GRIFFITH UNIVERSITY

THESIS

Molecular toxicity prediction using deep learning

Abdul Karim

BSc, MSc

Submitted in fulfillment of the requirements

for the degree of Doctor of Philosophy

in the

Griffith Sciences

School of Information and Communication Technologies

Griffith University

March 2021

Statement of Originality

I, Abdul Karim, declare that this thesis titled, "Molecular toxicity prediction using deep learning" and the work presented in it are my own. I confirm that:

"This work was done wholly or mainly while in candidature for a research degree at this University. This thesis has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself."

Signed: 

Date: 12/03/2021

Abstract

In this thesis, we address the black-box nature of deep learning models for molecular toxicity prediction as well as propose methods for aggregating various chemical features to have an improved accuracy. An ideal toxicity prediction model is characterized with high accuracy, capable of handling descriptors/features diversity, ease of training and interpretability. Considering these attributes of an ideal model, in the first quarter of this thesis we present a novel hybrid framework based on decision trees (DT) and shallow neural networks (SNN). This method paves a path to feature interpretability while enhancing the accuracy by selecting only the relevant features for model training. Using this approach, the run-time complexity of developed toxicity model is substantially reduced. The idea is to create a contextual adaptation of the models by hybridizing the decisions trees to enhance the features interpretability and accuracy both.

In the later quarters of this thesis, we argue for the idea of effective aggregation of chemical knowledge about molecules in toxicity prediction. Molecules are represented in various data formats such that each format has its own specific role in predicting molecular activities. We propose various deep learning ensemble approaches to effectively aggregate different chemical features information. We have applied these methods to quantitative and qualitative molecular toxicity prediction problems and have obtained new state-of-the-art accuracy improvements with respect to existing deep learning methods. Our ensembling methods also prove helpful in making the model's prediction robust over a range of performance metrics for toxicity prediction.

Acknowledgements

I would like to extend thanks to the many people, who so generously contributed to the work presented in this thesis confirmation report.

Special mention goes to my supervisor, Professor **Abdul Sattar** and Associate Supervisor, **MAHakim Newton**. My PhD is an amazing experience and I thank Abdul Sattar wholeheartedly, not only for his tremendous academic support, but also for giving me so many wonderful opportunities. I would extend my thanks to Griffith University for providing me generous scholarship to complete my PhD. I would also like to thank my friends specially **Jaswinder Singh** and **Jaspreet Singh** for their encouragement and guidance throughout my PhD studies.

Finally, to my caring, loving, and supportive fiancé, **Rabia Raza**: my deepest gratitude. Your encouragement when the times got rough are much appreciated and duly noted.

Publications

The main contributions of this study published in either peer reviewed journals or conferences/workshops in the field.

- Karim, A., Mishra, A., Newton, M. H., Sattar, A. (2019). Efficient toxicity prediction via simple features using shallow neural networks and decision trees. Acs Omega, 4(1), 1874-1888.

- Karim, A., Singh, J., Mishra, A., Dehzangi, A., Newton, M. H., Sattar, A. (2019, August). Toxicity Prediction by Multimodal Deep Learning. In Pacific Rim Knowledge Acquisition Workshop (pp. 142-152). Springer, Cham.

- Karim, A., Lee, M., Balle, T., Sattar, A. (2021). CardioTox net: A robust predictor for hERG channel blockade via deep learning meta ensembling approaches. **(Accepted for publication in BMC Journal of cheminformatics on 30-07-2021)**

- Karim, A., Riahi, V., Mishra, A., Newton, M. H., Dehzangi, A., Balle, T., Sattar, A. (2021) Quantitative toxicity prediction via meta ensembling of multi task deep learning models. Acs Omega, 6, 18, 12306–12317.

Contents

List of Figures

List of Tables

Dedicated to the ONE who gives me life.....…

Chapter 1

Introduction

In this thesis, we address the challenge of molecular toxicity prediction using various deep learning methods. Molecular toxicity prediction is one of the most crucial steps in drug discovery process in pharmaceutical industries. It involves a series of wet lab experiments and animal trials for any new drug candidate or a molecule. Usually it takes decades of experiments and trials to market a newly discovered molecule or drug. One of the critical component in a typical drugs discovery pipeline is determining if a molecule used in developing a drug will have any harmful affects, thus called chemical toxicity. Chemical toxicity is an important measure in environmental, agricultural, and pharmaceutical science. In the environmental context, toxic chemicals may cause varieties of chronic diseases. In pharmacology, toxicity prediction plays a vital role in the drug discovery pipeline. This makes toxicological screening to be mandatory for the development of new drugs and for the extension of the therapeutic potential of existing molecules. In this chapter, the main motivation of this research, the specific problems addressed, an overview of the contribution and thesis outline is described.

1.1 Motivation

Several in vitro/in vivo techniques have been devised to determine varieties of toxic effects. However,these techniques for examining chemical toxicity

are highly cost and time-intensive. In most cases, research with animals is the most reliable means of detecting important toxic properties of chemical substances. The usage of animals however for toxicology screening has raised ethical concerns. Therefore, there is an increased demand for cost-and time-efficient as well as animal safe toxicological screening methods. This gives an inspiration of using computational methods based on structure activity relation (SAR) techniques to determine or predict the toxicity of a chemical compound without using animals. SAR techniques uses computational modeling methods to predict the toxic activities of molecules based on the structural properties. Amongst SAR techniques, classical machine learning methods such as k-nearest neighbors (KNN), support vectors machines (SVM), decision trees (DT) and random forest (RF) have been used extensively to predict molecular toxicity. These classical SAR methods however does not perform very well in terms of performance accuracy when presented with large amount of data with physical and chemical descriptors. These methods also depend upon the feature engineering heavily which requires to design specific types of informative features for prediction. Therefore, state of the art deep learning methods such as deep neural networks (DNN) and its numerous variants are are widely adopted. There are two main challenges in using DNN and their variants.

- In toxicity prediction area, the black box nature of prediction models makes them hard to interpret. Moreover, large number of features are used in deep neural networks, which makes the model very prone to curse of dimensionality. Therefore, there is a need of an effective way of features to be used in neural networks for toxicity prediction. Also, it makes sense to devise new machine learning techniques which can combine the classical machine learning methods with deep learning approaches. This will not only help in making the prediction model more

transparent to obtain the importance of various chemical features for toxicity tasks but also boost the prediction performance.

- Each DNN variant requires the chemical data to be formulated in its own specific type that might restrict the performance to a specific type of features and model architecture used. There is a need of effective aggregation of various chemical features together to improve the overall performance over a range of performance metrics. Our motivation is to make a combined model that utilizes different types of features and architectures to obtain better collective performance that could go beyond the performance of each individual predictor.

1.2 Research problems and contributions

In this thesis, we address two problems related to molecular toxicity prediction using deep learning/machine learning approaches. The first problem is concerned with the effective use of features by jointly optimizing a shallow neural network with decision trees. The second problem concerned with using meta ensemble approaches to boost the overall performance of toxicity prediction models. Here we describe each problem and an overview of the contribution towards its solution in this thesis.

1.2.1 Research problems

- *Computationally intensive and black-box nature:* An ideal chemical toxicity model is characterized by its high accuracy, capability to deal with molecular descriptor diversity, ease of training, and slightly more interpretability. Unfortunately, most machine learning approaches act like black boxes; which means no insights are available from them about the problem or the solution structures, making them less trustworthy

from human perspective. Large number of chemical/molecular features in very deep neural network architectures makes them not only compute intensive, but also very prone to over fitting. This may also lead to curse of dimensionality which in turn plays a role in model's performance degradation.

- *Features specific performance restriction:* Chemical data can be expressed in various data formats and representations. These data formats represents molecules at various levels. Each format has its own merits and de-merits with respect to machine learning prediction models. These molecular data representations are also termed as chemical or molecular features. In molecular toxicity prediction area, each DNN variant requires the chemical data to be formulated in its own specific type which might restrict the performance to a specific type of features and model architecture used. Therefore, there is a need of effective aggregation of various chemical features together to improve the overall performance over a range of performance metrics.

1.2.2 Contributions

- *Effective use of molecular features:* In first quarter of this thesis, we argue for the models and methods that are simple in machine learning characteristics, efficient in computing resource usage, and powerful to achieve very high accuracy levels. We therefore present a novel hybrid framework that uses decision trees and shallow neural networks to build a simple machine learning model that paves a path to feature interpretability while achieving similar reasonable accuracy by selecting only the relevant features to train the model. To demonstrate this, we develop a single task-based chemical toxicity prediction framework using only 2D features that are less compute intensive. We effectively

use a decision tree to obtain an optimum number of features from a collection of thousands of them. We use a shallow neural network and jointly optimize it with decision tree taking both network parameters and input features into account. Our model needs only a minute on a single CPU for its training while existing methods using deep neural networks need about 10 min on NVIDIA Tesla K40 GPU. We obtain similar or better performance on several toxicity benchmark tasks. Moreover, we have developed a cumulative feature ranking method which enables to identify features that can help chemists perform prescreening of toxic compounds effectively. These contributions are published with a title "Efficient toxicity prediction via simple features using shallow neural networks and decision trees" in ACS Omega [68].

- *Meta ensemble approaches:* In this thesis, we propose various techniques to aggregate the outputs of deep learning models for toxicity prediction in single task and multi-task learning fashion. In the second quarter of this thesis, we study quantitative toxicity prediction and propose a machine learning model for the same. Our model uses an ensemble of heterogeneous predictors instead of typically using homogeneous predictors. The predictors that we use vary either on the type of features used or on the deep learning architecture employed. Each of these predictors presumably has its own strengths and weaknesses in terms of toxicity prediction. The outputs of all these predictors are averaged out to obtain the final output. We use six predictors in our model and test the model on four standard quantitative toxicity benchmark data sets.

  In the third quarter, we propose a meta ensemble technique for the single task and multitask quantitative toxicity data set to boost the over-all prediction performance. In ensemble technique, we train the base deep learning models on base molecular features to produce meta features.

A separate fully connected neural network is trained on meta features to produce the final output.

In the fourth quarter of this thesis, we apply our meta ensemble technique to cardio-toxicity data set and obtain state-of-the-art results over a range of classification accuracy metrics. For cardio-toxicity, we evaluate our meta ensemble technique against various classification metrics using two oppositely biased independent test sets and obtain a robust performance with respect to various state of the art methods. These contributions are published/under revision/under review [70, 71, 72].

1.3 Thesis outline

Following this introductory chapter, the rest of the thesis is organized in a number of related chapters.

- Chapter 2: reviews the key concepts of molecular toxicity and its prediction using machine learning techniques.
- Chapter 3: presents the use of simple chemical 2D features with joint optimization of shallow neural networks and decision trees to combat the compute intensive and black box nature of deep learning models in toxicity prediction.
- Chapter 4: presents an ensemble approach based on averaging of outputs of heterogeneous deep learning predictors to boost the quantitative toxicity prediction performance.
- Chapter 5: presents meta features ensemble technique for multi-task quantitative toxicity prediction.
- Chapter 6: presents the same meta features ensemble technique used for classifying molecules with cardio-toxicity properties. It also shows

that this technique can prove more robust as compared to other approaches for classifying cardio-toxic molecules.

- Chapter 7: presents a summary of this work, outlines some future directions to extend this work and concludes the thesis.

Chapter 2

Preliminaries

This chapter introduces the concepts molecular toxicity and computational methods to predict the same. The computational methods ranges from classical machine learning such as support vector machines, random forests and k-nearest neighbors to deep learning approaches such as fully connected neural networks, convolution neural networks and recurrent neural networks. Each of these techniques are discussed in this chapter and used throughout this thesis. Then we present an overview of single task vs multi task machine learning approaches later used in the chapter discussing quantitative toxicity prediction. We also shed some light on various types of chemical/molecular features and their compatibility with machine learning models used in this chapter. At the end of this chapter, we discuss various accuracy metrics which are used for performance evaluation of our toxicity prediction models presented in this thesis.

2.1 Molecular toxicity

Every year a broad spectrum of chemical compounds are produced in various laboratories all over the world. A large number of these chemical compounds are suspected to be toxic or hazardous for human life, and at the end, many of them are proven so. Toxicity is the degree to which a chemical compound can affect organisms, tissues, or cells. The main metric employed

to measure the toxicity level of a chemical compound is the concentration of the compound at the time of its exposure to a given organism [98]. The toxic concentration of a compound is ascertained by *endpoints* measuring experiments. Toxicity of a compound could vary with individuals and their ages, genders, and body weights. Thus, different toxicity indicators are devised to measure the toxicity over the population such as eye irritancy test [27, 154], mutagenicity [38, 43], toxicokinetics [156], neurotoxicity [40], embryotoxicity [78] and genetic toxicity [106].

Toxicity estimation, similar to other attributes of chemical compounds, is calculated using sophisticated experimental techniques on *in-vivo* or *in-vitro* models. However, these techniques are very time consuming and cost intensive. They also raise ethical concerns because of the involvement of animals. To address these issues, *in-silico* methods (computer-aided methods) have recently attracted much attention due to their lower cost and better time efficiency. There exist many *in-silico* methods, but the structure activity relationship (SAR) methods are one of the most successful ones.

2.2 Structure activity relationship and toxicity prediction

The main rationale behind structure activity relationship (SAR) methods is that chemical molecules that are similar in the structure should have similar activities [74]. The activity can be a quantitative which is related to a toxicity end point level. Quantitative toxicity can be modeled as a regression problem in machine learning [82]. Unlike determining the end point toxicity level, qualitative toxicity prediction is related to classifying the molecule to be toxic or non toxic in binary or multi-class machine learning context [68].

2.2.1 Eye ball toxicity

Usually, a medicinal chemist is capable of identifying specific types of substructures (toxicophores) in a molecule to classify them either toxic or non-toxic. For instance Figure 2.1 shows two molecules both of which consists of different toxicophores which can be identified easily by any medicinal chemist. Left side of Figure 2.1 shows neurotoxin VX which is toxic because of the presence of toxicophore which is phosphorous atom (orange arrow) bound to one sulfur and two oxygen atoms. Benzaldehyde cyanohydrin on the right side of Figure 2.1 is also toxic because of other well known toxicophore i.e. carbon atom (green arrow) which has both a hydroxyl (OH) group and a cyanide (CN) group (orange arrows) bound to it. It is easier to identify toxic molecules with an eye ball toxicity if they have well known toxicophore present in their structure.

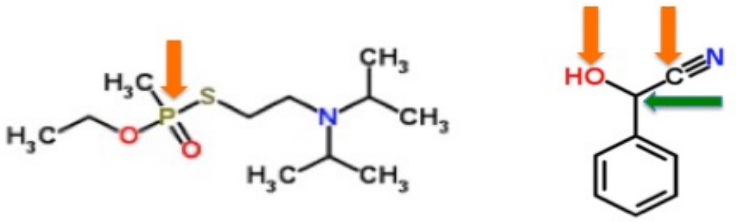

FIGURE 2.1: Toxicophore of the deadly neurotoxin VX (Left) and in benzaldehyde cyanohydrin (Right). Image taken from ACSH blog post [6].

2.2.2 Toxicity without obvious toxicophores

There is a large number of molecules with no obvious toxicophores and they still show toxic activity. For instance Figure 2.2, three molecules with no obvious toxicophores with very similar structures and yet, molecule A (urushiol) is toxic, molecule B (vitamin A) is good and molecule C (resveratrol) is neutral. All the three molecules share similar structure of long chain of carbon atoms with hydroxyl groups attached to them [6]. The challenge arises when we have similar structures and yet very different activities. There might be

some hidden features/structural attributes which are not obvious but might be responsible for the molecule's activity. In these situations when there is no obvious toxicophore, computational methods such as machine learning techniques capturing the structure activity relationships could prove very useful to predict the toxicity of molecule.

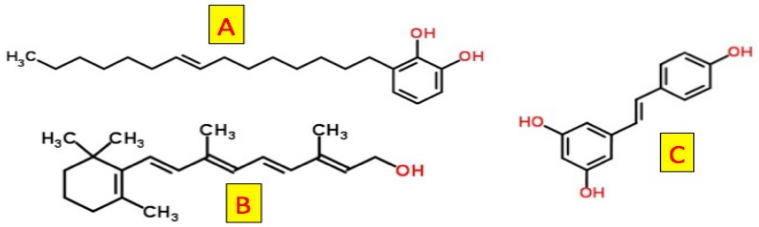

FIGURE 2.2: Three molecules with no obvious toxicophore [6].

2.3 In-silico methods for toxicity prediction

In recent years, machine learning methods have been widely used in drug discovery [88]. Because these techniques have proven to be capturing structure activity relationships where there is no obvious toxicophores [9, 23, 75]. Performance of traditional machine learning algorithms depends heavily upon the quantity and quality of training data along with domain knowledge based feature engineering [23]. Under the umbrella of machine learning, methods like K-Nearest Neighbors (KNN) and Support Vectors Machines (SVM) were used for Structure Activity Relation (SAR) techniques [9, 23, 75]. For instance, a KNN model used for hazard evaluation support systems was designed on carefully selected eight fingerprints as input features for a relatively small data set of 94 chemicals in the training set and 24 chemicals in the test set [122]. Similarly in another study, 74 topological descriptors with 314 training instances were used for specific COX-2 inhibitors [75]. These models perform relatively better on smaller data sets with fewer pre-selected features. One key limitation of KNN algorithm is the exponential rise of computational cost with the size of the input samples [8, 32, 140]. In contrast, non-linear

SVMs can manage high dimensional data but do not exhibit sufficiently robust performance on diverse chemical descriptors [132].

Besides KNN and SVMs, naive Bayes and random forest (RF) methods were also used extensively for toxicity prediction [13, 109, 137, 149]. Although RF is a decision tree (DT) method capable of handling high dimensional and diverse features, yet in many cheminformatics data sets, it shows a relatively low classification accuracy when compared to deep neural networks (DNN) [132, 139]. DNN is an artificial neural network with more than one hidden layer between the input and output while a shallow neural network (SNN) has only one hidden layer [14, 100, 124, 144]. In order to achieve high accuracy in a DNN, relatively a large data set is preferred with numerous features [105, 138]. In RF, features are used in raw form while DNN converts them to complex features using hidden layers [31, 139]. Moreover, hyper-parameter tuning in DNN gives a better control over a granular level optimization unlike in other machine learning approaches. In terms of features used to predict toxicity of molecules, DNN in most of these previous studies utilize single type of features such physicochemical, fingerprints or graph features [16, 18, 85, 86]. The key to success for these previous methods for toxicity prediction is elucidating correct structure-property relationships from existing data using high level physicochemical features along with fingerprints. There is also substantial literature for combining various types of features and features selection for molecular activity prediction, but no clear winner is concluded as yet because performance depends on the characteristics of the molecules used for modeling [110]. In several cases though, it was observed that the accuracy of the models can be improved by feature aggregation because of complementary information [110, 129]

2.3.1 Tox21 challenge and DNN

DNN attracted considerable attention in chemical information modelling community when Ma et al. won the "Merck Molecular Activity Challenge" using DNN networks in predicting the bio-molecular target for a drug [73, 93]. Later in 2014, "Tox21 Challenge" was also won by a group who used deep neural networks [139]. Following this trend, many other groups in computational chemistry used DNN models to achieve high accuracy to predict various chemical and biological characteristics including toxicity [102, 150], activity [29, 94, 112], reactivity [62, 63, 64], solubility [92], ADMET [76], docking [142], and QM-compound energies [102, 126, 128]. Even after achieving the state-of-the-art accuracy in various cheminformatics tasks, limited model interpretability of DNN made it less preferred in real world health informatics applications.

2.3.2 Quantitative toxicity

Quantitative structure activity relationship (QSAR) modelling using deep learning techniques have become very popular in recent years [74]. Many of these methods use 2D features calculated from the one dimensional representation of the molecules called SMILES, which is used to describe the chemical structure of a molecule as a string of characters [143]. There is a special grammar for SMILES to represent atoms, type and chemical bonds among them. SMILES strings are used to calculate various types of numerical features (e.g. Physicochemical descriptors) and molecular graphs by using different featurization methods [113, 153]. Traditional machine learning approaches such KNN, SVM, RF, and Fully Connected Neural Networks (FCNN) are based on numerical features, mainly to predict activity or properties of a chemical compound [89]. Besides, numerical features, SMILES strings can also be used to generate molecular graphs or images, which then can be used in

various types of convolutional neural network (CNN) to predict molecular activities [50]. Using CNN for molecular graphs or images needs relatively less domain expertise. It should be noted that SMILES strings can also be transformed into a vector representation or their respective fingerprints (fingerprints are bit strings composed of 0's and 1's) to be used in Recurrent Neural Networks (RNN) for molecular activity/property prediction [49].

Recently in the area of toxicity prediction, specialized type of features called element-specific topological descriptors (ESTDs) are used in deep neural networks and consensus models by TopTox to predict toxicity level [145]. Another recent software named AdmetSAR used molecular fingerprints to predict toxicity using RF, SVM, and KNN models [151]. The performance of all these quantitative prediction methods is restricted by the specific type of features or model used in prediction.

2.3.3 hERG toxicity

The human ether-à-go-go-related gene (hERG) encodes a voltage-dependent ion channel (Kv11.1, hERG) involved in controlling the electrical activity of the heart by mediating the re-polarisation current in the cardiac action potential. [111, 114]. Malfunction or inhibition of hERG-channel activity by drug molecules can lead to cardiac arrhythmias in the form of prolonged QT intervals and may lead to sudden cardiac arrest. Therefore, unwanted drug-induced arrhythmias are great concern for pharmaceutical companies and have led to blockbuster drugs being withdrawn from the market and discontinuation of drugs in late stages of development [11]. To prevent new drugs with unwanted hERG-related cardiotoxicity to enter the market, guidelines for assessment of potential for QT interval prolongation by non-cardiovascular medicinal products were decided at the International Conference on Harmonization of Technical Requirements for the Registration of Pharmaceuticals

for Human Use (ICH) [30, 141].

These procedures are time-consuming and expensive and therefore, to prevent product depletion due to cardiotoxicity at late preclinical and clinical stages, there is focus on preventing drugs with hERG channel activity form entering drug discovery pipelines in the first instance. To avoid this, computational methods to predict hERG liability have been established and can help prioritise molecules during the early phase of drug development [141]. Most of these methods are based on either machine learning techniques, including random forest, support vector machine, deep neural networks and graph convolutional neural networks (GCN) or on structure based methods including pharmacophore searching, quantitative structure activity relationships and molecular docking [18, 21, 35, 39, 85]. Publicly available high quality datasets consisting of molecules classified as hERG and non-hERG blockers are available and often utilized by these computational tools [18, 85, 86]. The datasets annotate chemical structure by SMILES strings which is a chemical language that describes the chemical structure using ASCII character strings. The SMILES strings are human readable and are considered a low-level representation of molecular structure [143]. For ease of computational processing, the SMILES strings are often converted into binary vectors of fixed length called fingerprints which is another low level representation [66]. From these molecular representations, similarly, high level features such as 2D and 3D physico-chemical descriptors can be computed from SMILES strings which are then used in various machine learning models [68, 85]. Alternatively, molecular graph representations have been used with graph convolutional neural networks [90]. This intermediate level molecular graph representation offers a compromise between high level physico-chemical features and low level SMILES and fingerprints [121]. Under this category, each molecule can be represented via a molecular graph which consists of node features and an adjacency matrix.

2.4 Machine learning techniques

Here we describe main supervised machine learning techniques which are used in later chapter of this thesis. These techniques include linear and logistic regressions, decision trees and random forests, support vector machines, deep neural neural networks and its variants. We also sheds light on each of these methods interpretability and computational cost.

2.4.1 Linear and logistic regressions

Both linear and logistic regression methods are the part of simple supervised machine learning methods. Linear regression is used for regression problems whereas logic regression is used for classification problems. In linear regression shown in Equation 2.1, the approach is to find the best fit line to predict the output In the logistic regression as shown in Equation 2.2, the approach is to try for curved graphs that classify between the two classes that are 0 and 1 [3]. Linear regression can be made interpretable with explaining the coefficients given in Equation 2.1. It works better for small data sets but in most of real world cases, it is hard to satisfy the pre-requisite conditions for using linear regression. These conditions include linearity, normality, homoscedasticity, independence, fixed features and absence of multicollinearity [104]. In case of logistic regression, the interpretations always come with a clause that all other features stay the same. In logic regression, interactions has to be hand crafted and it shows poor performance in most real world tasks. It fails when relation between target and features is non-linear and the features are interacting with each other.

$$y = a_0 + a_1 x^1 + a_2 x^2 + a_3 x^3 .... \tag{2.1}$$

$$p(y=1) = \frac{1}{1+e^{-(a_0+a_1x^1+a_2x^2+a_3x^3)}} \tag{2.2}$$

2.4.2 Decision trees and random forests

Decision tree is considered as one of the most important interpretable machine learning algorithm. It is very intuitive and covers interactions between the features. As it is based on hard splits, so it is very inefficient to handle linear relationships between the feature and the target variable. This leads to lack of smoothness and instability. Moreover, to achieve high performance, we need to use many trees instead of one tree, which eventually makes it hard to interpret it again [59]. A large number of decision trees (as building blocks) work in an ensemble like way to predict or classify an instant in random forest. The ensemble of the trees is based on the idea of boosting performance by combining weak learners RF is a decision tree method capable of handling high-dimensional and diverse features [28]. Ensemble learning combines predictions from each individual decision tree and averages them to produce a more accurate prediction, as shown in Figure 2.3, where each individual decision tree is itself a weak learner which then becomes into a strong learner together with other decision trees. At every node in the tree, a small subset of variables is chosen to find a variable or a value of that variable which optimises the split [28]. Again, the compute cost and interpretability depends upon the number of trees in random forest. Usually large number of trees are used in random forest which makes them computationally expensive and black box in nature [67].

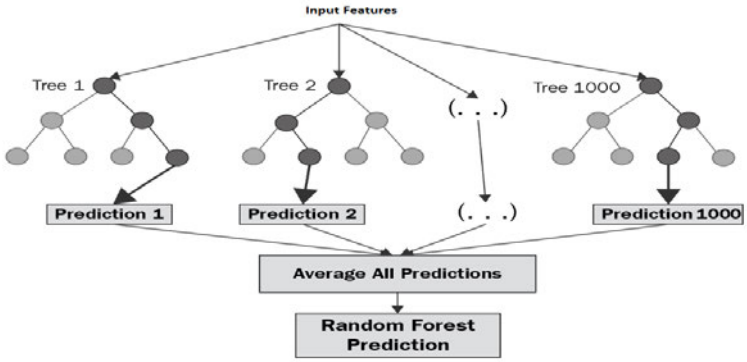

FIGURE 2.3: Trees in decision tree forming random forest.

2.4.3 Support vector machines

Support vector machines (SVM) belongs to a class of supervised machine learning methods. It attempts to find a line/hyper-plane (in multidimensional space) that separates classes of data under observation or ranges for regression [10]. SVM designates the new point depending on whether it lies on the positive or negative side of the hyper-plane. A function or kernel is used to map data from a lower dimension to a higher dimension. A kernel is useful in extracting a hyper-plane by transferring the data into high dimensional space without any computational cost [10]. The reason to transform the data into high dimensional space is that it is located in high dimensional space, wherein a linear hyper-plane can be used to separate the data of two classes, as shown in Figure 2.4. A non-linear SVM can handle high dimensional data but not robust enough to handle the diversity of chemical descriptors but mostly not the state of art classification accuracy [134].

2.4.4 Deep neural networks

Deep neural networks (often termed as deep learning) is a set of machine learning algorithms extensively used for predictions in supervised and unsupervised manners. Recently the advancement of deep learning technologies along with better computing resources have made them very popular

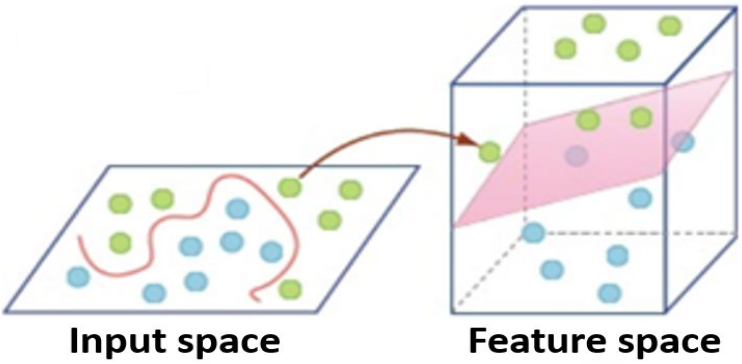

FIGURE 2.4: An illustration of creating a linear hyper-plane from a high dimensional transformation in a support vector machine model.

in fields such as computer vision [115], natural language processing [146], drugs discovery [68, 69, 70] and wide variety of related fields [33]. Here we will briefly discuss three main deep neural networks types such as fully connected neural network, convolution neural networks and recurrent neural networks. In later chapter of this thesis, we have used various variants of these three neural networks which are discussed in their respective chapters.

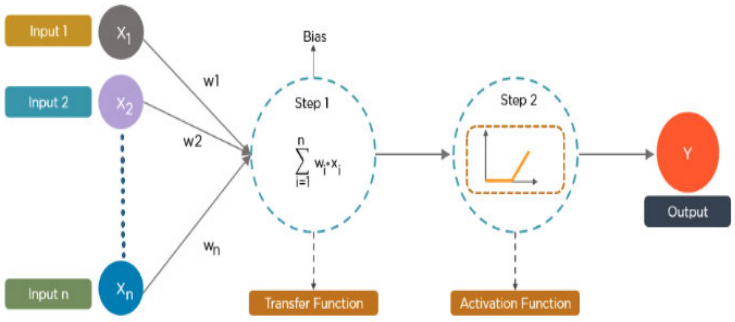

FIGURE 2.5: Transfer and activation functions used in various architectures of deep neural networks [4].

Fully connected neural networks

Fully connected neural networks (FCNN) can be viewed as a complex mapping function, where the fundamental unit of a FCNN is called a neuron and takes the input and computes the output after applying non-linearity

as shown in Figure 2.5 . There are three different types of layers in a network, namely, input, hidden and output layer as shown in Figure 2.6. Each layer is composed of neurons. The input layer takes the feature vector and multiplies it with a weight matrix followed by nonlinear activation as shown in Figure 2.5. The weight matrix, which is usually initialised randomly in the beginning, is then adjusted on the basis of the error at the output unit (layer). A gradient descent based back-propagation algorithm is used to adjust the weight matrix on the basis of the error [119]. The tuning or updating of the weight matrix in each step is called iteration. In each iteration, a chunk of data called a mini-batch is selected from the complete data to adjust the weights. The complete pass over the data is called an epoch. The overall process of weight adjustment is called training the network. To avoid over-fitting, a dropout technique is added after any hidden layer. Drop-out is randomly dropping the neurons in each iteration to reduce over-fitting and increase generality [130].

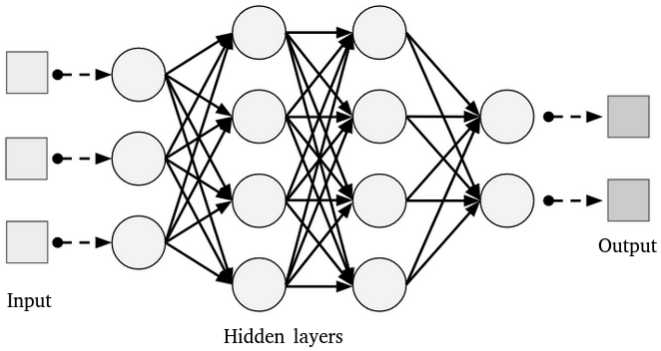

FIGURE 2.6: Sample architecture for fully connected neural network with input, output and hidden layers. Image taken from Deep Learning by Josh Patterson [1].

Convolution neural networks

Convolution Neural Network (CNN) is a special type of neural network for the image data. CNNs can extract low level features from images and compute more complex features as we go deeper in the networks [135]. Variants

of CNN like Inception, Alexnet and Resnet have been developed and employed as highly accurate image classification models [56]. CNNs undergoes the process of convolution which is applying number of filter (also called feature maps) with various dimensions to create convolutional layer. A non linear function is usually used after creating the convolutional layer. After convolution process, a pooling layer is applied to reduce the spatial size of the representation [107]. At the end after flattening, one or few fully connected layer are used in CNN to form class probabilities or regression values based on the features CNN has learnt as shown in Figure 2.7. Training process for CNN is the same as discussed in FCNN.

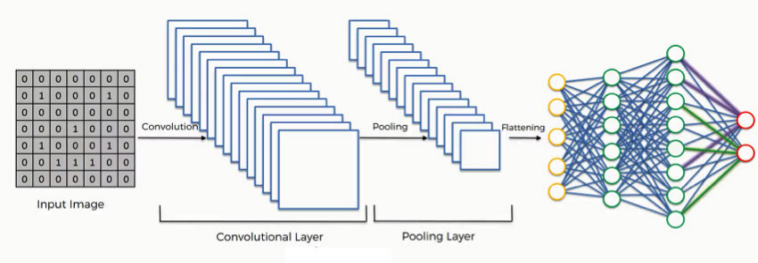

FIGURE 2.7: Classical convolutional neural network for image data. Image taken from SuperDataScience [2].

