Chapter 4

Identifying critical amino acids in the Antibacterial molecules using Explainable Artificial Intelligence

Antimicrobial resistance (AMR) is the outcome of several mutations in the genomes of bacteria that help them survive in the presence of antibiotics. Antibacterial peptides (ABPs) are innate immune system molecules that protect the host against infections. Compared to traditional antibacterial drugs, ABPs offer several advantages. As a result, they have recently attracted much attention as an alternative to the present antibiotics. A wide population of living things produces ABPs. To find novel ABPs from these natural resources, wet lab researchers do several tests in the lab, which involves a lot of time and money. As a result, to undertake a preliminary screening of natural sources to discover potential ABPs, the in-silico tool is needed. Thus, this chapter introduces an explainable artificial intelligence-based framework named XAI-INVENT, which has been made available as a web app at https://xai-invent.anvil.app/. This app will aid wet lab researchers in the fight against antibacterial resistance by accelerating the discovery of peptide-based antibacterial medications. The proposed framework not only discovers

the peptide-based antibacterial medications but also points out the critical amino acids that make a given peptide antibacterial, which enables wet-lab researchers to make informed decisions.

4.1 Introduction

Multiple mutations in the genome of microbes facilitate their survival in the presence of drugs, resulting in antimicrobial resistance (AMR). Antibacterial peptides (ABPs) are molecules of the innate immune system that protect the host from infections. ABPs provide many benefits over conventional antibacterial medications. They occur naturally, destroy bacteria in several ways (reducing their intracellular activity or, in most cases, disrupting their cell membrane [81, 82, 83]) and have few adverse effects. Because of this, ABPs have recently received a lot of interest as an alternative to currently available antibiotics [84]. As a consequence, peptide-based drugs account for 7% of all drugs authorised by the Food and Drug Administration (FDA) in the previous five years [85]. Antimicrobial peptides have been shown to have therapeutic uses by the authors in [86, 87, 88]. The FDA has authorised seven AMP-based drugs so far, and a thorough analysis of those drugs can be found in [89].

A diverse population of living organisms produces ABPs. Wet lab researchers conduct various trials in the lab to identify novel ABPs from these natural resources, which involves a lot of time and money. As a result, to undertake a preliminary screening of natural sources to discover potential ABPs, the *in-silico* tool is needed. In literature, several tools are available as web servers which wet-lab researchers can utilize for preliminary screening of natural sources. However, the aforementioned tools suffer from the limitation of being black boxes. As a result, they cannot provide information about the amino acids that play an essential role in the classification of a peptide as ABP/Non-ABP (known as critical amino acids).

For better understanding, we can relate the task of classifying the peptide as

ABP/Non-ABP to the sentiment analysis task, where given a sentence, the goal is to determine whether expressed opinion in the sentence is positive or negative. In the case of sentiment analysis, we can see that not all words contribute equally in determining the sentiment of a sentence (some words have more contribution, some words have less contribution and even some words have negative contribution). Consider the statement, "This film is not good". The overall sentiment of this statement is negative. The word that contributed most towards the prediction is "not", while the word "good" contributed against the prediction and the remaining words contributed little or nothing. In the sentiment analysis task, identifying critical words is possible for humans, but identifying critical amino acids in the case of peptides is not at all possible for a lay person. Therefore, there is a strong need for explainability in the task of peptide classification, which can be combined with domain expertise by wet-lab researchers while preparing ABPs in the lab.

Therefore, there is a need to develop an explainable *in-silico* tool that wet-lab researchers can utilize for preliminary screening of ABPs from the natural sources. Deep learning algorithms can automatically learn the optimal features from the data, thus reducing our reliance on domain experts. Moreover, the concept of ensemble learning, which aggregates the decisions of more than one deep learning models, can be applied to improve performance.

Taking into consideration the need to develop explainable *in-silico* tool and the advantage of deep learning and ensemble learning, we proposed XAI-INVENT, an explainable artificial intelligence-based framework for rapid discovery of novel peptide antibiotics. The proposed framework not only identifies potent ABPs from protein sequences but also provides information about the amino acids that play an essential role in classifying a peptide as ABP/Non-ABP. For building XAI-INVENT, first, the probability scores provided by the bidirectional gated recurrent unit (Bi-GRU), bidirectional long short-term memory (Bi-LSTM), bidirectional temporal convolutional network (Bi-

TCN) and 1D convolutional Neural Network (1DCNN) are fused using soft voting, and then the fused scores are utilised with local interpretable model-agnostic explanations (LIME) [90] for identifying critical amino acids.

To understand the contribution of ensemble learning, we also performed the ablation studies and found that there is a decrease in the performance on the removal of the ensemble learning technique.

To get more insights, we conducted a pilot study in which (i) We analysed all the peptides and determined amino acids which are present as critical amino acids in most of the ABPs / Non-ABPs. (Top five critical amino acids for ABPs are W, K, R, D, and M, while E, D, T, S, and N are the top five critical amino acids for Non-ABPs). (ii) We analysed some sample peptides that corresponded to each of the True Positive, True Negative, False Positive and False Negative categories and identified critical amino acids (see Section 4.3.7.2). (iii) We have also shown that the ABP/Non-ABP can be transformed into Non-ABP/ABP by removing the critical amino acid(s) (see Section 4.3.7.3).

We identified ABPs in bacteriocins obtained from ESKAPEE (Enterococcus faecium, Staphylococcus aureus, Klebsiella pneumoniae, Acinetobacter baumannii, Pseudomonas aeruginosa, Enterobacter spp and Escherichia coli) group of bacteria. We chose the ESKAPEE group of bacteria because they are highly threatening, drug-resistant, world health organization (WHO) critical priority I and II pathogens that have the inherent and acquired ability to develop resistance mechanisms in response to environmental threat rapidly and thus pose a significant threat among the drug-resistant microbes [91, 92].

The details of XAI-INVENT are shown in Figure 4.1, which contains three major components: (i) **Extraction of peptides from proteins:** Substrings of length $\in$[5,50] are generated to obtain peptides from the input protein sequence. (ii) **Ensemble of base classifiers:** Corresponding to each peptide, the probability scores provided by

Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN are fused using soft voting. (iii) **Explanation for the prediction using LIME:** Corresponding to each peptide, the fused score obtained from the ensemble classifier is provided to LIME, which provides a heatmap representing the critical amino acids in it. Wet lab researchers can combine the information from this heatmap with their domain expertise to make certain decisions. The main contributions of our chapter are summarized as follows: (i) We proposed an explainable artificial intelligence-based framework named XAI-INVENT for the rapid discovery of novel peptide antibiotics. (ii) To the best of our knowledge, proposed framework XAI-INVENT is the first of its kind, which not only identifies potent ABPs from protein sequences but also provides information about the amino acids that play an essential role in the classification of a peptide. (iii) Using XAI-INVENT, we identified ABPs in bacteriocin obtained from the ESKAPEE group of bacteria, which are highly threatening drug-resistant WHO priority I and II pathogens. (iv) To help researchers find new ABPs from protein sequences, the model has been set up as a web server and is freely accessible online at https://xai-invent.anvil.app/.

