Prediction of Antibiofilm peptides for drug discovery

Nivedha Balakrishnan, Siddarth Magidewar, Chidroop Sagar, Jashwanth Kumar

1. Problem Statement and Motivation

Biofilm development is frequently caused by microbial pathogens. Pathogens such as fungi, bacteria that have colonized surfaces like animal or plant tissues, medical equipment like catheters, and mechanical heart valves can create biofilms. Finding a different way to manage biofilm infections is crucial given the rise in chronic infections and antibiotic resistance caused by biofilm. These pathogens that produce biofilms can be inhibited from growing, becoming virulent, and forming biofilms by antibiofilm peptide interactions.

In this project, we developed a binary machine learning classification system that can classify peptides with antibiofilm properties. The positive dataset is the antibiofilm peptides which has the properties to eliminate biofilms, and the negative dataset contains peptides that aid in the formation of biofilms. We applied four different machine learning algorithms including Support Vector Classification (SVC) with RBF kernel, K Nearest Neighbor, Random Forest and XGBooster. Since our dataset is imbalanced, we applied SMOTE to increase the number of positive samples. Feature Importance method has been applied as a feature selection technique to reduce the number of features. On comparing the performance between each model, Random Forest and XGBooster provided the best F1 scores around 98.9% and 97.4% respectively.

2. Background

Drugs can be used to cure or prevent various illnesses and ailments. As human health has been declining globally for the past few decades, drugs have become increasingly important in the modern day. Finding new medications, however, is a laborious and expensive process. It takes more than 8 to 12 years to discover and develop a new medicine. It is crucial to evaluate and look into efficient and effective strategies in order to reduce the rising costs and speed up the discovery process. Machine Learning is one such strategy that can help speed up the process especially in Virtual Screening. Our project is focused on building a Machine Learning Classification model that could predict the proteins that can eliminate the biofilm formation.

The most difficult task in the fight against bacterial resistance is to create a promising antibiofilm peptide while reducing cytotoxicity and boosting efficacy. Although dealing with a large library of peptides takes a lot of time and money, computational methods offer hope for quickly and affordably screening further new antibiofilm peptides [2]. In the past ten years, as machine learning and artificial intelligence have become more prevalent in the healthcare sector, the in silico method of screening new peptides has gained a lot of popularity. In this study, we used a computer model to find peptides with the potential to prevent the establishment of biofilms. Finding peptides that are effective against biofilm now requires very little data from available sources. We included features such as primary features, amino acid composition (AAC), dipeptide composition (DPC), composition transition and distribution (CTD) features. in total, one peptide sequence consists of 572 features.

3. Literature Review

The development of peptide and protein-based therapeutics to treat a variety of ailments has been the subject of scientific study over the last few decades. Nevertheless, the toxicity of peptide-based therapies is one of its limitations. In a study Pallavi et al developed models to predict the toxicity of peptides. Using different properties of peptides they came up hybrid Support Vector Machine and it achieved an accuracy of 90% independent datsets. A web server called ToxinPred has been created based on the mentioned work which is useful for predicting toxicity [1].

Gupat et al propsed a computational method to predict biofilm inhibiting peptides. To capture sequence-based attributes and find distinctive sequence patterns, the empirically verified biofilm inhibitory

peptide sequences were used. They constructed Support Vector Machine-based model for prediction and, in addition, created hybrid models by incorporating information about sequence motifs patterns. By utilizing 10-fold cross validation, the model gave an accuracy of 97.19% and Matthews Correlation Coefficient (MCC) of 0.84 on validation dataset [3].

Bipasa et al, have discussed how biofilms and microbial illnesses are related and how targeting them is an effective strategy to limit microbial virulence while minimizing the development of antibiotic resistance. In order to categorize new peptides with potential antibiofilm activities, authors have created machine learning models to recognize the distinctive properties of existing antibiofilm peptides and to extract peptide databases from a variety of environments. There are two types of datasets one has data about 242 different antibiofilm peptides and other has peptides which promote formation of biofilm. The model achieved accuracy of 98% and F1 score greater than 0.90 and Matthews correlation coefficient (MCC) greater than 0.81. This paper discovers potential new members for biofilm eradication and presents a novel silico method for predicting antibiofilm success [4].