Recurrent neural networks

Recurrent Neural Network (RNN) is a specialized neural network for sequential data. RNNs can learn features directly from the sequence data without explicitly computing features. RNN is recurrent in nature and use their internal state (memory) to process the sequence of data. It performs the same function for every input of data while the output of the current input depends on the past one computation. After producing the output, it is copied and sent back into the recurrent network as shown in Figure 2.7. For making a decision, it considers the current input and the output that it has learned from the previous input [5]. RNN have shown great success in natural language processing and machine translation [101]. RNNs usually are prone

to short term memory problem [58]. The information flows from one cell to another sequentially and might be corrupted later in the network for longer sequences. Long short-term memory (LSTM) units or gated recurrent units (GRU) in RNN offer solutions to the short term memory problem [24].

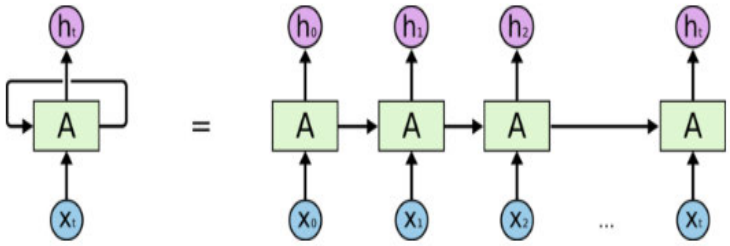

FIGURE 2.8: Classical recurrent neural network for sequence data. Image taken from towards data science [5].

2.5 Single task and multi task learning

In machine learning, we typically optimize for a single task or problem in hand by training a single model on a specific data set. We fine tune our model and optimize all its parameters for one specific task to achieve acceptable performance. By doing so, we laser focus on a single task and might be ignoring some relevant signals from other tasks that can improve the over-all performance. If we share the lower level representation between tasks by training a single model for multiple tasks, we might be able to generalize better on our original task under consideration. Multitask learning has been successfully applied across various domains such as natural language processing, computer vision and drug discovery [118]. In the context of Deep Learning, multi-task learning is typically done with either hard or soft parameter sharing of hidden layers [118]. In hard parameters sharing generally, hidden layers between multiple tasks are shared while some task specific layers near the output are kept un-shared [20, 118]. In soft parameter sharing on the other hand, each task has its own model with its own parameters. The

distance between the parameters of the model is then regularized in order to encourage the parameters to be similar [37, 118]. In this thesis, we only utilize hard parameter sharing for multi-task quantitative toxicity prediction model because of less chance of over-fitting.

2.6 Evaluation metrics

In this thesis, we used various types of evaluation metrics to measure the prediction performance of our proposed models for toxicity data sets. We broadly divide the evaluation metrics into quantitative and qualitative toxicity prediction based problems.

2.6.1 Evaluation criteria for quantitative toxicity prediction

We use three evaluation metrics for reporting the performance of our models for quantitative toxicity prediction.

- **Coefficient of determination r^2:** The first metric used in the paper is the coefficient of determination r^2 shown in Equation (2.3) where y_j and $\hat{y}_j$ respectively denote the predicted and actual value, and $\bar{y}_j$ denotes the mean of actual values. The coefficient of determination r^2 explains the relationship between the predicted and actual values. It varies between 0 and 1, and the higher the value of r^2, the better the model's performance.

$$r^2 = 1 - \frac{\sum_{j=1}^{n}(y_j - \hat{y}_j)}{\sum_{j=1}^{n}(y_j - \bar{y}_j)} \tag{2.3}$$

- **Mean absolute error (mae):** The second metric is the mean absolute error (mae) shown in Equation (2.4). The mae is the mean difference between the prediction y_j and the actual observation $\hat{y}_j$).

$$mae = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{2.4}$$

- **Root mean squared error (rmse):** The third metric is the root mean squared error (rmse) shown in Equation (2.5). The rmse is the square root of the mean of squared errors. In rmse, the errors are squared, so the large errors will have the higher weights.

$$rmse = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \tag{2.5}$$

2.6.2 Evaluation criteria for qualitative toxicity prediction

In order to measure the classification performance of qualitative toxicity prediction, we used the following metrics: Area under curve of receiver operating curve (AUC-ROC), specificity (SPE), sensitivity (SEN), negative predictive value (NPV), positive predictive value (PPV), accuracy (ACC) and Matthew's correlation coefficient (MCC). The details of these metrics are as follows:

- **Area under curve of receiver operating curve (AUC-ROC):** AUC-ROC takes into account all the threshold. The higher the value of AUC-ROC, the better the model is distinguishing between classes. It can be computed by taking area under the curve for true positive rate (TPR) on the y-axis and false positive rate (FPR) on the x-axis for a given dataset. TPR which is also called sensitivity (SEN) describes how good the model is at classifying a molecule as a positive class when the actual outcome is also a positive class. FPR describes how often a positive class is predicted when the actual outcome is negative class.

$$SEN = TPR = \frac{TP}{TP + FN} \tag{2.6}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.7}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives, SEN = Sensitivity.

- **Specificity (SPE):** SPE is the total number of true negatives divided by the sum of the number of true negatives and false positives. Specificity would describe what proportion of the negative class got correctly classified by our model.

$$SPE = \frac{TN}{TN + FP} \tag{2.8}$$

- **Negative predictive value (NPV):** NPV describes the probability of a molecule predicted as negative class to be actually as negative class.

$$NPV = \frac{TN}{TN + FN} \tag{2.9}$$

- **Positive predictive value (PPV):** PPV describes the probability of a molecule predicted as positive class to be actually as positive class.

$$PPV = \frac{TP}{TP + FP} \tag{2.10}$$

- **Accuracy (ACC):** ACC is the fraction of prediction our model got right. i.e it predicted positive class and negative class correctly.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

- **Matthews correlation coefficient (MCC):** MCC has a range of -1 to 1 where -1 indicates a completely wrong binary classifier while 1 indicates a completely correct binary classifier.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.12}$$

In this chapter, we discussed the concepts of molecular toxicity, features used to represent molecules, machine learning models used for activity relationship predictions, single and multitask learning and various evaluation metrics used in this thesis. In the next chapter, we present an efficient way of predicting molecular toxicity using a hybrid approach based on shallow neural networks and decision trees.

Chapter 3

Efficient toxicity prediction

This chapter is published in the following peer reviewed journal.

- Karim, A., Mishra, A., Newton, M. H., Sattar, A. (2019). Efficient toxicity prediction via simple features using shallow neural networks and decision trees. Acs Omega, 4(1), 1874-1888.

In this chapter, we strongly argue for the models and methods that are simple in machine learning characteristics, efficient in computing resource usage, and powerful to achieve very high accuracy levels. To demonstrate this, we develop a single task-based chemical toxicity prediction framework using only 2D features that are less compute intensive. We effectively use a decision tree to obtain an optimum number of features from a collection of thousands of them. We use a shallow neural network and jointly optimize it with decision tree taking both network parameters and input features into account. We call our model as hybrid2D because it is using decision trees and shallow neural networks with 2D features only. Our model needs only a minute on a single CPU for its training while existing methods using deep neural networks need about 10 min on NVIDIA Tesla K40 GPU. However, we obtain similar or better performance on several toxicity benchmark tasks. We also develop a cumulative feature ranking method which enables us to identify features that can help chemists perform pre-screening of toxic compounds effectively.

3.1 Introduction

In recent years, machine learning methods have been widely used in drug discovery [88]. Classical machine learning methods like k-nearest neighbors (KNN) and support vectors machines (SVM) were used for structure activity relation (SAR) techniques [9, 23, 75]. Performance of classical machine learning algorithms depends heavily upon the quantity and quality of training data along with domain knowledge based feature engineering. For instance, a KNN model used for hazard evaluation support systems (HESS) was designed on carefully selected eight fingerprints as input features for a relatively small data set of 94 chemicals in the training set and 24 chemicals in the test set [122]. Similarly in another study, 74 topological descriptors with 314 training instances were used for specific COX-2 inhibitors [75]. These models perform relatively better on smaller data sets with fewer pre-selected features. One key limitation of KNN algorithm is the exponential rise of computational cost with the size of the input samples [8, 32, 140]. In contrast, non-linear SVMs can manage high dimensional data but do not exhibit sufficiently robust performance on diverse chemical descriptors [132].

Besides KNN and SVMs, naive Bayes and random forest (RF) methods were also used extensively for toxicity prediction [13, 109, 137, 149]. Although RF is a decision tree (DT) method capable of handling high dimensional and diverse features, yet in many cheminformatics data sets, it shows a relatively low classification accuracy when compared to deep neural networks (DNN) [132, 139]. DNN is an artificial neural network with more than one hidden layer between the input and output while a shallow neural network (SNN) has only one hidden layer [14, 100, 124, 144]. In order to achieve high accuracy in a DNN, relatively a large data set is preferred with numerous features [105, 138]. In RF, features are used in raw form while DNN converts them to complex features using hidden layers [31, 139]. Moreover,

hyper-parameter tuning in DNN gives a better control over a granular level optimization unlike in other machine learning approaches.

An ideal classification model is characterized by its high classification accuracy, capability to deal with molecular descriptor diversity, ease of training, and somewhat more importantly interpretability [133]. Unfortunately, most machine learning approaches act like black box; which means no insights are available from them about the problem or the solution structures, making them less trustworthy from human perspective. Considering the attributes of an ideal classification model, in this chapter, we present a novel hybrid framework that uses DTs and SNNs to build a simple machine learning model that paves a path to feature interpretability while enhances the accuracy by selecting only the relevant features to train the model.

Using the proposed hybrid framework, we then construct a prediction model and train it on nuclear receptor (NR), stress response (SR) and ames mutagenicity (AM) data sets. The NR and SR data sets are from Tox21 data repository [60] while the AM data set [55] from Hansen et al. For all three data sets, we calculate only 2D chemical descriptors, which are less multifarious in nature and easy to calculate. The SNN in our model has only one hidden layer with 10 neurons and is trained with significantly fewer features (in the range of hundreds) than existing methods. The training times for our prediction models are reduced to $\approx$ 1 minute on Intel Core i5 CPU while the same was reported $\approx$ 10 minutes in the previous study using NVIDIA Tesla K40 GPU [97]. However, our model still achieved better ensembled average accuracy of 0.836 AUC-ROC (area under the receiver operating characteristic curve), 0.862, and 0.878 for NR, SR, and AM respectively while the best known existing methods achieved 0.826, 0.858, and 0.860 respectively [55, 97]. It is worth noting that our main objective is not merely to improve the accuracy, but also to focus more on the compute intensiveness, obtaining

simpler prediction models in terms of numbers of features used and architecture of the neural network, and interpretability of the classification results. We show that our model enables us to elucidate the interpretation of the descriptors that are the most responsible for NR, SR and AM toxicity types. These descriptors showed high classification strength to discriminate toxic compounds and could be used as initial indicators for detecting NR, SR and AM toxicity types.

3.2 Materials and methods

The work flow of our hybrid framework is composed of three main blocks as shown in Figure 3.1. All these three main blocks with sub-modules in each are explained below.

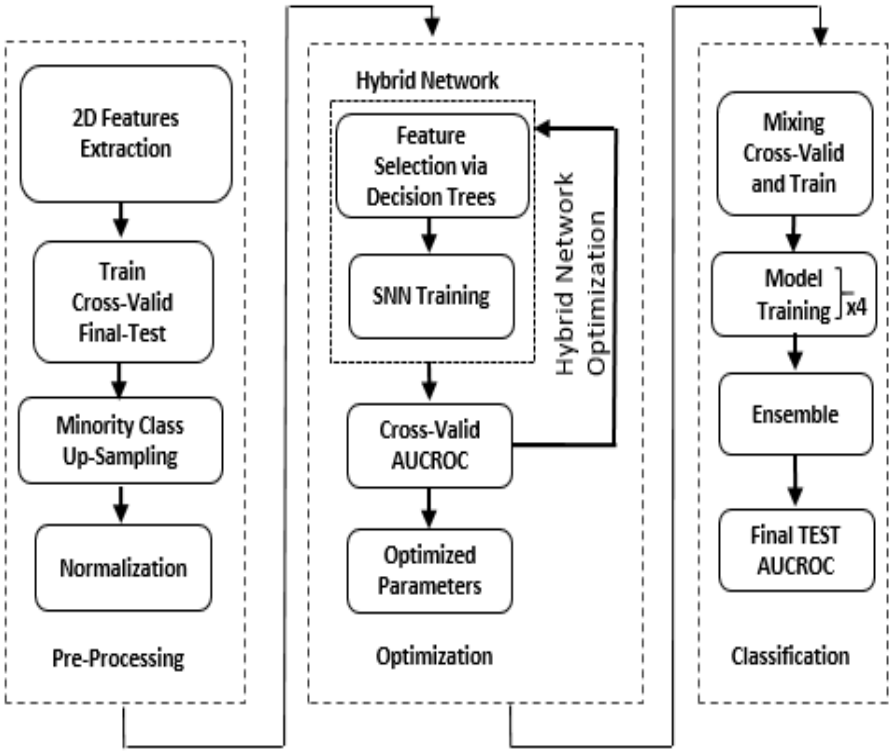

FIGURE 3.1: Prediction model flowchart.

3.2.1 Pre-processing

In pre-processing, 1422 2D chemical descriptors for NR and SR while 1249 for AM were calculated using an open source package called PADEL [152]. Data is split into train, CV and test sets. The split for NR and SR was predefined by the Tox21 Challenge [61] where a separate held out CV set of $\approx$ 296 instances is provided for an in-house cross validation purpose. On AM data, no such division was given so we divided it into 60% train, 20% CV and 20% test sets. The train and test sets for NR and SR consists of $\approx$ 8000 and $\approx$ 647 unique instances respectively. In order to avoid the bias of the model towards majority instances, minority class was up-sampled. Data was normalized using a data scaling method before up-sampling.

3.2.2 Hybrid framework

Considering a feature selection approach, we designed a novel hybrid framework that consists of two components: a decision tree and an SNN. Decision trees acted as a coarse filter to select a reduced number of features in order to train the SNN. Decision trees with feature selection technique helps in interpretability and provides with a criterion for pre-screening the compounds in all three toxicity data sets while SNN helps improve the accuracy. Training with selected feature subspace reduces time and model complexity which leads to better interpretability [57].

3.2.3 Optimization

In model optimization, both components (decision tree and the SNN) of hybrid framework were conjointly optimized. Here the chosen objective function (AUC-ROC) of a neural network is dependent on its own parameters

as well as on parameters of the feature selection module. A held out predefined CV set was used to optimize both components of the hybrid model as discussed below:

Feature selection via decision tree

In feature selection module, we used an extremely randomized extra tree classifier (a type of decision tree) [42] with gini index, also called mean decrease impurity (MDI) [91, 131] to perform initial coarse filtering for features ranking [99]. As our aim is to tweak the number of selected features, so only those parameters were optimized that affect the process of selecting the features. The extra tree classifier has several optimization parameters but the most critical ones are (1) n_estimators that represent the number of trees in the forest and (2) threshold that limits the number of features selected during optimization [108]. All the features were ranked on the basis of the gini index. The higher the gini index value, the greater the importance of that feature in predicting a specific class [99].

TABLE 3.1: Hybrid model feature selection optimization.

(A) Threshold (Grid Search)

0.08×mean	0.09	0.1	0.2	0.3
0.4	0.5	0.6	0.7	0.8
0.9	1	1.1	1.2	1.3
1.4	1.5	1.6	1.7	1.8
1.9	2	2.1	2.2	2.3

Fixed Parameters

Epochs	20
Initialization Function	he-normal
Dropout	0.5
Activation	ReLU
Mini-batch	512

During feature selection process via threshold parameter optimization, parameters of the SNN were fixed as shown in Table 3.1. Because of the single parameter optimization, a grid search was applied on threshold value to achieve maximum AUC-ROC. A higher value of the threshold reflects a smaller number of features while a lower value, a large number of available features. The range of the threshold for grid search was set such that it can select a small number of features up to the all available features.

SNN hyper-parameters tuning

Once the reduced feature subspace was obtained in the feature selection process, then with the selected features, hyper-parameters were tuned for the SNN as shown in Table 3.2. Then a random search was performed for SNN hyper-parameters tuning because it is more efficient than the grid search in case of more parameters to optimize [15].

TABLE 3.2: Hybrid model shallow neural network optimization.

SNN hyper-parameter tuning (Random Search)

Epochs	10, 20, 40, 60
Initialization Function	He-Normal, He-Uniform Normal, Uniform
Dropout	0.0, 0.1, 0.2, 0.3, 0.4 0.5, 0.6, 0.7, 0.8, 0.9
Activation	ReLU, Sigmoid
Mini-batch	32, 64, 128, 512 1024, 2048, 4096, 8192

3.2.4 Toxicity classification

In classification, the CV and the training set were mixed together after obtaining all the optimized parameters. Optimized parameters were used to train the SNN for each individual toxicity task of all three data sets. A set of four similar SNNs were trained and their outputs were averaged to form

a more robust model to compute AUC-ROC. Complete pipeline of hybrid prediction framework is shown in Figure 3.1.

3.3 Results and discussion

In this section, we discuss the benchmark data sets, performance on three case studies and final test sets, investigate prediction potential of 2D descriptor, analyse the comparative landscape and explain feature interpretability of our classification results.

3.3.1 Benchmark data sets

NR and SR data sets were collected from Tox21 challenge [60]. NR assays were classified into subtasks pathways: (1) aryl hydrocarbon receptor, (2) androgen receptor-full, (3) androgen receptor-luciferase, (4) aromatase, (5) estrogen receptor alpha, (6) estrogen receptor alpha-luciferase, and (7) peroxisome proliferator-activated receptor gamma. SR assays were classified into 5 subtasks pathways:(1) antioxidant response element, (2) heat shock response/ unfolded protein response, (3) mitochondrial membrane potential, (4) DNA damage p53 pathway, and (5) geno-toxicity indicated by ATAD5. A separate benchmark data set for AM was also obtained [55]. It should be noted that for SR and NR, the data was pre-divided into training, held out cross validation (CV) and separate test sets by the Tox21 repository. For AM, no such division was given, so we divided it into train (60%), CV (20%) and test (20%) sets. Each set contains toxic and non-toxic compounds, the detailed description is provided in Table 3.3. It should be noted that Table 3.3 mentions the data setting after the cleaning and quality control.

TABLE 3.3: NR, SR and AM data division: train, cross validation (CV) and test sets.

Task	Train	Toxic/Non-Toxic	CV	Toxic/Non-Toxic	Test	Toxic/Non-Toxic
NR-AHR	7863	937/6926	268	30/238	594	73/521
NR-AR	9036	374/7950	288	3/285	573	12/559
NR-AR-LBD	8234	284/7950	249	4/245	567	8/559
NR-Aromatase	6959	352/6607	211	18/193	515	37/478
NR-ER	7421	916/6505	261	27/234	505	50/455
NR-ER-LBD	8431	415/8016	283	10/273	585	20/565
NR-PPARG	7883	193/7690	263	15/248	590	30/560
SR-ARE	6915	1040/5875	230	47/183	540	90/450
SR-HSE	7879	386/7493	263	10/253	594	19/575
SR-MMP	7071	1117/5954	234	38/196	530	58/472
SR-p53	8349	509/7840	265	28/237	601	40/561
SR-ATAD5	8775	317/8458	268	25/243	606	36/570
AM	3900	2097/1803	1300	699/601	1300	699/601

3.3.2 Prediction potential of 2D descriptors

Representation of chemical compounds in 2D form as a connection table is used to calculate their 2D descriptors. These descriptors are relatively easier to calculate and computationally less intensive. PADEL descriptor tool was used to calculate 1422 2D descriptors [152]. Primarily, prediction (classification) potential of these features was evaluated by performing a dry run using a neural network model on training data set of each task (2nd column of Table 3.3). Training sets for NR and SR were up-sampled and split into internal training/validation set with 70/30 ratio for examining the prediction potential of 2D features. Here, the CV and the test set (4th and 6th column of Table 3.3) were not considered, as the aim was not to build final prediction model rather to estimate the prediction power of the 2D features. AUC-ROC for each toxicity task of NR and SR was calculated using internal validation set as shown in Figure 3.2. It showed that 2D features have high potential to discriminate toxic and non-toxic compounds.

The highest AUC-ROC of 0.95 was obtained for mitochondrial membrane potential (MMP) task which belongs to SR panel while estrogen receptor (ER) from NR panel showed the lowest AUC-ROC of 0.74. Although AUC-ROCs

shown in Figure 3.2 is overestimated as it is on the internal validation set created from training set of Tox21, but it clearly shows a good performance of 2D descriptors as features in the prediction model. Thus, the results shown in Figure 3.2 confirmed that 2D descriptors alone have the potential to discriminate between toxic and non-toxic compounds for NR and SR signaling pathways. The same procedure was repeated for AM data set as well and the AUC-ROC is included in the figure. It should be noted that results shown in Figure 3.2 are not the final results, instead it shows that there is a prediction potential in 2D features for all the three toxicity tasks. The results shown in Figure 3.2 were obtained without any feature optimization and no hybrid framework of neural network and decision tree is used. This result could be improved with proper optimization as discussed in latter sections.

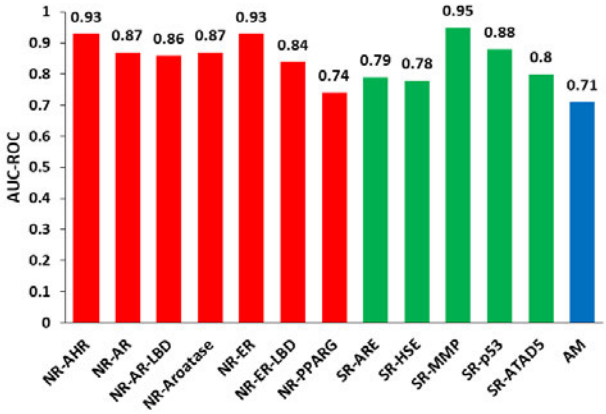

FIGURE 3.2: Area under the curve (AUC-ROC) for each of the three toxicity data sets was calculated on the internal validation set to evaluate the prediction potential of 2D features.

3.3.3 Case study-I: Series vs parallel optimization

Our hybrid model is composed of two main components, i.e. a shallow neural network (SNN) and a decision tree classifier (detail is given in methods section). Optimization of different parameters involved in two components of our hybrid framework is an essential phase to achieve a high accuracy. Parameters of both components (i.e. decision tree and SNN) of the hybrid

model could be optimized simultaneously (parallel mode) or one after another (series mode). A case study was conducted to compare the performance of series and parallel optimization on SR, NR and AM data sets. Estrogen receptor (ER) task of NR has shown the lowest accuracy in earlier studies by different groups in Tox21 challenge while Mitochondrial Membrane Potential (MMP) of SR has showed the best result [139]. Thus, NR-ER, SR-MMP and AM were selected for this case study. Similarly, two most critical parameters, one from the decision tree and the other one from the neural network were selected for optimization.

Threshold is an important parameter of a decision tree classifier that sets cut-off value for the selection of features and "dropout" refers to dropping out units in hidden layers of the neural network to prevent over-fitting. These two parameters one from decision tree and one from neural network were optimized in series and parallel mode using grid search technique considering AUC-ROC for the held out CV set (the train/CV/test split for the hybrid model is explained in method section as well as in Table 3.3) as an objective function. In addition to "threshold" in the decision tree classifier, the number of trees (n_estimator) [42, 108] were also tested in the range of 10 to 2000 on selected tasks, i.e. NR-ER, SR-MMP and AM. Figure 3.3(a) shows the behaviour of n_estimator with the AUC-ROC on CV set. Initially AUC-ROC showed ripples but then it became stable after 1000 number of trees. This suggests that n_estimator could be fixed to 1000 to make the model robust. The AUC in Figure 3.3(a) is computed using all available 2D features without considering any feature selection. Once the number of trees was fixed, threshold values were taken in grid search over [0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3] range while dropout was taken over [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]. During series mode, threshold was optimized first for the AUC-ROC of CV set which resulted in the optimized value of 1.6, 1.1 and 1.5 for NR-ER,

NR-MMP and AM respectively. Later, with these optimized threshold values selected, the dropout parameter was optimized in its search space. Optimum values for the dropout were 0.4, 0.2 and 0.1 with AUC-ROC 0.811, 0.949 and 0.864 for NR-ER, SR-MMP and AM respectively. Hence, these values were considered as optimized values for threshold and dropout. In parallel mode optimization, each combination of threshold and dropout were explored simultaneously and respective AUC-ROCs were calculated.

The parallel optimization resulted in several pairs of values (1.6,0.7), (1.2,0.4) and (1.3,0.3) for threshold and dropout with the best AUC-ROC of 0.789, 0.946 and 0.846 for NR-ER, SR-MMP and AM respectively. Results of series and parallel optimization are shown in Figure 3.3(b). In all the three cases, series and parallel optimizations perform very close to one another based on their AUC-ROC. However, the series mode achieved marginally higher AUC-ROC than the parallel mode. Additionally, the parallel optimization between two or more parameters from decision tree and SNN was found to be compute-intensive. This concluded to deployment of series parameter optimization across the components of hybrid framework (decision tree and SNN) to the build our predication model.

3.3.4 Case study-II: Do we really need a large set of 2D features?

In this case study, we wanted to know the number of 2D features which are sufficient for very good performance. This case study was inspired by a theorem called "curse of dimensionality" which states that beyond a certain point, the inclusion of additional features may lead to higher probabilities of error [138]. Moreover, there is a need of reducing the number of features to make the model simple and less compute intensive. The reduced number of features should be nearly optimum for a good performance and thus

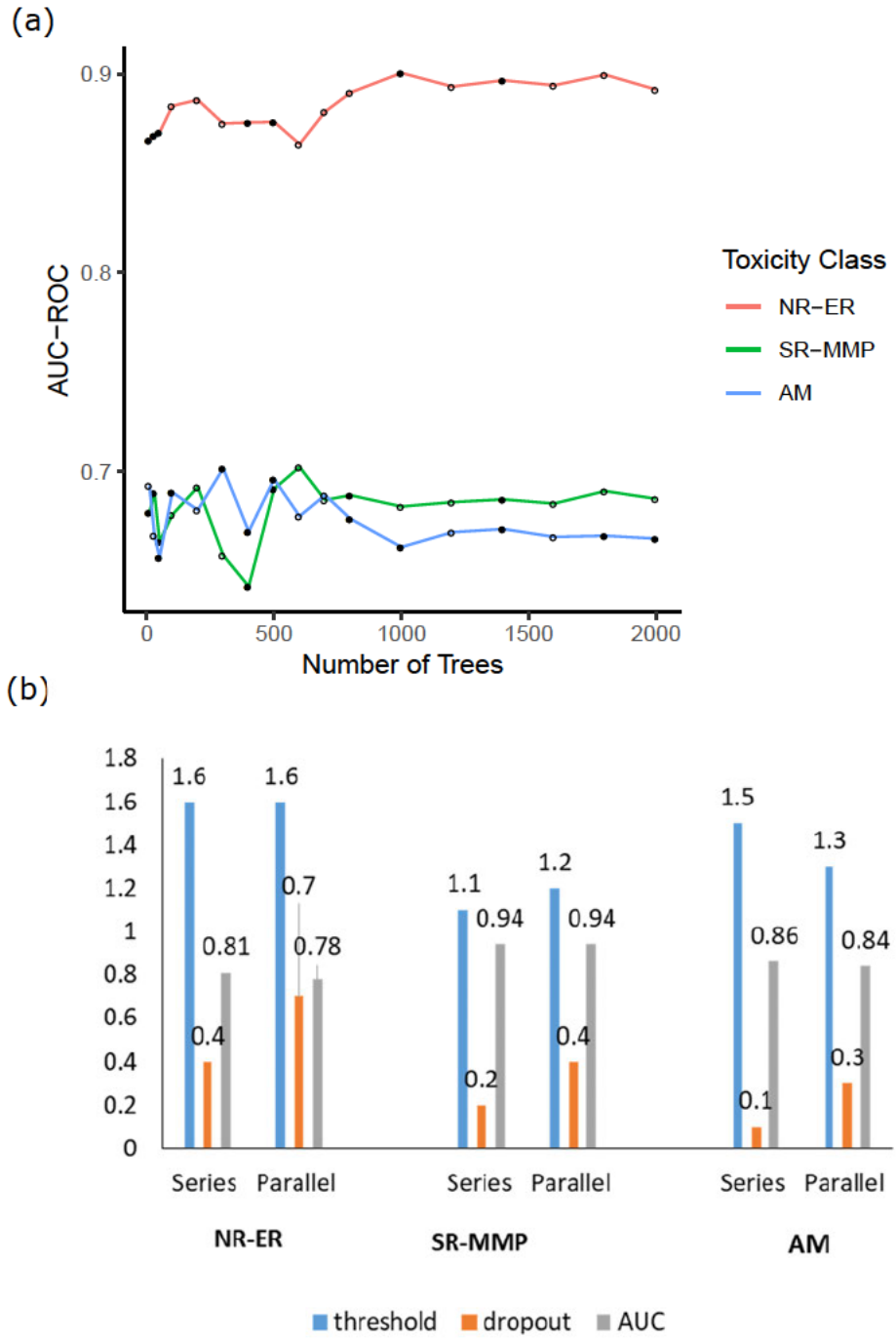

FIGURE 3.3: Variable parameters of hybrid learning model for series and parallel optimization to achieve betters results (a) number of trees in decision tree classifier vs AUC-ROC for NR-ER, SR-MMP and AM. (b) Selected values of threshold and dropout are shown in series and parallel optimization for NR-ER and SR-MMP task with AUC-ROC as an objective function.

may help in feature interpretability. It should be noted that in this work, the term "nearly optimum" is referring to the reduced number of 2D features

which can give better performance while searching over different values of thresholds in decision tree component of our hybrid model.

We plotted the AUC-ROC of all the three toxicity data sets (for external CV sets) against the number of features selected to know whether we can achieve better performance with fewer number of features. The threshold value of a decision tree classifier component was varied over a space of [0.0, 0.5, 1.0, 1.5, 2.0, 2.5]. The greater the threshold value, the lesser the number of features selected. The details of how the threshold changes the number of features selected is given in the methods section. A shallow neural network was trained for different number of selected features and the results are shown in Figure 3.4. In each case, we see that for better performance, we need not to train our model using all available 2D features, but instead a reduced number of features is sufficient to get the better performance. In case of AM in Figure 3.4, only 145 selected features achieved the highest AUC-ROC on CV set. If we further increase the number of features, the performance degrades. Similar trend can be seen for NR-ER and SR-MMP as well, although the performance does not degrade much with the increase of number of features. In these cases, It is a better choice to select the smaller number of features to make the model simple, less compute intensive and improve feature interpretability.

Considering case studies I and II, we developed a hybrid model (explained in the methods section) which enables the shallow neural network to select the small number of effective features (nearly optimum) to be trained on while jointly optimizing parameters of the shallow neural network and a decision tree.

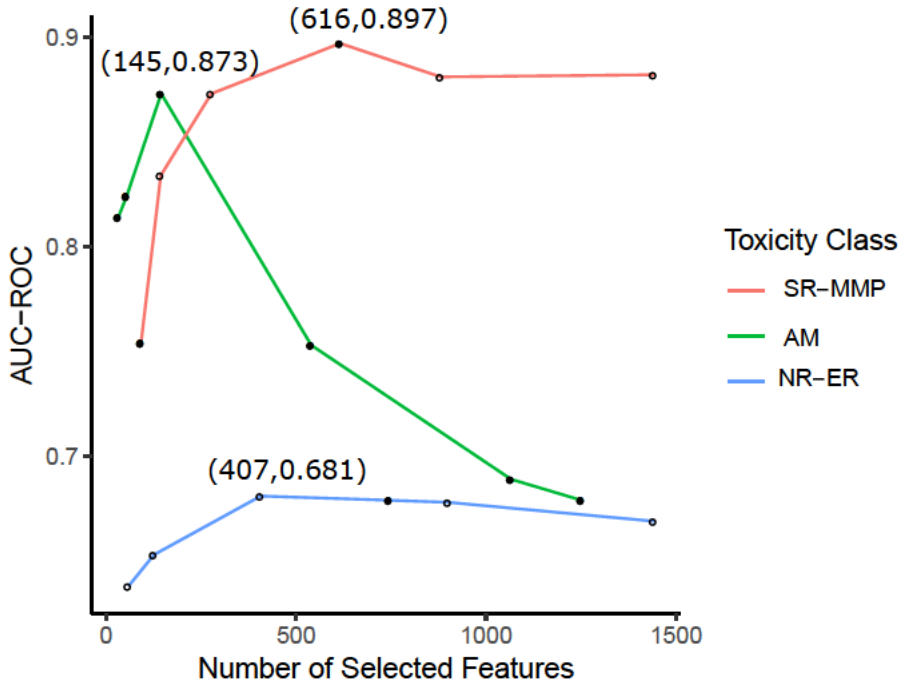

FIGURE 3.4: Selected number of effective 2D features used to build prediction models to achieve higher AUC-ROC for NR-ER, SR-MMP and AM toxicity classes.

