Predicting Anti-Microbial Peptides (AMP) through ensemble-based machine learning approach

Atharva Mandar Phatak[1,†], **G K Harish Balaji**[2,†], **Sanjeev M**[3,†], **Hrutik Ravindra Pawar**[4,†], **Krithika M**[5,†] **and Ramavath Yamuna**[6,†]

[1]*BE21B009*
[2]*BE21B017*
[3]*BE21B034*
[4]*BE21B026*
[5]*BE21B022*
[6]*BE21B030*
[†]*These authors contributed equally to this work*

Date of Submission on May 24, 2024

Abstract

Predicting which peptides have antimicrobial properties (AMPs) is essential for developing new antibiotics. This study outlines a method to accurately distinguish AMPs from non-antimicrobial peptides (non-AMPs) using machine learning techniques. We collected peptide data from databases such as APD, CAMP, DBAASP, and UniProt, extracted important features, performed preliminary analysis, and successfully differentiated AMPs from non-AMPs using ensemble based machine learning methods. The final model can be accessed through an user-friendly online tool. Our extensive collection of data and feature selection helped to develop a more accurate prediction tool for predicting AMP based on the input sequence.

Keywords: *AMP, Amino Acid Encoding, Machine Learning, Database, Visualisation*

1. Introduction

Antimicrobial peptides (AMPs), are essential components of the innate immune system, have broad-spectrum activity against pathogens and are less likely to induce resistance compared to traditional antibiotics. The rise of antibiotic-resistant bacteria has spurred intense research into AMPs as potential next-generation antibiotics. AMPs work by disrupting microbial membrane integrity or targeting intracellular components. The regulation of antimicrobial peptide expression plays a crucial role in inflammation and host defense (Navarro, Saravanan, Naglik, 2013). Accurate classification of peptides as AMPs or non-AMPs is crucial for therapeutic development and understanding innate immunity. This study employs machine learning to distinguish AMPs from non-AMPs by analyzing structural and physio-chemical properties, achieving high accuracy and efficiency. The research enhances AMP prediction accuracy, contributing to new antimicrobial therapies and offering a strong foundation for future studies in AMP discovery.

2. Material and Methods

2.1. Data Collection

The effects of antimicrobial peptides were studied (Usmani et al., 2018). The application of AMPs in microbial resistance was investigated (Lee, Chong, Yoon, 2017). The regulation of antimicrobial peptide expression was discussed (Navarro, Saravanan, Naglik, 2013). The role of AMPs in immune response was explored (Jain, Gupta, Jain, 2011). We collected AMP and Non-AMP data from these research papers. The data includes 61587 AMP (29112) and Non-AMP (32475) sequences. Of this we used 80% data to train the models while 20% data to test (test size = 0.2) the model. Please refer to the features mentioned in the analysis from Table-1 in Annexure.

2.2. Feature Selection

We employed Recursive Feature Elimination (RFE) with a Random Forest classifier to identify the top 10 features.

2.3. Descriptive Analysis

We calculated the mean, median, mode, skewness, kurtosis, and percentiles for top ten features (the names of features are mentioned in Annexure).

2.4. Inferential Analysis

We conducted Z-tests, T-tests, ANOVA, and Mann-Whitney U tests to determine the statistical significance of these features

2.5. Visual Analysis

We created histograms, density plots, boxplots, jitter plots, violin plots, and pair plots to visualize the data distributions and relationships.

2.6. Feature Extraction

A feature in machine learning and pattern recognition is a single, quantifiable attribute or aspect of a phenomena.(Bhadra et al., 2018) One of the first crucial steps in creating a potent machine learning based method for identifying AMP peptides and their functional types based on sequence information is to formulate the peptide samples with a mathematical expression that accurately captures the intrinsic correlation with the target that needs to be identified and using these as features for our model. A total of 69 features were generated, thus to capture the maximum information of the peptide. This features were generated using protpy and propy python libraries availiable online.

