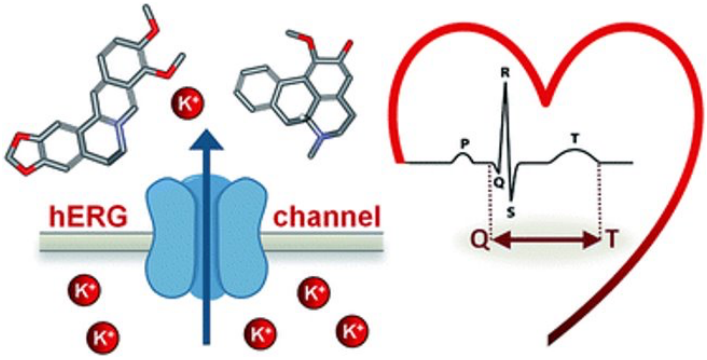
Building Machine Learning Models to Determine hERG Blockade Potential of Compounds

Jackson B. Sands,[a] H. Hoang,[a] S. Oner, [a] Y. Abo-Dahab [a]
[a] University of California San Francisco, 94143 San Francisco, USA
* Correspondence should be addressed to J.B.S. (Jackson.Sands@UCSF.edu)

In drug development the hERG blockade via small molecules can be catastrophic for drug approval. When drugs block the hERG channel it can prolong the QT interval. This can lead to many very dangerous side effects. In building our machine learning projects we used the Karim Toxicity dataset to classify if a compound would block the hERG channel or not. We evaluated several classifier models including logistic regression, random forest, gradient boosting, and support vector machines. We were able to achieve our best performance with our random forest achieving an accuracy of 0.85. While this is a solid performance overall it is likely that to be used in a clinical setting or for drug development, we may need to achieve better accuracy as we do not want to limit our chemical space unnecessarily or test many hERG blocking compounds. While 0.85 accuracy may be good for small projects when testing compounds at much higher magnitude that accuracy threshold may be less helpful. Our group believes that there is hope to increase the accuracy threshold both with increased computational resources and/or more sophisticated models.



Introduction

Blockade of the hERG or human ether-a-go-go-related potassium channel has become a very important topic in the drug development field as it has the potential to kill any drug development project. When a compound is able to block the hERG channel it can prolong the QT interval which represents an increase in the time it takes for the heart's ventricles to depolarize and repolarize. This can lead to very dangerous side effects such as cardiac arrhythmias which may lead to sudden cardiac arrest. These drug-induced arrhythmias have led to blockbuster drugs being withdrawn from the market and discontinuation of drugs in late stages of development. Therefore it is very useful to know when starting a drug development project on a specific compound if a compound has the potential to block the hERG channel. It can be catastrophic if a team runs into this issue when they have already spent significant resources to develop a drug. Classifying this issue was the primary goal of the machine learning project.

Methods

Datasets

In our model we used the Karim Toxicity dataset. This dataset was created for the project CardioTox net which used deep learning meta-feature ensembles to predict hERG blocking [1]. The dataset is made up of 13445 drugs in SMILES format labeled 1 for a hERG blocker and 0 for a non hERG blocker. Per the criteria of half maximum inhibitory concentration the (IC50) values < 10μM considered to be hERG blockers and (IC50) values >= 10μM considered to be hERG non-blockers. These values have been experimentally determined and collected from databases: BindingDB database [2], ChEMBL bioactivity database [3] and three literature derives[4][5][6]. The overall dataset was very balanced with 6718 class 1 samples and 6727 class 0 samples. While this balance is a bit bizarre it is very possible that this was done intentionally though not explicitly stated. We did not make any modifications to the datasets SMILES strings however we did check to make sure that all were correctly formatted and labeled, which they were. We split our dataset into 70% (9412 compounds) training set, 10% (1344 compounds) validation set, and 20% (2689 compounds) test set. The average compound length was 24 molecules.