3.3.5 Case Study-III: Why a shallow neural network?

The use of shallow neural network as one of the components in our hybrid model was motivated by "universal approximation theorem". It states that a shallow neural network (having one hidden layer with finite number of neurons) is a universal function approximator [26]. However, in practice a deep neural network performs better on large set of raw features. As the feature selection module of our hybrid model effectively selects a reduced number of features, we expect that a simple shallow neural network with a small number of neurons will be able to perform better or similar to a deep neural network. The main idea was to make a hybrid model in which a shallow neural network would extract relevant knowledge (effectively selecting the features to be trained on) from the decision tree classifier. In order to know if a shallow neural network will perform similar or better on selected features, we performed another case study in which the number of hidden

layers were varied from 1 to 5 for all three types of toxicity data sets as shown in Figure 3.5. It was found that a neural network with one hidden layer has relatively higher AUC-ROC on external CV set (on selected features) than a deep neural network for all three toxicity data sets. This concludes the implementation of a shallow neural network for better results.

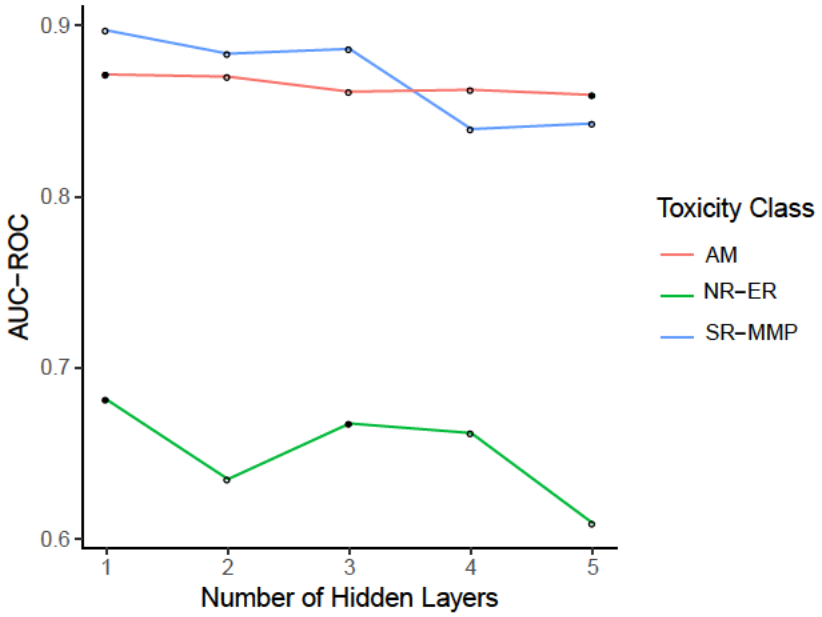

FIGURE 3.5: Classification performance with varying number of hidden layers used in neural network, for NR-ER, SR-MMP and AM toxicity classes.

3.3.6 Final test sets performance

The hybrid model was evaluated using the final test sets for each of the three toxicity data sets (7 tasks for NR, 5 for SR and 1 for AM) shown in Table 3.3 (6th column). The test set was not part of the training or validation, hence it was considered as blind testing of the prediction model. These parameters for the individual tasks were optimized on CV given in Table 3.3 (4th column). Table 3.4 shows AUC-ROC on test data (3rd column) as the final result for NR, SR and AM toxicity data sets. It also shows the number of features selected which is a subset of the total $\approx$ 1422 features for NR and SR while

$\approx$ 1249 for AM to build each model. Average AUC-ROC of our hybrid model on test set of NR is 0.836, on SR is 0.862 and on AM is 0.878.

TABLE 3.4: Performance on the final test sets of NR, SR and AM toxicity.

Task	Features Selected	Our Method AUC	Random Forest AUC	Support Vector Machine AUC
NR-AHR	270	0.921	0.907	0.889
NR-AR	284	0.743	0.638	0.730
NR-AR-LBD	365	0.881	0.800	0.702
NR-Aromatase	815	0.794	0.792	0.782
NR-ER	292	0.822	0.778	0.791
NR-ER-LBD	755	0.836	0.768	0.786
NR-PPARG	528	0.858	0.789	0.744
NR Average	472	0.836	0.782	0.775
SR-ARE	615	0.828	0.774	0.779
SR-HSE	1028	0.832	0.859	0.798
SR-MMP	685	0.958	0.978	0.916
SR-p53	223	0.875	0.847	0.810
SR-ATAD5	390	0.820	0.812	0.765
SR Average	588	0.862	0.854	0.814
AM	145	0.878	0.842	0.810
AM 5 Fold CV	145	0.879	0.831	0.815

In order to reduce the chance of error in the final result and to show robustness of our hybrid model, we performed ensemble averaging on all three toxicity data sets. An additional 5 fold cross validation is also performed on AM data set, which gave an AUC-ROC of 0.879. The average number of 2D features used to achieve this accuracy is 472 for NR, 588 for SR and 145 for AM. We also developed RF and SVM based models for all the three toxicity tasks and report the AUC-ROC on the final test data using 2D features as shown in Table 3.4 (4th and 5th column). Although in two cases (SR-HSE and SR-MMP), RF performed better, yet both the RF and SVM showed relatively less average AUC-ROC than AUC-ROC of our hybrid model.

3.3.7 Comparative landscape

On SR and NR data, our hybrid model was compared with the winning model of Tox21 challenge [97, 139]. For AM, we compared our results with

the state of the art methods [55]. We outperformed other methods in AUC-ROC for all the three toxicity data sets. The winning model of Tox21 Challenge is based on DNN and is trained on $\approx$ 273577 features for NR and SR data sets using a multitask approach. In this approach, DNNs up to four layers with thousands of neurons in each layer were tested. By harnessing the ability of a DNN to create intermediate complex features for prediction, they were able to achieve the average AUC-ROC of 0.826 for NR and 0.858 for SR on the final test set. Training of the model was computationally very expensive and took $\approx$ 10 minutes to train on NVIDIA Tesla K40 GPU. The large numbers of features used by the model made it very hard to interpret which features are playing vital role in decision making [97].

The second ranked team, AMAZIZ developed consensus models using associative neural network (ASNN) to achieve an average AUC-ROC of 0.816 for NR and 0.854 for SR. ASNN represents a combination of an ensemble of feed-forward neural networks and the KNN technique [7]. The information about the total number of features used and the training time is not reported [7]. The third ranked group, dmlab developed ensemble models with combining various fingerprinting tools using random forest and extra tree classifier (ET) to achieve an average AUC-ROC of 0.811 for NR and 0.850 for SR [12]. Post Tox21 challenge, other groups developed prediction models for NR and SR data sets [19, 46, 48]. Chemception developed convolutional neural networks (CNN) to predict toxicity using 2D images of compounds without explicitly calculating chemical descriptors and achieved average AUC-ROC 0.787 for NR and 0.739 for SR [46]. Capuzziet et al. used DNN with an ensemble of 2489 molecular descriptors to achieve a very good overall average AUC-ROC of 0.840 for both NR and SR [19]. SMILES2vec used deep recurrent neural networks that automatically learns features from the SMILES data and the reported average AUC-ROC is 0.799 for both NR and SR [48].

On AM data, the best performing method was reported Hansen et al. with

TABLE 3.5: Comparative analysis of different methods used for NR and SR toxicity prediction.

Name	**NR Average AUC-ROC**	**SR Average AUC-ROC**
Our Method	0.836	0.862
DeepTox [97]	0.826	0.858
AMAZIZ [7]	0.816	0.854
Capuzziet [19]	0.831	0.848
dmlab [7]	0.811	0.85
T	0.798	0.842
microsomes	0.785	0.814
filipsPL	0.765	0.817
Charite	0.75	0.811
RCC	0.751	0.781
frozenarm	0.759	0.768
ToxFit	0.753	0.756
CGL	0.72	0.791
SuperTox	0.682	0.768
kibutz	0.731	0.731
MML	0.7	0.753
NCI	0.651	0.791
VIF	0.702	0.692
Toxic Avg	0.659	0.607
Swamidass	0.596	0.593
Chemception [46]	0.787	0.739

an AUC-ROC OF 0.860. They compared with several commercial tools such as DEREK, MultiCASE and Pipeline Pilot with off-the-shelf methods such as SVM, Random Forests, KNN and Gaussian Processes [55]. The SVM achieved the highest performance using different types of constitutional, topological, geometrical, functional group count,and atom-centered fragments feature, though the exact number of features was not reported. Their model outperformed the DEREK and MultiACSE by extracting rich information from the training data. We outperform Hansen et al. SVM based model by achieving 0.878 AUC-ROC with only 145 2D features.

Our hybrid framework used reduced number of simple (easy to compute) 2D features to achieve the state of the art average AUC-ROCs. In contrast to other methods, we used shallow neural network (1 hidden layer, 10 neurons)

TABLE 3.6: Training time and model complexity of the top 5 models from Tox21 challenge and AM benchmark data set.

Task	Name	Method	Number of Features	Training Time	AUC-ROC
NR	Our Method	DT+SNN	472	≈ 1 min CPU	0.836
NR	DeepTox [97]	DNN	273577	≈ 10 min GPU	0.826
NR	AMAZIZ [7]	ASNN	NA	NA	0.816
NR	Capuzziet [19]	DNN	2489	NA	0.831
NR	dmlab [7]	RF + ET	681	≈ 13 sec CPU	0.811
SR	Our	DT+SNN	588	≈ 1 min CPU	0.862
SR	DeepTox [97]	DNN	273577	≈ 10 min GPU	0.858
SR	AMAZIZ [7]	ASNN	NA	NA	0.854
SR	Capuzziet [19]	DNN	2489	NA	0.848
SR	dmlab [7]	RF + ET	681	≈ 13 sec CPU	0.850
AM	Our Method	DT+SNN	145	≈ 1 min CPU	0.878
AM	Hansen et al. [55]	SVM	NA	NA	0.860

that makes the model computationally efficient and opens the avenue for interpretability. The average training time for our hybrid framework method is always less than a minute for all the tasks. Effectively reduced number of features selected in an optimization loop using decision tree and SNN improves the model to achieve the highest accuracy. Table 3.5 shows the comprehensive comparison of our model with others for SR and NR. In addition to accuracy we also compared our method on model complexity ground with top 5 models in Tox21 challenge. Table 3.6 shows methods, training time and number of feature for top 5 models of Tox21 challenge and for AM benchmark data set. DeepTox model achieved AUC-ROC close to our method but it used DNN with 273577 features. Table 3.5 and Table 3.6 jointly demonstrate the performance of our hybrid framework on accuracy and complexity verticals. We achieved the highest accuracy while utilizing the least computing resources with introducing interpretability of the chemical descriptors in terms of the decisions made by model.

3.3.8 Regression modeling of additional toxicity data sets

In order to verify the general applicability of 2D features predictive power and robustness of our model, we performed additional experiments using four new category of toxicity data. These data sets namely, 96 h fathead minnow LC50 data set (LC50 set), 48 h Daphnia magna LC50 data set (LC50-DM set), 40 h Tetrahymena pyriformis IGC50 data set (IGC50 set), and oral rat LD50 data set (LD50 set) were obtained from Wu and Wei while the setting (train test split) was kept the same as given in their recent work on toxicity [145]. In this work the authors used various types of approaches to verify the predictive power of element specific topological descriptors (ESTD), auxiliary molecular descriptors (AUX) and a combination of both for the four types of toxicity data sets. They named their predictive model as TopTox. In order to verify the predictive power of 2D features with our model, we compared our results with the single task deep neural network (ST-DNN) approach of TopTox as our hybrid model is also based on single task. We consider each toxicity task separately and independently. In 11 out of 12 single task cases, we obtain better squared Pearson correlation coefficient than TopTox models as shown in Table 3.7. In IGC50 data set, our model achieves squared Pearson correlation coefficient value of 0.805 which is slightly better than the state of the art.

TABLE 3.7: Single task (ST) performance comparison of our method with TopTox for various toxicity data sets .

Method	TopTox ST-DNN			Our Method
Descriptors	ESTD	AUX	ESTD+AUX	2D
R^2 for IGC50	0.708	0.678	0.749	**0.805**
R^2 for LC50-DM	0.446	0.430	0.459	**0.616**
R^2 for LC50	0.675	0.598	**0.692**	0.678
R^2 for LD50	0.601	0.593	0.614	**0.615**

3.3.9 Feature interpretability

Machine learning models predominantly behave as "black box" which usually do not provide any explanation of the decisions made. In this study, we tried to interpret the outcome in terms of feature importance. For this, physico-chemical 2D descriptors calculated using PADEL package were used to build predictive model. These features were ranked based on their gini index in the decision tree classifier. Gini index for individual toxicity task (7 NR tasks and 5 SR tasks) was calculated and added up to get the cumulative gini index to assign a single score to each feature across NR and SR toxicity data sets.

Figure 3.6(a) shows the cumulative gini index of 1422 features for NR and SR data sets. These features are arranged in a descending order of their gini index, the top 29 features in this list showed vertical drops in their gini index values, thus suggests substantial difference in their importance while others showed small variances (shown as the break point in Figure 3.6(a)). Similarly, average rank of each feature was calculated across NR and SR data sets. The relation between the average rank and the cumulative gini index score is shown in Figure 3.6(b). The Proportional behaviour between these parameters confirms a consistent nature of features as per their importance score among all toxicity tasks of NR and SR data sets. Later, the top 29 gini index descriptors detected in gini index plot were identified separately. These 29 features with their average ranks are shown in Figure 3.6(c). Here, it is observed that "path count descriptor" class is the most abundant class in the top features list. The top 3 features showed average rank below 10 are: (1) pipC10 (2) pipC9 and (3) pipC8 and their average ranks are 3.91, 9.91 and 6.41 respectively (marked with red stars in Figure 3.6(c)). These 3 features from the path count descriptor class played the most critical role in

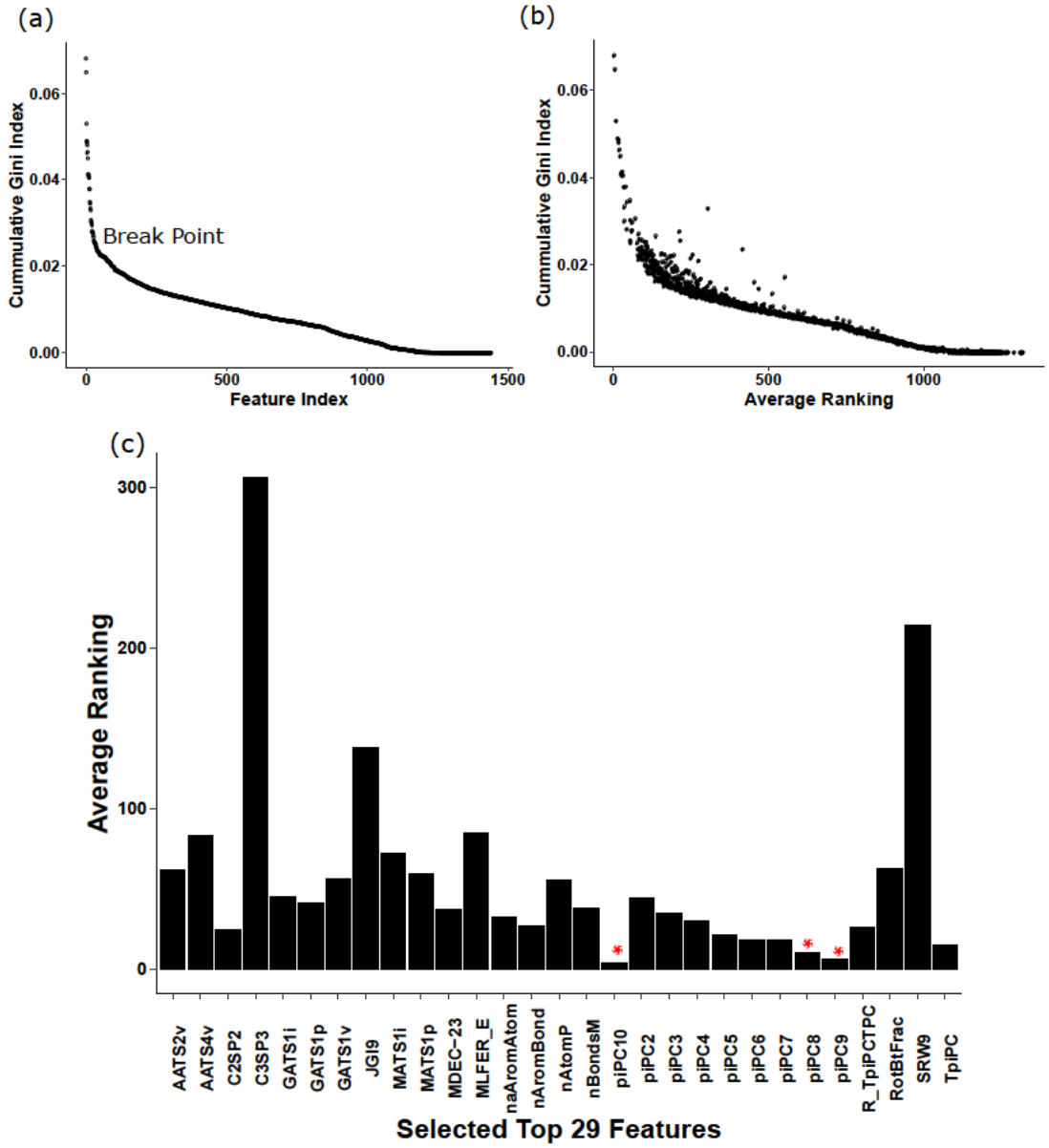

FIGURE 3.6: (a) Cumulative gini index score of 1422 features across 12 NR and SR toxicity data sets (b) Average ranking of 1422 features against cumulative gini index score in all 12 NR and SR data sets (c) Ranking of top 29 features arranged in alphabetical order, top 3 features piPC10, piPC9 and piPC8 showed average rank below 10 and are marked with red star.

classifying the molecule as toxic and non-toxic for NR and SR data sets. Relatively higher importance of these descriptors made them appropriate for coarse initial screening of molecules. In order to observe the importance of these top features, piPC10 values are plotted against all molecules. Figure 3.7

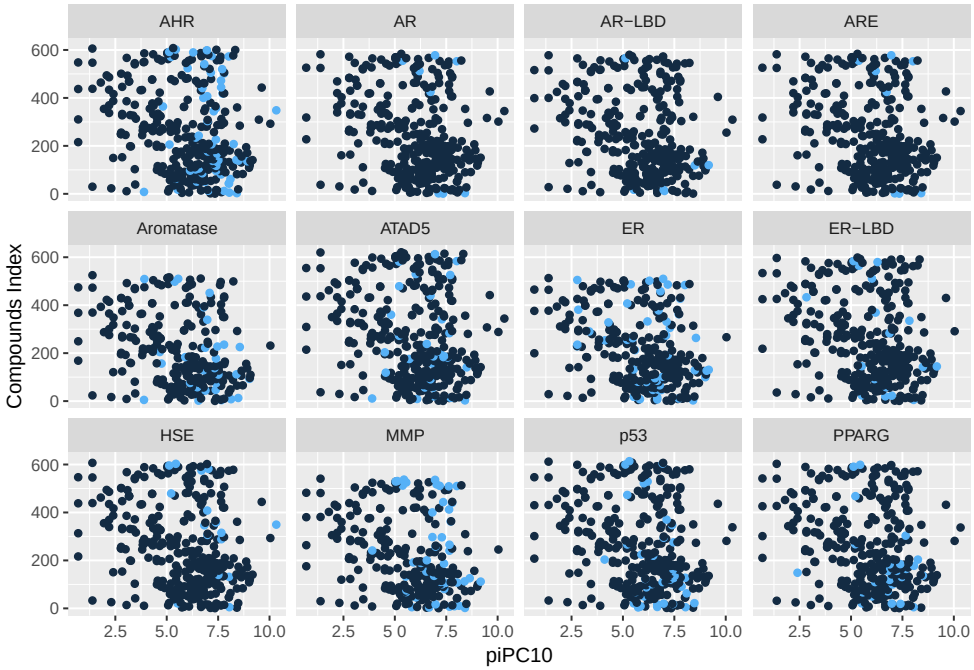

FIGURE 3.7: Classification of toxic and non-toxic molecule based on cut-off values of pipC10 features derived from decision tree classifier. Toxic molecules are shown in light blue while non-toxic are represented dark blue dots.

shows piPC10 values for toxic and non-toxic molecules, light blue circles represent toxic molecules while dark blue represent non-toxic molecules. As it can be clearly observed in Figure 3.7 that toxic molecules make cluster in certain range of piPC10 value leaving a large area as safe zone (non-toxic). This shows the classifying property of piPC10 between toxic and non-toxic molecule around a fixed value. The decision tree classifier used for feature importance in the presented hybrid framework has assigned a cut-off value to each feature at every node of the tree. These cut-offs of top features could be used as discriminating planes for toxic molecules.

Feature cut-offs in decision trees are defined as the values that divides the population in the highest ratios. Each tree has its own cut-off for each feature. Average cut-off values across these 1000 trees grown in building model for 3 most important features were calculated. It is suggested, that any molecule has value for these descriptors less than the respective cut-off would have more possibility to be found in the toxic spectrum. Top 3 features with their

respective cut-offs were combined together to improve the discriminating power. Molecules that have values of these top 3 features less than their respective cut-offs are taken in one group. Table 3.8 shows that this group has less than 0.03 fraction toxic molecules to the total available toxic molecules for all 12 tasks of SR and NR while on an average 0.50 fraction non-toxic molecule to the total non-toxic molecules for respective classes. This suggest that combined criteria for piPC10, piPC9 and piPC8 could be used to find the probability of a given molecule to be toxic or non-toxic for SR and NR. Similarly, individual cut-offs of pipC10, pipC9 and pipC8 are 5.29, 5.10 and 5.0 for AM dataset. Later, these features and their respective cutoffs were used cumulatively on AM data set as we done for NR/SR dataset. Here, again toxic molecules have low fraction 0.05 below the combined cut-off while non-toxic molecules have 0.10 fraction. Although the fractional discrimination between toxic and non-toxic molecules on AM data is weaker that NR and SR dataset, but it clearly shows that piPC10, piPC9 and piPC8 can be used to determine the initial Ames mutagenicity probability of any new molecules. Thus, these features could be used as initial indicators during molecule assessment.

TABLE 3.8: Toxic and non-toxic molecule fraction using combined criteria of pipC10, pipC9 and pipC8 for NR, SR and AM.

Task	Toxic Molecule Fraction	Non-Toxic Molecule Fraction
NR-AHR	0.01	0.48
NR-AR	0.01	0.47
NR-ARLBD	0.00	0.55
NR-Aromatase	0.02	0.55
NR-ER	0.02	0.43
NR-ERLBD	0.01	0.44
NR-PPARG	0.01	0.48
SR-ARE	0.00	0.44
SR-HSE	0.01	0.45
SR-MMP	0.02	0.51
SR-P53	0.00	0.49
SR-ATAD5	0.01	0.51
AM	0.05	0.10

3.4 Conclusion

The joint optimization of feature selection by a decision tree and classification using a shallow neural network enabled us to achieve the highest average AUC-ROC of 0.836 for NR, 0.862 for SR and 0.878 for AM which are better than the state of art results. In our approach, a shallow neural network is allowed to choose its feature set from the complete feature space. These features would be used for the training to achieve high accuracy on cross-validation data without any field expert intervention. The model complexity as well as the training time is reduced by a large extent. Instead of utilizing thousands of features, only selected reduced number of important features made the model more comprehensible. This hybrid method reduces the dimensionality curse by using only reduced effective features. One of the aims of this study is to achieve comparable toxicity prediction results by using simpler machine learning model. This opens an avenue to highlight the insight of a prediction process in order to understand the specific problem in comprehensive manner.

In our hybrid framework, a coarse filter for feature selection in the form of a decision tree prior to a classification model based on gini-index was applied. Decision trees helped in feature analysis using cumulative gini index. This was performed to find global relevance of features across toxicity tasks. Additionally, individual rankings of these features were used to calculate average ranking of each feature. The correlation between the average rank and cumulative gini index suggests the similar importance pattern of these features among diverse toxicity tasks. Eventually, the top features based on the gini index were plotted and 3 features were observed (1) pipC10 (2) pipC9 and (3) pipC8 have average ranks below 10. They belong to single descriptor class called path count. There individual cut-offs at first node were extracted

from 1000 decision trees and average score was used to observe the classification potential of these top features on toxic and non-toxic compounds. piPC10 was initially plotted for all toxicity tasks and clear discrimination was observed between toxic and non-toxic molecules for SR and NR. Further, piPC9 and piPC8 were combined with piPC10 to design a cumulative criteria for classification. The cumulative criteria indicates a safe zone where the probability of finding toxic compounds is less than 0.05%. This can allow users for initial screening of toxic and non-toxic compounds based on only piPC10, piPC9 and piPC8 scores.

We conclude, our hybrid model of a decision tree and an SNN can be used for toxicity prediction or any similar tasks to achieve high accuracy in comparably lesser time and lesser resources. This technique enabled us to use certain features for rapid and prior toxicity estimation. In addition, better performance than any other existing methods were achieved for these toxicity classes. We believe that our hybrid framework can be applied on various other toxicity or related tasks to achieve high accuracy and to obtain interpretable behaviour of the descriptors. It will also be interesting to apply a coarse feature selection method using a heuristic approach to improve feature space optimization. Following are the main concluding points of our study.

- 2D features if selected effectively, have the power to predict NR,SR and AM Toxicity with the state of the art accuracy.
- We propose a hybrid algorithm which effectively select a feature subset of 2D features for training.
- The use of significantly reduced number of effective 2D features helps in interpretability.
- The computational complexity of AM, NR and SR toxicity prediction can be reduced to a great extent with our hybrid algorithm.

- Using features interpretation, we help the chemists effectively screen out the toxic compounds with just three features.

In this chapter, we developed and demonstrated a novel hybrid framework based on decision tree and a shallow neural network. Using this hybrid framework, we then build prediction model for nuclear receptor (NR) toxicity, stress response (SR) toxicity and ames mutagenicity (AM) toxicity. The software code along with data for this chapter can be found at `https://github.com/Abdulk084/HybridTox2D`. In the next chapter, we present the idea of average ensembling of heterogeneous predictors for quantitative toxicity tasks.

Chapter 4

Average ensemble of heterogeneous predictors

This chapter is published as pre-print in the research square as follows.

- Karim, A., Riahi, V., Mishra, A., Dehzangi, A., Newton, M. H., Sattar, A. (2019, December). Quantitative Toxicity Prediction via Ensembling of Heterogeneous Predictors

In this chapter, we study quantitative toxicity prediction and propose a machine learning model for the the same. Our model uses an ensemble of heterogeneous predictors instead of typically using homogeneous predictors. The predictors that we use vary either on the type of features used or on the deep learning architecture employed. Each of these predictors presumably has its own strengths and limitations in terms of toxicity prediction. We aim to design an ensemble model utilizes different types of features and architectures to obtain better collective performance that could go beyond the performance of each individual predictor. We use six predictors in our model and test the model on four standard quantitative toxicity benchmark data sets. Experimental results show that our model outperforms the state-of-the-art toxicity prediction models in 8 out of 12 accuracy measures.

Our experiments show that ensembling heterogeneous predictor improves the performance over single predictors and homogeneous ensembling of single predictors. The results also show that each data representation or deep learning based predictor has its own strengths and weaknesses, thus employing a model ensembling multiple heterogeneous predictors could go beyond the individual performance of each data representation or each predictor type.

4.1 Introduction

The toxic concentration of the compounds is measured by *endpoints* measuring experiments. Toxicity of compound could vary for different individual depending on their age, gender and body weight. Thus different toxic indicators are devised to measure the toxicity on the population. Population based toxicity measures are considered in this study using an ensemble of deep learning methods. Toxicity estimation, similar to other attributes of chemical compounds, also calculated using sophisticated experimental techniques on *in-vivo* or *in-vitro* models. However, these techniques are heavily time-consuming and cost-intensive. It also raises ethical concerns because of the involvement of animals. To address these issues, *in-silico* methods (computer-aided methods) have recently attracted much attention due to their cost and time efficiency. There exist many *in-silico* methods, but the quantitative structure activity/property relationship (QSAR/QSPR) method is one of the most successful ones.

In this chapter, we propose a model comprising an ensemble of heterogeneous predictors (*HPE*). The *HPE* uses six different deep learning methods, thus called predictors in the chapter hereafter, to predict the regression values of four benchmark quantitative toxicity data sets. These predictors are:

(1) fully connected physicochemicalical (FCPC) (2) fully connected physicochemicalical extended (FCPCe) (3) convolution 1D SMILES (C1DS) (4) convolution 2D fingerprints (C2DF) (5) molecular graph convolution (MGC) and (6) molecular weave convolution (MWC). FCPC and FCFCe are fully connected neural networks, C1DS and C2DF are two types of convolutional neural networks, and MGC and MWC are two types of graph convolutional networks. In our *HPE* model, we ensembled the outputs of these predictors to achieve the overall performance. It should be noted that these predictors vary (heterogeneity) on either class, architecture or feature levels as shown in Table 4.1. For instance, FCPC and FCPCe vary on feature level only. They both use numerical features (different in number only) but share the same architecture. C1DS and C2DF vary on the architecture and the feature level both. C1DS uses SMILES directly as input while C2DF converts SMILES into fingerprints first. MGC and MWC also vary on the architecture and the feature level. The details of these predictors are given in the methods section.

Thus by introducing heterogeneity in each predictor with respect to the others, we were able to make a single model that utilizes different types of features and architectures to obtain collective performance that could go beyond the individual performance of single predictor type. Toxicity measures used in this study for different data sets are: (1) IGC_{50} (2) LD_{50} (3) LC_{50} and LC_{50}-DM. On four benchmark toxicity data sets, the proposed method outperformed in 8 out of 12 cases of evaluation metrics compared to the state-of-the-art method. Toxicity measures where we performed better than any existing techniques are: IGC_{50}, LD_{50} and LC_{50}-DM. Moreover, it also showed that *HPE* model significantly improves the performance over individual predictors and their homogeneous ensembling for all four toxicity data sets.

4.2 Materials and methods

In this section, we first overview four data sets used in this chapter. We also present the implementation details of the individual predictors of our *HPE* model used in this chapter.

4.2.1 Data sets

Mathematical representation of toxicity is the simplest way to understand the unwanted effect of a given compound on cells, tissues and living organism. These mathematical formulas for toxicity are based on two factors: (i) dose and (ii) time of exposure. These two factors combine and formulate quantitative toxicity of a compound. Quantitative toxicity, due to its mathematical characteristics, is not only easy to illustrate but is also proven to be compatible with supervised learning prediction algorithms. Four different quantitative toxic data sets are used in this study. These data sets are labelled as: LC_{50}, LC_{50}-DM (Lethal Concentration, 50%), LD_{50} (Lethal Dose, 50%) and IGC_{50} (Inhibition Growth Concentration, 50%) with different toxicity measure indicators on the population.

All these endpoint measures are being used in toxicology for estimating the toxicity behaviour of any given chemical compound on a given population of the organism. LC_{50} and LD_{50} are the concentrations of the compound that kills half members of the tested animal population. Here, the LC_{50} data set is showing the toxicity for a given compound on fathead minnow, a species of temperate freshwater fish after 96 hr exposure. LC_{50}-DM data set records the concentration of test chemicals in water in milligrams per litre that cause 50% population of *Daphnia maga* to die after 48 h. LD_{50} data set has the lethal dose data for killing 50% rat population when the given compound is administered orally. LD_{50} depends on the route of administration: oral administration could cause less toxicity than intravenous route.

IGC_{50} data set shows the concentration of the chemical compound to arrest the growth of Tetrahymena pyriformis when exposed for 40 hr. In addition to the concentration, these measures also depend on the duration of exposure of a given organism to the compound. LC_{50} data set shows the LC_{50} record on fathead minnow species after 96 hr duration of exposure while LC_{50}-DM data set shows the LC_{50} values on *Daphnia magna* after 48 hr of exposure. IGC_{50} data set shows the IGC_{50} record on *Tetrahymena pyriformis* after 40 hrs of exposure.

The units of LC_{50}, LC_{50}-DM, IGC_{50} end points are $-log10(T\ \ mol/L)$, where T represents corresponding end point. For LD50 set, the units are $-log10(LD_{50}\ mol/kg)$. Pre-processed data sets (in the form of SMILES strings and activity measures) are sourced from Wu and Wei while the original repository is available at `http://cfpub.epa.gov/ecotox/`, `http://cfpub.epa.gov/ecotox/` and `http://chem.sis.nlm.nih.gov/chemidplus/chemidheavy.jsp`, `http://chem.sis.nlm.nih.gov/chemidplus/chemidheavy.jsp`. These data sets have different sizes ranging from hundreds to thousands. For instance, LC_{50}-DM contains 353, LC_{50} contains 823, IGC_{50} contains 1792 and LD_{50} contains 7413 molecules.