2.6.1. Amino Acid Composition (ACC) (20)

It is the percentage of each residue occurring in the sequence. This is the simplest set of features. However, if the peptide were represented only by the ACC model, all of its sequence-order effects would be eliminated, which might significantly reduce the prediction quality.

2.6.2. Pseudo Amino Composition (PseAA) (28)

The main idea behind it is to maintain the sequence-order information of a protein while representing it using a discrete model. In light of this, the PseAA composition of a protein, which differs from the standard AA composition and may contain sequence order or pattern information, is essentially a set of discrete numbers obtained from the amino acid sequence of the protein.

2.6.3. Physical Property Composition (21)

This is the part of CTD (Composition, Transition, Distribution) descriptor set. Features of amino acid sequences, such as hydrophobicity, normalized van der Waals volume, polarity, polarizability, charge, secondary structure, and solvent accessibility. The groupings of amino acids based on the physical properties is given in Table 5

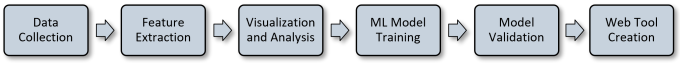

Figure 1. Pipeline of the project

2.7. Machine Learning Model

In our study, we employed four ML models, Support Vector Machine (SVM), Logistic Regression, Decision Tree, Catboost. These are widely renowned for their effectiveness in classification tasks. The SciKit library was used for model training (Pedregosa et al., 2011), while concurrent-features was used for multi-threading due to the large dataset size (Reitz Schlusser, 2016).

2.7.1. Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm, used for both classification and regression. Although it can handle regression problems, it is best suited for classification (Cortes Vapnik, 1995; Vapnik, 1998). The main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that separates data points of different classes. The hyperplane aims to maximize the margin between the closest points of different classes. The dimension of the hyperplane depends on the number of features. For two input features, the hyperplane is a line; for three input features, it becomes a 2-D plane. With more than three features, it becomes challenging to visualize. Data points on one side of the hyperplane are classified as AMPs, and those on the other side as Non-AMPs.

2.7.2. Logistic Regression

Logistic regression is widely used for binary classification problems due to its simplicity and effectiveness in estimating the probability of a given input belonging to a specific class (Hosmer, Lemeshow, Sturdivant, 2013). In this study, we used logistic regression with a sigmoid activation function to classify the input features into either 0 or 1. This enabled us to determine whether the sequences are AMP or Non-AMPs.

2.7.3. Decision Tree

Decision trees are popular for their interpretability and ability to handle both numerical and categorical data (Quinlan, 1986; Breiman et al., 1984). They create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The larger the number of features, the more relationships can be identified and branches can be created, resulting in greater efficiency. Generally, the entropy measure is used to make branches. Gini impurity is another option for deciding how to branch out (Breiman, 1996). Therefore, decision trees are among the best algorithms for classification, which we have used to classify AMPs from the rest.

2.7.4. CatBoost

CatBoost, short for Categorical Boosting, is an open-source boosting library developed by Yandex (Prokhorenkova et al., 2018). It is an advanced gradient boosting algorithm that handles categorical features efficiently and reduces the chances of overfitting, making it highly suitable for classification tasks. CatBoost is designed for use in problems like regression and classification that involve a very large number of independent features.

2.7.5. Dimensionality Reduction Techniques

To enhance the performance of our classification models and visualize the high-dimensional data, we employed two dimensionality reduction techniques:

1. Principal Component Analysis (PCA).
   Principal Component Analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of correlated variables into a set of uncorrelated variables (Jolliffe, 2002). PCA is the most widely used tool in exploratory data analysis and in machine learning for predictive models. Additionally, PCA can be used as a supervised learning technique to examine the interrelations among a set of variables (Abdi Williams, 2010). It is also known as a general factor analysis, where regression determines a line of best fit. The main goal of PCA is to reduce the dimensionality of a dataset while preserving the most important patterns or relationships between the variables without any prior knowledge of the target variables.
2. Uniform Manifold Approximation and Projection (UMAP).
   Dimensionality reduction is a powerful tool for machine learning practitioners to visualize and understand large, high-dimensional datasets. One of the most widely used techniques for visualization is t-SNE, but its performance suffers with large datasets and using it correctly can be challenging (van der Maaten Hinton, 2008). UMAP, developed by McInnes et al. (2018), offers several advantages over t-SNE, most notably increased speed and better preservation of the data's global structure. In this study, we explore the theory behind UMAP to better understand its workings, how to use it effectively, and how its performance compares with t-SNE.