Feature Engineering

Our data was presented in a SMILES string, this is a string of symbols and letters that is able to represent a molecule in a fairly simple way however it encapsulates a lot of information or a higher dimensional feature. To use this higher dimensional data we can convert our SMILES to Morgan Fingerprints using RDkit. Morgan fingerprints represent the structure of a molecule by encoding its circular atomic neighborhoods. The radius determines the size of the circular neighborhood to consider around each atom. Morgan fingerprints use multiple radiuses. For example, radius 1 includes the atom and its immediate neighbors, radius 2 extends to atoms two bonds away, and so on. This becomes a very high dimensional form of data,

and its only input is a SMILES string. From here we had to decide about the resolution of the Morgan Fingerprints. It is common to use 1024 or 2048 bits. This decision affects the resolution and ability to capture structural diversity of the molecules. We wanted to determine if it would be more useful to use a lower resolution for the Morgan fingerprint and incorporate more features or try to purely use the higher resolution data and leave other features out. We decided to use the higher resolution Morgan fingerprints because we thought the incremental increase for the resolution and capture of structural diversity would benefit us more than adding additional features such as molecules that fit to Lipinski's rules. This did however restrict us in our ability to run the models to some degree as we did not have additional compute power and were already dealing with high dimensional data.

Random Forest Classifier Performance:

	precision	recall	f1-score	support
0	0.85	0.84	0.85	1346
1	0.84	0.86	0.85	1343
accuracy			0.85	2689
macro avg	0.85	0.85	0.85	2689
weighted avg	0.85	0.85	0.85	2689

Accuracy: 0.8482707326143548

Gradient Boosting Classifier Performance:

	precision	recall	f1-score	support
0	0.68	0.73	0.71	1346
1	0.71	0.66	0.68	1343
accuracy			0.70	2689
macro avg	0.70	0.70	0.70	2689
weighted avg	0.70	0.70	0.70	2689

Accuracy: 0.6969133506879881

Gradient Boosting Classifier Performance Hypertuned:

	precision	recall	f1-score	support
0	0.85	0.83	0.84	1346
1	0.83	0.85	0.84	1343
accuracy			0.84	2689
macro avg	0.84	0.84	0.84	2689
weighted avg	0.84	0.84	0.84	2689

Accuracy: 0.8397173670509483

Table 1: Model Comparisons Performance comparison of three classification models Random Forest Classifier, Gradient Boosting Classifier (group parameters) (white), and Gradient Boosting Classifier (with my hyperparameter tuning) (black) on the same dataset. Each model's evaluation metrics include precision, recall, F1-score, and support for two classes (0 and 1), as well as overall accuracy, macro average, and weighted average scores.

Determining Model

To assess which model to use for our final project we looked at four different machine learning models including logistic regression, random forest, gradient boosting, and support vector machines. We attempted to optimize our hyperparameters slightly before running the models. We assessed attempting to use both GridSearch and then Random Search. We struggled to produce Grid Search optimized hyperparameters because of run time. We were able to produce hyperparameters for Random Search on some of our models however the parameters chosen indicated that we had done a solid job tuning the hyper parameters manually. After running each of our models we decided to continue with a random forest algorithm. We chose this for a few reasons. First of all, random forest models are great for classification (particularly with two classes). They can handle large high dimensional datasets. Random forest is also difficult to overfit and is an efficient algorithm which reduces runtimes. In addition to this we also had a solid initial performance and a good understanding of hyperparameter tuning for our random forest.

Here we also started to look at some of the shortcomings of the other models we initially tested. We determined that for the complexities of the dataset that we were using it was unlikely that our logistic regression would be able to sufficiently capture the data that we were inputting because of its reliance on linear features. We also thought it would be possible that the linear regression model would struggle with some degree of feature redundancy due to the high dimensional data and exhibit some degree of overfitting. Next, we looked at our support vector machines which we assumed would be very good for our classification problem. Support vector machines are effective on high dimensional data, handle non-linear relationships well and are robust to overfitting. However, they are computationally expensive and while our initial tests indicated good performance our support vector machines proved impossible to finetune the hyperparameters on the computational resources we had access to in this project. Finally, we looked at the gradient boosting model we assumed to see high predictive accuracy and good handling of non-linear features however were also worried about the computational complexities of the algorithm and without more computational resources it was difficult to do sufficient finetuning of hyperparameters to reduce the potential of overfitting. I decided to compare the performance of our random forest model with the gradient boosting model.