4.2.2 Methods

In our *HPE* model, we ensembled six various deep learning based predictors to achieve the overall performance. It should be noted that these predictors vary (heterogeneity) on either class, architecture or feature levels as shown in Table 4.1. We used an ensemble averaging method to combine the output of each individual predictor and to compute the final output of our model. We refer the reader to [125] for the concepts and mathematics of deep learning and neural networks. In the rest of this section, we explain these predictors in terms of their classes, architectures and features. The FCPC and FCPCe vary

on feature levels only, C1DS, C2DF and MGC, MWC vary on architectures and feature levels both.

TABLE 4.1: Predictors with their attributes.

Predictor Name	Class	Architecture	Features
FCPC	Fully Connected Deep Neural Network	Standard feed-forward	2D physicochemical features [153]
FCPCe	Fully Connected Deep Neural Network	Standard feed-forward	2D+3D physicochemical features [103]
C1DS	Convolutional Neural Network	1D Convolution	SMILE Strings
C2DF	Convolutional Neural Network	2D Convolution	Fingerprints
MGC	Geometric Neural Network	Graph Convolution	Molecular Graph Coordinates (Atom Features)
MWC	Geometric Neural Network	Weave	Molecular Graph Coordinates (Atom and Pair Features)

Fully connected physicochemical (FCPC) and fully connected physicochemical extended (FCPCe)

The first challenge in any machine learning algorithm is selecting a specific representation of the training data. The most common type of representation is numerical value based features. Usually, for numerical features, a standard fully connected neural network is used. A neural network that has each unit of each layer connected to all the units of the next layer is termed as a *fully connected neural network* (FCNN). FCNN operates on a fixed shape input by passing information through multiple non-linear transformations. The first two predictors of our method (FCPC and FCPCe) use standard fully connected neural networks as shown in Figure 4.1. FCNN in both FCPC and FCPCe predictors consists of 10 layers with 1000 neurons in each layer. The final layer consists of a single unit with a linear function. The non linear activation function of the sigmoid is used after each layer except the final layer. A dropout value of 0.5 is used after each layer. The learning rate was kept $5e^{-6}$ with a batch size of 32. Optimization was performed using the ADAM optimizer [79]. Both of these predictors are built using a Keras deep learning framework on a system with NVIDIA Tesla K40 GPU [25].

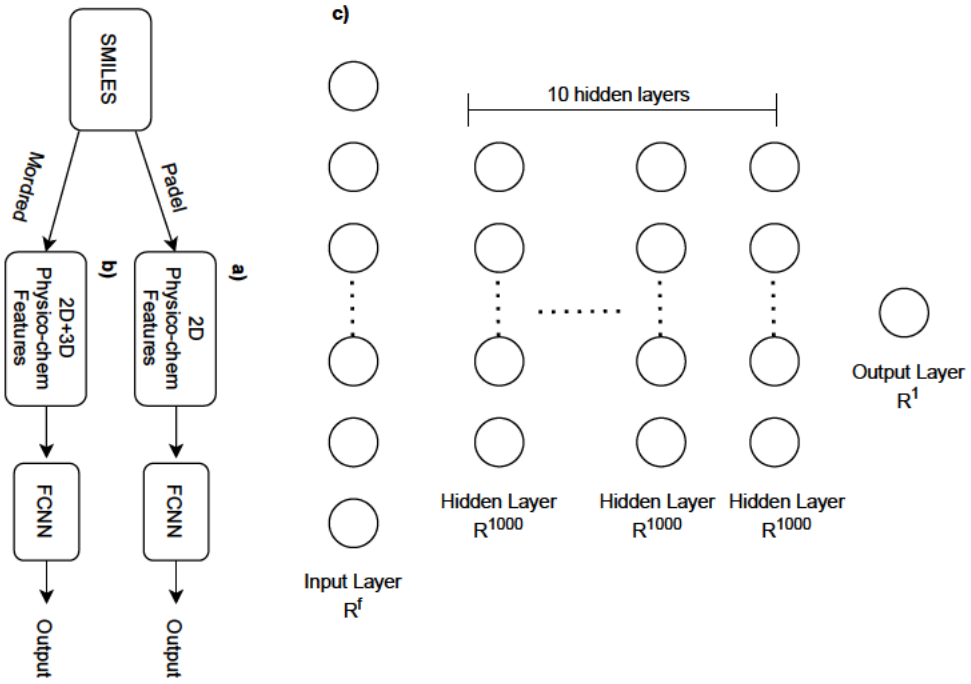

FIGURE 4.1: a) FCPC predictor which takes only 2D features computed using PADEL descriptor software. b) FCPCe predictor which takes both 2D and 3D features computed using Mordred descriptor software. c) FCNN used with physicochemical and physicochemical extended features. Each unit of one layer is connected to all units of the succeeding layer. R^f shows the input features where f takes values of 1148 for FCPC and 1826 for FCPCe.

In the FCPC component of our model, we used only 2D pysicochemical features. These 2D pysicochemical features are numerical in nature and are computed using PADEL software [153]. Out of total 1444 features, we computed 1148 features because PADEL fails to compute features for large molecules due to time and memory constraint. For FCPCe component, we extended the feature set to 3D (total of 1826) using Mordred [103].

Convolution 1D SMILES (C1DS) and convolution 2D fingerprint (C2DF)

A *convolutional neural network* (CNN) is a special type of neural network for the image data. CNNs can extract low level features from images and compute more complex features as we go deeper into the networks [135]. Variants of CNN like Inception, Alexnet and Resnet have been developed and

employed as highly accurate image classification models [56]. 1D convolution is a special type of convolution which uses convolution operation over one dimension such as sequence or time series data as opposed to 2D convolution which works for 2 dimensional data such as images. It should be noted that there is another type of specialized neural network called recurrent neural network (RNN) which also works for sequential data but suffers from high computational cost as compared to 1D convolutional neural network [58].

We developed an 1D convolutional neural network (C1DS) as a third predictor of our model. C1DS was trained directly on SMILES strings of the molecules. SMILES is a chemical language that describes the chemical structure of a molecule in a string of characters [143]. There is a special grammar for SMILES strings. Different characters represent atoms or bonds between the atoms. For instance, a small c represents aromatic carbon whereas capital C represents aliphatic carbon. To represent a single or double bond between atoms, special characters like "=" and "-" are used between the atom characters. An example of a SMILES string is "COc(c1)cccc1C#N", which represents 3-cyanoanisole. It should be noted that SMILES strings are canonically normalised before feeding into C1DS predictor.

The architecture of CIDS predictor is shown in Figure 4.2a. The SMILES strings of molecules are of different lengths. We pad each smile with '0' and make them all equal to the length of the longest smile in a particular data set. The longest SMILES string is 52, 103, 75 and 181 for IGC50, LC50-DM, LC50 and LD50 respectively. Each character of the SMILES is encoded into a numerical value. Thus we obtain equal length vectors of each SMILE to be used in convolution 1D predictor. This fixed dimensional feature vector goes into the embedding layer of convolution 1D predictor. Each integer value of the fixed sized vector is embedded into 400 dimensional vector, thus creating a matrix of the shape *[maximum length of a SMILES string, 400]*. This matrix is

trained along with the rest of the model training. After the embedding layer, we applied three 1D convolution layers, each with 192 filters with the size of 10, 5 and 3 respectively. A ReLu activation function and batch normalization is used after each convolution layer. After flattening out, a fully connected dense layer with 100 units followed by a ReLu activation and dropout of 0.5 is applied. The output layer is a single neuron with a linear activation function. It should be noted that the learning rate, batch size and optimization algorithm are kept the same as of the FCPC and FCPCe components. We used Keras with NVIDIA Tesla K40 GPU for building convolution 1D SMILES predictor [25].

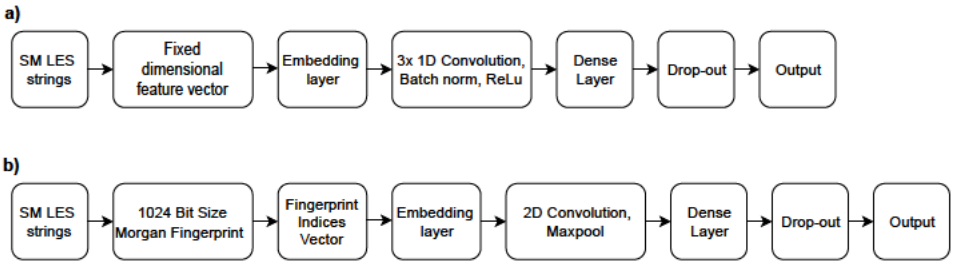

FIGURE 4.2: a) Architecture of convolution 1D SMILES (C1DS) predictor. b) Architecture of convolution 2D Fingerprints (C2DF) predictor.

As described before, 2D convolutional neural network is a special type of neural network used for 2 dimensional data such as images. This predictor (C2DF) of our model (*HPE*) is based on 2D convolutional neural network inspired by FP2VEC model [66] as shown in Figure 4.2b. Each SMILES string of a molecule in all 4 data sets is first converted into their respective fingerprint. We used RDKit to convert the SMILES strings into 1024 bit Morgan fingerprints of a radius 2 [83]. Fingerprints are bit strings composed of 0's and 1's. The position at which there is 1 represents a chemical feature defined by a specific design of fingerprint [123]. We computed a fingerprint indices vector by only taking those indices of the fingerprints with the value "1". The length of the fingerprint indices vector is computed to be 92 as the maximum number of "1s" in 1024 bit fingerprint of any molecule is 92 for

all the four data sets under consideration. Those molecules with less than 92 "1s" in their 1024 bit fingerprint were padded with zero at the end. Thus we obtain fixed length vectors called fingerprint indices vector of each 1024 bit size fingerprint. This fixed length fingerprint indices vector goes into the embedding layer of C2DF predictor. Similar to C1DF, each integer value of the fingerprint index vector is embedded into 400 dimensional vector, thus creating a matrix of the shape *[92, 400]*. This matrix is trained along with the rest of the model training as similar to C1DS.

Unlike C1DS, in C2DF we used 2D convolution layer followed by max-pool layer. The output of the embedding layer in C2DF is fed into a 2D convolutional layer. The number of filters in this layer is chosen to be 2024 each with a size of *[4, 400]*. A maxpool layer with a kernel size of 89 followed by a dense layer with 100 units in it is applied. Rest of the hyper-parameters were kept same as that of C1DS. It should be noted that parameters like embedding size (which is chosen to be 400 for both C1DS and C2DF), filter/kernel sizes, number of filters, learning rate, batch size and optimizer type are chosen to be inspired from the previously published research [45, 47, 66, 68, 70] and initial experimentation.

Molecular graph convolution (MGC) and molecular weave convolution (MWC)

Molecular Graph Convolution (MGC) and Molecular Weave Convolution (MWC) belong to the third category of our developed predictors. They use similar features and classes but different architectures as given in Table 4.1. As the name suggests, Graph Convolution Networks (GCN) are inspired from the convolutional neural network by redefining them for graphs instead of typical pixel based images [80]. Typical neural networks like fully connected, recurrent and convolution neural networks extract latent representation from Euclidean space but they fail to work efficiently on graph

data applications [148]. For instance, in the space of chemistry, a molecule can be represented in the form of a molecular graph, where the nodes represent the atoms and the bonds are represented by edges in the graph as shown in Figure 4.3.

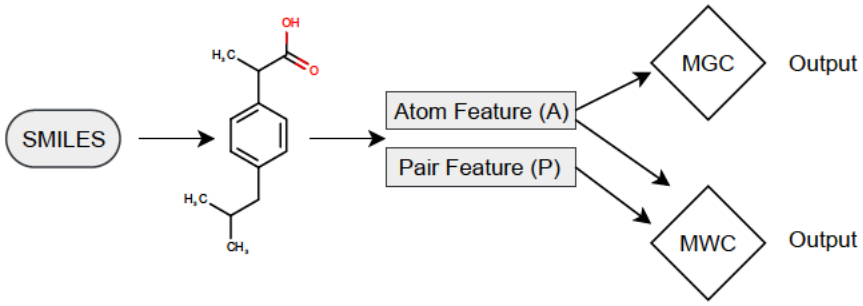

FIGURE 4.3: SMILES string is converted into 2D Molecular graph of Ibuprofen. The unmarked vertices/nodes represent carbon atoms whereas the edges represent bonds in the graph. Atom (A) and pair (P) features are computed from the molecular graph. MGC is trained only with the atom features whereas MWC takes both atom and pair features together.

MGC and MWC are graph convolution neural networks trained on molecular graphs as input data. Conceptually, MGC only requires the structure or graph of a molecule and a vector of features for every atom (A) that describes the surrounding local chemical environment whereas MWC requires pair features (P) as well. SMILES of each molecule is converted into their respective molecular graphs using RDKit [83]. Atom features such as atom type, chirality, formal charge, partial charge, ring sizes, hybridization, hydrogen bonding and aromaticity are computed using deepchem library [77, 147]. Pair features include bond type, graph distance, and same ring as described in previous papers [77, 147]. MGC predictor applies convolution layers to the central and its surrounding atoms, thus capturing the local chemical environment. As opposed to MGC, MWC predictor applies global convolutions to central atom along with all other atoms in a molecule while taking into account their corresponding atoms pair features as well. We used MGC and MWC as our two predictors for our *HPE* model from deepchem library

with default settings [113]. The specific architecture details of both MGC and MWC can be found in the original molecular graph convolution paper by Google and deepchem open source library [77, 113, 147].

4.3 Results

We report the prediction results of the proposed *HPE* model. In order to evaluate our predictors, we used K-fold cross validation with $K = 10$. Data was split into 10 equal random parts. One part was kept for testing while other 9 parts were used for training. This process was repeated for 10 times. All the results shown later in the section represent an average value of 10 fold cross validation. We have compared the proposed model with each of the single predictor (i.e., FCPC, FCPCe, C1DS, C2DF, MGC and MWC) used in our *HPE* model, and also with their homogeneous ensembles. The homogeneous ensembles (*Hom*) of each predictor are obtained by ensembling each individual predictor with itself six times. We also have compared the proposed model against known best-performing models in the literature: TopTox [145], Hybrid2d [68] and various methods used in the development of TEST software [96].

4.3.1 Comparison of *HPE* against individual predictors and their homogeneous ensembles

Table 4.2 presents the prediction results of individual predictors, their homogeneous ensembles (*Hom*), and our final model *HPE* in four data sets using three metrics. It should be noted that *HPE* is the ensemble of all six predictors. Comparing columns *Ind* and *Hom* in each data set, in each metric, we see that each *Hom* obtains better performance compared to the corresponding individual predictor (*Ind*). These are expected results and we include

to reaffirm the strength of homogeneous ensembles. Our main results come from the ensembling heterogeneous predictors or *HPE*. Comparing columns *Hom* and *HPE*, we see that the *HPE* outperforms the homogeneous ensembles in all metrics in all data sets. The difference is in the range of 0.018–0.084 with an average of 0.03825 on a scale of 1.00. This clearly demonstrates the strength of the *HPE* over the homogeneous ensembles.

As can be seen from Table 4.2, in all four data sets, and in all three metrics, the proposed *HPE* model outperforms all six predictors. These results confirm that using a heterogeneous predictors ensembling (*HPE*) model using 6 different predictors is better than using just a single predictor. The results show that each data representation or neural network type has its own strengths and weaknesses, thus employing a model ensembling multiple predictors could go beyond the individual performance of each data representation or each neural network type. For further clarification, the results are discussed below for each data set in detail.

- **IGC_{50}:** the proposed *HPE* obtained a correlation coefficient (R^2) of 0.831, RMSE of 0.426 $log(mol/L)$, and MAE of 0.182 $log(mol/L)$. However, among those 6 various individual predictors, MGC obtained the best R^2 value with of 0.782. FCPC obtained the best RMSE and MAE values with of 0.472 $log(mol/L)$ and 0.223 $log(mol/L)$, respectively. *HPE* improves the R^2 by 6.26% and 4.52%, RMSE by 9.74% and 7.59% , MAE by 9.03% and 8.73% from the best *Ind* and best *Hom* respectively.

- **LD_{50}:** the proposed *HPE* model obtains better results in all three metrics with R^2 of 0.680, RMSE of 0.536 $log(mol/kg)$, and MAE of 0.407 $log(mol/kg)$. *HPE* improves the R^2 by 7.59% and 4.61%, RMSE by 10.96% and 4.79% , MAE by 8.94% and 4.23% from the best *Ind* and best *Hom* respectively.

- **LC_{50}-DM :** as table shows, for this data set, the proposed *HPE* model obtains better results in all three metrics as well. It obtains R^2 of 0.811, RMSE of 0.787 $log(mol/L)$, and MAE of 0.620 $log(mol/L)$. *HPE* improves the R^2 by 8.13% and 6.29%, RMSE by 3.14% and 2.95% , MAE by 8.01% and 5.05% from the best *Ind* and best *Hom* respectively.

- **LC_{50}:** for this data set, the proposed *HPE* obtained R^2 of 0.742, RMSE of 0.788 $log(mol/L)$, and MAE of 0.621 $log(mol/L)$. *HPE* improves the R^2 by 7.53% and 4.50%, RMSE by 8.26% and 6.52% , MAE by 15.85% and 11.91% from the best *Ind* and best *Hom* respectively.

TABLE 4.2: Comparison of prediction results (10 fold cross-validation) of individual predictors, their homogeneous ensembles, and our proposed *HPE* model on four data sets. In the table, columns *Ind*, *Hom*, and *HPE* are respectively for individual predictors, their homogeneous ensembles, and the heterogeneous ensemble. For each metric, the bold numbers are the best ones in the respective columns, and the underlined number is the best among all.

Metric	Predictor	IGC_{50} Data set			LD_{50} Data set			LC_{50}-DM Data set			LC_{50} Data set		
		Ind	*Hom*	*HPE*	*Ind*	*Hom*	*HPE*	*Ind*	*Hom*	*HPE*	*Ind*	*Hom*	*HPE*
R^2	FCPC	0.781	0.785		0.564	0.572		0.740	0.751		0.671	0.685	
	FCPCe	0.683	0.698		0.563	0.581		0.642	0.658		0.675	0.689	
	C1DS	0.699	0.715		0.538	0.539		0.702	0.713		0.646	0.653	
	C2DF	0.632	0.645		0.557	0.564		0.665	0.671		0.601	0.615	
	MGC	**0.782**	**0.795**		**0.632**	**0.650**		0.669	0.675		**0.690**	**0.710**	
	MWC	0.771	0.785		0.586	0.623		**0.750**	**0.763**		0.687	0.693	
				0.831			0.680			0.811			0.742
RMSE	FCPC	**0.472**	0.471		0.621	0.610		0.864	0.850		0.874	0.860	
	FCPCe	0.564	0.550		0.617	0.604		1.085	1.055		0.872	0.859	
	C1DS	0.544	0.542		0.659	0.643		1.036	1.026		0.926	0.910	
	C2DF	0.605	0.602		0.623	0.616		0.985	0.961		0.967	0.962	
	MGC	0.480	0.476		**0.602**	**0.563**		0.969	0.962		0.986	0.967	
	MWC	0.478	**0.461**		0.625	0.589		**0.820**	**0.811**		**0.859**	**0.843**	
				0.426			0.536			0.787			0.788
MAE	FCPC	0.315	0.311		0.461	0.458		0.747	0.736		0.764	0.743	
	FCPCe	0.318	0.310		0.473	0.443		1.177	1.160		0.761	0.740	
	C1DS	0.353	0.334		0.514	0.497		1.074	1.070		0.857	0.834	
	C2DF	0.366	0.351		0.467	0.462		0.972	0.960		0.935	0.920	
	MGC	**0.310**	**0.309**		**0.447**	**0.425**		0.939	0.913		0.972	0.961	
	MWC	0.313	0.310		0.469	0.442		**0.674**	**0.653**		**0.738**	**0.705**	
				0.282			0.407			0.620			0.621

4.3.2 Evaluation of *HPE* model against several best-performing models

After finding the effectiveness of the proposed *HPE* model over various individual predictors and *Hom*, here we are to examine its performance against the state-of-the-art algorithms in the literature; the models used in the development of TEST software [96], TopTox [145] and Hybrid2d [68]. The results are shown in Table 4.3. As can be seen, from total 12 metrics, the proposed *HPE* model obtain the best results in 8 of them, especially in two of the data sets, it dominates other algorithms with obtaining better results in all three metrics. The detailed results are discussed below.

- **IGC_{50}:** As can be seen, for this data set, TEST consensus obtained the highest R^2 among different models in TEST software with of 0.764, while TopTox model achieved R^2 of 0.802. However, the proposed model obtained R^2 of 0.831 which is better than all 6 models compared including TopTox. The proposed model also obtained better RMSE and MAE values with of 0.426 $log(mol/L)$ and 0.282 $log(mol/L)$, respectively.

- **LD_{50}:** for this data set, the proposed model dominates other algorithms in all three metrics with R^2 of 0.680, RMSE of 0.536 $log(mol/kg)$, and MAE of 0.407 $log(mol/kg)$. The results of TopTox model, in all three metrics, was better than TEST software models but worse than the proposed model in this study.

- **LC_{50}-DM:** For R^2 and MRSE, the proposed model obtained 0.811 and 0.787 $log(mol/L)$ which was the better than all other models compared. However, for MAE, the proposed model obtained 0.620 $log(mol/L)$ which was better than all other models but TopTox with of 0.592 $log(mol/L)$.

- **LC_{50}:** As this table indicates, the proposed model obtained better R^2 results than 6 comparing models yet TopTox [145] with R^2 of 0.788. The TopTox [145] also obtained better results in terms of RMSE and MAE with of 0.677 $log(mol/L)$ and 0.446 $log(mol/L)$ respectively.

TABLE 4.3: Comparison of prediction results For *HPE* model vs. the sate-of-the-art models on four data sets

Model_Name	R^2	RMSE	MAE	R^2	RMSE	MAE
		IGC_{50}			LD_{50}	
HPE	**0.831**	**0.426**	**0.282**	**0.680**	**0.536**	**0.407**
hierarchical[96]	0.719	0.539	0.358	0.578	0.650	0.460
FDA[96]	0.747	0.489	0.337	0.557	0.657	0.474
group contribution[96]	0.682	0.575	0.411	–	–	–
nearest neighbor[96]	0.6	0.638	0.451	0.557	0.656	0.477
TEST consensus[96]	0.764	0.475	0.332	0.626	0.594	0.431
TopTox[145]	0.802	0.438	0.305	0.653	0.568	0.421
Hybrid2D[68]	0.810	–	–	0.629	–	–
		LC_{50}-DM			LC_{50}	
HPE	**0.811**	**0.787**	0.62	0.742	0.788	0.621
hierarchical [96]	0.695	0.979	0.757	0.71	0.801	0.574
single model [96]	0.697	0.993	0.772	0.704	0.803	0.605
FDA [96]	0.565	1.19	0.909	0.626	0.915	0.656
group contribution [96]	0.671	0.803	0.62	0.686	0.81	0.578
nearest neighbor [96]	0.733	0.975	0.745	0.667	0.876	0.649
TEST consensus [96]	0.739	0.911	0.727	0.728	0.768	0.545
TopTox [145]	0.788	0.805	**0.592**	**0.788**	**0.677**	**0.446**
Hybrid2D [68]	0.616	–	–	0.678	–	–

4.4 Discussion

Representing molecules in a single type of representation and then using homogeneous modeling techniques might not help to capture the whole information about that molecule. For instance, basic molecular graph representation does not capture the quantum mechanical structure of molecules or necessarily express the information. Similarly, the models which use molecular graphs as input like graph convolution will not be able to distinguish between chiral molecules (molecules having the same graph structure with

a mirror image to each other). In the case of fingerprints as an input, it is also possible that different molecules may have identical fingerprints which will make it difficult for a model to distinguish if it only takes fingerprints as input. There is also some information loss when one type of feature is converted into another type of feature.

In our experiments on the quantitative toxicity data sets, *HPE* obtains the highest performance followed by *Hom* and then individual predictors. The percentage improvement of *HPE* over *Hom* and *Ind* in all four data sets indicates that various predictors might be learning different knowledge from the same data set. As it can be seen in Table 4.2, graph based predictors like MGC and MWC achieves better performances in most of the metrics and data sets. Specifically in maximizing R^2 for IGC_{50}, LD_{50}, and LC_{50} data sets, MGC produces the best results whereas for LC_{50}-DM data set, MWC produces the best results. The quantitative toxicity data sets considered in this study contain relatively smaller molecules which make them more suitable for graph based predictors. The second highest performers on the average are FCPCe and FCPC which use the features based on physicochemicalical properties. These features have proved to have high predictive power in literature [68, 74, 155]. It can be noticed that predictors like C1DS and C2DF struggle to perform as compared to other predictors. Yet, when all of them are ensembled to form an *HPE* model, they help in improving the results.

Even though various heterogeneous predictors ensembling enhance the overall accuracy, yet it would be interesting to see the commonality between the learnt representation of various individual predictors and to what degree one predictor's captured knowledge differ to the others.

4.5 Conclusion

Toxicity prediction methods of chemical compounds recently achieved enhanced performance in terms of accuracy after the introduction of various deep learning models in this space. Usually, molecules are represented in a fixed representation which is then used as features with a specific machine learning method to predict the toxicity. Among various other types of compounds toxicity, quantitative toxicity measurement has paramount importance in pharmaceuticals [145]. The performance of any quantitative toxicity prediction method depends upon the specific features and model used. This restricts the overall performance to a single type of features and a model.

Our approach eliminates the restriction of model and data representation bound performance. Each of our model's predictor vary either on features level, deep learning architecture level or both. These predictors include FCPC, FCPCe, C1DS, C2DF, MGC and MWC. The FCPC and FCPCe vary only on feature level. They both use numerical features (different in number only) but share the same architecture. The C1DS and C2DF vary on both architecture and feature level. The C1DS uses SMILES directly as input while C2DF first converts SMILES into fingerprints. Molecular graph convolution (MGC) and molecular weave convolution (MWC) also vary on both architecture and feature level. Our motivation is to make a single model that utilizes different types of feature and architecture to obtain collective performance that could go beyond the individual performance of a single predictor type. We also performed experiments which showed that the heterogeneous ensembling method performs better than ensembling the homogeneous predictors. We achieved better performance in 8 out of 12 accuracy metrics for four quantitative toxicity data sets compared to the best-existing methods in the literature.

In this chapter, we proposed a method which uses various heterogeneous

predictors ensembling (*HPE*) to achieve better accuracy in quantitative toxicity prediction of four benchmark data sets. The software code along with data for this chapter can be found on https://github.com/Abdulk084/HPE. In the next chapter, we explore the idea of using meta features to train and test a meta ensemble neural neural in multitask way for quantitative toxicity tasks.

Chapter 5

Meta ensemble of multi-model deep learning

This chapter is published in the following peer review venues.

- Karim, A., Singh, J., Mishra, A., Dehzangi, A., Newton, M. H., Sattar, A. (2019, August). Toxicity Prediction by Multimodal Deep Learning. In Pacific Rim Knowledge Acquisition Workshop (pp. 142-152). Springer, Cham.

- Karim, A., Riahi, V., Mishra, A., Newton, M. H., Dehzangi, A., Balle, T., Sattar, A. (2021). Quantitative toxicity prediction via meta ensembling of multi task deep learning models. In ACS Omega 2021, 6, 18, 12306–12317

In this chapter, we propose a deep learning framework for toxicity prediction called QuantitativeTox which uses five individual base deep learning models and their own base feature representations. We then propose to adopt a meta ensembling approach using another separate deep learning model to perform aggregation of the outputs of the individual base deep learning models. We train our deep learning models in a weighted multi-task fashion combining four quantitative toxicity data sets of LD_{50}, IGC_{50}, LC_{50}, and LC_{50}-DM and minimising the root mean square errors. Compared

to the current state-of-the-art toxicity prediction method TopTox, on LD_{50}, IGC_{50}, and LC_{50}-DM, i.e. three out of four data sets, our method respectively obtains (5.457%, 16.666%, ,6.335%) better root means square errors, (6.413%, 11.803%, 12.162%) better mean absolute errors, and (5.206%, 7.356%, 2.538%) better coefficients of determination.

5.1 Introduction

QSAR modelling using deep learning techniques has become very popular in recent years [74]. Many of these methods use 2D features calculated from the one dimensional representation of the molecules called SMILES. SMILES is a language used in describing the chemical structure of a molecule as a string of characters [143]. There is a special grammar for SMILES to represent atoms, types, and chemical bonds among them. SMILES strings are used in calculating various types of numerical features (e.g. physicochemical descriptors) and molecular graphs by using different featurization methods [113, 153]. These numerical features can then be used by traditional machine learning approaches such as K-Nearest Neighbours (KNN), Support Vector Machines (SVM), Random Forest (RF), and Fully Connected Neural Networks (FCNN) to predict activity or properties of a chemical compound [89]. Besides, numerical features, SMILES strings can also be used to generate molecular graphs or images, which then can be used in various types of convolutional neural network (CNN) to predict molecular activities [50]. Using CNN for molecular graphs or images needs relatively less domain expertise. It should be noted that SMILES strings can also be transformed into a vector representation or into their respective fingerprints. Fingerprints are bit strings composed of 0's and 1's and can be used in Recurrent Neural Networks (RNN) for molecular activity/property prediction [49]. Recently in

the area of toxicity prediction, a specialized type of features called element-specific topological descriptors (ESTDs) are used in deep neural networks and in consensus models by TopTox to predict toxicity level [145]. A recent software named AdmetSAR uses molecular fingerprints to predict toxicity using RF, SVM, and KNN models [151]. Another method, named Hybrid2D, uses joint optimization of shallow neural networks and decision trees on 2D features only to predict toxicity measurement levels [68]. The performance of all these quantitative prediction methods is restricted by the specific type of features or models used in prediction. A yet another method named DeepHIT [120] utilize a reasonably diverse feature set, but it still suffers from the lack of an effective way for combining the outputs of individual models to obtain a robust performance over a range of metrics. Moreover, DeepHit is optimized for toxicity classification to enhance the sensitivity of the model.

In this chapter, for quantitative toxicity prediction, we hypothesize that an effective aggregation of various chemical information captured within various feature representations extracted from SMILES strings can improve the prediction accuracy. For this purpose, we propose a three stage deep learning framework: A featurization stage first generates a number of base features. A base learning stage then trains a number of deep learning models, one for each base feature. A meta learning stage uses the outputs of the base learning stage as input meta features and trains a separate deep learning model for meta ensembling and producing the final output. The five types of base features generated in the featurization stage are 2D and 3D descriptors, molecular graph features, extended-connectivity fingerprints (ECFPs), SMILES vocabulary based embedded vectors, and fingerprint based embedded vectors. The base learning stage comprises five deep learning models such as 2 $\times$ deep neural networks, 1 $\times$ graph convolutional neural network, and 2 $\times$ 1D convolutional neural network. Each of these deep learning models essentially is for one of the base features. The meta learning stage comprises

a fully connected deep neural network. We train all of our deep learning models in an weighted multi-task fashion combining four quantitative toxicity data sets of LD_{50}, LC_{50}, IGC_{50} and LC_{50}-DM and minimising the root mean square error. Compared to the current state-of-the-art toxicity prediction method TopTox, on LD_{50}, IGC_{50}, and LC_{50}-DM, i.e. three out of four data sets, our method respectively obtains (5.457%, 16.666%, ,6.335%) better root means square error, (6.413%, 11.803%, 12.162%) better mean absolute error, and (5.206%, 7.356%, 2.538%) better coefficient of determination.

5.2 Materials and methods

We describe the four quantitative data sets, the evaluation criteria, and the weighted loss function used in this work. As shown in Figure 5.1, our deep learning framework has three stages: featurization, base learning, and meta learning. The featurization stage is to generate base features which are used in training base learning models. The output of the base learning models are then used as meta features for the meta learning model to produce the final predictions. We describe the three stages in more details.

5.2.1 Featurization stage

The featurization stage of our framework consists of various types of featurizers. Each featurizer takes SMILES strings as input and produces fixed length base features as output. Figure 5.1 shows the five featurizers and their output base features.