Table 1. Model Performance with Different PCA Components

Modelcomponents	No PCA	5	10	20
SVM	88.45%	83.30%	86.30%	88.70%
Logistic regression	81.10%	70.10%	76.00%	79.67%
Decision tree	83.80%	77.60%	79.80%	80.65%
Catboost	89.50%	*	*	*

2.8. Web-tool

Streamlit, an open-source python framework was used to build the web-tool. The web-tool is hosted on the local PC. Hence video is provided in the Github repository showing its working

2.8.1. AMP Prediction Web tool

This Application predicts Anti Microbial Peptides (AMPs) using multiple machine learning models and Principal Component Analysis (PCA) models. The app allows users to input a peptide sequence and returns predictions from various models, along with an ensemble prediction.

1. Features of the web tool
   (a) User Input Sequence: Enter a peptide sequence of length greater than four.
   (b) Extracted Features: Displays the features extracted from the input sequence.
   (c) Individual Model Predictions: Shows predictions from individual models (SVM, Logistic Regression, Decision Tree, CatBoost).
   (d) PCA Model Predictions: Displays predictions from PCA-transformed models.
   (e) Ensemble Prediction: Provides a combined prediction based on all models.
   (f) Feature Information: Toggle button to display detailed feature information.

3. Results

3.1. Descriptive Statistics Results

The descriptive statistics provide an overview of the distribution and variability of each feature. Notably, features such as APAAC6 and E have higher means, indicating their potentially greater influence in distinguishing AMP samples. The skewness and kurtosis values suggest varying degrees of asymmetry and peakedness across features, with some features showing extreme values.

3.2. Visualisation Results

Refer to the Figures in Annexure

3.3. Inferential Results

Refer to the Figures in Annexure

4. Analysis

4.1. Analysis of Descriptive Statistics

4.1.1. Mean

The mean values provide the central tendency of the data for each feature. For instance, APAAC6 and E have high means (4.478175 and 4.591795, respectively), suggesting that these features generally have higher values in the dataset

4.1.2. Median

The median values are close to the mean values for most features, indicating a relatively symmetric distribution. However, features like APAAC6 and E have slightly higher medians, suggesting a slight right skew.

4.1.3. Mode

Many features have a mode of 0, indicating that zero values are common. This might reflect the presence of many samples where these particular features are absent or very low.

4.1.4. Skewness

Positive skewness values for most features indicate a right-skewed distribution, meaning the data has a long tail on the right side. Features like D (4.066732) and APAAC15 (4.380212) exhibit significant skewness. APAAC23 and APAAC24 have negative skewness, indicating a left-skewed distribution.

4.1.5. Kurtosis

High kurtosis values indicate a distribution with heavy tails or outliers. Features like D (72.435296) and APAAC15 (36.967099) have extremely high kurtosis, suggesting the presence of significant outliers or a heavy-tailed distribution.

4.1.6. Percentiles

The 25^{th}, 50^{th}, and 75^{th} percentiles provide insights into the spread and central tendency. For example, APAAC6 and E show substantial spread in their data values, with a wide range between the 25^{th} and 75^{th} percentiles.

4.2. Analysis of Inferential Statistics

4.2.1. Z-test

The Z-test results show extremely high Z-scores for all features, with p-values effectively at zero, indicating strong statistical significance. This suggests that the means of these features are significantly different between AMP and non-AMP groups.

4.2.2. T-test

The T-test results are consistent with the Z-test results, showing very high T-statistics and p-values at zero, confirming the significant differences in feature means between the two groups.

