HIGH SCHOOL EDITION
Journal of Student Research

Volume 13 Issue 3 (2024)

A Novel ML Approach to Detect Possible Protein Allosteric Sites for Drug Discovery

Sriram Susarla

John Foster Dulles High School, USA

ABSTRACT

In recent years, allosteric sites on proteins, located away from the primary active binding site, have gained prominence as promising drug targets because of the significant control they provide over biological pathways. The process of identifying allosteric sites through methods such as x-ray crystallography is often time-consuming, where the protein structure and the structure of potential allosteric-site binding molecules are analyzed to discover the allosteric site. Thus, machine learning models have been applied to expedite the process of allosteric drug discovery by contributing to allosteric site identification. In this study, I investigated different machine learning models and evaluated their performance and found that the best models were the random forest classifier and a 5-layer neural network. The random forest had an accuracy of 98.89%, a precision score of 98.20%, a recall score of 99.69%, an F1 score of 98.94%, and an AUC of .9991. The neural network had an accuracy of 97.70%, a precision score of 99.15%, a recall score of 96.04%, an F1 score of 97.57%, and an AUC of .9939. The high accuracy of these models along with their efficiency in running could potentially help the pharmaceutical industry identify these allosteric sites which would help with the allosteric drug manufacturing process.

Introduction

Proteins are composed of 2 or more binding sites, which enable corresponding molecules to bind either to the active (orthosteric) site (Fig. 1), or the allosteric site, resulting in regulation of the protein's function or activity (Fig. 2).[1] This is of interest to the pharmaceutical industry, which manufactures drugs complementary to a binding site of a protein, in order to regulate its role in a biological pathway. The majority of the aforementioned drugs mainly target the active site of the protein, competing with other molecules for the binding site in order to take effect. However, the conformation of active sites remains largely similar across proteins, and may appear to have the same conformation when the protein is mutated.[3, 7] Active site targeted drugs thus can modulate a broad variety of proteins, which may cause unintended consequences if the drug is intended for only one protein.[7] However, allosteric drugs target a unique binding site away from the active site on the protein, changing the conformation of the protein's active site without competition from other molecules trying to bind. The lack of competition allows for a greater control over biological pathways, dampening or heightening the protein's role in the pathway.[7, 11] However, discovery of allosteric sites is required in order to create a compatible drug, and current methods of enzymatic assays[13], fluorescence resonance energy transfer (FRET)[9] and x-ray crystallography[5] are time-consuming and resource-intensive. By applying machine learning to this field, the process of uncovering the allosteric sites can be expedited, aiding in the creation of allosteric site targeted drugs.

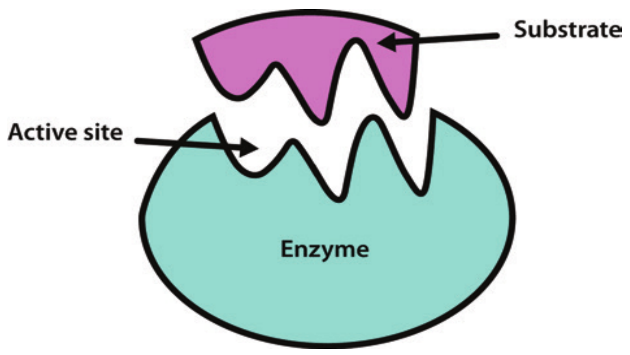


Figure 1. Binding action of molecules (substrates) to active sites, other substrates competing for the binding sites results in competitive inhibition, where the substrate in highest concentration will occupy the active site.[1]

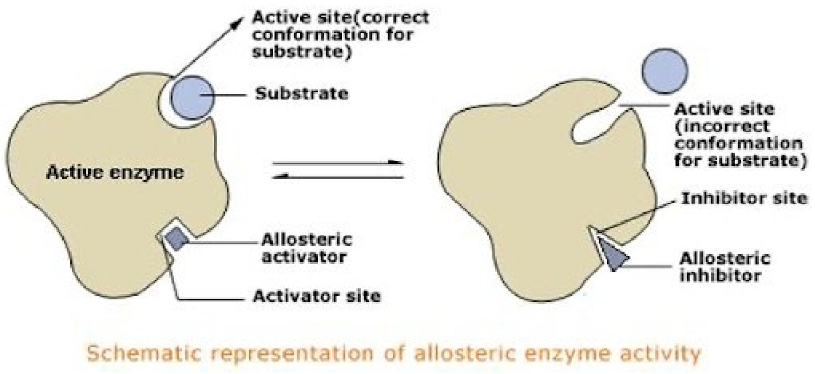

Active site(correct conformation for substrate)
Substrate
Active enzyme
Allosteric activator
Activator site
Active site (incorrect conformation for substrate)
Inhibitor site
Allosteric inhibitor
Schematic representation of allosteric enzyme activity

Figure 2. When an allosteric modulator binds to an allosteric activator or inhibitor site, it changes the conformation of the protein, either activating it or inhibiting it. This phenomenon is known as noncompetitive inhibition because the state of the enzyme cannot be modified by any substrate until the allosteric modulator unbinds from the allosteric site.[3]

Background

The discovery of allosteric sites on enzymes has been attempted through several non-computational and computational methods, which have proved to be time-consuming in several cases. One such method is through x-ray crystallography. Günther et. al. describes a method to take an x-ray crystallography of a library of 5000+ drug and drug candidates as well as the protein it binds to in order to identify the location of an allosteric site.[5] However, performing x-ray crystallography requires crystallization of the proteins, x-ray diffraction, and extensive analysis for each protein and possible drug candidate, which is extremely time-consuming. extremely time-consuming.