The rest of this chapter is structured as follows. The information regarding the dataset and the proposed framework is provided in Section 4.2. Section 4.3 presents the experimental configuration, performance metrics, assessment procedure, experiments and results. Identification of ABPs in bacteriocin obtained from the ESKAPEE group of bacteria is provided in Section 4.4. The details about the web server are provided in Section 4.5. The conclusion and future work are provided in Section 4.6.

4.2 Materials and methods

4.2.1 Dataset Collection

In this study, we prepared dataset D_s containing 20,648 peptides (ABPs: 10,324, non-ABPs: 10,324). These peptides are made up of natural amino acids and have length

$\in$ [5,50]. The ABPs were obtained from the DBAASP v3 database [93], whereas the non-ABPs were obtained from the Swiss-Prot database [45] using the similar approach used in the previous works [71, 94, 95, 70, 96, 97]. Specifically, these non-ABPs were generated from the reviewed, manually annotated proteins from the Swiss-Prot that did not contain any of the following keywords: antimicrobial, antibiotic, antioxidant, anticancer, defensin, bacteriocin, antiviral, anti-protist, antiendotoxin, antiparasitic, antitumor, antiprotozoal, anti-inflammatory, antibacterial, anti-gram +, anti-gram-, insecticidal, cytokine, antimalarial, anti-MRSA, antifungal, anti-diabetic, anti-TB, anti-HIV, antibiofilm, anti-toxin, secreted, excreted, effector.

We further divided the D_s into three sets, namely Training set (S^{Train}), Validation set (S^{Val}) and Test set (S^{Test}). S^{Train} contains $\approx$ 60% peptides and can be defined as follows:

$$\begin{aligned} & S^{Train} = S^{Train}_{ABPs} \cup S^{Train}_{non-ABPs} \\ & \text{where,} \\ & S^{Train}_{ABPs} \cap S^{Train}_{non-ABPs} = \emptyset \\ & |S^{Train}_{ABPs}| = 6195 \\ & |S^{Train}_{non-ABPs}| = 6195 \\ & |S^{Train}| = 12390 \end{aligned} \tag{4.1}$$

S^{Val} contains $\approx$ 20% peptides and can be defined as follows:

$$\begin{aligned} & S^{Val} = S^{Val}_{ABPs} \cup S^{Val}_{non-ABPs} \\ & \text{where,} \\ & S^{Val}_{ABPs} \cap S^{Val}_{non-ABPs} = \emptyset \\ & |S^{Val}_{ABPs}| = 2065 \\ & |S^{Val}_{non-ABPs}| = 2064 \\ & |S^{Val}| = 4129 \end{aligned} \tag{4.2}$$

S^{Test} contains the remaining 20% peptides and can be defined as follows:

$$
\begin{aligned}
& S^{Test} = S^{Test}_{ABPs} \cup S^{Test}_{non-ABPs} \\
& \text{where,} \\
& S^{Test}_{ABPs} \cap S^{Test}_{non-ABPs} = \emptyset \\
& |S^{Test}_{ABPs}| = 2064 \\
& |S^{Test}_{non-ABPs}| = 2065 \\
& |S^{Test}| = 4129
\end{aligned}
\tag{4.3}
$$

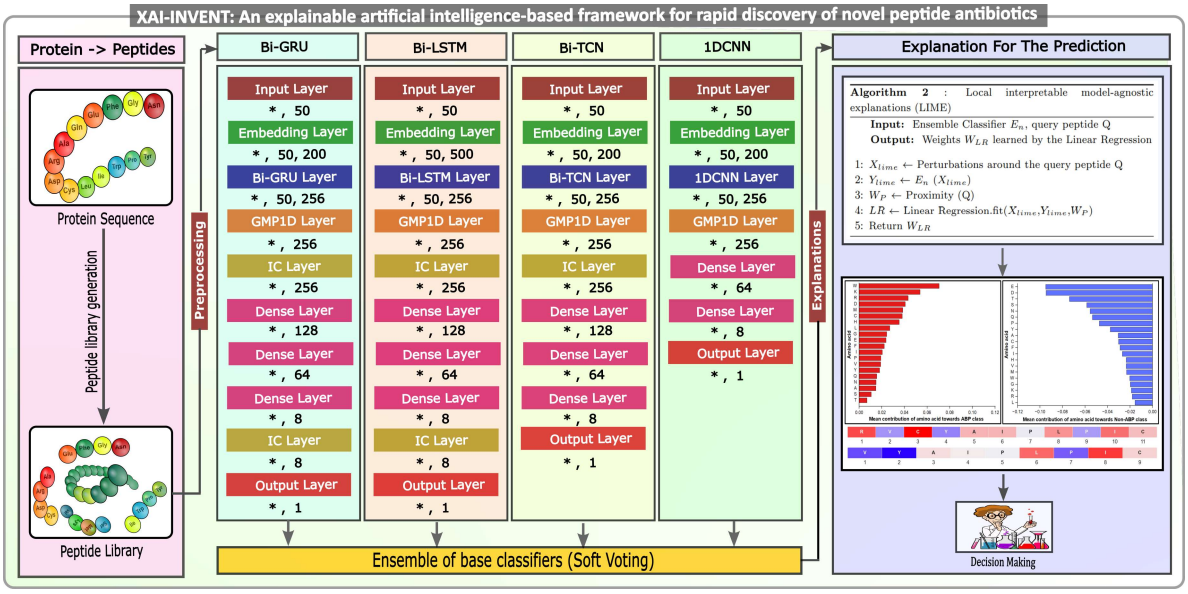


Figure 4.1: Proposed Framework

4.2.2 Proposed Framework

In the current work, we proposed XAI-INVENT, an explainable artificial intelligence-based framework for the rapid discovery of novel peptide antibiotics. For building XAI-INVENT, first, the probability scores provided by Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN deep learning algorithms are fused using a soft voting technique, and then the fused scores are utilised with LIME for identifying the critical amino acids.

We performed different experiments (related to the dimensionality of embedding layer, learning rate, number of dense layers, number of filters, size of filter, number of neurons, etc.) while choosing the architecture of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN. This section provides the optimal architecture obtained for Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN after different experiments. The size of input and output corresponding to each layer of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN is shown in Figure 4.1.