2D and 3D descriptors (DESC)

A total of 995 high level features such as 2D and 3D physicochemical descriptors are computed using Mordred [103]. The feature names are in Table S2 of

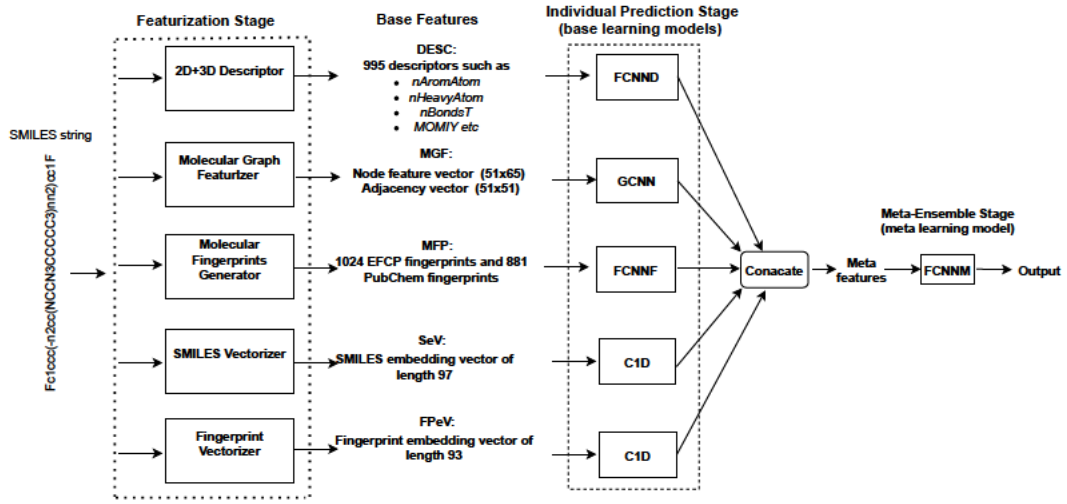

FIGURE 5.1: End to end flow diagram of all the stages of the proposed framework.

DeepHIT supplementary material [120]. These features are numerical in nature and describe the physical and chemical properties of molecules [22]. 2D descriptors represent information related to size, shape, distribution of electrons, ocatnol-water distribution coefficient (LogP) measuring lipophilicity, nAromAtom denoting the number of aromatic atoms, nHeavyAtom denoting the number of heavy atoms, and nBondsT denoting the number of triple bonds. 3D descriptors relate to the 3D conformation of the molecules and include the moment of inertia along Y axis (MOMIY) [22]. The value of each descriptor is normalized between 0 and 1.

Molecular graph features (MGF)

Topological information of molecules can be intuitively and concisely expressed via molecular graph features. In this featurizer [120, 121], molecular graph features such as node vectors and adjacency matrix are computed. Node vectors represent atoms in the SMILE strings. Adjacency matrix represent the bonds between atoms. In this study, we extract $[51 \times 65]$ node vectors and $[51 \times 51]$ adjacency matrix. Here 51 is the maximum number of atoms and 65 is the length of the one hot-encoded feature vector computed

from atom descriptors. The details of these are in Table S3 of DeepHIT supplementary material [120].

Molecular fingerprints (MFP)

The third featurizer deals with fingerprints, where structural features are represented either by bits in a bit string or by counts in a count vector [116, 136]. 1024 extended-connectivity fingerprints with a maximum diameter parameter of 2 (EFCP2) fingerprints and 881 pubChem fingerprints are computed using the Python package PyBioMed [36, 120]. EFCPs are also referred to as circular fingerprints and are specifically designed for structure-activity relationship modeling [117] whereas pubChem fingerprints are mainly designed for similarity neighboring and similarity searching [53].

SMILES strings embedded vectors (SeV)

We compute low level features SMILES strings embedded vectors [45, 70]. These features do not directly describe any biological attribute of the molecules, but have been proven to have a reasonable predictive power in various QSAR tasks. In the SMILES vectorizer, we create a vocabulary based on the valid SMILES tokens. Based on the training data, SMILES vocabulary is generated using tokenizer module developed by Reverie Labs, the link of which is given below.

`https://blog.reverielabs.com/transformers-for-drug-discovery/`.

Each SMILES string is converted into fixed sized numerical vectors based on dictionary mapping of SMILES vocab as shown Figure 5.2. The dictionary maps each SMILES vocab element to a numerical value. The length of the longest SMILES string is 97 in terms of SMILES vocab element in the training data considered for this work. A total of 64 unique tokens are determined based on the training data. Each SMILES string is converted into a one-hot encoded vector based on the SMILES vocabulary.

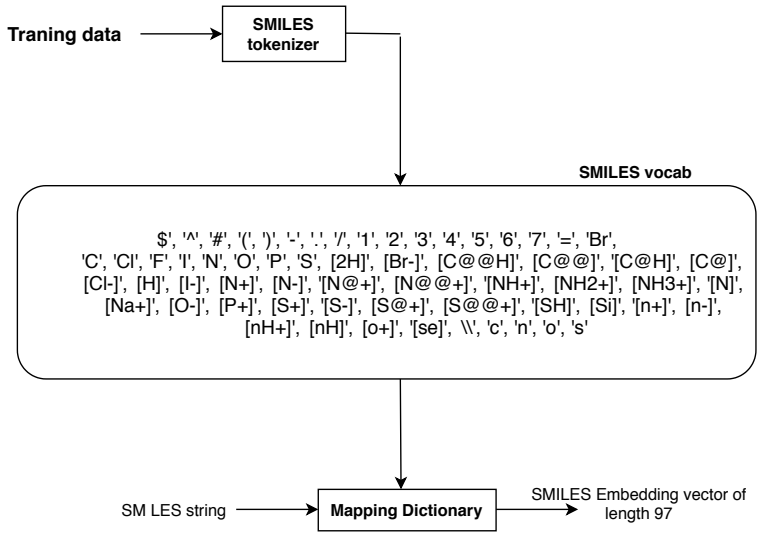

FIGURE 5.2: SMILES embedding vectors based on the vocab elements

Fingerprints embedded vectors (FPeV)

We also compute fingerprint based embedded vectors [66]. In the fingerprint vectorizer, SMILES string are converted into 1024-bit Morgan (or circular) fingerprints with a radius of 2 via RDKIT [84]. As per an existing technique [66], we extract fingerprint indices, which are marked 1 in the fingerprints generated. Thus, we obtain a vector of length 93 where the vector consists of integers representing presence of specific substructures in a molecule. The procedure for fingerprint embedded vector is described FP2VEC [66].

5.2.2 Base learning stage

The base learning stage consists of five base deep neural network models, which are trained on respective base features from the featurization stage. All of the base models are trained at a learning rate of $10e^{-4}$ with Adam optimizer and 100 epochs with a batch size of 32. Selection of parameters, hyper-parameters and network architecture of base models are inspired from previous published research in this area [45, 66, 68, 69, 70, 85, 120]. Each of these base models produce 4 regression values for 4 tasks as output; only the

output corresponding to given task is counted for a given input. The base models are shown in Figure 5.3a,b,c,d and are described below. The Keras deep learning framework and Spektral package are used in developing the base models [25, 52].

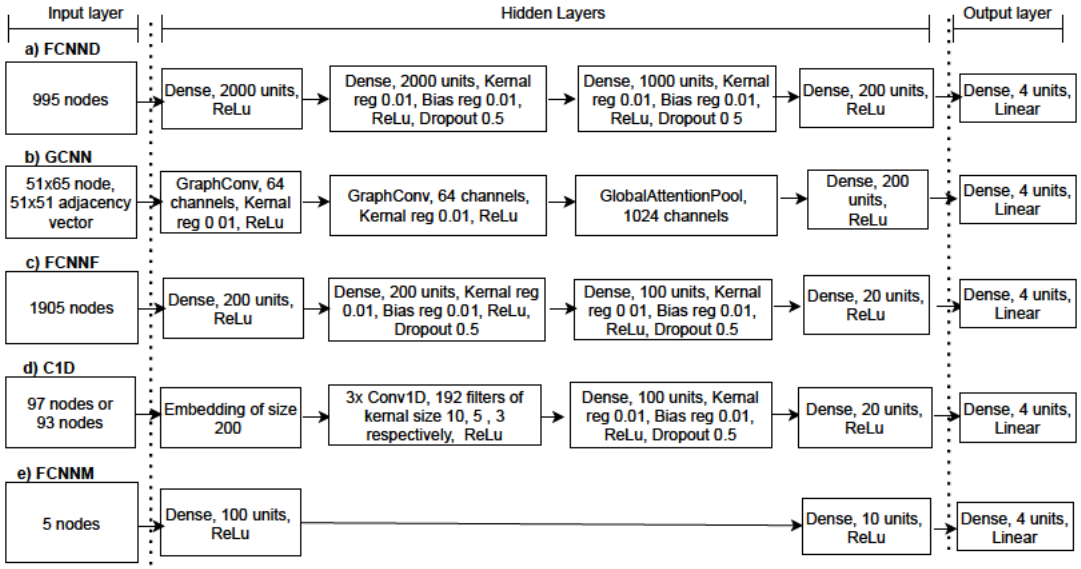

FIGURE 5.3: a) A fully connected neural network for 995 2D+3D descriptors as base features. b) A convolutional neural network for node vectors of size 51×65 and adjacency vectors of size 51×51 as base features. c) A fully connected neural network for 1024 EFCP and 881 pubChem fingerprints as base features. d) A 1D convolution neural network for each of both SMILES embedded vectors and fingerprint embedded vectors as base features. e) A meta ensemble fully connected neural network for meta features.

Fully connected neural network for DESC (FCNND)

As shown in Figure 5.3a), a fully connected deep neural network with 4 hidden layers was trained and validated on 995 2D+3D physicochemical descriptors. The input layer consists of 995 nodes for the 995 physicochemical descriptors. The output layer has 4 units, one for each of the 4 tasks. All the layers in FCNND are densely connected and receive input from all the units in the previous layer. The number of units in each hidden layer decreases gradually and a ReLu activation [51, 65] is applied at the end of each

layer except the output layer. Various regularization parameters such as Kernel regularizer that applies penalties to the Kernel (main units in layer), bias regularizer that applies penalties to the bias units to reduce the over-fitting during optimization [51, 54] are used. We also apply a drop-out rate of 0.5 to the middle layers [130].

Graph convolutional neural network for MGF (GCNN)

As shown in Figure 5.3b), a graph convolutional neural network (GCNN) was trained using the molecular graph features. GCNN consists of two graph convolution layers [80], one global attention pool layer [87], and a dense layer before the output. Each of the graph convolutional layers are initiated with 64 channels with a Kernel regularization value of 0.01 and a ReLu activation. The number of channels in the global attention pool layer is made equal to 1024, the number of units in the following dense layer.

Fully connected neural network for MFP (FCNNF)

As shown in Figure 5.3c), a FCNN is used with fingerprints as the base feature. Unlike FCNND, FCNNF uses a much smaller number of units in each layer. Except the number of units, other parameters are kept the same as in FCNND. The number of input nodes in the input layer is kept at 1905 to match the sum of 1024 EFCP fingerprints and 881 pubChem fingerprints.

Convolution 1D Neural Network for SeV (C1DS) and for FPeV (C1DF)

As shown in Figure 5.3d), a variant of a Convolution 1D Neural Network (C1D) is used for each of SMILES embedded vectors and fingerprint embedded vectors as base features. The only difference between the two C1D is in the number of input-layer nodes: 97 for SMILES embedded vectors and 93 for fingerprint embedded vectors. Input vectors are converted to a trainable

embedded matrix of size [97 or 93 $\times$ 200], which was then fed into a series of three 1D convolution layers. Each of these 1D convolution layers used ReLu activation, 192 filters with a Kernel size of 10, 5 and 3 respectively. Two densely connected layers are also used before the output layer.

5.2.3 Meta learning stage

As mentioned before, each base model produce four outputs for four data sets. The outputs of the base models are used as meta features for the meta learning model. As shown as FCNNM in Figure 5.3e), the meta learning model is an FCNN with an input layer, an output layer, and two hidden layers. It is trained at a learning rate of $10e^{-3}$ with an Adam optimizer and 300 epochs with a batch size of 32. The meta learning model acts as an ensembling method for the whole framework. Our hypothesis is that for quantitative toxicity prediction, the meta ensembling will be able better aggregate the output of individual base models than the typical average ensembling approaches.

5.2.4 Data sets

We use end points for four quantitative toxicity data sets (also called tasks). These data sets are LD_{50}, IGC_{50}, LC_{50}, and LC_{50}-DM [145]. These endpoint measures have been used in toxicology for estimating the toxicity behaviour of a given chemical compound on a given population of a given organism. These measures depend on the concentration of the compound as well as the duration of exposure of a given organism to the compound. LC_{50} and LD_{50} are the compound's concentrations that kill half members of the tested animal population. LC_{50} records the toxicity of a given compound on fathead minnow, a species of temperate freshwater fish after 96 hour exposure. LC_{50}-DM records the concentration of a compound in water in milligrams

per litre causing 50% population of *Daphnia maga* to die after 48 hour. LD_{50} data set has the lethal dose data for killing 50% rat population when a given compound is administered orally, given oral administration could cause less toxicity than intravenous route. IGC_{50} data set shows the concentration of a chemical compound to arrest the growth of *Tetrahymena pyriformis* when exposed for 40 hour. The units of LC_{50}, LC_{50}-DM, IGC_{50} end points are $-\log_{10}(T\ \ mol/L)$, where T represents corresponding end point. For LD_{50} set, the units are $-\log_{10}(LD_{50}\ mol/kg)$.

TABLE 5.1: Description of data sets after standardization. LD_{50} data set actually has 5924 and 1479 train and test compounds before standardization.

data set	train	test	total
LD_{50}	5901	1475	7376
IGC_{50}	1434	358	1792
LC_{50}	659	164	823
LC_{50}-DM	283	70	353

Original data is available at http://cfpub.epa.gov/ecotox/, http://cfpub.epa.gov/ecotox/, http://chem.sis.nlm.nih.gov/chemidplus/chemidheavy.jsp, and http://chem.sis.nlm.nih.gov/chemidplus/chemidheavy.jsp. We obtained pre-processed train and test sets (as pairs of SMILES strings and toxicity measures) from TopTox [145]. As shown in Table 5.1, these data sets have different sizes ranging from hundreds to thousands compounds. In LD_{50}, some molecules were removed as part of standardization using using RDKIT http://www.RDKIT.org/ and MolVS https://molvs.readthedocs.io/en/latest/. The train set from each of the four tasks is randomly split into four types of subsets: 70% for base train set, 10% for base validation set, 10% for meta train set, and 10% for meta validation set. Next, for each type of subset, the corresponding subsets for the four tasks are concatenated to obtain a combined set of that type. The test sets from the four tasks are also concatenated to obtain a combined test set. These combined train and test sets are available from a GitHub repository https://github.com/Abdulk084/

QuantitativeTox/blob/master/training_multitask.tar.xz. The data split procedure is shown in Figure 5.4.

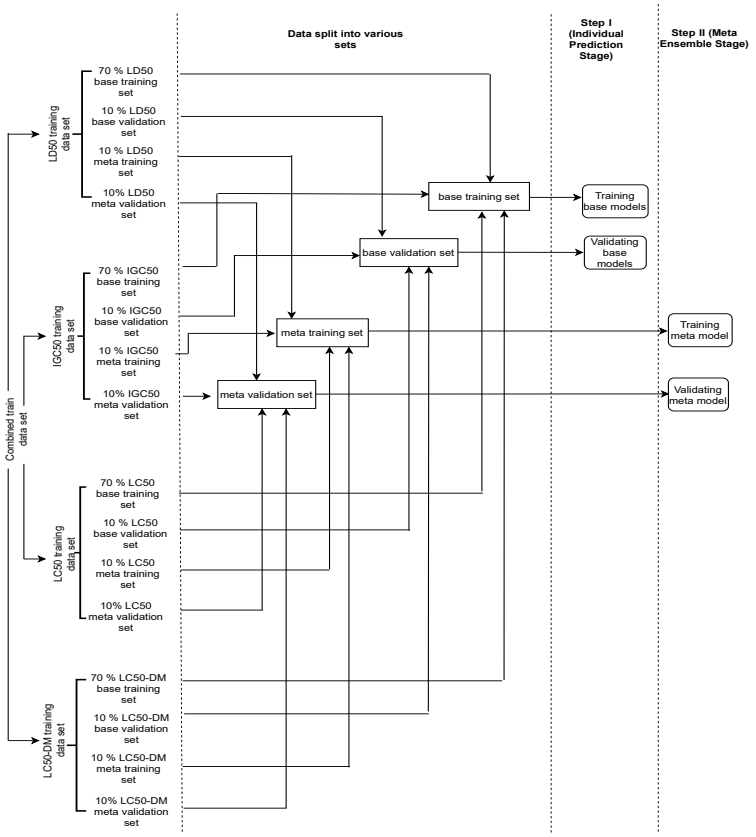

FIGURE 5.4: Multi-task data split of train and test data

5.2.5 Weighted loss functions

Although, as noted before, we use three measures to report our performances, we only use the third metric rmse, as the loss function in the training of the deep neural networks in both of base and meta learning stages. However, each learning model has four outputs for four data sets. So in each learning model (either base or meta), we compute the rmse value for each output separately and then we take a weighted sum of the rmse values to compute the total rmse value for all outputs of the learning models. The weight is w when a given input belongs to the task associated with the output and 1 when not. This means for a given input from a given task, the weight of the loss functions associated with the other three outputs are all 1. In the experiments, we

try various values from $\{1, 3, 5, 7, 9\}$ for weight w as shown in Equation (5.1), (5.2), (5.3) and (5.4). Once a w is selected, the same w is used in all based models and the meta model.

$$LD_{50}loss = \boldsymbol{w} \times LD_{50}rmse + IGC_{50}rmse + LC_{50}rmse + LC_{50-DM}rmse \tag{5.1}$$

$$IGC_{50}loss = LD_{50}rmse + \boldsymbol{w} \times IGC_{50}rmse + LC_{50}rmse + LC_{50-DM}rmse \tag{5.2}$$

$$LC_{50}loss = LD_{50}rmse + IGC_{50}rmse + \boldsymbol{w} \times LC_{50}rmse + LC_{50-DM}rmse \tag{5.3}$$

$$LC_{50-DM}loss = LD_{50}rmse + IGC_{50}rmse + LC_{50}rmse + \boldsymbol{w} \times LC_{50-DM}rmse \tag{5.4}$$

5.3 Results

We select weights to be used in our weighted loss functions. Then, we evaluate our base features and meta features. These experimental results are reported from 10-fold cross validation processes. Finally, we compare our experimental results with that of the state-of-the-art toxicity prediction methods using our independent test sets.

5.3.1 Weight selection in multi-task loss function

For these experiments, we use all components of our method: five types of base features, five base models, and the meta model. We consider the output of the the meta model. Figure 5.5 shows our method achieves the best performance with weight 5 for LD_{50}, IGC_{50} and LC_{50} and with weight 9 for LC_{50}-DM. Henceforth, we will use these weights in our further experiments.

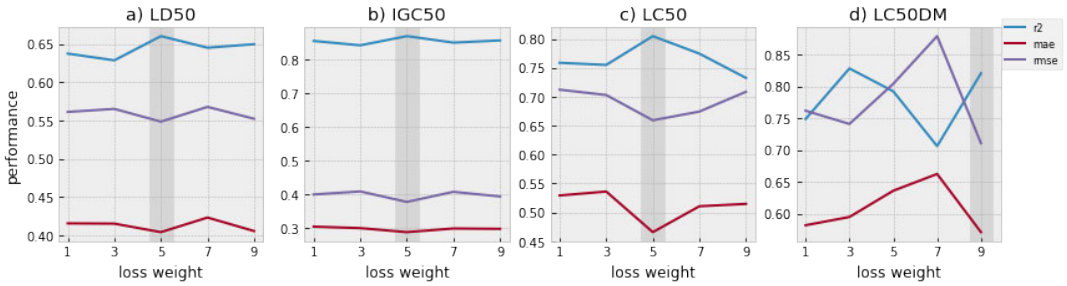

FIGURE 5.5: 10 fold cross validation performance with various loss weight values evaluated against meta validation sets.

5.3.2 Performance evaluation of base features

Table 5.2 shows the performance of the base models using respective base features and using the base validation set. The table shows the averages of the ten runs in the ten fold cross validation process. The standard deviation values are given in Table 5.3.

TABLE 5.2: 10 fold cross validation performance of the base models using respective base features and using the base validation set.

Base features	r^2	mae	rmse	r^2	mae	rmse
	LD_{50}			IGC_{50}		
DESC	0.553	0.465	0.627	0.788	0.355	0.482
MGF	0.544	0.469	0.630	0.764	0.358	0.502
MFP	0.566	0.457	0.621	0.737	0.417	0.549
FPeV	0.361	0.563	0.754	0.432	0.628	0.800
SeV	0.267	0.597	0.800	0.572	0.532	0.703
	LC_{50}			LC_{50}-DM		
DESC	0.700	0.574	0.814	0.623	0.792	1.053
MGF	0.568	0.674	0.935	0.544	0.875	1.158
MFP	0.617	0.630	0.892	0.456	0.738	1.008
FPeV	0.369	0.909	1.157	0.297	0.937	1.270
SeV	0.567	0.699	0.962	0.431	0.955	1.289

As shown in Table 5.2, DESC performed better in all three metrics for LC_{50} and IGC_{50}. For LD_{50} and LC_{50}-DM, MFP obtains the best results in all metrics except in one case. MGF shows reasonable performance close to DESC and MFP. The possible reason behind these performance might be the

direct biological relevance of DESC, MFP, and MGF to activity prediction. SeV and FPeV showed the lowest performance for all tasks in three metrics possibly because of their no direct biological relevance to the activity prediction. From Table 5.3, we observe that MGF in LD_{50} and LC_{50} and DESC in IGC_{50} and LC_{50}-DM obtain the overall most stable performances with least deviations values.. The standard deviation values are large for FPeV and SeV compared to that for DESC, MGF and MFP. Comparatively smaller data sets such as LC_{50} and LC_{50}-DM show the least stable results in terms of standard deviation values.

TABLE 5.3: Standard deviation values for 10 fold cross validation performance of the base models on base validation set.

Base features	r^2	mae	rmse	r^2	mae	rmse
		LD_{50}			IGC_{50}	
DESC	0.053	0.018	0.029	0.030	0.030	0.042
MGF	0.034	0.015	0.026	0.058	0.034	0.060
MFP	0.036	0.015	0.028	0.045	0.036	0.040
FPeV	0.040	0.023	0.033	0.093	0.067	0.083
SeV	0.154	0.071	0.095	0.062	0.056	0.067
		LC_{50}			LC_{50}-DM	
DESC	0.039	0.071	0.094	0.095	0.103	0.141
MGF	0.113	0.046	0.069	0.130	0.179	0.190
MFP	0.140	0.089	0.135	0.429	0.114	0.252
FPeV	0.200	0.108	0.132	0.193	0.139	0.197
SeV	0.143	0.087	0.159	0.161	0.207	0.298

5.3.3 Performance evaluation of meta features

Our goal in this study is to effectively aggregate the chemical information extracted from various base features for quantitative toxicity data sets so that the regression performance can be improved. We have five types of base features DESC, MGF, MFP, SeV, and FPeV. Hence, we have five based models. We consider all possible subsets of these five types of base features. This gives us $2^5 - 1 = 31$ possible subsets. For each subset, we consider only the

base features in the subset and then use only the corresponding base models. However, for each of the 5 subsets having just one type of base features, in order to have ensembling effects, we use two base models using the same base features but trained separately. The set of four outputs of a subset of features are denoted by M*i*-*j* where *i* is the number of types of features in the subset and *j* is a unique index within the subsets all having *i* types of features. These outputs are used as the meta features for the meta learning models. To summarise, we consider 31 possible meta ensembling models. For convenience, M*i*-*j* is also used to denote the corresponding meta ensembling model. Moreover, M*i* denotes the set of meta models all have *i* types of features. Table 5.4 shows 10 fold cross validation performance of the 31 meta ensembling models. The corresponding standard deviation values are given Table 5.5. We also compute the mean of standard deviation values for M1 meta features from Table 5.5 and for base models from Table 5.3 for all tasks to see the effect of meta ensembling on the stability of results. These mean values are plotted and compared in Figure 5.6.

Table 5.4 shows that meta features in M3, M4 and M5 show overall better performance for most of the metrics for all four tasks. M3-1 which represents only those three features which are directly related to a biological activity prediction achieves better performance in 4 cases across three tasks. Similarly, M3-10, M4-1, M4-2 and M5-1 achieve better performance in two metrics across one or two tasks. Note M3-1, M4-1, M4-2 and M5-1 are associated with at least two direct biologically relevant base features. Surprisingly, SeV and FPeV which lack in direct biological relevance helps LC_{50}-DM task to improve r^2, mae and rmse from 0.707, 0.636 and 0.841 to 0.773, 0.551 and 0.702 respectively when used with DESC. Using SeV and FPeV individually or together result in worst performance in all metrics. Overall, meta features used with meta ensemble models results in more stable performances as shown in Table 5.5. M2-8 shows the most stable performances across two tasks in 5

TABLE 5.4: 10 fold cross validation results for meta features on meta validation set.

Meta features	Base features	LD_{50}			IGC_{50}			LC_{50}			LC_{50}-DM		
		r^2	mae	rmse	r^2	mae	rmse	r^2	mae	rmse	r^2	mae	rmse
M1-1	DESC, DESC	0.585	0.455	0.613	0.828	0.329	0.434	0.767	0.547	0.737	0.707	0.636	0.841
M1-2	MGF, MGF	0.550	0.470	0.628	0.803	0.345	0.465	0.662	0.650	0.860	0.736	0.670	0.848
M1-3	MFP, MFP	0.578	0.459	0.620	0.764	0.390	0.518	0.673	0.604	0.831	0.704	0.672	0.847
M1-4	FPeV, FPeV	0.365	0.554	0.742	0.431	0.612	0.781	0.491	0.834	1.082	0.507	0.832	1.106
M1-5	SeV, SeV	0.287	0.589	0.791	0.582	0.519	0.679	0.648	0.641	0.883	0.697	0.697	0.904
M2-1	MGF, MFP	0.637	0.419	0.571	0.846	0.303	0.411	0.736	0.542	0.726	0.714	0.624	0.813
M2-2	MGF, DESC	0.629	0.433	0.580	0.851	0.291	0.397	0.772	0.512	0.710	0.738	0.649	0.871
M2-3	MGF, SeV	0.577	0.458	0.617	0.820	0.331	0.444	0.736	0.573	0.760	0.710	0.600	0.804
M2-4	MGF, FPeV	0.564	0.468	0.629	0.837	0.318	0.426	0.712	0.578	0.784	0.719	0.674	0.861
M2-5	MFP, DESC	0.633	0.418	0.566	0.856	0.313	0.409	0.771	0.509	0.678	0.722	0.723	0.910
M2-6	MFP, SeV	0.593	0.455	0.614	0.779	0.376	0.496	0.702	0.584	0.770	0.745	0.613	0.786
M2-7	MFP, FPeV	0.597	0.447	0.593	0.773	0.381	0.501	0.671	0.616	0.828	0.690	0.676	0.868
M2-8	DESC, SeV	0.586	0.451	0.603	0.817	0.331	0.440	0.715	0.570	0.768	0.779	0.589	0.762
M2-9	DESC, FPeV	0.602	0.449	0.597	0.818	0.323	0.435	0.724	0.568	0.756	0.692	0.688	0.878
M2-10	SeV, FPeV	0.414	0.536	0.720	0.641	0.500	0.656	0.623	0.653	0.864	0.645	0.721	0.933
M3-1	MGF, MFP, DESC	0.647	0.412	0.559	**0.866**	**0.289**	0.385	0.770	0.499	**0.676**	**0.812**	0.566	0.724
M3-2	MGF, MFP, SeV	0.627	0.431	0.576	0.846	0.316	0.415	0.760	0.505	0.690	0.708	0.643	0.821
M3-3	MGF, MFP, FPeV	0.625	0.426	0.580	0.842	0.316	0.421	0.730	0.569	0.784	0.713	0.618	0.807
M3-4	MGF, DESC, SeV	0.614	0.433	0.577	0.853	0.302	0.403	0.756	0.512	0.693	0.778	0.589	0.756
M3-5	MGF, DESC, FPeV	0.634	0.431	0.578	0.860	0.306	0.407	0.777	0.514	0.684	0.737	0.607	0.813
M3-6	MGF, SeV, FPeV	0.573	0.458	0.613	0.821	0.339	0.450	0.752	0.535	0.720	0.719	0.637	0.798
M3-7	MFP, DESC, SeV	0.626	0.425	0.580	0.847	0.306	0.410	0.754	0.513	0.702	0.757	0.613	0.786
M3-8	MFP, DESC, FPeV	0.634	0.423	0.575	0.845	0.315	0.426	0.744	0.530	0.707	0.780	0.622	0.807
M3-9	MFP, SeV, FPeV	0.589	0.448	0.606	0.799	0.367	0.482	0.704	0.547	0.768	0.712	0.689	0.894
M3-10	DESC, SeV, FPeV	0.589	0.440	0.588	0.833	0.323	0.433	0.697	0.542	0.755	0.773	**0.551**	**0.702**
M4-1	MGF, MFP, DESC, SeV	0.644	0.413	**0.551**	0.860	0.290	**0.384**	0.783	0.516	0.701	0.736	0.558	0.754
M4-2	MGF, MFP, DESC, FPeV	0.637	0.413	0.555	0.859	0.294	0.400	**0.786**	**0.489**	0.678	0.783	0.563	0.731
M4-3	MGF, MFP, SeV, FPeV	0.629	0.421	0.569	0.840	0.313	0.422	0.768	0.544	0.725	0.725	0.618	0.810
M4-4	MGF, DESC, SeV, FPeV	0.619	0.430	0.578	0.851	0.298	0.403	0.761	0.525	0.718	0.741	0.584	0.784
M4-5	MFP, DESC, SeV, FPV	0.632	0.421	0.569	0.850	0.311	0.411	0.756	0.518	0.691	0.756	0.605	0.811
M5-1	MGF, DESC, SeV, FPeV, MFP	**0.652**	**0.411**	0.553	0.860	0.302	0.399	0.785	0.506	0.686	0.784	0.615	0.784

performance metrics.

5.3.4 Effectiveness of meta models over base models

In order to investigate the effectiveness of meta models M2-M5 compared to the best meta models in M1 that have only one types of base features, we compute % improvement and show in Figure 5.7. An overall improvement can be observed in r^2, mae and rmse for all four tasks. As expected, meta model M2-10, which refers to using SeV and FPeV only, causes increase in both types of errors and decrease in the correlation substantially. For each task separately, we highlight in Figure 5.7a,b,c,d the best meta model. We select M5-1 with improvement of 11.44% for r^2, 9.75% for mae and 9.74% for rmse on LD_{50}; M3-1 with improvement of 4.53% for r^2, 12.26% for mae, and 11.39% for rmse on IGC_{50}; M4-2 with improvement of 2.50% for r^2, 10.58% for mae, and 8.00% for rmse on LC_{50}; and M3-1 with improvement of 10.39% for r^2, 11.04% for mae, and 13.92% for rmse on LC_{50}-DM. We develop our final

TABLE 5.5: Standard deviation values for 10 fold cross validation results for various meta features on meta validation set.