4.2.3. ANOVA

The F-statistics from ANOVA are extremely high with p-values at zero for all features, indicating significant differences in variance between AMP and non-AMP groups. Features like APAAC6 (12113.467668) and E (11530.462142) show particularly high F-statistics, highlighting their importance.

4.2.4. Mann-Whitney U Test

The Mann-Whitney U test results also indicate significant differences between the groups for all features, with p-values at zero. This non-parametric test supports the findings from the Z-test, T-test, and ANOVA, confirming that the distributions of the features are significantly different between AMP and non-AMP groups.

4.3. Analysis of Visual Statistics

The following visualizations provide further insights into the distribution and relationships of the top features selected by RFE.

4.3.1. Violin Plots

Violin plots visually compare the distribution of each feature across the AMP (1) and non-AMP (0) groups. These plots reveal clear differences in feature distributions, with most features showing distinct shapes and spreads. For instance, APAAC6 and E display wider and more dispersed distributions in the AMP group.[Refer Figure 1]

4.3.2. Pair Plots

Pair plots display scatter plots of each feature against every other feature, allowing the observation of potential correlations and relationships. Features like APAAC6 and E exhibit broader distributions, consistent with their higher means, confirming variability and potential correlations.[Refer Figure 2]

4.3.3. Jitter Plots

Jitter plots visualize the spread and overlap of feature values within each group. The spread of points indicates greater variability in the AMP group for most features, with minimal overlap between groups for key features like APAAC6 and E.[Refer Figure 3]

4.3.4. Histograms

Histograms show the frequency distribution of feature values. Features such as APAAC6 and E exhibit right-skewed distributions with long tails, confirming the presence of high values in some samples and consistent with their high skewness and kurtosis values.[Refer Figure 4 in Annexure]

4.3.5. Density Plots

Density plots combine histograms with a smoothed density curve, providing a clearer view of the distribution shape. Features like APAAC6 and E show clear peaks and long right tails, confirming the shapes observed in histograms and descriptive statistics.[Refer Figure 5 in Annexure]

4.3.6. Boxplots

Boxplots display the median, quartiles, and outliers for each feature. The numerous outliers, particularly for features like D and APAAC15, indicate significant variability, which is consistent with the high kurtosis values observed in the descriptive statistics.[Refer Figure 6 in Annexure]

4.3.7. Heatmap

The correlation heatmap shows the relationships between the top 10 RFE features, with correlation coefficients ranging from -1 to 1. Strong positive correlations are observed between APAAC23 and APAAC24 (0.85) and between APAAC6 and E (1), suggesting potential redundancy. Most feature pairs show low correlations, indicating they capture different information, which is beneficial for model robustness.

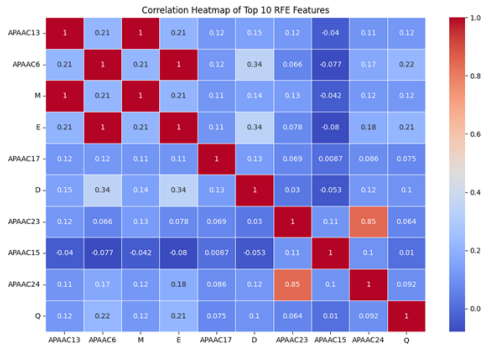

Figure 2. Correlation plot for the top ten features

5. Conclusion

Our study highlights the effectiveness of using multiple machine learning models, including SVM, Logistic Regression, Decision Tree, and CatBoost, for the classification of AMP and non-AMP sequences. Usage of extensive data helped the model to gain better accuracy than the existing tools. The incorporation of PCA and UMAP further enhanced our ability to handle and visualize the data. Each model demonstrated varying degrees of accuracy and predictive power in classifying AMP and non-AMP sequences. The combination of feature extraction from research papers and the application of robust classification models allowed us to achieve high accuracy in our predictions. These findings provide a strong foundation for future research in antimicrobial peptide discovery and the development of new antimicrobial therapies.

