*. (Klepsch, Vasanthanathan, & Ecker, 2014) applied PCA to a curated P-glycoprotein inhibitors data set of 1608 compounds, where the first two principal components were reported to explain 71.7% of the variance in the data set. This approach was applied to classification, and an analysis into the effect of the initial descriptors on these two components showed that hydrophobic information, such as the number of aromatic bonds and the partition coefficient, was the major contributor to the principal components. According to (Hemmateenejad, Miri, Jafarpour, Tabarzad, & Foroumadi, 2006), 2-aryl-1,3,4-Thiadiazole derivatives were classified into distinct clusters of active or inactive molecules when PCA was performed instead of using all of the descriptors calculated.

Considering that principal components are combinations of the original features, all the original features are still available within the components. This is useful for interpretation of models because knowing the original features that contribute to a component can reveal the types of features that are closely related. A key challenge with PCA is that it is unable to handle data with complicated structures that may not be represented in a linear subspace (Manikandan & Abirami, 2018). Kernel PCA (KPCA) (Reverter et al., 2014; Q. Wang, 2011) was designed to serve as the nonlinear form of PCA. KPCA is based on kernel functions that intrinsically perform a nonlinear mapping of the input space to a feature space followed by performing linear PCA in this feature space. KPCA generated vectors have been used to train SVM models (Hemmateenejad et al., 2006), and it was shown that KPCA is efficient over a wide range of virtual screening dataset inputs using MACCS and ECFP fingerprints. It was also observed that the KPCA embedding largely depended on the properties of the underlying representation as its performance on the ECFP fingerprint varied with the hashing employed.

2.3.2 Autoencoder

Autoencoders (Baldi, 2012; Goh et al., 2017) are unsupervised neural networks with an odd number of hidden layers that can be applied for nonlinear feature extraction. They employ the backpropagation algorithm to try to create a set of output values which are equal to the input by minimizing the error between the output and the input layer. As shown in Figure 7.2, The network architecture can be designed such that the middle layer is smaller, i.e. has fewer nodes than the input and output layers.

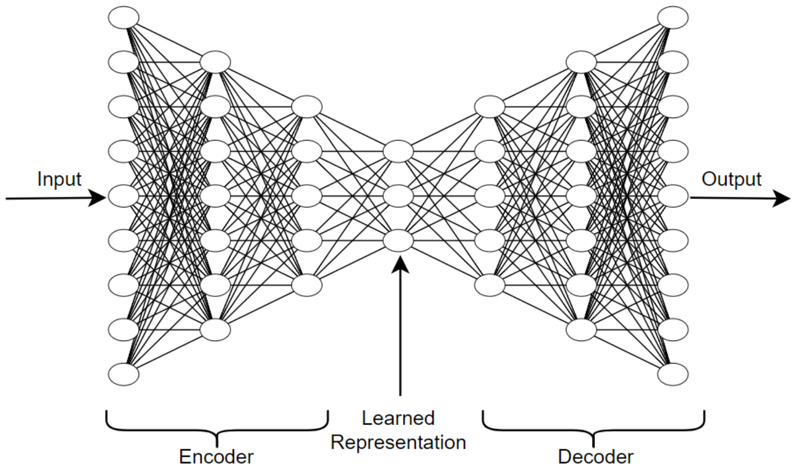


Figure 2.2 An autoencoder indicating the reduced dimension in the middle layer.

The network is forced to learn a compact representation (embedding) of the input data (Chandra & Sharma, 2015). In an early work, Hinton *et al*. (Hinton & Salakhutdinov, 2006) demonstrated that autoencoders generated embeddings of images that were used to reconstruct images. A major drawback of autoencoders is that physical meaning for theoretical insight will be lost. They are also complex to train because they typically require a large amount of training data and a search through many possible hyperparameter values. Blaschke *et al*. (Blaschke, Olivecrona, Engkvist, Rgen Bajorath, & Chen, 2018) employed generative autoencoders to design new molecules *in silico* based on the recreated output layer. (Burgoon, 2017) used autoencoders to screen chemicals for potential estrogenic activity by projecting the two neurons in the middle layer into a Cartesian plane. The application of autoencoders for toxicity prediction has not been widely reported, especially for feature extraction. This provides an opportunity for a future area of research.

2.3.3 Linear Discriminant Analysis

Like PCA, Linear Discriminant Analysis (LDA) (Chandra & Sharma, 2015; Ye, n.d.) is a linear transformation technique commonly used for dimensionality reduction. However, LDA is supervised since the discrimination power of the features is taken into consideration. LDA computes an optimal transformation (projection) of the input data on to a line such that classes are separated as clusters. The goal of the projection is to ensure maximum class discrimination by minimizing the within-class distance while maximizing the between-class distance (Van Der Maaten et al., 2009). A weakness of LDA is that if the distribution of a dataset is significantly non-Gaussian, the LDA projections will not be able to preserve any complex structure of the data (Yan & Dai, 2011). Thus, the resulting features may not have good discriminative power. Features extracted with LDA were used by Ren *et al*. (Ren et al., 2016) in a stepwise forward manner from a combined pool of experimental data, and chemical structure-based descriptors were employed for predicting aquatic toxicity mode of action. In this work, logistic regression was shown to have a better predictive performance than LDA using the extracted features, with a 7.3% improvement over previously reported classification rates.

In addition to the above-mentioned non-linear dimensionality reduction techniques, there are also spectral and manifold learning methods, such as t-distributed Stochastic Neighbor Embedding (t-SNE) (Maaten & Hinton, 2008), Multi-dimensional Scaling (MDS) (*Modern Multidimensional Scaling*, 2005), Spectral Embedding (Belkin & Niyogi, 2003), and Isomap (Tenenbaum, de Silva, & Langford, 2000). Manifold learning, a class of unsupervised non-linear algorithms, assumes that the dimensionality of a datasets is only artificially high, and thus attempts to uncover the intrinsic low dimensionality. Typically, these algorithms work by computing the similarities between points to find a nearest-neighbor, and then an eigenproblem for embedding high-dimensional points into a lower dimensional space (Izenman, 2012).

2.4 Miscellaneous

2.4.1 Feature Stability

It is common to use the performance of a model as the metric to evaluate the suitability of a feature reduction algorithm. Therefore, it is an obvious choice to optimize the selection process to obtain the best prediction power possible. However, the stability or degree of variance of feature selection methods becomes a crucial challenge when the task at hand goes beyond optimizing prediction accuracy to include improving interpretability. A simple scenario may be the case for using substructure-based descriptors for SAR modeling. It is common to consider a substructure that is very relevant for prediction as a major contributor to the activity of that molecule, implying a potential research target. However, many feature selection algorithms tend to be unstable and would yield a different subset if a little perturbation is applied (i.e. when new training samples are added or when some training samples are removed). If every perturbation results in wide variation in the selected subset, then it is difficult to conclude that a feature may be important to the molecule's activity.

Kalousis *et al.* (Kalousis, Prados, & Hilario, 2007) defined the stability of a feature selection algorithm as "the robustness of the feature subset the algorithm produces in the presence of perturbations in training sets drawn from the same generating distribution." Essentially, stability quantifies how different training sets affect the variation in the selected feature subset. Hence, a similarity measure is often employed to measure the stability of feature selection algorithms. A reliable algorithm should produce the same or similar subset for any perturbations in the training data. Alelyani *et al.* (Alelyani, Liu, & Wang, 2011) performed experiments to investigate the causes of instability and reported that dimension, sample size and the distribution of the training data influenced stability. Larger sample size translated to improved stability, while larger dimensions caused negative effects. Thus, researchers should pay attention to the characteristics of a training data set. Certain algorithms are also more prone to instability than others. *ReliefF* based feature selection is affected by the order of samples in a training set, while stochastic search algorithms like GA that use random initialization parameters tend to yield subsets that are unstable (P. Yang, Ho, Yang, & Zhou, 2011; P. Yang, Zhou, Yang, & Zomaya, 2013). Various metrics for measuring stability have been proposed (P. Yang et al., 2013). To overcome the stability challenge, it has been suggested to employ ensemble selection algorithms based on the technicalities of the selection algorithm in use (Abeel, Helleputte, Van de Peer, Dupont, & Saeys, 2010; Feng Yang & Mao, 2011; P. Yang et al., 2013). Some of these algorithms include Bootstrap sampling, random data partitioning, parameter randomization, or the combination of several of these. Developing algorithms for feature selection that are stable and possess high predictive power is still an open and challenging area. SAR based toxicity prediction stands to gain a lot from such techniques that can improve speed and accuracy of predictions for regulatory as well as lead optimization purposes.

2.4.2 Validation of Feature Selection

In selecting the optimal feature subset, it is common to evaluate the performance of a learner based on its prediction error. A very common and overlooked mistake is to select features using the entire data set as a preprocessing step. While this appears to be obviously wrong, it has been reported that many researchers, especially in the biomedical fields, continue to make this mistake and successfully publish in top ranking journals (Ambroise & McLachlan, 2002; Hastie, Tibshirani, & Friedman, n.d.). If a test set is to be used to evaluate the performance of a feature set, it must not be involved in the feature selection step as that will result in a selection bias that will yield overly optimistic performance estimates. This is because the features used will have an unfair advantage since they were chosen based on all of the samples. As a result, the model would have gained insight about the features which are more important in the test set. This challenge is more common with wrapper methods (Ambroise & McLachlan, 2002).

In many practical cases of SAR-based toxicity modeling, there are rarely a large number of compounds across the different endpoints to be predicted. This makes it difficult to set aside a reasonable batch of data for evaluation purposes. Methods such as cross-validation and bootstrap sampling can be used to avoid sampling bias (Ambroise & McLachlan, 2002; Guyon & Elisseeff, 2003; Hastie et al., n.d.). Cross-validation techniques like leave-one-out cross-validation (LOOCV) and the *k*-fold method were suggested. Feature selection is to be done in the inner loop of the cross-validation procedure, hence the algorithm takes the following form for a k-fold technique (Hastie et al., n.d.):

```
(i) Randomly shuffle the data set
(ii) Randomly split the dataset into $K$ folds
(iii) For each fold k = 1, 2, . . . , $K$;
    Perform feature selection to obtain an optimal subset with good univariate correlation with the desired endpoint using all the data except the $k^{th}$ fold
    Use the selected features and build a multivariate model with all data except the $k^{th}$ fold. Perform an evaluation using the $k^{th}$ fold
(iv) Aggregate the performance across all $K$ folds to get an unbiased evaluation.
```

2.5 Summary

QSAR-based predictive toxicity modeling methods are faced with input spaces of thousands of features. To improve the ability of a learner to find a generalizable relationship between molecular descriptors and the toxicity endpoint of interest, it is expedient to provide the learning algorithm with the minimum number of descriptors while ensuring that the resulting model is interpretable and computationally inexpensive to build. The relevance of a descriptor is assessed by its ability to discriminate between classes in qualitative classification or its correlation to a scalar in quantitative prediction.

This chapter discussed different feature selection and extraction methods applicable to SAR-based toxicity modeling. The strengths and weaknesses of each method are highlighted. The choice of which to use should largely depend on the available data set, and it is suggested to beginn a new task with a few baseline performance values from a number of methods since no single approach is universally superior. Where the importance of descriptors is sought, feature selection methods such as *filter*, *wrapper*, *embedded* or their combinations (*hybrid* and *ensemble*) may apply.

Feature extraction methods transform the features into a lower dimension while altering the physical meaning of the features. More analysis may be required to interpret the selected features. The stability of selected features and proper feature subset validation methods are often overlooked. Feature selection bias can be avoided by embedding the feature selection process within the inner loop of a cross-validation process to avoid an overly optimistic performance value. Although dimensionality reduction has been shown to improve model performance, there is still room for improvement when it comes to evaluating and validating feature selection and extraction methods and their stability. For the sake of reproducibility, researchers are encouraged to publish important parameters for feature selection or extraction methods they employed, such as the threshold for a variance score. Regardless of the choice of features (molecular descriptors, fingerprints or a combination) used for modeling, SAR models can benefit from dimensionality reduction techniques.

CHAPTER III - STRUCTURE-ACTIVITY RELATIONSHIP-BASED CHEMICAL CLASSIFICATION OF HIGHLY IMBALANCED TOX21 DATASETS

3.1 Introduction

Structure-activity relationship (SAR) has been frequently used to predict the biological activities of chemicals from their molecular structures. One of the major challenges in SAR-based chemical classification or drug discovery is the extreme imbalance between active and inactive chemicals (Czarnecki & Rataj, 2015). Despite the existence of as many as 10^7 commercially available molecules (Irwin, Sterling, Mysinger, Bolstad, & Coleman, 2012), there is almost always a skew in the distribution of molecules across bioactivity or toxicity classes. Biomacromolecules such as proteins are often highly selective in their binding to small molecular ligands. Regardless of the huge chemical space, only a few compounds are likely to interact with a target biomacromolecule causing biological effects and are consequently labelled as active compounds, whereas the remaining majority are labelled as inactive compounds. This gives rise to a common problem of class imbalance for SAR-based predictive modeling, particularly in chemical classification and activity quantification using machine learning approaches (Dahl et al., 2014; Darnag et al., 2010; Polishchuk et al., 2009).

In machine learning, classifiers are built on data statistics and require a balanced data distribution to achieve optimal performance. Classifiers trained from imbalanced data tend to have a bias towards the majority class. This leads to low sensitivity and precision for the minority class (Galar, Fernández, Barrenechea, Bustince, & Herrera, 2012), even though the minority class is usually of greater importance than the majority class (Hido, Kashima, & Takahashi, 2009; Krawczyk & Krawczyk, 2016). In such fields as toxicology and disease diagnosis, bias towards the majority class may result in a higher rate of false negative predictions (Czarnecki & Rataj, 2015).

The problem of data imbalance has been studied in the context of machine learning for more than two decades (Nitesh V. Chawla, 2005; H. He & Ma, 2013; Krawczyk & Krawczyk, 2016). As a result, a plethora of methods have been proposed to alleviate the skewness of class distribution. These methods can be grouped into three categories: data-level, algorithm-level, and hybrid (Branco, Torgo, & Ribeiro, 2015; Krawczyk & Krawczyk, 2016) . Data-level methods aim to rebalance the training dataset's class distribution either by undersampling the majority class or oversampling the minority class (N. V. Chawla et al., 2002). They also include methods that clean overlapping samples and remove noisy samples that may negatively affect classifiers (Stefanowski, 2016). Algorithm-level methods attempt to alter a given learning algorithm by inducing cost sensitivity that biases a model towards the minority class, which, for instance, may be achieved by imposing a high misclassification cost for the minority class (Branco et al., 2015; Krawczyk & Krawczyk, 2016). Hybrid methods combine the use of resampling strategies with special-purpose learning algorithms (Branco et al., 2015). Ensemble approaches (e.g., bagging and boosting), known to increase the accuracy of single classifiers, have also been hybridized with resampling strategies (Galar et al., 2012).

The selection of appropriate metrics plays a key role in evaluating the performance of imbalanced learning algorithms (Branco et al., 2015; Haibo He & Garcia, 2009). In consideration of user preference (e.g., identifying rare active chemicals) and data distribution, a number of metrics have been proposed, including precision, recall, Area Under the Precision-Recall Curve (AUPRC) (Davis & Goadrich, 2006), Area Under the Receiver Operating Characteristics (AUROC) (Provost, Fawcett, & Kohavi, 1998), F-measure, geometric mean (G-mean), balanced accuracy, etc. (Capuzzi, Politi, Isayev, Farag, & Tropsha, 2016; Drwal et al., 2015; Mayr et al., 2016; Ribay, Kim, Wang, Pinolini, & Zhu, 2016). For instance, precision is not affected by a large number of negative samples because it measures the number of true positives out of the samples predicted as positives (i.e., true positive + false positive). A high AUPRC represents both high recall and high precision. High precision relates to a low false positive rate, and high recall relates to a low false negative rate (Davis & Goadrich, 2006; Saito et al., 2015).

The present study was motivated by the scarcity of reported efforts in the application of the above-mentioned methods to the SAR-based chemical classification domain. A literature survey was done which revealed a few studies in this domain where cost-sensitive learning (J. Chen et al., 2012; Pham-The et al., 2016), resampling (T. Lei et al., 2017; Pham-The et al., 2016) and extreme entropy machines (Czarnecki & Rataj, 2015; Czarnecki & Tabor, 2017) were employed to specifically deal with data imbalance. Although predictive modeling was improved for certain datasets, a consistent performance enhancement was not observed as a result of resampling and algorithm modification. Apparently, more studies are warranted to further examine such questions as: (1) Does imbalance ratio (IR), i.e., inactive-to-active sample ratio, affect the effectiveness of data-level methods (particularly resampling methods)? (2) Would different data rebalancing techniques affect the performance of a classifier differentially, and does the SMOTEENN imbalance handling technique perform better? (3) What metrics can better evaluate the results of imbalanced learning in SAR-based chemical classification? This study attempted to address all three of these questions.

To address the first question, twelve binary datasets of 10K compounds with varying degrees of imbalance were selected, which were generated within the Toxicology in the 21st Century (Tox21) program (NCATS, n.d.) and used for the Tox21 Data Challenge 2014 (R. Huang & Xia, 2017a; R. Huang, Xia, Nguyen, et al., 2016) (https://tripod.nih.gov/tox21/challenge/about.jsp). To address the other two questions, 8 evaluation metrics were chosen, compared three resampling algorithms integrated with the base classifier (random forest - RF), and performed statistical analysis to rank the metrics.

In this work, RF was selected as the base classifier and bagging as the ensemble learning algorithm to improve the stability and accuracy of model predictions. Then, three representative resampling methods for data imbalance handling were applied, i.e., random under-sampling (RUS), synthetic minority over-sampling technique (SMOTE) and SMOTEENN (i.e., a combination of SMOTE and Edited Nearest Neighbor (ENN) algorithms). Consequently, four hybrid learning methods, i.e., RF without imbalance handling (RF), RF with RUS (RUS), RF with SMOTE (SMO), and RF with SMOTEENN (SMN) were tested. Here, it was not intend to conduct a comprehensive or exhaustive comparative investigation of all existing imbalance handling methods, but rather to use this case study to demonstrate that appropriate handling of imbalanced data and the choice of appropriate evaluation metrics could improve SAR-based classification modelling. This chapter investigates the performance of these existing approaches and highlights their limitations regarding imbalance ratio. The rest of the chapter is organized as follows: Section 3.2 (Materials and Methods) covers the study design, data curation and preprocessing steps, imbalance handling methods, and performance metrics. Section 3.3 (Results and Discussion) presents classification performance results, statistical analysis, and a comparison with published results for the Tox21 datasets. Lastly, Section 3.4 (Conclusions) briefly summarizes the major findings from this study.

3.2 Materials and Methods

3.2.1 Study Design

The workflow of this study design is outlined in Figure 3.1. It consists of data preprocessing, feature generation and selection, resampling, model training (ensemble learning), model testing and performance evaluation. The data preprocessing and feature generation steps were applied to a total of 12,707 compounds in the raw dataset of 12 assays. However, feature selection, resampling and training of classifiers were conducted separately for each individual assay. For each assay, the preprocessed compounds in the training set were split into *N* stratified bootstrap samples with replacement (i.e., randomly select samples but maintain the same imbalance ratio). This was followed by ensemble learning either without resampling (RF) or with the application of a resampling technique (RUS, SMOTE, or SMOTEENN). Optimal parameters for each base learner were obtained via grid search with 5-fold cross validation. Optimized base learners were combined to form the final ensemble learner. Evaluation metrics were calculated using the prediction results of RF, RUS, SMO and SMN to statistically compare their performance. Details of the workflow are presented below.

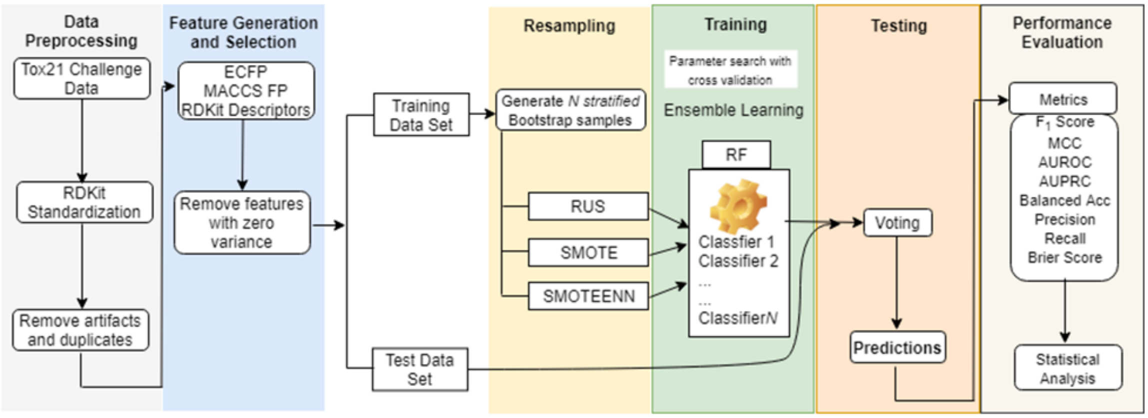


Figure 3.1 Workflow of structure-activity relationship (SAR)-based chemical classification with imbalanced data processing designed for this study.

3.2.2 Chemical in vitro toxicity data curation

The Tox21 Data Challenge dataset used in this study consisted of 12 quantitative high throughput screening (qHTS) assays for a collection of over 10K compounds (with redundancy within and across assays). The 12 *in vitro* assays included a nuclear receptor (NR) signaling panel and a stress response (SR) panel. The NR panel comprised 7 qHTS assays for identifying compounds that either inhibited aromatase or activated androgen receptor (AR), aryl hydrocarbon receptor (AhR), estrogen receptor (ER), or peroxisome proliferator-activated receptor γ (PPAR-γ). The SR panel contained 5 qHTS assays for detecting agonists of antioxidant response element (ARE), heat shock factor response element (HSE) or p53 signaling pathways, disruptors of the mitochondrial membrane potential (MMP), or genotoxicity inducers in human embryonic kidney cells expressing luciferase-tagged ATAD5. There were three sets of chemicals: a training set of 11,764 chemicals, a leaderboard set of 296 chemicals and a test set of 647 chemicals (R. Huang, Xia, Nguyen, et al., 2016). For this study, the leaderboard set was merged with the original training set to form the "training set" and retained the original test set as the "test set". The Tox21 dataset was downloaded in SDF format at https://tripod.nih.gov/tox21/challenge/data.jsp. There were four possible assay outcomes for each compound: active, inactive, inconclusive or not tested. Only those chemicals labeled as either active (1) or inactive (0) were retained for this study.

3.2.3 Compound preprocessing and chemical descriptor (feature) generation

Chemical structures were also downloaded at https://tripod.nih.gov/tox21/challenge/data.jsp as SMILES files. Data cleaning/standardization was carried out in three steps. First, a fragmentation step was performed as previously described (Mayr et al., 2016) where compounds possessing distinct structures not linked by covalent bonds were split into separate "compound fragments". The second step was performed to identify problematic molecules with inconsistent resonance structures and tautomers (Alexander Tropsha et al., 2003), which should not contribute to the biological effect of a compound (Stefaniak, 2015). Standardization was executed using MolVS ("MolVS: Molecule Validation and Standardization — MolVS 0.0.9 documentation," n.d.), a publicly available tool built on RDKit (Greg, n.d.). Briefly, a SMILES entry was canonicalized by standardizing chemotypes such as nitro groups and aromatic rings, and the largest uncharged fragment of the compound was retained. In the third step, the resulting fragments were merged based on their reported activity to exclude replicates and conflicting instances. Specifically, only one instance of a set of duplicates was retained with the most frequent activity label, while duplicates with ambiguous activity labels (i.e., equal number of active and inactive outcomes for the same chemical) were removed. Three types of molecular features (>2000 in total), i.e., RDKit descriptors, MACCS (Molecular ACCess System) keys and Extended-Connectivity Fingerprints (ECFPs) (Rogers & Hahn, 2010) with a radius of 2 and a fixed bit length of 1024, were generated using RDKit (Greg, n.d.) to characterize the final set of compounds. All features with zero variance were dropped.

3.2.4 Sampling and classification methods

Briefly, this section describes the three resampling techniques (i.e., RUS, SMOTE and SMOTEENN) used for handling imbalanced data with RF chosen as the base classifier.

3.2.4.1 RUS

RUS is a widely used undersampling technique which randomly removes samples from the majority class. In this study, RUS was used to randomly remove inactive compounds. While RUS alleviates imbalance in the dataset, it may potentially discard useful or important samples and increase the variance of the classifier. Recent studies have shown that the integration of RUS with ensemble learning can achieve better results (Galar et al., 2012; Seiffert et al., 2010). To overcome its drawbacks, RUS was combined with bagging (an ensemble learning algorithm) for SAR-based chemical classification.

3.2.4.2 SMOTE

SMOTE is an oversampling technique that creates synthetic samples based on feature space similarities between existing examples in the minority class (N. V. Chawla et al., 2002). It has shown a great deal of success in various applications (Haibo He & Garcia, 2009). To create a synthetic data sample, first, a sample was taken from the dataset of the minority class and considered its K-nearest neighbors based on Euclidian distance to form a vector between the current data point and one of those k neighbors. The new synthetic data sample was obtained by multiplying this vector by a random number X between 0 and 1 and adding the product to the current data point. More technical details can be found in (N. V. Chawla et al., 2002; Haibo He & Garcia, 2009). Applying SMOTE to the minority class instances can balance class distributions (N. V. Chawla et al., 2002) and augment the original data set in a manner that generally significantly improves learning (Haibo He & Garcia, 2009).

3.2.4.3 SMOTEENN

Despite many promising benefits, the SMOTE algorithm also has its drawbacks, including over generalization and variance (Haibo He & Garcia, 2009). In many cases, class boundaries are not well defined since some majority class instances may appear in the minority class space, especially for nonlinear data with a large feature space (V. García, Sánchez, & Mollineda, 2012). As a result, some new synthetic samples in the minority class may be mislabeled and attempting to learn from such datasets often results in overfitting (Galar et al., 2013). To remove the mislabeled samples created by the SMOTE technique, SMOTEENN was applied, which is a combination of SMOTE and the Edited Nearest Neighbor (ENN) (Wilson, 1972) algorithm to clean the synthetic data samples.

In the ENN algorithm, the label of every synthetic instance is compared with the vote of its *K*-nearest neighbors. The instance is removed if it is inconsistent with its *K*-nearest neighbors; otherwise, it remains in the data set. A higher *K* value in the edited nearest neighbors algorithm leads to a more stringent cleaning rule that allows more synthetic instances to be eliminated. Applying SMOTEENN to an imbalanced dataset does not automatically result in a perfectly balanced set after resampling, but it creates more meaningful synthetic samples in the minority class and reduces the imbalance ratio to a more manageable level.

3.2.4.4 RF and ensemble learning

RF is a robust supervised learning algorithm that has been widely used for classification in many applications in data science (Breiman, 2001). An RF model consists of many individual decision trees that operate as an ensemble. The individual decision trees are generated using a random selection of features at each node to determine the split. During the classification, each tree votes and the class with most votes becomes the model's prediction.

RF can be built (Han, Kamber, & Pei, 2011) and improved (Altman & Krzywinski, 2017) using bagging (short for bootstrap aggregation). Bagging is a common ensemble method that uses bootstrap sampling in which several base classifiers are combined (usually by averaging) to form a more stable aggregate classifier (Khoshgoftaar, Van Hulse, & Napolitano, 2011). Each base classifier (RF in this study) in the ensemble is trained on a different subset of the training dataset obtained by random selection with replacement, thus introducing some level of diversity and robustness. It is well known that the bagging classifier is more robust in overcoming the effects of noisy data and overfitting, and it often has greater accuracy than a single classifier because the ensemble model reduces the effect of the variance of individual classifiers (Galar et al., 2012; Khoshgoftaar et al., 2011; Laszczyski, Stefanowski, & Idkowiak, 2013).

In this case, the Tox21 dataset was both highly dimensional and highly imbalanced (Nitesh V. Chawla et al., 2003; Galar et al., 2012). With a large feature space and a small number of minority class samples, classification of such datasets often suffers from overfitting. Bagging was the ensemble method of choice because it is less susceptible to model overfitting. Combining the base classifier RF with three sampling techniques (RUS, SMO and SMOTEENN) and bagging, four hybrid classification methods were assembled: (1) RF without resampling, (2) RF + RUS, (3) RF + SMO, and (4) RF + SMOTEENN. For the convenience of result analysis, the four methods were simply denoted as RF, RUS, SMO and SMN, respectively.

Using SMN as an example to illustrate the algorithm that integrates resampling with ensemble learning (see Algorithm 1 and Figure 3.1). First, a subset, S_i, was obtained by taking a stratified bootstrap sampling from the training set, X, and this sampling process was repeated *N* times, where *i* = 1 to *N*, with N ranging between 5 and 100 in steps of 5. Stratification was employed to ensure that each bootstrap had the same class distribution as the entire training set. Each subset is used to train a classifier in the ensemble, hence *N* is also equivalent to the number of classifiers. Then, the SMOTEENN algorithm was applied to S_i to oversample the minority class and obtain an augmented training subset $S_i{}'$, which was used to train a random forest classifier $f_i(x)$. The parameters for each classifier in the ensemble were selected using a grid search with a 5-fold cross-validation. This would give every individual classifier a chance to attain its best performance and contribute optimally to the ensemble. The final ensemble model was a bagged classifier that would count the votes of the *N* classifiers and assign the class with the most votes to a chemical in the test dataset. The other three methods RF, RUS and SMO also employed Algorithm 1 with the only difference being the resampling technique, i.e., no resampling, RUS and SMOTE, respectively. All classifiers were implemented using the Scikit-learn package (Pedregosa et al., 2011) and Imbalanced-learn in a Python toolbox (Lemaˆıtre, Nogueira, & Aridas, 2017).

```
Algorithm 1: $\boldsymbol{N}$ = Number of classifiers, $\boldsymbol{X}$ = Training set
For $i$ from 1 to $N$ (number of classifiers):
    (1) Take a stratified bootstrap sample, $S_i$, from training set, $X$
    (2) Apply SMOTEENN to $S_i$ in order to obtain $S_i{}'$
    (3) Build a classifier $f_i(x)$ using $S_i{}'$ as the training set and 5-fold cross validation with a grid
        parameter search
Obtain the ensemble model, $F(x)$, a collection of the classifiers given as $(f_i(x)|i = 1, \dots, N)$
Prediction of $F(x)$ = majority votes of all $N$ classifiers for a test instance
```

3.2.5 Performance evaluation metrics

The output of a binary classification model can be primarily represented by four terms: (1) true positive (TP) defined as the number of true active chemicals that are correctly predicted as active by the model; (2) false positive (FP) as the number of true inactive chemicals incorrectly predicted as active; (3) true negative (TN) as the number of true inactive chemicals correctly predicted as inactive; and (4) false negative (FN) as the number of true active chemicals incorrectly predicted as inactive. Most evaluation metrics are derived from these four terms. True positive rate (TPR), also referred to as sensitivity or recall, represents the fraction of correctly predicted active chemicals. In SAR modeling, recall is also considered as a measure of the accuracy of the active (minority) class. True negative rate (TNR) or specificity provides a similar measure (accuracy) for the inactive (majority) class. Precision estimates the probability of a model to make a correct active class prediction. F_1 score is the harmonic mean of precision and recall. Similarly, balanced accuracy (BA) is the average of correct predictions for both classes. Matthews correlation coefficient (MCC) offers a good index for the performance of imbalanced classification tasks as it incorporates all the components of the confusion matrix (Boughorbel, Jarray, & El-Anbari, 2017). MCC has been widely used to evaluate the performance of SAR-based chemical classification (Bergmann & Hommel, 1988; R. Huang & Xia, 2017b). The MCC value varies in the range of [-1, 1] with -1 implying disagreement, 1 complete agreement and 0 no correlation between the prediction and the known truth. The Brier score is a measure of the average squared difference between the predicted probabilities and the known value for a class, and it assesses the overall accuracy of a probability model. The formulas of these evaluation metrics are given as follows:

Recall = Sensitivity = $\frac{TP}{TP+FN}$

Specificity = $\frac{TN}{TN+FP}$

Precision = $\frac{TP}{TP+FP}$

F_1 score = $2 \times \frac{Precision \times Recall}{Precision + Recall}$

Balanced accuracy (BA) = $\frac{Sensitivity + Specificity}{2}$

MCC = $\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Brier score = $\frac{1}{N} \sum_{i=1}^{N} (p_i - \ o_i)^2$

where N is the total number of chemicals in a dataset, p_i ($\in [0{,}1]$) is the predicted probability, and o_i is the ground truth for the i^{th} chemical (equal to 1 for active and 0 for inactive). In addition, the two widely used metrics AUROC and AUPRC were also calculated using Scikit-learn (Pedregosa et al., 2011) to evaluate and compare the overall performance of a classifier against another.

