Chapter 2

Using the stacked ensemble technique for the prediction and discovery of antibacterial peptides

The rise in antimicrobial resistance (AMR) among disease-causing pathogens has rendered conventional antibiotics ineffective. Consequently, researchers are seeking alternative therapeutic options such as antibacterial peptides (ABPs), which have proved useful against multi-drug resistant (MDR) microorganisms. The process of identifying new ABPs through wet lab trials is both resource-intensive and time-consuming. Numerous machine learning models have been proposed in the literature for the *in silico* identification of novel ABPs in protein chains. However, there exists an opportunity to build a highly accurate and precise model for classifying and identifying ABPs. This study introduces StaBle-ABPpred, a machine and deep learning (ML and DL)-based framework built using the stacked ensemble technique. At the base level, it uses the bidirectional long-short-term memory (biLSTM) algorithm along with the attention mechanism. At the meta level, it uses a group of ML algorithms, namely random forest, gradient boosting, and logistic regression. The objective of this classifier is to

accurately classify peptides as either antibacterial or non-antibacterial. Also, since algorithms like biLSTM take a lot of time to train, the stacked ensembling with the ML techniques helps reduce this overhead. The performance of this model was compared with the state-of-the-art classifiers and also analyzed statistically. The findings state that our model outperforms the existing classifiers. In addition, a web application has been created and deployed at `https://stable-abppred.anvil.app`. This application serves the purpose of detecting previously undiscovered ABPs within protein sequences. Utilizing this application, some novel ABPs were discovered in the proteins of the Streptococcus phage T12. The identified ABPs had substantial similarities with annotated antimicrobial peptides (AMPs). Hence, they can be chemically synthesized and experimentally validated for their efficacy against various bacterial strains.

2.1 Introduction

The multi-drug resistant (MDR) pathogens have acquired resistance to several antibiotics, resulting in reduced efficacy of these drugs [22]. There is a growing demand to advance the development of novel and enhanced antibiotics that specifically target these microorganisms. While certain antibiotics have demonstrated potential, it is important to acknowledge that they may induce temporary or permanent side effects in the human body. Therefore, it is imperative to explore the development of a novel class of pharmaceuticals utilizing antimicrobial peptides (AMPs) to manage the rise of antimicrobial resistance (AMR) effectively. AMPs are a crucial component of an organism's innate defense system [23]. They impede microbial growth and dissemination through disruption of their cell membrane or interference with their intracellular activity. One of the most notable characteristics of AMPs is their low toxicity and limited or no side effects. As of the present time, the US Food and Drug Administration (FDA) has granted approval to a total of seven medications that are based on AMPs [24].

The AMPs are broadly divided into four classes: antibacterial peptides (ABPs), an-

tiviral peptides (AVPs), antifungal peptides (AFPs), and antiparasitic peptides (APPs). The discovery of new ABPs received a boost during the SARS-CoV-2 pandemic [1]. However, intense hit-and-trial-based wet lab procedures were very costly and time-consuming. To solve this, the ML and DL experts built ABP classifiers such as AntiBP [2], Deep-ABPpred [3], iAMPpred [4], CAMP [7], ClassAMP [11], BIPEP [25], IAMPE [26], ADAM [27], etc. Some models employ DL techniques like convolutional neural networks (CNNs), recurrent neural networks (RNNs), long-term memory networks (LSTMs), etc., that take the amino acid (AA) chains constituting a peptide sequence Pep, as shown in Eq. 2.1, as input. Each standard AA is represented by a letter (except B, J, O, U, X, Z). E.g., AMPScanner v2 [28] uses CNNs and RNNs, Deep-ABPpred [3], which uses bidirectional LSTMs (biLSTMs), etc. These models have not fully utilized the advantages of the attention mechanism [29] in capturing dependencies between amino acid residues separated by a long distance. On the other hand, some other models use hand-crafted physicochemical properties (e.g., molecular weight, net charge, isoelectric point, GRAVY index), compositional properties (e.g., amino acid composition (AAC), Pseudo-AAC (Pse-AAC), Amphiphilic Pse-AAC (APAAC)), and structural properties (e.g., beta-sheet propensity (BSP), beta-turn propensity (BTP), and alpha-helix propensity (AHP)) of peptides. Such models use classic ML algorithms like support vector machine (SVM), random forest (RF), extreme gradient boosting (XGBoost), etc. E.g., AntiBP2 [30] used artificial neural networks (ANNs), CAMP [7] used RF, SVM, and XGB, and ClassAMP [11] employed RF and SVM. The selection of relevant features to train these algorithms is a time-consuming process that requires domain expertise as well.