As a result, new computational models are emerging to help identify potential allosteric sites.[6, 15] Recently, Xiao et. al. conducted a study using an AutoML framework to identify allosteric sites using machine learning.[16] However, there are many limitations to computational models, due to potential bias in the dataset, which makes it inapplicable until the model is constantly reliable for allosteric site identification.

Dataset

The dataset was obtained from the Protein Allosteric Sites Server (PASSer)[16], and is a combination of two other datasets, Allosteric Sites Database (ASD) and Catalytic and Allosteric Sites Benchmarking Dataset (CASBench), containing examples of proteins with allosteric sites. Each of these examples was passed through FPocket, an algorithm for identifying pockets (components) of a protein. For each pocket detected, it calculated 19 numerical features (Fig. 3), which were then added to the dataset. This dataset was downloaded with features (generated by FPocket) and numerical labels (1 or 0 indicating whether or not the pocket is an allosteric site, respectively), intended for use in a supervised learning problem (Tab. 1). When the ASD portion of the dataset (which was used for this experiment) was unpacked, it was discovered that pockets belonging to a protein were grouped together, which resulted in varying numbers of pockets in each group (as each protein has a different number of pockets) which machine learning models would not accept. I performed preprocessing where I assumed that every pocket is independent, and thus separated each group of pockets into individual entries in the dataset (Fig. 4). After preprocessing steps, the dataset consisted of 4412 samples of protein pockets. As the dataset consisted of a majority of non-allosteric site pocket examples (95.3%), baseline machine learning models were biased (see Methodology/Models section). The Synthetic Minority Over-sampling Technique (SMOTE)[4] algorithm was implemented to rebalance the dataset, adding more samples of allosteric site examples (the minority class) to the dataset. This raised the number of samples in the dataset to 8410, consisting of a 50/50 division of allosteric site and non-allosteric site examples. For the purposes of training and validating the models, an 85/15 train-test ratio was used to split the dataset.

No.	Feature
1	Score
2	Druggability Score
3	Number of Alpha Spheres
4	Total SASA
5	Polar SASA
6	Apolar SASA
7	Volume
8	Mean local hydrophobic density
9	Mean alpha sphere radius
10	Mean alp. sph. solvent access
11	Apolar alpha sphere proportion
12	Hydrophobicity score
13	Volume score
14	Polarity score
15	Charge score
16	Proportion of polar atoms
17	Alpha sphere density
18	Cent. of mass - Alpha Sphere max dist
19	Flexibility

Figure 3. 19 Features Generated by FPocket.[16]

Table 1. Sample entry of data in the PASSer Allosteric Site Database, where each row is a pocket, and pockets sharing the same color (yellow or blue) belong to the same protein grouping.

Feature/Label	Feature 1	Feature 2	Feature X…
1	0.937	0.86	…
0	0.028	0.001	...

…	…	…	…
1	1.004	0.982	…
…	…	…	…

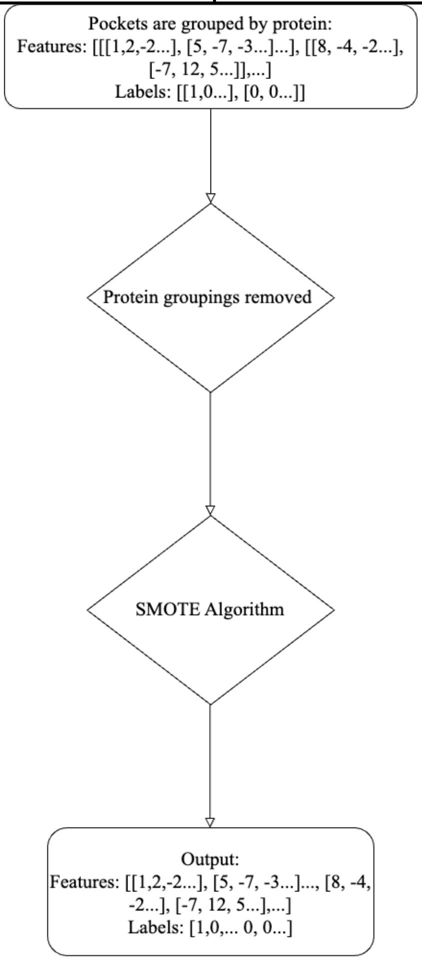

Pockets are grouped by protein:
Features: [[[1,2,-2...], [5, -7, -3...]...], [[8, -4, -2...], [-7, 12, 5...]],...]
Labels: [[1,0...], [0, 0...]]
Protein groupings removed
SMOTE Algorithm
Output:
Features: [[1,2,-2...], [5, -7, -3...]..., [8, -4, -2...], [-7, 12, 5...],...]
Labels: [1,0,... 0, 0...]