4. Methodology

4.1 Data collection

The dataset for the project is extracted from two sources: National Center for Biotechnology Information (NCBI) and Uniprot website. The data extracted from the sources is in a FASTA, a text-based format that represents the peptide sequence. The final data set which is to be given to the model is divided into two sets positive and negative. The positive set contains information on antibiofilm peptide which has 349 records. The negative set contains information on biofilm peptide which has 134867 records.

4.2 Feature Engineering

4.2.1 Feature extraction:

We used the Biopython module to extract different peptide features, which are numerical representations of the peptide sequence, structure, and physicochemical properties. Our features are of 4 sets corresponds to protein sequences they are mentioned below.

Feature set	**No. of features**
Primary features	5
AAC (Amino Acid Composition)	20
DPC (Dipeptide Composition)	400
CTD (Composition Transition Distribution)	147
Total	**572**

4.2.2 Feature reduction:

We found that we've abundant number of features in our data. So, we plotted a correlation matrix for all the features sets to understand the relationship between them. In Figure 1, we can see that the sequence length and weight are highly correlated among primary features. This makes sense because increase in sequence length means adding more amino acid which in turn will increase the molecular weight of the protein. The correlation between the rest of the features is not more than 10%, this indicates that these features are independent to each other.

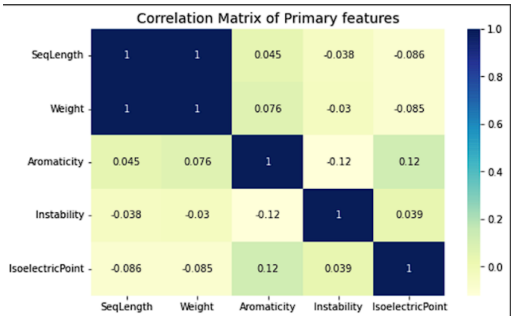

Figure 1: Correlation Matrix of Primary features

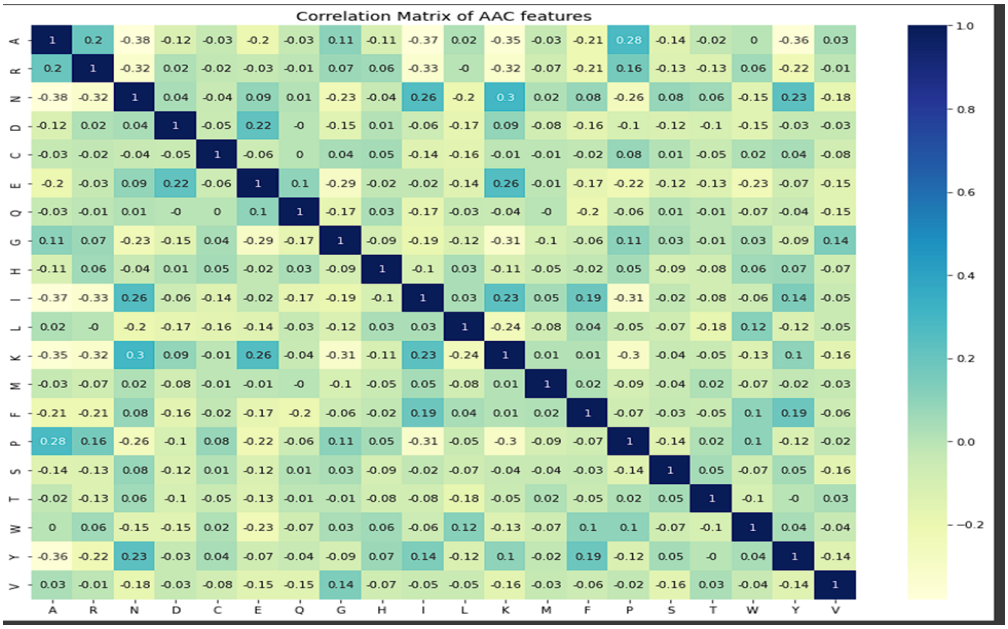

Figure 2: Correlation Matrix of AAC features

In AAC, no two features have more than 45% of correlation as shown in figure 2. Since, there are high number of features in DPC and CTD features, we wrote a python code to get highly correlated features. We found that there are 26 pairs of highly correlated features, and one feature from those pairs were removed.

4.3 Data Pre-processing

Regardless of how good a ML model is, good results cannot be obtained from bad data. So, before passing the data into the model, we performed several operations on the data to ensure its quality. The steps taken are:

4.3.1 Data cleaning:

First, we removed duplicate values from both positive and negative dataset. We ended up with 227 positives and 124759 negatives. Since negative peptides are too large compared to positive ones, we randomly selected 2270 negatives. Thus, our datasets contains 1:10 ratio of positive and negative sets.