$$Pep = \{AA_1, AA_2, \cdots, AA_n\} \tag{2.1}$$

Such shortcomings led to the development of StaBle-ABPpred (**sta**cked ensem**ble** technique-based **ABP pred**ictor), which is essentially a stacked ensemble classifier that

comprises biLSTM at the base level and an ensemble of XGBoost, RF, and Logistic Regression (LR) at the meta level. This work puts forth a novel technique in which the base level outputs a feature vector that is combined with some handcrafted features and given to the meta level as input. Then, the ensemble at the meta level classifies a given peptide as an ABP or a non-ABP.

The proposed model was used to develop an *in silico* app that identifies ABPs in protein sequences. This app was used to find potential ABPs in the genome of Streptococcus phage T12. On BLAST analysis, most of the sequences with high classifier predicted probability values showed significant similarity with annotated AMPs in various public databases, indicating that they might possess good antibacterial potential (which can be experimentally validated). The major contributions of this work are listed as follows.

- A stacked ensemble model has been proposed to build a novel ABP classifier (StaBle-ABPpred) for identifying and classifying ABPs.
- The proposed model uses the biLSTM network at the base level (with peptide sequences as input) and an ensemble of GB, RF, and LR (with certain handcrafted features and output of base level as input).
- A global attention mechanism has been incorporated at the base level to boost performance.
- The model has been compared and analyzed statistically to prove that it outperforms the state-of-the-art models.
- A web app has been deployed online at `https://stable-abppred.anvil.app/` to find novel ABPs in proteins.

The rest of this chapter is organized as follows. Section 2.2 presents the techniques and the dataset used in this work. The proposed work is elucidated in section 2.3. The results and outcomes have been elaborated in section 2.4. The concluding remarks and future works are discussed in section 2.5.

2.2 Data and preliminaries

2.2.1 Dataset

The dataset comprises 12936 peptides collected from sources such as the antimicrobial peptide database [31], the data repository of antimicrobial peptides [32], the milk antimicrobial peptides (MilkAMP) [33] database, the starPepDB [34], etc. The non-ABPs were collected from the UniProt database [35]. To constitute the dataset used in this work, the peptides with lengths not in $\{4, 5, \cdots, 30\}$, or having a net charge less than +2, or having non-standard AAs, were discarded. This is because very large peptides may be unstable and toxic, and very small peptides may lack antibacterial properties. Also, only cationic or positively charged peptides can effectively interact electrostatically with the anionic or negatively charged cell membrane. The final dataset was divided into training (60%), validation (20%), and test (20%) sets. The test set was used to compare the model's performance with the state-of-the-art models.

Moreover, various open-source packages and libraries such as Biopython [36] and propy3 package [37] were used to compute the hand-crafted features, namely, the molecular weight, net charge, isoelectric point, GRAVY index, and APAAC. These features, along with the output of the base level of the model, were given as input to the meta level.

