Virtual Screening of Pharmaceutical Compounds with hERG Inhibitory Activity (Cardiotoxicity) using Ensemble Learning

Aditya Sarkar and Arnav Bhavsar

Indian Institute of Technology, Mandi, India

Keywords: Ensemble Learning, Feature Selection, Virtual Screening, Cardiotoxicity, Pharmaceuticals.

Abstract: In silico prediction of cardiotoxicity with high sensitivity and specificity for potential drug molecules can be of immense value. Hence, building machine learning classification models, based on some features extracted from the molecular structure of drugs, which are capable of efficiently predicting cardiotoxicity is critical. In this paper, we consider the application of various machine learning approaches, and then propose an ensemble classifier for the prediction of molecular activity on a Drug Discovery Hackathon (DDH) (1st reference) dataset. We have used only 2-D descriptors of SMILE notations for our prediction. Our ensemble classification uses 5 classifiers (2 Random Forest Classifiers, 2 Support Vector Machines and a Dense Neural Network) and uses Max-Voting technique and Weighted-Average technique for final decision.

1 INTRODUCTION

It is well known that drug discovery is complex, long-drawn, and requires interdisciplinary expertise to discover new molecules. Drug safety is an important issue in the process of drug discovery. Failure in Clinical trials in the 2000s was majorly due to efficacy and safety (approx 30%) (Kola, I. and Landis, J., 2004). One important aspect of drug safety is drug toxicity. Frequently observed toxicities are cardiotoxicity, hepatotoxicity, genotoxicity, and phototoxicity (Keiji Ogura, 2019). Toxicological screening is very important for the development of new drugs and for the extension of the therapeutic potential of existing molecules. The US Food and Drug Administration (FDA) states that it is essential to screen new molecules for pharmacological activity and toxicity potential in animals (21CFR Part 314). The toxic effects of chemicals, food substances, pharmaceuticals, etc., have attained great significance in the 21st century (Parasuraman, S, 2011).

The h-ERG (human Ether-à-go-go-Related Gene) is a gene that codes for a protein known as K_v11.1, the alpha subunit of a potassium ion channel ("hERG safety",2018). The h-ERG potassium channels (Snyders, 1999) are essential for normal electrical activity in the heart. When this channel’s conductivity of electric current is inhibited by some action of drugs, it can lead to a fatal disorder called Long QT Syndrome.

It is found that many drugs have the h-ERG inhibitory activity which can prolong the QT and thereby resulting in irregularity of the heartbeat called Torsades de Pointes (Keiji Ogura et al, 2019). As a result, many drugs, that are inhibiting the h-ERG channel’s conductivity, have been withdrawn from the markets. Hence, it is regarded as a major anti-target for drug discovery.

Over the years, many works have been done to achieve the goal of classifying compounds having h-ERG inhibitory activity. In early drug discovery stages such as screening h-ERG inhibitory activity, performing costly and time-consuming assays is difficult. Hence, developing an in-silico model to predict hERG inhibition is very useful. (Keiji Ogura et al, 2019) Machine Learning techniques can be used for classifying if any compound is having the inhibitory activity or not. Machine Learning model can learn features that can classify any compound on the basis of h-ERG inhibition activity.

There have been some recent classification models reported for h-ERG inhibition which have used Neural Networks (B. Mehlig, 2019), Random Forest Classifiers (RF) (Leo Breiman, 2001) and Support Vector Machines (SVM) (Yongli Zhang et al, 2012). Below we provide a brief review of these.

1.1 Related Work

Czodrowski (Czodrowski,,2013) constructed RF models using descriptors calculated by Rdkit , based...
Table 1: Details of the various datasets.

<table>
<thead>
<tr>
<th>Database</th>
<th>hERG inhibitors</th>
<th>Inactive compounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChEMBL (version 22)</td>
<td>4790</td>
<td>5275</td>
</tr>
<tr>
<td>GOSTAR</td>
<td>3260</td>
<td>3509</td>
</tr>
<tr>
<td>NCGC</td>
<td>232</td>
<td>1234</td>
</tr>
<tr>
<td>hERGCentral</td>
<td>4,321</td>
<td>274,536</td>
</tr>
<tr>
<td>hERG integrated dataset</td>
<td>9,890</td>
<td>281,329</td>
</tr>
</tbody>
</table>

on 3,721 compounds measured in a binding assay and 765 compounds measured in a functional assay collected from ChEMBL. The prediction models were constructed separately from each data set, and showed prediction accuracies of 79.7%–80.1% and 69.2%–90.7%, respectively.