Statistical analysis was performed to assess if there existed significant difference among the four investigated classification methods in their performance metrics across the twelve bioassays (Table 1). A nonparametric was adopted to test for multiple comparisons as described in Garcia et al (S. García, Fernández, Luengo, & Herrera, 2010). Using the Statistical Comparison of Multiple Algorithms in Multiple Problems (scmamp) library in R (Calvo & Santafé, 2016), Friedman's aligned-rank test was conducted (Hodges & Lehmann, 2012). The Friedman test was chosen over other statistical tests such as ANOVA because it does not require the assumption of data normality. The Bergmann-Hommel post-hoc test was carried out for pairwise comparisons between SMN and the other three methods (RF, RUS and SMO) (Bergmann & Hommel, 1988).

3.3 Results and Discussion

This section presents (1) a summary of the curated and preprocessed Tox21 dataset, (2) the preliminary comparative results to justify the selection of RF as the base classifier, (3) parameter optimization for RF and ENN algorithms, (4) performance metrics of four classification methods for the twelve imbalanced Tox21 datasets, (5) the impact of IR and classification methods on prediction performance, and (6) a comparison between this study and published Tox21 studies.

3.3.1 Data curation and preprocessing

A summary of the preprocessed training and test datasets of chemicals and their activities from 12 qHTS *in vitro* assays is presented in Table 3.1. Although the original raw Tox21 dataset contained more than 12K chemicals, approximately 50% of them or fewer were retained for each assay after preprocessing. This was primarily due to duplication and the absence of testing data for individual assays. The imbalanced ratio (IR), defined as the ratio of the number of the majority class (inactive compounds) to that of the minority class (active compounds) (V. García et al., 2012), varied widely between assays and between the training and the test sets. Such large disparities offered a great opportunity to investigate the performance of different ensemble-resampling approaches as a function of IR (see below for detailed results). In the training datasets, the highest IR of 41.7 appeared in the dataset of the NR-PPAR-γ assay, whereas the lowest IR of 5.7 was observed with the SR-MMP assay. The test datasets generally had IRs larger than or equivalent to those of their corresponding training datasets, e.g., measuring as high as ~70 for NR-AR-LBD (except for NR-Aromatase, NR-PPAR-γ, and SR-ATAD5).

Table 3.1 Class distribution and imbalance ratio (IR) of the preprocessed training and test chemical datasets from Tox21 Data Challenge. The highest and lowest IRs for the training and test sets are in bold.

***In vitro* qHTS assay ID**	**Total number of chemicals**	**Training set**			**Test set**		
		Inactive	Active	IR	Inactive	Active	IR
NR-AR	6436	5698	166	34.3	560	12	46.7
NR-AR-LBD	5931	5223	143	36.5	557	8	**69.6**
NR-AhR	5596	4445	561	7.9	520	70	7.4
NR-Aromatase	4901	4193	193	21.7	478	37	12.9
NR-ER	5171	4167	500	8.3	455	49	9.3
NR-ER-LBD	6043	5239	221	23.7	563	20	28.2
NR-PPAR-γ	5712	5005	120	**41.7**	558	29	19.2
SR-ARE	4808	3669	603	6.1	448	88	**5.1**
SR-ATAD5	6320	5515	203	27.2	568	34	16.7
SR-HSE	5529	4733	206	23.0	573	17	33.7
SR-MMP	4955	3763	666	**5.7**	472	54	8.7
SR-p53	6009	5110	303	16.9	558	38	14.7

3.3.2 Selecting RF as the base classifier

A comparison of six popular machine learning algorithms, i.e., RF, K-nearest neighbors (KNN), decision trees (CART), Naïve Bayes (NB), support vector machine (SVM) and multilayer perceptron (MLP), was performed using the training datasets of all twelve assays and a stratified 5-fold cross validation. The purpose of this preliminary study was to select a base classifier from these algorithms that were all implemented in Scikit-learn (Pedregosa et al., 2011) with default parameter settings. F_1 score was calculated and used as the metric to evaluate classification performance. As shown in Figure 3.2, RF was the frontrunner for four of the 12 assay datasets, including NR-AR-LBD, SR-ARE, SR-HSE, and SR-MMP. RF was the second best performer for another five assays (i.e., NR-AR, NR-ER, NR-ER-LBD, NR-PPAR-γ, and SR-p53). The average F_1 score of RF for all 12 assays was the highest (0.2783) among all six algorithms, and the runner-up was MLP with an average F_1 score of 0.2487. Clearly, RF outperformed the other five algorithms on the Tox21 dataset, which informed the decision to proceed with choosing RF as the base classifier and to focus this study on imbalance handling methods.

Furthermore, the RF classifier was widely used by the participating teams in the Tox21 Data Challenge [28] [48]. Two of the winning teams developed RF models that achieved the best performance in predicting compound activities against AR, aromatase, and p53 (Barta, 2016) as well as ER-LBD (Uesawa, 2016). Using the same RF classifier and the same dataset made it convenient to compare this results with those from the participating teams and allowed us to better investigate the impact of resampling methods on improving imbalanced learning and, consequently, improving classification performance (see the section 3.3.8 for more info).

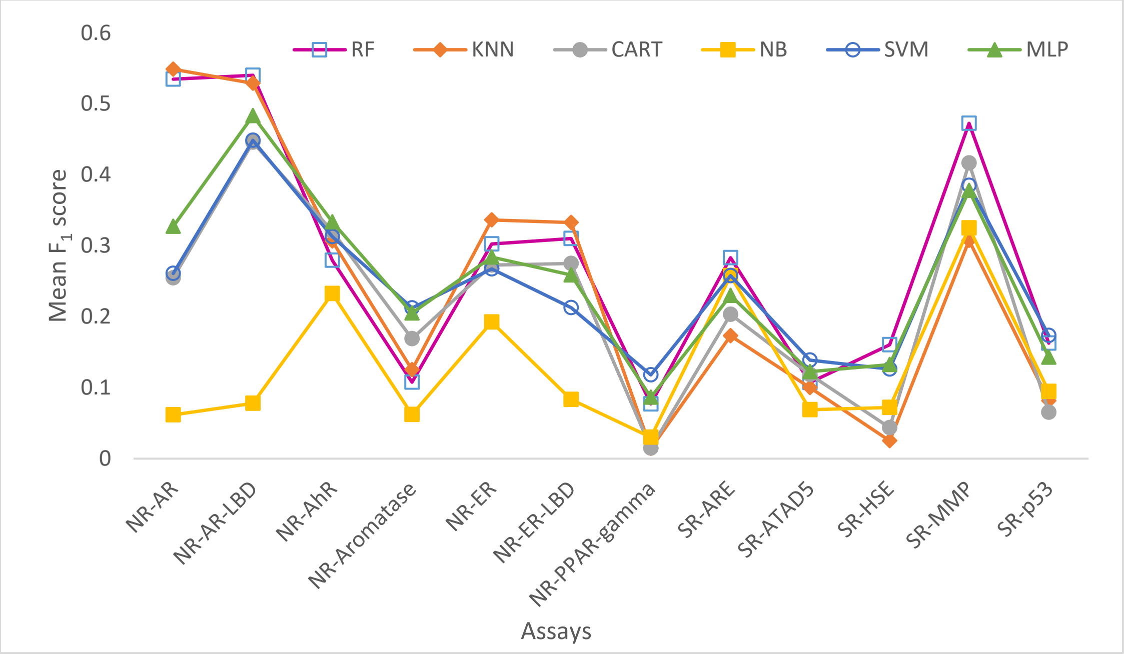


Figure 3.2 A spot check of six popular machine learning algorithms: performance of classifiers trained using the preprocessed Tox21 training datasets as evaluated using F_1 score.

3.3.3 Parameter optimization for the RF classifier

It is generally accepted that the accuracy of a classifier ensemble is positively correlated with ensemble diversity (Kuncheva & Whitaker, 2003). Adjustment to the ensemble diversity was achieved by randomly selecting data instances to create the bootstrap samples and by increasing the number of classifiers included in the ensemble. Figure 3.3 shows that the performance of classifier ensembles measured by the average F_1 score, AUPRC, AUROC and MCC for all four methods changes with the varying number of classifiers in the ensemble. A plateau was encountered when the number of classifiers reached 30, which might be the optimal number of classifiers. After this point, there was little improvement in performance as the number of classifiers increased. Even if minor improvements were noticed using 100 classifiers for some metrics (e.g., MCC), this dramatically increased the computational time and resources needed to train the model. The relationship between performance and the number of classifiers may be explained by the importance of diversity in ensemble learning. With every bootstrap sample being different from another in terms of chemical composition and fingerprint features, diversity in the bagging ensemble was inherent. However, as the number of classifiers increased, the number of times (frequency) that a sample was selected from the same population also increased. This would result in a decline in the variance between such bootstrap samples or a flat line in ensemble diversity. Consequently, a flat line was observed in performance metrics as the number of classifiers in an ensemble increased from 30 to 100 (Figure 3.3). In the subsequent experiments, the optimal number of 30 classifiers for ensemble learning was adopted.

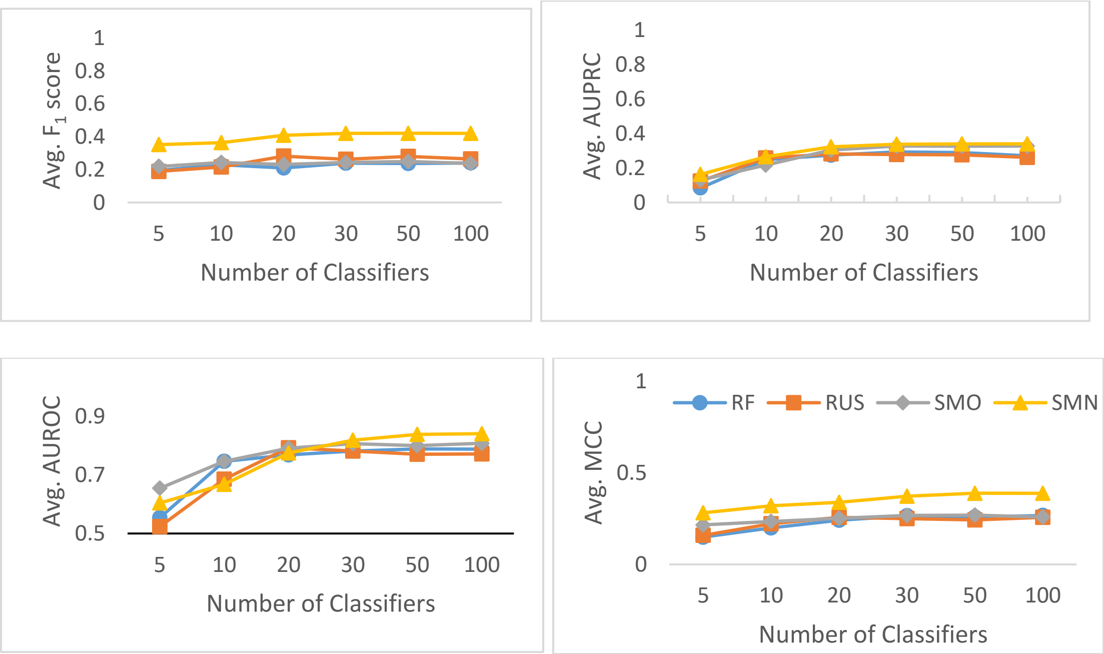


Figure 3.3 Relationship between model performance and the number of classifiers in the RF base classifier.

3.3.4 Optimal number of nearest neighbors (*K*) in the ENN algorithm of SMN models

Another parameter optimized was the *K* value in the ENN algorithm. As shown in Figure 3.4, the number of nearest neighbors *K*, was varied from 1 to 5, and 3 appeared to be the optimal *K* value for most of the five measured performance metrics. F_1 score and AUPRC peaked at *K* =3, BA plateaued when *K* = 3 or 4, whereas MCC peaked earlier at *K* = 2. AUROC was the only metric not affected by the change in *K* value. Thus, the *K* value was set at 3 for SMN in this study.

By setting *K* at this optimal value, ENN may help increase the classifier’s generalizability by removing noisy (mislabeled) synthetic instances introduced in the SMOTE step. By reducing the amount of noise in the dataset while reducing imbalance, it is expected that the class boundaries between active and inactive compounds can be better defined. A reduction in noisy instances can also reduce the chance of over-fitting. This is essentially where the power of SMN lies. However, further increments in the *K* value beyond the optimum led to a decline in classifier performance.

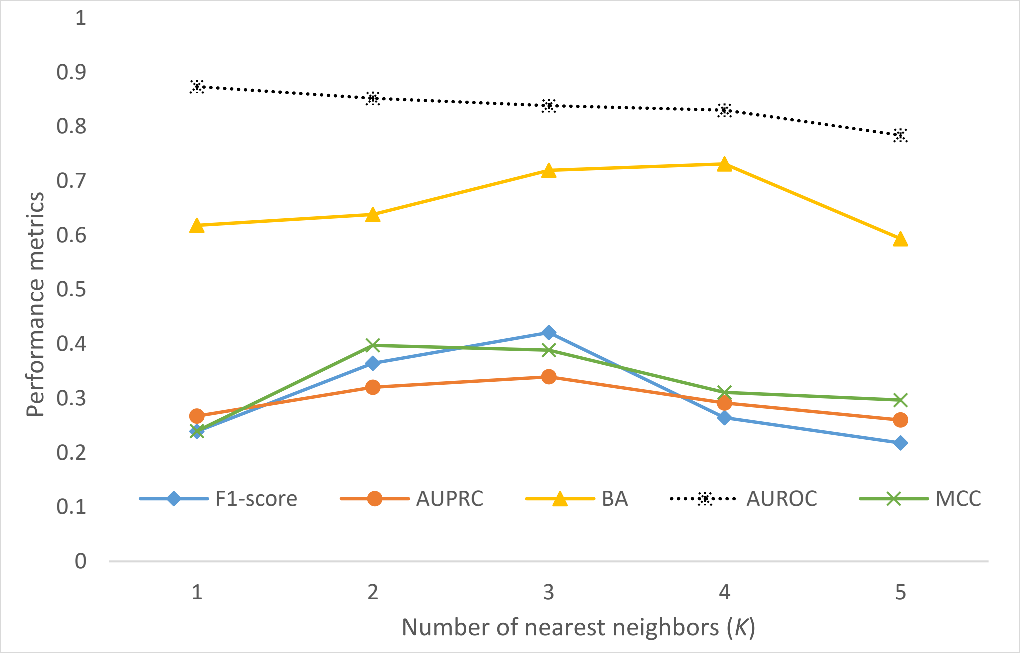


Figure 3.4 Performance metrics of SMN models measured as the number of nearest neighbors (*K*) varied in the ENN.

3.3.5 Performance evaluation metrics

Table 3.2 reports the eight performance metrics of four classification methods (RF, RUS, SMO and SMN) for the 12 assays, with the best performer highlighted in bold for each evaluation metric and assay. For each assay, the training dataset was employed to train a classifier using four different algorithms, and then the trained classifier was applied to the test dataset to determine performance metrics as described in the section3.2 section (also see Figure 3.1). The reported values varied greatly depending on metrics, assays and algorithms.

In general, AUROC has the highest values averaged at 0.8049, whereas MCC has the lowest mean value of 0.2945. This is not surprising as different metrics measure different aspects of learning algorithm performance and trained model quality (Ferri, Hernández-Orallo, & Modroiu, 2009). Accuracy (the ratio of correct predictions to the total number of chemicals) was excluded and specificity because accuracy may be misleading in evaluating model performance for highly imbalanced classification (Provost et al., 1998). Specifically, a high accuracy does not translate into a high capability of the prediction model to correctly predict the rare class, whereas specificity is less relevant as there is more interest in the positive class (active minority). However, the eight chosen metrics are not necessarily the ideal ones for evaluating the performance of classification with a skewed class distribution. For instance, both AUROC and AUPRC can provide a model-wide evaluation of binary classifiers (Saito et al., 2015). Although AUROC, proposed as an alternative to accuracy (Provost et al., 1998), is unaffected by data skewness (Jeni, Cohn, & De La Torre, 2013), it may provide an excessively optimistic view of an algorithm's performance on highly imbalanced data (Davis & Goadrich, 2006). AUPRC, on the other hand, is affected by data imbalance (Jeni et al., 2013), but it is a more informative and more realistic measure than AUROC for imbalanced classification (Saito et al., 2015). Another example is precision and recall, both of which depend on a threshold selected to determine if a chemical compound is active or inactive. A higher recall may be obtained by setting a lower threshold (increasing the number of TP predictions and decreasing the number of FN predictions), which results in a lower precision (more FP predictions). On the other hand, raising the threshold for labeling active chemicals may benefit precision but hurt recall. Optimizing both precision and recall occurs with a tradeoff, especially with imbalanced data. F1 score appears to be a balanced trade-off between precision and recall. Nevertheless, like AUPRC, F1 score is also attenuated by data skewness (Jeni et al., 2013). Given the pros and cons of these metrics, it is necessary to use a suite of metrics for performance evaluation.

Table 3.2 Eight evaluation metrics of four classification methods (RF, RUS, SMO and SMN) for twelve Tox21 qHTS assay datasets. The metrics were calculated using the test datasets (see Table 3.1). The best performer among the four classifiers is highlighted in bold for each assay and each evaluation metric. The highest value represents the best performer except for Brier score which is the opposite (i.e., the lower the better).

Metrics	Classifier	NR-AR	NR-AR-LBD	NR-AhR	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-γ	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	Mean	CV[#]
F_1 score	**RF**	0.1538	0.0000	0.4340	0.2326	0.2727	0.2400	0.0606	0.3359	0.2500	**0.2500**	0.5106	0.1364	0.2397	251%
	RUS	0.1176	**0.1667**	0.4507	0.2222	0.2605	0.1849	**0.4051**	0.4185	0.2063	0.1058	**0.5867**	0.2527	0.2815	189%
	SMO	**0.2500**	0.0000	0.3883	0.1905	0.3692	0.2857	0.1765	0.2927	0.2439	0.1905	0.3902	0.1395	0.2431	193%
	SMN	0.1951	0.1111	**0.5856**	**0.5070**	**0.6078**	**0.3636**	0.3929	**0.6791**	**0.3636**	0.2400	0.5850	**0.4225**	**0.4211**	101%
MCC	**RF**	**0.2859**	-0.0050	0.4101	0.3202	0.2726	0.2891	0.0767	0.2770	**0.3377**	**0.2619**	0.4701	0.1801	0.2647	187%
	RUS	0.1056	**0.1602**	0.4209	0.1914	0.1816	0.1908	**0.3810**	0.2950	0.2049	0.1190	**0.5537**	0.2769	0.2568	205%
	SMO	0.2805	-0.0071	0.3669	0.2792	0.3990	0.3018	0.2355	0.2498	0.3091	0.2327	0.3662	0.2019	0.2679	147%
	SMN	0.1886	0.0975	**0.5342**	**0.4711**	**0.5643**	**0.3404**	0.3627	**0.6177**	0.3261	0.2226	0.5492	**0.3872**	**0.3885**	109%
AUROC	**RF**	**0.8232**	0.7963	0.9063	0.7356	0.7601	0.6963	0.6640	0.7867	0.7827	0.7610	0.9194	0.7443	0.7813	12%
	RUS	0.6785	**0.9133**	0.8852	0.7627	0.7174	0.7619	**0.7937**	0.7698	0.7791	0.7065	0.9295	0.8168	0.7929	13%
	SMO	0.7780	0.7509	0.8936	0.8112	0.7296	0.8072	0.7872	0.7714	**0.8151**	0.7983	0.8893	0.8510	0.8069	8%
	SMN	0.6810	0.7969	**0.9196**	**0.8500**	**0.8628**	**0.8233**	0.7713	**0.8910**	0.8093	**0.8483**	**0.9294**	**0.8785**	**0.8384**	10%
AUPRC	**RF**	**0.3521**	0.0565	**0.5846**	0.2825	0.3203	0.1887	0.1120	0.4224	0.2881	0.1608	**0.5632**	0.1881	0.2933	194%
	RUS	0.1444	**0.1068**	0.4836	0.2043	0.2420	0.1545	**0.5067**	0.4140	0.2423	0.0622	0.5237	0.2295	0.2762	214%
	SMO	0.3290	0.0821	0.5065	0.3504	0.3895	**0.2658**	0.2806	0.4052	**0.3350**	**0.1993**	0.4928	0.2913	0.3273	110%
	SMN	0.0685	0.0639	0.5660	**0.3845**	**0.5688**	0.2018	0.3736	**0.6443**	0.2422	0.1134	0.5234	**0.3254**	**0.3396**	178%
Balanced accuracy (BA)	**RF**	0.5417	0.4991	0.6518	0.5665	0.5830	0.5732	0.5146	0.6016	0.5726	0.5847	0.7053	0.5368	0.5776	17%
	RUS	0.5929	**0.6124**	0.8129	0.6828	0.6513	**0.6968**	**0.7454**	0.6977	**0.7133**	**0.6665**	**0.8523**	**0.7777**	0.7085	15%
	SMO	0.5815	0.4982	0.6304	0.5530	0.6181	0.5964	0.5499	0.5833	0.5718	0.5571	0.6354	0.5377	0.5761	12%
	SMN	**0.6443**	0.5544	**0.8228**											

Table 3.2 Eight evaluation metrics of four classification methods (RF, RUS, SMO and SMN) for twelve Tox21 qHTS assay datasets. The metrics were calculated using the test datasets (see Table 3.1). The best performer among the four classifiers is highlighted in bold for each assay and each evaluation metric. The highest value represents the best performer except for Brier score which is the opposite (i.e., the lower the better), Continued

Metrics	Classifier	NR-AR	NR-AR-LBD	NR-AhR	NR-Aromatase	NR-ER	NR-ER-LBD	NR-PPAR-γ	SR-ARE	SR-ATAD5	SR-HSE	SR-MMP	SR-p53	Mean	CV[#]
Precision	**RF**	**1.0000**	0.0000	**0.6389**	**0.8333**	0.5294	**0.6000**	0.2500	0.5116	**0.8333**	0.4286	**0.6000**	0.5000	**0.5604**	85%
	RUS	0.0769	0.1250	0.2991	0.1302	0.1604	0.1111	0.3200	0.2869	0.1193	0.0576	0.4583	0.1464	0.1909	333%
	SMO	0.5000	0.0000	0.6061	0.8000	**0.7500**	0.5000	**0.6000**	0.5143	0.7143	**0.5000**	0.5714	**0.6000**	0.5547	66%
	SMN	0.1379	0.1000	0.4775	0.5294	0.5849	0.3333	0.4074	**0.5748**	0.2963	0.1818	0.4624	0.4545	0.3784	117%
Recall	**RF**	0.0833	0.0000	0.3286	0.1351	0.1837	0.1500	0.0345	0.2500	0.1471	0.1765	0.4444	0.0789	0.1677	445%
	RUS	0.2500	**0.2500**	**0.9143**	**0.7568**	**0.6939**	**0.5500**	**0.5517**	0.7727	**0.7647**	**0.6471**	**0.8148**	**0.9211**	**0.6573**	52%
	SMO	0.1667	0.0000	0.2857	0.1081	0.2449	0.2000	0.1034	0.2045	0.1471	0.1176	0.2963	0.0789	0.1628	332%
	SMN	**0.3333**	0.1250	0.7571	0.4865	0.6327	0.4000	0.3793	**0.8295**	0.4706	0.3529	0.7963	0.3947	0.4965	87%
Brier score	**RF**	**0.3817**	0.5425	0.3404	0.3997	0.3883	0.4163	0.3961	0.3725	0.3947	0.4257	0.3215	0.3810	0.3967	35%
	RUS	0.4461	**0.3874**	0.3104	0.3724	0.3793	0.4299	**0.3204**	0.3735	0.3829	0.4871	0.3892	0.3936	0.3894	32%
	SMO	0.4263	0.6739	0.3281	0.3379	0.4205	0.4067	0.4138	0.3881	0.3924	0.4146	0.3467	0.3814	0.4109	53%
	SMN	0.4303	0.4156	**0.2583**	**0.3327**	**0.3134**	**0.3670**	0.3503	**0.2761**	**0.3431**	**0.3491**	**0.2371**	**0.3014**	**0.3312**	53%

[#] Coefficient of variation (CV) = standard deviation/mean of 12 assays

3.3.6 Impact of imbalance ratio on performance metrics

The variation in the same performance metrics between different assay datasets is as high as 445% CV (Table 3.2), suggesting that dataset properties (IR in particular) have a significant impact. The NR-AR-LBD assay with the second highest IR among the training datasets and the highest IR among the test datasets has the lowest average value of the 8 metrics (0.2773), whereas the SR-MMP assay with the lowest IR among the training datasets and the third lowest IR among the test datasets has the highest average metrics score (0.5800) (Table 3.2). This result implies that IR may adversely affect classifier performance.

Nevertheless, systematic assessment of the impact of IR on prediction accuracy remains a challenging problem. The IRs in the assay datasets varied from 5 to 70 (Table 3.1). Correlation coefficients (CCs) between $\log_2$(IR) and the score of five evaluation metrics were calculated (Table 3.3). Except for the CCs between AUROC and RF/RUS/SMO, there exists a strong negative correlation between IR and the performance evaluation metrics F_1 score, MCC, BA, AUPRC and AUROC, which is consistent with earlier reports on the adverse effects of IR on these metrics (Jeni et al., 2013).

Table 3.3 Correlation coefficients (CCs) between $\log_2 IR$ and five performance metrics for all four classification algorithms. Insignificant CCs are highlighted in bold and are those whose absolute values are smaller than 0.5760, the critical value at $\alpha = 0.05$ significance level for the degree of freedom *df* = 10 (i.e., n-2, where n = 12 assays).

Metrics	Algorithms			
	RF	RUS	SMO	SMN
F_1 score	-0.6941	-0.7394	-0.7217	-0.9817
MCC	-0.6419	-0.6180	-0.5778	-0.9761
BA	-0.6227	-0.6274	-0.6539	-0.9461
AUPRC	-0.8418	-0.7148	-0.7034	-0.9628
AUROC	**-0.3713**	**-0.1589**	**-0.2770**	-0.7417

To investigate how IR affects the extent of performance improvement obtained by different resampling techniques, the scores of two metrics (F_1 score and MCC) of all twelve assays are plotted against their $\log_2 IR$ (see Figure 3.5). For both metrics, the trend line of SMN is well above those of SMO, RUS and RF, indicating that SMN performed better than other classifiers. The trend lines of SMO and RUS intertwine with that of RF, suggesting that both SMO and RUS did not consistently improve the performance metrics over the base classifier RF. However, the SMN trend line intercepts with the other three at about $\log_2 IR = 5.5$ (for MCC) or 6 (for F_1 score), suggesting that an IR of 40 is likely the threshold at which SMN can outperform other classifiers. The lower the IR value is, the more improvements SMN can achieve, compared to the RF, RUS and SMO classifiers. When IR approaches 40, the improvements are insignificant. These results demonstrate the limitation of data rebalancing techniques and also provide useful feedback for data acquisition. Whenever possible, practitioners should increase the number of active compounds to reduce the imbalance ratio in order to obtain more accurate predictions in SAR-based chemical classification.