2.2.2 Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks are a variant of RNNs whose primary characteristic is to capture long-distance dependencies between the components of a given sequence. This is achieved using a gating mechanism in which there is a cell comprising input, output, and forget gates, which regulate the flow of information in and out of the cell. This enables the network to choose the information it wants to remember or forget selectively. Bidirectional Long Short-Term Memory (biLSTM) is characterized

by the presence of both forward and backward connections within the network layers. One of the primary benefits of biLSTMs is their capacity to concurrently capture contextual information from both preceding and subsequent elements of a sequence while processing a given element. This becomes particularly advantageous in tasks where understanding the entire context is important. The biLSTM networks consist of forward (to process sequence in the forward direction), backward (to process in the backward direction), and concatenation (combines the outputs of forward and backward) layers. The computations involved in the forward, backward, and concatenation layers have been given in Eqs. 2.2-2.4.

Forward LSTM Layer:

$$
\begin{aligned}
&\text{Input Gate: } \overrightarrow{i}_t = \sigma(\overrightarrow{W}_{ix}.x_t + \overrightarrow{W}_{ih}.\overrightarrow{h}_{t-1} + \overrightarrow{b}_i)\\
&\text{Forget Gate: } \overrightarrow{f}_t = \sigma(\overrightarrow{W}_{fx}.x_t + \overrightarrow{W}_{fh}.\overrightarrow{h}_{t-1} + \overrightarrow{b}_f)\\
&\text{Output Gate: } \overrightarrow{o}_t = \sigma(\overrightarrow{W}_{ox}.x_t + \overrightarrow{W}_{oh}.\overrightarrow{h}_{t-1} + \overrightarrow{b}_o)\\
&\text{Memory cell candidate: } \tilde{c}_t = (\overrightarrow{W}_{cx}.x_t + \overrightarrow{W}_{ch}.\overrightarrow{h}_{t-1} + \overrightarrow{b}_c)\\
&\text{Memory cell state: } \overrightarrow{c}_t = \overrightarrow{f}_t \otimes \overrightarrow{c}_{t-1} \oplus \overrightarrow{i}_t \otimes \tanh(\tilde{c}_t)\\
&\text{Hidden state: } \overrightarrow{h}_t = \overrightarrow{o}_t \otimes \tanh(\overrightarrow{c}_t)
\end{aligned}
\tag{2.2}
$$

Backward LSTM Layer:

$$
\begin{aligned}
&\text{Input Gate: } \overleftarrow{i}_t = \sigma(\overleftarrow{W}_{ix}.x_t + \overleftarrow{W}_{ih}.\overleftarrow{h}_{t+1} + \overleftarrow{b}_i)\\
&\text{Forget Gate: } \overleftarrow{f}_t = \sigma(\overleftarrow{W}_{fx}.x_t + \overleftarrow{W}_{fh}.\overleftarrow{h}_{t+1} + \overleftarrow{b}_f)\\
&\text{Output Gate: } \overleftarrow{o}_t = \sigma(\overleftarrow{W}_{ox}.x_t + \overleftarrow{W}_{oh}.\overleftarrow{h}_{t+1} + \overleftarrow{b}_o)\\
&\text{Memory cell candidate: } \tilde{c}_t = (\overleftarrow{W}_{cx}.x_t + \overleftarrow{W}_{ch}.\overleftarrow{h}_{t+1} + \overleftarrow{b}_c)\\
&\text{Memory cell state: } \overleftarrow{c}_t = \overleftarrow{f}_t \otimes \overleftarrow{c}_{t+1} \oplus \overleftarrow{i}_t \otimes \tanh(\tilde{c}_t)\\
&\text{Hidden state: } \overleftarrow{h}_t = \overleftarrow{o}_t \otimes \tanh(\overleftarrow{c}_t)
\end{aligned}
\tag{2.3}
$$

$$
\textbf{Concatenation Layer: } h^{bilstm} = \overrightarrow{h}_t \oplus \overleftarrow{h}_t \tag{2.4}
$$