4.3.2 Data preparation and normalization:

After combining and splitting the datasets into train, validation and test sets (60:20:20 ratio), we performed data normalization on the each sets to bring all the data into the range [-1, 1]**.**

4.4 Data Modeling

4.4.1 SVM RBF:

Support Vectors are the data points that are located at the margin. They are extremely important in establishing the direction and location of the ideal hyperplane [11]. In fact, these alone have an impact on the model's performance, making them the most crucial points in the dataset. The algorithm aims to maximize the margin which is the distance between the closest point (from either class) to the optimal hyperplane. If the data cannot be separated linearly, the Kernel technique is used to change the original dimensional space into a higher dimensional one. One such kernel is radial bias which is used in this project.

4.4.2 KNN:

The "k-nearest neighbors" (KNN) technique calculates the chance that a data point will belong to one group or another based on which group the data points closest to it do. It is a supervised machine learning algorithm used in problems like classification and regression. It is mainly employed for classification problems. KNN is a lazy learning algorithm or lazy learner because it doesn't train until it is supplied with the training data. It makes this algorithm ideal for data mining.

4.4.3 Random Forest

A supervised machine learning algorithm which consists of many base learners known as decision trees. With the help of multiple base learners, it is possible to train and make predictions on previously unseen data. The Random Forest algorithm is highly efficient as it has low bias and low variance, thus making the performance invariant to new unseen data. It is an ensemble algorithm that utilizes a technique known as Bootstrap sampling: randomly sampling the rows from the training set along with a replacement [5].

The peptide data are divided into the random sample with the help of bootstrapping method. The data is passed into multiple base learners, the majority vote is taken into consideration, and the model predicts if it is a negative or positive classification.

4.4.4 XGBoost

Extreme Gradient Boosting is an ensemble technique that increases the performance of weak learners, using gradient descent architecture through algorithmic enhancements and systems optimizations. It basically increases the weights of the weak learners from the past and thus improves the model's performance. It utilizes regularization techniques such as L1 and L2 regularisation techniques to reduce the overfitting of the data. It can handle missing values well along with tree pruning using a depth-first approach [6]. The peptide data are passed through multiple base learners and then the weights are updated while passing them to the next set of decision trees. The weak learners are given more weights and thus the classification is done once all the weights are updated.

4.4.5 SMOTE

SMOTE is a method for building classifiers from Imbalanced datasets. The data where we observe the unequal distribution of target classes is referred to as Imbalanced data. In Smote we normalize the data. We extrapolate or interpolate and create new or fake data by averaging the lower-class data. It generalizes the samples by making more features available to each class. The problem with SMOTE is it may oversample noise. The peptide data has imbalanced data where the

negative dataset has 227 features but the negative dataset has 2270 features. After applying smote both positive and negative sets became equal in number.

4.4.6 Principal Component Analysis (PCA)

When there are fewer links between features to consider, it is less likely to overfit the model, and It can be done by lowering the dimension of feature space. PCA helps in dimensionality reduction. PCA is helpful when there is uncertainty to identify the features that must be removed for dimensionality reduction and to ensure components are independent. The problem with PCA is that there is a chance of information loss []. In the peptide dataset, there are around 500 features. With the help of PCA, we have selected 289 components based on a cumulative proportion of 95% [9].

4.4.7 Feature Importance (FI)

For building a machine learning model, feature selection is one of the essential steps which is often overlooked. Many techniques help in feature selection. Feature importance is one such method. Feature Importance is a method that scores the input features with respect to the impact it has on predicting a particular variable. It helps to comprehend the features which are irrelevant to the model. The higher the score, the impact factor of the component is higher. We have used feature importance.

5. Model Evaluation metrics

In this paper we have used two model evaluation metrics which are the F1- measure and Matthews Correlation Coefficient (MCC) in order to evaluate our models performance. The scores of all four models were evaluated and plotted using a histogram as seen below. Figure 3 shows the 10 fold cross validation scores of the baseline models. We can see that random forest and xgbooster are overfitting and SVM RBF is giving poor performance.