Figure 4. Preprocessing pipeline for preparing the balanced supervised allosteric site dataset. Each grouping of numbers is a pocket, and pockets are grouped into proteins. Each label corresponds to one pocket.

Methodology/Models

After converting the data into a form that is acceptable by models and splitting it with an 85/15 split (see Dataset section), several baseline models were trained and evaluated on performance through the python library Scikit-learn.[12]

Logistic Regression

The logistic regression model is popularly used as a baseline classifier due to its interpretability.[2] A simple logistic regression is given by:

$$f(z) = 1/(1 + e^{-z}) \text{ (Fig. 5)}$$

where z is given by:

$$z = b + w_1x_1 + w_2x_2 + \cdots + w_nx_n$$

The above equation is used to train the logistic regression model. Each feature x is multiplied by a weight w which is then summed with a bias term b. Both the weights and the bias are optimized through model training. For every test example, when z is found, it is passed through $f(z)$, a sigmoid function which returns a value between 0 and 1. To classify the input as either 0 or 1, a decision boundary exists at $f(z) = 0.5$. If the output value of $f(z)$ is greater than or equal to the decision boundary, then the input is classified as 1; otherwise, it is classified as a 0.

Random Forest

The random forest classifier is a combination of various decision trees optimized to classify an example during testing (Fig. 6).[14] During training, instances of the data are selected through bootstrapping several times, and a tree is generated for each time data is bootstrapped. A random subset of features is selected that will be used in each tree. After each tree makes a classification based on the test data, a tally is taken of each tree's classification, and the decision with the greatest number of tallies is outputted as the classification. It allows for a generalization to the data, and increases diversity in classification with various features to prevent incorrect classification.

Other Baseline Models

In addition to the above baseline classifiers, the ridge classifier, Support Vector Classifier (SVC), and decision tree models were trained. However, they did not perform nearly as well as the aforementioned models described above.

Smote

Baseline classifiers such as logistic regression were biased to classifying a given test set example as non-allosteric, as the models were trained on a greater number of non-allosteric examples. However, they still achieved a relatively high accuracy because non-allosteric examples made up the majority of the test dataset. As a result, SMOTE, a resampling algorithm was used to resize the data.[4] More samples were added to the minority class in the dataset (allosteric site examples) in order for the data to consist of 50% allosteric site samples and 50% non-allosteric site samples (see Dataset section). By implementing SMOTE, the model will learn the patterns from the data instead of classifying the majority of examples as non-allosteric sites.

Neural Network

Using Keras[8], a 5-layer fully connected neural network was implemented. Each layer in a neural network is composed of neurons, each capturing a key feature from the inputted data and multiplying it by a weight to pass it to the next neuron in the network after an activation function (Fig. 7). After the neural network goes through 1 iteration, backpropagation occurs after calculating the loss (comparison of output to target values). After this, the weights are re-optimized based on the learning rate, and the model is trained again. In the neural network in this experiment, each of our 5 layers contain 128, 64, 32, 16, and 2 neurons respectively. The 5th layer is the output layer, using the softmax activation function to convert it into a probability from 0 to 1, while the other 4 layers use the ReLU activation function. The aforementioned activation function is given by the formula

$f(x) = max(0,x)$, and determines if a neuron fires to the next neuron or not, allowing for more effective learning.

Hyperparameter Tuning

GridSearchCV-facilitated hyperparameter tuning[10] allowed for automatic optimization of model hyperparameters by testing and finding the hyperparameters that returned the best accuracy (Tab. 2). GridSearchCV takes in the parameters we want to test and evaluates every possible model using a combination of each of the hyperparameters. The number of combinations that GridSearchCV tests is given by the formula $f(i) = nP_1 * nP_2 ... nP_i$ where nP_i is the number of values tested for hyperparameter i.

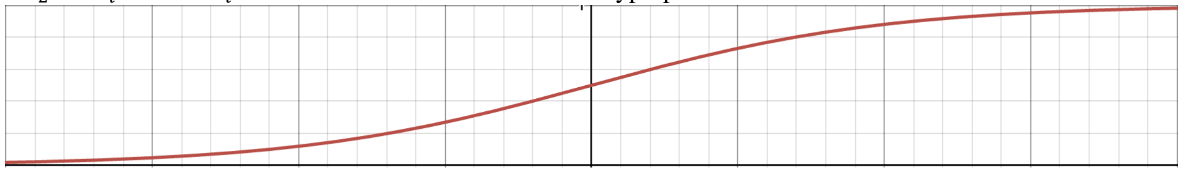


Figure 5. Visualization of the general equation of a logistic regression curve.