Here, x_t is the embedding vector of a word at timestep t. For a forward lstm cell, at t, $\overrightarrow{h}_t$ represents the hidden state, $\overrightarrow{i}_t$, $\overrightarrow{f}_t$, and $\overrightarrow{o}_t$ is the computation performed by the input, forget and output gates, respectively. A cell's current state is denoted by $\overrightarrow{c}_t$, and $\tilde{c}_t$ is the candidate value for replacing it. Also, the weight matrices are represented by $\overrightarrow{W}_{ih}, \overrightarrow{W}_{fh}, \overrightarrow{W}_{oh}, \overrightarrow{W}_{ch}$, and $\overrightarrow{W}_{cx}$, and the biases are given by $\overrightarrow{b}_i$, $\overrightarrow{b}_f$, $\overrightarrow{b}_o$, and $\overrightarrow{b}_c$. The backward layer parameters can be explained in a similar way. Moreover, σ is the activation function, $\otimes$ represents the Hadamard product [38], and $\oplus$ denotes the addition operation, respectively.

2.2.3 Attention mechanism

Bahdanau et al. [39] developed the concept of global attention mechanism that calculates the global alignment scores (score$(h_t, h_{t'})$) using an additive function to compute the attention weights ($\alpha_{t,t'}$) for each timestep t, with respect to other timesteps, t'. This determines the relative importance of other timesteps (t') for a given timestep (t). Then, the context vector (cv_t) is computed for each timestep t, and using it, the entire sequence is represented in form of a single vector known as the attention vector (a_t) (a function of hidden states, context vectors and a weight matrix W_{cv}). The computations involved in the attention layer are given in Eq. (2.5).

$$\begin{aligned}
&\text{Attention Weights: } \alpha_{t,t'} = \frac{\exp(\text{score}({h_t}^{bilstm}, {h_{t'}}^{bilstm}))}{\sum_{s=1}^{n} \exp(\text{score}({h_t}^{bilstm}, {h_s}^{bilstm}))} \\
&\text{Context Vector: } cv_t = \sum_{t'} \alpha_{t,t'} h_{t'}^{bilstm} \\
&\text{Attention Vector: } a_t = \tanh(W_{cv}[cv_t; {h_t}^{bilstm}])
\end{aligned} \tag{2.5}$$

2.3 Proposed work

The proposed work has been illustrated using Figure 2.1 and explained as follows.