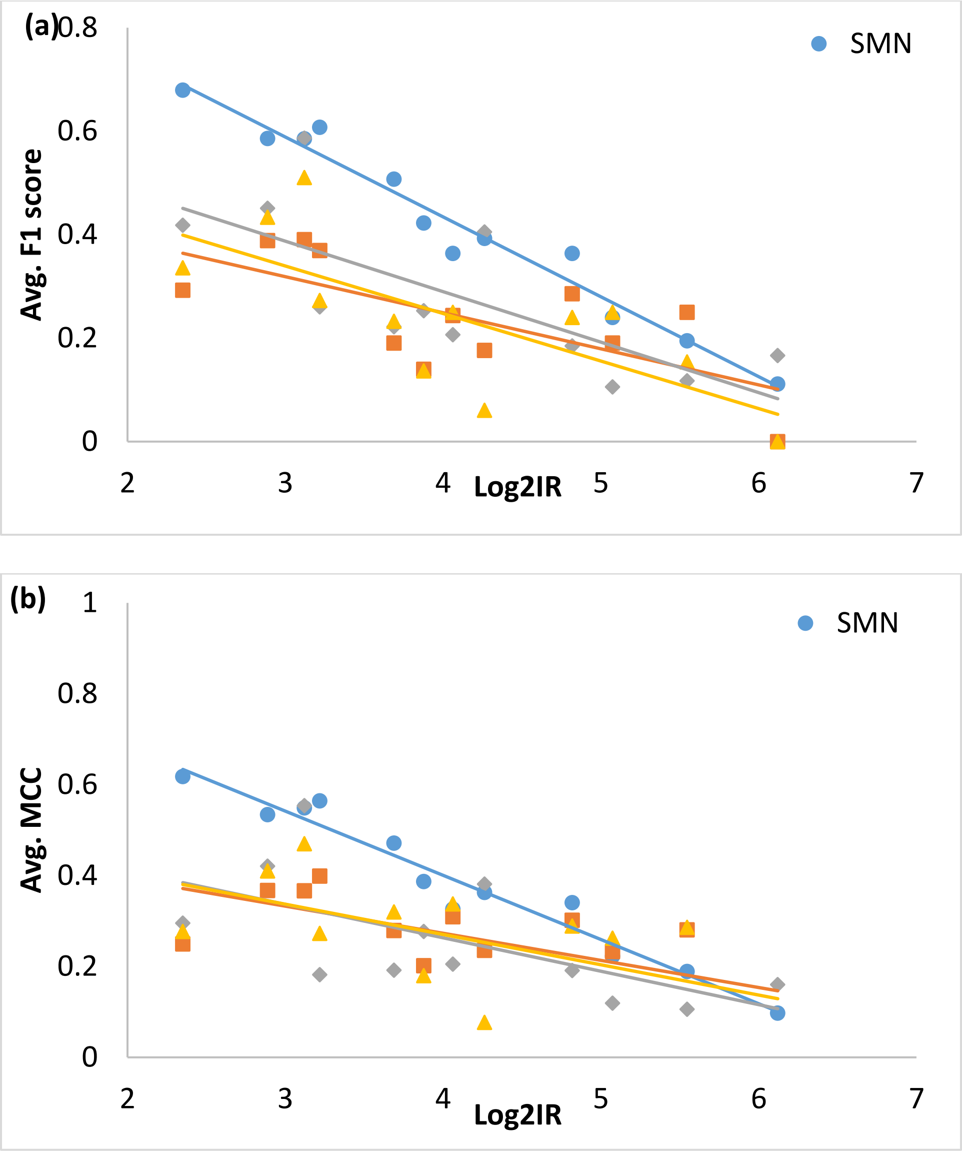


Figure 3.5 The relationship between imbalance ratio (Log$_2$IR) and two prediction performance metrics calculated for four classification methods (SMN, SMO, RUS and RF): (a) F$_1$ score and (b) MCC.

3.3.7 Impact of resampling techniques on classifier performance

The effect of using different algorithms is reflected by a change of 0.0790 in the average metrics score from RF (0.4102) to SMN (0.4892) (Table 3.2). The average Friedman ranking was calculated for each classifier (S. García et al., 2010) by ranking the four algorithms from 1 to 4 based on their performance on each assay dataset. The best classifiers were assigned a rank of 1 and the worst classifiers were assigned a rank of 4. The algorithm with the lowest average rank is considered the best for a specific metric. As shown in Figure 3.6, SMN outperformed the other algorithms (RF, RUS and SMO) in terms of four metrics (F_1 score, AUPRC, AUROC and MCC) and was only slightly surpassed by the frontrunner RUS for the BA metric. Taking F_1 score as an example, SMN performed better in seven of the 12 assay datasets, followed by RUS which was the best performer for three assays (Table 3.2). More interestingly, the magnitude of improvement offered by SMN from the next best method ranged from approximately 8% for the NR-ER-LBD dataset to as much as 27% for the SR-ARE and NR-Aromatase datasets. Understandably, the baseline classifier RF had the worst average performance even though its parameters were also optimized. SMN demonstrated a better F_1 score in most cases because of its ability to improve recall without excessively lowering precision. A moderately higher recall value with comparable precision positively impacts the F_1 score.

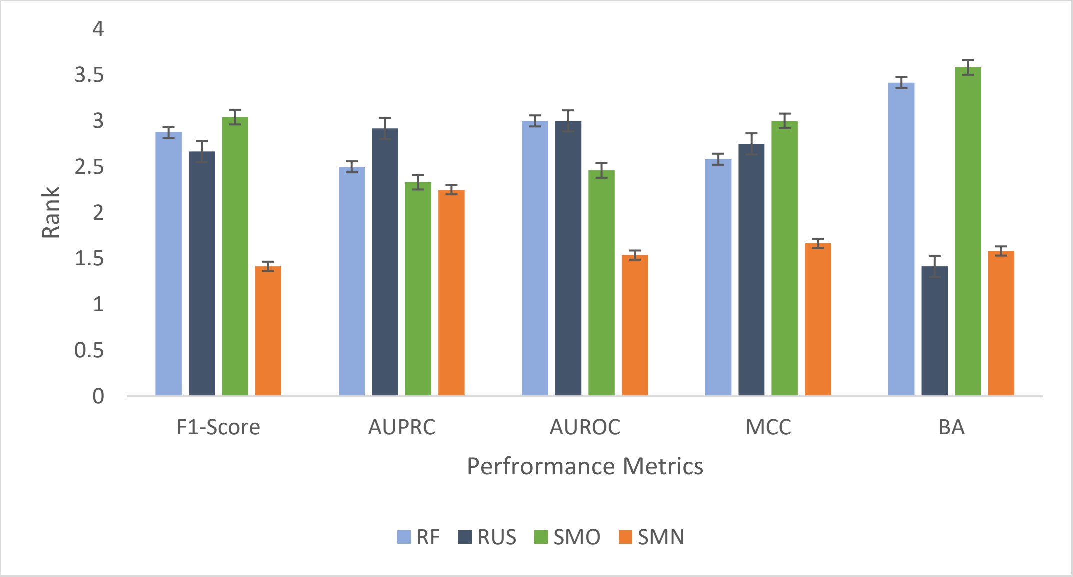


Figure 3.6 Average Friedman ranks of the four classification methods (RF, RUS, SMO and SMN) with respect to five metrics (F_1 score, AUPRC, AUROC, MCC and BA).

The Friedman's Aligned Rank Test for Multiple Comparisons (S. García et al., 2010) was performed to further examine the statistical significance of the algorithmic effects of resampling techniques. The null hypothesis was that all four algorithms had similar capability in classification measured by eight metrics for 12 datasets. Results shown in Table 3.4 suggest that all metrics except AUPRC were significantly affected by the resampling algorithm ($p < 0.05$). The Bergmann-Hommel post hoc analysis was applied to compare pairwise performance metrics of SMN against the other three classifiers. SMN differed more from RF than from SMO and RUS because one, two, and five metrics were insignificantly different ($p > 0.05$) between SMN and RF, SMN and SMO, and SMN and RUS, respectively. F_1 score, MCC and Brier score showed significant difference among the four classifiers in both multiple and pair-wise comparisons. For instance, SMN had the lowest average Brier score of 0.3312 ± 0.0509 (average ± standard error) in comparison with SMO (0.4109 ± 0.0627), RUS (0.3894 ±

0.0361), and the baseline classifier RF (0.3967 ± 0.0395). A lower Brier score indicates that the predictions of a classifier are more accurate because they are closer to the ground truth. MCC, a metric widely used to evaluate the performance of SAR-based chemical classification (Sakkiah et al., 2017; Tong et al., 2003), embodies all the components of the confusion matrix and hence presents a reliable summary of the performance of models trained on imbalanced data.

On the contrary, AUPRC was the sole metric that did not differ significantly in any of the comparisons. AUPRC computes the area under the precision-recall curve that is obtained by using the output of the precision function at different recall levels to assess the overall performance of a prediction model (Pedregosa et al., 2011). SMN showed improved AUPRC scores compared to the other algorithms. However, this improvement was not very substantial. Unlike F_1 score, which benefits from a varied classification threshold, minor improvements in the probabilities for each class do not translate to a marked improvement in the AUPRC score. This is because, being a threshold-independent metric, AUPRC computes the entire area under the curve for the plot of precision versus recall at all possible thresholds. Nevertheless, SMN still showed the best performance in 33% (4/12) of cases tested, RF and SMO in 25% (3/12) each, and RUS in 16% (2/12). The above results suggest that AUPRC is not sensitive to algorithmic effects, whereas F_1 score, MCC and Brier score are sensitive metrics that can distinguish among the classifiers by their performance.

Table 3.4 Friedman's aligned rank test and Bergmann-Hommel post hoc analysis results showing corrected *p*-values for multiple and pair-wise comparisons between SMN and the other three classifiers, respectively. Insignificant statistics (*p* > 0.05) are highlighted in bold.

Comparisons	F_1 score	AUPRC	AUROC	MCC	BA	Precision	Recall	Brier score
All four classifiers	0.0005	**0.1322**	0.0462	0.0111	5.4e-06	9.0e-05	1.8e-06	0.0017
SMN vs RF	0.0003	**0.5253**	0.0168	0.0088	0.0001	0.0278	0.0013	0.0009
SMN vs RUS	0.0051	**0.1008**	**0.0504**	0.0062	**1.0000**	**0.0948**	**0.2307**	0.0022
SMN vs SMO	0.0003	**0.7818**	**0.3320**	0.0088	0.0001	0.0278	0.013	0.0007

3.3.8 Comparison with Tox21 Data Challenge winners

This section presents the comparison between the prediction performance of the four classifiers in this study with those developed by the winning teams for each of the assays in the Tox21 Data Challenge (R. Huang & Xia, 2017b). The winning team for each sub-challenge was judged by AUROC (and BA if there was a tie in AUROC (R. Huang, Xia, Nguyen, et al., 2016)). The AUROC and BA scores of the top ten ranked teams are posted at (https://tripod.nih.gov/tox21/challenge/leaderboard.jsp). The 12 assay sub-challenges were won by four teams: Bioinf@JKU, Amaziz, Dmlab and Microsomes. Bioinf@JKU developed DeepTox models using deep learning (Mayr et al., 2016) and won six out of the 12 assay sub-challenges (NR-AhR, NR-AR-LBD, NR-ER, NR-PPAR-γ, SR-ARE, and SR-HSE) in addition to the Grand Challenge and two additional sub-challenges for the Nuclear Receptor Panel and the Stress Response Panel. Amaziz (Abdelaziz, Spahn-Langguth, Schramm, & Tetko, 2016) employed associative neural networks to develop winning models for SR-ATAD5 and SR-MMP assays, and had the best overall BA score. Dmlab (Barta, 2016) used multi-tree ensemble methods, such as

Random Forests and Extra Trees, to produce winning models for three assays (i.e., NR-AR, NR-aromatase and SR-p53). Microsomes (Uesawa, 2016) chose Random Forest for descriptor selection and model generation, and produced the best performing NR-ER-LBD model. For the purpose of comparison, Dmlab and Microsomes were selected because they used Random Forest. Also the best classifier was compared with the winner of each assay sub-challenge. Given the over-optimistic nature of AUROC, the BA metric provides a more realistic and reliable measure for performance comparison. The titles of the best BA scores were shared by five teams: Kibutz (1 assay), Bioinf@JKU (2), Amaziz (2), T (3), and StructuralBioinformatics@Charite (4). The AUROC and BA scores of the winning teams are shown in Table 3.5 side by side with those of the best performing classifiers because they are the only metrics available for the Tox21 Data Challenge.

Although the AUROC and BA metrics are not ideal for evaluating imbalanced classification, a comparison is made to demonstrate that the improvement obtained from imbalance pre-processing enabled the classifiers to perform equally well or outperform the winning models of the Tox21 Data Challenge. This is primarily reflected by the following observations: (1) the best classifiers outperformed Dmlab and Microsomes in terms of both AUROC and BA by large margins with only four exceptions (NR-AR, NR-PPAR-γ, SR-ATAD5 and SR-MMP), where Dmlab exceeded the best classifiers in AUROC by less than 4%; (2) the best classifiers had the same or higher AUROC and a higher BA than challenge winners for six and three assays, respectively, with less than 8% (AUROC) or 17% (BA) difference for the remaining assays; and (3) on average, the best classifiers performed almost equally well as the challenge winners as a whole (Table

3.5). These results (particularly the BA scores) not only establish the validity, credibility and scientific soundness of the approach, methodology and algorithms implemented in this study, but also demonstrate that the excellence of this work reached levels comparable to that of the Tox21 Data Challenge winners.

Table 3.5 Comparison between this study and Tox21 Data Challenge winners in terms of classification performance metrics AUROC and balanced accuracy. The red-colored values are the highest among all the classifiers (both this study and Tox21 Data Challenge) whereas the values in bold font are the best among the Tox21 Data Challenge participating teams.

Assay ID	AUROC					Balanced accuracy (BA)					Best classifier / Challenge winner	
	Best classifier (this study)		Dmlab	Microsomes	Challenge winner	Best classifier (this study)		Dmlab	Microsomes	Challenge winner		
	value	name				value	name				AUROC	BA
NR-AR	0.8232	RF	**0.83**	N/A	0.828	0.6443	SMN	0.61	N/A	0.736	0.99	0.88
NR-AR-LBD	0.9133	RUS	0.82	N/A	0.879	0.6124	RUS	0.49	N/A	0.650	1.04	0.94
NR-AhR	0.9196	SMN	0.78	0.901	0.928	0.8228	SMN	0.56	0.698	0.853	0.99	0.96
NR-Aromatase	0.8500	SMN	**0.84**	N/A	0.838	0.7265	SMN	0.56	N/A	0.737	1.01	0.99
NR-ER	0.8628	SMN	0.77	0.783	0.810	0.7922	SMN	0.66	0.621	0.749	1.07	1.06
NR-ER-LBD	0.8233	SMN	0.77	**0.827**	0.827	0.6968	RUS	0.59	0.550	0.715	1.00	0.97
NR-PPAR-γ	0.7937	RUS	0.83	0.718	0.861	0.7454	RUS	0.55	N/A	0.785	0.92	0.95
SR-ARE	0.8910	SMN	0.77	0.804	0.840	0.8545	SMN	0.52	0.605	0.729	1.06	1.17
SR-ATAD5	0.8151	SMO	0.80	0.812	0.828	0.7133	RUS	0.61	0.539	0.741	0.98	0.96
SR-HSE	0.8483	SMN	0.86	N/A	0.865	0.6665	RUS	0.56	N/A	0.799	0.98	0.83
SR-MMP	0.9295	RUS	0.95	N/A	0.950	0.8523	RUS	0.69	N/A	0.904	0.98	0.94
SR-p53	0.8785	SMN	**0.88**	0.826	0.880	0.7777	RUS	0.58	0.523	0.765	1.00	1.02
Average	0.8624		0.83	0.810	0.861	0.7421		0.58	0.589	0.764	1.00	0.97

3.4 Conclusions

Due to the specificity of toxicant-target biomolecule interactions, SAR-based chemical classification studies are often impeded by the imbalanced nature of many toxicity datasets. Furthermore, class boundaries are often blurred since active toxicants often appear in the minority class. In order to address these issues, common resampling techniques can be applied. However, removing majority class instances using an undersampling technique can result in information loss, whereas increasing minority instances by interpolation tends to further obfuscate the majority class space, giving rise to over-fitting. In order to improve the prediction accuracy attained from imbalanced learning, SMOTEENN, a combination of SMOTE and ENN algorithms, is often employed to oversample the minority class by creating synthetic samples, followed by cleaning the mislabeled instances. Here, an ensemble approach (bagging) was integrated with a base classifier (RF) and various resampling techniques to form four learning algorithms (RF, RUS, SMO and SMN). They were then applied to binary classification of 12 highly imbalanced Tox21 *in vitro* qHTS bioassay datasets.

Multiple sets of chemical descriptors or fingerprints were generated and down-selected small groups of features for use in class prediction model generation. After data preprocessing, parameters were optimized for both resampling and classifier training. The performance of the four learning methods was compared using eight evaluation metrics, among which F_1 score, MCC and Brier score provided more consistent assessment of the overall performance across the 12 datasets. The Friedman's aligned ranks test and the subsequent Bergmann-Hommel *post hoc* test showed that SMN significantly outperformed the other three methods. It was also found that there was a strong negative correlation between prediction accuracy and IR. It was observed that SMN became less effective when IR exceeded a certain threshold (e.g., >40). Therefore, SAR-based imbalanced learning can be affected by the degree of dataset skewness, resampling algorithms, and evaluation metrics.

The ability to separate the small number of active compounds from the vast amounts of inactive ones is of great importance in computational toxicology. This work demonstrates that the performance of SAR-based, imbalanced chemical toxicity classification can be significantly improved through imbalance handling. Although the best classifiers of this study achieved the same level of performance as the winners of the Tox21 Data Challenge as a whole, it is believed that there is still plenty of room for further improvement. Given the exceptionally outstanding performance of DeepTox (Mayr et al., 2016) and previous experience with deep learning-based chemical toxicity classification (Idakwo et al., 2019), future plans involve replacing RF with a deep learning algorithm like deep neural networks as the base classifier and combine it with class rebalancing techniques to build novel deep learning models for SAR-based chemical toxicity prediction.

CHAPTER IV - DEEP LEARNING-BASED STRUCTURE-ACTIVITY RELATIONSHIP MODELING FOR MULTI-CATEGORY TOXICITY CLASSIFICATION: A CASE STUDY OF 10K TOX21 CHEMICALS WITH HIGH-THROUGHPUT CELL-BASED ANDROGEN RECEPTOR BIOASSAY DATA

4.1 Introduction

Toxicity caused by chemical exposure can be manifested sequentially at ascending organismal levels, which often begins as a molecular initiating event and escalates into adverse effects measured as toxicological endpoints for the cell, tissue, organ, organism or population (Allen, Goodman, Gutsell, & Russell, 2014; Ankley et al., 2010; OECD (Organization for Economic Co-operation and Development), 2013). There exist three categories of chemical toxicity testing strategies: *in vivo*, *in vitro* and *in silico*. Due to the prohibitively high costs and ethical concerns over animal welfare associated with *in vitro* and *in vivo* assays, there has been an increasing demand for reduced animal use as well as a shift in toxicity testing paradigms from *in vivo*/*vitro* to *in silico* (National Research Council, 2007). This demand has also been driven by the 3Rs (Replacement, Reduction, Refinement) movement (Stokes, 2015) and by government policies, regulations and legislation (e.g., REACH by the European Union (European Union, 2006)). Despite significant advances made in the past decades, *in silico* prediction of chemical toxicity without performing any biochemical (ligand binding) or *in vitro*/*vivo* assays remains an unresolved challenge (Li, Yan; Idakwo, Gabriel; Thangapandian, Sundar; Chen, Minjun; Hong, Huixiao; Zhang, Chaoyang; Gong, 2018). Among all *in* *silico* approaches, structure-activity relationship (SAR)-based modeling has become the predominant one, and it is capable of both qualitative classification and quantitative prediction.

Once the toxicity endpoint or biological activity for prediction is set, the performance of SAR-based predictive modelling is largely determined by the choice of molecular descriptors relevant to toxicity (Shao et al., 2013) and of the prediction modelling algorithms (Plewczynski, Spieser, & Koch, 2006). The latter varies from linear methods, such as multiple linear regression (MLR), partial least squares (PLS) and linear discriminant analysis (LDA) to nonlinear methods, such as *k*-nearest neighbors (KNN), artificial neural networks (ANN), decision trees and support vector machines (SVM) (Dudek, Arodz, & Gálvez, 2006). Recently, deep learning, with the Rectified Linear Unit (ReLU) activation function and such architectures as recurrent neural networks (RNN) and convolutional neural networks (CNN), has emerged as a promising tool for *in silico* toxicity or bioactivity prediction modeling (Gao, Igata, Takeuchi, Sato, & Ikegaya, 2017; Hughes, Dang, Miller, & Swamidass, 2016; Hughes, Miller, & Swamidass, 2015; Hughes & Swamidass, 2017; Y. Wu & Wang, 2018; Youjun Xu et al., 2015). Deep learning, also called deep structured learning or hierarchical learning, allows computational models that are composed of multiple processing layers to be fed with raw data and automatically learn multiple levels of abstract representations of data for performing detection and classification (LeCun, Bengio, & Hinton, 2015). The success of deep learning has been well documented in such diverse fields as image and speech recognition (Cummins, Baird, & Schuller, 2018; D. Shen, Wu, & Suk, 2017), visual art (S. Huang et al., 2016), natural language processing (Névéol, Zweigenbaum, & Section Editors for the IMIA

Yearbook Section on Clinical Natural Language Processing, 2018), drug discovery (Dana et al., 2018), bioinformatics (Min, Lee, & Yoon, 2016), computational biology (Angermueller, Pärnamaa, Parts, & Stegle, 2016), and the game of GO (AlphaGo) (Silver et al., 2016).

One of the earliest case studies of applying deep learning in SAR-based toxicity prediction was reported by Mayr and co-workers (Mayr et al., 2016) who developed the DeepTox pipeline. The authors trained deep neural networks (DNNs) using the Tox21 Data Challenge dataset (i.e., training data) that consisted of approximately 12,000 compounds and 12 *in vitro* bioassays (R. Huang & Xia, 2017a; R. Huang, Xia, Nguyen, et al., 2016), and then they predicted the toxicity of approximately 650 chemicals (test data). Although the multi-task DNN exceled in terms of the average AUC (Area Under the Curve of receiver operating characteristics) of the overall 12 bioassays, the nuclear receptor (NR) signaling panel (7 assays), and the stress response (SR) panel (5 assays), it did not perform as well for 5 out of the 12 bioassays as conventional shallow learning techniques did (e.g., SVM, random forest (RF), and elastic net) (Mayr et al., 2016). These results are consistent with the performance of DeepTox in the Tox21 Data Challenge competition where the DeepTox pipeline ranked behind several shallow learning techniques for half of the 12 bioassays even though it won 9 sub-challenges, including those for the other 6 bioassays, the NR and the SR panels, and for the 12 bioassays overall (R. Huang, Xia, Nguyen, et al., 2016; Mayr et al., 2016).

In the past three years, more than a dozen papers have been published with conflicting conclusions on comparative performance between deep learning and shallow learning. For instance, the deepAOT (deep learning-based acute oral toxicity) models constructed using a molecular graph encoding convolutional neural network (MGE-CNN) architecture outperformed previously reported shallow learning models in both quantitative toxicity prediction and toxicant category classification (Youjun Xu, Pei, & Lai, 2017). By pairing element specific topological descriptors (ESTDs) with multitask DNN, TopTox (topology-based multitask deep neural networks) was demonstrated to be more accurate than RF and gradient boosting decision tree (GBDT) using four benchmark ecotoxicity datasets (K. Wu & Wei, 2018). On the contrary, SVM outperformed DNN in predictive classification of chemical-induced hepatocellular hypertrophy (Ambe et al., 2018), and multiple layer perceptron (MLP) exceeded the performance of 2DConvNet (2D Convolutional neural network) in the aforementioned twelve Tox21 bioassays (Fernandez et al., 2018). Meanwhile, Liu *et al*. (R. Liu, Madore, Glover, Feasel, & Wallqvist, 2018) found that the overall performance of DNN models was similar to that of RF and variable nearest neighbor methods. They also concluded that neither a larger number of hidden neurons nor a larger number of hidden layers necessarily leads to better neural networks for regression problems. This contradicted previous observations that deeper and wider networks generally performed better than shallower and narrower ones (Koutsoukas, Monaghan, Li, & Huan, 2017; Lenselink et al., 2017). Recently, Mayr *et al*. conducted a large-scale comparison of drug target prediction between deep learning (Feed-forward neural networks or FNN, CNN and RNN) and shallow learning (RF, SVM, KNN, naïve Bayes (NB), and similarity ensemble approach) methods using a large benchmark dataset (456,331 compounds and more than 1000 assays) from the ChEMBL database (Mayr et al., 2018). Although FNN was statistically identified as the frontrunner across a wide variety of assay targets, the authors observed that RF and SVM had higher average AUC scores than CNN and RNN.

As a new domain with a few years of application history, it is yet to see overwhelmingly significant and convincingly consistent improvements in both quantitative prediction and qualitative classification of chemical toxicity using deep learning. Evidence has indicated that deep learning sometimes does enhance prediction accuracies over shallow learning. However, obtaining such results appears to occur on a case-by-case basis, and the opposite outcomes have also been reported. More studies are warranted to look into many confounding factors such as descriptors, assay targets, chemical space, hyper-parameters, and deep learning architectures, all of which may impact the performance of deep learning in QSAR-based chemical toxicity prediction.

Motivated by the aforementioned controversy, this study was conducted to further investigate if deep learning algorithms could be optimized to offer a significant improvement over representative shallow learning algorithms for a suite of performance metrics. In the following section, two Tox21 quantitative high throughput screening (qHTS) assay datasets with more than 10,000 compounds are described. These cell-based qHTS assays were conducted to identify small molecule agonists and antagonists of the androgen receptor (AR) signaling pathway (R. Huang, Xia, Sakamuru, et al., 2016). Then, such structural features as 1D to 3D molecular descriptors and fingerprints were computed for each chemical. Two algorithms, i.e., DNN (representing deep learning) and RF (representing shallow learning), were employed to build SAR-based classification models so as to compare the accuracy of these methods for predicting chemical class labels (i.e., agonist, antagonist, inactive, and inconclusive). The results suggest that DNN outperformed RF not only significantly by statistical analysis, but by a large margin of more than 20% in four of the five performance metrics. Further in-depth analyses of chemical scaffolding shed insights on the structural alerts for the four classes of chemicals in AR activity, which may aid in future drug discovery and improvement of toxicity prediction modeling.

4.2 Materials and Methods

4.2.1 Bioassay Dataset Curation and Preprocessing

Toxicology in the 21st century (Tox21) is a collaborative initiative launched by the consortium of the NIH, EPA and FDA aiming to develop better toxicity assessment methods.[1] The Tox21 program has tested over 10,000 chemicals against a panel of NR and SR signaling pathways (Attene-Ramos et al., 2013; R. Huang, Xia, Sakamuru, et al., 2016). AR, a nuclear hormone receptor, plays a critical role in AR-dependent prostate cancer and other androgen related diseases (Tan, Li, Xu, Melcher, & Yong, 2015). Two *in vitro* assays were carried out in both agonist mode and antagonist mode to assess the agonistic and antagonistic properties of Tox21 chemicals, respectively. The first assay (BLA assay) used the AR-UAS-bla-GripTite^{TM} cell line that contained the ligand-binding domain (LBD) of the rat AR and stably expressed a beta-lactamase reporter gene under the transcriptional control of an upstream activator sequence (UAS). The second assay (MDA assay) used a human breast carcinoma cell line (MDA-kb2 AR-luc) stably transfected with a luciferase reporter gene. A total of 10,496 chemicals were tested, and their assay outcomes were downloaded from the Tox21 Data Challenge website[2]. The

[1] https://ncats.nih.gov/tox21/about/goals
[2] https://tripod.nih.gov/tox21

downloaded datasets (2 assay modes × 2 assays) were merged using PubChem Substance IDs (SID) because SID was unique for each entry in the datasets. Of the 10,496 compounds, 149 compounds were mixtures of chemicals such as oils and solvents and another 96 compounds contained atoms for which reliable force field parameters were unavailable to perform molecular docking as shown in Figure A.1. Thus, these 245 compounds were removed. There was redundancy in the remaining compounds because, on some occasions, multiple SIDs were found corresponding to the same PubChem Compound ID (CID). Hence, CIDs were used to identify and remove redundant chemicals, resulting in 7665 unique chemicals (see Figure A.1).

For each SID entry, there were up to four records of qualitative assay outcomes that resulted from two assays (BLA and MDA) in two assay modes (agonist and antagonist). There were three possible assay outcomes, i.e., active agonist, active antagonist, or inactive. One of four class labels, namely “agonist”, “antagonist”, “inactive”, or “inconclusive”, was assigned to each chemical by adopting the following rules: a chemical was labeled (i) ‘agonist’ only if both assays in the agonist mode determined it to be an active agonist, (ii) ‘antagonist’ only if both assays in the antagonist mode determined it to be an active antagonist, (iii) ‘inactive’ if all assay outcomes for this chemical were negative, or (iv) ‘inconclusive’ if any other combination was true. In the case of chemical entry redundancy, i.e., multiple SIDs corresponding to the same CID, a consensus was reached on the class label by selecting the most frequently occurring response (i.e., the assay outcome with the highest incidence of occurrence), or the chemical was removed if the assay outcomes were evenly split among multiple categories. Finally, 7665 unique chemicals with unambiguous consensus assay outcomes were obtained and used in the downstream steps (see Figure A.1).

4.2.2 Chemical Dataset Curation and Preprocessing

4.2.2.1 Chemical Structure Preparation

The SMILES of the 7665 unique chemicals were downloaded from PubChem via its PUG REST interface[3] (Kim, Thiessen, Cheng, Yu, & Bolton, 2018) using a custom R script. The Open Babel program (O'Boyle et al., 2011) was used to perform the following steps to clean and optimize the downloaded chemical structures (also see Figure A.1). Salts and other small fragments were removed and only the largest fragment of each entry was retained. SMILES were converted to 2D structures and hydrogens were added when necessary. Then, 3D conformations were generated and partial charges were assigned using the *Electronegativity Equalization Method* followed by energy minimization using the *steepest descent* algorithm (Bultinck et al., 2002; Geidl et al., 2015). Finally, molecular docking was performed to generate biologically relevant 3D ligand conformations within the binding site of the AR because the bound ligand conformation was typically different from the conformations obtained in its unbound state (Sundarapandian, Shalini, Sugunadevi, & Woo, 2010; Tirado-Rives & Jorgensen, 2006). Molecular docking was performed using the AutoDock Vina program (Trott & Olson, 2010) and the X-ray crystal structure of AR-testosterone complex (PDB ID. 2AM9) (de Jésus-Tran Karine et al., 2006). A cubic box of 16×16×16 Å^3 centered at the binding site was used to dock the chemicals in the data set. The docking-generated ligand conformations were used for 3D descriptor calculations (see 2.2.2 below).

[3] https://pubchemdocs.ncbi.nlm.nih.gov/pug-rest

4.2.2.2 Feature Generation and Dimensionality Reduction

A total of 17,967 molecular descriptors and fingerprints (termed features) were generated using PaDEL (Yap, 2011), including 1444 1D or 2D descriptors, 431 3D descriptors, and 16,092 unique fingerprints belonging to 12 different pattern types. The 3D descriptors were calculated using the binding conformations obtained above from molecular docking. In case PaDel failed to compute certain features for certain compounds, the mean-imputation method as implemented in Scikit-Learn (Pedregosa et al., 2011) was employed to replace those missing values. A variance thresholding method was used to reduce feature dimensionality. Any feature vector with at least 85% of its entries being identical was removed, resulting in a final set of 2544 features.

4.2.2.3 Feature Standardization

For many algorithms, it is necessary to rescale the features to keep certain features from getting more influence than they should. This particularly holds true for neural networks where certain weights may update faster than others, thus making optimization methods converge less quickly (LeCun, Bengio, Hinton, et al., 2015). Also, the generated features were of varying scales and distributions, and they were also comprised of count and binary features. To resolve this, the features in the final set were standardized (rescaled) individually such that they assumed a standard normal distribution with a mean of zero and unit standard deviation. Using the StandardScaler function in Scikit-Learn (Pedregosa et al., 2011), the training dataset was rescaled by subtracting the mean and dividing the resulting difference by the standard deviation. The mean and standard deviation used in the training dataset were used to transform the test dataset.