Wang et al. (Wang, S. et al, 2012) developed hERG classification models using naive Bayesian classification and recursive partitioning based on molecular properties and the ECFP_8 fingerprints, and recorded 85% accuracy (Wang, S. et al, 2016).

Schyman et al. (Schyman, P, 2016) combined 3D (David C. Kombo et al,2012) similarity conformation and 2D similarity ensemble approach, and achieved 69% sensitivity and 95% specificity on an independent external data set.

Recently, Keiji Ogura, Tomohiro Sato, Hitomi Yuki and Teruki Honma (Keiji Ogura et al, 2019) used Support Vector Machines (SVMs) on an integrated h-ERG database having more than 291,000 structurally diverse compounds. They achieved kappa statistics of 0.733 and accuracy of 98.4%. Table 1 provides a summary of various datasets used across existing works. They have made the dataset publicly available for research purposes.

1.2 Contributions of this Work

Most of the works (Wang, S. et al, 2012),(Wang, S. et al, 2016),(Schyman, P, 2016),(Keiji Ogura et al, 2019) which we have reviewed have either used descriptors (2-D, 3-D and 4-D) or fingerprints. On the other hand, unlike the above, we have used only 2-D descriptors for our classification model. 2D-descriptors deal with the molecular topology of the compounds i.e. topological indices and fragment counts. 2D-Descriptors incorporate precious chemical information like size, degree of branching, flexibility etc. Generating 2D descriptors of the SMILES compounds usually takes less time than 3D descriptors. Even with just 2D descriptors, we demonstrate that the proposed ensemble model achieves a very good performance.

In this study, we have developed an ensemble model having two Random Forest Classifiers, two Support Vector Machines and one Dense Neural Network which achieved a AUC score (Area Under the ROC Curve) of 0.96 and Cohen’s Kappa of 0.9195.

Most of the existing models have used Support Vector Machines, Random Forest Classifiers and Naive Bayesian Classifiers for prediction. However, in addition to these models, we have also used Deep Neural Networks and two different Ensemble classifiers for the task. We have found that the Deep Neural Networks and the Ensemble classifier yield the highest performance.

We have also worked with data augmentation for our class-imbalanced dataset. We have used SMOTE (Synthetic Minority Oversampling Technique) (N. V. Chawla, et al, 2011) for augmenting data. Data augmentation is a very useful procedure because the data that it generates is almost similar to the training data. For some diseases, the drugs available can be quiet less, and doing prediction with less data points can lead to over-fitting. Data Augmentation can prove useful by not only creating new data but may also help in understanding the underlying distribution of each property (descriptors or fingerprints) of drugs.

Unlike most existing works, we also suggest an automatic approach based on information gain/entropy to shortlist (or select) features from a larger set. To our knowledge, the only exception among the existing methods, which are considered such a selection is work by (Keiji Ogura et al, 2019) which involves the NSGA-II (Non-dominated Sorting Genetic Algorithm-II) for descriptor selection.

Finally, towards the end of the paper, we provide a consolidated summary of the various works in this domain, the datasets used, the feature descriptors and methods employed, and their performance across several metrics. We note that although this work is not analysing imaging data, it involves core machine learning on an important problem in biology. Considering that this is a relatively recent application domain, such an overview provides a good perspective of the trade-offs of the approaches and paves the way for more standardized benchmarking and extensions in this area.

2 DATASET AND DESCRIPTORS

In this work, we have used a dataset provided in one of the competitions of the Drug Discovery Hackathon, organized by the Govt. of India; The dataset has been made by Dr Kunal Roy, professor from Department of Pharmaceutical Technology, Jadavpur University. http://sites.google.com/
site/kunalroyindia/home. It contains the SMILE notations of 8227 pharmaceutical compounds, along with their h-ERG inhibitory activity label (i.e. blocker or non-blocker). Out of these, 6878 compounds were blockers and 1349 were non-blockers.