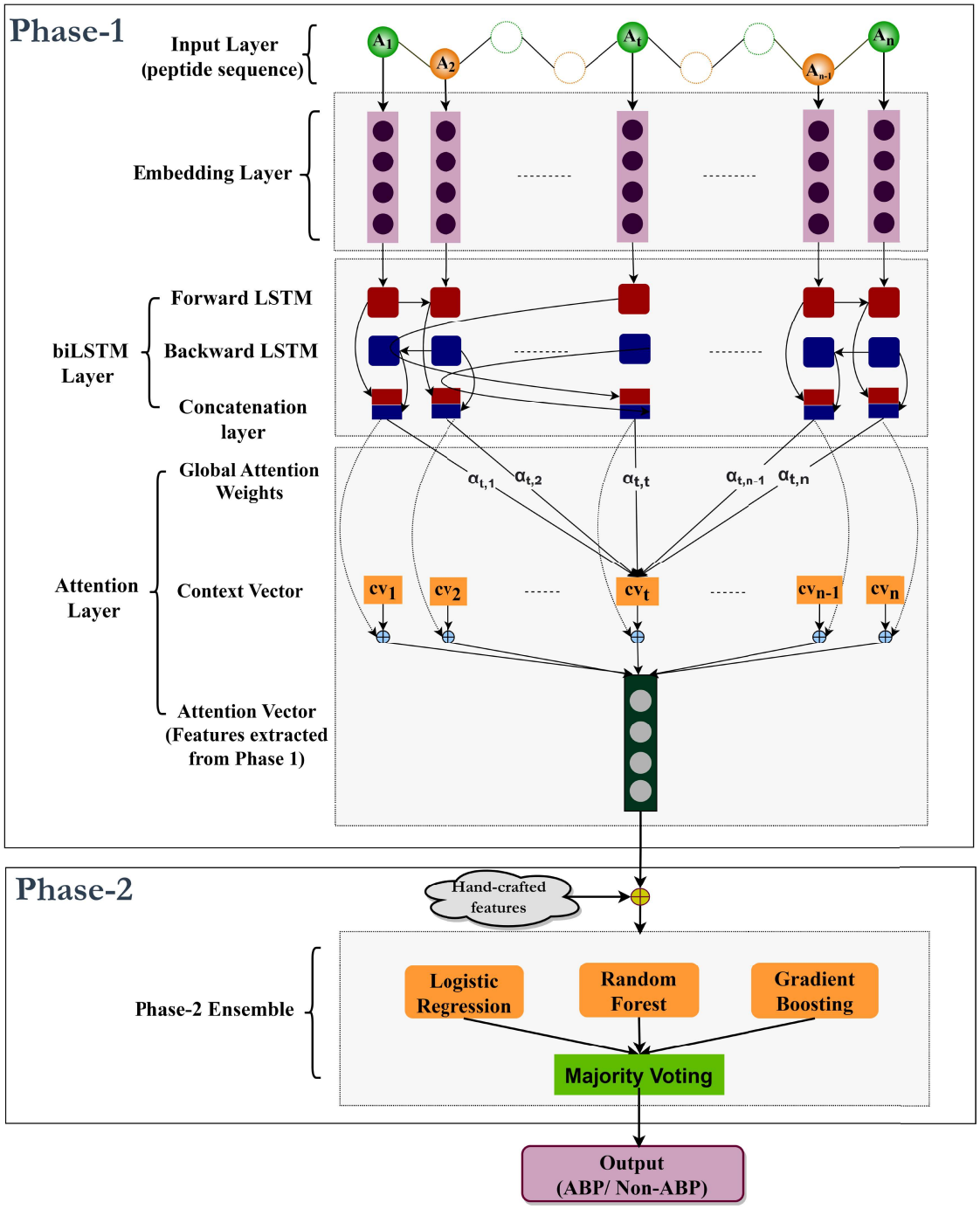


Figure 2.1: The architecture of the StaBle-ABPpred model

2.3.1 The base level

The base level consists of an embedding layer that takes a peptide sequence as input and outputs feature vectors corresponding to all the elements of that sequence. This is done using the skip-gram algorithm, a word2vec technique that analyzes the dataset and computes the feature vector for each token (AA) after taking its context into cognizance. These feature vectors are fed into a biLSTM layer, which processes these vectors in both forward and backward directions, after which the concatenated results are given to the attention layer. Then, the attention vector is calculated and passed onto the meta level.

2.3.2 The meta level

The output of the base level is integrated with some hand-crafted features as input to the meta level. These features include molecular weight, net charge, iso-electric point, APAAC, and GRAVY index. This level comprises an ensemble of XGBoost, LR, and RFs. The optimal hyperparameters of XGBoost and RF classifiers were found using the grid search cross-validation (GSCV) method. An unweighted majority voting of the results given by these classifiers decides the final classification of a peptide.

2.4 Experiments, results, and discussions

Extensive experimentation was conducted on the proposed model, StaBle-ABPpred, which was coded using the Python programming language. The trials were conducted on a central processing unit (CPU) compute node equipped with an Intel Xeon Skylake 6148 processor running at a clock speed of 2.4 GHz. The compute node also included 192 gigabytes of random access memory. In order to execute our deep learning model, the Keras framework with Tensorflow was employed as the underlying platform [40]. The comparison of our model has been conducted with several contemporary models,

including iAMPpred, Deep-ABPpred, IAMPE-KNN, IAMPE-RF, IAMPE-SVM, and IAMPE-XGBOOST, using the test set. In addition, a comprehensive statistical analysis was conducted utilizing the analysis of variance (ANOVA) followed by a post-hoc test.