4.2.2.4 Chemical Space Visualization

The chemical space of the 7665 unique Tox21 chemicals was visualized in two-dimensional vectors. The space of the final set of 2544 features was further reduced to two abstract features using an autoencoder (Baldi, 2012; Chandra & Sharma, 2015). By trying to reconstruct the input at the output layer, the autoencoder was forced to learn the underlying feature space in a lower dimension. The innermost layer of the autoencoder, an embedding of the input, was set to two units. The encoder component of the autoencoder had 2544 units in the input layer corresponding to the number of features in the input data and {1024, 512, 128, 32, 2} features in the hidden layers. The decoder component of the autoencoder was ordered as the reverse of the encoder. For activation functions, ReLU was used in the hidden layers while sigmoid functions were used in the output layer. The Adam optimizer was used to minimize the mean squared error. The autoencoder model was trained using the Keras (Chollet, 2015) Python library with a Tensorflow backend.

4.2.3 Machine Learning Methods

4.2.3.1 Machine learning-based SAR modeling approach

The overall workflow of the machine learning-based SAR modeling approach is illustrated in Figure 4.1. It began with data curation, followed by preprocessing of chemical structure and *in vitro* assay data. Nested double-loop cross-validation strategy was employed to ensure robust model development and to alleviate the impact of selection bias and overfitting (Cawley & Talbot, 2010). Similar to most other typical SAR datasets, the 7665 unique chemicals displayed an imbalanced distribution across the four assay outcome classes, i.e., agonist, antagonist, inactive, and inconclusive. As a result of the imbalance, a stratified sampling strategy was adopted to ensure that the partitioning of chemicals across all classes remained the same between the cross-validation folds and between the training and test datasets.

The 7665 chemicals were split randomly using the stratified strategy into 5 subsets. For each run of the outer loop, one subset (20%) was withheld as the test set while the remaining four subsets (80%) were used as the training set. Each of the five runs in the outer loop used a different subset. In the inner loop, the training set was further randomly split into 10 folds using the stratified strategy. Nine folds were used for model (classifier) training or hyper-parameter tuning, while the remaining one fold was used for validation. Thus, a 10-fold cross-validation was implemented in the inner loop for classifier training, whereas a 5-fold cross-validation was executed in the outer loop for model testing and evaluation. The overall performance was assessed using the average metrics values of all five runs in the outer loop (see Section 4.2.4 for metrics definition).

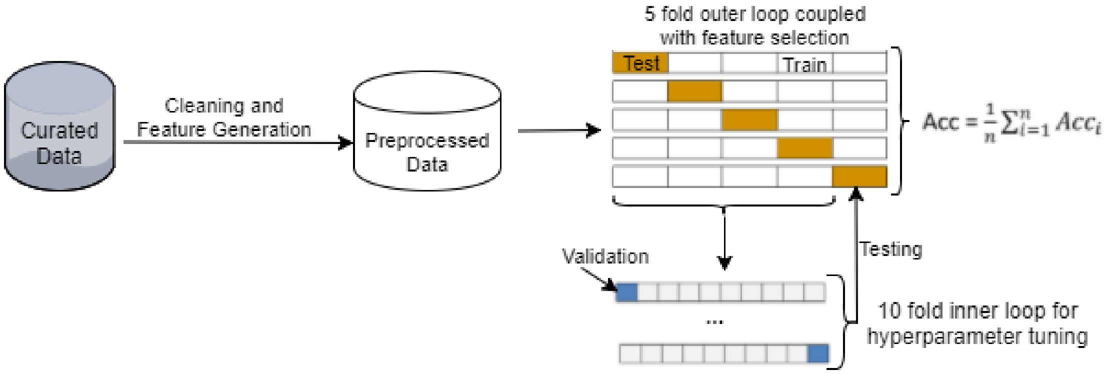

5 fold outer loop coupled with feature selection
Test
Train
Curated Data
Cleaning and Feature Generation
Preprocessed Data
Acc = $\frac{1}{n}\sum_{i=1}^{n} Acc_i$
Validation
Testing
...
10 fold inner loop for hyperparameter tuning

Figure 4.1 Experimental Workflow

4.2.3.2 Shallow and Deep Learning Algorithms

Six commonly used and popular machine learning algorithms were compared in a preliminary study. They included KNN, RF, classification and regression trees (CART), NB, SVM, and DNN, all of which ran under their respective default settings as implemented in Scikit-Learn (Pedregosa et al., 2011). Their performance without optimization was determined by following the workflow presented in Figure 4.1. Based on their performance metrics as shown in Figure A.2, the top two algorithms, DNN and RF were selected, for further optimization and chemical toxicity classification in this study.

4.2.3.3 Random Forest and Optimization

Random forests are a collection of decision trees whose predictions are averaged to obtain an ensemble performance. Randomness is achieved by allowing each tree in the forest to use bootstrap samples of the training data and random molecular features selection for prediction. Decision Trees are drawn upside down and begin with a trunk that splits into multiple branches before eventually arriving at the leaves. The leaf nodes represent the endpoint to be predicted, while all other nodes are assigned a molecular feature. To construct a robust decision tree, the features (nodes) that most clearly differentiate the endpoints (leaf nodes) are chosen. *Gridsearch* with 10-fold cross validation was employed in optimizing the RF models.

4.2.3.4 Deep Learning and Optimization

4.2.3.4.1 Deep learning architecture

This section briefly describes the Deep Neural Networks (DNNs) algorithm its hyper-parameters to facilitate discussion of the optimization and performance analysis process. A DNN is an artificial neural network with one input layer, multiple hidden layers and one output layer, as shown in Figure 4.2. The number of hidden layers is defined as k. Each layer consists of a number of units (or neurons), denoted by n. The number of units at the input layer corresponds to the number of features in the input data $(\boldsymbol{x})$. The number of units in the output layers is equal to the number of classes to be predicted. In this study, there were 4 units in the output layer that corresponded to four classes: (i) agonist, (ii) antagonist, (iii) inactive, and (iv) inconclusive. The number of units in each hidden layer usually depends on specific details of various classification problems and datasets. Typically, it is determined by multiple trials of different network topologies. For a fully connected network as used for this study, each pair of units i and j in two consecutive layers are connected by a link with a weight $W_{i,j}$. There is an input and an output for each unit. In the input layer, the output is the same as the input for each unit. For each unit in the hidden layer, the input is comprised of the weighted sum of the units in the previous layers and the bias of the current unit. The output of each hidden layer unit is obtained by applying an activation function to its input. The ReLU activation function is applied to all units in all the hidden layers and computes the function $f(x) = \max(0, x)$. This allows for easy gradient computation, which in turn results in faster training for large networks. By feeding the training data in batches to the input layer (with a specified batch size), the DNN with a given network topology and weights can compute the predictions in the output layer.

During the training process, a dropout regularization technique is used to ignore some randomly selected neurons in order to prevent the neural networks from overfitting. Dropout rate is a parameter that needs to be tuned in deep learning. The softmax function is applied to the output layer to obtain a categorical probability distribution with values between 0 and 1, indicating the likelihood that any of the four classes are true. The highest probability determines the class label of each sample.

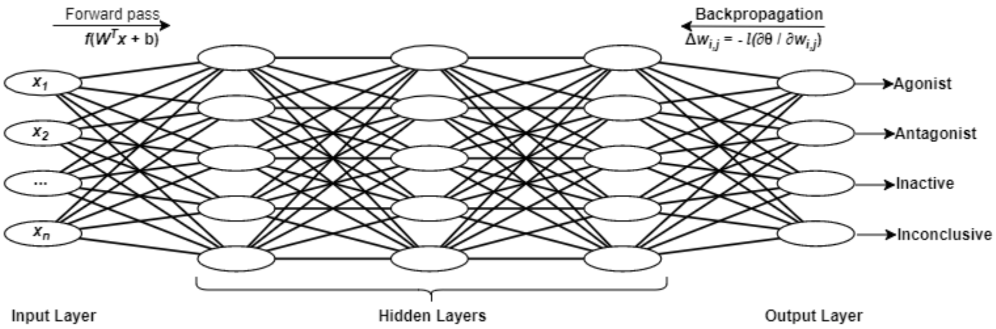


Figure 4.2 Deep Learning Architecture

4.2.3.4.2 Learning process

Training a neural network with a given architecture is a process performed to find a combination of weights of units so as to minimize the error between the predictions in the output layer and the known truth. In this study, categorical cross entropy θ is used as the loss function to compute the error. The objective function θ can be minimized by iteratively applying optimization methods such as mini-batch gradient descent, Adam, RMSprop, and Adagrad. Backpropagation is used in gradient descent methods to update the weights of units by computing the gradient $\nabla\theta$ of the loss function with respect to weight $W_{i,j}$.

The weights are updated in the opposite direction of $\nabla\theta$. The update of the weight $w_{i,j}$ is defined as $\Delta w_{i,j} = -l\frac{\partial\theta}{\partial w_{i,j}}$

where l refers to the learning rate that determines the size of the steps taken at each iteration to reach the minimum of the objective function. The weights are updated iteratively, and the learning process repeats until the neural networks are trained adequately. This means that the loss function decreases to a certain threshold.

4.2.3.4.3 Hyper-parameter optimization

The hyper-parameters in deep learning need to be tuned to get the best model suited for the dataset. These hyper-parameters include the number of hidden layers, the number of units in the input layer, the number of units in the hidden layers, the number of units in the output layer (e.g., set to 4 in this study because of the four categories of the chemical activity classification), batch size, dropout rate, learning rate and optimizer.

Bayesian hyper-parameter optimization has been shown to perform faster and more accurately than grid and random parameter search, respectively (Snoek, Larochelle, & Adams, 2012). The rationale for Bayesian optimization is to liken the optimization of hyper-parameters to a function minimization challenge. In Bayesian hyper-parameter optimization, a probability model of the objective function is constructed, which is often referred to as a surrogate function and denoted as $p(score|parameters)$. Instead of randomly selecting parameters or going through a grid in a blind manner, the results of the surrogate function are used to select the next parameters to try on the objective function, thus minimizing the number of calls to the objective function. The hyper-parameters with the best score or least validation set error computed by the objective function are considered the optimal. In this study, the search for optimal hyper-parameters was conducted using Bayesian optimization as implemented in Hyperas, a tool that combines the Keras deep learning library (Chollet, 2015) with Hyperopt's Sequential Model-Based Optimization (SMBO) methods using the Tree-structured Parzen Estimator (TPE) algorithm (Bergstra, Bardenet, Bengio, & Kégl, 2011). The search space included hidden layers {2,3,4}, Neurons {32,64,128,256,512,1024}, optimization methods {mini-batch gradient descent, Adam, RMSprop, Adagrad}, batch size {8,16,32,64,128}, and learning rate {random uniform distribution between 0 and 1}.

4.2.4 Model Evaluation Metrics

Five metrics were computed for model performance evaluation. They included precision, recall, F1-score (also called F-measure), the area under the receiver operating characteristic curve (AUROC), and the area under the precision-recall curve (AURPC). Macro-averages of the performance metrics were calculated and used for evaluation throughout this study because of the imbalanced nature of the data and the multi-category classification task. Macro-averaging independently computes the average for every class prior to averaging. By giving the same weight to all classes, it can show how effective a model is on the minority classes, e.g., AR agonists and AR antagonists that are of greater importance in this study. Micro-averaging was not considered as it gives equal weight to every sample; hence, the majority classes contribute more to the average metric than the minority classes. The following formulas describe computing the macro-averages of precision, recall and F-measure.

$$Precision_{macro} = \frac{\sum_{i=1}^{m} \frac{tp_i}{tp_i+fp_i}}{m}$$

$$Recall_{macro} = \frac{\sum_{i=1}^{m} \frac{tp_i}{tp_i+fn_i}}{m}$$

$$F\text{-}measure_{macro} = \frac{\sum_{i=1}^{m} \left(\frac{2*Precision_i*Recall_i}{Precision_i+Recall_i}\right)}{m}$$

where *m* = number of classes, *tp* = true positive, *fp* = false positive, *fn* = false negative,

The AUROC and the AUPRC were determined in Scikit-Learn (Pedregosa et al., 2011) by computing the area under the plot of true positive rate versus false positive rate and that of precision versus recall, respectively. The macro-averages of AUROC and AUPRC were calculated in a similar fashion to those of precision and recall above.

4.2.4.1 Implementation Environment

The machine learning models were developed in Python 3.5.4 using Jupyter Notebook within the Anaconda 4.3.27 (64-bit) environment. Other important libraries include Scikit-Learn 0.19.0, Keras 2.1.4, Tensorflow 1.9, and Hyperas 0.4. All models were trained on a server (Intel Xeon E5-1650) running Ubuntu 16.04.5 LTS with six cores, 32GB memory and four Nvidia Titan Xp GPUs.

4.2.5 Chemical Scaffolding and Similarity Analysis

Chemical scaffolding and similarity analysis were performed on one of the five chemical subsets used as the external test set in the first run (i.e., Fold 1 as seen in Figure 4.1). The R packages *Rcdk* and *Rcpi* were used for calculating chemical scaffolds and similarity analysis, respectively. The true labels (not predicted labels) of chemicals were used for both analyses.

In chemical scaffolding, the structural information of a chemical can be organized into rings and frameworks (Bemis & Murcko, 1996). Any cycles that share an edge are defined as rings, whereas any unions of rings via linkers are defined as frameworks. For instance, benzene, naphthalene, and anthracene are single ring systems, whereas diphenylmethane is a framework. Using Murcko chemical scaffolding, a list of rings and frameworks present in the test chemicals was generated.

The Tanimoto coefficient or scores (Bajusz, Rácz, & Héberger, 2015) are a widely accepted metric for evaluating similarity between two chemicals. Tanimoto scores were calculated using the PubChem fingerprints as the input, for every interclass pairing (e.g., an agonist vs. an antagonist, an agonist vs. an inactive, an antagonist vs. an inconclusive) in order to compare interclass similarity. The score of 0.5 was selected as the cutoff threshold, i.e., any pairs of chemicals with a score ≥ 0.5 were considered similar to each other.

4.3 Results and Discussion

4.3.1 Data distribution and evaluation metrics

As shown in Figure 4.3A, the 7665 unique compounds were unevenly distributed across four AR activity classes with the two active classes (222 compounds) being the minority (2.9%) and the inactive (2476) or inconclusive (4967) classes being the majority (97.1%).

An autoencoder was used to reduce chemical feature dimensionality. As a result, the chemical space distribution of the final set of 7665 compounds can be visualized in a 2-D plot (Figure 4.3B). The plot shows that no class forms a distinct cluster, the two inactive classes are more widely dispersed than the two active classes, and that all the active compounds reside within the space of inactive or inconclusive ones. These observations suggest that it was a challenging task to separate the four classes based on the structural features of the compounds.

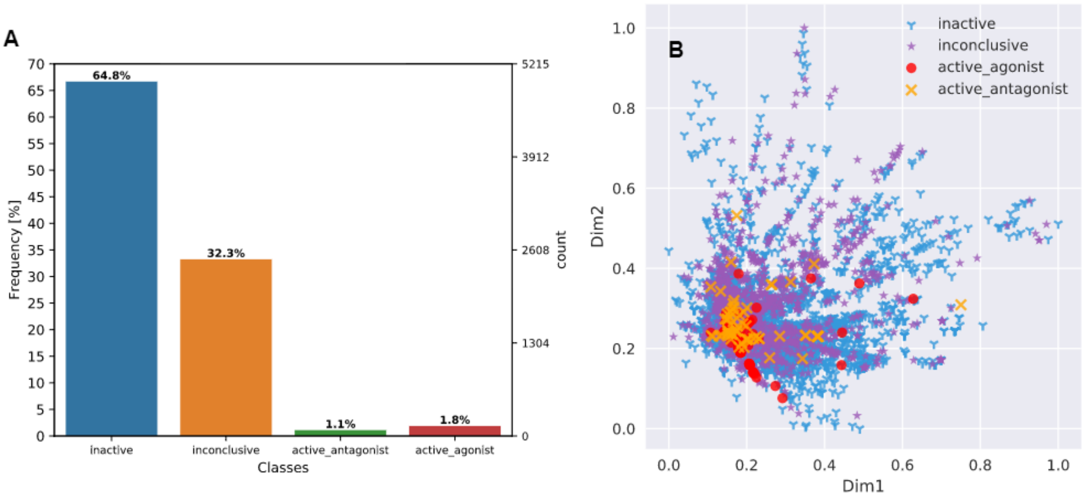


Figure 4.3 Data Distribution

Owing to the skewed class distribution, one of the main objectives was to develop a classification model with high performance for the minority classes because the two less populated active classes were of higher toxicological importance. Meanwhile, the model should not sacrifice the accuracy of the more abundant inactive and inconclusive classes, which would compromise the overall prediction performance for the entire dataset. Therefore, macro-averages was used over micro-averages (see section 4.2.4 above) and evaluation metrics that are sensitive to class imbalance or favorable to minority classes such as F-measure and AUPRC were selected (Jeni et al., 2013). F-measure is considered a better metric than precision (P) and recall (R) because it is a harmonic mean of P and R and also a tradeoff between P and R (Powers, 2011). Although AUROC and AUPRC both provide model-wide evaluation, a classifier that optimizes the area under ROC is not guaranteed to result in an optimal AUPRC (Davis & Goadrich, 2006). When the positives are the minority and more important than the negatives, AUROC is an overly optimistic measure of model performance, whereas AUPRC provides a more informative and accurate depiction of model prediction performance as it evaluates the fraction of true positives among positive predictions (Saito et al., 2015).

4.3.2 Performance Comparison between DNN and RF

Only F-measure was determined in the preliminary performance study of six machine learning algorithms without parameter optimization, and RF showed the highest F-measure with a low variance (Figure A.2). Therefore, RF was selected to represent shallow learning algorithms for further optimization as well as to compare with DNN.

Following the workflow depicted in Figure 4.1, hyper-parameters were optimized, built multi-class prediction models, and assessed the model performance. Details of the hyper-parameter optimization approach for RF and DNN are described earlier in Section 4.2.3. For DNN, it was noticed that (a) the architecture of the best performing classifier had three hidden layers with (1024,1024,512) units; (b) regularization was achieved using dropout rates of (0.25, 0.341 and 0.5) applied on these three hidden layers, respectively; and (c) Mini-Batch Gradient Descent with a batch size of 16 allowed for frequent updates in the weights of the network and a more robust convergence.

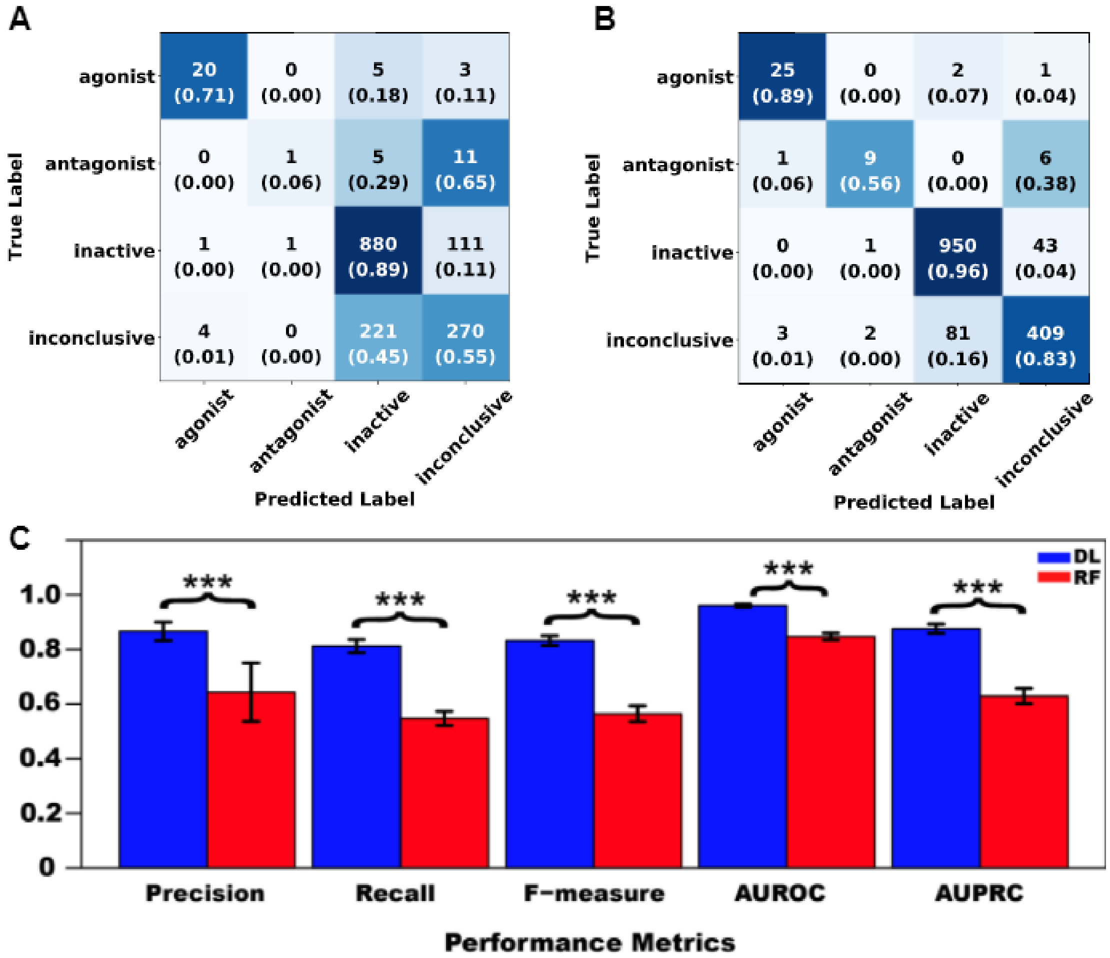


Figure 4.4 Comparison of DL versus RF Performance

Then, DNN and RF models were separately trained using the same preprocessed data. Figures 4.4A and 4.4B present the confusion matrices and the average recall scores for all four classes calculated from the external 5-fold cross-validation. Figure 4.4C provides the average performance metrics for DNN and RF side-by-side. These results clearly indicate that DNN consistently outperformed RF in both of the following measures: (1) the average number of correctly classified compounds (recall) for all four classes (Figures 4.4A and 4.4B), and (2) the macro-averages of all five performance metrics across all four classes (Figure 4.4C).

Specifically, DNN correctly predicted 50% more antagonists and 28% more inconclusive compounds than RF did, whereas the other two classes were not improved as much (i.e., 18% for agonists and 7% for inactive compounds) (Figures 4.4A and 4.4B). Furthermore, the performance enhancement was statistically significant ($p < 0.001$, ANOVA) for each metric (Figure 4.4C), regardless of whether the metric is insensitive (AUROC) or sensitive (the other four metrics) to imbalanced class distribution (Jeni et al., 2013). It is worth noting that the four imbalance-sensitive metrics were improved by 22% to 27%, while AUROC was boosted by only 11%. The coefficient of variation (CV = standard deviation/mean) for each metric was less than 5% except for the precision of RF (17%), suggesting that both DNN and RF models had stable performance. However, the performance of DNN models was more stable than that of RF and with lower error bars as seen in Figure 4.4C).

However, performance did not differ between RF and DNN prior to hyper-parameter optimization in terms of F-measure: 0.548±0.038 for RF vs. 0.536±0.052 for DNN ($p = 0.654$, paired *t*-test). Parameter optimization did not enhance RF performance (F-measure): 0.548±0.038 pre-optimization vs. 0.564±0.029 post-optimization (Figure 4.4C) ($p = 0.579$, paired *t*-test). This was due to the fact that the default parameters for RF in Scikit-Learn were not arbitrary (i.e., they are pre-optimized for normal tasks) and were similar or comparable to the selected optimal ones. On the contrary, hyper-parameter tuning greatly contributed to the improvement of DNN performance as reflected in the F-measure: 0.536±0.052 pre-optimization (Figure A.2) vs. 0.832±0.018 post-optimization (Figure 4.4C) ($p < 0.001$, paired *t*-test). In some studies (e.g., (Ambe et al., 2018; Fernandez et al., 2018)) where suboptimal performance of DL was reported in comparison with shallow learning, adequate hyper-parameter optimization was not reported. These studies along with the finding in this chapter demonstrate the dependence of DL performance on hyper-parameter optimization.

4.3.3 Chemical scaffolding analysis

Using the chemicals in Fold 1 (20% of the entire preprocessed dataset) as an example, scaffolding analysis was conducted. Class-wise Murcko decomposition revealed that the majority of chemicals contain single-ring systems and no Murcko frameworks (Figure A.3). Only 2 out of 28 agonists and 3 out of 17 antagonists contain scaffolding systems with more than one ring. These single-ring systems predominantly contain cyclopentanophenanthrene, a fused 4-membered ring system like in testosterone. About 20-30% inactive and inconclusive compounds contain systems with 2 to 4 rings (Figure A.3A). Both agonists and antagonists displayed a maximum of only 3 frameworks, whereas inactive and inconclusive compounds contained as many as 16 frameworks. This meant that the AR active compounds were more compact than the other two classes (Figure A.3B).

The obtained scaffolds (both rings and frameworks) were compared to explain the differences in prediction accuracy between different classes. The decomposed Murcko rings and frameworks revealed the total and unique chemical backbones present in each class (Table 4.1) as well as the class-specific backbones and those shared between classes (Figure 4.5). There were 8 and 3 class-specific rings identified for AR agonists and antagonists, respectively (Figure 4.5A), as well as 4 frameworks unique to these two AR active classes (Figure 4.5B).

Table 4.1 Number of total and unique Murcko rings and frameworks in the test set

	Rings		**Frameworks**	
	Total	**Unique**	**Total**	**Unique**
Agonists	30	14	4	4
Antagonists	20	9	7	6
Inactives	932	195	471	382
Inconclusives	648	167	611	497

Among the 4 agonist-specific frameworks, the 1,3-dioxole (a five-membered heterocycle consisting of two oxygen atoms at the 1 and 3 positions) and thiozetoquinoline (quinoline fused to a four-membered 1,3-thiazetidine) rings are each present in two frameworks, whereas piperazine (a six-membered ring containing two nitrogen atoms at para positions in the ring) is present in three frameworks (Figure 4.6A). A higher structural diversity is displayed in the antagonist-exclusive frameworks, including N-phenyl-azobicyclohexane-, naphthyridine-, piperidine-, and thiophene-containing frameworks, with only the structure of thiazole and piperidine connected by an ethyl linker present in two frameworks (Figure 4.6B). The 8 agonist- and 3 antagonist-specific rings are shown in Figures 4.6C and 4.6D, respectively. The low scaffold overlapping between agonists and antagonists (2 rings and 0 framework, Figures 4.5A and 4.5B) may explain why these two classes were rarely mistaken for each other during classification (Figures 4.4A and 4.4B). Furthermore, these class-specific scaffolds may serve as potential structural alerts for AR agonists or antagonists and as additional features in future machine learning-based classification or quantitative prediction modeling.

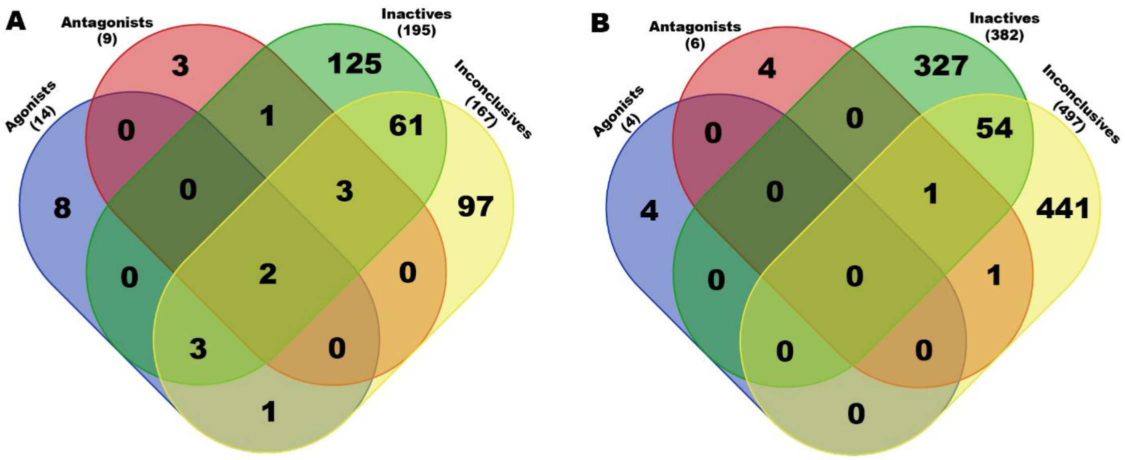


Figure 4.5 Breakdown of exclusive and shared rings (A) and frameworks (B) present in each chemical class of AR activity. Only chemicals in the Fold 1 subset (20% of the final set of preprocessed compounds) were used in this analysis. Total numbers of non-redundant scaffolds are given in parentheses

Among the four classes of chemicals, 65% (Figure 4.4A) vs. 38% (Figure 4.4B) of antagonists were misclassified as inconclusive compounds by RF and DNN, respectively; whereas 45% (Figure 4.4A) vs. 16% (Figure 4.4B) of inactive compounds were wrongly predicted to be inconclusive compounds by RF and DNN, respectively. These high rates of misclassification may be attributed to the high rates of non-redundant rings (5/9) and frameworks (2/6) present in antagonists that also appear in inconclusive compounds, and of non-redundant scaffolds (69/195 rings and 55/382 frameworks) in inactive compounds overlapping with those in inconclusive compounds (Figure 4.5). For instance, the overlapping scaffolds between antagonist and inconclusive classes include five rings (benzene, pyrazoline, thiophene, piperidine and reduced cyclopentaphenanthrene) (Figure 4.7A), and two frameworks (diphenylmethane and 4-phenylamino-piperidine) (Figure 4.7B). These overlapping scaffolds may confound the learning process in classification modeling, leading to lower prediction accuracies.