Results

Accuracy Building

Our group's random forest Classifier had a threshold of 0.85 accuracy, and I wanted to see if I could use the gradient boosting model to get a comparable accuracy. Our baseline gradient boosting model was able to get an accuracy performance of 0.70. I attempted to do a Grid Search and Random Search and was unable to get either to run to execution. I manually hypertunned my parameters until the run times began to exceed an hour. I

started by increasing the number of trees iteratively from 300 all the way up to 4000. After incorporating early stopping in my model I determined that 3000 trees was most appropriate to improve the accuracy without the chance of overfitting. With this I was at an accuracy of around 0.79 I then increased the max depth of the gradient boosting model which allows for a higher number of nodes within each tree. I increased the max depth from 3 all the way up to 8 and was able to improve the accuracy of the model to 0.84 (table 1). In addition to this precision, recall, and F1-score all proportionally increased. I would have likely increased the max depth more to improve the accuracy performance however at this point the run time was around 40 minutes.

Area Under the Receiver Operating Characteristic

The Area Under the Receiver Operating Characteristic is a useful metric in measuring overall performance when we have equal class distributions, and we care equally about false positives and false negatives. The overall AUC ROC was nearly identical with both the random forest and the gradient boosting performing at and AUC = 0.92 (fig. 1).

Confusion Matrix

Finally, we compared a Confusion matrix of our random forest and my gradient boosting model (fig. 2). Here we are able to see the random forest is still slightly better both in terms of true positives and negatives and minimizing false positives and negatives but they are relatively close. It is possible that with increased compute power we would be able to exceed the accuracy of our random forest model with the gradient boosting model however with the current compute power we have access to this would be challenging due to long run times.

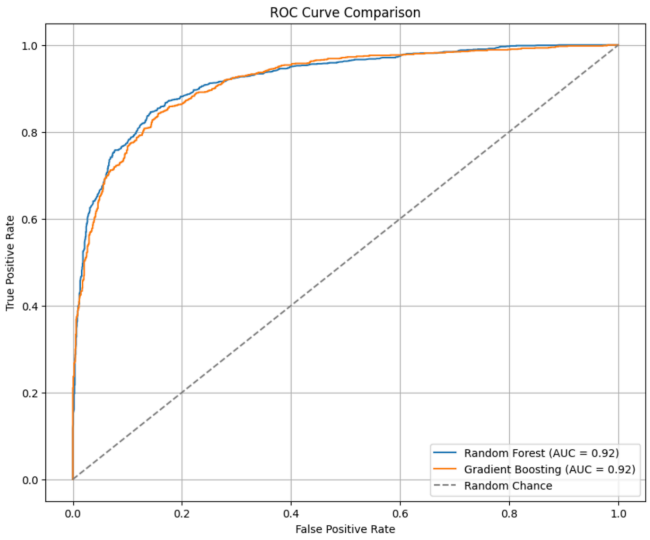


Figure 1: AUC ROC Curve Comparisons Receiver Operating Characteristic (ROC) curve comparison for the Random Forest and Gradient Boosting classifiers. The ROC curve plots the True Positive Rate (sensitivity) against the False Positive Rate at various classification thresholds, providing a visual representation of the models' discriminative ability. Random forest (blue) and gradient boosting (orange).

Discussion

Compute Power

In this project I was able to build a model that is of comparable accuracy to the model that we worked on in our group however the run times were drastically different. The random forest model took less than 3 minutes while the gradient boosting model took around 40 minutes. It is likely that the gradient boosting model and support vector machine have the potential to exceed the accuracy of the random forest because of their increased complexities in handling high dimensional data. However, to do this it is likely that we would want access to more compute power. This could allow us to run libraries like Grid Search and Random Search to better fine tune hyperparameters.