2.4.1 Evaluation Criteria

Several evaluation metrics, like accuracy, recall, precision, f1-score, and area under the receiver operating characteristic curve (AUC), were used to evaluate the proposed model. These metrics are described in Eqs. 5.11-5.16.

$$Accuracy\ (Acc) = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.6}$$

$$Precision\ (Pr) = \frac{TP}{TP + FP} \tag{2.7}$$

$$Recall\ (Rec)\ (\text{or}\ True\ Positive\ Rate\ (TPR)) = \frac{TN}{TN + FP} \tag{2.8}$$

$$F1\text{-}score\ (Fs) = \frac{2 \times Pr \times Rec}{Pr + Rec} \tag{2.9}$$

$$False\ Positive\ Rate\ (FPR) = 1 - \frac{TN}{FP + TN} \tag{2.10}$$

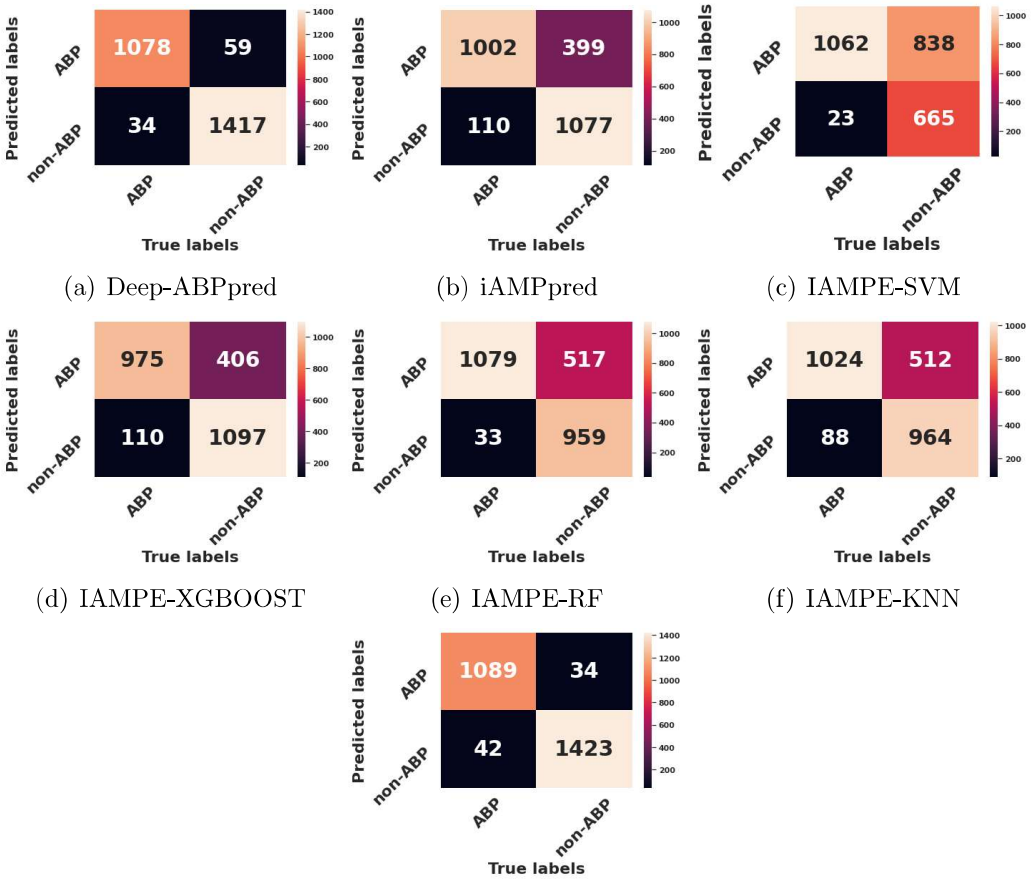
$$AUC = \int TPR.\ \mathrm{d}(FPR) \tag{2.11}$$

