

Block or Not to Block: Predicting hERG Channel Blockade with Machine Learning: A Data-Driven Approach

Shiyang Xin,a Tushar Agashe,**a** Jeffrey Campbell, **a** and Somayeh Motevalli**a**

aUniversity of California, San Francisco, 94158 California, U.S.

The goal of this project is to predict if a given molecule is a hERG channel blocker, using a machine learning approach. The hERG channel is involved in regulating the electrical activity of the heart, and drugs that inhibit it can lead to dangerous side effects, including arrhythmia. Therefore, predicting if a given molecule is a hERG channel blocker is essential for the drug development process. The methodologies we used include Feature Engineering, model training and comparison, and molecular interpretation.

In terms of feature engineering, in order to predict if a given molecule is a hERG channel blocker, the molecular structures of the drugs need to be represented as numerical features. This is done through two methods—MACCS Fingerprints and Morgan Fingerprints. MACCS Fingerprints are predefined binary fingerprints that represent the presence or absence of certain substructures in the molecule, while Morgan Fingerprints are circular fingerprints based on the molecular graph, capturing more detailed local structural information compared to MACCS. In addition, we aim to extract relevant chemical features from drug molecules, using molecular descriptors, which can be used as inputs for machine learning models. In this study, these include various numerical values that describe properties of the molecule, such as molecular weight, LogP, TPSA, and the number of rotatable bonds. All the features are then combined into a single vector for each molecule.

In terms of model training and comparison, the project involves training and evaluating two machine learning models (Random Forest and XGBoost) on the dataset to assess which model performs the best in terms of accuracy and area under the curve. Random forest is an ensemble learning method that builds multiple decision trees and aggregates their results for prediction. It's known for handling large datasets and providing feature importance scores. In comparison, XGboost is a gradient boosting algorithm that iteratively improves a model by focusing on the mistakes made by previous models. It is tuned using hyperparameters such as learning rate, max depth, and regularization parameters to optimize performance. Both models are trained on the training data and evaluated on the validation data using metrics like accuracy, classification report, and area under the curve. The models are then tested on an unseen test dataset to evaluate their generalization ability.

In terms of molecular interpretation, for both two models, feature importance is computed to understand which molecular features (e.g., fingerprints, descriptors) are most predictive of whether a given molecule is a hERG channel blocker. The importance scores are ranked, which can provide insights into which chemical properties are most critical in determining whether a drug is a hERG channel blocker or not.

# Introduction

The human Ether-à-go-go-related gene channel is a type of ion channel found in the heart. It is responsible for conducting potassium ions out of cardiac cells, which is crucial for the regulation of the heart's electrical activity. Specifically, the hERG channel is involved in the repolarization phase of the cardiac action potential, which helps the heart muscle relax after a contraction and prepares it for the next electrical signal.

Certain drugs, particularly pharmaceuticals that target various biological pathways, have been shown to block or inhibit the hERG channel. When this happens, the electrical activity of the heart can be disrupted. Specifically, blocking the hERG channel can lead to prolonged repolarization of cardiac cells, which can cause abnormal heart rhythms, including arrhythmias. Some of these arrhythmias can be severe, leading to potentially life-threatening conditions.

In our study, the data used comes from the Toxicology Data Commons, specifically the "hERG\_Karim" dataset. This dataset contains a set of drug molecules represented as SMILES strings along with binary hERG-blocking labels (block or not to block). The dataset is split into three parts—training (70%, 9412 molecules), validation (10%, 1344 molecules), and testing (20%, 2689 molecules)—enabling the training of models on one set of data while evaluating them on another.

The molecular features are then scaled using StandardScaler to ensure that the models treat all features equally, preventing features with larger numerical ranges from dominating the learning process. After training both two models, their performances are evaluated based on accuracy, the proportion of correctly classified instances; F1-score, which combines precision and recall into a single value; and area under the curve, a measure of the model's ability to distinguish between the two classes, with a higher area under the curve indicating better performance.