In Bi-GRU first layer is the Input layer, using which we feed the data (peptides). Before feeding peptides, we performed the following pre-processing steps: (i) Mapped 20 amino acids to numerical values (K: 1, L: 2, R: 3, A: 4, G: 5, I: 6, V: 7, S: 8, T: 9, F: 10, P: 11, N: 12, E: 13, C: 14, Q: 15, D: 16, Y: 17, W: 18, H: 19, M: 20). (ii) Made all peptides of uniform length 50 (= maximum length of peptide). After Input layer, Embedding layer is used, which learns the vector representation for every amino acid present in the dictionary. Like other layers Embedding layer can be initialised with random weights, or pre-trained embeddings can be used to initialise it. In the current study, we utilised the pre-trained embeddings because it realises the concept of transfer learning which helps this layer to learn better representation for each amino acid. These pretrained embeddings were obtained using the word2vec technique (for each amino acid, word2vec learns a 200-length vector representation). Following the Embedding layer, we used a 256-neuron Bi-GRU layer (128 neurons in each GRU layer). After Bi-GRU layer, we used GlobalMaxPooling1D (GMP1D) layer, which performs downsampling of input by taking the maximum value over the time dimension. After GMP1D layer, we used Independent component (IC) layer. The IC layer was presented for the first time in [46], where the authors merged batch normalisation and dropout. They ran multiple experiments and observed that IC layer leads to a stable training process, quicker convergence and improved generalisation. The output from the preceding IC layer is fed to dense layers D_1, D_2 and D_3 comprising 128, 64 and 8 neurons, respectively. Next, we used IC layer followed by output layer comprising one neuron.

In Bi-LSTM first layer is the Input layer using which we fed the peptides to the model. Following the input layer, Embedding layer is used, which utilises a 500-dimensional word2vec embedding vector. After Embedding layer, we utilised a 256-neuron Bi-LSTM layer (128 neurons in each LSTM layer). After Bi-LSTM layer, we used GMP1D layer, followed by the IC layer. The output from the preceding IC layer is fed to dense layers D_1 , D_2 and D_3 comprising 128, 64 and 8 neurons, respectively. Next, we used IC layer followed by output layer comprising one neuron.

In Bi-TCN first layer is the Input layer using which we fed the peptides to the model. Following the input layer, Embedding layer is used, which utilises a 200-dimensional word2vec embedding vector. After Embedding layer, we utilised a 256-neuron Bi-TCN layer (128 neurons in each TCN layer). After Bi-TCN layer, we used GMP1D layer, followed by IC layer. The output from the preceding IC layer is fed to dense layers D_1 , D_2 and D_3 comprising 128, 64 and 8 neurons, respectively. Next, we used an output layer comprising one neuron.

In 1DCNN first layer is the Input layer using which we fed the peptides to the model. Following the input layer, Embedding layer is used, which utilises a 200-dimensional word2vec embedding vector. After Embedding layer, we utilised a 256-neuron 1DCNN layer. After 1DCNN layer, we used GMP1D layer. The output from the preceding GMP1D layer is fed to dense layers D_1 and D_2 comprising 64 and 8 neurons, respectively. Next, we used an output layer comprising one neuron.

We used rectified linear unit (ReLU) activation function with each of Bi-GRU, Bi-LSTM, Bi-TCN, 1DCNN and dense layer.

The activation function used with the output layer of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN is sigmoid (σ), which can be defined as follows:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{4.4}$$

This output layer provides the probability of being ABP for each peptide (If probability

ABP > 0.5, then the peptide is predicted as ABP; else, predicted as non-ABP).

The network weights were updated using the Adam (Adaptive Moment Estimation) optimizer.

We employed the EarlyStopping technique to prevent overfitting. This technique monitors the validation loss and training loss and stops training the model when the validation loss stops decreasing.

Finally, using the soft voting approach, we integrated the predictions from the base classifiers BiGRU, BiLSTM, BiTCN, and 1DCNN. The algorithm for the soft voting technique is presented in Algorithm 1. The net probability P is determined by averaging the predictions made by the base classifiers BiGRU, BiLSTM, BiTCN, and 1DCNN, as demonstrated in Algorithm 1. Based on the calculated probability value, the class C is determined. The class is 1 if the probability is greater than 0.5; otherwise, it is 0. The LIME algorithm (Section 4.2.3, Algorithm 2) is then utilised with this fused score for determining the critical amino acids.

Algorithm 1 : Ensemble of base classifiers

Input: Base classifiers (Bi-GRU, Bi-LSTM, Bi-TCN, 1DCNN), query peptide Q.
Output: Class C (= 0 for Non-ABP and 1 for ABP) and
Probability P ($\in$ [0,1]) corresponding to ABP
for query peptide Q

/* Probability values $(P_1, .., P_4)$ are provided by Models (Bi-GRU, Bi-LSTM, Bi-TCN, 1DCNN) for query sequence Q */

$P_1 \leftarrow$ Bi-GRU (Q)
$P_2 \leftarrow$ Bi-LSTM (Q)
$P_3 \leftarrow$ Bi-TCN (Q)
$P_4 \leftarrow$ 1DCNN (Q)
P $\leftarrow \frac{(P_1+P_2+P_3+p_4)}{4}$
if $P > 0.5$ **then**
    C$\leftarrow$1
else
    C$\leftarrow$0
end if
Return C, P

4.2.3 Local interpretable model-agnostic explanations (LIME)

LIME stands for local interpretable model-agnostic explanations. It is a technique that faithfully explains any black-box model's predictions by approximating it locally with an interpretable model. To do so, LIME first generates a new dataset by executing random perturbations around the query point. After this, predictions are made for the newly created dataset using the black-box model (for which interpretability is required). In addition to prediction, the newly created samples are weighted according to their closeness to the query point. Then an interpretable linear model is trained, and the weights learned are used to unbox the black-box model for the query point. LIME is explained as Algorithm 2. LIME associates a weight value w ∈[-ve, 0, +ve] corresponding to each amino acid of the peptide,. If w is +ve, then it means that amino acid is contributing towards ABP. If w is -ve, then it means that amino acid is contributing towards Non-ABP. For better visualisation, these weights are shown as a heatmap where +ve values are represented by red colour, and -ve values are represented by blue colour. If the query peptide is predicted as ABP, then the critical amino acids, in that case, are amino acids which are coloured red, whereas if the query peptide is predicted as Non-ABP, then the critical amino acids, in that case, are amino acids which are coloured as blue.

Algorithm 2 : Local interpretable model-agnostic explanations (LIME)

Input: Ensemble Classifier E_n, query peptide Q
Output: Weights W_{LR} learned by the Linear Regression

1: $X_{lime} \leftarrow$ Perturbations around the query peptide Q
2: $Y_{lime} \leftarrow E_n\ (X_{lime})$
3: $W_P \leftarrow$ Proximity (Q)
4: $LR \leftarrow$ Linear Regression.fit(X_{lime},Y_{lime},W_P)
5: Return W_{LR}

4.3 Experiments and Results

This section briefly describes the experimental configuration, performance metrics, assessment procedure used, results obtained from the proposed framework. We have also conducted the ablation studies to understand the contribution of ensemble technique in the proposed framework. This section also provides the results obtained from these ablation studies. Additionally, the outcomes of different experiments carried out using LIME are also presented in this section.