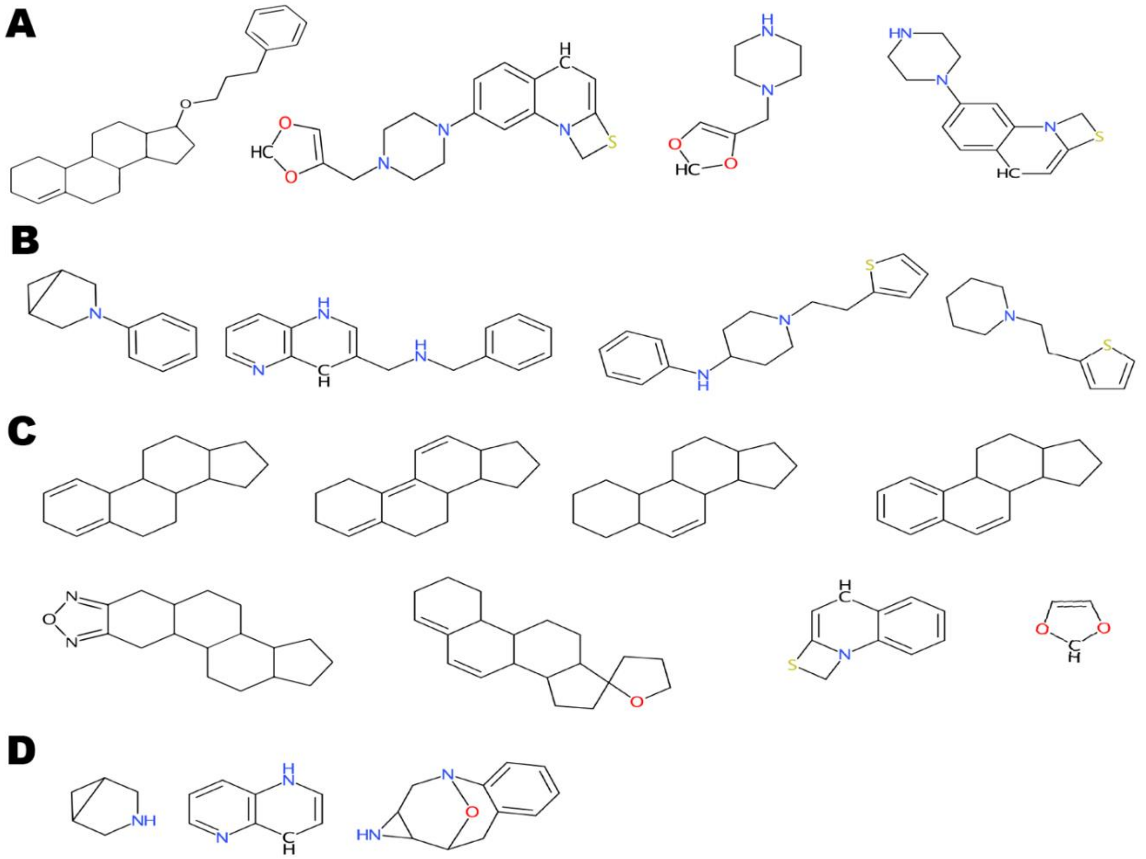


Figure 4.6 Murcko frameworks exclusively present in agonists (A) and antagonists (B) as well as Murcko rings exclusively present in agonists (C) and antagonists (D). Also see Figure 3.5 for the numbers of class-specific frameworks and rings for these two classes.

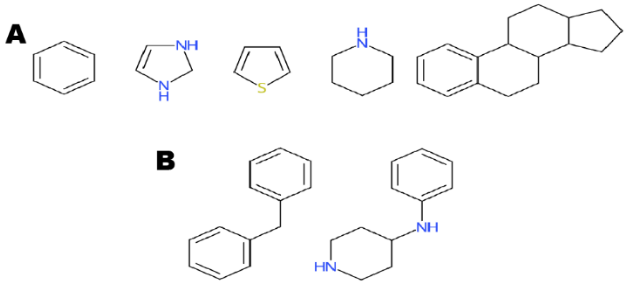


Figure 4.7 Murcko rings (A) and frameworks (B) present in both antagonists and inconclusive 782 compounds. Also see Figure 3.5 for the breakdown of scaffolds among classes.

4.3.4 Chemical similarity analysis

The Tanimoto scores (TS) determined using PubChem fingerprints have revealed the degree of chemical similarity among the four AR activity classes. For the Fold-1 subset of Tox21 compounds, five types of inter-class, pairwise chemical similarity were determined: agonist-inactive, agonist-inconclusive, antagonist-inactive, antagonist-inconclusive, and agonist-antagonist (Figure A.4). It was observed that 4.1% (=1133/(28×994)) of agonist-inactive pairs and 4.0% (=544/(496×28)) of agonist-inconclusive pairs were chemically similar (TS ≥0.5), whereas 11.9% (=1788/(17×994)) of antagonist-inactive pairs and 10.5% (=875/(17×496) of antagonist-inconclusive pairs were 50% or more similar (Table 4.2). Similar to scaffolding analysis results, the higher degree of chemical property similarity between antagonists and inconclusive or inactive compounds may have contributed to the high misclassification rates of antagonists (Figures 4.4A and 4.4B). In contrast, agonists, chemically less similar to inactive and inconclusive classes, were predicted with a much higher accuracy than antagonists (Figures 4.4A and 4.4B). The mean Tanimoto scores did not differ significantly among the four types of comparisons, likely due to an equalizing effect caused by high numbers of less similar chemical pairs (Figure A.4 and Figure A.5).

Table 4.2 Mean values of inter-class Tanimoto scores (TS) using the test set

	Inactives (994)			Inconclusives (496)		
	# true pairs with (TS =>0.5)	Mean TS	%	# true pairs with (TS =>0.5)	Mean TS	%
Agonists (28)	**1133**	**0.25 (±0.13)**	**4.1**	**544**	**0.29 (±0.13)**	**4.0**
Antagonists (17)	**1788**	**0.26 (±0.16)**	**11.9**	**875**	**0.31 (±0.17)**	**10.5**

4.4 Conclusion

Using the multi-class AR dataset from the Tox21 Data Challenge, a study was conducted that demonstrated that deep learning (represented by DNNs) was far superior to shallow learning (represented by RFs) for predicting their AR activities. The results suggest that the performance of DNN was highly dependent on hyper-parameter optimization. Meanwhile, appropriate data preprocessing (e.g., feature generation and standardization), stratified data splitting, a double-loop cross-validation strategy and performance evaluation metrics also played an important role in ensuring high quality data, avoiding over-fitting, and alleviating the impact of skewed class distribution. By performing scaffolding and similarity analyses, potential causes for antagonists being frequently misclassified as inconclusive or inactive compounds were discovered and for inactive compounds being wrongly predicted as inconclusive compounds. The high similarity in chemical properties and structural scaffolding between antagonist and inconclusive compounds and between inactive and inconclusive compounds was identified as a confounding factor that impaired classifier performance. Meanwhile, several class-specific scaffolds have been identified as candidate structural alerts for AR agonists and antagonist, which may serve as additional chemical features to improve prediction performance in future studies.

CHAPTER V – LEARNING CONTINUOUS MOLECULAR VECTOR REPRESENTATIONS USING SELF-SUPERVISED MULTI-HEAD ATTENTION MODEL

5.1 Introduction

Machine learning based Quantitative Structure–Activity/Property Relationship (QSAR) modeling plays a key role in virtual screening of chemical compounds for several purposes such as drug design, toxicological and material science studies (R. Huang & Xia, 2017b; R. Huang, Xia, Nguyen, et al., 2016; Lo et al., 2018; A. Tropsha, 2007). For drug-like substances alone, over 10^{8} chemical substances have been synthesized and as much as 10^{60} can potentially be synthesized (Irwin et al., 2012). This provides a vast field of candidates to search through. This vast search space is where *in silico* methods like QSAR thrive to narrow down promising candidates that serve as leads. Regardless of the abundance of molecules in the drug-like search space, there is still a high attrition rate as most candidates fail at different phases in the drug design process (Arrowsmith & Miller, 2013; Di Veroli, Davies, Zhang, Abi-Gerges, & Boyett, 2013; Segall & Barber, 2014). This implies the need for more accurate QSAR methods. Like any machine learning or QSAR task, the use of information loaded features plays a vital role in predictive accuracy of the model (Danishuddin & Khan, 2016; Eklund et al., 2014; Goodarzi et al., 2012; Ponzoni et al., 2017). The most relevant features are those that enhances the ease of differentiating instances of the chemical compounds into categorical classes or continuous spectrum. Benchmarking studies of the predictive performance of QSAR models have shown that the choice of molecular descriptors used is of greater importance that the statistical method used (Shao et al., 2013).

Molecular descriptors are wide ranging, each with its shortcoming. Constitutional descriptors (0D) describe the molecular composition of the compound such as molecular weight, number and type of atoms and bonds. Constitutional descriptors do not account for isomers as they do not represent conformational changes in molecules. Topological descriptors are structure-explicit descriptors calculated from the topological representation of molecules. Topological indices note the connectivity of atoms within molecules in form of a molecular graph. Typical topological indices hold information about bonds, branching, shape of molecules but it does not account for conformational information ("Molecular Descriptors," 2007; Shahlaei, 2013; Todeschini et al., 2000). Geometric descriptors are computed from the 3D coordinates of atoms in the molecule. They contain good structure and conformation information for describing molecules such as molecular size and atom distribution. However, this ability is also their setback. The complexity of geometry optimization for flexible molecules makes these descriptors extremely expensive to compute (Duan et al., 2010; Health, n.d.). Another widely used numeric representation of molecular features is fingerprints (Shahlaei, 2013). Fingerprints encode the presence or absence of substructures into a binary vector. Common types include ECFP and PubChem fingerprints. Fingerprints like ECFP tend to split molecules into several substructures and recombine into variable length bit vectors, hence models built from such bit vectors are scarcely interpretable (Rogers & Hahn, 2010).

It has been reported that using computational linguistics methods, the structural information of organic chemicals can be expressed in natural human languages like English in terms of molecular fragments and text fragments (Cadeddu, Wylie, Jurczak, Wampler-Doty, & Grzybowski, 2014; Nam & Kim, 2016). As a result, computational methods applied to corpuses of natural human language may also be applicable to the text representation of molecules. Molecules can be represented as text sequences in line notation format, such as the SMILES arbitrary target specification (SMART), IUPAC International Chemical Identifier (InChI)(O'Boyle, 2012) and the more popular simplified molecular-input line-entry system (SMILES) (Jastrzębski, Leśniak, & Czarnecki, 2016; Weininger, 1988).

Computational linguistic methods such as machine translation involves mapping an input text sequence to a target text sequence. Machine learning algorithms have become the more common way to achieve this (Nam & Kim, 2016). At the basic level, the architecture of a machine learning model for translation involves an encoder for the input text sequence which yields a set of continuous (latent) vector that serve as input to a decoder model. The decoder maps the continuous vector to the target text sequence. This architecture is similar to autoencoders. Recurrent neural networks such as BiLSTM are typically used for the encoder and decoder components because of their ability to encode sequence ("GitHub - tensorflow/nmt: TensorFlow Neural Machine Translation Tutorial," n.d.).

Based on the knowledge that human language and organic chemistry have the same structure, a possible solution to the feature generation problem may be to transform it into a text translation task. Several preceding works have explored the use of autoencoders to generate latent continuous vectors (Cadeddu et al., 2014; Gómez-Bombarelli et al., 2018; Nam & Kim, 2016; Schwaller, Gaudin, Lányi, Bekas, & Laino, 2018; Schwaller et al., 2019; Winter, Montanari, Noé, & Clevert, 2019) and de novo molecular generation (Blaschke, Olivecrona, Engkvist, Bajorath, & Chen, 2017; Segler, Kogej, Tyrchan, & Waller, 2018).

Gómez-Bombarelli et al.(Gómez-Bombarelli et al., 2018) employed a deep Variational AutoEncoder (VAE) network whose encoder and decoders composed of a blend of 1D convolutional layers and recurrent neural networks to generate continuous encoding of molecules from the latent space. The use of this continuous encoding in place of discrete representations such as fingerprints allowed the use of gradient-based optimization to search for new functional molecules and generation of new molecules via random latent vector decoding and interpolation. A separate predictor model from the VAE was used to estimate molecular properties.

Nam and Kim (Nam & Kim, 2016) first proposed a sequence–to–sequence model with hyperparameters tuned to predict the outcomes of organic chemical reactions without requiring manual encoding the rules of chemical transformations. Using a similar logic, Philippe Schwaller et al (Schwaller et al., 2018) used a similar method with LSTM variants of RNN for the encoder and decoder to translate reactants/reagents to products. Luong (Luong, Pham, & Manning, 2015) and Badhanau (Bahdanau, Cho, & Bengio, n.d.) attention mechanisms were used to compute the latent vector. This architecture outperformed the state-of-the-art results on Jin’s USPTO (Jin, Coley, Barzilay, & Jaakkola, n.d.) and Lowe’s (“Chemical reactions from US patents (1976-Sep2016),” n.d.) datasets.

Philippe Schwaller et al (Schwaller et al., 2019) also adapted a multi-head attention transformer model to their earlier work from (Schwaller et al., 2018). They claimed that the transformer model was better at accounting for subtle properties such as regioselectivity, stereoselectivity and chemo selectivity which are responsible for chemical transformation. The transformer model outperformed their earlier model which was based on RNN and gave a score to estimate its own uncertainty. This superiority of transformer models over traditional deep sequence–to–sequence models is in alignment with results obtained in human language tasks. A major reason why transformer models outperform LSTM is that they are more naturally able to capture long-term dependencies in input sequences by operating on all entities of the sequence at the same time (Vaswani et al., n.d.). This same property allows transformers to be parallelizable. LSTMs employs recurrence through backpropagation through time while transformers use attention an decode the symbol position in sequence ("The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time.," n.d.).

Autoencoders on the other hand, reconstruct the input at the output layer of the decoder using constricted latent space from the decoder (Baldi, 2012; Chandra & Sharma, 2015). Such reconstruction can result in a model that inadvertently learn the syntactic features and not so much of the semantic features that encode molecular properties. Translation instead of reconstruction is one way this challenge may be circumvented (Bjerrum, n.d.; Blaschke et al., 2018; Gómez-Bombarelli et al., 2018).

One work that focused on translating between semantically equivalent but syntactically different representations of molecules like ours is (Winter et al., 2019). The authors employed tokenized string representations of molecules such as SMILES, IUPAC and InChI (International Chemical Identifier) interchangeably as input and target. The architecture composed of a blend of both convolutional neural network (CNN) and recurrent neural network (RNN) in the encoder and decoder set up. The continuous vector from the latent space was used for modelling quantitative structure–activity relationships and the authors reported that it performed competitively and consistently in comparison to extended-connectivity fingerprints (ECFPs).

The goal of this work is to create a reliable means of generating a numerical definition vector capable of capturing a molecule's representation, referred to here as transformer embedding features (TEF). The variable-length feature vectors generated, unlike fingerprints and descriptors, do not refer to specific fragments or features of the chemical compound but should be capable of inferring chemical properties and activity of the chemical compound as required in QSAR. The method addressed in this work as highlighted in section 2 utilizes a pretrained Neural Machine Translation (NMT) technique using a transformer model, translating from SMILES structures of chemical compounds as input and the corresponding SMARTS representation as the target output. The latent vector between the input SMILES and target SMARTS may be considered as a numeric representation of the chemical compound. In section three, the suitability of the generated vectors as per structure-activity relationship modeling is assessed and compared with conventional descriptors and fingerprints. In section four, recommendations are made for future work.

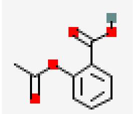
5.2 Method

5.2.1 Data Curation and Preprocessing

ChEMBL version 26 is an open large-scale bioactivity database of drug-like small molecules, manually curated from the medicinal chemistry literature. A random subset of 0.93 million unique small molecules were selected from the 2 million chemicals in ChEMBL's v26 repository of compounds in SDF format. There was no reason for the size of the selected subset beyond limitations of computational resources. As with most machine learning techniques, more data will be beneficial to the model's ability to learn.

Using the downloaded SDF data and RDKit,(Greg, n.d.) canonical SMILES were generated to ensure that each SMILES is a unique representation of the corresponding compound. These canonical SMILES are sequence of characters denoting topological properties such atoms, bonds, branches and rings. The SMILES were used as the input and SMARTS as the target output sequence. SMARTS are an extension of the SMILES notation with wildcards to specify chemical patterns such as atoms and bonds. SMARTS are mostly employed for substructure searching. They provide several primitive symbols describing atomic properties that are not used in SMILES. All SMILES expressions are also valid SMARTS expressions, but the reverse is rarely the case. This helps ensure that translation from one notation to the other is semantically and syntactically feasible.

Table 5.1 Table BB: Different representation of aspirin

2D Graph	
SMILES	CC(=O)OC1=CC=CC=C1C(=O)O
IUPAC	2-acetyloxybenzoic acid
InChI	1S/C9H8O4/c1-6(10)13-8-5-3-2-4-7(8)9(11)12/h2-5H,1H3,(H,11,12)

5.2.2 Translation Model Architecture

The multi-head self-attention architecture, also referred to as transformer as described by Vaswani et al (Vaswani et al., n.d.), was adapted for this study. Transformer follows the architecture of other state-of-the-art neural sequence transduction models that are comprised of linked encoder-decoder operations.

The encoder maps the input sequence in the form of a feature vector $X = (x_1, \dots, x_n)$ to a latent continuous vector $Z = (z_1, \dots, z_n)$. The decoder then computes the target sequence $Y = (y_1, \dots, y_m)$ one element at a time using representations of previous elements in the sequence (Schwaller et al., 2019). The encoder is composed of N stacks of identical layers, each with a multi-head self-attention mechanism and a fully connected feed-forward network with positional encoding. Each of these components of a layer is wrapped in a layer normalization operation. The decoder section of the transformer is like the encoder section. However, an additional multi-head attention component is introduced to process the incoming output of the encoder. The self-attention components in the decoder are also modified by masking and moving the output embedding by one position to the right. This auto-regressive property of decoders guarantees that the computation of the next element in a sequence at any state depends not only on the feature vector of that state but that of the previous elements in the sequence (Bahdanau et al., n.d.; Schwaller et al., 2019; Vaswani et al., n.d.).

The basic architecture of a fully connected feed-forward network (FFN) consists of several processing units called neurons combined as layers. Neurons in different layers are connected by weights (*W*) and between each layer is an activation function (σ), ReLU (Arora, Basu, Mianjy, & Mukherjee, 2016) was used in this study. The output of the feed-forward network is made of two linear operations (Vaswani et al., n.d.). This is given as:

$$FFN(x) = max(0, xW_1 + \ b_1)W_2 + \ b_2$$

$$y = \sigma(\sum W^T x)$$

At the core of a transformer is the multi-head self-attention mechanism units that replace the conventional units such as RNN or convolutional neural networks. The encoded representation of the input sequence is viewed by the transformer as a set of key-value pairs, (K, V). The key-value pairs are the hidden state of the encoder. The pair have the same dimension as the input sequence, n. The vector representing elements in the input sequence is represented as a Query, (Q).

The output of an attention unit is defined as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key (Vaswani et al., n.d.).

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Where $Q \in \mathbb{R}^{n*d_x}$, $K \in \mathbb{R}^{m*d_x}$, $V \in \mathbb{R}^{m*d_x}$

Encoded representations of input sequences mostly hold the semantics of elements in a sequence. However, the matrix vector generated by multi-head self-attention mechanism better captures the semantics as well as the internal relationship between elements in a sequence. Accuracy of machine translations are dependent on the meaning as well as the relationship between each word or elements and the others in the sequence. Instead of performing one attention operation at a time, multi-head self-attention mechanisms compute multiple scaled dot-product attention at the same time. The output from all attentions are added together, followed by a linear transformation. The simultaneous and independent computation of several scaled dot-product attention allows for parallelization and for the mechanism to handle information from different representations.

$$MultiHead(Q, K, V) = Concat[head_1, \dots, head_h]W^0$$

Where $head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$

$W_i^Q, W_i^K, W_i^V and\ W^0$ are trainable parameter matrices (Schwaller et al., 2019; Vaswani et al., n.d.).

The recurrent part of RNN-based seq-2-seq models that allows it to understand the relative position of elements in a sequence is absent in transformer models (Nam & Kim, 2016; Schwaller et al., 2018). This challenge is resolved by using a positional encoding. Positional encodings add a position-dependent trigonometric vector to the input encoding. The positional encoding is calculated from sine and cosine functions to get a vector with the same dimension as the input encoding. The addition will result in elements of the sequence being closer to each other depending on the similarity of meaning and their position in the input sequence.

$$PE_{(pos,2i)} = sin\left({}^{pos}/_{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = cos\left({}^{pos}/_{10000^{2i/d_{model}}}\right)$$

where pos is the position and i is the dimension ("GitHub - tensorflow/nmt: TensorFlow Neural Machine Translation Tutorial," n.d.; Vaswani et al., n.d.).

In designing the optimal model architecture, several parameters such as the number of attention heads, input, and inner layer dimensions and batch size were varied. ADAM optimizer (Kingma & Ba, 2014) with a varied learning rate schedule was used as suggested by (Vaswani et al., n.d.) and loss was computed as sparse categorical cross entropy. The goal of developing this model is not to merely translate between SMILES and SMARTS, hence its success is measured by the ability of the latent embeddings to improve the accuracy of the classification models and to serve on downstream QSAR tasks.

For the purpose of comparison against a baseline, Ames data set as presented in (Winter et al., 2019) was used to evaluate TEF generated by the pretrained transformer model. Morgan Fingerprints were also generated for the Ames test data using RDKit (Greg, n.d.).

5.3 Results

The goal of this proof of concept (PoC) study is to compute continuous vector representation of compounds from sequence (string) based representations. This method minimizes human specific knowledge of chemistry. It adopts a data centric approach translating between SMILES and SMARTS. As a result, the measure of performance begins with the translation quality of the NMT model. Subsequently, the suitability of the generated embedding to be used in place of fingerprints/descriptors for basic cheminformatics tasks such as similarity searches and clustering is evaluated.

5.3.1 Pretrained Translation Model Performance

The underlying assumption in assessing the performance of the NMT model is that as it gets better at translating the input SMILES sequence into the target SMARTS sequence, the better the descriptive and predictive ability of the latent embedding vector. Table 1 shows the possible range of hyperparameters that can be used to train the multi-head self-attention NMT model as well as the values used for each hyperparameter in this model. The entire range of hyperparameters were not tested for this PoC due to data and computing resource limitations.

Table 5.2 Transformer Model Hyperparameters

Hyperparameters	**Options**	**Value Used**
Number of layers	3, 4, 5, 6	4
Inner layer dimension	256, 512, 1024, 2048	512
Input/Output Dimension	64, 128, 256, 512, 1024	128
Number of Attention heads	4, 6, 8,10	8
Dropout rate	0.1, 0.15, 0.2, 0.25	0.1
Batch Size	16, 32,64,128,256	64

The parameters employed resulted in an accuracy of 82% in the translation task. This accuracy compares the translation encoding to the target encoding (ground truth) using sparse categorical accuracy. Table 2 shows specific examples of the NMT model's attempt at translating SMILES to SMART. A visual inspection shows a strong similarity between the model's output and the target. The translation maintains and to a large extent, obeys the rules for SMARTS indicating that the NMT model was able to learn both semantic and syntactic properties of the input sequence. These properties account for chemical phenomena such as valency and ionic attraction which are important for formation of compounds in the real world.

Table 5.3 Input SMILES and output SMARTS examples from the pretrained translation model

• **Input**: Cc1ccc(S(=O)(=O)N2CC3(C[C@H]2C(=O)NO)OCCO3)cc1 • **Translation**: [#6]-[#6]1:[#6]:[#6]:[#6](-[#16](=[#8])(=[#8])-[#7]2-[#6]-[#6]3-[#6](-[#6]-[#6@H](-[#8])-[#7]-[#8])-[#8]-[#6]-[#6]-[#8]-3):[#6]:[#6]:1 • **Target**: [#6]-[#6]1:[#6]:[#6]:[#6](-[#16](=[#8])(=[#8])-[#7]2-[#6]-[#6]3(-[#6]-[#6@H]-2-[#6](=[#8])-[#7]-[#8])-[#8]-[#6]-[#6]-[#8]-3):[#6]:[#6]:1
• **Input**: Cc1cc(C)n(-c2nc(-c3ccccc3)nc3c2C2CCCN2C(=O)N3c2ccccc2)n1 • **Translation**: [#6]-[#6]1:[#6]:[#6](-[#6]):[#7](-[#6]2:[#7]:[#6](-[#6]-[#6]-[#7]-2-[#6]):[#6]:[#6]:1 • **Target**: [#6]-[#6]1:[#6]:[#6](-[#6]):[#7](-[#6]2:[#7]:[#6](-[#6]3:[#6]:[#6]:[#6]:[#6]:[#6]:3):[#7]:[#6]3:[#6]:2-[#6]2-[#6]-[#6]-[#6]-[#7]-2-[#6](=[#8])-[#7]-3-[#6]2:[#6]:[#6]:[#6]:[#6]:[#6]:2):[#7]:1
• **Input**: Cc1[nH]c(C)c(C(=O)OC2CCN(CCc3ccccc3)CC2)c1C • **Translation**: [#6]-[#6]1:[#7H]:[#6](-[#6]):[#6](-[#6](=[#8])-[#8]-[#6]2-[#6]-[#6]-1-[#6]-[#6]-2):[#6]:1-[#6] • **Target**: [#6]-[#6]1:[#7H]:[#6](-[#6]):[#6](-[#6](=[#8])-[#8]-[#6]2-[#6]-[#6]-[#7](-[#6]-[#6]-[#6]3:[#6]:[#6]:[#6]:[#6]:[#6]:3)-[#6]-[#6]-2):[#6]:1-[#6]
• **Input**: O=C(NC1CCCCC1)c1ccc(N2CCCC2=O)cc1 • **Translation**: [#8]=[#6](-[#7]-[#6]1-[#6]-[#6]-[#6]-[#6]-[#6]-1)-[#6]1:[#6]:[#6]:[#6](-[#7]2=[#8]):[#6]:1 • **Target**: [#8]=[#6](-[#7]-[#6]1-[#6]-[#6]-[#6]-[#6]-[#6]-1)-[#6]1:[#6]:[#6]:[#6](-[#7]2-[#6]-[#6]-[#6]-[#6]-2=[#8]):[#6]:[#6]:1
• **Input**: CCOCCn1cc(C2CCN(CCOc3cc(Cl)ccc3C(=O)O)CC2)c2ccccc21 • **Translation**: [#6]-[#6]1:[#6]:[#6]:[#6]2:[#6](:[#7]:[#6](:[#7]:1-[#6]-[#6]1-[#6]-[#6](-[#17])-[#8])-[#6]:[#6]:3-[#6](=[#8])-[#8])-[#6]-[#6]-2):[#6]2:[#6]:[#6]:[#6]:[#6]:[#6]:1:2 • **Target**: [#6]-[#6]-[#8]-[#6]-[#6]-[#7]1:[#6]:[#6](-[#6]2-[#6]-[#6]-[#7](-[#6]-[#6]-[#8]-[#6]3:[#6]:[#6](-[#17]):[#6]:[#6]:[#6]:3-[#6](=[#8])-[#8])-[#6]-[#6]-2):[#6]2:[#6]:[#6]:[#6]:[#6]:[#6]:1:2

5.3.2 Classification Algorithm Spot-check

A quick algorithm spot-check provides an idea of the best type of machine learning algorithm that will yield the optimal predictive ability of the learned latent space embedding. Figure 5.1 shows that random forest clearly outperforms other algorithms such as logistic regression, KNN, Decision Trees, Naïve Bayes, SVM and deep neural net. This indicates that ensemble learners (extreme gradient boosted machines) within the class of random forest can exploit the informative, discriminative, and potentially independent. DNN performs comparably to random forest but was not selected based on Occam's razor as it is a more complex and less interpretable algorithm.

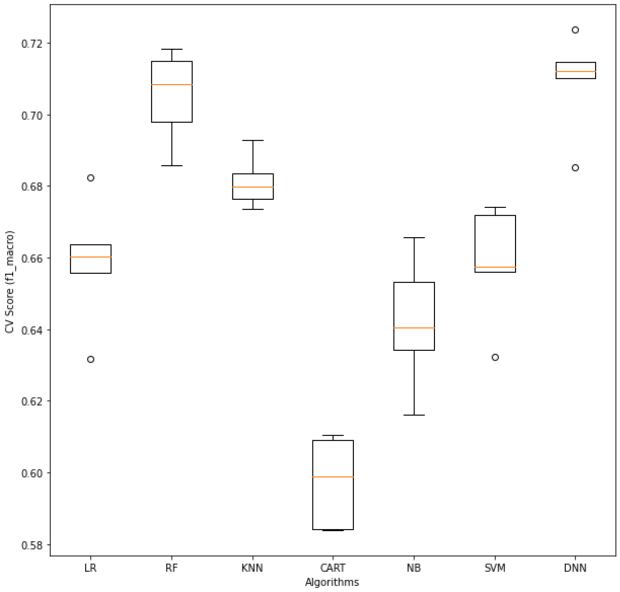


Figure 5.1 Performance comparison of classic machine learning algorithms

5.3.3 Classification Model Performance

Further optimization of the random forest model on the Ames data set as reported by [] produced an AUROC score of 0.83 on a balanced test set. This outperforms the use of circular fingerprints (0.8), graph convolutions (0.8) and RNN-based embedding (Figure 5.2). It however performs less than Canonical SMILES translation with an accuracy of over 0.95. with more data and hyperparameter tuning, latent representation of molecules from multi-head self-attention models as used in this study can perform either comparably or even better than those reported in (Winter et al., 2019).

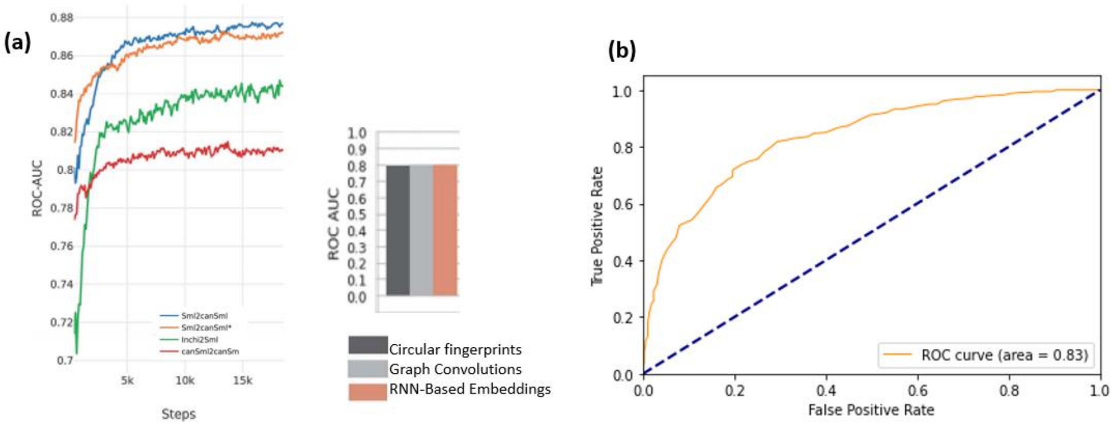


Figure 5.2 Comparison of performance of (a) baseline fingerprint and RNN-based fingerprints and (b) TEF on classification task

The comparative performance of the multi-head self-attention translation model can be attributed inherent forced learning from both the input SMILES and target SMARTS sequence. The bottleneck in translation models cannot simply encode sequence-based features or patterns in the latent space. They must learn to extract the pattern that is common in the input and the output sequence, thus increasing the chances of encoding more information of the molecule in question into the latent embedding. This is unlike autoencoder based method (Gómez-Bombarelli et al., 2018) that are forced to reconstruct the input at the output layer, hence autoencoder learn from only the input sequence.