# Method

﻿The dataset we selected is the hERG Karin et al. dataset. This dataset contains 13,445 drugs and provides an integrated collection of molecular structures labeled as hERG blockers (<10 µM) or non-hERG blockers (≥10 µM) in SMILES string format. The data was sourced from DeepHIT, the BindingDB database, the ChEMBL bioactivity database, and other scientific literature. It is essentially a binary classification task. Given a drug’s SMILES string, predict whether it acts as a hERG channel blocker (1: <10 µM) or does not block (0: ≥10 µM). No modification was made to the dataset.

Our group selected XGboost as the machine learning algorithm for this task. XGboost brings high predictive power as well as efficiency. In terms of high predictive power, XGBoost is a gradient-boosting framework that excels in predictive tasks. Its ability to capture complex, non-linear relationships between features makes it well-suited for tasks like binary classification. In terms of efficiency, XGBoost is highly optimized for both speed and resource efficiency. It supports parallel processing and includes features like tree pruning and sparse data handling, which help in training large datasets quickly. In contrast, XGboost does have the downside of complexity, as it requires careful tuning of hyperparameters to achieve optimal performance. This can be time-consuming and computationally expensive.

Individually, I selected random forest as the machine learning algorithm for this task. Random forest brings robustness to overfitting, handles high-dimensional data well, and allows feature importance analysis. In terms of robustness to overfitting, random forest reduces the risk of overfitting by averaging predictions across multiple decision trees. This makes it a good choice for complex tasks like predicting hERG channel blockade, where the data may contain noise or non-linear patterns. In terms of handling high-dimensional data, it can effectively handle datasets with many features, making it ideal for molecular structure data (e.g., SMILES strings and derived molecular descriptors). Random Forest can identify important features while ignoring irrelevant ones. In terms of feature importance analysis, random forest provides a built-in mechanism to measure feature importance, helping to understand which molecular properties are most influential in determining hERG blocking potential. This interpretability is particularly useful in drug discovery. In contrast, random forest does have the downside of high computational cost, specifically in training time and memory usage. A longer training time is due to the fact that building multiple decision trees can be computationally expensive and time-consuming, especially for large datasets with many features. Similarly, random forest has a more intense memory usage since all the trees need to be stored in memory, which can be a limitation for very large datasets.