2.4.2 Performance Evaluation

The dataset was split into ten different combinations, and the proposed model was trained, tuned, and tested on each of these splits. The motive behind ten independent split-based experiments was to ascertain the reliability of the reported performance. The average performance of StaBle-ABPpred and other state-of-the-art models has been reported in Table 2.1. It has been found that StaBle-ABPpred outperforms other models as per all the performance metrics. The actual performance of the classifiers can be seen from the confusion matrices given in Figure 2.2, using which it can be

Table 2.1: Performance of various models on the test set

Model	Accuracy(%)	Precision(%)	Recall(%)	F1-score(%)	AUC(%)
StaBle-ABPpred	97.60	97.60	97.60	97.60	99.58
Deep-ABPpred	96.40	94.92	96.70	95.94	99.10
iAMPpred	80.50	81.50	81.70	80.50	81.66
IAMPE-KNN	77.40	79.40	78.90	77.30	78.91
IAMPE-SVM	68.00	77.70	71.40	66.90	71.43
IAMPE-RF	78.20	82.10	80.40	77.80	80.32
IAMPE-XGBOOST	76.20	81.20	78.50	75.70	78.54



(a) Deep-ABPpred (b) iAMPpred (c) IAMPE-SVM

(d) IAMPE-XGBOOST (e) IAMPE-RF (f) IAMPE-KNN

(g) StaBle-ABPpred (Proposed)

Figure 2.2: Confusion matrices for various models on test set

ascertained that our model gave the highest number of TPs and TNs.

The ANOVA test was employed to find the statistical significance of the difference in performance of StaBLe-ABPpred compared to other models. It checks whether

Table 2.2: ANOVA on accuracy (%) of various models

(a) Input summary

Group	Count	Sum	Average	Variance
StaBle-ABPpred	10	976	97.60	0.27
Deep-ABPpred	10	964	96.40	0.27
iAMPpred	10	805	80.50	0.50
IAMPE-KNN	10	774	77.40	1.16
IAMPE-SVM	10	680	68.00	0.89
IAMPE-RF	10	782	78.20	1.52
IAMPE-XGBOOST	10	762	76.20	0.63

(b) ANOVA result

Source of Variation	SS	df	MS	F stat	P-value	F crit
Between Groups	7181.97	6	1196.99	1607.90	6.48E-67	2.25
Within Groups	46.90	63	0.74	-	-	-
Total	7228.87	69	-	-	-	-

(c) Post Hoc Analysis

Model compared with StaBle-ABPpred	Difference of mean	LSD	Lower-bound	Upper-Bound
Deep-ABPpred	01.20	0.77	0.43	01.97
iAMPpred	17.10	0.77	16.33	17.87
IAMPE-KNN	20.20	0.77	19.43	20.97
IAMPE-SVM	29.60	0.77	28.83	30.37
IAMPE-RF	19.40	0.77	18.63	20.17
IAMPE-XGBOOST	21.40	0.77	20.63	22.17

the difference in means (for a performance metric) of the given models is statistically significant. This condition is true only if the null hypothesis (H_0), which states that means of all groups are equal, is false. For rejecting H_0, the p-value and F-critical reported by ANOVA must be lower than the α-level (0.05) and F-statistic, respectively. The null hypothesis can be expressed using Eq. 2.12 [41, 42]. The ANOVA test was performed on accuracy, precision, and F1-score, and the results have been given in Tables 2.2-2.4.

$$\begin{aligned} H_0 : \mu_{StaBle-ABPpred} &= \mu_{iAMPpred} = \mu_{Deep-ABPpred} \\ &= \mu_{IAMPE-KNN} = \mu_{IAMPE-SVM} \\ &= \mu_{IAMPE-RF} = \mu_{IAMPE-XGBOOST} \end{aligned} \tag{2.12}$$

The value of F-critical is much lower than the F-statistic, and the p-value is much lower than the alpha level (0.05) for all the performance metrics mentioned before. Hence, the