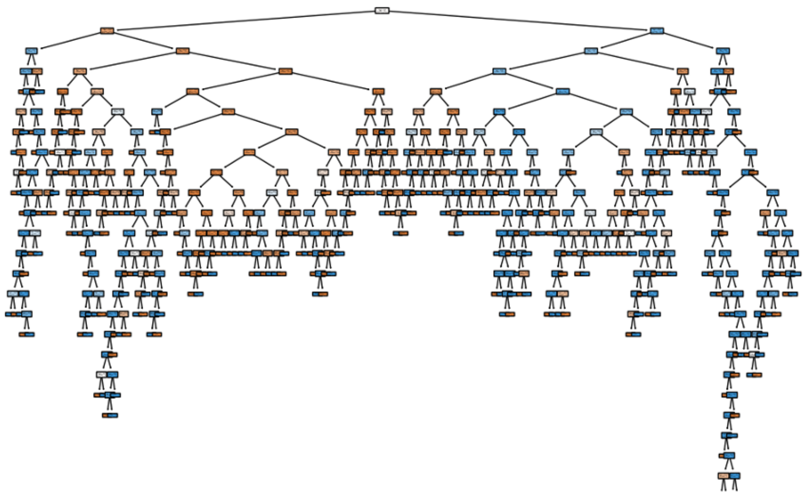


Figure 6. One of the several decision trees that comprise the random forest model. This decision tree coordinates with all the other trees in order to output a final result.

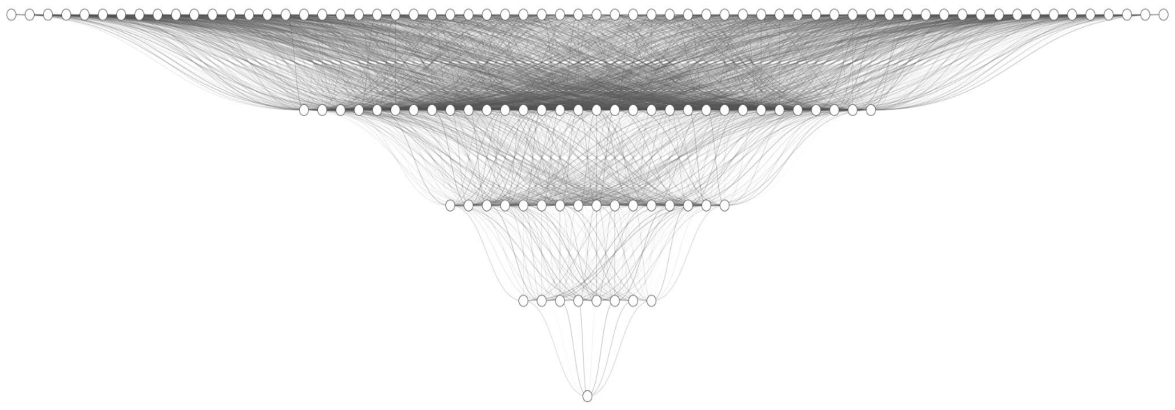


Figure 7. Architecture of the neural network model (number of neurons in each layer is scaled down by 2)

Table 2. Best hyperparameters found by GridSearchCV for each model during hyperparameter tuning.

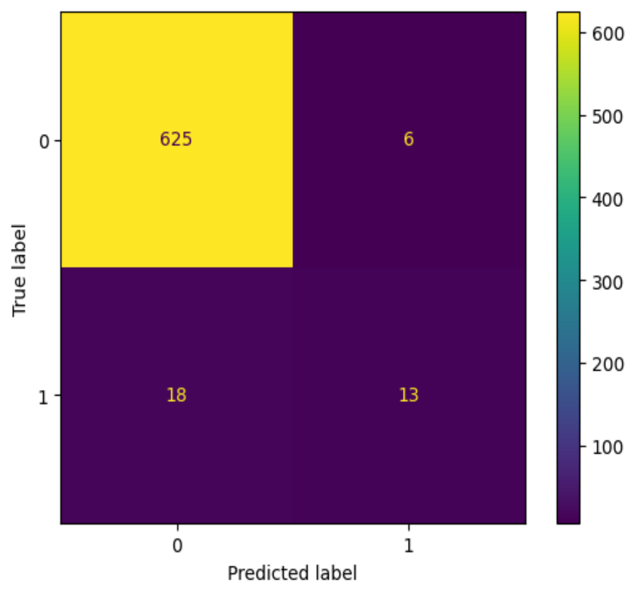
Model	Optimal Hyperparameters
Logistic Regression	{'C': 0.1, 'max_iter': 100, 'penalty': 'l2', 'solver': 'newton-cg'}
Ridge Classifier	{'alpha': 10, 'max_iter': 100, 'solver': 'sparse_cg'}
Random Forest	{'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Decision Tree	{'criterion': 'entropy', 'min_samples_leaf': 2, 'min_samples_split': 2}
SVC	{'C': 100, 'gamma': 'scale'}
Neural Network	{'epochs': 70, 'learning_rate': 0.001}