# Results

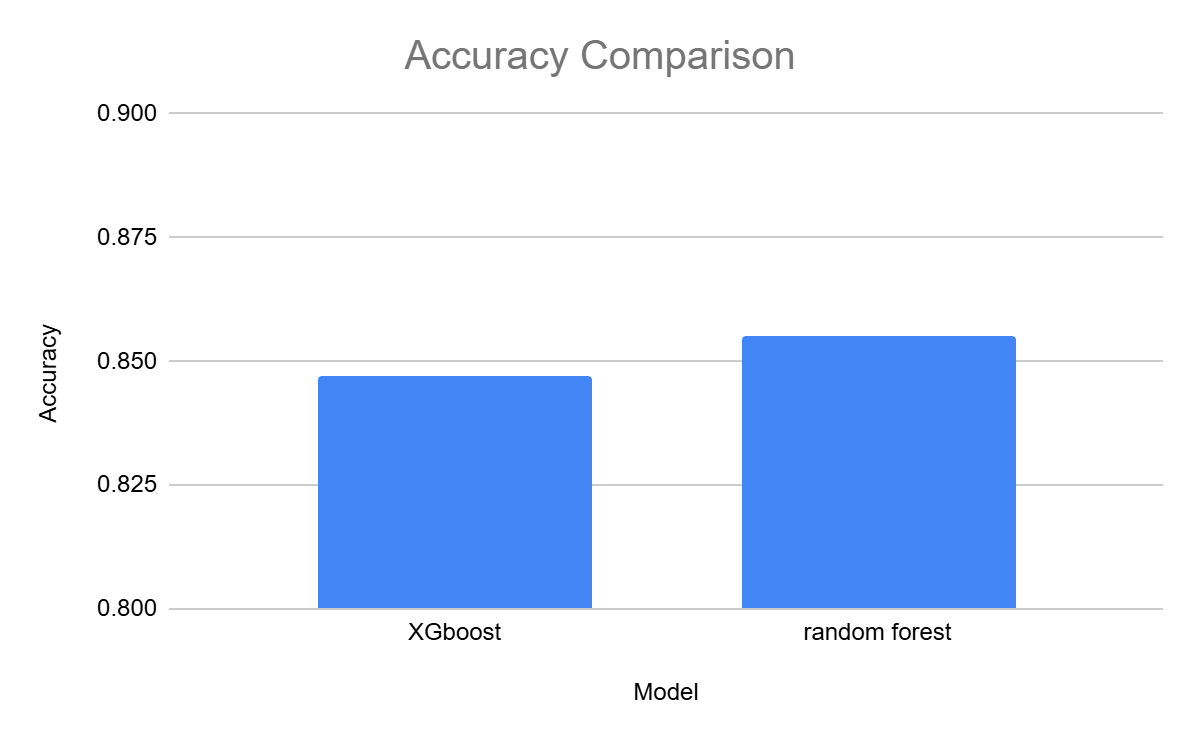


Figure 1: Accuracy comparison between XGboost and random forest.

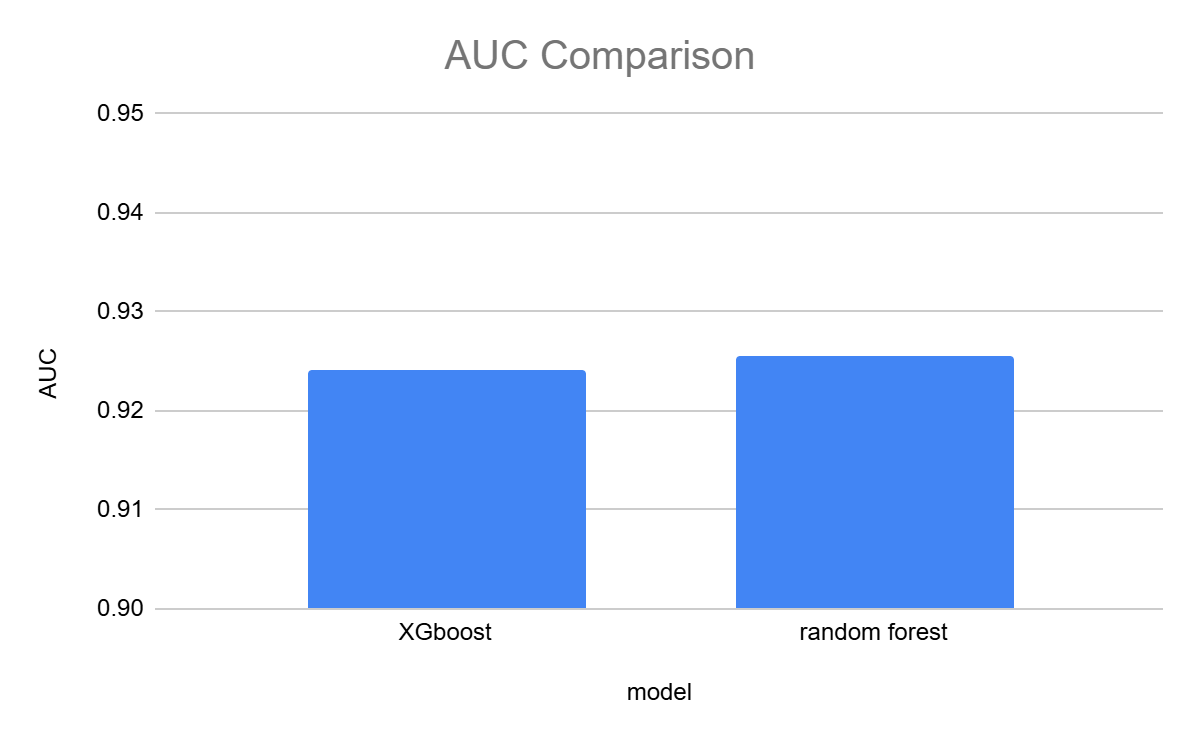


Figure 2: AUC comparison between XGboost and random forest.

The XGboost model yields an accuracy value of 0.8471550762, while the random forest model yields an accuracy value of 0.8549646709. In addition, the XGboost model yields an AUC value of 0.9241361017, while the random forest model yields an AUC value of 0.9255686024.