5.3.4 Similarity and Clustering Studies

The structural similarity of ten compounds (Figure 5.3a) which are a subset of the test set was assessed using cosine similarity. Figure 5.3b shows that the similarity matrix of the transformer embedding has a similar pattern to the matrix produced using morgan fingerprints. This is indicative of the relevance of transformer embedding in conventional cheminformatics studies. The embeddings support the assumption that similar molecules are more likely to have similar structural properties and as a result, similar biological and physicochemical properties. Clustering studies show that transformer embedding (Figure 5.3b,c) produced better clusters with more delineated boundaries than morgan fingerprints shown in Figure 5d. for instance, the similarities between molecules 6 and 7 (both having double benzene rings) is better highlighted by the matrix based on the transformer embeddings.

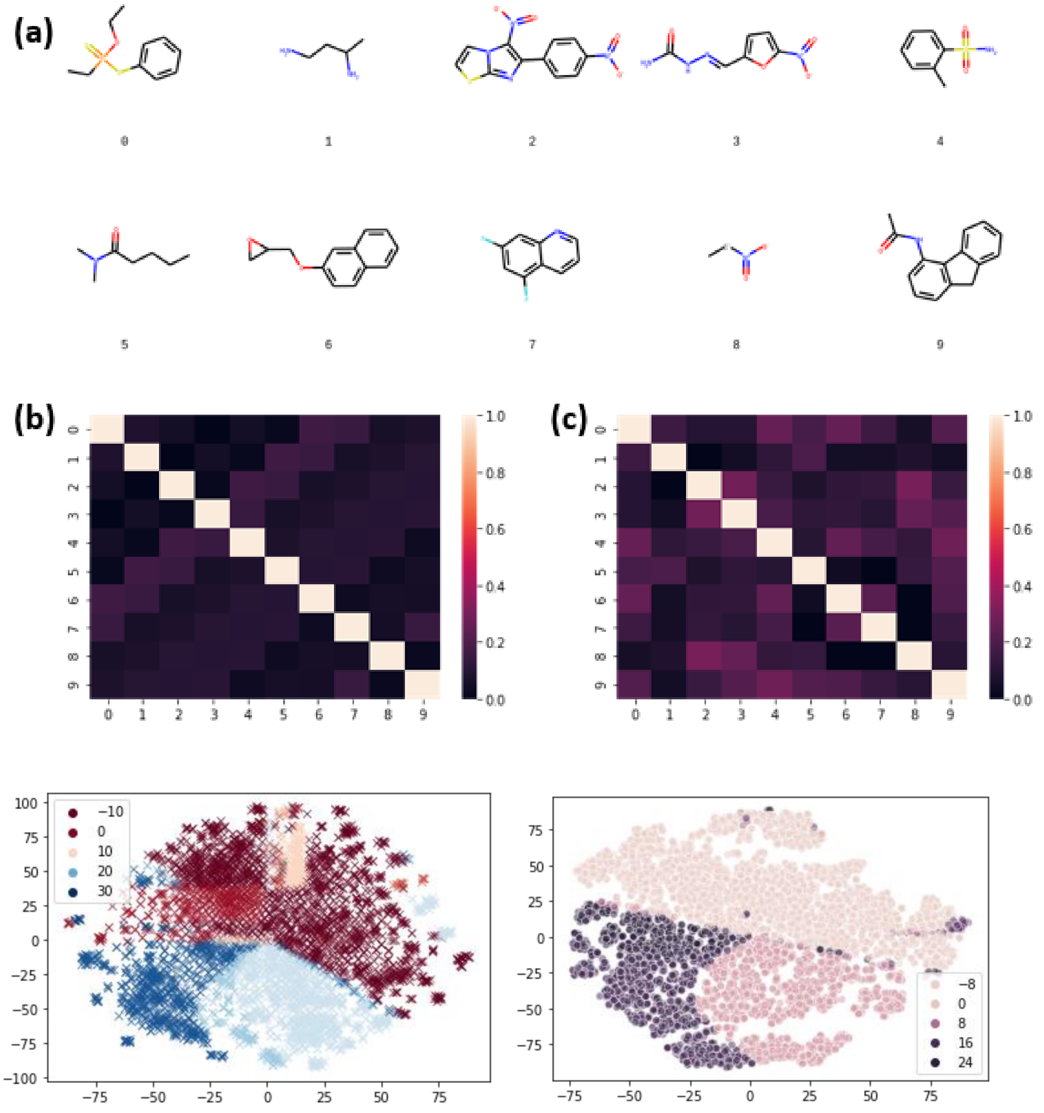


Figure 5.3 Two dimensional (2D) representation of some compounds (a), their similarity matrix using Morgan fingerprints (b) or TEF(c), and their respective clustering results ((b) and (c)).

5.4 Conclusion

This work proposes, as a proof of concept, the use of latent space embeddings from multi-head self-attention translation models. The embedding is representative of the latent space between the encoder and the decoder. The embedding is shown to perform comparably to fingerprints regardless to not extensively training the translation model. It also performs comparably to embeddings from RNN based autoencoders trained on 72 times more data than the model used in this study. This performance can be attributed to the fact the the embedding in translation models as used in this study are able to learn the underlying properties of a molecules from both the input SMILES and the target SMARTS of that molecule.

The performance of the embeddings on conventional tasks that are deemed important in cheminformatics for drug discovery, particularly in ligand-based virtual screening such as bioactivity classification, similarity search and clustering were evaluated.

Regardless of the performance shown by the embedding as features, the quality can be improved upon. For future studies, more compounds can be obtained from the ZINC database amongst others to further train and improve the quality of the embedding. Less than a million compounds were used for training the model in this study in comparison to the baseline that was trained on 72 million compounds. Further hyperparameter optimization and training barring limitations of compute resources will lead to better performing embeddings.

CHAPTER VI – SUMMARY AND PERSPECTIVES

6.1 Summary

This work focused on developing methods for improving the performance and reliability of Quantitative Structure-Activity/Property Relationship (QSAR) studies. The importance of QSAR in speeding up toxicology and drug design studies is immense. Solutions to challenges that affect the performance of QSAR studies such as class imbalance, feature dimension and relevance and selection of appropriate model complexity. *Chapter I* provides an overview of the application of machine learning algorithms to QSAR. From raw data to model validation, the importance of data quality is stressed as it greatly affects the predictive power of derived models. Commonly overlooked challenges such as data imbalance, activity cliff, model evaluation, and definition of applicability domain are highlighted, and plausible solutions for alleviating these challenges are discussed. *Chapter II* reviews current methods used for feature reduction in cheminformatics. Descriptors and fingerprints are usually of high dimension and sparse, these methods help reducing the dimension and increasing the concentration of useful information for learning properties of compounds.

The class imbalance problem is tackled in *Chapter III*. The specificity of toxicant-target biomolecule interactions lends to the very imbalanced nature of many toxicity datasets, causing poor performance in Structure-Activity Relationship (SAR)-based chemical classification. Undersampling and oversampling are representative techniques for handling such an imbalance challenge. However, removing inactive chemical compound instances from the majority class using an undersampling technique can result in information loss, whereas increasing active toxicant instances in the minority class by interpolation tends to introduce artificial minority instances that often cross into the majority class space, giving rise to over-fitting. In this study, in order to improve the prediction accuracy of imbalanced learning, SMOTEENN, a combination of Synthetic Minority Over-sampling Technique (SMOTE) and Edited Nearest Neighbor (ENN) algorithms, to oversample the minority class by creating synthetic samples, followed by cleaning the mislabeled instances. The highly imbalanced Tox21 dataset was chosen, which consisted of 12 *in vitro* bioassays for >10,000 chemicals that were distributed unevenly between binary classes. With Random Forest (RF) as the base classifier and bagging as the ensemble strategy, four hybrid learning methods were applied, i.e., RF without imbalance handling (RF), RF with Random Undersampling (RUS), RF with SMOTE (SMO), and RF with SMOTEENN (SMN). The performance of the four learning methods was compared using eight evaluation metrics, among which F_1 score, Matthews correlation coefficient and Brier score provided a more consistent assessment of the overall performance across the 12 datasets. The Friedman's aligned ranks test and the subsequent Bergmann-Hommel post hoc test showed that SMN significantly outperformed the other three methods. It was also found that a strong negative correlation existed between the prediction accuracy and the imbalance ratio (IR), which is defined as the number of inactive compounds divided by the number of active compounds. SMN became less effective when IR exceeded a certain threshold (e.g., >40). The ability to separate the few active compounds from the vast amounts of inactive ones is of great importance in computational toxicology. This work demonstrates that the performance of SAR-based, imbalanced chemical toxicity classification can be significantly improved through rebalancing the imbalanced data.

In *Chapter IV*, the application of Deep learning in improving prediction performance over classical machine learning algorithms is evaluated. Deep learning has attracted the attention of computational toxicologists as it offers a potentially greater power for *in silico* predictive toxicology than existing shallow learning algorithms. To further explore the advantages of deep learning over shallow learning, I conducted a case study using two cell-based androgen receptor (AR) activity datasets with 10K chemicals generated from the Tox21 program. A nested double-loop cross-validation approach was adopted along with a stratified sampling strategy for partitioning chemicals of multiple AR activity classes (i.e., agonist, antagonist, inactive, and inconclusive) at the same distribution rates amongst the training, validation and test subsets. Deep neural networks (DNN) and random forest (RF), representing deep and shallow learning algorithms, respectively, were chosen to carry out structure-activity relationship-based chemical toxicity prediction. Results suggest that DNN significantly outperformed RF ($p < 0.001$, ANOVA) by 22-27% for four metrics (precision, recall, F-measure, and AUPRC) and by 11% for another (AUROC). Further in-depth analyses of chemical scaffolding shed insights on structural alerts for AR agonists/antagonists and inactive/inconclusive compounds, which may aid in future drug discovery and improvement of toxicity prediction modeling. A major factor of success for deep learning is having sufficient data. While there is no science as to the amount of data required, effort should be made to curate as much balanced data as possible. Deep learning should not be the primary option except for problems such as object recognition that cannot be solved by classical algorithms.

Molecules are often represented as descriptors or as bit-vectors in the form of fingerprints. These descriptors are often very sparse and limited by the logic and mathematical processes used to compute them. In *Chapter V*, I attempt to develop new descriptors/features using the latent space embedding from a multi-head self-attention often referred to as transformer architecture. The transformer embedding features (TEF) is obtained as the continuous numeric vector in the latent space while translating between two string representations of a molecule. TEF learns its encoding of a molecule from both the input SMILES and target SMARTS representation. The significance of TEF as new descriptors was evaluated by applying them to tasks such as predictive modeling, clustering, and similarity search. An accuracy of 84% on predicting the chemicals' Ames mutagenicity test indicates that these new features have a good correlation with biological activity. TEF also showed very defined clusters on a set of mutagenic compounds. In this study, only 0.93 million unique molecules were used. Based on the results of this study, and in comparison with similar neural machine translation studies, much more data is required to achieve state-of-the-art results.

6.2 Perspectives

The challenge of learning from imbalanced data is a major concern in the field of cheminformatics. In *Chapter III* of this work, hybrid resampling techniques were applied to handling the class imbalance challenge often encountered in machine learning based SAR modeling. These techniques are based on the properties of the data in question. As shown in the findings of this work, a lot of room exist for improvement. The application of algorithmic methods that are not tied to the properties of the data being studied may offer opportunities. Algorithmic methods like cost-sensitive learning resampling techniques are likely to be less computationally intensive. For instance, XGBoost (T. Chen & Guestrin, n.d.) as an ensemble learning algorithm, uses gradient descent optimization to minimize loss a regularized (L1 and L2) objective function which is comprised of a convex loss function and a penalty term for model complexity when adding a new tree to the ensemble. Only trees that minimize the loss are added. The training proceeds of XGBoost continues iteratively, adding new trees that predict the residuals of prior trees that are then combined with previous trees to make the final prediction. In a similar vein, new samples can be selected based on prior residuals to handle imbalance. Each new model is a tree built from a subset of the entire data. Future efforts can be made to apply gradient descent in the selection of samples that are used for building each tree. Random sampling and stratified sampling are currently common in ensemble learning. By allowing the algorithm to make its own selection of samples use to train each tree using gradient descent, challenges of imbalance can be inherently dealt with.

The challenge of selecting a model with the appropriate complexity for a SAR modeling task is important to achieving good prediction performance. Newer and more complex algorithms do not necessarily translate to better prediction performance. In Chapter IV, deep learning is used to compare its performance with that of Random Forest in a classification task. This comes with a need for extensive hyperparameter tuning. The application of deep learning in machine learning based SAR has gained a lot of ground in recent times: from bioactivity classification and regression to predicting products of organic reactions. Deep Learning models are still very difficult to optimize. Regardless of such extensive use, there are less defined architectures and weights for faster training and more accurate training in SAR modeling. Fields such as computer vision (ResNet (K. He, Zhang, Ren, & Sun, n.d.) and ImageNet (Deng et al., 2010)) and Natural Language Processing (BERT (Devlin, Chang, Lee, Google, & Language, n.d.) and GPT (Openai, Openai, Openai, & Openai, n.d.)) have state-of-the-art pretrained models that can be used as a starting point for most tasks. This allows tasks in such fields to achieve commendable performance with minimal data, time restrictions or computational restraints. The availability of large amounts of data in databases such as ChEMBL, ZINC and PubChem along with increased computation resources provides an opportunity for more transfer learning in Cheminformatics. Using transfer learning, the chemical representations/information (often in the form of model weights and architecture) learned by one trained model can be applied to other models that needs to be trained on different data for either a similar or completely different task. More effort should be made in developing SAR transfer learning and pre-trained models that can boost accuracy without taking much time to converge, as compared to a model trained from scratch. This can reduce the need for extensive and resource intensive hyperparameter tuning, and in turn, better performing models with less resources.

The importance of relevant features for SAR modeling cannot be overstated. *Chapter V* provides a proof of concept for generating information rich features that are less dependent on domain experience. Although human language translation models were shown to be useful for extracting features of chemical compounds, it is worth noting that string representations of chemical compounds such as SMILES and SMARTS are different from human language. Human sentences are words. However, each SMILES or SMARTS is a long string of characters without spaces. As a result, common tokenization methods applied to human language cannot be used directly for string notations of chemical compounds. Tokenization involves splitting a sentence of paragraph into an array of words. To achieve a similar sentence structure may help SAR studies better enjoy the benefits offered by cutting edge natural language processing (NLP) techniques. Character level encoding was used in this study to account for the underlying single string format of SMILES and SMARTS. A future direction can consider splitting SMILES and SMARTS into constituent substructures, atoms, and bonds. For example: glycine represented as *[NH3+][CH2]C(=O)[O-]* can be split to resemble as sentence as *[[NH3+], [CH2], C, (=, O,), [O-]]*. This array of strings with some semblance of natural language sentences can be further used to explore more input encoding techniques from simple bag of words to more complex byte encoding and word vector encoding.

Overall, this body of work presents a set of promising data-driven solutions to challenges faced by practitioners in the field of cheminformatics such as the generation of informative features for small molecules, managing class imbalance and selecting the appropriate algorithm for machine learning tasks. At least one or all the challenges addressed are evident in every machine learning based SAR modeling exercise. The improvement in performance of these SAR models by the suggested solutions can translate to better toxicological assessment of everyday chemicals in our environment as well as reducing the cost of development and rate of attrition of drugs.

APPENDIX A

Table A.1: Sources of data for in silico toxicity modeling

Source	**URL**	**Description**
Tox21 10K	https://tripod.nih.gov/tox21	A collection of thousands of environmental chemicals and approved drugs tested for their potential to disrupt biological pathways.
ToxCast	https://www.epa.gov/chemical-research/exploring-toxcast-data-downloadable-data	
ACToR	https://actor.epa.gov/	The EPA's CompTox warehouse containing high-throughput screening, chemical exposure, sustainable chemistry, and virtual tissues data.
DSSTox	https://www.epa.gov/chemical-research/distributed-structure-searchable-toxicity-dsstox-database	A database that provides mapping of bioassay and physicochemical property data associated with chemical substances to their corresponding chemical structures.
TOXNET	https://toxnet.nlm.nih.gov/	A portal for searching several databases for toxicology related information.
ToxBank	http://www.toxbank.net/data	A repository containing protocols and experimental results to support the development of a replacement for *in vivo* repeated dose toxicity testing.
ChEMBL	https://www.ebi.ac.uk/chembldb/	A public repository of curated binding, functional, and ADMET information for a large number of drug-like bioactive compounds.
PubChem	http://pubchem.ncbi.nlm.nih.gov/	A publicly accessible platform for mining the biological information of small molecules.
eChemPortal	echemportal.org/echemportal/index.action	A chemical property data search portal.
ChemProt	http://potentia.cbs.dtu.dk/ChemProt/	A repository of 1.7 million unique compounds and biological activity information for 20,000 proteins.
BindingDB	http://www.bindingdb.org/bind/index.jsp	A database containing binding affinities of drug-like small molecules and proteins.
STITCH	http://stitch.embl.de/	A database of known and predicted interactions between chemicals and proteins.
admetSAR	http://lmmd.ecust.edu.cn/admetsar1/	A manually curated data source for diverse chemicals associated with known Absorption, Distribution, Metabolism, Excretion and Toxicity profiles.
DrugBank	http://www.drugbank.ca/	A source for combined drug (experimental and approved) and target data.
SIDER	http://sideeffects.embl.de/	A dabase containing information about approved drugs and their known adverse reactions.

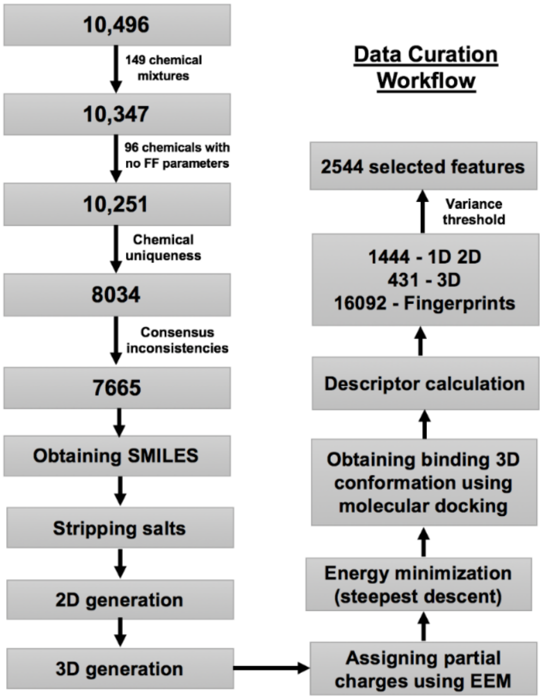


Figure A.1 Data curation workflow followed to obtain the preprocessed data to be used in DL modeling.

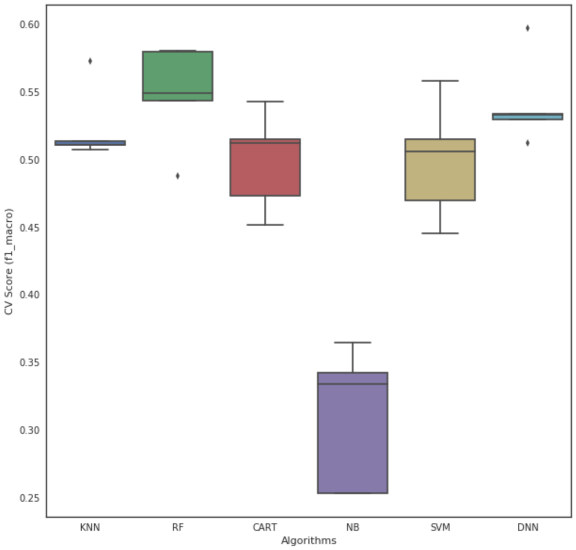


Figure A.2 Algorithm Spot check

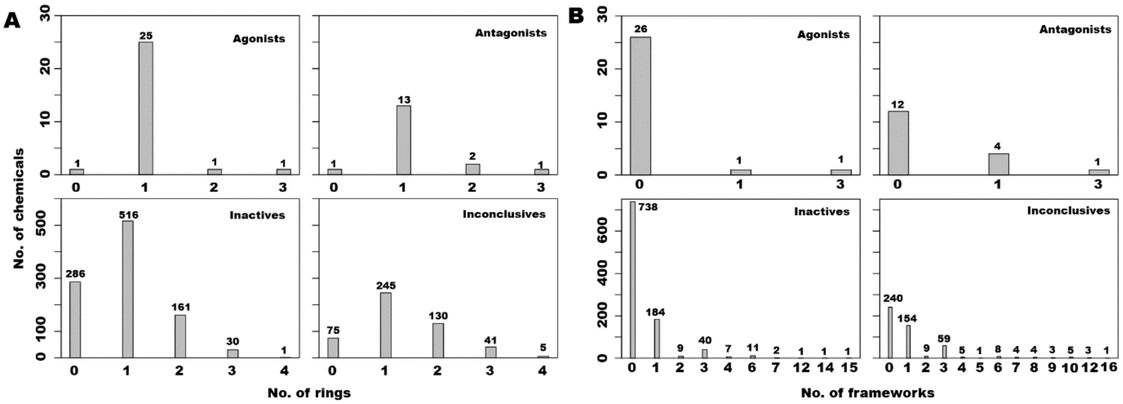


Figure A.3 Number of rings (A) and frameworks (B) present in test set.

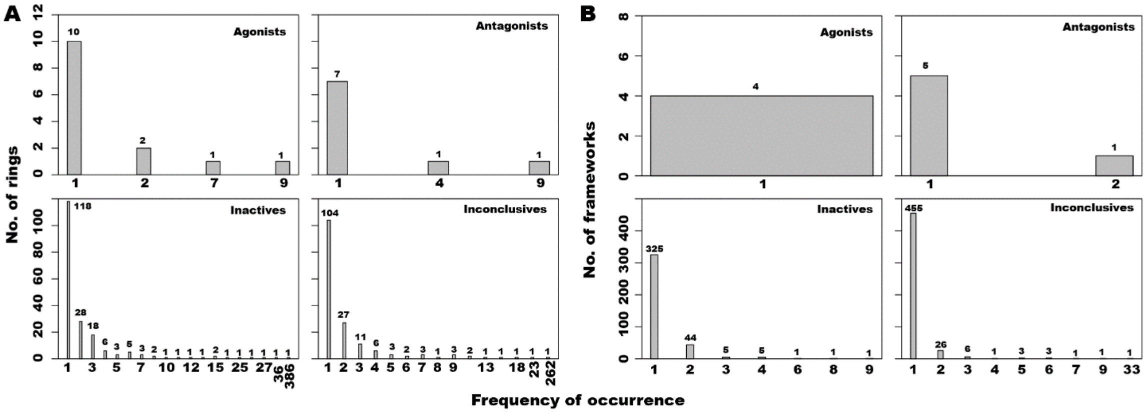


Figure A.4 Frequencies of occurrence of (A) rings and (B) frameworks present in test set.

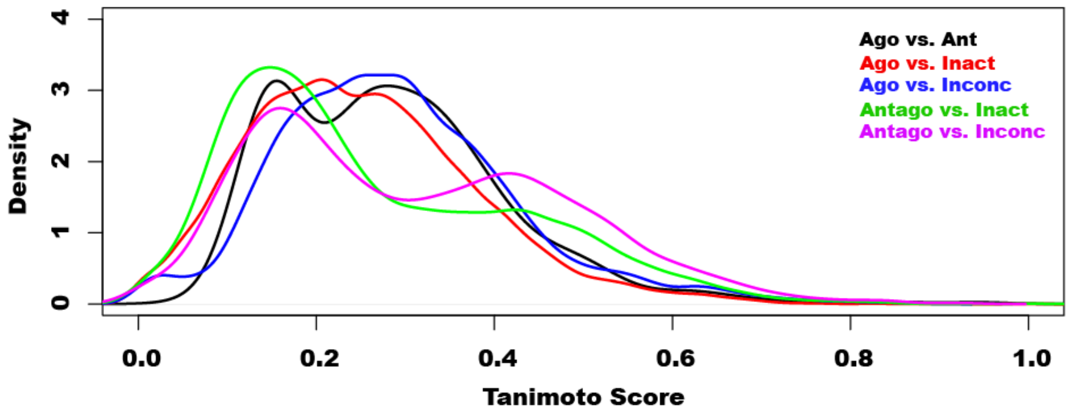


Figure A.5 Density of similar compounds present in each combination of classes using the test set. Density on the y-axis was calculated by binning the data. Number of data points in each bin was divided by the total data points and further by bin width to obtain the height of the bar along y-axis. Antagonists contain more similar chemicals in inactives (green color) and inconclusives (magenta color) compared to agonists (red and blue colors).

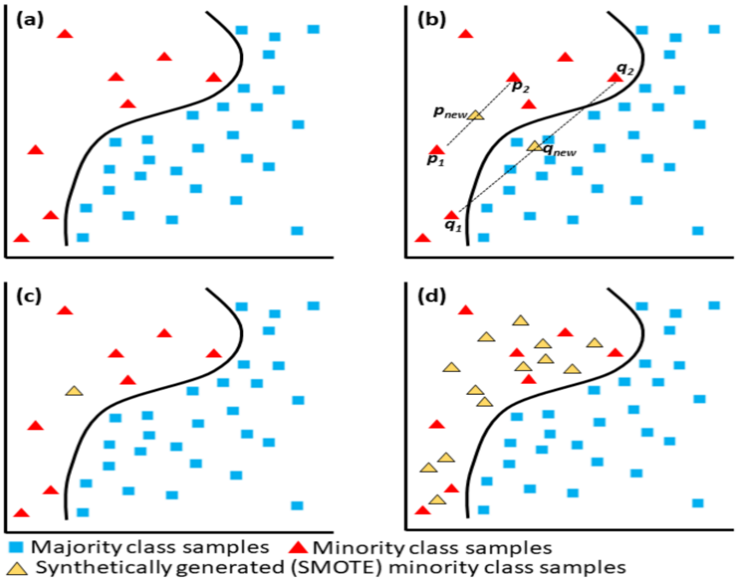


Figure A.6 Illustration of SMOTE and ENN techniques. (a) Stratified samples of imbalanced data that include minority class samples (red) and majority class samples (blue); (b) Synthetic samples (*pnew* and *qnew*) of the minority class are generated using SMOTE; (c) Retain the synthetic sample *pnew* and remove the synthetic sample *qnew* using the ENN technique. ; (d) Cleaned data with more valid synthetic minority samples to reduce the imbalance across the classes.

REFERENCES

.Abdelaziz, A., Spahn-Langguth, H., Schramm, K.-W., & Tetko, I. V. (2016). Consensus Modeling for HTS Assays Using In silico Descriptors Calculates the Best Balanced Accuracy in Tox21 Challenge. *Frontiers in Environmental Science*, *4*, 2. https://doi.org/10.3389/fenvs.2016.00002

Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., & Saeys, Y. (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, *26*(3), 392–398. https://doi.org/10.1093/bioinformatics/btp630

Alelyani, S., Liu, H., & Wang, L. (2011). The Effect of the Characteristics of the Dataset on the Selection Stability. In *2011 IEEE 23rd International Conference on Tools with Artificial Intelligence* (pp. 970–977). IEEE. https://doi.org/10.1109/ICTAI.2011.167

Allen, T. E. H., Goodman, J. M., Gutsell, S., & Russell, P. J. (2014). Defining Molecular Initiating Events in the Adverse Outcome Pathway Framework for Risk Assessment. *Chemical Research in Toxicology*, *27*(12), 2100–2112. https://doi.org/10.1021/tx500345j

Altman, N., & Krzywinski, M. (2017). Ensemble methods: bagging and random forests. *Nature Methods*, *14*(10), 933–934. https://doi.org/10.1038/nmeth.4438

Ambe, K., Ishihara, K., Ochibe, T., Ohya, K., Tamura, S., Inoue, K., … Tohkin, M. (2018). In Silico Prediction of Chemical-Induced Hepatocellular Hypertrophy Using Molecular Descriptors. *Toxicological Sciences*, *162*(2), 667–675. https://doi.org/10.1093/toxsci/kfx287

Ambroise, C., & McLachlan, G. J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(10), 6562–6566. https://doi.org/10.1073/pnas.102102699

and, R. G., & John H. Van Drie*, †. (2008). Structure−Activity Landscape Index: Identifying and Quantifying Activity Cliffs. https://doi.org/10.1021/CI7004093

Ang, J. C., Mirzal, A., Haron, H., & Hamed, H. N. A. (2016). Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *13*(5), 971–989. https://doi.org/10.1109/TCBB.2015.2478454

Angermueller, C., Pärnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Molecular Systems Biology*, *12*(7), 878.

Ankley, G. T., Bennett, R. S., Erickson, R. J., Hoff, D. J., Hornung, M. W., Johnson, R. D., … Villeneuve, D. L. (2010). Adverse outcome pathways: a conceptual framework to support ecotoxicology research and risk assessment. *Environmental Toxicology and Chemistry*, *29*(3), 730–741. https://doi.org/10.1002/etc.34

Arora, R., Basu, A., Mianjy, P., & Mukherjee, A. (2016). Understanding Deep Neural Networks with Rectified Linear Units. Retrieved from http://arxiv.org/abs/1611.01491

Arrowsmith, J., & Miller, P. (2013). Trial Watch: Phase II and Phase III attrition rates 2011–2012. *Nature Reviews Drug Discovery*, *12*(8), 569–569. https://doi.org/10.1038/nrd4090

Attene-Ramos, M. S., Miller, N., Huang, R., Michael, S., Itkin, M., Kavlock, R. J., …

Xia, M. (2013). The Tox21 robotic platform for the assessment of environmental chemicals – from vision to reality. *Drug Discovery Today*, *18*(15–16), 716–723. https://doi.org/10.1016/J.DRUDIS.2013.05.015

Bahdanau, D., Cho, K., & Bengio, Y. (n.d.). *NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE*.

Bajorath, J. (n.d.). Activity Landscapes. Retrieved from http://infochim.u-strasbg.fr/CS3_2012/Lectures/Bajorath.pdf

Bajorath, J. (2001). Selected Concepts and Investigations in Compound Classification, Molecular Descriptor Analysis, and Virtual Screening. https://doi.org/10.1021/CI0001482

Bajusz, D., Rácz, A., & Héberger, K. (2015). Why is Tanimoto index an appropriate choice for fingerprint-based similarity calculations? *Journal of Cheminformatics*, *7*(1), 20. https://doi.org/10.1186/s13321-015-0069-3

Baldi, P. (2012). Autoencoders, Unsupervised Learning, and Deep Architectures. In *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning workshop - Volume 27* (Vol. 27, pp. 37–50). JMLR.org. Retrieved from https://dl.acm.org/citation.cfm?id=3045801

Barandela, R., Sánchez, J. S., & Valdovinos, R. M. (2003). New Applications of Ensembles of Classifiers. *Pattern Anal Applic*, *6*, 245–256. https://doi.org/10.1007/s10044-003-0192-z

Barril, X. (2017). Computer-aided drug design: time to play with novel chemical matter. *Expert Opinion on Drug Discovery*, *12*(10), 977–980. https://doi.org/10.1080/17460441.2017.1362386

Barta, G. (2016). Identifying Biological Pathway Interrupting Toxins Using Multi-Tree Ensembles. *Frontiers in Environmental Science*, *4*. https://doi.org/10.3389/fenvs.2016.00052

Belkin, M., & Niyogi, P. (2003). Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, *15*(6), 1373–1396. https://doi.org/10.1162/089976603321780317

Bellman, R. (n.d.). *Adaptive control processes : a guided tour*.