We have used mordred (Moriwaki H, et al, 2018) Python module to decrypt the SMILE (Anderson, E. et al, 1987) notations to 2-D descriptors. As a result, we got 1613 features for each compound. Figure 1(a) shows a snippet of the Pandas view of the dataset. It shows the IDs of the pharmaceutical compounds, the class to which it belongs i.e. Blocker or Non-Blocker (Blocker means that the compound possesses the h-ERG inhibitory activity and Non-Blocker means that it does not possess the h-ERG inhibitory activity) and few 2D descriptors. The list of descriptors along with their descriptions can be found at mordred documentation link: https://mordred-descriptor.github.io/documentation/master/descriptors.html. Initially, we have extracted all the 2-D descriptors that mordred has offered. In section III-C, we have described a method for further selecting features out of these.

3 METHOD

We have first worked on data cleaning, then augmented our data because it was having an imbalanced class problem, selected important features using an Entropy/Information Gain based method, and finally performed the classification using two ensembles - Max-Voting Ensemble and Weighted-Average Ensemble using two variants of Random Forest Classifiers, two variants of Support Vector Machines and one Deep Neural Network. We have divided this section into 5 subsections - Data Cleaning, SMOTE Application, Feature Selection and Base Models and Ensemble-Learning.

3.1 Data Cleaning

Data Cleaning is important part of analysing this data. This is done so as to eliminate outliers present in the data. These outliers are due to miscalculations made by the mordred python module. Since the range each column is different, so to normalize them, Z-score/Standard Score is used.

Z-score or standard score of a particular column is defined as the number of standard deviations by which the value of a datapoint value is above or below the mean value of datapoints present in the column. Raw scores above the mean have positive Z-scores, while those below the mean have negative Z-scores (Spiegel, Murray R.; Stephens, Larry J, 2008). Mathematically, it is defined as:

$$Z = \frac{x - \mu}{\sigma}$$

where x is the sample datapoint, $\mu$ is mean of all the datapoints in the sample column and $\sigma$ is standard deviation in the sample column.

We have considered a datapoint, an outlier when...
A section of the dataset view obtained after cleaning is shown in Figure 1(b). It shows the class to which the pharmaceutical compound belongs to i.e. Blocker or Non-Blocker and few 2D descriptors. However, unlike Figure 1(a), this contains Z-score. Also, we have removed those features which were containing NaN values. Hence, our number of features decreased from 1613 to 1375.

### 3.2 SMOTE for Data Augmentation

As mentioned earlier, our dataset has 6878 Blocker compounds and 1349 non-Blocker compounds. Hence, it is imbalanced and can lead to high bias. To tackle this problem, we made use of the popular data augmentation method SMOTE (Synthetic Minority Oversampling Technique). SMOTE (N. V. Chawla, 2011) can be used to create synthetic examples for the minority class. It works by first choosing a random example from the minority class then $k$ of the nearest neighbors for that example are found. A randomly selected neighbor is interpolated between the two examples in feature space. We chose SMOTE considering its popularity, with which we were also getting an improvement in our results. However, one can also consider other data augmentation methods. We have used Imblearn Python Module (Guillaume Lemaître et al, 2017) for applying SMOTE. After SMOTE, we achieved a total size of 13756 datapoints, including 6878 Blocker compounds and 6878 Non-Blocker compounds.

### 3.3 Feature Selection

Feature selection is an important part of the model. Considering we have numerous 2-D features, it is important to consider the ones which can contribute for the task of the classification.

Gini-index and Entropy are used as criterion for calculating information gain. Decision tree algorithms use information gain to split a node. Entropy and gini are used for measuring impurity of a node. Node having multiple classes is considered impure and node having single class is considered pure. In this project, we have used Entropy as our impurity measure. While training a tree, we can compute how much each feature decreases the impurity. The more a feature decreases the impurity, the more important that feature is. In random forests, the impurity decrease from each feature can be averaged across trees to determine the final importance of the variable. For this we have used the features selected by Random Forests with measure of impurity as Entropy. Initially, we started with 1375 features for each compound. After applying feature selection, we had 592 features, which is a significant reduction.

### 3.4 Base Models and Ensemble Learning

We have used 5 base models in our ensemble - 2 Random Forest Classifiers, 2 Support Vector Machines and 1 Dense Neural Network. These well known methods are discussed briefly in the following subsections, and the corresponding parameters are provided in Table 2. We have used Scikit-Learn Python module for Random Forest Classifiers and Support Vector Machines. For implementing Dense Neural Networks, we have used Tensorflow 2 and Keras.