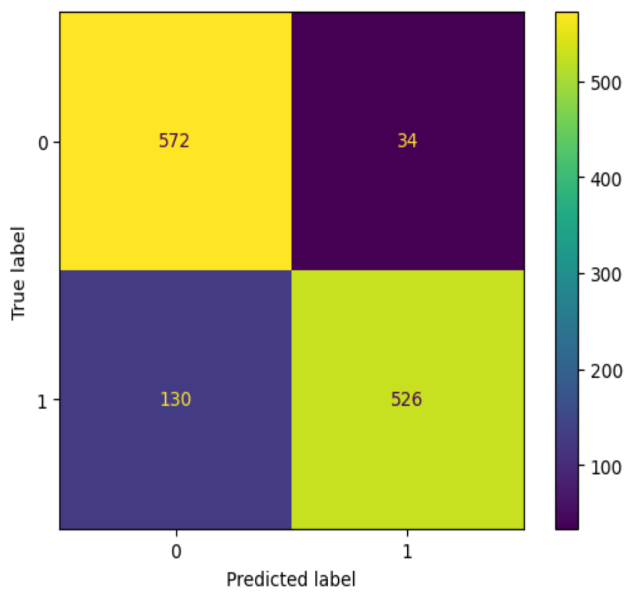
Results/Discussion

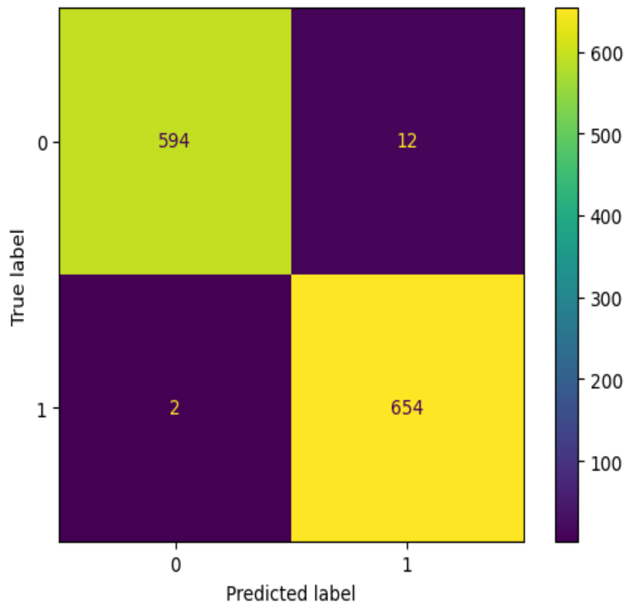
After all necessary rebalancing and hyperparameter tuning, we found that random forest achieved the highest accuracy while minimizing error. To evaluate our models, 6 different metrics were used, which were available through the python library Scikit-learn[12]: accuracy, precision, recall, F1, confusion matrix, and ROC-AUC, and a learning curve was used to evaluate the neural network. (Tab. 3; Figs. 8, 9, 10).

Table 3. Numerical metrics for all tested models, after hyperparameter tuning (LR = Logistic Regression, RC = Ridge Classifier, RF = Random Forest, DT = Decision Tree, SVC = Support Vector Classifier, NN = Neural Network).

Model	Accuracy	Precision	Recall	F1 Score	AUC
LR	87.00	93.93	80.18	86.51	.9359
RC	85.10	94.82	75.46	84.04	.9444
RF	98.89	98.19	99.70	98.94	.9991
DT	95.64	94.92	96.80	95.85	.9610
SVC	85.97	94.77	80.18	96.34	.9411
NN	97.70	99.15	96.04	97.57	.9939







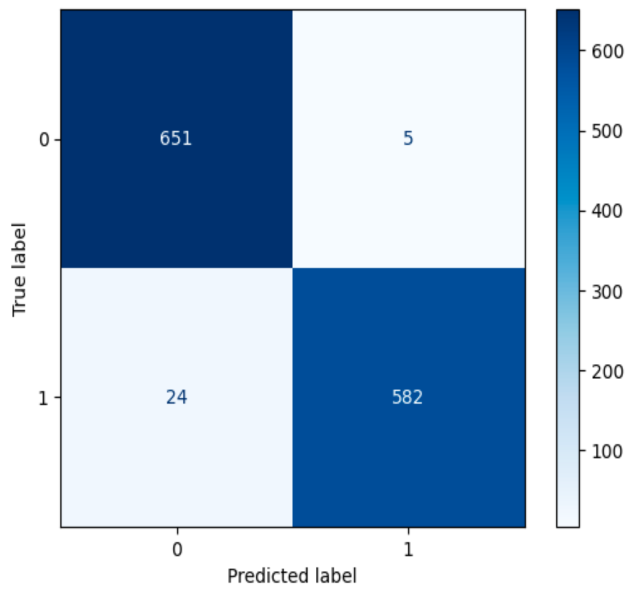


Figure 8. Confusion Matrices for logistic regression (before resampling data), logistic regression (hyperparameter tuned), random forest, and neural network (in order).