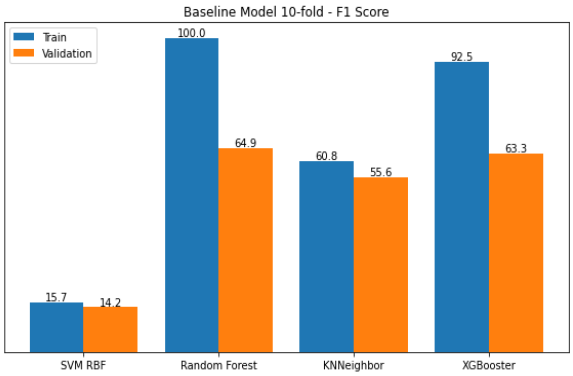

Figure 3: 10 fold validation of Baseline Model

Since our dataset is highly imbalanced, we calculated 10 fold cross validation F1 scores of all the model with different proportion of datasets. We also performed smote and plotted the scores for comparison as shown in figure 4. We found that applying oversampling with SMOTE gave best scores for all the model. Thus, we decided to use SMOTE for the rest of the process.

As mentioned before, the dataset has high number of features. To reduce the dimensions we applied two techniques. First, PCA is applied and first 289 components are selected based on 95% of cumulative proportion. Second, feature importance method is applied and we selected 303 features based on the 0.001 threshold. Hyperparameter tuning is applied to select the best hyperparameters for each model with the selected feature sets. Figure 5, 6 shows the results of after applying PCA and feature importance method respectively. We can see that the process eliminated overfitting and the scores are increased for all the model. However, Random Forest and XGBooster provided the best scores.