4.3.1 Experimental Configuration

The deep learning algorithms were implemented using Keras deep learning library [48] with Tensorflow as the backend. All experiments were carried out on a CPU compute node configured with a 2.4 GHz Intel-Xeon Skylake 6148 processor and 192 GB RAM.

4.3.2 Performance Metrics

To access the performance, we used various performance metrics namely Accuracy (A_{cc}), Sensitivity(S_n), Precision(P_r), F1-Score (F_s), Specificity (S_p), Matthews correlation coefficient (MCC).

4.3.3 Assessment Procedure

Dataset D_s containing 10,324 ABPs and 10,324 non-ABPs was divided into three sets, namely Training set (S^{Train}), Validation set (S^{Val}), and Test set (S^{Test}). S^{Train} contains 60% (12390) peptides . S^{Val} contains 20% (4129) peptides, and S^{Test} contains the remaining 20% (4129) peptides. We used S^{Train} for training, S^{Val} for hyperparameter tuning and identifying the best framework (the model obtained from the best framework is termed as XAI-INVENT) among the frameworks available from different methods. S^{Test} was not utilised for hyperparameter tuning or training. Therefore, it was used test the generalization performance of our proposed model.

4.3.4 Result obtained from the proposed Model

We proposed a model E_n that is an ensemble of various deep learning algorithms. The results obtained for S^{Val} using the E_n are provided in Table 4.1. As can be seen from this Table, E_n obtained the A_{cc}, S_n, P_r, F_s, S_p and MCC values $\approx$ 96, 95, 97, 96, 97, and 93 % respectively. Before arriving at the proposed framework, we performed ablation studies . These ablation studies are evaluated on the validation data, and their findings are presented in the subsequent Section.

Table 4.1: Result obtained from the ensemble classifier constructed by combining Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms.

Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	MCC (*100)
LGTC-SV	96.29	95.39	97.14	96.26	97.18	92.60

4.3.5 Ablation Studies

The concept of ensemble technique (Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN deep learning algorithms are combined using the soft voting ensemble technique) is utilized in the proposed framework. Therefore, before finalizing the proposed framework to understand the role of ensemble learning in it , we performed the following ablation experiments: (i) We build classifiers using Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN algorithms taking one at a time (ii) Build ensemble classifiers by combining 2 of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms (iii) We build ensemble classifiers by combining 3 of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms (iv) We build ensemble classifiers by combing all Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms. For conducting the aforementioned experiments S^{Train} was used as training data, and S^{Val} was used as validation data.

Table 4.2: Result obtained from Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN .

Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	MCC (*100)
Bi-GRU	95.37	94.91	95.79	95.35	95.83	90.75
Bi-LSTM	95.22	94.62	95.78	95.20	95.83	90.46
Bi-TCN	95.64	95.69	95.59	95.64	95.59	91.28
1DCNN	95.03	95.10	94.97	95.03	94.96	90.07

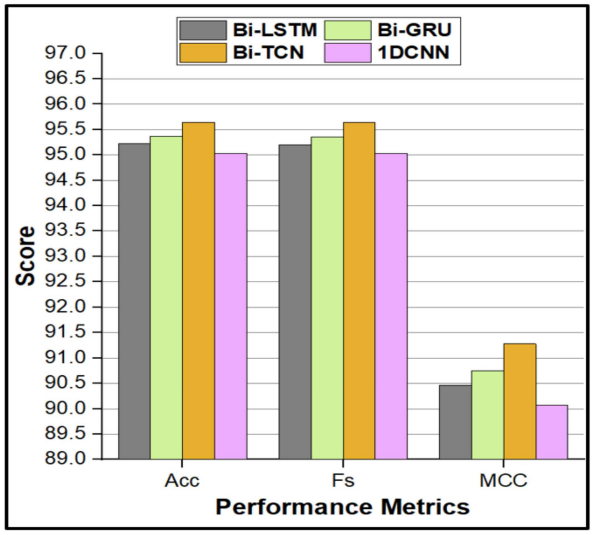


Figure 4.2: Comparison of results obtained from Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN taking one at a time.

Table 4.3: Results obtained from the ensemble classifiers constructed by combining any two of the Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms.

Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	MCC (*100)
LG-SV	95.44	94.76	96.07	95.41	96.12	90.90
LT-SV	96.00	95.54	96.43	95.98	96.46	92.01
LC-SV	95.64	94.72	96.49	95.60	96.56	91.29
GT-SV	96.02	95.73	96.29	96.01	96.31	92.05
GC-SV	95.81	95.35	96.23	95.79	96.26	91.62
TC-SV	95.49	95.35	95.62	95.48	95.63	90.99

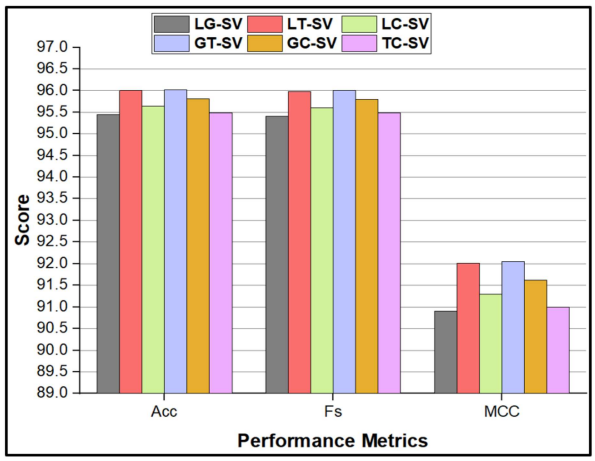


Figure 4.3: Comparison of results obtained by combining any two of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN.

Table 4.4: Result obtained from the ensemble classifiers constructed by combining any three of the Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms.

Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	MCC (*100)
LGT-SV	96.19	95.49	96.85	96.17	96.89	92.40
LGC-SV	96.24	95.35	97.09	96.21	97.14	92.50
LTC-SV	95.90	95.30	96.47	95.88	96.51	91.82
GTC-SV	95.93	95.35	96.47	95.90	96.51	91.86

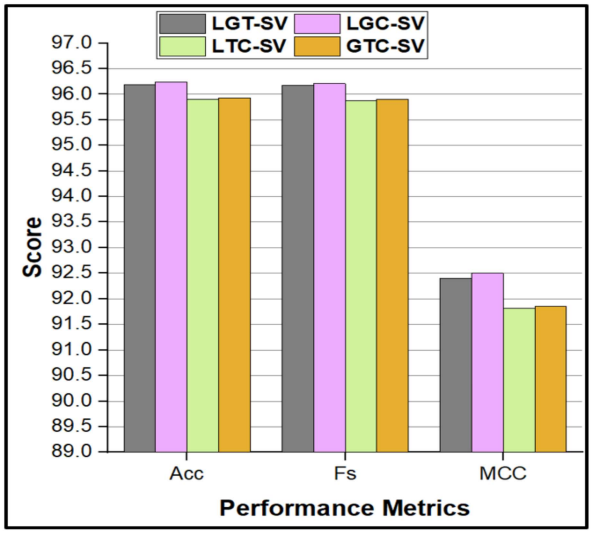


Figure 4.4: Comparison of results obtained by combining any three of Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN.