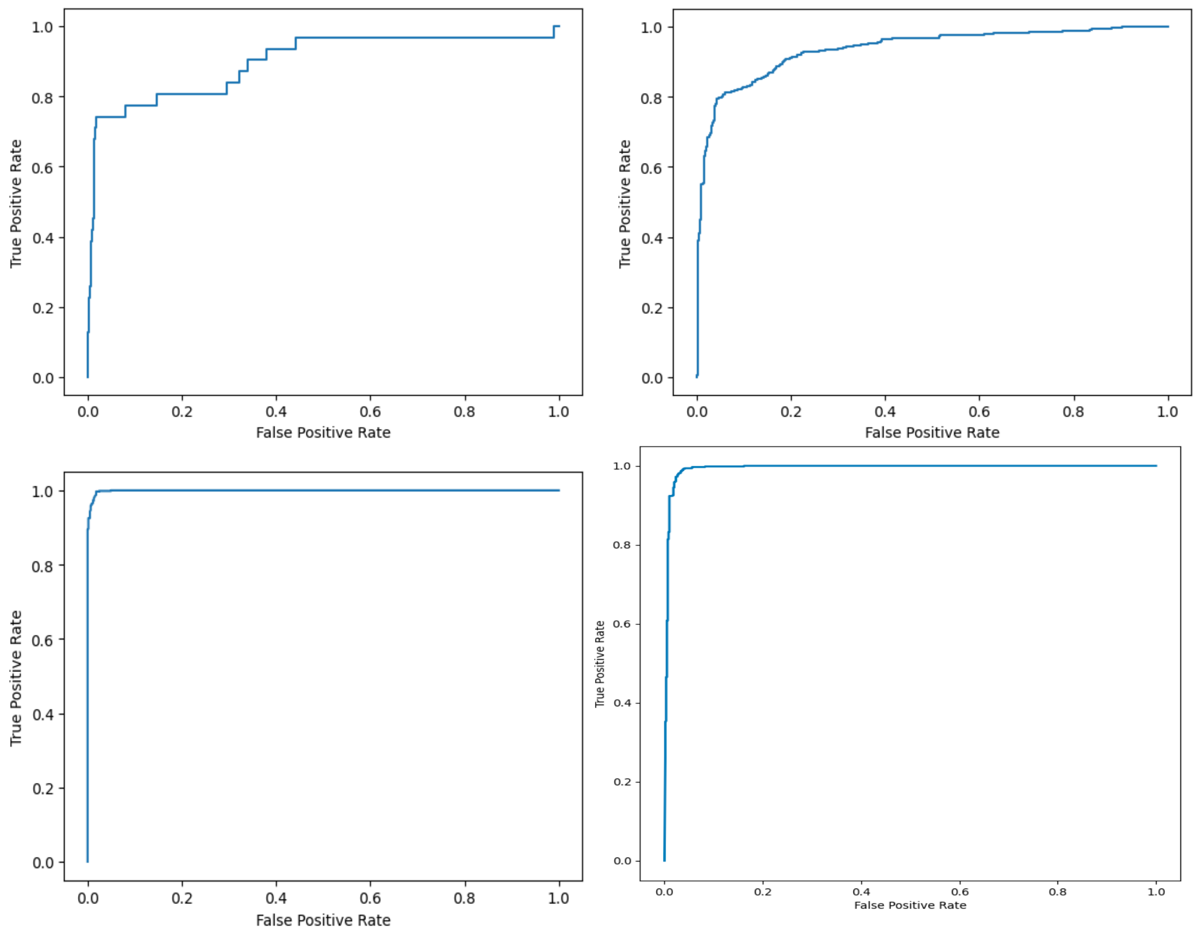


Figure 9. ROC curves for logistic regression (before resampling data), logistic regression (hyperparameter tuned), random forest, and neural network (in order).

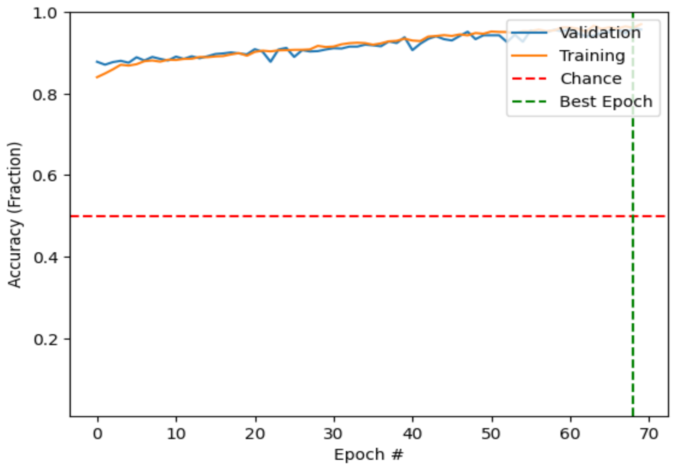


Figure 10. Learning curve for the neural network model. This curve shows that the model is learning, as both the training and validation (testing) accuracy both increase.