Furthermore, for XGboost, the precision is 0.85 (for non-hERG blockers) and 0.84 (for hERG blockers); the recall is 0.84 (for non-hERG blockers) and 0.86 (for hERG blockers); and the f1 score is 0.85 (for non-hERG blockers) and 0.85 (for hERG blockers). For random forest, the precision is 0.86 (for non-hERG blockers) and 0.85 (for hERG blockers); the recall is 0.86 (for non-hERG blockers) and 0.85 (for hERG blockers); and the f1 score is 0.86 (for non-hERG blockers) and 0.85 (for hERG blockers).

Despite the small differences in performance metrics, the Random Forest model has a higher accuracy and AUC than XGBoost. Possibly, such a difference arises from model structure and hyperparameter tuning.

In terms of model structure, random forest builds multiple decision trees and aggregates their predictions (through bagging), which helps reduce variance and overfitting. Each tree is trained on a different random subset of the data, and this aggregation of multiple trees often leads to stable, generalized performance. In contrast, XGBoost is a boosting algorithm, which works by training decision trees sequentially. Each tree corrects the errors made by the previous ones. While this typically results in higher predictive accuracy, boosting models like XGBoost can be more prone to overfitting.

In terms of hyperparameter tuning, XGBoost models are sensitive to hyperparameter choices. If the hyperparameters are not finely tuned (such as learning rate, tree depth, subsample rate, etc.), XGBoost may not perform as optimally as it could. The slightly lower performance of XGBoost in terms of accuracy and AUC could be due to suboptimal hyperparameter settings. In contrast, Random Forest is less sensitive to hyperparameter tuning, and defaults often perform quite well, as evidenced by its slightly better performance here.

The hyperparameter tuning approach used in XGboost is RandomizedSearchCV, which is a hyperparameter tuning method from the scikit-learn library that searches for the best combination of hyperparameters by randomly sampling from a predefined distribution or set of values. We performed this method on the following parameters: reg\_lambda, reg\_alpha, max\_depth, learning\_rate, n\_estimators, and colsample\_bytree. For example, for max\_depth, which controls model complexity, we tested it in the range from 3 to 15. The result shows that the model performs the best when max\_depth is equal to 9. The upsides of embracing this approach is that 9 is the max\_depth value between 3 and 15 that achieved the best balance between performance and generalization. The downside of it is that, due to computational complexity, values beyond this range were not tested, as there is a possibility that the optimal value lies outside this range.

Since n\_estimators is the only parameter that needed to be tuned in a random forest model, the hyperparameter tuning approach used in random forest was a random search over a list of values [50, 100, 200, 500, 1000]. As more trees reduce overfitting and improve performance but increase computational cost, we decided to stop at a value in which further increase in accuracy becomes negligible. The upside of taking this approach is that we found n\_estimators=100 gives the optimal balance between performance and training time. The downside of it is that we traded off the chance to maximize the model’s accuracy via using a very large number of trees.

Both models we tried worked despite yielding different performances. We did not use any other models.

# Discussion

After checking the importance of each feature, we concluded the three most important features for both models are respectively LogP (importance factor 0.041), TPSA (importance factor 0.027), and molecular weight (importance factor 0.018).

Both models do not have overfitting problems, as the training accuracy is similar to validation and test accuracy. All three values are around 0.85.

Our approaches do face limitations with regards to feature engineering as well as model capacities. In Karim’s approach, 995 descriptors were used, compared to 7 we used, suggesting more descriptors can be used for our study in the future. Hence, while molecular descriptors provide a robust representation of molecules, they may not fully capture subtle molecular characteristics or interactions relevant to hERG blocking activity. In addition, random forest and XGBoost, while powerful, have limited capacity for feature extraction compared to deep learning methods. The manually engineered features may restrict the models' ability to learn more complex patterns. Hence, we should consider exploring more deep learning algorithms for this study in the future.

# Code Availability

Code used in this study can be found in https://github.com/abcdefucsb/cdd203finalproj

# References

1. Lamothe SM, Guo J, Li W, Yang T, Zhang S. The Human Ether-a-go-go-related Gene (hERG) Potassium Channel Represents an Unusual Target for Protease-mediated Damage. J Biol Chem. 2016 Sep 23;291(39):20387-401. doi: 10.1074/jbc.M116.743138. Epub 2016 Aug 8. PMID: 27502273; PMCID: PMC5034037.

2. 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-z3.