Bemis, G. W., & Murcko, M. A. (1996). The Properties of Known Drugs. 1. Molecular Frameworks. *Journal of Medicinal Chemistry*, *39*(15), 2887–2893. https://doi.org/10.1021/jm9602928

Ben Brahim, A., & Limam, M. (2017). Ensemble feature selection for high dimensional data: a new method and a comparative study. *Advances in Data Analysis and Classification*, 1–16. https://doi.org/10.1007/s11634-017-0285-y

Bergmann, B., & Hommel, G. (1988). Improvements of General Multiple Test Procedures for Redundant Systems of Hypotheses (pp. 100–115). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-52307-6_8

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). *Algorithms for Hyper-Parameter Optimization*. Retrieved from https://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf

Bhhatarai, B., & Gramatica, P. (2011). Oral LD50 toxicity modeling and prediction of per- and polyfluorinated chemicals on rat and mouse. *Molecular Diversity*, *15*(2), 467–476. https://doi.org/10.1007/s11030-010-9268-z

Bjerrum, E. J. (n.d.). *Improving Chemical Autoencoder Latent Space and Molecular De*

Novo Generation Diversity with Heteroen-coders. Retrieved from https://arxiv.org/pdf/1806.09300.pdf

Blaschke, T., Olivecrona, M., Engkvist, O., Bajorath, J., & Chen, H. (2017). Application of generative autoencoder in de novo molecular design. *Molecular Informatics*, *37*(1). Retrieved from http://arxiv.org/abs/1711.07839

Blaschke, T., Olivecrona, M., Engkvist, O., Rgen Bajorath, J., & Chen, H. (2018). Application of Generative Autoencoder in De Novo Molecular Design. https://doi.org/10.1002/minf.201700123

Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLOS ONE*, *12*(6), e0177678. https://doi.org/10.1371/journal.pone.0177678

Branco, P., Torgo, L., & Ribeiro, R. (2015, May 7). A Survey of Predictive Modelling under Imbalanced Distributions. Retrieved August 8, 2017, from http://arxiv.org/abs/1505.01658

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brendel, M., Zaccarelli, R., & Devillers, L. (n.d.). *A Quick Sequential Forward Floating Feature Selection Algorithm for Emotion Detection from Speech*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.185.588&rep=rep1&type=pdf

Bultinck, P., Langenaeker, W., Lahorte, P., De Proft, F., Geerlings, P., Waroquier, M., & Tollenaere, J. P. (2002). The Electronegativity Equalization Method I: Parametrization and Validation for Atomic Charge Calculations, *106*(34), 7887–

7894. https://doi.org/10.1021/jp0205463

Burgoon, L. D. (2017). Autoencoder Predicting Estrogenic Chemical Substances (APECS): An improved approach for screening potentially estrogenic chemicals using in vitro assays and deep learning. *Computational Toxicology*, *2*, 45–49. https://doi.org/10.1016/J.COMTOX.2017.03.002

Cadeddu, A., Wylie, E. K., Jurczak, J., Wampler-Doty, M., & Grzybowski, B. A. (2014). Organic Chemistry as a Language and the Implications of Chemical Linguistics for Structural and Retrosynthetic Analyses. *Angewandte Chemie International Edition*, *53*(31), 8108–8112. https://doi.org/10.1002/anie.201403708

Cai, J., Luo, J., Wang, S., & Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, *300*, 70–79. https://doi.org/10.1016/j.neucom.2017.11.077

Calvo, B., & Santafé, G. (2016). scmamp: Statistical comparison of multiple algorithms in multiple problems. *R Journal*, *8*(1), 248–256. https://doi.org/10.32614/rj-2016-017

Capuzzi, S. J., Politi, R., Isayev, O., Farag, S., & Tropsha, A. (2016). QSAR Modeling of Tox21 Challenge Stress Response and Nuclear Receptor Signaling Toxicity Assays. *Frontiers in Environmental Science*, *4*, 3. https://doi.org/10.3389/fenvs.2016.00003

Cawley, G. C., & Talbot, N. L. C. (2010). *On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation*. *Journal of Machine Learning Research* (Vol. 11).

Chandra, B., & Sharma, R. K. (2015). Exploring autoencoders for unsupervised feature selection. In *2015 International Joint Conference on Neural Networks (IJCNN)* (pp.

1–6). IEEE. https://doi.org/10.1109/IJCNN.2015.7280391

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, *40*(1), 16–28. https://doi.org/10.1016/J.COMPELECENG.2013.11.024

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal Of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

Chawla, Nitesh V. (2005). Data Mining for Imbalanced Datasets: An Overview. In *Data Mining and Knowledge Discovery Handbook* (pp. 853–867). New York: Springer-Verlag. https://doi.org/10.1007/0-387-25465-X_40

Chawla, Nitesh V., Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving Prediction of the Minority Class in Boosting (pp. 107–119). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-39804-2_12

ChemAxon. (n.d.). Retrieved from https://chemaxon.com/

Chemical reactions from US patents (1976-Sep2016). (n.d.). Retrieved May 19, 2020, from https://figshare.com/articles/Chemical_reactions_from_US_patents_1976-Sep2016_/5104873

Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., & Blaschke, T. (2018). The rise of deep learning in drug discovery. *Drug Discovery Today*, *23*(6), 1241–1250. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S1359644617303598

Chen, J., Tang, Y. Y., Fang, B., & Guo, C. (2012). In silico prediction of toxic action mechanisms of phenols for imbalanced data with Random Forest learner. *Journal of Molecular Graphics and Modelling*, *35*, 21–27.

https://doi.org/10.1016/J.JMGM.2012.01.002

Chen, T., & Guestrin, C. (n.d.). *XGBoost: A Scalable Tree Boosting System*. Retrieved from https://github.com/dmlc/xgboost

Cherkasov, A., Muratov, E. N., Fourches, D., Varnek, A., Baskin, I. I., Cronin, M., … Tropsha, A. (2014). QSAR modeling: where have you been? Where are you going to? *Journal of Medicinal Chemistry*, *57*(12), 4977–5010. https://doi.org/10.1021/jm4004285

Chollet, F. (2015). Keras. Retrieved from https://keras.io/

Clarke, R., Ressom, H. W., Wang, A., Xuan, J., Liu, M. C., Gehan, E. A., & Wang, Y. (2008). The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature Reviews. Cancer*, *8*(1), 37–49. https://doi.org/10.1038/nrc2294

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cruz-Monteagudo, M., Medina-Franco, J. L., Pé Rez-Castillo, Y., Nicolotti, O., Natá, M., Cordeiro, D. S., & Borges, F. (2014). Activity cliffs in drug discovery: Dr Jekyll or Mr Hyde? *Drug Discovery Today*, *19*(8), 1069–1080. https://doi.org/10.1016/j.drudis.2014.02.003

Cummins, N., Baird, A., & Schuller, B. W. (2018). Speech analysis for health: Current state-of-the-art and the increasing impact of deep learning. *Methods*. https://doi.org/10.1016/j.ymeth.2018.07.007

Czarnecki, W. M., & Rataj, K. (2015). Compounds Activity Prediction in Large Imbalanced Datasets with Substructural Relations Fingerprint and EEM. In *2015*

IEEE Trustcom/BigDataSE/ISPA (pp. 192–192). Helsinki: IEEE. https://doi.org/10.1109/Trustcom.2015.581

Czarnecki, W. M., & Tabor, J. (2017). Extreme entropy machines: robust information theoretic classification. *Pattern Analysis and Applications*, *20*(2), 383–400. https://doi.org/10.1007/s10044-015-0497-8

Dahl, G. E., Jaitly, N., & Salakhutdinov, R. (2014, June 4). Multi-task Neural Networks for QSAR Predictions. Retrieved October 6, 2017, from http://arxiv.org/abs/1406.1231

Dana, D., Gadhiya, S., St. Surin, L., Li, D., Naaz, F., Ali, Q., … Narayan, P. (2018). Deep Learning in Drug Discovery and Medicine; Scratching the Surface. *Molecules*, *23*(9), 2384. https://doi.org/10.3390/molecules23092384

Danishuddin, & Khan, A. U. (2016). Descriptors and their selection methods in QSAR analysis: paradigm for drug design. *Drug Discovery Today*, *21*(8), 1291–1302. https://doi.org/10.1016/J.DRUDIS.2016.06.013

Darnag, R., Mostapha Mazouz, E. L., Schmitzer, A., Villemin, D., Jarid, A., & Cherqaoui, D. (2010). Support vector machines: development of QSAR models for predicting anti-HIV-1 activity of TIBO derivatives. *European Journal of Medicinal Chemistry*, *45*(4), 1590–1597. https://doi.org/10.1016/j.ejmech.2010.01.002

Davis, J., & Goadrich, M. (2006). The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine Learning* (pp. 233–240). Pittsburgh: ACM. Retrieved from http://pages.cs.wisc.edu/~jdavis/davisgoadrichcamera2.pdf

de Jésus-Tran Karine, P., Pierre-Luc, C., Line, C., Jonathan, B., Fernand, L., & Rock, B.

(2006). Comparison of crystal structures of human androgen receptor ligand-binding domain complexed with various agonists reveals molecular determinants responsible for binding affinity, *15*(5), 987–999. https://doi.org/10.1110/ps.051905906

Debojyoti Dutta, Rajarshi Guha, David Wild, and, & Chen, T. (2007). Ensemble Feature Selection: Consistent Descriptor Subsets for Multiple QSAR Models. https://doi.org/10.1021/CI600563W

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2010). ImageNet: A large-scale hierarchical image database (pp. 248–255). Institute of Electrical and Electronics Engineers (IEEE). https://doi.org/10.1109/cvpr.2009.5206848

Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (n.d.). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Retrieved from https://github.com/tensorflow/tensor2tensor

Di Veroli, G. Y., Davies, M. R., Zhang, H., Abi-Gerges, N., & Boyett, M. R. (2013). High-throughput screening of drug-binding dynamics to HERG improves early drug safety assessment. *American Journal of Physiology. Heart and Circulatory Physiology*, *304*(1), H104-17. https://doi.org/10.1152/ajpheart.00511.2012

Dimitrov, S., Dimitrova, G., Pavlov, T., Dimitrova, N., Patlewicz, G., Niemela, J., & Mekenyan, O. (2005). A Stepwise Approach for Defining the Applicability Domain of SAR and QSAR Models. *Journal of Chemical Information and Modeling*, *45*(4), 839–849. https://doi.org/10.1021/ci0500381

Dorfer, M., Kelz, R., & Widmer, G. (2015). Deep Linear Discriminant Analysis. Retrieved from http://arxiv.org/abs/1511.04707

Drwal, M. N., Siramshetty, V. B., Banerjee, P., Goede, A., Preissner, R., & Dunkel, M.

(2015). Molecular similarity-based predictions of the Tox21 screening outcome. *Frontiers in Environmental Science*, *3*, 54. https://doi.org/10.3389/fenvs.2015.00054

Duan, J., Dixon, S. L., Lowrie, J. F., & Sherman, W. (2010). Analysis and comparison of 2D fingerprints: Insights into database screening performance using eight fingerprint methods. *Journal of Molecular Graphics and Modelling*, *29*(2), 157–170. https://doi.org/10.1016/J.JMGM.2010.05.008

Dudek, A. Z., Arodz, T., & Gálvez, J. (2006). *Computational Methods in Developing Quantitative Structure-Activity Relationships (QSAR): A Review*. *Combinatorial Chemistry & High Throughput Screening* (Vol. 9).

Dy, J. G., & Brodley, C. E. (2004). *Feature Selection for Unsupervised Learning*. *Journal of Machine Learning Research* (Vol. 5). Retrieved from http://www.jmlr.org/papers/volume5/dy04a/dy04a.pdf

Eduati, F., Mangravite, L. M., Wang, T., Tang, H., Bare, J. C., Huang, R., … Saez-Rodriguez, J. (2015). Prediction of human population responses to toxic compounds by a collaborative competition. *Nature Biotechnology*, *33*(9), 933–940. https://doi.org/10.1038/nbt.3299

Eklund, M., Norinder, U., Boyer, S., & Carlsson, L. (2014). Choosing Feature Selection and Learning Algorithms in QSAR. *J. Chem. Inf. Model*, *54*. https://doi.org/10.1021/ci400573c

European Union. Regulation (EC) No 1907/2006 - Registration, Evaluation, Authorisation and Restriction of Chemicals (REACH) (2006).

Feng Yang, & Mao, K. Z. (2011). Robust Feature Selection for Microarray Data Based on Multicriterion Fusion. *IEEE/ACM Transactions on Computational Biology and*

Bioinformatics, *8*(4), 1080–1092. https://doi.org/10.1109/TCBB.2010.103

Fernandez, M., Ban, F., Woo, G., Hsing, M., Yamazaki, T., LeBlanc, E., … Cherkasov, A. (2018). Toxic Colors: The Use of Deep Learning for Predicting Toxicity of Compounds Merely from Their Graphic Images. *Journal of Chemical Information and Modeling*, *58*(8), 1533–1543. https://doi.org/10.1021/acs.jcim.8b00338

Ferri, C., Hernández-Orallo, J., & Modroiu, R. (2009). An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, *30*(1), 27–38. https://doi.org/10.1016/J.PATREC.2008.08.010

Fooshee, D., Mood, A., Gutman, E., Tavakoli, M., Urban, G., Liu, F., … Baldi, P. (2018). Deep learning for chemical reaction prediction. *Molecular Systems Design & Engineering*, *3*(3), 442–452. https://doi.org/10.1039/C7ME00107J

Fourches, D., Muratov, E., & Tropsha, A. (2016). Trust, but Verify II: A Practical Guide to Chemogenomics Data Curation. *Journal of Chemical Information and Modeling*, *56*(7), 1243–1252. https://doi.org/10.1021/acs.jcim.6b00129

Galar, M., Fernández, A., Barrenechea, E., Bustince, H., & Herrera, F. (2012). A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, *42*(4), 463–484. https://doi.org/10.1109/TSMCC.2011.2161285

Galar, M., Fernández, A., Barrenechea, E., & Herrera, F. (2013). EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. *Pattern Recognition*, *46*(12), 3460–3471. https://doi.org/10.1016/J.PATCOG.2013.05.006

Gao, M., Igata, H., Takeuchi, A., Sato, K., & Ikegaya, Y. (2017). Machine learning-based prediction of adverse drug effects: An example of seizure-inducing compounds.

Journal of Pharmacological Sciences, *133*(2), 70–78. https://doi.org/10.1016/j.jphs.2017.01.003

García, S., Fernández, A., Luengo, J., & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, *180*(10), 2044–2064. https://doi.org/10.1016/J.INS.2009.12.010

García, V., Sánchez, J. S., & Mollineda, R. A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. https://doi.org/10.1016/j.knosys.2011.06.013

Gawehn, E., Hiss, J. A., & Schneider, G. (2016). Deep Learning in Drug Discovery. *Molecular Informatics*, *35*(1), 3–14. https://doi.org/10.1002/minf.201501008

Geidl, S., Bouchal, T., Raček, T., Svobodová Va\vreková, R., Hejret, V., K\vrenek, A., … Koča, J. (2015). High-quality and universal empirical atomic charges for chemoinformatics applications. *Journal of Cheminformatics*, *7*(1), 59. https://doi.org/10.1186/s13321-015-0107-1

GitHub - tensorflow/nmt: TensorFlow Neural Machine Translation Tutorial. (n.d.). Retrieved May 19, 2020, from https://github.com/tensorflow/nmt

Goh, G. B., Hodas, N. O., & Vishnu, A. (2017). Deep learning for computational chemistry. *Journal of Computational Chemistry*, *38*(16), 1291–1307. https://doi.org/10.1002/jcc.24764

Goldberg, D. E. (David E., & E., D. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Longman Publishing Co., Inc. Retrieved from https://dl.acm.org/citation.cfm?id=534133

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., … Aspuru-Guzik, A. (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, *4*(2), 268–276. https://doi.org/10.1021/acscentsci.7b00572

Goodarzi, M., Dejaegher, B., & Heyden, Y. Vander. (2012). Feature selection methods in QSAR studies. *Journal of AOAC International*. https://doi.org/10.5740/jaoacint.SGE_Goodarzi

Goodfellow, Ian; Bengio, Yoshua; Courville, A. (2016). Deep Feedforward Networks. Retrieved July 31, 2018, from https://www.deeplearningbook.org/contents/mlp.html

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (n.d.). *Generative Adversarial Nets*. Retrieved from http://www.github.com/goodfeli/adversarial

Greene, N., & Pennie, W. (2015). Computational toxicology, friend or foe? *Toxicol. Res.*, *4*(5), 1159–1172. https://doi.org/10.1039/C5TX00055F

Greg, L. (n.d.). RDKit: Open-source cheminformatics Software. Retrieved from http://rdkit.org/

Guan, D., Yuan, W., Lee, Y.-K., Najeebullah, K., & Rasel, M. K. (2014). A Review of Ensemble Learning Based Feature Selection. *IETE Technical Review*, *31*(3), 190–198. https://doi.org/10.1080/02564602.2014.906859

Guha, R. (2011). The ups and downs of structure-activity landscapes. *Methods in Molecular Biology (Clifton, N.J.)*, *672*, 101–117. https://doi.org/10.1007/978-1-60761-839-3_3

Guo, G., Neagu, D., & Cronin, M. T. D. (2005). A Study on Feature Selection for

Toxicity Prediction (pp. 31–34). Springer, Berlin, Heidelberg. https://doi.org/10.1007/11540007_4

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, *3*(Mar), 1157–1182. Retrieved from http://www.jmlr.org/papers/v3/guyon03a.html

Haibo He, & Garcia, E. A. (2009). Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284. https://doi.org/10.1109/TKDE.2008.239

Han, J., Kamber, M., & Pei, J. (2011). *Data mining : concepts and techniques* (3rd ed). Elsevier Science.

Hanser, T., Barber, C., Marchaland, J. F., & Werner, S. (2016). Applicability domain: towards a more formal definition. *SAR and QSAR in Environmental Research*, *27*(11), 865–881. https://doi.org/10.1080/1062936X.2016.1250229

Hastie, T., Tibshirani, R., & Friedman, J. (n.d.). *The Elements of Statistical Learning*. Retrieved from https://web.stanford.edu/~hastie/Papers/ESLII.pdf

He, H., & Ma, Y. (2013). *Imbalanced learning : Foundations, Algorithms, and Applications*. (H. He & Y. Ma, Eds.). John Wiley & Sons, Inc.

He, K., Zhang, X., Ren, S., & Sun, J. (n.d.). *Deep Residual Learning for Image Recognition*. Retrieved from http://image-net.org/challenges/LSVRC/2015/

Health, N. I. of. (n.d.). *PubChem Substructure Fingerprint*. Retrieved from http://pubchem.

Héberger, K., & Rajkó, R. (2002). Variable selection using pair-correlation method. Environmental applications. *SAR and QSAR in Environmental Research, 13*(5), 541–554. https://doi.org/10.1080/10629360290023368

Hemmateenejad, B., Miri, R., Jafarpour, M., Tabarzad, M., & Foroumadi, A. (2006). Multiple Linear Regression and Principal Component Analysis-Based Prediction of the Anti-Tuberculosis Activity of Some 2-aryl-1,3,4-Thiadiazole Derivatives. *QSAR & Combinatorial Science*, *25*(1), 56–66. https://doi.org/10.1002/qsar.200530006

Hewitt, M., & Ellison, C. M. (2010). Developing the Applicability Domain of In Silico Models: Relevance, Importance and Methods (pp. 301–333). https://doi.org/10.1039/9781849732093-00301

Hido, S., Kashima, H., & Takahashi, Y. (2009). Roughly balanced bagging for imbalanced data. *Statistical Analysis and Data Mining*, *2*(5â□’6), 412–426. https://doi.org/10.1002/sam.10061

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, *313*(5786), 504–507. https://doi.org/10.1126/science.1127647

Hira, Z. M., & Gillies, D. F. (2015). A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data. *Advances in Bioinformatics*, *2015*, 198363. https://doi.org/10.1155/2015/198363

Hodges, J. L., & Lehmann, E. L. (2012). Rank Methods for Combination of Independent Experiments in Analysis of Variance. In *Selected Works of E. L. Lehmann* (pp. 403–418). Boston, MA: Springer US. https://doi.org/10.1007/978-1-4614-1412-4_35

Hsu, H.-H., Hsieh, C.-W., & Lu, M.-D. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, *38*(7), 8144–8150. https://doi.org/10.1016/J.ESWA.2010.12.156

Huang, R., & Xia, M. (2017a). Editorial: Tox21 Challenge to Build Predictive Models of Nuclear Receptor and Stress Response Pathways As Mediated by Exposure to Environmental Toxicants and Drugs. *Frontiers in Environmental Science*, *5*. https://doi.org/10.3389/fenvs.2017.00003

Huang, R., & Xia, M. (2017b). Editorial: Tox21 Challenge to Build Predictive Models of Nuclear Receptor and Stress Response Pathways As Mediated by Exposure to Environmental Toxicants and Drugs. *Frontiers in Environmental Science*, *5*, 3. https://doi.org/10.3389/fenvs.2017.00003

Huang, R., Xia, M., Nguyen, D.-T., Zhao, T., Sakamuru, S., Zhao, J., … Simeonov, A. (2016). Tox21Challenge to Build Predictive Models of Nuclear Receptor and Stress Response Pathways as Mediated by Exposure to Environmental Chemicals and Drugs. *Frontiers in Environmental Science*, *3*, 85. https://doi.org/10.3389/fenvs.2015.00085

Huang, R., Xia, M., Sakamuru, S., Zhao, J., Shahane, S. A., Attene-Ramos, M., … Simeonov, A. (2016). Modelling the Tox21 10 K chemical profiles for in vivo toxicity prediction and mechanism characterization. *Nature Communications*, *7*. https://doi.org/10.1038/ncomms10425

Huang, S., Li, X., Zhang, Z., He, Z., Wu, F., Liu, W., … Zhuang, Y. (2016). Deep Learning Driven Visual Path Prediction From a Single Image. *IEEE Transactions on Image Processing*, *25*(12), 5892–5904. https://doi.org/10.1109/TIP.2016.2613686

Hughes, T. B., Dang, N. Le, Miller, G. P., & Swamidass, S. J. (2016). Modeling Reactivity to Biological Macromolecules with a Deep Multitask Network. *ACS Central Science*, *2*(8), 529–537. https://doi.org/10.1021/acscentsci.6b00162

Hughes, T. B., Miller, G. P., & Swamidass, S. J. (2015). Modeling Epoxidation of Drug-like Molecules with a Deep Machine Learning Network. *ACS Central Science*, *1*(4), 168–180. https://doi.org/10.1021/acscentsci.5b00131

Hughes, T. B., & Swamidass, S. J. (2017). Deep Learning to Predict the Formation of Quinone Species in Drug Metabolism. *Chemical Research in Toxicology*, *30*(2), 642–656. https://doi.org/10.1021/acs.chemrestox.6b00385

Hyvärinen, A., Hyvärinen, A., & Oja, E. (2000). Independent component analysis: a tutorial. *NEURAL NETWORKS*, *13*, 4--5. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.87.5796

Idakwo, G., Thangapandian, S., Luttrell, J., Zhou, Z., Zhang, C., & Gong, P. (2019). Deep Learning-Based Structure-Activity Relationship Modeling for Multi-Category Toxicity Classification: A Case Study of 10K Tox21 Chemicals With High-Throughput Cell-Based Androgen Receptor Bioassay Data. *Frontiers in Physiology*, *10*, 1044. https://doi.org/10.3389/fphys.2019.01044

Indigo Toolkit. (n.d.). Retrieved from http://lifescience.opensource.epam.com/indigo/

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., & Coleman, R. G. (2012). ZINC: A Free Tool to Discover Chemistry for Biology. *Journal of Chemical Information and Modeling*, *52*(7), 1757–1768. https://doi.org/10.1021/ci3001277

Iyer, P., Stumpfe, D., Vogt, M., Bajorath, J., & Maggiora, G. M. (2013). Activity Landscapes, Information Theory, and Structure - Activity Relationships. *Molecular Informatics*, *32*(5–6), 421–430. https://doi.org/10.1002/minf.201200120

Izenman, A. J. (2012). Introduction to manifold learning. *Wiley Interdisciplinary Reviews: Computational Statistics*, *4*(5), 439–446. https://doi.org/10.1002/wics.1222

Jain, D., & Singh, V. (2018). Feature selection and classification systems for chronic disease prediction: A review. *Egyptian Informatics Journal*. https://doi.org/10.1016/J.EIJ.2018.03.002

Janecek, A. G. K., Gansterer, W. N., Demel, M. A., & Ecker, G. F. (n.d.). *On the Relationship Between Feature Selection and Classification Accuracy* (Vol. 4). Retrieved from http://proceedings.mlr.press/v4/janecek08a/janecek08a.pdf

Jastrzębski, S., Leśniak, D., & Czarnecki, W. M. (2016). Learning to SMILE(S). Retrieved from http://arxiv.org/abs/1602.06289

Jaworska, J., Nikolova-Jeliazkova, N., & Aldenberg, T. (2005). QSAR applicabilty domain estimation by projection of the training set descriptor space: a review. *Alternatives to Laboratory Animals : ATLA*, *33*(5), 445–459. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/16268757

Jeni, L. A., Cohn, J. F., & De La Torre, F. (2013). Facing Imbalanced Data--Recommendations for the Use of Performance Metrics. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction* (Vol. 2013, pp. 245–251). IEEE. https://doi.org/10.1109/ACII.2013.47

Jin, W., Coley, C. W., Barzilay, R., & Jaakkola, T. (n.d.). *Predicting Organic Reaction Outcomes with Weisfeiler-Lehman Network.*

Johnstone, I. M., & Titterington, D. M. (2009). Statistical challenges of high-dimensional data. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, *367*(1906), 4237–4253. https://doi.org/10.1098/rsta.2009.0159

Kadurin, A., Nikolenko, S., Khrabrov, K., Aliper, A., & Zhavoronkov, A. (2017).

druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico. *Molecular Pharmaceutics*, *14*(9), 3098–3104. https://doi.org/10.1021/acs.molpharmaceut.7b00346

Kalousis, A., Prados, J., & Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, *12*(1), 95–116. https://doi.org/10.1007/s10115-006-0040-8

Kavlock, R., & Dix, D. (2010). Computational Toxicology as Implemented by the U.S. EPA: Providing High Throughput Decision Support Tools for Screening and Assessing Chemical Exposure, Hazard and Risk. *Journal of Toxicology and Environmental Health, Part B*, *13*(2–4), 197–217. https://doi.org/10.1080/10937404.2010.483935

Kavlock, R. J., Ankley, G., Blancato, J., Breen, M., Conolly, R., Dix, D., … Weber, E. (2008). Computational Toxicology—A State of the Science Mini Review. *Toxicological Sciences*, *103*(1), 14–27. https://doi.org/10.1093/toxsci/kfm297

Kennedy, J., Eberhart, R., & gov, bls. (n.d.). *Particle Swarm Optimization*. Retrieved from https://www.cs.tufts.edu/comp/150GA/homeworks/hw3/_reading6 1995 particle swarming.pdf

Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2011). Comparing Boosting and Bagging Techniques With Noisy and Imbalanced Data. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *41*(3), 552–568. https://doi.org/10.1109/TSMCA.2010.2084081

Kim, S., Thiessen, P. A., Cheng, T., Yu, B., & Bolton, E. E. (2018). An update on PUG-

REST: RESTful interface for programmatic access to PubChem. *Nucleic Acids Research*, *46*(W1), W563–W570.

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. Retrieved from http://arxiv.org/abs/1412.6980

Klepsch, F., Vasanthanathan, P., & Ecker, G. F. (2014). Ligand and Structure-Based Classification Models for Prediction of P-Glycoprotein Inhibitors. *Journal of Chemical Information and Modeling*, *54*(1), 218–229. https://doi.org/10.1021/ci400289j

KNIME. (n.d.). Retrieved July 30, 2018, from https://www.knime.com/

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324. https://doi.org/10.1016/S0004-3702(97)00043-X

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, *43*(1), 59–69. https://doi.org/10.1007/BF00337288

Koutsoukas, A., Monaghan, K. J., Li, X., & Huan, J. (2017). Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of Cheminformatics*, *9*(1), 42. https://doi.org/10.1186/s13321-017-0226-y

Krawczyk, B., & Krawczyk, B. B. (2016). Learning from imbalanced data: open challenges and future directions. *Prog Artif Intell*, *5*, 221–232. https://doi.org/10.1007/s13748-016-0094-0

Kruhlak, N. L., Benz, R. D., Zhou, H., & Colatsky, T. J. (2012). (Q)SAR Modeling and Safety Assessment in Regulatory Review. *Clinical Pharmacology & Therapeutics*, *91*(3), 529–534. https://doi.org/10.1038/clpt.2011.300

Kuncheva, L. I., & Whitaker, C. J. (2003). Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning*, *51*(2), 181–207. Retrieved from https://link.springer.com/content/pdf/10.1023/A:1022859003006.pdf

Kwasnicka, H., Michalak, K., Kwa´snicka, H., & Kwa´snicka, K. (2006). *Correlation-based feature selection strategy in classification problems*. *Article in International Journal of Applied Mathematics and Computer Science* (Vol. 16). Retrieved from https://www.researchgate.net/publication/230856547

Laszczyski, J., Stefanowski, J., & Idkowiak, L. (2013). Extending Bagging for Imbalanced Data. In *Burduk R., Jackowski K., Kurzynski M., Wozniak M., Zolnierek A. (eds) Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013. Advances in Intelligent Systems and Computing* (pp. 269–278). Heidelberg: Springer. https://doi.org/https://doi.org/10.1007/978-3-319-00969-8_26

Lauria, A., Ippolito, M., & Almerico, A. M. (2009). Combined Use of PCA and QSAR/QSPR to Predict the Drugs Mechanism of Action. An Application to the NCI ACAM Database. *QSAR & Combinatorial Science*, *28*(4), 387–395. https://doi.org/10.1002/qsar.200810062

Lavecchia, A. (2015). Machine-learning approaches in drug discovery: methods and applications. *Drug Discovery Today*, *20*(3), 318–331. https://doi.org/10.1016/j.drudis.2014.10.012

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

LeCun, Y., Bengio, Y., Hinton, G., Y., L., Y., B., & G., H. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lei, S. (2012). A Feature Selection Method Based on Information Gain and Genetic Algorithm. In *2012 International Conference on Computer Science and Electronics Engineering* (pp. 355–358). Hangzhou: IEEE. https://doi.org/10.1109/ICCSEE.2012.97

Lei, T., Sun, H., Kang, Y., Zhu, F., Liu, H., Zhou, W., … Hou, T. (2017). ADMET Evaluation in Drug Discovery. 18. Reliable Prediction of Chemical-Induced Urinary Tract Toxicity by Boosting Machine Learning Approaches. *Molecular Pharmaceutics*, *14*(11), 3935–3953. https://doi.org/10.1021/acs.molpharmaceut.7b00631

Lemaˆıtre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*. Retrieved from http://jmlr.org/papers/v18/16-365.html.