#### 3.4.1 Random Forest Classifiers

The random forest classifier is essentially an ensemble of decision tree-based classifiers. It operates by constructing a number of decision trees during its training and outputs a class decision that is the mode of the classes estimates of each decision tree. It is based on the principle of bagging to mitigate the bias-variance trade-off in decision tree-based classifiers.

#### 3.4.2 Support Vector Machines

SVM is a popular maximum-margin classification framework boasting advantages of good generalization and non-linear classification via the use of kernel functions. Support Vector machines find a maximum-margin-hyperplane that divides the data points of the classes such that the distance between the hyperplane and the nearest point of either class to this hyperplane is maximised.

#### 3.4.3 Dense Neural Networks

Dense Neural Networks have gained popularity as contemporary classifiers due to their ability to learn highly non-linear classification models, given enough data. A neural network is a network of neurons that can well approximates a highly nonlinear boundary between the classes, given enough data.

#### 3.4.4 Ensemble Learning

For our Ensemble, we have used Max-Voting and Weighted-Average with our 5 base models described above. In max-voting, each base model makes a prediction and votes for each sample. Only the sample class with the highest votes is included in the final predictive class. In weighted-average, we have
Table 2: Accuracy of various classification models.

<table>
<thead>
<tr>
<th>Base Model Accuracies and ROC-AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Models</strong></td>
</tr>
<tr>
<td>-----------------------------------</td>
</tr>
<tr>
<td>Dense Neural Networks</td>
</tr>
<tr>
<td>Random Forest Classifier-1</td>
</tr>
<tr>
<td>Random Forest Classifier-2</td>
</tr>
<tr>
<td>Support Vector Machines-1</td>
</tr>
<tr>
<td>Support Vector Machines-2</td>
</tr>
</tbody>
</table>

placed weights on predictions of each of the base model for the final prediction. The weights we have placed is 0.75 for DNN, 0.1 for RF-1, 0.07 for RF-2, 0.05 for SV-1 and 0.03 for SV-2. We have given more weights to that classifiers among the base models, which yields a higher accuracy. Since, our Neural network is showing maximum accuracy, we have placed maximum weightage to it.

4 EXPERIMENTS AND OUTCOMES

For our experiments, we have divided our dataset in the training-testing ratio of 70% and 30%. As a result, our training set contains 9629 datapoints and testing set has 4127 datapoints. The training data is split into training and validation sets, automatically by the in-built models in the packages that we employ. We use testing data only for prediction.

We have tested the performance of the base classifiers as well as the ensemble classifiers using several metrics, in addition to the overall accuracy. These are defined below:

4.1 Description of Metrics

The metrics we have used for classifying our final classification models are described below:

4.1.1 AUC-ROC Score

AUC-ROC score computes Area Under the Receiver Operating Characteristic Curve (ROC) from prediction scores. The ROC curve is shown in Figure 3. We note from all ROC curves that a high True Positive is achieved at fairly low value of False positives.
4.1.2 Sensitivity, Precision and Specificity

Abbreviating TN for True Negative, TP for True Positive, FN for False Negative and FP for False Positive, the other metrics are defined as:

\[
\text{Sensitivity} = \frac{TP}{TP+FN}, \quad \text{Specificity} = \frac{TN}{TN+FP}.
\]

Bal. Acc. = \(\frac{\text{Sensitivity} + \text{Specificity}}{2}\), Precision = \(\frac{TP}{TP+FP}\).

4.1.3 Cohens-Kappa (κ)

We have also used Cohens-Kappa (κ) (J. Cohen, 1960), which determines the level of agreement between two annotators in a classification problem.

\[
\kappa = \frac{p_o - p_e}{1 - p_e}
\]

where \(p_0\) is defined as empirical probability of agreement on the label assigned to any sample and \(p_e\) is defined as expected agreement when both annotators assign labels randomly.

4.1.4 Matthews Correlation Coefficient (MCC)

Matthews correlation coefficient (MCC) (Matthews, B. W., 1975) is generally regarded as a balanced measure which can be used in case of data imbalance between classes.

4.2 Results

The accuracy and the ROC-AUC values of the base models as well as the ensemble models are provided in Table 3. We note that among the base classifiers, the neural network model performs the best. The accuracy of the ensemble model with average weighting is similar to the DNN model. It is likely that the ensemble learning method with max voting is performing relatively lower, because of the some low performing base classifiers.