Confusion Matrices

A confusion matrix depicts the classification results of the test set for each model. It is used to calculate the rest of the metrics for model evaluation and visualize the performance of a model. On a confusion matrix a “positive” is defined as an example being assigned a label of one, in this case, meaning that the example was classified an allosteric site. An “negative” is an example assigned a label of zero, which would happen if a model classified an example as not being an allosteric site. The number of correctly identified or “true” examples are the squares where the true and predicted labels match up, and the number of incorrectly identified or “false” examples are the squares where the true and predicted labels are different. A high performing model will maximize true negatives and true positives (top left and bottom right squares), while minimizing false negatives and false positives (bottom left and top right squares).

Accuracy

The accuracy of the model is an indication of the total number of true positives and true negatives that the model identified correctly with respect to the total number of samples in the test dataset. However, accuracy is not used as the sole metric for the model, as it can be misleading in scenarios of data imbalance and bias. Accuracy was high before data rebalance, as the models were biased to classifying an example in the test set as non-allosteric; however, it decreased after data rebalanced while other metrics increased. After hyperparameter tuning and data resizing, accuracy was highest in the random forest model (98.89%) and the neural network model (97.70%).

Precision, Recall, F1 Score

Precision, recall, and the F1 score were used in addition to the model’s accuracy to evaluate its effectiveness on the dataset. The formula for precision is given by $TP/(TP + FP)$, where TP is the amount true positives, and FP is the amount of false positives. Precision thus measures the amount of correctly identified allosteric sites with respect to the total number of examples predicted as allosteric sites. However, precision wasn’t an issue for most of the models before hyperparameter tuning, as there were sufficient non-allosteric examples in the dataset, reducing the amount of false positives. After rebalancing the data, the precision score for some of the models decreased slightly, but the models continued to be highly precise. After hyperparameter tuning, the neural network model was the most precise (99.15%) followed by the random forest classifier (98.19%).

Recall is another metric that measures the total number of true positives with respect to the amount of false negatives and true positives combined. It is given by the formula $TP/(TP + FN)$ where FN represents the count of false negatives. In this case, recall would measure the amount of correctly identified allosteric sites with respect to the total amount of allosteric sites in the test set. This metric was especially low before SMOTE was implemented, as the models were biased to classifying an allosteric example as non-allosteric, increasing the number of false negatives. However, after SMOTE was implemented, recall increased greatly in all models. Recall was maximized in the random forest (99.70%) and neural network models (96.04%).

The F1 score is the harmonic mean of precision and recall, and is used to find the trade-off between precision and recall. The F1 score is given by the formula $2 * (Precision * Recall)/(Precision + Recall)$, returning a value between 0 and 1. As outputs approach closer to 1, it indicates a higher model performance, whereas an output closer to 0 indicates a lower performance. The F1 score was greater in models which had a high precision and recall, including in the neural network (97.57%) and random forest classifier (98.94%).

ROC-AUC

The ROC Curve shows the trade-off between the false positive rate and the true positive rate (Fig. 12). Each point on the graph represents a decision threshold, or the trade-off between false positives and true positives that the graph chooses. A model with 100% accuracy will have an ROC curve that is a vertical line from the origin to 1.0, and then the plateaus. To calculate the quality of the model's predictions, the AUC (area under the curve will be calculated). For a random binary classification prediction, the AUC is 0.5, as there is an equal probability of getting a true positive and a false positive. However with training, the AUC will increase as the curve goes above the line $y = 0.5x$. As the neural network and the random forest classifiers had the greatest accuracies, their AUC was also the highest, at .9939 and .9991 respectively.

Conclusion

In this study, I tested several machine learning models in order to determine whether allosteric sites can accurately be classified. Due to the lack of data availability, data bias was combatted using the SMOTE algorithm, allowing for models to learn the patterns in the data. After optimizing each model by tuning their hyperparameters, the random forest model was by far the best model, followed by the neural network, after evaluation with various metrics (accuracy, precision, recall, F1 score, confusion matrices, and ROC-AUC). As the random forest algorithm increases diversity in classification using a random selection of features for each tree, it was able to generalize to the patterns in the data, and incorrect classifications were minimized by taking a tally of each tree's classification and choosing the majority output. The same accuracy from the random forest model was unmet by other models such as the logistic regression, as the relationship may have been too complex to be captured, or the other models failed to generalize to the data. The high 98.89% accuracy of the random forest and the 97.7% accuracy of the Keras neural network model can help many pharmaceutical industries who currently do not focus on allosteric site discovery due to constraints, if these models are implemented. By making advances in the field on allosteric site discovery, drug manufacturers are closer to implementing allosteric site targeted drugs in order to gain greater control over biological pathways. In the future, I would like to address certain limitations of the model to see if it changes the results of this study. For example I would like to verify whether isolating the pockets during preprocessing still yields the same results, comparing the accuracy with a larger dataset versus a smaller dataset resized by SMOTE, or whether conformation data of the protein can help identify the allosteric site in addition to other features.