The results obtained from Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN taking one at a time are shown in Table 4.2 and their comparison in terms of composite metrics is provided in Figure 4.2. As can be seen from this figure, Bi-TCN performed the best.

The results obtained by combining any 2 of Bi-GRU, Bi-LSTM, Bi-TCN, 1DCNN algorithms using soft voting (Where LG-SV, LT-SV, LC-SV, GT-SV, GC-SV, TC-SV represent soft voting used with (Bi-LSTM and Bi-GRU), (Bi-LSTM and Bi-TCN), (Bi-LSTM and 1DCNN), (Bi-GRU and Bi-TCN), (Bi-GRU and 1DCNN), (Bi-TCN and 1DCNN), respectively) are shown in Table 4.3 and their comparison in terms of composite metrics is provided in Figure 4.3. As can be seen from this figure, GT-SV performed the best.

The results obtained by combining any 3 of Bi-GRU, Bi-LSTM, Bi-TCN, and 1DCNN algorithms using soft voting (Where LGT-SV, LGC-SV, LTC-SV, GTC-SV represent soft voting used with (Bi-LSTM, Bi-GRU and Bi-TCN), (Bi-LSTM, Bi-GRU and 1DCNN), (Bi-LSTM, Bi-TCN and 1DCNN), (Bi-GRU, Bi-TCN and 1DCNN), respectively) are shown in Table 4.4 and their comparison in terms of composite metrics are provided in Figure 4.4. As can be seen from this figure, LGC-SV performed the best.

The results obtained by combining all Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN algorithms using soft voting are shown in Table 4.1 (LGTC-SV).

The comparison of the best results obtained from all the cases (Bi-TCN, GT-SV, LGC-SV and LGTC-SV) is given in Figure 4.5. As can be seen from this figure, LGTC-SV performed the best.

Table 4.5: Results obtained from the proposed model LGTC-SV on test data

Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	MCC (*100)
Bi-GRU	95.08	94.52	95.59	95.05	95.64	90.17
Bi-LSTM	95.34	95.00	95.65	95.33	95.69	90.70
Bi-TCN	95.54	95.63	95.45	95.54	95.44	91.08
1DCNN	95.34	95.00	95.65	95.33	95.69	90.70
LGTC-SV	96.17	95.63	96.66	96.15	96.70	92.35

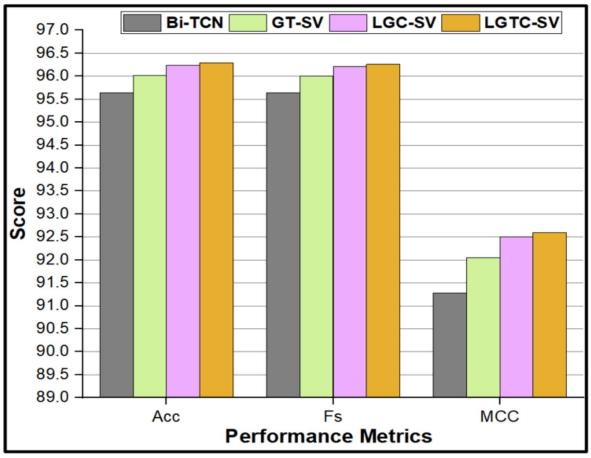


Figure 4.5: Comparison of the best results obtained for all the cases.

4.3.6 Performance of the proposed model on test data

By performing ablation studies, we found that the proposed framework is better than the alternatives. Further, we evaluated the generalization performance of E_n on S^{Test}. The results obtained by E_n for S^{Test} are shown in Table 4.5. The performance obtained by E_n for S^{Test} is almost similar to that obtained for S^{Val}, as can be seen from Tables 4.5 and 4.1. The reason for the better performance of E_n and similar performance attained by E_n for both S^{Test} and S^{Val} is its stability (The deep learning algorithms which are utilized in E_n are heterogeneous. As a result, the peptide misclassified by one model got correctly classified by another. Therefore, the aggregate result of multiple models is less noisy than the individual model, leading to model stability and robustness).

4.3.7 Results from various experiments conducted using LIME

4.3.7.1 Predictions based on LIME for peptides in D_s

We applied LIME to peptides in D_s and identified amino acids which are present as critical amino acids in most of (i) ABPs: obtained by calculating the mean of positive weights attained by each amino acid (see Figure 4.6). (ii) Non-ABPs: obtained by calculating the mean of negative weights attained by each amino acid (see Figure 4.7).

Top five critical amino acids for ABPs are W, K, R, D, and M, while E, D, T, S, and N are the top five critical amino acids for Non-ABPs. We came across that certain amino acids are working as critical amino acids in both ABPs and Non-ABPs (like D is present in the top 5 critical amino acids for both ABPs and Non-ABPs). The reason for this can be understood by considering the following two sentences from the sentiment analysis task: (i) This movie is not bad (ii) This movie is not good. The word " not " contributes towards positive sentiment in (i), whereas it contributes towards negative sentiment in (ii).

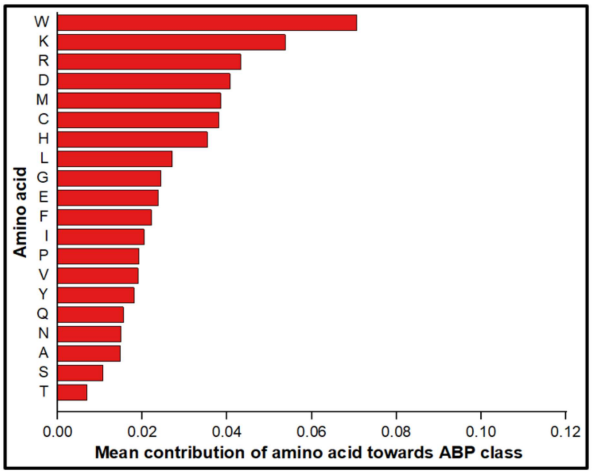


Figure 4.6: Mean contribution of amino acid towards ABP class.

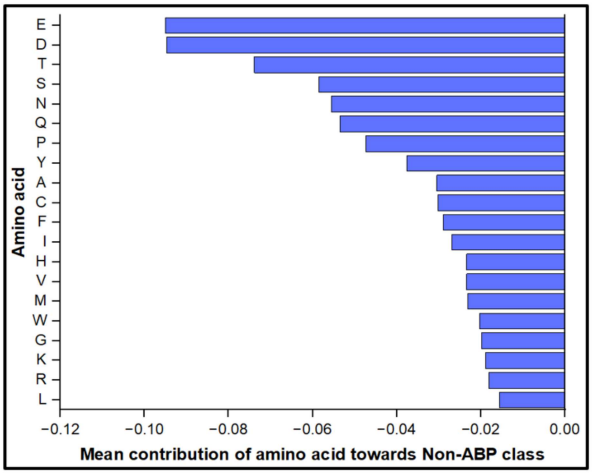


Figure 4.7: Mean contribution of amino acid towards Non-ABP class.