Meta features	Base features	LD_{50}			IGC_{50}			LC_{50}			LC_{50}-DM		
		r^2	mae	rmse	r^2	mae	rmse	r^2	mae	rmse	r^2	mae	rmse
M1-1	DESC, DESC	0.037	0.020	0.031	0.026	0.031	0.041	0.034	0.052	**0.062**	0.128	0.118	0.180
M1-2	MGF, MGF	0.030	0.015	0.020	0.052	0.043	0.058	0.087	0.072	0.087	0.087	0.132	0.156
M1-3	MFP, MFP	0.034	0.012	0.021	0.041	0.026	0.038	0.109	0.100	0.167	0.117	0.105	0.146
M1-4	FPeV, FPeV	0.053	0.018	0.033	0.102	0.060	0.071	0.120	0.100	0.131	0.156	0.152	0.211
M1-5	SeV, SeV	0.117	0.059	0.072	0.096	0.078	0.092	0.066	0.050	0.065	0.108	0.064	**0.096**
M2-1	MGF, MFP	0.025	0.011	**0.015**	0.037	0.020	0.029	0.101	0.084	0.113	0.133	0.110	0.152
M2-2	MGF, DESC	0.026	0.012	0.016	0.032	0.021	0.040	0.063	0.082	0.133	0.083	0.139	0.161
M2-3	MGF, SeV	0.020	0.013	0.020	0.031	0.032	0.047	0.049	0.067	0.083	0.097	**0.062**	0.133
M2-4	MGF, FPeV	0.048	0.015	0.026	0.026	0.020	0.034	0.080	0.061	0.079	0.098	0.081	0.097
M2-5	MFP, DESC	0.035	0.010	0.021	0.026	0.024	0.032	0.045	0.062	0.084	0.122	0.124	0.152
M2-6	MFP, SeV	0.036	0.023	0.034	0.038	0.033	0.043	0.097	0.093	0.143	0.072	0.092	0.122
M2-7	MFP, FPeV	0.032	0.016	0.025	0.024	0.023	0.030	0.067	0.080	0.120	0.090	0.099	0.112
M2-8	DESC, SeV	0.016	0.013	0.016	0.025	**0.014**	**0.023**	**0.023**	**0.042**	**0.062**	0.083	0.078	0.113
M2-9	DESC, FPeV	0.025	0.020	0.030	0.030	0.024	0.036	0.061	0.062	0.082	0.118	0.100	0.148
M2-10	SeV, FPeV	0.041	0.013	0.026	0.045	0.039	0.041	0.074	0.086	0.124	0.088	0.086	0.111
M3-1	MGF, MFP, DESC	0.018	0.013	0.021	0.026	0.026	0.043	0.066	0.063	0.088	0.072	0.129	0.185
M3-2	MGF, MFP, SeV	**0.019**	0.015	0.016	0.024	0.024	0.027	0.053	**0.042**	0.058	0.135	0.104	0.136
M3-3	MGF, MFP, FPeV	0.033	0.016	0.027	0.038	0.026	0.039	0.060	0.075	0.084	0.106	0.125	0.170
M3-4	MGF, DESC, SeV	0.022	0.012	0.016	0.016	0.027	0.031	0.046	0.068	0.109	**0.046**	0.123	0.124
M3-5	MGF, DESC, FPeV	0.030	0.019	0.033	0.025	0.020	0.030	0.062	0.072	0.118	0.105	0.127	0.222
M3-6	MGF, SeV, FPeV	0.032	0.022	0.029	0.028	0.027	0.038	0.037	0.045	0.064	0.089	0.092	0.131
M3-7	MFP, DESC, SeV	0.029	**0.008**	0.027	0.021	0.026	0.034	0.081	0.086	0.113	0.075	0.099	0.131
M3-8	MFP, DESC, FPeV	0.028	0.012	0.026	0.023	0.018	0.031	0.053	0.060	0.084	0.079	0.120	0.171
M3-9	MFP, SeV, FPeV	0.045	0.030	0.042	0.034	0.033	0.042	0.066	0.064	0.088	0.082	0.091	0.112
M3-10	DESC, SeV, FPeV	0.030	0.021	0.031	0.019	0.019	0.028	0.066	0.051	0.070	0.111	0.109	0.166
M4-1	MGF, MFP, DESC, SeV	0.027	0.019	0.030	0.039	0.035	0.053	0.032	0.057	0.070	0.109	0.113	0.145
M4-2	MGF, MFP, DESC, FPeV	0.026	0.017	0.021	**0.014**	0.016	0.026	0.049	0.054	0.066	0.086	0.102	0.153
M4-3	MGF, MFP, SeV, FPeV	0.027	0.017	0.029	0.025	0.028	0.040	0.045	0.061	0.080	0.187	0.124	0.161
M4-4	MGF, DESC, SeV, FPeV	0.030	0.016	0.035	0.035	0.030	0.044	0.068	0.071	0.096	0.118	0.100	0.186
M4-5	MFP, DESC, SeV, FPV	0.025	0.012	0.028	0.027	0.025	0.038	0.072	0.057	0.081	0.108	0.132	0.234
M5-1	MGF, DESC, SeV, FPeV, MFP	0.031	0.017	0.026	0.028	0.030	0.040	0.052	0.046	0.072	0.084	0.098	0.139

model with these selected meta models for each task individually as follows for external independent testing.

5.3.5 Comparative landscape using the external independent test sets

We compare our method's performance against the state-of-the-art methods in the literature e.g. the models used in the development of TEST software [96], TopTox [145] and Hybrid2d [68]. The results are shown in Table 5.6. As can be seen, from total 12 metrics, the proposed method obtains best results in 11 of them. Especially in three of the four data sets, it dominates other algorithms with significant margin in all three metrics. The detailed results are discussed below for each task separately.

- **LD_{50}:** For this data set, which is the largest for the four, our proposed method dominate other methods in all three metrics with r^2 0.687, mae

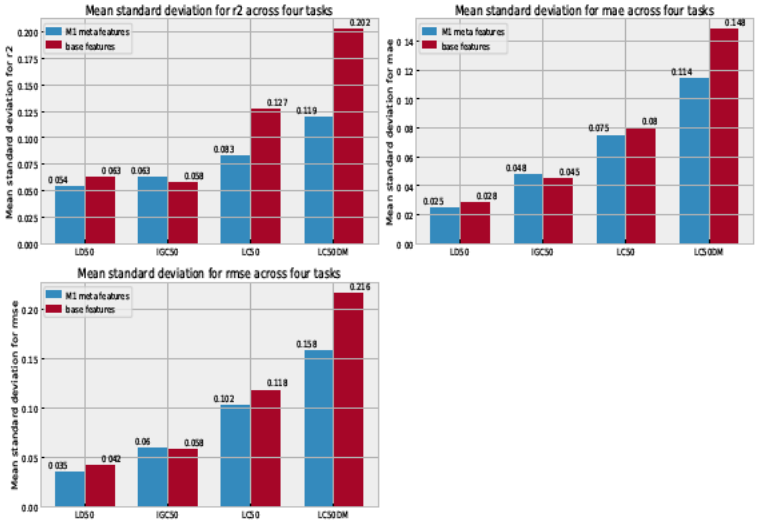

FIGURE 5.6: Comparison between mean standard deviation for base and meta models M1 across all tasks

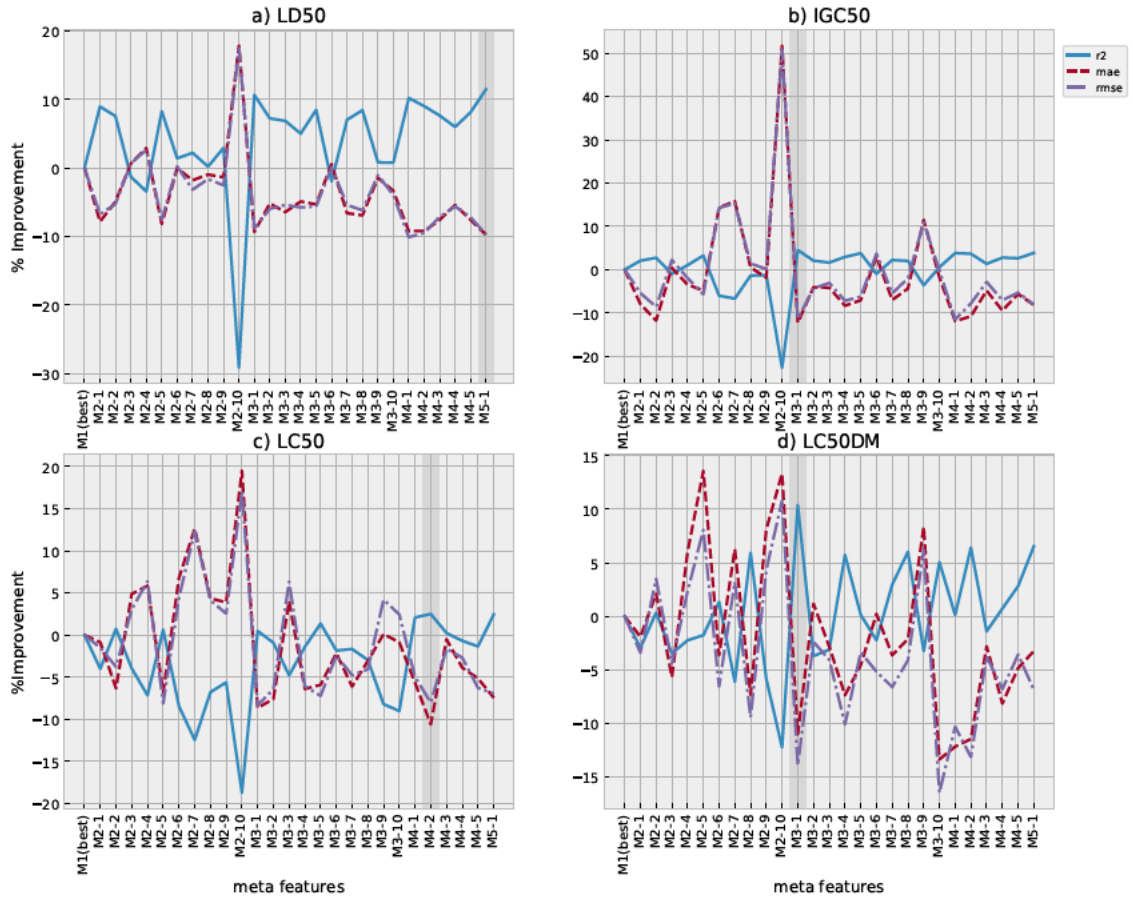

FIGURE 5.7: Performance of meta models in M2-M5 in terms of % improvement over the corresponding best meta models in M1 on the meta validation set.

TABLE 5.6: Comparison of our method with other methods using external independent test sets for all four tasks.

Methods	r^2	mae	rmse	r^2	mae	rmse
	LD_{50}			IGC_{50}		
QuantitativeTox	**0.687**	**0.394**	**0.537**	**0.861**	**0.269**	**0.365**
hierarchical [96]	0.578	0.460	0.650	0.719	0.358	0.539
FDA [96]	0.557	0.474	0.657	0.747	0.337	0.489
group contribution [96]	–	–	–	0.682	0.411	0.575
nearest neighbor [96]	0.557	0.477	0.656	0.6	0.451	0.638
TEST consensus [96]	0.626	0.431	0.594	0.764	0.332	0.475
TopTox [145]	0.653	0.421	0.568	0.802	0.305	0.438
Hybrid2D [68]	0.629	–	–	0.810	–	–
	LC_{50}			LC_{50}-DM		
QuantitativeTox	**0.792**	0.479	**0.668**	**0.808**	**0.520**	**0.754**
hierarchical [96]	0.71	0.574	0.801	0.695	0.757	0.979
single model [96]	0.704	0.605	0.803	0.697	0.772	0.993
FDA [96]	0.626	0.656	0.915	0.565	0.909	1.19
group contribution [96]	0.686	0.578	0.81	0.671	0.62	0.803
nearest neighbor [96]	0.667	0.649	0.876	0.733	0.745	0.975
TEST consensus [96]	0.728	0.545	0.768	0.739	0.727	0.911
TopTox [145]	0.788	**0.446**	0.677	0.788	0.592	0.805
Hybrid2D [68]	0.678	–	–	0.616	–	–

0.394, and rmse 0.537. The results of TopTox method, in all three metrics, are better than those of the TEST software methods but worse than that of our proposed methods. The improvements of our method over the state-of-the-art TopTox method are r^2 5.206%, mae 6.413%, rmse 5.457%.

- **IGC_{50}:** As can be seen, for this data set, TEST consensus obtains the highest r^2 of 0.764 among all the models in TEST software while TopTox model achieved r^2 of 0.802. Our proposed model obtains r^2 of 0.861, which is better than all 6 models including TopTox. The proposed model also obtains better mae and rmse values with 0.269 and 0.365 respectively. The improvements over the state-of-the-art TopTox method is r^2 7.356%, mae 11.803%, rmse 16.666%.

- **LC_{50}:** In this task, our proposed method achieves the best results: 0.792 for r^2 and 0.668 for rmse than those of all other methods. For mae, our proposed method achieves the second best results of 0.479, which is after 0.446 of TopTox.

- **LC_{50}-DM:** This is the smallest data set among all four tasks. In this task, our method obtains the best results for all the three metrics with r^2 0.808, mae 0.520, and rmse 0.754. The improvements over the state-of-the-art method TopTox are r^2 2.538%, mae 12.162%, rmse 6.335%.

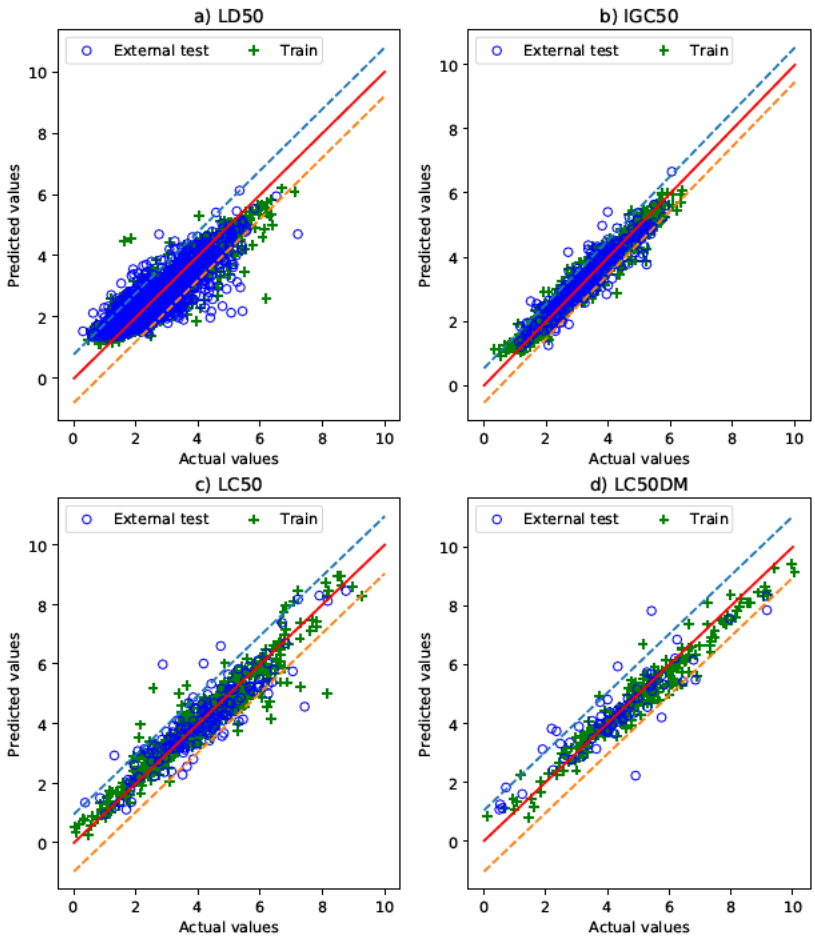

FIGURE 5.8: Prediction confidence interval of each task for full training and external test sets.

5.3.6 Chemical space and prediction confidence

A diverse dataset covering a broad sample space is a prerequisite for building predictive models [17]. For all SMILES strings in the combined train and test sets, we computed the 2048-bit Morgan fingerprints using RDKIT [84]. We then use the t-SNE dimensional reduction technique [95] to convert the 2048 dimensional vectors into two t-SNE dimensions; which are shown in Figure 5.9 with a perplexity value of 30. From the charts, train sets are observed to be covering the test sets for LD_{50}, IGC_{50} and LC_{50} indicating the possibility of highly accurate predictions. However, in LC_{50}-DM, the train set does not cover the test set, which is more diverse. This indicates that prediction will be more challenging for this data set. As shown in Figure 5.5, Table 5.2, Table 5.4, and Table 5.6, this finding is consistent with the comparatively lower performance of our proposed method on LC_{50}-DM data set than on the other three data sets.

For each task separately, we show the prediction confidence for full training as well as external test sets in Figure 5.8. By taking 2 times the mean absolute error for external test sets of each task, above 85% of the predictions are covered in the confidence interval. Prediction confidence for external test sets of LD_{50}, IGC_{50}, LC_{50} and LC_{50}-DM is 87.93%, 87.98%, 87.80% and 85.71% respectively. The prediction confidence for external test set of IGC_{50} is the highest whereas it is lowest for the external test set of LC_{50}-DM. The possible reason for lowest prediction confidence interval for the external test set of LC_{50}-DM might be its exceptionally diverse chemical space as shown in Figure 5.9d.

5.4 Discussion

We discuss how meta ensembling and multi-task learning bring new insights into quantitative toxicity end points and can improve overall performances

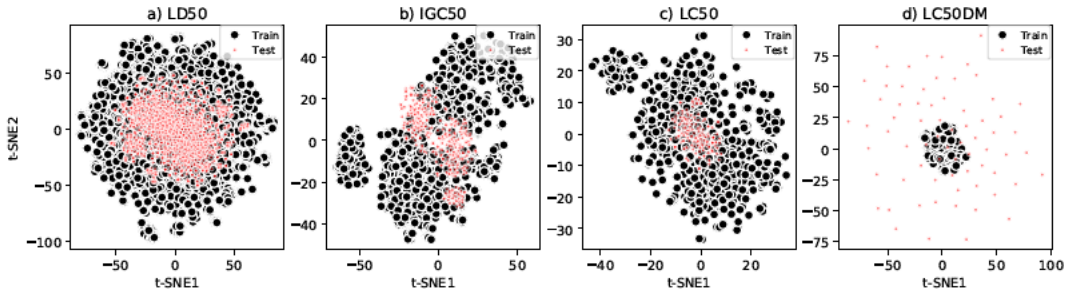

FIGURE 5.9: Two dimensional t-SNE charts showing the dispersion and coverage of the chemical sample space of the train sets over the test sets for the four tasks.

using off the shelf general features.

5.4.1 Impact of multi-task weighted loss function

The effectiveness of multi-tasking which is being proven to be very useful for quantitative toxicity prediction [145] can be further improved using it in meta-ensembling approaches with multi-task weighted loss function optimization. As shown in Figure 5.5, it is clear that a smaller data set such as LC_{50}-DM requires more weight in multi-task loss optimization. Also, tasks with relatively smaller data sets show more fluctuations with changing the loss weight value. For instance, LD_{50} and IGC_{50} are relatively larger data sets than LC_{50} and LC_{50}-DM and thus they are less affected by the loss weight value.

5.4.2 Impact of aggregation of various base features

Representing molecules using just one type of features might not help capture all information about that molecule. For instance, basic molecular graph representations do not capture the quantum mechanical structure of molecules or do not necessarily express the information. Similarly, the models which

use molecular graphs as input like graph convolution will not be able to distinguish between chiral molecules (molecules having the same graph structure with a mirror image of each other). In the case of fingerprints as an input, it is also possible that different molecules may have identical fingerprints and this will make it difficult for a model to distinguish if it only takes fingerprints as input. Features with direct biological relevance such as DESC, MGF and MFP in Table 5.2 prove very useful individually as opposed to those features which do not have any direct biological relevance such as FPeV and SeV. The predictive power DESC, MGF and MFP can be further enhanced when aggregated effectively with FPeV and SeV as shown in Table 5.4 and Figure 5.7. Specifically in case of LC_{50}-DM, the r^2, mae, and rmse are substantially improved by using SeV and FPeV with DESC (M3-10 meta model). This paves a path towards the idea of using weak features (SeV and FPeV) along with strong features (DESC) to further enhance the performance due to strong features. The chemical information captured by weak features might not be significant by itself but can play a vital role when aggregated with the chemical information extracted using strong features. Besides performance improvements, meta ensembling helps stabilize the results by producing low standard deviation in most metrics. In case of smaller data sets such as LC_{50}-DM, meta models M1 decrease the mean standard deviation value substantially as compared to the the mean stand deviation value for base models.

5.5 Conclusion

Quantitative toxicity measurement has paramount importance in pharmaceuticals. Toxicity prediction for chemical compounds recently achieved enhanced performance in terms of accuracy after the introduction of various deep learning models in this space. Usually, molecules are represented by a given type of features and a specific machine learning method is then used to

predict the toxicity. The performance of any quantitative toxicity prediction method depends upon the specific type of features and the learning model used. This restricts the overall performance to a single type of features and a learning model.

In this chapter, we have introduced a deep learning based framework called QuantitativeTox for predicting quantitative toxicity end points or data sets such as LD_{50}, IGC_{50}, LC_{50} and LC_{50}-DM. Our approach has three stages: generating base features, training base learning models on the base features, and training a meta learning model. We use 5 types of base features, and then use five base learning models on them. The outputs of the base learning models are used as the meta features for the meta learning model. To support multi-task training, each model produces four outputs for four data sets and the loss function uses an weighted sum over the data sets.

We have found that high level physicochemical, low level fingerprints, SMILES embedded vectors, and fingerprint embedded vectors when used to create meta features for the meta ensemble model, enhance the performance over a wide range of metrics for the quantitative toxicity prediction tasks. We evaluated our framework against three main regression metrics using independent test sets and obtained a robust performance compared to state of the art methods. Our framework can serve as a robust tool for quantitative toxicity prediction with better aggregation strategy for various features along with individual models and multi-tasking. The software code along with data can be found on https://github.com/Abdulk084/QuantitativeTox. In the next chapter, we will test the idea of meta ensembling approach developed in this chapter for cardiotoxicity in single task manner. We also will show the effectiveness of meta ensembling approach for robust classification of molecular cardiotoxity.

Chapter 6

Robust cardiotoxicity classifier

This chapter is accepted for publication in journal of cheminformatics.

- Karim, A., Lee, M., Balle, T., Sattar, A. (2021). CardioTox net: A robust predictor for hERG channel blockade via deep learning meta ensembling approaches. **(Accepted for publication)**

In this chapter, we use similar architecture which we developed in chapter 5 and applied it to cardiotoxicity data. The cardiotoxicity data consists of molecules which are evaluated for their potential to block ether-a-go-go-related gene (hERG) channel. Ether-a-go-go-related gene (hERG) channel blockade by small molecules is a major concern during drug development in the pharmaceutical industry. Blockade of hERG channels may cause prolonged QT intervals that potentially could lead to cardiotoxicity. Various in-silico techniques including deep learning models are widely used to screen out small molecules with potential hERG related toxicity. Most of the published deep learning methods utilize a single type of features which might restrict their performance. Methods based on more than one type of features struggle with the aggregation of extracted information and show better performance when evaluated against a single accuracy metric but struggle when evaluated against others. Therefore, there is a need for a method that

can efficiently aggregate information gathered from models based on different chemical representations and boost hERG toxicity prediction over a range of performance metrics.

In this chapter, we use deep learning framework developed in chapter 5 to predict hERG channel blocking activity of small molecules. As described 5, our approach utilizes five individual deep learning base models with their respective base features and a separate neural network to combine the outputs of the five base models. By using the same training and external test data with potency activity of IC_{50} at a threshold of 10 µM as that of state-of-the-art DeepHIT, we improved Matthew correlation coefficient (MCC) by 25.84%, specificity (SPE) by 22.23%, positive predictive value (PPV) by 7.20% and accuracy (ACC) by 4.78% with no loss of sensitivity (SEN) and negative predictive value (NPV) over DeepHIT. In addition, on newly prepared independent external test set with the same potency activity and threshold, our method improved MCC by 13.56%, SPE by 12.57%, PPV by 9.11% and ACC by 4.71% over DeepHIT. We also investigate the effective aggregation of chemical information extracted for robust hERG activity prediction. In summary, our framework (CardioTox) can serve as a robust tool for screening small molecules for hERG channel blockade in drug discovery pipelines and performs better than previously reported methods on a range of classification metrics.

6.1 Introduction

Models in most of the previous studies either utilize a single type of features which might restrict the performance or struggle with the aggregation of extracted chemical information from various types of features [16, 18, 85, 86]. Thus they achieve reasonable performance when evaluated against one accuracy metric, but falls behind when evaluated against others. For instance,

CardPred used a total of 3456 physicochemical descriptors and fingerprints with six individual machine learning models [85] to achieve reasonable performance when evaluated against accuracy (ACC) and positive predictive value (PPV) but performed poorly when evaluated against other metrics such as Matthew correlation coefficient (MCC), negative predictive value (NPV), specificity (SPE), sensitivity (SEN) (evaluated on external test sets as reported in the results section) [120]. A method reported by Cai et al.. relies on physicochemical descriptors and molecular vectors combined together as a single input for a fully connected multi-task deep neural network to achieve better performance for various metrics except NPV (for their internal cross validation datasets).

Li et al. used 8 different types of machine learning models and their ensemble with physicochemical descriptors and fingerprints performed well when evaluated against SPE and PPV but less so for other metrics. The key to success for these previous methods for hERG activity prediction is elucidating correct structure-property relationships from existing data using high level physicochemical features along with fingerprints. Recently the DeepHIT method was introduced which utilizes physicochemical descriptors, fingerprints and graph features with fully connected deep neural networks and graph convolution neural networks to achieving better performance for hERG activity prediction [120]. DeepHIT classifies a molecule as a hERG blocker if at least one model out of the three models used predicts a given molecule as a hERG blocker [120], thus enhancing the sensitivity of the model. Although DeepHIT utilize reasonably diverse feature set, it still lacks in an effective way of combining the outputs of individual models for robust performance over a range of metrics.

We hypothesize that if we can extract chemical information from all or the subsets of three levels of features (low, high and intermediate) and their variants for molecular hERG activity prediction, we can improve upon the

performance over a wide range of accuracy metrics. For this purpose, we customized our developed framework in chapter 5 for single task classification and called it "CardioTox". For two different external test sets, CardioTox net improves the Matthew correlation coefficient (25.84%, 13.56%), accuracy (4.78%, 4.71%), positive predictive value (7.20%, 9.11%) and specificity (22.23%, 12.57%) while keeping the sensitivity same as the so far best in class method, DeepHIT. Our framework consists of three stages; A featurization stage which generates base features; an individual prediction stage which uses base features with the base individual deep learning models to generate the outputs also called meta features; and a meta ensemble stage which uses meta features generated by the previous stage to classify the molecule as hERG blocker or hERG non-blocker.

6.2 Materials and methods

In this section, we explain the data preparation method for training as well as both external test sets. We then describe the architecture details of the deep learning framework used in this study. It should be noted that the main architecture including the featurization stage, individual prediction stage and meta ensemble stage is similar to the one used in chapter 5. For the purpose of this study, we customized it to a single task classification.

6.2.1 Data preparation

A dataset consisting of molecular structures labelled as hERG and non-hERG blockers in the form of SMILES strings was obtained from the DeepHIT authors [120] and was curated from five sources, the BindingDB database (3056 hERG blockers, 3039 hERG non-blockers) [44], ChEMBL bioactivity database (4859 hERG blockers, 4751 hERG non-blockers) [41], and literature derived (4355 hERG blockers, 3534 hERG non-blockers) [18], (1545 hERG blockers,

816 hERG non-blockers) [35], (2849 hERG blockers, 1202 hERG non-blockers) [34] and unlike in the DeepHIT procedure, we did not use any in-house data. A total of 30000 molecular structures were obtained and were standardized using using RDKIT `http://www.rdkit.org/` and MolVS `https://molvs.readthedocs.io/en/latest/` as described by Ryu et al [120]. We further removed duplicates and mislabeled compounds. Thus we obtained total of 12620 molecules with 6643 labelled as hERG blockers and 5977 as hERG non-blockers to constitute our training set. We evaluated our framework against two external independent test sets, one of which was obtained from the authors of DeepHIT [120], hereafter called test-set I which is positively imbalanced (i.e. more blockers (30) than non-blockers (14)). We also retrieved another independent test set (test-set II) from [127] and [81] as per the criteria of half maximal inhibitory concentration (IC_{50}) values < 10 µM considered to be hERG blockers and (IC_{50}) values ≥ 10 µM considered to be hERG non-blockers. Test-set II was negatively imbalanced with 11 blockers and 30 non-blockers. The Tanimoto similarity [120] criteria was also ensured for all molecules in both test and training sets (explained in upcoming section of similarity and chemical diversity). The training set was subdivided into four sets, 70% for training the base models, 10% for validating base models, 10% for training the meta ensemble model and 10% for validating the meta ensemble model.

All redundant molecules were removed and respective sets were merged together to form a combined base training set, base validation set, meta training set and meta validation set. We used test set-I from DeepHIT "as is" which contains more hERG blockers than non-blockers. Pairwise Tanimoto similarity [120] was computed between all molecules of combined data sets with those of molecules in test set-I obtained from DeepHIT. All those molecules in the combined data sets, the Tanimoto similarity of which are >0.7 to any of the molecule in test set-I were removed, thus forming a gold standard

training and validation data as shown in Figure 6.1a.

In order to evaluate our model on another independent test set which should contain more non blockers molecules, we curated 110 hERG blockers and 336 hERG non-blockers from "E3 training" set of Siramshetty et al.. The reason we curated from E3 training is because it contains molecules with potency threshold (IC_{50}) values < 10 µM considered to be hERG blockers and (IC_{50}) values ≥ 10 µM considered to be hERG non-blockers which is compatible with other datasets used in our study. Besides, E3 training is also negatively imbalanced which contains more non-blockers than blocker molecules, as test set-II is aimed to be negatively imbalanced unlike test set-I which is positively imbalanced. We also obtained 9250 molecules from Konda et al. with pIC_{50} values as potency threshold. We converted the unit of potency from pIC_{50} to IC_{50} and labelled molecules with (IC_{50}) values < 10 µM as hERG blockers and (IC_{50}) values ≥ 10 µM as hERG non-blockers. Both data sets were merged together and all those molecules with Tanimoto similarity > 0.7 to any molecule in gold standard data training and validation or test set-I were removed. Thus we obtained test set-II which contains more non blocker molecule than blockers and is dissimilar to both gold standard training and validation as well as test set-I as shown in Figure 6.1b.

6.2.2 Similarity and chemical diversity

A diverse dataset covering a broad chemical space is a prerequisite for building predictive models [17]. For all SMILES strings in training as well as in both external test sets, we computed the 2048 bit Morgan fingerprints using RDKIT [**?**]. The t-SNE dimensional reduction technique [95] was then used to convert the 2048 dimensional vector into two t-SNE dimensions for each SMILES string. As demonstrated by the chemical space defined by the t-SNE components in Figure 6.2, diverse chemical space distributions for classified

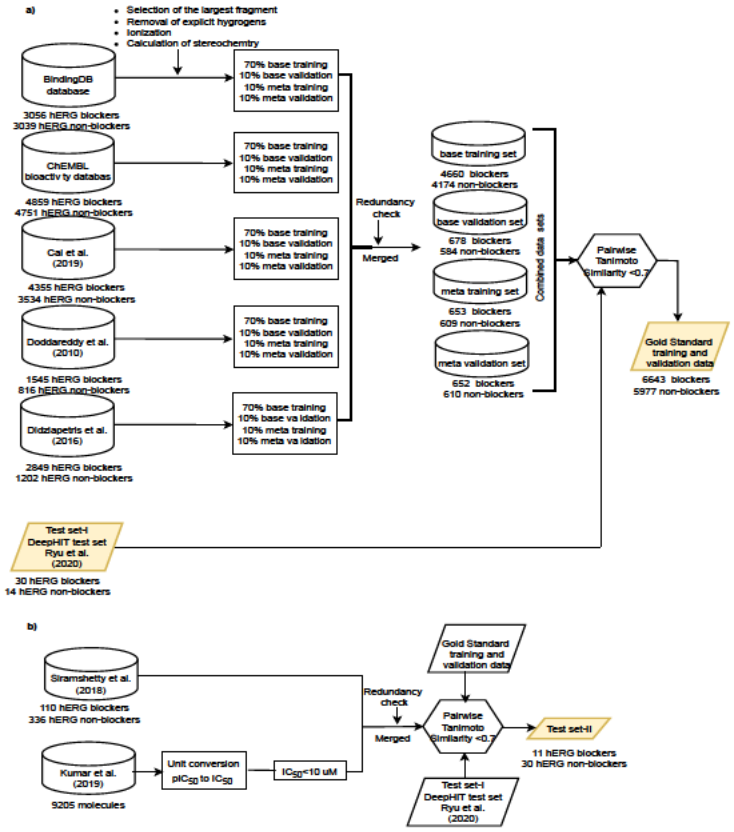

FIGURE 6.1: a) Preparation of gold standard training data as per the procedure described in DeepHIT. b) Preparation of external independent negatively biased test set-II

blockers and non-blockers as well as overlap with the external tests sets was observed. We computed the Tanimoto mean value for each of the datasets separately given in Table 6.1 and a pairwise Tanimoto similarity shown in Figure 6.3 for all three datasets. The Tanimoto mean value shows the mean Tanimoto similarity within each data set whereas pairwise Tanimoto similarity shows similarity between different datasets. The lower the Tanimoto mean value is, the better the diversity of the compounds within the data set. As illustrated in Table 6.1, the Tanimoto mean value is 0.124 for the training set, 0.126 for the external test-set I, and 0.116 for the external test-set II, which means all the three data sets are diverse. Pairwise Tanimoto similarity as shown in Figure 6.3 for external test sets, with respect to the training set is always less than 0.7. The external test-set I is also substantially dissimilar to the external test-set II as the maximum pairwise Tanimoto similarity value is

less than 0.5 as shown in Figure 6.3c.

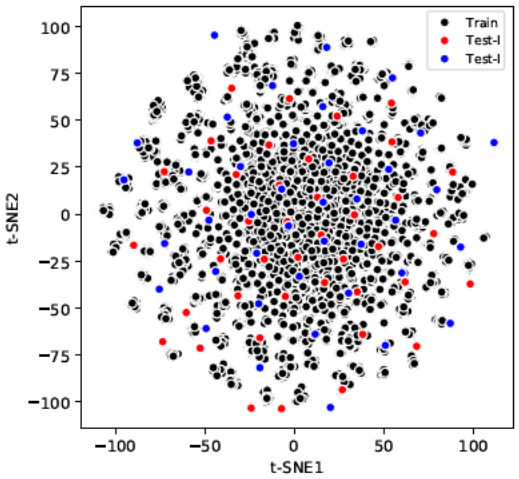

FIGURE 6.2: Two dimensional t-SNE components showing the chemical space diversity of training and the two external test sets.