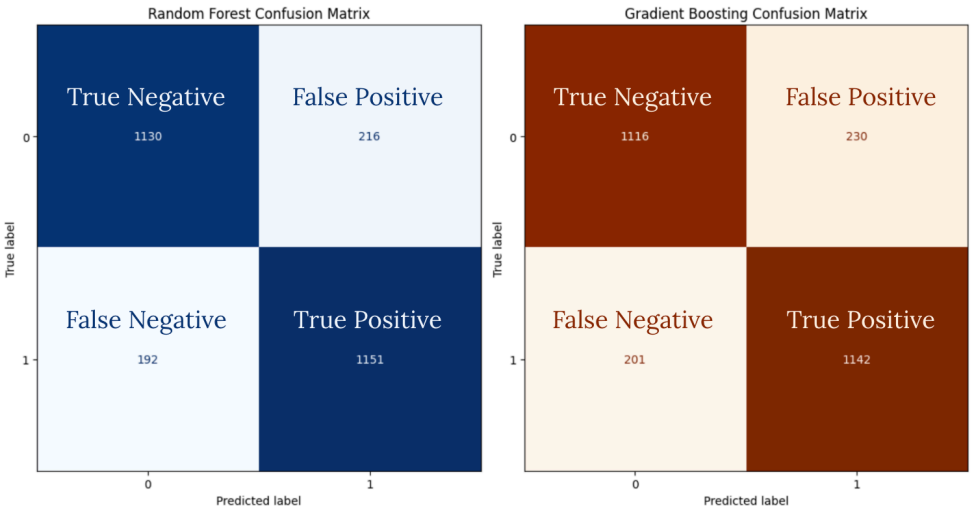


Figure 2: Confusion Matrix Comparisons Confusion matrix comparison for the Random Forest and Gradient Boosting classifiers, illustrating the distribution of predictions for two classes (0 and 1) against the true labels. Random forest (blue) and gradient boosting (orange).

Model Definition

Another area we could examine further is a better definition of what we are trying to use our model for. If we were using it for determining potential drug development candidates, we may value false negatives (true blockers that come up as non-blockers) as more of a problem then false positives (true non blockers that come up as blockers). This is because it is more costly to start developing a drug that we later find out is a hERG blocker then it is to remove a drug that may not be a hERG blocker (assuming infinite chemical space). In this case we may want to optimize our hyperparameters to maximize the highest score for recall to remove as many potential blockers as possible.

This too would benefit from increased computational resources as we can specific our optimizations using the Grid Search and Random Search Libraries.

Expanded Use

Finally, it is unlikely that using the algorithms are complex enough to get an accuracy high enough to be used for anything significant. It would be interesting to test our model on a different dataset to evaluate its performance. This could show us a lot about or model generalizability and scope.

References

1. Karim, A., Lee, M., Balle, T. et al. CardioTox net: a robust predictor for hERG channel blockade based on deep learning meta-feature ensembles. J Cheminform 13, 60 (2021). https://doi.org/10.1186/s13321-021-00541-z
2. Gilson MK, Liu T, Baitaluk M, Nicola G, Hwang L, Chong J (2016) Bindingdb in 2015: a public database for medicinal chemistry, computational chemistry and systems pharmacology. Nucleic Acids Res 44(D1):1045–1053
3. Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, Light Y, McGlinchey S, Michalovich D, Al-Lazikani B et al (2012)Chembl: a large-scale bioactivity database for drug discovery. Nucleic Acids Res 40(D1):1100–1107
4. Cai C, Guo P, Zhou Y, Zhou J, Wang Q, Zhang F, Fang J, Cheng F (2019) Deep learning-based prediction of drug-induced cardiotoxicity. J Chem Inform Model 59(3):1073–1084
5. Doddareddy MR, Klaasse EC, IJzerman AP, Bender A (2010) Prospective validation of a comprehensive in silico herg model and its applications to commercial compound and drug databases. ChemMedChem 5(5):716–729
6. Didziapetris R, Lanevskij K (2016) Compilation and physicochemical classification analysis of a diverse herg inhibition database. J Comput Aided Mol Des 30(12):1175–1188
7. Lamothe, S. M., Guo, J., Li, W., Yang, T., & Zhang, S. (2016). The Human Ether-a-go-go-related Gene (hERG) Potassium Channel Represents an Unusual Target for Protease-mediated Damage. The Journal of biological chemistry, 291(39), 20387–20401. https://doi.org/10.1074/jbc.M116.743138