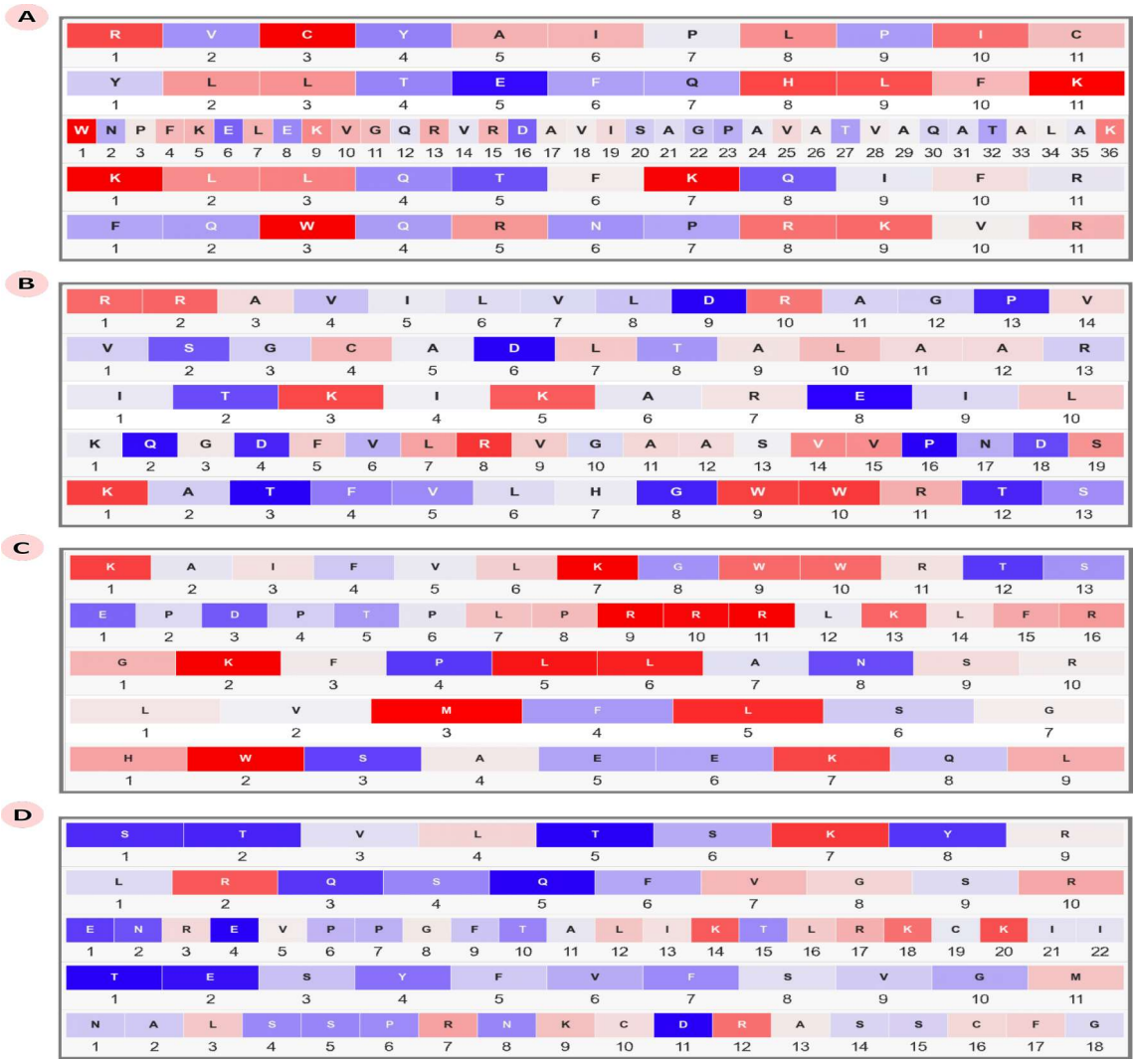


Figure 4.8: Predictions made by LIME for sample peptides from each of (A) True Positive, (B) True Negative, (C) False Positive and (D) False Negative categories

4.3.7.2 LIME for representative peptides

The predictions made by LIME for sample peptides from each of the True Positive, True Negative, False Positive and False Negative categories are shown in Figure 4.8(A), 4.8(B), 4.8(C) and 4.8(D), respectively. From Figure 4.8(A), we can see that the weightage of red-coloured amino acids is high compared to blue-coloured amino acids. Therefore model has classified the peptide as ABP. Similarly, Figure 4.8(C) shows the high weightage of red-coloured amino acids compared to blue-coloured amino acids, leading

to the wrong prediction of Non-ABP as ABP.

From Figure 4.8(B), we can see that the weightage of blue-coloured amino acids is high compared to red-coloured amino acids. Therefore model has classified the peptide as Non-ABP. Similarly, Figure 4.8(D) shows the high weightage of blue-coloured amino acids compared to red-coloured amino acids, leading to the wrong prediction of ABP as Non-ABP.

4.3.7.3 Transformation of ABP/Non-ABP to Non-ABP/ABP

By eliminating red-coloured amino acid(s) (responsible for moving the peptide prediction towards ABP), the ABP can be converted to Non-ABP. Figure 4.8(A) shows a heatmap of predictions made by LIME for sample ABP, and Figure 4.9(A) shows a heatmap of predictions made by LIME for the same ABP after removing critical amino acid (s). The critical amino acids removed from 1st peptide are R, and C, which are present at positions 1, and 3, respectively. The critical amino acids removed from 2nd peptide are H, L, and K, which are present at positions 8, 9, and 11, respectively. The critical amino acid removed from the 3rd peptide is W which is present at position 1. The critical amino acids removed from the 4th peptide are K, and K, which are present at positions 1, and 7, respectively. The critical amino acid removed from the 5th peptide is W which is present at position 3. The probability values obtained from the ensemble classifier E_n for these five peptides before and after removing the aforementioned critical amino acid(s) are (0.98, 0.85, 0.93, 0.94, 0.98) and (0.39, 0.06, 0.40, 0.13, 0.27), respectively. These probability scores (for each of the five peptides, initially probability score was greater than 0.5, and after the removal of critical amino acids, the probability score for each of the peptide became more than 0.5) shows the transformation of ABPs to Non-ABPs on the removal of critical amino acid(s).