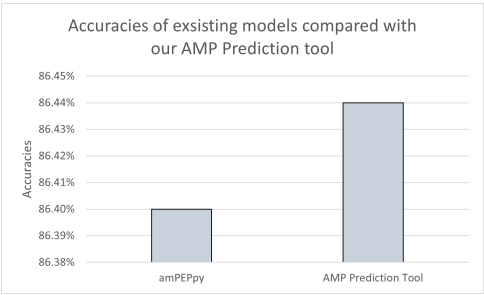

Figure 3. Comparison with existing tool

6. Discussion

Looking ahead, there are several avenues for further exploration and refinement of this work. Continual updates to the training dataset could help the model adapt to emerging trends and newly discovered peptides. Additionally, exploring the incorporation of additional features, could potentially enhance the model's predictive power.

7. References

1. Moretta, A., Scieuzo, C., Petrone, A. M., Salvia, R., Manniello, M. D., Franco, A., Lucchetti, D., Vassallo, A., Vogel, H., Sgambato, A., Falabella, P. (2021). Antimicrobial Peptides: A New Hope in Biomedical and Pharmaceutical Fields. Frontiers in cellular and infection microbiology, 11, 668632. https://doi.org/10.3389/fcimb.2021.668632
2. Cao, W., Zhou, Y., Ma, Y., Luo, Q., Wei, D. (2005). Expression and purification of antimicrobial peptide adenoregulin with C-amidated terminus in Escherichia coli. Protein Expression and Purification, 40(2), 404-410. https://doi.org/10.1016/j.pep.2004.12.007
3. Holaskova, E., Galuszka, P., Frebort, I., Oz, M. T. (2015). Antimicrobial peptide production and plant-based expression systems for medical and agricultural biotechnology. Biotechnology Advances, 33(6 Pt 2), 1005-1023. https://doi.org/10.1016/j.biotechadv.2015.03.007
4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830. https://jmlr.org/papers/v12/pedregosa11a.html
5. Xiao, X., Wang, P., Lin, W. Z., Jia, J. H., Chou, K. C. (2013). iAMP-2L: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types. Analytical biochemistry, 436(2), 168–177. https://doi.org/10.1016/j.ab.2013.01.019
6. Cortes, C., Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297. https://doi.org/10.1007/BF00994018
7. Vapnik, V. (1998). Statistical learning theory. Wiley.
8. Quinlan, J. R. (1986). Induction of decision trees. Machine Learning, 1(1), 81-106. https://doi.org/10.1007/BF00116251
9. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. arXiv preprint arXiv:1706.09516. https://doi.org/10.48550/arXiv.1706.09516
10. van der Maaten, L., Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9(Nov), 2579-2605. https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf
11. McInnes, L., Healy, J., Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for dimension reduction. arXiv preprint arXiv:1802.03426. https://doi.org/10.48550/arXiv.1802.03426
12. Abdi, H., Williams, L. J. (2010). Principal component analysis. Wiley Interdisciplinary Reviews: Computational Statistics, 2(4), 433-459. https://doi.org/10.1002/wics.101
13. Jolliffe, I. T. (2002). Principal Component Analysis (2nd ed.). Springer. https://doi.org/10.1007/b98835
14. Reitz, K., Schlusser, T. (2016). The Hitchhiker's Guide to Python: Best Practices for Development. O'Reilly Media.
15. Breiman, L., Friedman, J., Stone, C. J., Olshen, R. A. (1984). Classification and Regression Trees. Wadsworth International Group.
16. Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123-140. https://doi.org/10.1007/BF00058655

8. Additional Information

8.1. Annexure

Annexure is provided seperately. Annexure can be found in the github repository.

8.2. Github Repository

The link to the github repostitory can be found here: (It contains all the Plots, Tables, Web Tool [AMP Predicting web tool], Demonstration video, Model Codes, etc. in respective folders)
https://github.com/BT3041-TP-Report-Team-7/BT3041-TPA-AMP