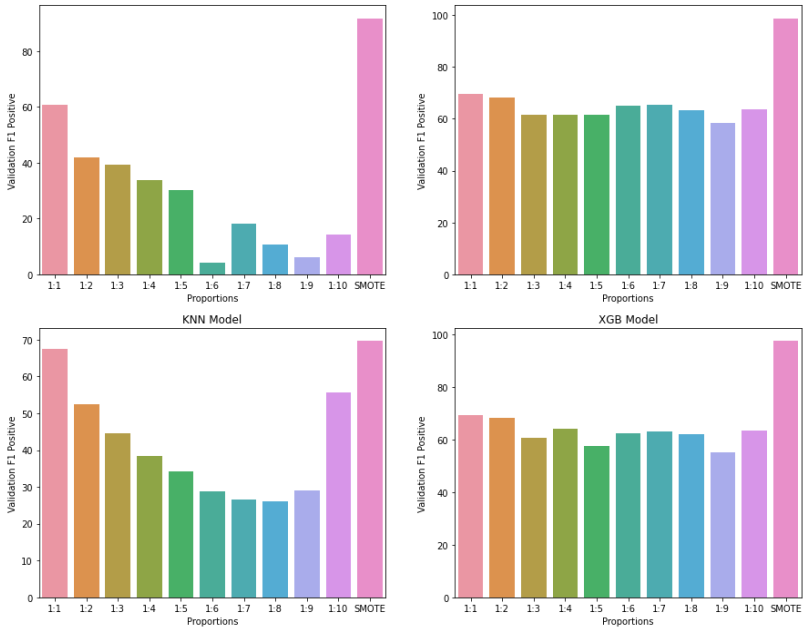

Figure 4: 10 fold CV F1 scores for different proportions of datasets

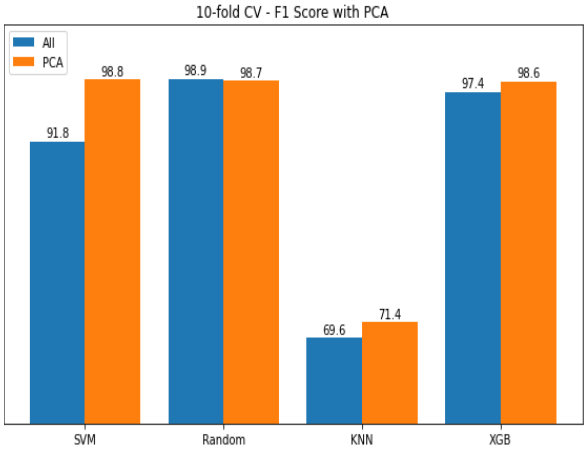

Figure 5: Scores after PCA

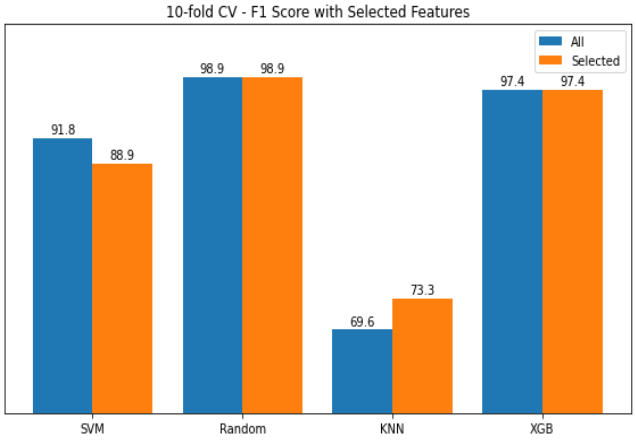

Figure 6: Scores after FI

6. Conclusions

While performing different proportions of the dataset, we noticed the models achieve better performance using SMOTE. Evaluation of SMOTE models on selected features showed an increase in F1 score and MCC score in Random Forest and KNN while it reduced the performance of the SVM model. XGBoost remained invariant. Evaluation of SMOTE models on PCA (Principal Component Analysis) showed an increase in F1 score and MCC score in SVM and XGBoost while it reduced the performance of KNN, Random Forest remain invariant. Hyperparameter tuning of the models increased the performance of all the models with respect to F1-score and MCC score. Random Forest Model and XGBoost provided the best results.

7. Future scope

Implement NLP to classify the sequences by tokenizing each amino acid. ProteinGAN to generate similar sequences and use discriminator to identify sequences that could have anti biofilm properties in order to generate the number of positives [12]. Implement DeepLearning models such as Bi-LSTM, Transformers and BERT models which can be used in NLP.

8. Contributions

Nivedha Balakrishnan	Feature extraction and Data Modelling
Chidroop Sagar	Modeling and Evaluation
Siddharth Magidewar	Data Collection and preparation
Jashwanth Kumar	Data Engineering

References

[1] P, B., J, Y., S, F., & SWI, S. (n.d.). *AMPEP: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest*. Scientific reports. Retrieved November 30, 2022, from https://pubmed.ncbi.nlm.nih.gov/29374199/

[2] D. Pletzer and R. E. W. Hancock, “Antibiofilm peptides: Potential as broad-spectrum agents,” Journal of Bacteriology, vol. 198, no. 19, pp. 2572–2578, 2016.

[3] Gupta, S., Kapoor, P., Chaudhary, K., Gautam, A., Kumar, R., Consortium, O. S. D. D., & Raghava, G. P. S. (n.d.). *In silico approach for predicting the toxicity of peptides and proteins*. PLOS ONE. Retrieved November 30, 2022, from https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0073957

[4] Bose, B., Downey, T., Ramasubramanian, A. K., & Anastasiu, D. C. (1AD, January 1). *Identification of distinct characteristics of antibiofilm peptides and prospection of diverse sources for efficacious sequences*. Frontiers. Retrieved November 30, 2022, from https://www.frontiersin.org/articles/10.3389/fmicb.2021.783284/full

[5] S. E. R, “Random Forest: Introduction to random forest algorithm,” *Analytics Vidhya*, 30-Nov-2022. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/. [Accessed: 30-Nov-2022].

[6] H. Singh, “Understanding random forests,” *Medium*, 24-Mar-2019. [Online]. Available: https://medium.com/@harshdeepsingh_35448/understanding-random-forests-aa0ccecdbbbb. [Accessed: 30-Nov-2022]

[7] V. Morde, “XGBoost algorithm: Long may she reign!,” *Medium*, 08-Apr-2019. [Online]. Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d. [Accessed: 30-Nov-2022].

[8] J. Korstanje, “The F1 score,” *Medium*, 31-Aug-2021. [Online]. Available: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6. [Accessed: 07-Dec-2022].

[9] M. Brems, “A one-stop shop for principal component analysis,” *Medium*, 26-Jan-2022. [Online]. Available: https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c. [Accessed: 07-Dec-2022].

[10] “SKLEARN.METRICS.MATTHEWS_CORRCOEF,” *scikit*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.matthews_corrcoef.html. [Accessed: 07-Dec-2022].

[11] Gunn, S., (1998). *Support vector machines for classification and regression.* Technical Report, ISIS, Department of Electronics and Computer Science, University of Southampton.

[12] Gupta, A., Zou, J. Feedback GAN for DNA optimizes protein functions. *Nat Mach Intell* 1, 105–111 (2019). https://doi.org/10.1038/s42256-019-0017-4