In Table 4, we provide the results of the top two performing classifiers from Table 3, for the other metrics that were defined above. We note that for both the approaches yield high quality classification across all the metrics, and their performance is close to each other. While for this dataset, the DNN model performs somewhat better than the ensemble learning approach, it is important to acknowledge the high performance of the ensemble learning, as for larger datasets, it is known that the ensemble strategies can typically better mitigate the bias-variance tradeoff.

4.3 A Review of the Results for Existing Methods

In Table 5, we provide a overall summary of the various existing methods for the task of classification of pharmaceutical compounds based on their hERG inhibitory activity. We note that while there have been a few (but not many) methods to address this task, these involve different datasets, and different features. Thus, while such a summary is not a comparison, it does provide, under one roof, a perspective on data, methods, and can help in identifying scope of improvements in this area.

In the table, some cells are blank as not all metrics are provided for all methods. We note that the methods do not yield a high performance across all the metrics. Some methods employ relatively small data. The work by Ogura et al, involves the largest data, but yields a low sensitivity and kappa coefficient. Such discrepancy suggests an issue with the data imbalance. Importantly, most of the approaches use various different features in their methodology. In contrast our approach uses only 2D features, and yields a high performance across all metrics. A limitation of this work is that it also involves relatively less data, which we plan to address in future.

5 CONCLUSIONS AND FUTURE WORK

In this work we have compared various standard machine learning methods for the task of pharmaceutical compound classification based on their hERG inhibi-
Table 5: A summary of the various approaches and their performance.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Database</th>
<th>Data size</th>
<th>Classifiers</th>
<th>Features</th>
<th>AUC</th>
<th>Sensitivity</th>
<th>Specificity</th>
<th>Cohen’s Kappa</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Czodrowski P., 2013</td>
<td>ChemBL</td>
<td>11958</td>
<td>RF</td>
<td>Pharmaco- phore hypothesis, Pipeline descriptors</td>
<td>0.564</td>
<td>0.029-0.243</td>
<td>-</td>
<td>-</td>
<td>0.907</td>
</tr>
<tr>
<td>Wang S et al, 2016</td>
<td>-</td>
<td>587</td>
<td>Naïve Bayes, SVM</td>
<td>2D 3D descriptors, ECFP-4 structural fingerprints, Pipeline descriptors</td>
<td>0.966</td>
<td>0.715</td>
<td>0.933</td>
<td>0.733</td>
<td>0.98</td>
</tr>
<tr>
<td>Schyman P., 2016</td>
<td>National Cancer Inst. Database</td>
<td>25000</td>
<td>-</td>
<td>Accelrys extended connectivity fingerprints, conformations</td>
<td>-</td>
<td>0.69</td>
<td>0.95</td>
<td>-</td>
<td>0.79</td>
</tr>
<tr>
<td>Doddareddy et al</td>
<td>Dube203, Literature368, Thai313 datasets</td>
<td>7360</td>
<td>LDA, SVM</td>
<td>Extended connectivity fingerprints, functional class fingerprints</td>
<td>0.94</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.91</td>
</tr>
<tr>
<td>Kwang-Eun et al</td>
<td>Pipeline (PP,PCFP_2, PCFP_4 and PCFP_6), R package</td>
<td>5252</td>
<td>DNN, NB, SVM, RF, Bagging</td>
<td>integer and binary type fingerprints</td>
<td>0.95</td>
<td>0.626</td>
<td>0.986</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Chaipu Cai et al</td>
<td>ChEMBL, hERG K+ channel binding affinity, radioligand binding measurements on mammalian and non-mammalian cell lines, literature-derived data</td>
<td>7889</td>
<td>DNN, GCNN</td>
<td>Molecular Operating Environment descriptors, Mol2vec descriptors</td>
<td>0.97</td>
<td>0.912</td>
<td>0.817</td>
<td>-</td>
<td>0.91</td>
</tr>
</tbody>
</table>

tion activity. Some of the important aspects that we have considered is the used of only 2D features, data augmentation, feature selection, and ensemble learning. The accuracies we have achieved with our model is quite high for a small dataset.

The work encourages us to further explore more ensemble strategies considering DNN features, stacking, bagging etc.

**REFERENCES**


B. Mehlig, 2019, Artificial Neural Networks. *arxiv* - 1901.05639


Leo Breiman, 2001. Random Forests, Statistics Department, University of California, Berkeley, CA 94720. *Springer*