TABLE 6.1: Statistical description of cardiotoxicity data

Data set	Activity	Threshold	hERG blockers	hERG non-blockers	Total	Tanimoto mean
Training set	IC_{50}	10 µM	6643	5977	12620	0.124
Test set-I	IC_{50}	10 µM	30	14	44	0.136
Test set-II	IC_{50}	10 µM	11	30	41	0.116

6.2.3 Featurization stage

The featurization stage of our framework consists of various types of featurizers which takes SMILES string as an input and produce fixed length base features as shown in Figure 6.4a.

Descriptors

A total of 995 high level features such as 2D and 3D physicochemical descriptors (DESC) were computed using Mordred [103], names of which are also given in Table S2 of DeepHIT supplementary material [120]. These features are numerical in nature and describe the physical and chemical properties of molecules [22]. 2D descriptors represents information related to size, shape, distribution of electrons, ocatnol-water distribution coefficient (LogP) which

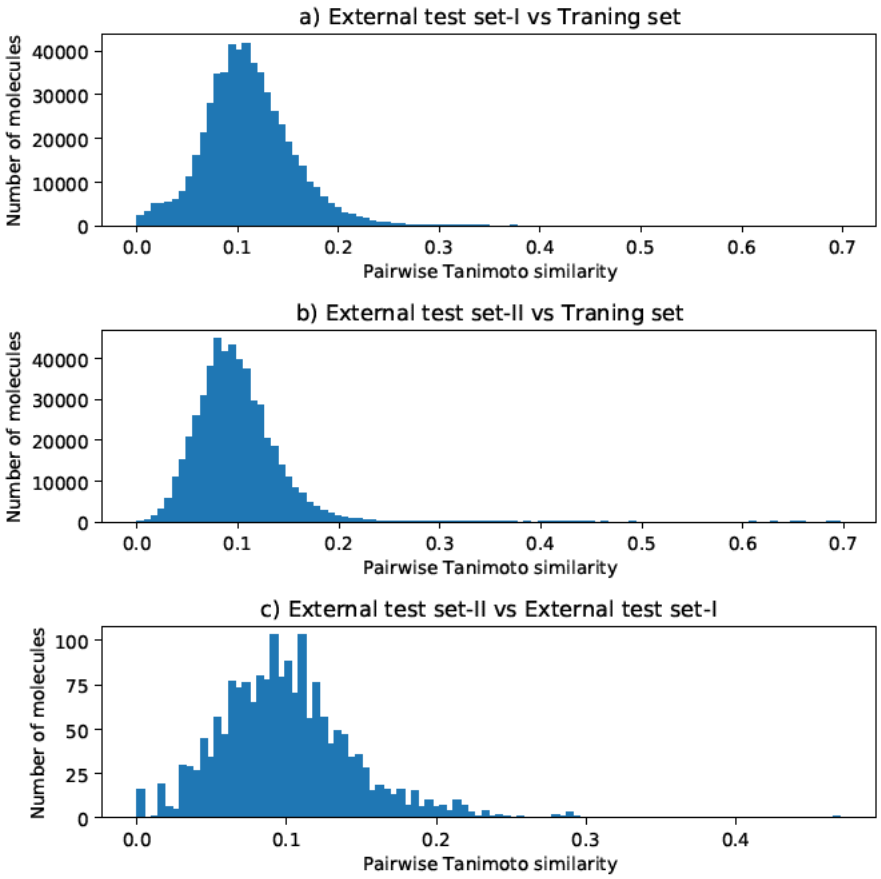

FIGURE 6.3: Pairwise Tanimoto similarity for each molecule in a) external test-set I with all molecules in training set. b) external test-set II with all molecules in training set. c) external test-set I with all molecules external test-set II.

is a measure for lipophilicity, nAromAtom which shows number of aromatic atoms, nHeavyAtom which shows number of heavy atoms, nBondsT shows number of triple bonds. 3D descriptors relates to the 3D conformation of the molecules such as moment of inertia along Y axis (MOMIY) [22]. The value of each descriptor was normalized between 0 and 1.

Molecular graph featurizer

Topological information of molecules can be intuitively and concisely expressed via molecular graph features. This intermediate level featurizer computes molecular graph features such as node vectors which represents atoms

in the SMILES string and an adjacency matrix which shows the bonds between atoms [121]. In this study, we extracted the same graph features as were extracted for DeepHIT [120], i.e a [50x65] node vector and a [50x50] adjacency matrix, details of which are also given in Table S3 of DeepHIT supplementary material [120]. Here 50 refers to the maximum number of atoms and 65 refers to the one hot-encoded feature vector computed from atom descriptors [120].

Molecular fingerprint generator

The third featurizer deals with fingerprints where structural features are represented by either bits in a bit string or counts in a count vector [116, 136]. 1024 extended-connectivity fingerprints with a maximum diameter parameter of 2 (EFCP2) fingerprints and 881 pubChem fingerprints were computed using using the Python package PyBioMed [36, 120]. EFCP are also referred to as circular fingerprints and are specifically designed for structure-activity relationship modeling [117] whereas pubChem fingerprints are mainly designed for similarity neighboring and similarity searching [53].

SMILES vectorizer

We also computed two variants of low level features, SMILES strings embedded vectors (SeV) [45, 70] and fingerprint based embedded vectors (FPeV) [66] which themselves do no directly describe any biological attribute of the molecules, but has proven to have a reasonable predictive power in various quantitative structure-activity relationship (QSAR) tasks. In the SMILES vectorizer, we created a vocabulary based on the valid SMILES tokens. A total of 64 unique tokens were determined based on the training data. The longest SMILES string in the data considered for this study was 97. Each SMILES string was converted into a one-hot encoded vector based on the SMILES vocabulary.

Fingerprints vectorizer

In the fingerprint vectorizer, SMILES string are converted into 1024 bit Morgan (or circular) fingerprints with a radius of 2 via RDKIT [**?**]. As per the previously published technique [66], we extracted fingerprint indices which were marked 1 in the fingerprint generated. Thus we obtained a vector of length 93 which consisted of integers representing presence of specific substructures in a molecule. The procedure for fingerprint embedding vector is described in Figure 1 of FP2VEC [66].

6.2.4 Individual prediction stage

The individual prediction stage consists of base models which are trained on respective base features from the featurization stage. All of the base models were trained at a learning rate of $10e^{-4}$ with an Adam optimizer and 100 epochs with a batch size of 32. Selection of parameters, hyper-parameters and network architecture of base models were inspired from the previous published research in this area [45, 66, 68, 69, 70, 85, 120]. Each of these base models produce an output which is a single probability of a molecule being a hERG blocker. Here we describe each base model in the individual prediction stage also shown in Figure 6.4b,c,d,e. The Keras deep learning framework and Spektral package was used in developing base models for the individual prediction stages [25, 52].

Fully connected neural network for descriptors (FCNND)

A fully connected deep neural network with 4 hidden layers was trained and validated on 995 2D and 3D physicochemical descriptors. The input layer consists of 995 nodes as per the number of total physicochemical descriptors and an output layer with 1 unit. All the layers in FCNND are densely connected and receives input from all the units present in the previous layer.

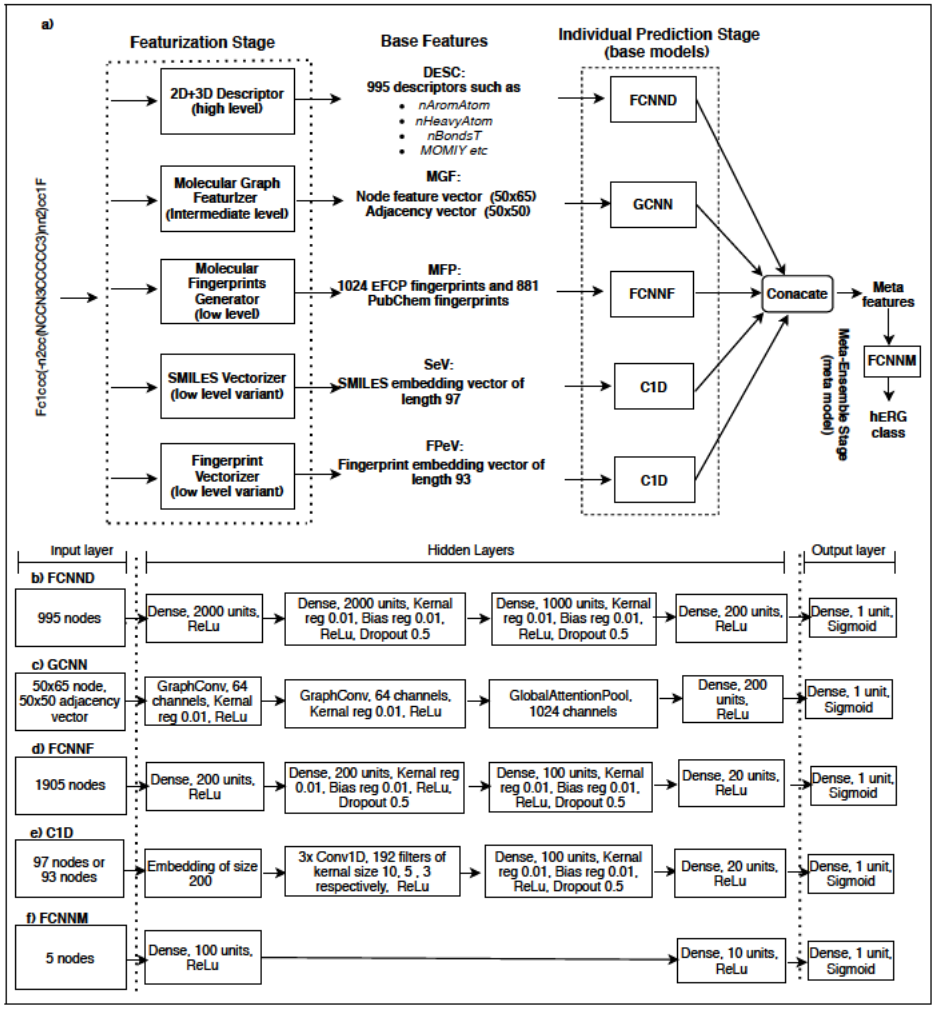

FIGURE 6.4: a) CardioTox framework: End to end flow diagram of all the stages of proposed framework. b) Architecture specifications of fully connected neural network for 995 2D and 3D descriptors as base features. c) Architecture specifications of graph convolutional neural network for node vector of size 50x65 and adjacency vector of size 50x50 as base features. d) Architecture specifications of fully connected neural network for 1024 EFCP and 881 pubChem fingerprints as base features e) Architecture specifications of 1D convolution neural network for SMILES and fingerprints embedding vectors as base features. f) Architecture specifications of meta ensemble fully connected neural network for meta features

The number of units in each hidden layer is decreased gradually and a ReLu activation [51, 65] is applied at the end of each layer. Various regularization parameters such as Kernel regularizer which applies penalties to the kernel (main units in layer), bias regularizer which applies penalties to the bias units to reduce the over-fitting during optimization [51, 54]. We also applied a drop-out rate of 0.5 to the middle layers [130].

Graph convolutional neural network for graph features (GCNN)

A graph convolutional neural network (GCNN) was trained using the graph features as shown in Figure 6.4c. GCNN consists of two graph convolution layers [80], one global attention pool layer [87] and a dense layer before the output. Each of the graph convolutional layers were initiated with 64 channels with a kernel regularization value of 0.01 and a ReLu activation. The number of channels in the global attention pool layer was made equal to the number of units in the following dense layer, i.e 1024.

Fully connected neural network for fingerprints (FCNNF)

A fully connected neural network was used with fingerprints (FCNNF) as the base feature. Unlike FCNND, FCNNF uses a much smaller number of units in each layer. Except the number of units, other parameters were kept the same as in FCNND. The number of input nodes in the input layer were kept at 1905 to match the sum of 1024 EFCP fingerprints and 881 pubChem fingerprints as shown in part Figure 6.4d.

Convolution 1D neural network for SMILES and fingerprint embedding vectors (C1D)

For models where SMILES and fingerprint embedding vectors were used as base features, we used a variant of a Convolution 1D Neural Network (C1D)

as base model as shown in Figure 6.4e. The only difference was in the number of input-layer nodes which was 97 for SMILES embedding vectors and 93 for fingerprint embedding vectors. Input vectors were converted to a trainable embedding matrix of the size [97 or 93 x 200] which was then fed into a series of three 1D convolution layers. Each of these 1D convolution layers used ReLu activation, 192 filters with a kernel size of 10, 5 and 3 respectively. Two densely connected layers with the parameters shown in Figure 6.4e are also used to before the output layer.

6.2.5 Meta ensemble stage

The outputs of each of the base models in the individual prediction stage were concatenated to produce meta features for the meta ensemble model. The Meta ensemble model is a fully connected neural network (FCNNM) with an input, output and two hidden layers as shown in Figure 6.4f. It is trained at a learning rate of $10e^{-3}$ with an Adam optimizer and 300 epochs with a batch size of 32.

6.3 Results and discussion

Our proposed framework employs step-wise training to produce the final classification of molecules as hERG or non-hERG blockers. For this purpose, data was divided into four sets, base training set: 70% for training base models , base validation set: 10% for validating base models, meta training set: 10% for training meta-ensemble model and meta validation set: 10% for validating the meta-ensemble model. In the first step of training, all the base models were trained on the base training set and validated using the base validation set. In the second step, the outputs of the best performing base models for the base validation set were used as meta features to train the

meta ensemble model with the meta training set. We used the meta validation set to obtain the best meta ensemble model and also to select which combination of the base models ensembling produces better results. We repeated this process for 10 fold times and report the results as follows.

6.3.1 Validation of base model performance

The 10 fold cross validated results for individual base models of our framework on base validation set are shown in Table 6.2. Each base model is trained and validated with its own respective base features independently. In the Table 6.2, DESC refers to high level features such as 2D and 3D descriptors feeding the FCNND, MGF refers to intermediate molecular graph features fed into GCNN, MFP refers to low level molecular fingerprints fed into FCNNF, SeV refers to one of the low level variant i.e, SMILES embedding vectors when used with C1D and FPeV refers to low level variant i.e, fingerprint embedding vectors when used with C1D.

TABLE 6.2: 10 fold cross validated performance of the base models in individual prediction stage on base valid set using their respective base features

Base features	MCC	NPV	ACC	PPV	SPE	SEN	AUC
DESC	0.689	0.813	0.845	0.870	0.868	0.822	0.911
MGF	0.620	0.805	0.810	0.817	0.794	0.826	0.888
MFP	0.683	0.830	0.841	0.855	0.837	0.845	0.915
FPeV	0.638	0.814	0.818	0.826	0.802	0.835	0.899
SeV	0.636	0.811	0.817	0.827	0.809	0.826	0.889

As shown in Table 6.2, DESC performed better in MCC, ACC and PPV whereas MFP performed better in NPV, SEN and AUC. The possible reason might be the the direct biological relevance of these base features (descriptors and fingerprints) to the activity prediction. Interestingly, SeV and FPeV showed better performance than MGF despite no biological relevance of the features used. FPeV and SeV achieved almost similar performance in

most the of performance metrics. MGF legs behind in most of the metrics except SEN where it achieved slightly better performance than DESC. Table 6.3 shows standard division for each split of base validation set in training the individual base models.

TABLE 6.3: Standard deviation values for 10 fold cross validated performance of the base models in individual prediction stage on base validation set using their respective base features.

Base features	MCC	NPV	ACC	PPV	SPE	SEN	AUC
DESC	0.024	0.019	0.012	0.021	0.025	0.028	0.011
MGF	0.020	0.022	0.010	0.025	0.034	0.024	0.006
MFP	0.016	0.023	0.008	0.026	0.035	0.027	0.007
MFP	0.016	0.031	0.008	0.026	0.042	0.038	0.007
FPeV	0.024	0.028	0.012	0.024	0.040	0.034	0.010
SeV	0.018	0.021	0.009	0.019	0.036	0.037	0.007

6.3.2 Validation of meta model performance

The overall goal of this study is to aggregate the chemical information extracted from various base features for cardio-toxicity data set so that the classification performance can be improved over a wide range of metrics. For that purpose, the outputs of the base models are concatenated to produce meta features for the use of a meta ensemble model as shown in Figure 6.4a. A separate meta training set and meta validation set is used for training and validating the meta ensemble model. Table 6.4 demonstrates 10 fold cross validation results for the meta validation set for ensembling all possible unique combinations of base features ranging from 1 to 5. For instance, M1 represents single type of base features used in creating meta features whereas M2, M3, M4 and M5 represents any two, three, four and 5 different types of the base features with no repetitions.

It can be seen from Table 6.4 that meta features in M3 and M4 show overall better performance for most of the metrics. In the M4 meta-feature category, M4-5 achieves the best results of MCC: 0.720, ACC: 0.860, PPV: 0.871 and

TABLE 6.4: 10 fold cross validation results for various meta features on meta validation set

Meta Features	Base features	MCC	NPV	ACC	PPV	SPE	SEN	AUC
M1-1	DESC, DESC	0.676	0.829	0.838	0.862	**0.868**	0.819	0.909
M1-2	MGF, MGF	0.599	0.784	0.799	0.815	0.792	0.806	0.878
M1-3	MFP, MFP	0.682	0.829	0.840	0.853	0.838	0.843	0.909
M1-4	FPeV, FPeV	0.636	0.820	0.817	0.819	0.795	0.839	0.897
M1-5	SeV, SeV	0.621	0.806	0.809	0.816	0.791	0.828	0.880
M2-1	MGF, MFP	0.691	0.826	0.846	0.864	0.850	0.842	0.919
M2-2	MGF, DESC	0.683	0.818	0.842	0.865	0.848	0.835	0.914
M2-3	MGF, SeV	0.685	0.837	0.842	0.848	0.830	0.854	0.916
M2-4	MGF, FPeV	0.682	0.828	0.841	0.854	0.833	0.848	0.916
M2-5	MFP, DESC	0.710	0.843	0.855	0.866	0.855	0.855	0.928
M2-6	MFP, SeV	0.698	0.838	0.849	0.861	0.844	0.853	0.921
M2-7	MFP, FPeV	0.690	0.831	0.845	0.859	0.840	0.850	0.920
M2-8	DESC, SeV	0.707	0.847	0.853	0.860	0.846	0.861	0.926
M2-9	DESC, FPeV	0.715	0.848	0.857	0.867	0.859	0.856	0.929
M2-10	SeV, FPeV	0.680	0.828	0.840	0.853	0.835	0.845	0.918
M3-1	MGF, MFP, DESC	0.707	0.851	0.853	0.857	0.841	0.866	0.924
M3-2	MGF, MFP, SeV	0.711	**0.855**	0.855	0.857	0.835	**0.874**	0.927
M3-3	MGF, MFP, FPeV	0.701	0.849	0.850	0.853	0.833	0.867	0.921
M3-4	MGF, DESC, SeV	0.710	0.847	0.855	0.864	0.849	0.861	0.926
M3-5	MGF, DESC, FPeV	0.706	0.853	0.852	0.855	0.831	**0.874**	0.928
M3-6	MGF, SeV, FPeV	0.697	0.844	0.849	0.854	0.838	0.859	0.925
M3-7	MFP, DESC, SeV	0.718	0.854	0.859	0.865	0.850	0.868	**0.930**
M3-8	MFP, DESC, FPeV	0.710	0.850	0.855	0.861	0.846	0.864	0.926
M3-9	MFP, SeV, FPeV	0.699	0.837	0.849	0.862	0.848	0.851	0.925
M3-10	DESC, SeV, FPeV	0.712	0.846	0.856	0.866	0.854	0.858	0.928
M4-1	MGF, MFP, DESC, SeV	0.711	0.850	0.855	0.861	0.841	0.869	0.927
M4-2	MGF, MFP, DESC, FPeV	0.719	0.851	**0.860**	0.869	0.853	0.867	0.929
M4-3	MGF, MFP, SeV, FPeV	0.705	0.846	0.852	0.859	0.846	0.859	0.921
M4-4	MGF, DESC, SeV, FPeV	0.707	0.849	0.853	0.859	0.841	0.865	0.926
M4-5	MFP, DESC, SeV, FPV	**0.720**	0.849	**0.860**	**0.871**	0.856	0.864	**0.930**
M5-1	MGF, DESC, SeV, FPeV, MFP	0.717	0.853	0.858	0.864	0.850	0.867	0.925

AUC: 0.930. In the M3 meta-feature category, M3-2 achieves the best results for NPV: 0.855 and SEN: 0.874. M3-5 also achieves similar performance of 0.874 for SEN to that of M3-2. Similarly for AUC, M3-7 achieves a similar performance of 0.930 compared to that of M4-5. For SPE however, none of the base-feature combinations (ranging from M2 to M5) improves the performance over M1-1 which is 0.868. Interestingly for SPE, the individual lower performance of MGF, FPeV and SeV (M1-2: 0.792, M1-4: 0.795 and M1-5: 0.791) is improved substantially with meta features comprised of any of the combinations (M2-3: 0.830, M2-4: 0.833 and M2-10: 0.835). This improvement offers some perspective on potentially better ensembling performance even if the individual performance is relatively lower for MGF, FPeV and

SeV. Table 6.5 shows standard division for each split of meta validation set in 10 fold validation process.

TABLE 6.5: Standard deviation values for 10 fold cross validation results for various meta features on meta validation set.

Meta Features	Base features	MCC	NPV	ACC	PPV	SPE	SEN	AUC
M1-1	DESC, DESC	0.022	0.013	0.011	0.014	0.017	0.025	0.008
M1-2	MGF, MGF	0.023	0.018	0.012	0.020	0.034	0.022	0.008
M1-3	MFP, MFP	0.021	0.023	0.011	0.025	0.038	0.030	0.008
M1-4	FPeV, FPeV	0.034	0.025	0.018	0.028	0.044	0.028	0.011
M1-5	SeV, SeV	0.019	0.025	0.010	0.025	0.042	0.031	0.007
M2-1	MGF, MFP	0.019	0.015	0.009	0.014	0.017	0.012	0.007
M2-2	MGF, DESC	0.019	0.015	0.009	0.014	0.017	0.012	0.007
M2-3	MGF, SeV	0.015	0.019	0.008	0.020	0.025	0.023	0.005
M2-4	MGF, FPeV	0.019	0.013	0.010	0.021	0.029	0.017	0.004
M2-5	MFP, DESC	0.014	0.012	0.007	0.016	0.022	0.015	0.007
M2-6	MFP, SeV	0.025	0.017	0.012	0.017	0.032	0.020	0.006
M2-7	MFP, FPeV	0.024	0.023	0.012	0.018	0.023	0.021	0.006
M2-8	DESC, SeV	0.020	0.021	0.010	0.020	0.024	0.021	0.005
M2-9	DESC, FPeV	0.023	0.018	0.011	0.021	0.027	0.023	0.009
M2-10	SeV, FPeV	0.019	0.013	0.009	0.012	0.020	0.012	0.005
M3-1	MGF, MFP, DESC	0.017	0.014	0.009	0.011	0.019	0.017	0.006
M3-2	MGF, MFP, SeV	0.023	0.019	0.012	0.025	0.032	0.022	0.006
M3-3	MGF, MFP, FPeV	0.025	0.019	0.013	0.021	0.027	0.017	0.009
M3-4	MGF, DESC, SeV	0.021	0.026	0.011	0.023	0.029	0.029	0.008
M3-5	MGF, DESC, FPeV	0.015	0.014	0.007	0.009	0.022	0.016	0.009
M3-6	MGF, SeV, FPeV	0.016	0.013	0.008	0.018	0.019	0.009	0.008
M3-7	MFP, DESC, SeV	0.014	0.027	0.006	0.020	0.024	0.027	0.006
M3-8	MFP, DESC, FPeV	0.009	0.018	0.004	0.013	0.015	0.019	0.005
M3-9	MFP, SeV, FPeV	0.008	0.018	0.004	0.015	0.023	0.022	0.004
M3-10	DESC, SeV, FPeV	0.021	0.026	0.010	0.012	0.017	0.028	0.007
M4-1	MGF, MFP, DESC, SeV	0.021	0.022	0.010	0.017	0.023	0.021	0.008
M4-2	MGF, MFP, DESC, FPeV	0.025	0.021	0.012	0.020	0.024	0.020	0.007
M4-3	MGF, MFP, SeV, FPeV	0.020	0.017	0.010	0.016	0.020	0.018	0.009
M4-4	MGF, DESC, SeV, FPeV	0.012	0.017	0.006	0.015	0.023	0.020	0.005
M4-5	MFP, DESC, SeV, FPV	0.020	0.015	0.010	0.013	0.017	0.017	0.006
M5-1	MGF, DESC, SeV, FPeV, MFP	0.021	0.017	0.010	0.015	0.021	0.020	0.008

6.3.3 Effectiveness of meta features

In order to investigate the effectiveness of meta features (M2-M5) as compared to the ones which use only single individual base features (M1), we computed % improvement of each of the meta feature ranging from M2 to M4 over best M1 on the meta validation set as shown in Figure 6.5a. An overall improvement can be observed in MCC, NPV, ACC, SEN and AUC. For PPV, more fluctuations across zero axis are observed for various meta features. For SPE, there is overall decrease in performance with relatively

bigger fluctuations on the negative side. It can be observed from Figure 6.5a and Table 6.4 that for meta feature M4-5, 4 out of 7 metrics shows improvement as compared to best M1. Thus we select meta feature M4-5 as the final unique combination of base features for our CardioTox net framework for further analysis and final evaluation against external test sets.

In Figure 6.5b, we show the % difference of CardioTox and DeepHIT from their respective best base model performances for various performance metrics. The values in Figure 6.5b are retrieved from Table 2 of the DeepHIT publication [120] and Table 6.4 for CardioTox. As shown in Table 2 of DeepHIT, the best performance is shown by Descriptor-based DNN for all metrics. DeepHIT is optimized for SEN and NPV with a substantial sacrifice of MCC, ACC, PPV and SPE. It improves SEN by 12.48% and NPV by 9.59% with a sacrifice of 4.47% MCC, 2.87% ACC, 10.63% PPV and 18.09% SPE. On the other hand, CardioTox improves MCC by 5.7%, NPV by 2.34%, ACC by 2.37%, PPV by 1.15% and SEN by 2.52% with a sacrifice of 1.39% in SPE only. With an overall improvement in nearly all the metrics for a relatively little sacrifice of SPE as compared to DeepHIT, CardioTox performance can be considered more robust.

6.3.4 Comparative landscape using the external independent test sets

We compared CardioTox net results with state of the art methods such as DeepHIT [120], CardPred [85], OCHM Predictor-I and OCHM Predictor-II [86] and Pred-hERG 4.2 [16] on two external test sets given in Table 6.6. For both test sets, CardioTox achieves improved performance for all metrics except SEN where its performance is the same as achieved by second best method DeepHIT. This improvement is significant for MCC (25.84%, 13.56%), PPV (7.20%, 9.11%) and SPE (22.23%, 12.57%) over DeepHIT. The SEN is

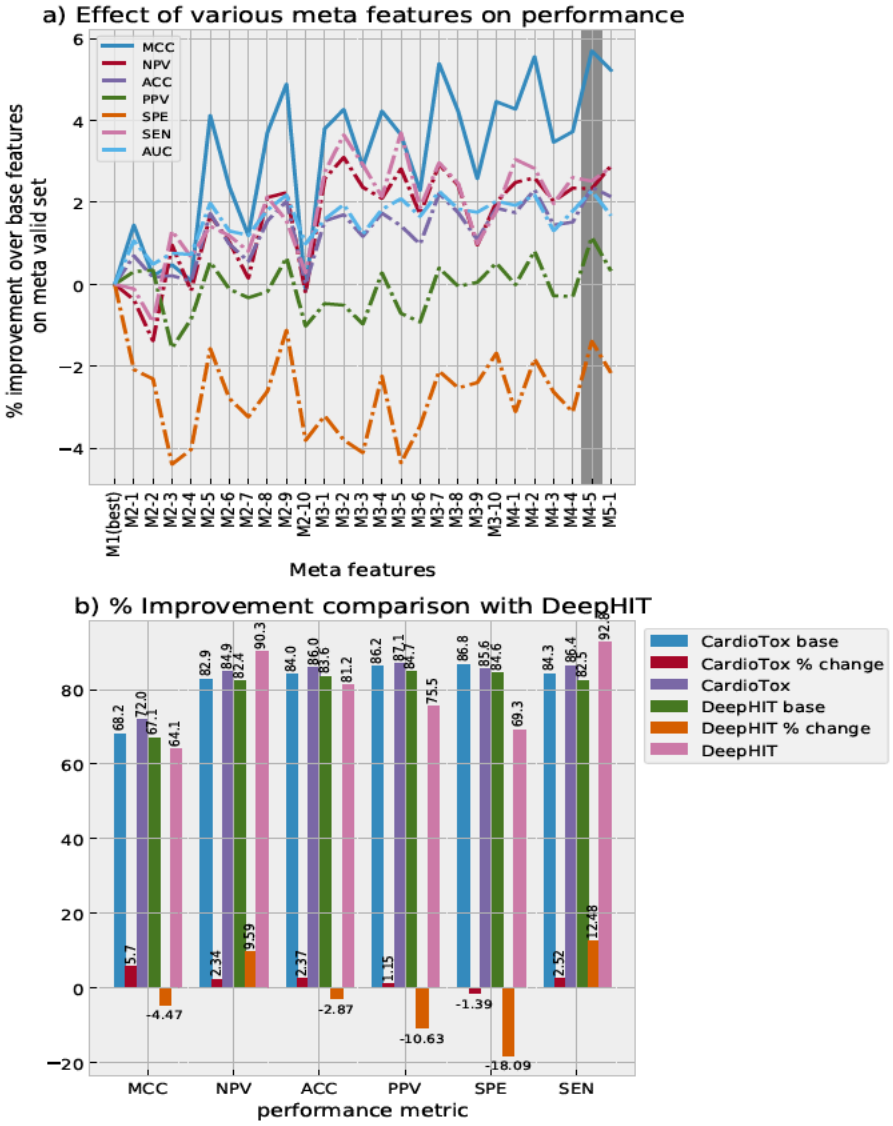

FIGURE 6.5: Figure a: shows the affect of various meta features in terms of % improvement over the base features using an ensemble stage of CardioTox framework on meta valid set. Figure b: shows the % difference of CardioTox and DeepHIT from their respective best base models performance for various performances metrics.

0.833 for test set-I and 0.909 for test set-II which is the same as achieved by DeepHIT. For ACC and NPV, the improvement over DeepHIT for test-sets I,II is (4.78%, 4.71%), and (6.99%,0.64%). OCHM-Predictor I, II achieves better performance for PPV and SPE but lags behind significantly in all other metrics for both test sets. Pred-hERG 4.2 performs reasonably well for SEN in both tests but performs worse in other metrics. Interestingly for test-set II, OCHEM-Predictor I and II performs reasonably well for PPV and SPE with less sacrifice in other metrics as compared to its performance on test set-I. DeepHIT is specifically designed and trained to obtain better NPV and SEN by using physicochemical descriptors, fingerprints and graph features with three deep learning base models. CardPred used an individual neural network model (out of six other models) with physicochemical descriptors and fingerprints. OCHMI and OCHMII used range of machine learning models trained on various types of high level physicochemical descriptors. Pred-hERG 4.2 used fingerprints and molecular descriptors with support vector machines to classify the molecules for hERG blocking activity. By using a step-wise training strategy with base and meta ensemble models, CardioTox shows robust performance against a range of accuracy metrics as compared to the state of the art methods on two independent test sets.

TABLE 6.6: Comparison of CardioTox with other methods using two external independent test sets

Evaluation data	Methods	MCC	NPV	ACC	PPV	SPE	SEN
Test set-I	CardioTox	**0.599**	**0.688**	**0.810**	0.893	0.786	**0.833**
	DeepHIT	0.476	0.643	0.773	0.833	0.643	0.833
	CardPred	0.193	0.421	0.614	0.760	0.571	0.633
	OCHEM Predictor-I	0.149	0.333	0.364	**1.000**	**1.000**	0.067
	OCHEM Predictor-II	0.164	0.351	0.432	0.857	0.929	0.200
	Pred-hERG 4.2	0.306	0.538	0.705	0.774	0.500	0.800
Test set-II	CardioTox	**0.452**	**0.947**	**0.755**	0.455	0.600	**0.909**
	DeepHIT	0.398	0.941	0.721	0.417	0.533	0.909
	CardPred	0.049	0.750	0.527	0.294	0.600	0.454
	OCHEM Predictor-I	0.372	0.800	0.648	**0.666**	**0.933**	0.364
	OCHEM Predictor-II	0.310	0.794	0.632	0.571	0.900	0.364
	Pred-hERG 4.2	0.146	0.813	0.580	0.320	0.433	0.727

6.4 Conclusion

In this chapter, we introduced a deep learning based framework called CardioTox for classifying drug-like molecules as hERG blockers and hERG non blockers . Our approach is based on step-wise training of base and meta ensemble deep learning models. In the first step, 5 deep learning base models are trained and validated. Each of these base models use different types of base features ranging from high level to low level descriptors and their variants. In the second step of training, the output of base models is concatenated to form meta features for training and validating the meta ensemble model. We found that high level physicochemical, low level fingerprints, SMILES embedding vectors and fingerprint embedding vectors when used to create meta features for the meta ensemble model, enhance the performance over a wide range of metrics for the cardio toxicity prediction task. We evaluated our framework against various classification metrics using two oppositely biased independent test sets and obtained a robust performance compared to state of the art methods. Our framework is a robust method for classifying small drug-like molecules as hERG blockers and hERG non blockers. The software code along with data for this chapter can be found on `https://github.com/Abdulk084/CardioTox`.