Lenselink, E. B., ten Dijke, N., Bongers, B., Papadatos, G., van Vlijmen, H. W. T., Kowalczyk, W., … van Westen, G. J. P. (2017). Beyond the hype: deep neural networks outperform established methods using a ChEMBL bioactivity benchmark set. *Journal of Cheminformatics*, *9*(1), 45. https://doi.org/10.1186/s13321-017-0232-0

Li, Yan; Idakwo, Gabriel; Thangapandian, Sundar; Chen, Minjun; Hong, Huixiao; Zhang, Chaoyang; Gong, P. (2018). Target-specific Toxicity Knowledgebase (TsTKb): A novel toolkit for in silico predictive toxicology. *Journal of Environmental Science* *and Health, Part C - Environmental Carcinogenesis and Ecotoxicology Reviews*, *36*(4), 21–36.

Ling Xue, Jeff Godden, Hua Gao, A., & Bajorath, J. (1999). Identification of a Preferred Set of Molecular Descriptors for Compound Classification Based on Principal Component Analysis. https://doi.org/10.1021/CI980231D

Liu, R., Madore, M., Glover, K. P., Feasel, M. G., & Wallqvist, A. (2018). Assessing Deep and Shallow Learning Methods for Quantitative Prediction of Acute Chemical Toxicity. *Toxicological Sciences : An Official Journal of the Society of Toxicology*, *164*(2), 512–526. https://doi.org/10.1093/toxsci/kfy111

Liu, Y. X., Zhang, N. N., He, Y., & Lun, L. J. (2015). Prediction of core cancer genes using a hybrid of feature selection and machine learning methods. *Genetics and Molecular Research*, *14*(3), 8871–8882. https://doi.org/10.4238/2015.August.3.10

Lo, Y.-C., Rensi, S. E., Torng, W., & Altman, R. B. (2018). Machine learning in chemoinformatics and drug discovery. *Drug Discovery Today*, *23*(8), 1538–1546. https://doi.org/10.1016/J.DRUDIS.2018.05.010

Lobo, J. M., Jiménez-Valverde, A., & Real, R. (2008). AUC: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography*, *17*(2), 145–151. https://doi.org/10.1111/j.1466-8238.2007.00358.x

Luong, M.-T., Pham, H., & Manning, C. D. (2015). *Effective Approaches to Attention-based Neural Machine Translation*. Retrieved from http://nlp.stanford.edu/projects/nmt.

Maaten, L. van der, & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, *9*(Nov), 2579–2605. Retrieved from http://jmlr.org/papers/v9/vandermaaten08a.html

Maggiora, G. M. (2006). On Outliers and Activity CliffsWhy QSAR Often Disappoints. https://doi.org/10.1021/CI060117S

Manikandan, G., & Abirami, S. (2018). A Survey on Feature Selection and Extraction Techniques for High-Dimensional Microarray Datasets. In *Knowledge Computing and its Applications* (pp. 311–333). Singapore: Springer Singapore. https://doi.org/10.1007/978-981-10-8258-0_14

Mansouri, K., Grulke, C. M., Richard, A. M., Judson, R. S., & Williams, A. J. (2016). An automated curation procedure for addressing chemical errors and inconsistencies in public datasets used in QSAR modelling An automated curation procedure for addressing chemical errors and inconsistencies in public datasets used in QSAR modelling $. *SAR and QSAR in Environmental Researchl Research*, *27*(11), 911–937. https://doi.org/10.1080/1062936X.2016.1253611org/10.1080/1062936X.2016.1253611doi.org/10.1080/1062936X.2016.1253611

Martin, Y. C. (2009). Let's not forget tautomers. *Journal of Computer-Aided Molecular Design*, *23*(10), 693–704. https://doi.org/10.1007/s10822-009-9303-2

Mayr, A., Klambauer, G., Unterthiner, T., Hochreiter, S., Mayr, A., Klambauer, G., … Hochreiter, S. (2016). DeepTox: Toxicity Prediction using Deep Learning. *Frontiers in Environmental Science*, *3*(80), 1–15. https://doi.org/10.3389/fenvs.2015.00080

Mayr, A., Klambauer, G., Unterthiner, T., Steijaert, M., Wegner, J. K., Ceulemans, H., … Hochreiter, S. (2018). Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical Science*, *9*(24), 5441–5451.

https://doi.org/10.1039/c8sc00148k

Merkwirth, C., Mauser, H., Schulz-Gasch, T., Roche, O., Martin Stahl, A., & Lengauer, T. (2004). Ensemble Methods for Classification in Cheminformatics. https://doi.org/10.1021/CI049850E

Min, S., Lee, B., & Yoon, S. (2016). Deep learning in bioinformatics. *Briefings in Bioinformatics*, bbw068. https://doi.org/10.1093/bib/bbw068

Modern Multidimensional Scaling. (2005). New York, NY: Springer New York. https://doi.org/10.1007/0-387-28981-X

Molecular Descriptors. (2007). In *An Introduction To Chemoinformatics* (pp. 53–74). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-1-4020-6291-9_3

MolVS: Molecule Validation and Standardization — MolVS 0.0.9 documentation. (n.d.). Retrieved February 6, 2018, from https://molvs.readthedocs.io/en/latest/

Nam, J., & Kim, J. (2016). Linking the Neural Machine Translation and the Prediction of Organic Chemistry Reactions. Retrieved from http://arxiv.org/abs/1612.09529

National Research Council. (2007). *Toxicity Testing in the 21st Century: A Vision and A Strategy*. (N. R. Council, Ed.). Washington, D.C.: National Academies Press. https://doi.org/10.17226/11970

NCATS. (n.d.). Toxicology in the 21st Century (Tox21). Retrieved May 11, 2017, from https://ncats.nih.gov/tox21

Netzeva, T. I., Worth, A., Aldenberg, T., Benigni, R., Cronin, M. T. D., Gramatica, P., … Yang, C. (2005). Current status of methods for defining the applicability domain of (quantitative) structure-activity relationships. The report and recommendations of ECVAM Workshop 52. *Alternatives to Laboratory Animals : ATLA*, *33*(2), 155–

173. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/16180989

Névéol, A., Zweigenbaum, P., & Section Editors for the IMIA Yearbook Section on Clinical Natural Language Processing. (2018). Expanding the Diversity of Texts and Applications: Findings from the Section on Clinical Natural Language Processing of the International Medical Informatics Association Yearbook. *Yearbook of Medical Informatics*, *27*(01), 193–198. https://doi.org/10.1055/s-0038-1667080

Newby, D., Freitas, A. A., & Ghafourian, T. (2013). Pre-processing Feature Selection for Improved C&amp;RT Models for Oral Absorption. *Journal of Chemical Information and Modeling*, *53*(10), 2730–2742. https://doi.org/10.1021/ci400378j

O'Boyle, N. M. (2012). Towards a Universal SMILES representation - A standard method to generate canonical SMILES based on the InChI. *Journal of Cheminformatics*, *4*(1), 22. https://doi.org/10.1186/1758-2946-4-22

O'Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., & Hutchison, G. R. (2011). Open Babel: An open chemical toolbox. *Journal of Cheminformatics*, *3*(1), 33. https://doi.org/10.1186/1758-2946-3-33

OECD (Organization for Economic Co-operation and Development). (2013). *Guidance document on developing and assessing adverse outcome pathways. OECD environment, health and safety publications - series on testing and assessment, No. 184.* Paris, France.

Openai, A. R., Openai, K. N., Openai, T. S., & Openai, I. S. (n.d.). *Improving Language Understanding by Generative Pre-Training*. Retrieved from https://gluebenchmark.com/leaderboard

OpenEye. (n.d.). Retrieved from https://www.eyesopen.com/

Osman, H., Ghafari, M., & Nierstrasz, O. (2017). Automatic feature selection by regularization to improve bug prediction accuracy. In *2017 IEEE Workshop on Machine Learning Techniques for Software Quality Evaluation (MaLTeSQuE)* (pp. 27–32). IEEE. https://doi.org/10.1109/MALTESQUE.2017.7882013

Patlewicz, G., Ball, N., Becker, R. A., Booth, E. D., Cronin, M. T. D., Kroese, D., … Hartung, T. (n.d.). Food for Thought … Read-Across Approaches-Misconceptions, Promises and Challenges Ahead. *Altex*, *31*, 4–14. https://doi.org/10.14573/altex.1410071

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830. Retrieved from http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Peltason, L., & Bajorath, J. (2007). SAR Index: Quantifying the Nature of Structure−Activity Relationships. *Journal of Medicinal Chemistry*, *50*(23), 5571–5578. https://doi.org/10.1021/jm0705713

Perez-Riverol, Y., Kuhn, M., Vizcaíno, J. A., Hitz, M.-P., & Audain, E. (2017). Accurate and fast feature selection workflow for high-dimensional omics data. *PLOS ONE*, *12*(12), e0189875. https://doi.org/10.1371/journal.pone.0189875

Perkins, R., Fang, H., Tong, W., & Welsh, W. J. (2003). Quantitative structure-activity relationship methods: perspectives on drug discovery and toxicology. *Environmental Toxicology and Chemistry*, *22*(8), 1666–1679. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/12924569

Pham-The, H., Casañola-Martin, G., Garrigues, T., Bermejo, M., González-Álvarez, I., Nguyen-Hai, N., … Le-Thi-Thu, H. (2016). Exploring different strategies for imbalanced ADME data problem: case study on Caco-2 permeability modeling. *Molecular Diversity*, *20*(1), 93–109. https://doi.org/10.1007/s11030-015-9649-4

Plewczynski, D., Spieser, S. A. H., & Koch, U. (2006). Assessing Different Classification Methods for Virtual Screening. *Journal of Chemical Information and Modeling*, *46*(3), 1098–1106. https://doi.org/10.1021/ci050519k

Polishchuk, P. G., Muratov, E. N., Artemenko, A. G., Kolumbin, O. G., Muratov, N. N., & Kuz'min, V. E. (2009). Application of Random Forest Approach to QSAR Prediction of Aquatic Toxicity. *Journal of Chemical Information and Modeling*, *49*(11), 2481–2488. https://doi.org/10.1021/ci900203n

Ponzoni, I., Sebastián-Pérez, V., Requena-Triguero, C., Roca, C., Martínez, M. J., Cravero, F., … Campillo, N. E. (2017). Hybridizing Feature Selection and Feature Learning Approaches in QSAR Modeling for Drug Discovery. *Scientific Reports*, *7*(1), 2403. https://doi.org/10.1038/s41598-017-02114-3

Powers, D. M. W. (2011). EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The Case Against Accuracy Estimation for Comparing Induction Algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 445–453). San Francisco: Morgan Kaufmann Publishers Inc. Retrieved from https://dl.acm.org/citation.cfm?id=657469

Pudil, P., Novovičová, J., & Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, *15*(11), 1119–1125.

https://doi.org/10.1016/0167-8655(94)90127-9

Putin, E., Asadulaev, A., Vanhaelen, Q., Ivanenkov, Y., Aladinskaya, A. V, Aliper, A., & Zhavoronkov, A. (2018). Adversarial Threshold Neural Computer for Molecular de Novo Design. *Molecular Pharmaceutics*, *15*(10), 4386‒4397. https://doi.org/10.1021/acs.molpharmaceut.7b01137

Raies, A. B., & Bajic, V. B. (2016). In silico toxicology: computational methods for the prediction of chemical toxicity. *WIREs Comput Mol Sci*, *6*, 147–172. https://doi.org/10.1002/wcms.1240

Rajarshi, G., & Jurs, P. C. (2004). Development of Linear, Ensemble, and Nonlinear Models for the Prediction and Interpretation of the Biological Activity of a Set of PDGFR Inhibitors. https://doi.org/10.1021/CI049849F

Ramsundar, B., Kearnes, S., Riley, P., Webster, D., Konerding, D., Pande, V., & Edu, P. Massively Multitask Networks for Drug Discovery (2015). Retrieved from https://tripod.

Rao, H., Yang, G., Tan, N., Li, P., Li, Z., & Li, X. (2009). Prediction of HIV-1 Protease Inhibitors Using Machine Learning Approaches. *QSAR & Combinatorial Science*, *28*(11â□’12), 1346–1357. https://doi.org/10.1002/qsar.200960021

Reddy, A. S., Kumar, S., & Garg, R. (2010). Hybrid-genetic algorithm based descriptor optimization and QSAR models for predicting the biological activity of Tipranavir analogs for HIV protease inhibition. *Journal of Molecular Graphics & Modelling*, *28*(8), 852–862. https://doi.org/10.1016/j.jmgm.2010.03.005

Ren, Y. Y., Zhou, L. C., Yang, L., Liu, P. Y., Zhao, B. W., & Liu, H. X. (2016). Predicting the aquatic toxicity mode of action using logistic regression and linear discriminant analysis. *SAR and QSAR in Environmental Research*, *27*(9), 721–746. https://doi.org/10.1080/1062936X.2016.1229691

Revathy, N., Revathy, N., & Balasubramanian, D. R. (n.d.). GA-SVM WRAPPER APPROACH FOR GENE RANKING AND CLASSIFICATION USING EXPRESSIONS OF VERY FEW GENES. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.299.7281

Reverter, F., Vegas, E., & Oller, J. M. (2014). Kernel-PCA data integration with enhanced interpretability. *BMC Systems Biology*, *8*(Suppl 2), S6. https://doi.org/10.1186/1752-0509-8-S2-S6

Ribay, K., Kim, M. T., Wang, W., Pinolini, D., & Zhu, H. (2016). Predictive Modeling of Estrogen Receptor Binding Agents Using Advanced Cheminformatics Tools and Massive Public Data. *Frontiers in Environmental Science*, *4*, 12. https://doi.org/10.3389/fenvs.2016.00012

Ringnér, M. (2008). What is principal component analysis? *Nature Biotechnology*, *26*(3), 303–304. https://doi.org/10.1038/nbt0308-303

Rogers, D., & Hahn, M. (2010). Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, *50*(5), 742–754. https://doi.org/10.1021/ci100050t

Roy, K., Kar, S., & Ambure, P. (2015). On a simple approach for determining applicability domain of QSAR models. *Chemometrics and Intelligent Laboratory Systems*, *145*, 22–29. https://doi.org/10.1016/J.CHEMOLAB.2015.04.013

Roy, K., Kar, S., Das, R. N., Roy, K., Kar, S., & Das, R. N. (2015). Validation of QSAR Models. In *Understanding the Basics of QSAR for Applications in Pharmaceutical*

Sciences and Risk Assessment (pp. 231–289). Elsevier. https://doi.org/10.1016/B978-0-12-801505-6.00007-7

Sahigara, F., Mansouri, K., Ballabio, D., Mauri, A., Consonni, V., & Todeschini, R. (2012). Comparison of Different Approaches to Define the Applicability Domain of QSAR Models. *Molecules*, *17*(5), 4791–4810. https://doi.org/10.3390/molecules17054791

Saito, T., Rehmsmeier, M., Hood, L., Franco, O., Pereira, R., & Wang, K. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, *10*(3), e0118432. https://doi.org/10.1371/journal.pone.0118432

Sakkiah, S., Selvaraj, C., Gong, P., Zhang, C., Tong, W., Hong, H., … Hong, H. (2017). Development of estrogen receptor beta binding prediction model using large sets of chemicals. *Oncotarget*, *8*(54), 92989–93000. https://doi.org/10.18632/oncotarget.21723

Schneider, G. (2018). Generative Models for Artificially-intelligent Molecular Design. *Molecular Informatics*, *37*(1–2), 1880131. https://doi.org/10.1002/minf.201880131

Schölkopf, B., & Smola, A. J. (2001). *Learning with Kernels*. Retrieved from https://www.cs.utah.edu/~piyush/teaching/learning-with-kernels.pdf

Schwaller, P., Gaudin, T., Lányi, D., Bekas, C., & Laino, T. (2018). “Found in Translation”: predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models. *Chemical Science*, *9*(28), 6091–6098. https://doi.org/10.1039/c8sc02339e

Schwaller, P., Laino, T., Gaudin, T., Bolgar, P., Hunter, C. A., Bekas, C., & Lee, A. A.

(2019). Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Central Science*, *5*(9), 1572–1583. https://doi.org/10.1021/acscentsci.9b00576

Segall, M. D., & Barber, C. (2014). Addressing toxicity risk when designing and selecting compounds in early drug discovery. *Drug Discovery Today*, *19*(5), 688–693. https://doi.org/10.1016/J.DRUDIS.2014.01.006

Segler, M. H. S., Kogej, T., Tyrchan, C., & Waller, M. P. (2018). Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, *4*(1), 120–131. https://doi.org/10.1021/acscentsci.7b00512

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., & Napolitano, A. (2010). RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *40*(1), 185–197. https://doi.org/10.1109/TSMCA.2009.2029559

Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., & Alonso-Betanzos, A. (2017). Ensemble feature selection: Homogeneous and heterogeneous approaches. *Knowledge-Based Systems*, *118*, 124–139. https://doi.org/10.1016/J.KNOSYS.2016.11.017

Shahlaei, M. (2013). Descriptor Selection Methods in Quantitative Structure−Activity Relationship Studies: A Review Study. https://doi.org/10.1021/cr3004339

Shao, C.-Y., Chen, S.-Z., Su, B.-H., Tseng, Y. J., Esposito, E. X., & Hopfinger, A. J. (2013). Dependence of QSAR Models on the Selection of Trial Descriptor Sets: A Demonstration Using Nanotoxicity Endpoints of Decorated Nanotubes. *Journal of Chemical Information and Modeling*, *53*(1), 142–158.

https://doi.org/10.1021/ci3005308

Sheikhpour, R., Sarram, M. A., Gharaghani, S., & Chahooki, M. A. Z. (2017). A Survey on semi-supervised feature selection methods. *Pattern Recognition*, *64*, 141–158. https://doi.org/10.1016/J.PATCOG.2016.11.003

Shen, D., Wu, G., & Suk, H.-I. (2017). Deep Learning in Medical Image Analysis. *Annual Review of Biomedical Engineering*, *19*(1), 221–248. https://doi.org/10.1146/annurev-bioeng-071516-044442

Shen, Q., Jiang, J.-H., Tao, J., Guo-li Shen, A., & Ru-Qin Yu. (2005). Modified Ant Colony Optimization Algorithm for Variable Selection in QSAR Modeling: QSAR Studies of Cyclooxygenase Inhibitors. https://doi.org/10.1021/CI049610Z

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., … Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, *529*(7587), 484–489. https://doi.org/10.1038/nature16961

Singh, S., & Gupta, P. (2014). *Comparative Study ID3, CART AND C4.5 Decision Tree Algorithms: A Survey*. *International Journal of Advanced Information Science and Technology (IJAIST) ISSN* (Vol. 27). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.685.4929&rep=rep1&type=pdf

Sitzmann, M., Ihlenfeldt, W.-D., & Nicklaus, M. C. (2010). Tautomerism in large databases. *Journal of Computer-Aided Molecular Design*, *24*(6–7), 521–551. https://doi.org/10.1007/s10822-010-9346-4

Snoek, J., Larochelle, H., & Adams, R. P. (2012). *Practical Bayesian Optimization of Machine Learning Algorithms*. Retrieved from https://papers.nips.cc/paper/4522-

practical-bayesian-optimization-of-machine-learning-algorithms.pdf

Solorio-Fernandez, S., Martinez-Trinidad, J. F., Carrasco-Ochoa, J. A., & Yan-Qing Zhang. (2012). Hybrid feature selection method for biomedical datasets. In *2012 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)* (pp. 150–155). IEEE. https://doi.org/10.1109/CIBCB.2012.6217224

Stefaniak, F. (2015). Prediction of Compounds Activity in Nuclear Receptor Signaling and Stress Pathway Assays Using Machine Learning Algorithms and Low-Dimensional Molecular Descriptors. *Frontiers in Environmental Science*, *3*, 77. https://doi.org/10.3389/fenvs.2015.00077

Stefanowski, J. (2016). Dealing with Data Difficulty Factors While Learning from Imbalanced Data. In *Challenges in computational statistics and data mining* (pp. 333–363). Switzerland: Springer, Cham. https://doi.org/10.1007/978-3-319-18781-5_17

Stokes, W. S. (2015). Animals and the 3Rs in toxicology research and testing : The way forward. *Human and Experimental Toxicology*, *34*(12), 1297–1303. https://doi.org/10.1177/0960327115598410

Subramanian, J., & Simon, R. (2013). Overfitting in prediction models – Is it a problem only in high dimensions? *Contemporary Clinical Trials*, *36*, 636–641. https://doi.org/10.1016/j.cct.2013.06.011

Sundarapandian, T., Shalini, J., Sugunadevi, S., & Woo, L. K. (2010). Docking-enabled pharmacophore model for histone deacetylase 8 inhibitors and its application in anti-cancer drug discovery. *Journal of Molecular Graphics and Modelling*, *29*(3), 382–

395. https://doi.org/https://doi.org/10.1016/j.jmgm.2010.07.007

Tan, M. E., Li, J., Xu, H. E., Melcher, K., & Yong, E. (2015). Androgen receptor: structure, role in prostate cancer and drug discovery. *Acta Pharmacologica Sinica, 36*(1), 3–23. https://doi.org/10.1038/aps.2014.18

Tang, J., Alelyani, S., & Liu, H. (n.d.). *Feature Selection for Classification: A Review*. Retrieved from https://pdfs.semanticscholar.org/310e/a531640728702fce6c743c1dd680a23d2ef4.pdf

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N.Y.), 290*(5500), 2319–2323. https://doi.org/10.1126/science.290.5500.2319

The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time. (n.d.). Retrieved May 19, 2020, from http://jalammar.github.io/illustrated-transformer/

Tirado-Rives, J., & Jorgensen, W. L. (2006). Contribution of Conformer Focusing to the Uncertainty in Predicting Free Energies for Protein−Ligand Binding. *Journal of Medicinal Chemistry*, *49*(20), 5880–5884. https://doi.org/10.1021/jm060763i

Todeschini, R., Consonni, V., & Wiley InterScience (Online service). (2000). *Handbook of molecular descriptors*. Wiley-VCH. Retrieved from https://books.google.com/books?hl=en&lr=&id=TCuHqbvgMbEC&oi=fnd&pg=PP2&ots=jvBAwfyPnb&sig=DTUfCkFm8CPqaDRl4tZInOYVSkU#v=onepage&q&f=false

Tong, W., Hong, H., Fang, H., Xie, Q., & Perkins, R. (2003). Decision Forest:

Combining the Predictions of Multiple Independent Decision Tree Models. *Journal of Chemical Information and Computer Sciences*, *43*(2), 525–531. https://doi.org/10.1021/ci020058s

Tropsha, A. (2007). Predictive Quantitative Structure–Activity Relationship Modeling. In *Comprehensive Medicinal Chemistry II* (pp. 149–165). Elsevier. https://doi.org/10.1016/B0-08-045044-X/00248-0

Tropsha, Alexander. (n.d.). Best practices for developing predictive QSAR models. Retrieved from http://infochim.u-strasbg.fr/CS3_2010/OralPDF/Tropsha_CS3_2010

Tropsha, Alexander. (2010). Best Practices for QSAR Model Development, Validation, and Exploitation. *Molecular Informatics*, *29*(6–7), 476–488. https://doi.org/10.1002/minf.201000061

Tropsha, Alexander, Gramatica, P., & Gombar, V. (2003). The Importance of Being Earnest: Validation is the Absolute Essential for Successful Application and Interpretation of QSPR Models. *QSAR & Combinatorial Science*, *22*(1), 69–77. https://doi.org/10.1002/qsar.200390007

Trott, O., & Olson, A. J. (2010). AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, EfficientOptimization, and Multithreading. *Journal of Computational Chemistry*, *31*(2), 455–461. https://doi.org/10.1002/jcc

Uesawa, Y. (2016). Rigorous Selection of Random Forest Models for Identifying Compounds that Activate Toxicity-Related Pathways. *Frontiers in Environmental Science*, *4*, 9. https://doi.org/10.3389/fenvs.2016.00009

Van Der Maaten, L., Postma, E., & Van Den Herik, J. (2009). *Dimensionality Reduction: A Comparative Review*. Retrieved from http://www.uvt.nl/ticc

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., … Polosukhin, I. (n.d.). *Attention Is All You Need*.

Venkatraman, V., Dalby, A. R., & Yang, Z. R. (2004). Evaluation of Mutual Information and Genetic Programming for Feature Selection in QSAR. https://doi.org/10.1021/ci049933v

Wang, Q. (2011). *Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models*. Retrieved from https://arxiv.org/pdf/1207.3538.pdf

Wang, S., & Yao, X. (2009). Diversity Analysis on Imbalanced Data Sets by Using Ensemble Models. In *2009 IEEE Symposium on Computational Intelligence and Data Mining* (pp. 324–331). IEEE. https://doi.org/10.1109/CIDM.2009.4938667

Wang, Y., Xiao, J., Suzek, T. O., Zhang, J., Wang, J., & Bryant, S. H. (2009). PubChem: a public information system for analyzing bioactivities of small molecules. *Nucleic Acids Research*, *37*(Web Server), W623–W633. https://doi.org/10.1093/nar/gkp456

Weininger, D. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Modeling*, *28*(1), 31–36. https://doi.org/10.1021/ci00057a005

Willighagen, E. L., Mayfield, J. W., Alvarsson, J., Berg, A., Carlsson, L., Jeliazkova, N., … Steinbeck, C. (2017). The Chemistry Development Kit (CDK) v2.0: atom typing, depiction, molecular formulas, and substructure searching. *Journal of Cheminformatics*, *9*(1), 33. https://doi.org/10.1186/s13321-017-0220-4

Wilson, D. L. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-2*(3), 408–421.

https://doi.org/10.1109/TSMC.1972.4309137

Winter, R., Montanari, F., Noé, F., & Clevert, D. A. (2019). Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chemical Science*, *10*(6), 1692–1701. https://doi.org/10.1039/c8sc04175j

Wu, K., & Wei, G.-W. (2018). Quantitative Toxicity Prediction Using Topology Based Multitask Deep Neural Networks. *Journal of Chemical Information and Modeling*, *58*(2), 520–531. https://doi.org/10.1021/acs.jcim.7b00558

Wu, Y., & Wang, G. (2018). Machine Learning Based Toxicity Prediction: From Chemical Structural Description to Transcriptome Analysis. *International Journal of Molecular Sciences*, *19*(8), 2358. https://doi.org/10.3390/ijms19082358

Xu, Youjun, Dai, Z., Chen, F., Gao, S., Pei, J., & Lai, L. (2015). Deep Learning for Drug-Induced Liver Injury. *Journal of Chemical Information and Modeling*, *55*(10), 2085–2093. https://doi.org/10.1021/acs.jcim.5b00238

Xu, Youjun, Pei, J., & Lai, L. (2017). Deep Learning Based Regression and Multiclass Models for Acute Oral Toxicity Prediction with Automatic Chemical Feature Extraction. *Journal of Chemical Information and Modeling*, *57*(11). https://doi.org/10.1021/acs.jcim.7b00244

Xu, Yuting, Ma, J., Liaw, A., Sheridan, R. P., & Svetnik, V. (2017). Demystifying Multitask Deep Neural Networks for Quantitative Structure−Activity Relationships. https://doi.org/10.1021/acs.jcim.7b00087

Xue, L., & Bajorath, J. (2000). Molecular Descriptors for Effective Classification of Biologically Active Compounds Based on Principal Component Analysis Identified by a Genetic Algorithm. https://doi.org/10.1021/CI000322M

Yan, H., & Dai, Y. (2011). The Comparison of Five Discriminant Methods. In *2011 International Conference on Management and Service Science* (pp. 1–4). IEEE. https://doi.org/10.1109/ICMSS.2011.5999201

Yang, H., Sun, L., Li, W., Liu, G., & Tang, Y. (2018). In Silico Prediction of Chemical Toxicity for Drug Design Using Machine Learning Methods and Structural Alerts. *Frontiers in Chemistry*, *6*, 30. https://doi.org/10.3389/fchem.2018.00030

Yang, P., Ho, J. W., Yang, Y., & Zhou, B. B. (2011). Gene-gene interaction filtering with ensemble of filters. *BMC Bioinformatics*, *12*(Suppl 1), S10. https://doi.org/10.1186/1471-2105-12-S1-S10

Yang, P., Zhou, B. B., Yang, J. Y.-H., & Zomaya, A. Y. (2013). Stability of Feature Selection Algorithms and Ensemble Feature Selection Methods in Bioinformatics. In *Biological Knowledge Discovery Handbook* (pp. 333–352). Hoboken, New Jersey: John Wiley & Sons, Inc. https://doi.org/10.1002/9781118617151.ch14

Yap, C. W. (2011). PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of Computational Chemistry*, *32*(7), 1466–1474. https://doi.org/10.1002/jcc.21707

Ye, J. (n.d.). *Discriminant Analysis for Dimensionality Reduction: An Overview of Recent Developments i Contributors*. Retrieved from https://pdfs.semanticscholar.org/8d4b/fd73c50545212a8646623ec76e49656421a5.pdf

Yoo, C., & Shahlaei, M. (2018). The applications of PCA in QSAR studies: A case study on CCR5 antagonists, *91*(1). https://doi.org/10.1111/cbdd.13064

Young, D., Martin, T., Venkatapathy, R., & Harten, P. (2008). Are the Chemical

Structures in Your QSAR Correct? *QSAR & Combinatorial Science*, *27*(11–12), 1337–1345. https://doi.org/10.1002/qsar.200810084

Zeng, Z., Zhang, H., Zhang, R., & Zhang, Y. (2014). A Hybrid Feature Selection Method Based on Rough Conditional Mutual Information and Naive Bayesian Classifier. *ISRN Applied Mathematics*, *2014*, 1–11. https://doi.org/10.1155/2014/382738

Zhao, L., Wang, W., Sedykh, A., & Zhu, H. (2017). Experimental Errors in QSAR Modeling Sets: What We Can Do and What We Cannot Do. *ACS Omega*, *2*(6), 2805–2812. https://doi.org/10.1021/acsomega.7b00274

Zhu, X.-W., Xin, Y.-J., & Ge, H.-L. (2015). Recursive Random Forests Enable Better Predictive Performance and Model Interpretation than Variable Selection by LASSO. https://doi.org/10.1021/ci500715e

Zhu, X., & Wu, X. (2004). Class Noise vs. Attribute Noise: A Quantitative Study. *Artificial Intelligence Review*, *22*(3), 177–210. https://doi.org/10.1007/s10462-004-0751-8