Acknowledgments

I would like to thank my mentor Matthew Radzihovsky for editing and reviewing this paper as well as providing input throughout this project. I would also like to acknowledge Xiao et. al. for making the PASSer Allosteric Sites Database available for use in projects.

References

1. *(PDF) Enzymes: principles and biotechnological applications*. (n.d.). ResearchGate. https://www.researchgate.net/publication/283779912_Enzymes_principles_and_biotechnological_applications
2. Bewick, V., Cheek, L., & Ball, J. (2005). Statistics review 14: Logistic regression. *Critical Care*, *9*(1), 112. https://doi.org/10.1186/cc3045

3. Buntz, B. (2023, September 27). *Why allosteric drugs represent a unique small molecule approach.* Drug Discovery and Development. https://www.drugdiscoverytrends.com/allosteric-drugs-a-differentiated-small-molecule-approach/
4. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*(16), 321–357. https://doi.org/10.1613/jair.953
5. Günther, S., Reinke, P. Y. A., Fernández-García, Y., Lieske, J., Lane, T. J., Ginn, H. M., Koua, F. H. M., Ehrt, C., Ewert, W., Oberthuer, D., Yefanov, O., Meier, S., Lorenzen, K., Krichel, B., Kopicki, J.-D., Gelisio, L., Brehm, W., Dunkel, I., Seychell, B., & Gieseler, H. (2021). X-ray screening identifies active site and allosteric inhibitors of SARS-CoV-2 main protease. *Science*. https://doi.org/10.1126/science.abf7945
6. Huang, W., Nussinov, R., & Zhang, J. (2017). Computational Tools for Allosteric Drug Discovery: Site Identification and Focus Library Design. *Methods in Molecular Biology (Clifton, N.J.)*, *1529*, 439–446. https://doi.org/10.1007/978-1-4939-6637-0_23
7. Jarvis, L. (2019, March 10). *Drug hunters explore allostery's advantages*. Chemical & Engineering News. https://cen.acs.org/pharmaceuticals/drug-development/Drug-hunters-explore-allosterys-advantages/97/i10#:~:text=An%20allosteric%20site%20is%20more
8. keras-team. (2019, January 22). *keras-team/keras*. GitHub. https://github.com/keras-team/keras
9. Lecat-Guillet, N., Monnier, C., Rovira, X., Kniazeff, J., Lamarque, L., Zwier, J. M., Trinquet, E., Pin, J.-P., & Rondard, P. (2017). FRET-Based Sensors Unravel Activation and Allosteric Modulation of the GABAB Receptor. *Cell Chemical Biology*, *24*(3), 360–370. https://doi.org/10.1016/j.chembiol.2017.02.011
10. Liashchynskyi, P., & Liashchynskyi, P. (2019). *Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS.*
11. Ni, D., Wei, J., He, X., Ashfaq Ur Rehman, Li, X., Qiu, Y., Pu, J., Lu, S., & Zhang, J. (2021). Discovery of cryptic allosteric sites using reversed allosteric communication by a combined computational and experimental strategy. *Chemical Science*, *12*(1), 464–476. https://doi.org/10.1039/d0sc05131d
12. Pedregosa, F., Varoquaux, G., Michel, V., Thirion, B., Grisel, O., Blondel, M., Müller, A., Nothman, J., Louppe, G., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Gramfort, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python Pedregosa, Varoquaux, Gramfort et al. *Journal of Machine Learning Research*, *12*, 2825–2830.
13. Qi, Y., Wang, Q., Tang, B., & Lai, L. (2012). Identifying Allosteric Binding Sites in Proteins with a Two-State Go¯ Model for Novel Allosteric Effector Discovery. *Journal of Chemical Theory and Computation*, *8*(8), 2962–2971. https://doi.org/10.1021/ct300395h
14. Tin Kam Ho. (1995, August 1). *Random decision forests*. IEEE Xplore. https://doi.org/10.1109/ICDAR.1995.598994
15. Wagner, J., Lee, C. S., Durrant, J. D., Malmstrom, R. D., Feher, V. A., & Amaro, R. E. (2016). Emerging Computational Methods for the Rational Discovery of Allosteric Drugs. *Chemical Reviews*, *116*(11), 6370–6390. https://doi.org/10.1021/acs.chemrev.5b00631
16. Xiao, S., Tian, H., & Tao, P. (2022). PASSer2.0: Accurate Prediction of Protein Allosteric Sites Through Automated Machine Learning. *Frontiers in Molecular Biosciences*, *9*. https://doi.org/10.3389/fmolb.2022.879251