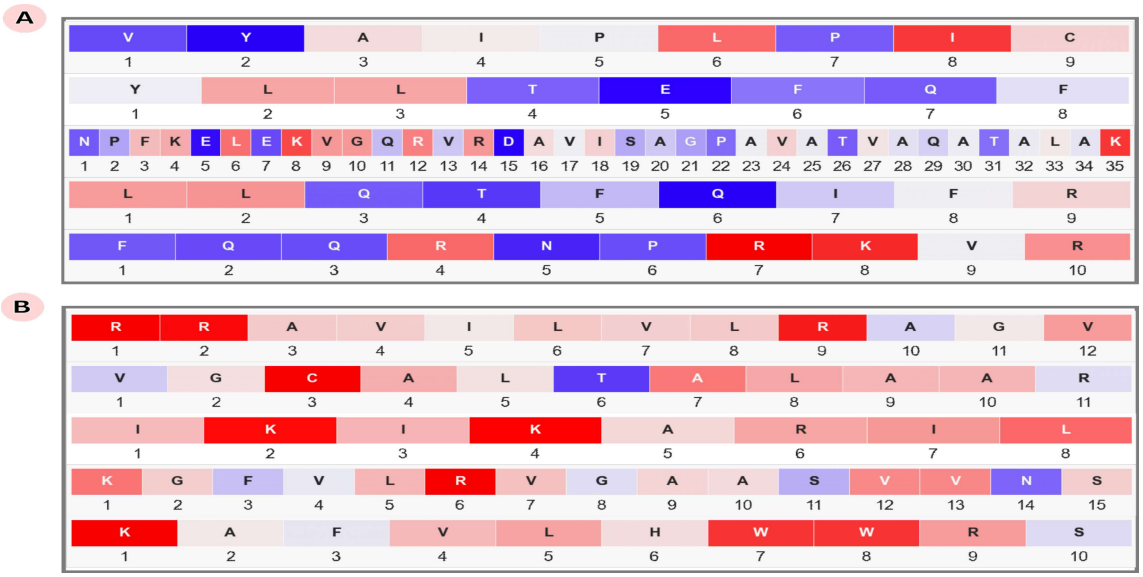


Figure 4.9: Conversion of ABP to Non-ABP after removing critical amino acid (s) (B) Conversion of Non-ABP to ABP after removing critical amino acid (s).

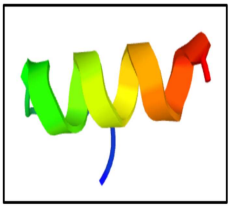
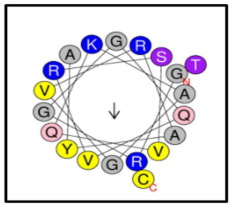
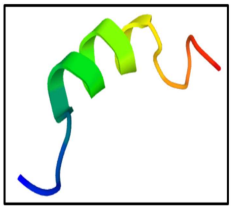
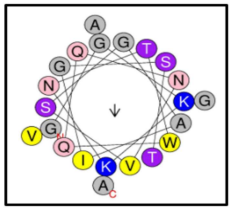
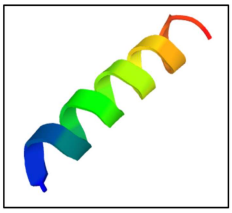
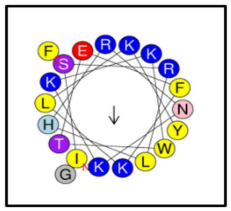
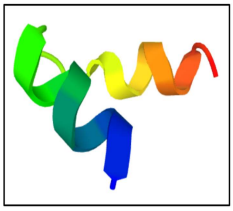
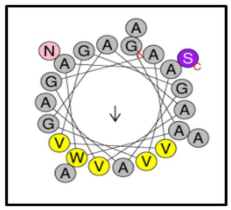
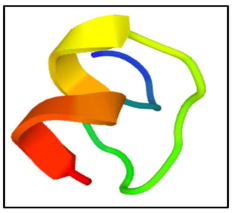
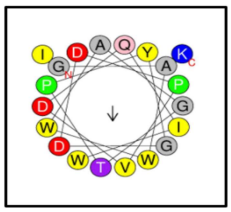
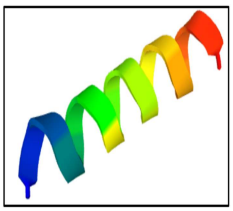
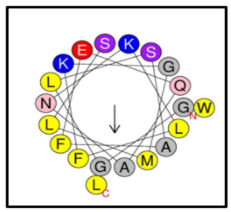
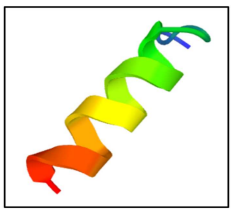
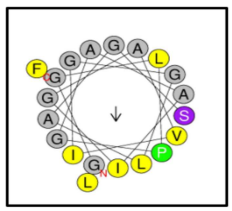
Similarly, by eliminating the blue-coloured amino acid(s) (responsible for moving the peptide prediction towards Non-ABP), the Non-ABP can be converted to ABP. Figure 4.8(B) shows a heatmap of predictions made by LIME for sample Non-ABP, and Figure 4.9(B) shows a heatmap of predictions made by LIME for the same Non-ABP after removing critical amino acid (s). The critical amino acids removed from 1st peptide are D and P, which are present at positions 9 and 13, respectively. The critical amino acids removed from the 2nd peptide are S and D which are present at positions 2 and 6, respectively. The critical amino acids removed from the 3rd peptide are T and E which are present at positions 2 and 8, respectively. The critical amino acids removed from the 4th peptide are Q, D, P and D, which are present at positions 2, 4, 16 and 18, respectively. The critical amino acids removed from the 5th peptides are T, G and T, which are present at positions 3, 8 and 12, respectively. The probability values obtained from the ensemble classifier E_n for these five peptides before and after removing the aforementioned critical amino acid(s) are (0.31, 0.16, 0.36, 0.25, 0.08) and (0.97, 0.93, 0.99, 0.95, 0.94), respectively. These probability scores (for each of the five

peptides, initially probability score was less than 0.5, and after the removal of critical amino acids, the probability score for each of the peptide became more than 0.5) shows the transformation of Non-ABPs to ABPs on the removal of critical amino acid(s).