Chapter 7

Conclusions

This chapter summarises the contributions of the research presented in this thesis. It also covers the objectives and aims set at the beginning of this thesis. We also outline a few possible potential future directions of this research and conclude the thesis.

7.1 Toxicity predictions via deep learning

In pharmaceutical industries, molecular toxicity prediction plays a crucial role in the process of drug design and development. Traditionally, quantitative activity relationships methods such as decision trees, support vector machines and random forests are used to screen molecules for their toxic properties before performing experiments. In QSAR methods, recently deep learning techniques have widely been used because of their ability to use non linear interactions of features to predict toxicity of molecules. In this thesis, we explore molecular toxicity prediction using deep learning techniques and report state of the art performances. In first quarter of the thesis, we explored joint optimization of decision tees and shallow neural networks for quantitative and qualitative toxicity prediction tasks. This enabled us to interpret the model prediction in the context of features importance values while still achieving state of the art performance. In the later quarters of this thesis, we proposed the idea of effectively aggregating various types of molecular

features and their predictions using ensemble approaches. We used single task and multi-task learning methods for quantitative and qualitative toxicity predictions. Most existing machine learning methods in toxicity prediction utilise only one type of feature representation and one type of neural network; which essentially restricts their performance. Our motivation was to effectively aggregate the information extracted from various molecular features to help in boasting the over all performance across a number of accuracy metrics. These contributions are published/under revision/under review [68, 70, 71, 72].

7.2 Joint optimization of Decision trees and shallow neural networks

Despite better prediction accuracy, deep neural networks are compute intensive and black box in nature. On the other hand, classical machine learning methods such as decision trees legs behind accuracy but wins with features interpretability. We proposed the idea of joint optimization of decision trees and shallow neural networks for various toxicity prediction tasks. We conclude that our hybrid model of a DT and an SNN can be used for toxicity prediction or any similar tasks to achieve better or near similar accuracy in comparably lesser time and lesser resources. This technique enabled us to use certain features for rapid and prior toxicity estimation.

- It used very simple 2D physicochemical features which are easy to be computed as compared to other complex 3D features.
- Decision trees with gini index are used to select effective number of features for the shallow neural network.

- We explored the idea using decisions trees as a coarse filter to feed the shallow neural network with important features only.
- Individual rankings of these features were used to calculate average ranking of each feature.
- The algorithm is tested for independent test data sets of three classification and four regression toxicity tasks.
- Our method achieves state of the art performance in various classification and regression toxicity tasks.
- The computational complexity of various toxicity end points can be reduced to a great extent with our hybrid algorithm while keeping the accuracy level similar or relatively better than the state of the art methods.
- Using our commutative feature ranking method, we help the chemists effectively screen out the toxic compounds with few features in hand.

7.3 Features specific performance restriction and meta ensemble approaches

Molecular data can be expressed in various representations/features. Each type of feature has its own pros and cons with respect to molecular toxicity prediction. Usually, single type of features are used with a fixed prediction model which restricts the performance. In this thesis, we proposed new set of approaches involving meta ensembling of base model predictions to boost the overall prediction accuracy of various toxicity tasks. Meta ensembling also helped in obtaining robust models for a range of accuracy metrics for cardiotoxicity data sets.

- We proposed a method which uses various heterogeneous predictors ensembling to achieve better performance in quantitative toxicity prediction of four benchmark data sets. Thus, eliminating the restriction of model and data representation bound approach, each of our model's predictor vary either on features level, deep learning architecture level or both. Our motivation was to make a single model that utilizes different types of feature and architecture to obtain collective performance that could go beyond the individual performance of a single predictor type. We also performed experiments which showed that the heterogeneous ensembling method performs better than ensembling the homogeneous predictors.

- We introduced a novel two step training framework for toxicity predictions. In the first step of training, base individual models are trained to produce meta features. In the second step of training, a separate fully connected neural network is trained on the meta features produced by the base models to predict the final level or class of toxicity. In each step of training, a separate validation set was used for optimization. We applied our proposed framework on for quantitative toxicity end points such as LD_{50}, IGC_{50}, LC_{50} and LC_{50}-DM. We also proposed a novel method of multi-task optimization for both base and meta ensembling model. In order to test the robustness our proposed framework, we predicted the hERG channel blockade property of molecules and achieved the state of the art performance over a range of classification metrics. Our framework is a robust method for classifying small drug-like molecules as hERG blockers and hERG non blockers.

7.4 Future directions

In this thesis, each main chapter deals with a different kind of molecular toxicity problem and specific respective approaches. Thus, this research leads to a number of possible future directions to solve these problems more efficiently.

- In this thesis, we developed an algorithm based on joint optimization of decisions trees and shallow neural networks for toxicity prediction. The important features are selected by gini index of decisions trees which are then used in training process of shallow neural network. These features are effective in toxicity prediction but are not well optimized. In future, heuristics methods instead of gini index can be used to select optimized number of features to train shallow neural network. Moreover, this algorithm can be extended to activities other than toxicity predictions such as absorption, distribution, metabolism and excretion. It would be interesting to explore the model agnostic interpretability methods to obtain the feature importance values as well.

- In case of heterogeneous predictors, we used simple averaging technique to obtain the final output. In future work, It will be interesting to use max voting ensemble instead of simple averaging of the outputs. For regression tasks, molecular chemistry inspired threshold can be defined for maximum voting strategy.

- For meta ensemble approaches using in this thesis, we used two steps training strategy. In the first step, base models are trained. In the second step, the trained base models are used as it is while training the meta ensemble model. In future, it would be very interesting to train both base and meta models in an end to end fashion by optimizing both in one go. Moreover, various ways of multi-tasking complimented with

meta ensemble learning can be explored to find out the most optimum method for any specific toxicity prediction task.

In the future, we plan to continue our investigation to improve the effectiveness and prediction performances along with the interpretability of deep learning in molecular toxicity predictions.

Bibliography

[1] Major architectures of deep networks - deep learning [book]. https://www.oreilly.com/library/view/deep-learning/9781491924570/ch04.html. (Accessed on 01/24/2021).

[2] Convolutional neural networks (cnn). https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-summary/. (Accessed on 01/24/2021).

[3] How does linear and logistic regression work in machine learning? https://www.analyticssteps.com/blogs/how-does-linear-and-logistic-regression-work-machine-learning. (Accessed on 01/23/2021).

[4] Top 10 deep learning algorithms you should know in (2020). https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm. (Accessed on 01/24/2021).

[5] Understanding rnn and lstm. https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e. (Accessed on 01/24/2021).

[6] Urushiol, the poison in poison ivy , american council on science and health. https://www.acsh.org/news/2018/06/12/urushiol-poison-poison-ivy-13042. (Accessed on 01/23/2021).

[7] Ahmed Abdelaziz, Hilde Spahn-Langguth, Karl-Werner Schramm, and Igor V Tetko. Consensus modeling for hts assays using in silico descriptors calculates the best balanced accuracy in tox21 challenge. *Frontiers in Environmental Science*, 4:2, 2016.

[8] Nir Ailon, Ragesh Jaiswal, and Claire Monteleoni. Streaming k-means approximation. In *Advances in neural information processing systems*, pages 10–18, 2009.

[9] Subhash Ajmani, Kamalakar Jadhav, and Sudhir A Kulkarni. Three-dimensional qsar using the k-nearest neighbor method and its interpretation. *Journal of chemical information and modeling*, 46(1):24–31, 2006.

[10] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, volume 2, page 7, 2002.

[11] Alex M Aronov. Common pharmacophores for uncharged human ether-a-go-go-related gene (herg) blockers. *Journal of medicinal chemistry*, 49(23):6917–6921, 2006.

[12] Gergo Barta. Identifying biological pathway interrupting toxins using multi-tree ensembles. *Frontiers in Environmental Science*, 4:52, 2016.

[13] Andreas Bender, Hamse Y Mussa, Robert C Glen, and Stephan Reiling. Molecular similarity searching using atom environments, information-based feature selection, and a naive bayesian classifier. *Journal of chemical information and computer sciences*, 44(1):170–178, 2004.

[14] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[15] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb): 281–305, 2012.

[16] Rodolpho C Braga, Vinicius M Alves, Meryck FB Silva, Eugene Muratov, Denis Fourches, Luciano M Lião, Alexander Tropsha, and Carolina H Andrade. Pred-herg: A novel web-accessible computational tool for predicting cardiac toxicity. *Molecular informatics*, 34(10):698–701, 2015.

[17] Chuipu Cai, Qihui Wu, Yunxia Luo, Huili Ma, Jiangang Shen, Yongbin Zhang, Lei Yang, Yunbo Chen, Zehuai Wen, and Qi Wang. In silico prediction of rock ii inhibitors by different classification approaches. *Molecular diversity*, 21(4):791–807, 2017.

[18] Chuipu Cai, Pengfei Guo, Yadi Zhou, Jingwei Zhou, Qi Wang, Fengxue Zhang, Jiansong Fang, and Feixiong Cheng. Deep learning-based prediction of drug-induced cardiotoxicity. *Journal of chemical information and modeling*, 59(3):1073–1084, 2019.

[19] Stephen J Capuzzi, Regina Politi, Olexandr Isayev, Sherif Farag, and Alexander Tropsha. Qsar modeling of tox21 challenge stress response and nuclear receptor signaling toxicity assays. *Frontiers in Environmental Science*, 4:3, 2016.

[20] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[21] Andrea Cavalli, Elisabetta Poluzzi, Fabrizio De Ponti, and Maurizio Recanatini. Toward a pharmacophore for drugs inducing the long qt syndrome: insights from a comfa study of herg k+ channel blockers. *Journal of medicinal chemistry*, 45(18):3844–3853, 2002.

[22] Balakumar Chandrasekaran, Sara Nidal Abed, Omar Al-Attraqchi, Kaushik Kuche, and Rakesh K Tekade. Computer-aided prediction of pharmacokinetic (admet) properties. In *Dosage Form Design Parameters*, pages 731–755. Elsevier, 2018.

[23] Swapnil Chavan, Ran Friedman, and Ian A Nicholls. Acute toxicity-supported chronic toxicity prediction: a k-nearest neighbor coupled read-across strategy. *International journal of molecular sciences*, 16(5): 11659–11677, 2015.

[24] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[25] François Chollet et al. Keras. https://keras.io, 2015.

[26] Balázs Csanád Csáji. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48, 2001.

[27] Rodger D Curren and John W Harbell. In vitro alternatives for ocular irritation. *Environmental Health Perspectives*, 106(Suppl 2):485, 1998.

[28] Adele Cutler, D Richard Cutler, and John R Stevens. Random forests. In *Ensemble machine learning*, pages 157–175. Springer, 2012.

[29] George E Dahl, Navdeep Jaitly, and Ruslan Salakhutdinov. Multi-task neural networks for qsar predictions. *arXiv preprint arXiv:1406.1231*, 2014.

[30] Borje Darpo, Thierry Nebout, and Philip T Sager. Clinical evaluation of qt/qtc prolongation and proarrhythmic potential for nonantiarrhythmic drugs: the international conference on harmonization of technical requirements for registration of pharmaceuticals for human use e14 guideline. *The Journal of Clinical Pharmacology*, 46(5):498–507, 2006.

[31] Yann N Dauphin and Yoshua Bengio. Big neural networks waste capacity. *arXiv preprint arXiv:1301.3583*, 2013.

[32] Cheng-Hao Deng and Wan-Lei Zhao. Fast k-means based on knn graph. *arXiv preprint arXiv:1705.01813*, 2017.

[33] Li Deng and Dong Yu. Deep learning: methods and applications. *Foundations and trends in signal processing*, 7(3–4):197–387, 2014.

[34] Remigijus Didziapetris and Kiril Lanevskij. Compilation and physicochemical classification analysis of a diverse herg inhibition database. *Journal of computer-aided molecular design*, 30(12):1175–1188, 2016.

[35] Munikumar R Doddareddy, Elisabeth C Klaasse, Adriaan P IJzerman, and Andreas Bender. Prospective validation of a comprehensive in silico herg model and its applications to commercial compound and drug databases. *ChemMedChem*, 5(5):716–729, 2010.

[36] Jie Dong, Zhi-Jiang Yao, Lin Zhang, Feijun Luo, Qinlu Lin, Ai-Ping Lu, Alex F Chen, and Dong-Sheng Cao. Pybiomed: a python library for various molecular representations of chemicals, proteins and dnas and their interactions. *Journal of cheminformatics*, 10(1):16, 2018.

[37] Long Duong, Trevor Cohn, Steven Bird, and Paul Cook. Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser. In *Proceedings of the 53rd annual meeting of the Association*

for Computational Linguistics and the 7th international joint conference on natural language processing (volume 2: short papers), pages 845–850, 2015.

[38] David A Eastmond, Andrea Hartwig, Diana Anderson, Wagida A Anwar, Michael C Cimino, Ivan Dobrev, George R Douglas, Takehiko Nohmi, David H Phillips, and Carolyn Vickers. Mutagenicity testing for chemical risk assessment: update of the who/ipcs harmonized scheme. *Mutagenesis*, 24(4):341–349, 2009.

[39] Sean Ekins, William J Crumb, R Dustan Sarazan, James H Wikel, and Steven A Wrighton. Three-dimensional quantitative structure-activity relationship for inhibition of human ether-a-go-go-related gene potassium channel. *Journal of Pharmacology and Experimental Therapeutics*, 301 (2):427–434, 2002.

[40] Makoto Ema, Yoshihiro Fukui, Hiroaki Aoyama, Michio Fujiwara, Junichiro Fuji, Minoru Inouye, Takayuki Iwase, Takahide Kihara, Akihide Oi, Hiroki Otani, et al. Comments from the developmental neurotoxicology committee of the japanese teratology society on the oecd guideline for the testing of chemicals, proposal for a new guideline 426, developmental neurotoxicity study, draft document (october 2006 version), and on the draft document of the retrospective performance assessment of the draft test guideline 426 on developmental neurotoxicity. *Congenital anomalies*, 47(2):74–76, 2007.

[41] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. Chembl: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.

[42] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[43] Saeid Gholami, Fatemeh Soleimani, Farshad Hoseini Shirazi, Maryam Touhidpour, and Massoud Mahmoudian. Evaluation of mutagenicity of mebudipine, a new calcium channel blocker. *Iranian journal of pharmaceutical research: IJPR*, 9(1):49, 2010.

[44] Michael K Gilson, Tiqing Liu, Michael Baitaluk, George Nicola, Linda Hwang, and Jenny Chong. Bindingdb in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic acids research*, 44(D1):D1045–D1053, 2016.

[45] Garrett B Goh, Nathan O Hodas, Charles Siegel, and Abhinav Vishnu. Smiles2vec: An interpretable general-purpose deep neural network for predicting chemical properties. *arXiv preprint arXiv:1712.02034*, 2017.

[46] Garrett B Goh, Charles Siegel, Abhinav Vishnu, Nathan O Hodas, and Nathan Baker. Chemception: A deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models. *arXiv preprint arXiv:1706.06689*, 2017.

[47] Garrett B Goh, Charles Siegel, Abhinav Vishnu, Nathan O Hodas, and Nathan Baker. Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models. *arXiv preprint arXiv:1706.06689*, 2017.

[48] Garrett B Goh, Nathan Hodas, Charles Siegel, and Abhinav Vishnu. Smiles2vec: Predicting chemical properties from text representations. 2018.

[49] Garrett B Goh, Nathan Hodas, Charles Siegel, and Abhinav Vishnu. Smiles2vec: Predicting chemical properties from text representations. 2018.

[50] Garrett B Goh, Charles Siegel, Abhinav Vishnu, Nathan Hodas, and Nathan Baker. How much chemistry does a deep neural network need to know to make accurate predictions? In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1340–1349. IEEE, 2018.

[51] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[52] Daniele Grattarola and Cesare Alippi. Graph neural networks in tensorflow and keras with spektral. *arXiv preprint arXiv:2006.12138*, 2020.

[53] Lianyi Han, Yanli Wang, and Stephen H Bryant. Developing and validating predictive decision tree models from mining chemical structural fingerprints and high–throughput screening data in pubchem. *BMC bioinformatics*, 9(1):401, 2008.

[54] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pages 1135–1143, 2015.

[55] Katja Hansen, Sebastian Mika, Timon Schroeter, Andreas Sutter, Antonius Ter Laak, Thomas Steger-Hartmann, Nikolaus Heinrich, and Klaus-Robert Muller. Benchmark data set for in silico prediction of ames mutagenicity. *Journal of chemical information and modeling*, 49(9): 2077–2081, 2009.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[57] Mark O Hill. Introduction to the exploration of multivariate biological data, 2002.

[58] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[59] Ned Horning. Introduction to decision trees and random forests. *American Museum of Natural History*, 2:1–27, 2013.

[60] Ruili Huang, Srilatha Sakamuru, Matt T Martin, David M Reif, Richard S Judson, Keith A Houck, Warren Casey, Jui-Hua Hsieh, Keith R Shockley, Patricia Ceger, et al. Profiling of the 10k compound library for agonists and antagonists of the estrogen receptor alpha signaling pathway. *Scientific reports*, 4:5664, 2014.

[61] Ruili Huang, Menghang Xia, Dac-Trung Nguyen, Tongan Zhao, Srilatha Sakamuru, Jinghua Zhao, Sampada A Shahane, Anna Rossoshek, and Anton Simeonov. 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, 2016.

[62] Tyler B Hughes, Grover P Miller, and S Joshua Swamidass. Modeling epoxidation of drug-like molecules with a deep machine learning network. *ACS central science*, 1(4):168–180, 2015.

[63] Tyler B Hughes, Grover P Miller, and S Joshua Swamidass. Site of reactivity models predict molecular reactivity of diverse chemicals with glutathione. *Chemical research in toxicology*, 28(4):797–809, 2015.

[64] Tyler B Hughes, Na Le Dang, Grover P Miller, and S Joshua Swamidass. Modeling reactivity to biological macromolecules with a deep multitask network. *ACS central science*, 2(8):529–537, 2016.

[65] Kevin Jarrett, Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. What is the best multi-stage architecture for object recognition? In *2009 IEEE 12th international conference on computer vision*, pages 2146–2153. IEEE, 2009.

[66] Woosung Jeon and Dongsup Kim. Fp2vec: a new molecular featurizer for learning molecular properties. *Bioinformatics*, 2019.

[67] Abdul Karim, Avinash Mishra, MA Newton, and Abdul Sattar. Machine learning interpretability: A science rather than a tool. *arXiv preprint arXiv:1807.06722*, 2018.

[68] Abdul Karim, Avinash Mishra, MA Hakim Newton, and Abdul Sattar. Efficient toxicity prediction via simple features using shallow neural networks and decision trees. *ACS Omega*, 4(1):1874–1888, 2019.

[69] Abdul Karim, Vahid Riahi, Avinash Mishra, Abdollah Dehzangi, MA Hakim Newton, and Abdul Sattar. Quantitative toxicity prediction via ensembling of heterogeneous predictors. 2019.

[70] Abdul Karim, Jaspreet Singh, Avinash Mishra, Abdollah Dehzangi, MA Hakim Newton, and Abdul Sattar. Toxicity prediction by multimodal deep learning. In *Pacific Rim Knowledge Acquisition Workshop*, pages 142–152. Springer, 2019.

[71] Abdul Karim, Matthew Lee, Thomas Balle, and Abdul Sattar. Cardiotox net: A robust predictor for herg channel blockade via deep learning meta ensembling approaches. *BMC Cheminformatics*, 2021. Under revision.

[72] Abdul Karim, Vahid Riahi, Avinash Mishra, Hakim Newton, Abdullah Dehzangi, Thomas Balle, and Abdul Sattar. Quantitative toxicity prediction via meta ensembling of multi-task deep learning models. *ACS Omega*, 2021. Under review.

[73] Yoshiki Kato, Shinji Hamada, and Hitoshi Goto. Molecular activity prediction using deep learning software library. In *Advanced Informatics: Concepts, Theory And Application (ICAICTA), 2016 International Conference On*, pages 1–6. IEEE, 2016.

[74] Yoshiki Kato, Shinji Hamada, and Hitoshi Goto. Molecular activity prediction using deep learning software library. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, pages 1–6. IEEE, 2016.

[75] Gregory W Kauffman and Peter C Jurs. Qsar and k-nearest neighbor classification analysis of selective cyclooxygenase-2 inhibitors using topologically-based numerical descriptors. *Journal of chemical information and computer sciences*, 41(6):1553–1560, 2001.

[76] Steven Kearnes, Brian Goldman, and Vijay Pande. Modeling industrial admet data with multitask networks. *arXiv preprint arXiv:1606.08793*, 2016.

[77] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[78] Karen L Kimm-Brinson and John S Ramsdell. The red tide toxin, brevetoxin, induces embryo toxicity and developmental abnormalities. *Environmental Health Perspectives*, 109(4):377, 2001.

[79] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[80] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[81] Leela Sarath Kumar Konda, S Keerthi Praba, and Rajendra Kristam. herg liability classification models using machine learning techniques. *Computational Toxicology*, 12:100089, 2019.

[82] Sunyoung Kwon, Ho Bae, Jeonghee Jo, and Sungroh Yoon. Comprehensive ensemble in qsar prediction for drug discovery. *BMC bioinformatics*, 20(1):521, 2019.

[83] Greg Landrum. Rdkit: Open-source cheminformatics. URL http://www.rdkit.org.

[84] Greg Landrum et al. Rdkit: Open-source cheminformatics. 2006.

[85] Hyang-Mi Lee, Myeong-Sang Yu, Sayada Reemsha Kazmi, Seong Yun Oh, Ki-Hyeong Rhee, Myung-Ae Bae, Byung Ho Lee, Dae-Seop Shin, Kwang-Seok Oh, Hyithaek Ceong, et al. Computational determination of herg-related cardiotoxicity of drug candidates. *BMC bioinformatics*, 20(10):250, 2019.

[86] Xiao Li, Yuan Zhang, Huanhuan Li, and Yong Zhao. Modeling of the herg k+ channel blockage using online chemical database and modeling environment (ochem). *Molecular Informatics*, 36(12):1700074, 2017.

[87] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

[88] Angélica Nakagawa Lima, Eric Allison Philot, Gustavo Henrique Goulart Trossini, Luis Paulo Barbour Scott, Vinícius Gonçalves

Maltarollo, and Kathia Maria Honorio. Use of machine learning approaches for novel drug discovery. *Expert opinion on drug discovery*, 11 (3):225–239, 2016.

[89] Angelica Nakagawa Lima, Eric Allison Philot, Gustavo Henrique Goulart Trossini, Luis Paulo Barbour Scott, Vinícius Goncalves Maltarollo, and Kathia Maria Honorio. Use of machine learning approaches for novel drug discovery. *Expert opinion on drug discovery*, 11 (3):225–239, 2016.

[90] Ke Liu, Xiangyan Sun, Lei Jia, Jun Ma, Haoming Xing, Junqiu Wu, Hua Gao, Yax Sun, Florian Boulnois, and Jie Fan. Chemi-net: a molecular graph convolutional network for accurate drug property prediction. *International journal of molecular sciences*, 20(14):3389, 2019.

[91] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. Understanding variable importances in forests of randomized trees. In *Advances in neural information processing systems*, pages 431–439, 2013.

[92] Alessandro Lusci, Gianluca Pollastri, and Pierre Baldi. Deep architectures and deep learning in chemoinformatics: the prediction of aqueous solubility for drug-like molecules. *Journal of chemical information and modeling*, 53(7):1563–1575, 2013.

[93] Junshui Ma, Robert P Sheridan, Andy Liaw, George E Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55(2):263–274, 2015.

[94] Junshui Ma, Robert P Sheridan, Andy Liaw, George E Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55(2):263–274, 2015.

[95] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[96] T Martin. User's guide for test (version 4.2)(toxicity estimation software tool) a program to estimate toxicity from molecular structure. us epa office of research and development, washington, dc. Technical report, EPA/600/R-16/058, 2016.

[97] Andreas Mayr, Günter Klambauer, Thomas Unterthiner, and Sepp Hochreiter. Deeptox: toxicity prediction using deep learning. *Frontiers in Environmental Science*, 3:80, 2016.

[98] James W McFarland. Parabolic relation between drug potency and hydrophobicity. *Journal of medicinal chemistry*, 13(6):1192–1196, 1970.

[99] Bjoern H Menze, B Michael Kelm, Ralf Masuch, Uwe Himmelreich, Peter Bachert, Wolfgang Petrich, and Fred A Hamprecht. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC bioinformatics*, 10(1):213, 2009.

[100] Hrushikesh Mhaskar, Qianli Liao, and Tomaso A Poggio. When and why are deep networks better than shallow ones? In *AAAI*, pages 2343–2349, 2017.

[101] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[102] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O Anatole Von Lilienfeld. Machine learning of

molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):095003, 2013.

[103] Hirotomo Moriwaki, Yu-Shi Tian, Norihito Kawashita, and Tatsuya Takagi. Mordred: a molecular descriptor calculator. *Journal of cheminformatics*, 10(1):4, 2018.

[104] Raymond H Myers and Raymond H Myers. *Classical and modern regression with applications*, volume 2. Duxbury press Belmont, CA, 1990.

[105] Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

[106] Carolina Dizioli Rodrigues Oliveira, Camila Queiroz Moreira, Lilian Rose Marques de Sá, Helenice de Souza Spinosa, and Mauricio Yonamine. Maternal and developmental toxicity of ayahuasca in wistar rats. *Birth Defects Research Part B: Developmental and Reproductive Toxicology*, 89(3):207–212, 2010.

[107] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[108] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[109] Pavel G Polishchuk, Eugene N Muratov, Anatoly G Artemenko, Oleg G Kolumbin, Nail N Muratov, and Victor E Kuz'min. Application of random forest approach to qsar prediction of aquatic toxicity. *Journal of chemical information and modeling*, 49(11):2481–2488, 2009.

[110] Ignacio Ponzoni, Víctor Sebastián-Pérez, Carlos Requena-Triguero, Carlos Roca, María J Martínez, Fiorella Cravero, Mónica F Díaz, Juan A Páez, Ramón Gómez Arrayás, Javier Adrio, et al. Hybridizing feature selection and feature learning approaches in qsar modeling for drug discovery. *Scientific reports*, 7(1):1–19, 2017.

[111] Birgit Priest, Ian M Bell, and Maria Garcia. Role of herg potassium channel assays in drug development. *Channels*, 2(2):87–93, 2008.

[112] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

[113] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837, 2019.

[114] WS Redfern, L Carlsson, AS Davis, WG Lynch, Il MacKenzie, S Palethorpe, PKS Siegl, I Strang, AT Sullivan, R Wallis, et al. Relationships between preclinical cardiac electrophysiology, clinical qt interval prolongation and torsade de pointes for a broad range of drugs: evidence for a provisional safety margin in drug development. *Cardiovascular research*, 58(1):32–45, 2003.

[115] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[116] Sereina Riniker and Gregory A Landrum. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *Journal of cheminformatics*, 5(1):26, 2013.

[117] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50(5):742–754, 2010.

[118] Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[119] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Neurocomputing: Foundations of research, 1988.

[120] Jae Yong Ryu, Mi Young Lee, Jeong Hyun Lee, Byung Ho Lee, and Kwang-Seok Oh. Deephit: a deep learning framework for prediction of herg-induced cardiotoxicity. *Bioinformatics*, 36(10):3049–3055, 2020.

[121] Seongok Ryu, Jaechang Lim, Seung Hwan Hong, and Woo Youn Kim. Deeply learning molecular structure-property relationships using attention-and gate-augmented graph convolutional network. *arXiv preprint arXiv:1805.10988*, 2018.

[122] Y Sakuratani, HQ Zhang, S Nishikawa, K Yamazaki, T Yamada, J Yamada, K Gerova, G Chankov, O Mekenyan, and M l Hayashi. Hazard evaluation support system (hess) for predicting repeated dose toxicity using toxicological categories. *SAR and QSAR in environmental research*, 24(5):351–363, 2013.

[123] Norberto Sánchez-Cruz and José L Medina-Franco. Statistical-based database fingerprint: chemical space dependent representation of compound databases. *Journal of cheminformatics*, 10(1):55, 2018.

[124] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[125] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[126] Kristof T Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature communications*, 8:13890, 2017.

[127] Vishal B Siramshetty, Qiaofeng Chen, Prashanth Devarakonda, and Robert Preissner. The catch-22 of predicting herg blockade using publicly accessible bioactivity data. *Journal of Chemical Information and Modeling*, 58(6):1224–1233, 2018.

[128] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

[129] Axel J Soto, Rocío L Cecchini, Gustavo E Vazquez, and Ignacio Ponzoni. Multi-objective feature selection in qsar using a machine learning approach. *QSAR & Combinatorial Science*, 28(11-12):1509–1523, 2009.

[130] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15 (1):1929–1958, 2014.

[131] Carolin Strobl, Torsten Hothorn, and Achim Zeileis. Party on! 2009.

[132] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6): 1947–1958, 2003.

[133] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar

modeling. *Journal of chemical information and computer sciences*, 43(6): 1947–1958, 2003.

[134] Vladimir Svetnik, Andy Liaw, Christopher Tong, J Christopher Culberson, Robert P Sheridan, and Bradley P Feuston. Random forest: a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6): 1947–1958, 2003.

[135] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[136] Roberto Todeschini and Viviana Consonni. *Handbook of molecular descriptors*, volume 11. John Wiley & Sons, 2008.

[137] Weida Tong, Huixiao Hong, Hong Fang, Qian Xie, and Roger Perkins. Decision forest: combining the predictions of multiple independent decision tree models. *Journal of Chemical Information and Computer Sciences*, 43(2):525–531, 2003.

[138] Gerard V Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on pattern analysis and machine intelligence*, (3):306–307, 1979.

[139] Thomas Unterthiner, Andreas Mayr, Günter Klambauer, and Sepp Hochreiter. Toxicity prediction using deep learning. *arXiv preprint arXiv:1503.01445*, 2015.

[140] Andrea Vattani. K-means requires exponentially many iterations even in the plane. *Discrete & Computational Geometry*, 45(4):596–616, 2011.

[141] Bruno O Villoutreix and Olivier Taboureau. Computational investigations of herg channel blockers: New insights and current predictive models. *Advanced drug delivery reviews*, 86:72–82, 2015.

[142] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

[143] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[144] David A Winkler and Tu C Le. Performance of deep and shallow neural networks, the universal approximation theorem, activity cliffs, and qsar. *Molecular informatics*, 36(1-2), 2017.

[145] Kedi Wu and Guo-Wei Wei. Quantitative toxicity prediction using topology based multitask deep neural networks. *Journal of chemical information and modeling*, 58(2):520–531, 2018.

[146] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[147] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[148] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[149] Xiaoyang Xia, Edward G Maliski, Paul Gallant, and David Rogers. Classification of kinase inhibitors using a bayesian model. *Journal of medicinal chemistry*, 47(18):4463–4470, 2004.

[150] Youjun Xu, Ziwei Dai, Fangjin Chen, Shuaishi Gao, Jianfeng Pei, and Luhua Lai. Deep learning for drug-induced liver injury. *Journal of chemical information and modeling*, 55(10):2085–2093, 2015.

[151] Hongbin Yang, Chaofeng Lou, Lixia Sun, Jie Li, Yingchun Cai, Zhuang Wang, Weihua Li, Guixia Liu, and Yun Tang. admetsar 2.0: web-service for prediction and optimization of chemical admet properties. *Bioinformatics*, 35(6):1067–1069, 2018.

[152] Chun Wei Yap. Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of computational chemistry*, 32(7):1466–1474, 2011.

[153] Chun Wei Yap. Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of computational chemistry*, 32(7):1466–1474, 2011.

[154] Michael York and Winfried Steiling. A critical review of the assessment of eye irritation potential using the draize rabbit eye test. In *Journal of Applied Toxicology: An International Forum Devoted to Research and Methods Emphasizing Direct Clinical, Industrial and Environmental Applications*, volume 18, pages 233–240. Wiley Online Library, 1998.

[155] Qingda Zang, Kamel Mansouri, Antony J Williams, Richard S Judson, David G Allen, Warren M Casey, and Nicole C Kleinstreuer. In silico

prediction of physicochemical properties of environmental chemicals using molecular fingerprints and machine learning. *Journal of chemical information and modeling*, 57(1):36–49, 2017.

[156] Herbert Zepnik, Wolfgang Völkel, and Wolfgang Dekant. Toxicokinetics of the mycotoxin ochratoxin a in f 344 rats after oral administration. *Toxicology and applied pharmacology*, 192(1):36–44, 2003.