4.4 Prediction of ABPs

Corresponding to bacteriocin obtained for each bacteria from the ESKAPEE group (details are provided in Table 4.6), we proposed one ABP using the following steps: (i) Utilised XAI-INVENT, which provided probability score and heatmap corresponding to each ABP identified from the protein sequences. (ii) Considered the ABPs from the previous step which have length $\in$ [20, 30] and then sorted them in decreasing order of probability (iii) Obtained the helical wheel representation (generated using the HeliQuest web server [98] available at https://heliquest.ipmc.cnrs.fr/). This helical wheel presents the alpha-helical property of a peptide. In the helical wheel, hydrophobic and hydrophilic residues are arranged in two different planes. As a result, the helical wheel possesses a hydrophobic moment, which is shown as an arrow inside the helical wheel. A large hydrophobic moment value means that the helix is amphipathic and, therefore, more likely to adopt a helical structure in solution, which will be beneficial for the antibacterial activity of the peptide. (iv) Obtained the structure (generated using PEP-FOLD3 web server [99] available at https://bioserv.rpbs.univ-paris-diderot.fr/services/PEP-FOLD3/) for the ABPs obtained from (ii). (v) Identified the topmost ABP, which displays good helical wheel representation and structure both. (vi) Proved the novelty of the ABP obtained from the previous step. For this, we first performed the protein-protein BLAST (Blastp) of the peptide identified in (v) with respect to the available ABPs. Then we used the commonly used alignment tool accessible at https://www.uniprot.org/align/ to align the highest-ranked ABP produced by Blastp with the corresponding query ABP. We noticed that 100 % alignment is not there, confirming the uniqueness of our peptide.

Table 4.6: Antibacterial peptides derived from bacteriocin of the ESKAPEE group of bacteria and proposed for wet-lab synthesis and experimentation.

Antibacterial peptides	Predicted Structure	Helical Wheel
Bacteriocin: Enterococcus faecium [OTO49785.1] | Proposed ABP: GRGGQVRSVQKAGVRAYATC Source of Existing ABP: Synthetic | Existing ABP: ACYCRIPACIAGERRYATCIYQGRLWAFCC		
Bacteriocin: Staphylococcus aureus [OIV48016.1] | Proposed ABP: GGKKNTWQQNVSGAIGSTVAGA Source of Existing ABP: Hymenochirus boettgeri | Existing ABP: IKVPAIVKDTQKKVAKGVISTVADALSKS		
Bacteriocin: Klebsiella pneumoniae [WP_109223240.1] | Proposed ABP: ISRLHRNKKKWTEFKLKYGF Source of Existing ABP: Synthetic | Existing ABP: GLLKKWLKKWKEFKRIVGY		
Bacteriocin: Acinetobacter baumannii [TPV09527.1] | Proposed ABP: GAWAAVGAAVGAVVGGAAAAANS Source of Existing ABP: Synthetic | Existing ABP: KKAAAVAAAAAVAAWAAVAAAKKKK		
Bacteriocin: Pseudomonas aeruginosa [AVZ37966.1] | Proposed ABP: GAWWAGTPYGDDPVDQIWIK Source of Existing ABP: Synthetic | Existing ABP: ILPWKWAWWAWRR		
Bacteriocin: Enterobacter quasiroggenkampii [WP_235404758.1] | Proposed ABP: GGLSAFEQANKLFKGMLSWL Source of Existing ABP: Synthetic | Existing ABP: KKLFKKILKWL		
Bacteriocin: Escherichia coli [UJY38014.1] | Proposed ABP: GGLPGAAIGAVIGGLAGSLF Source of Existing ABP: Bombina maxima | Existing ABP: ILGPVIKTIGGVIGGLLKNL		

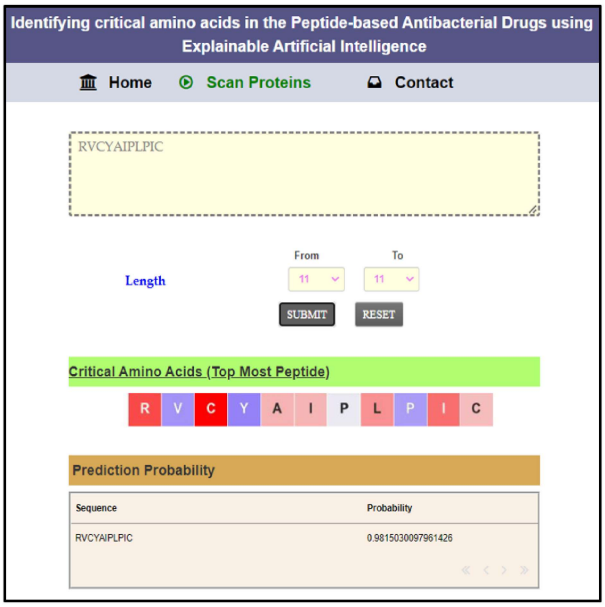


Figure 4.10: Identify critical amino acids in ABP.

4.5 Web Server

To aid wet-lab researchers, we made XAI-INVENT available as a web app at https://xai-invent.anvil.app/. The web server accepts as input the protein sequence and length $\in$ [5, 50] (on the basis of which the peptide library is generated) and returns ABPs with probability scores and a heatmap produced from LIME for the top peptide. Figure 4.10 shows the output from the web server for sample bacteriocin protein of Enterococcus faecium.

4.6 Summary

Traditional antibiotics can no longer effectively treat pathogens. Thus scientists are searching for new therapeutic alternatives, such ABPs. ABPs are found in all multicellular animals and serve as the first line of defence against harmful bacteria. However, finding ABPs from natural sources is time-consuming and expensive. As a result, wet-lab researchers use *in-silico* tools to detect the probable ABPs from natural sources. The existing *in-silico* tools available for this purpose suffer from the limitation of being

black boxes. Therefore, they are not able to provide information about the amino acids that play an essential role in the classification of the peptide as ABP/Non-ABP (known as critical amino acids). Thus, in the present work, we developed XAI-INVENT, an explainable artificial intelligence-based framework for the rapid discovery of novel peptide antibiotics. The proposed framework not only identifies potent ABPs from protein sequences but also provides information about the amino acids that play an essential role in the classification of the peptide. For building XAI-INVENT, first, the probability scores provided by the Bi-GRU, Bi-LSTM, Bi-TCN and 1DCNN are fused using a soft voting technique, and then the fused scores are provided to the LIME algorithm to identify the critical amino acids. Utilising the proposed model, we identified ABPs in the bacteriocin of the ESKAPEE group of bacteria and proposed peptides for wet lab synthesis and experimentation. The identified ABPs can be synthesised as either linear or branched peptides, and their antibacterial activity can be evaluated by similar techniques used by [53, 54]. To aid wet-lab researchers, we have also deployed XAI-INVENT as a web app and made it freely available at https://xai-invent.anvil.app/, which can be utilised to predict novel ABPs from protein sequences. To the best of our knowledge, the proposed framework XAI-INVENT is the first of its kind since it not only identifies ABPs but also provides information about critical amino acids, which will benefit wet-lab researchers in better comprehending the model's outcome.

The proposed model is trained on ABP sequences, with no bacteria-specific information provided during the training process. As a result, regardless of the bacteria it can target, the proposed model will produce ABP/non-ABP output (i.e., the predicted peptide will function as an ABP or not). The present work may be expanded in the future to create a two-stage cascade model. The first step may provide ABP/non-ABP results, while the second stage may reveal the bacteria (ESKAPEE group of bacteria) that the ABP found in the first stage may be targeting.