Table 2.3: ANOVA on precision (%) of various models

(a) Input summary

Group	Count	Sum	Average	Variance
StaBle-ABPpred	10	976	97.60	0.27
Deep-ABPpred	10	949	94.90	0.01
iAMPpred	10	815	81.50	0.50
IAMPE-KNN	10	794	79.40	0.27
IAMPE-SVM	10	777	77.70	0.46
IAMPE-RF	10	821	82.10	1.21
IAMPE-XGBOOST	10	812	81.20	0.40

(b) ANOVA result

Source of Variation	SS	df	MS	F stat	P-value	F crit
Between Groups	3744.20	6	624.03	1399.74	4.95E-65	2.25
Within Groups	28.08	63	0.45	-	-	-
Total	3772.29	69	-	-	-	-

(c) Post Hoc Analysis

Model compared with StaBle-ABPpred	Difference of mean	LSD	Lower-bound	Upper-Bound
Deep-ABPpred	02.58	0.59	01.98	03.17
iAMPpred	16.00	0.59	15.40	16.59
IAMPE-KNN	18.10	0.59	17.50	18.69
IAMPE-SVM	19.80	0.59	19.20	20.39
IAMPE-RF	15.40	0.59	14.80	15.99
IAMPE-XGBOOST	16.30	0.59	15.70	16.89

Table 2.4: ANOVA on f1-score (%) of various models

(a) Input summary

Group	Count	Sum	Average	Variance
StaBle-ABPpred	10	976	97.60	0.26
Deep-ABPpred	10	959	95.90	0.01
iAMPpred	10	805	80.50	0.50
IAMPE-KNN	10	772	77.20	0.62
IAMPE-SVM	10	669	66.09	0.76
IAMPE-RF	10	778	77.80	3.51
IAMPE-XGBOOST	10	757	75.70	0.67

(b) ANOVA result

Source of Variation	SS	df	MS	F stat	P-value	F crit
Between Groups	7476.32	6	1246.05	1373.67	8.90E-65	2.25
Within Groups	57.15	63	0.91	-	-	-
Total	7533.47	69	-	-	-	-

(c) Post Hoc Analysis

Model compared with StaBle-ABPpred	Difference of mean	LSD	Lower-bound	Upper-Bound
Deep-ABPpred	01.00	0.85	0.80	02.51
iAMPpred	17.10	0.85	16.25	17.95
IAMPE-KNN	20.40	0.85	19.55	21.25
IAMPE-SVM	30.70	0.85	29.85	31.55
IAMPE-RF	19.80	0.85	18.95	20.65
IAMPE-XGBOOST	21.90	0.85	21.05	22.75

null hypothesis can be rejected. In other words, the performance of all the models has a statistically significant difference. However, the ANOVA test does not help determine the best model. For this purpose, a least significant difference (LSD) test was done on the results of the ANOVA test. The LSD of StaBle-ABPpred was calculated with every other model on accuracy, precision, and f1-score. The 95% confidence interval (CI), which is given by the (Lower-bound, Upper-bound) was calculated using the LSD values. If this interval does not contain zero, the difference in means of the models under consideration is statistically significant. Also, a positive difference between the mean performance of StaBLe-ABPpred and another model (e.g., $\mu_{StaBle-ABPpred} - \mu_{other}$) implies that the former is better than the latter. On calculating the 95% CI (for the difference in means) of StaBle-ABPpred ($\mu_{StaBle-ABPpred}$) with all the models, it has been found that the 95% CIs do not contain zero. Moreover, from Eq. 2.13, it can be inferred that StaBle-ABPpred outperforms all the other models.

$$\begin{aligned}
&\mu_{StaBle-ABPpred} - \mu_{Deep-ABPpred} > 0 \\
&\mu_{StaBle-ABPpred} - \mu_{iAMPpred} > 0 \\
&\mu_{StaBle-ABPpred} - \mu_{IAMPE-KNN} > 0 \\
&\mu_{StaBle-ABPpred} - \mu_{IAMPE-SVM} > 0 \\
&\mu_{StaBle-ABPpred} - \mu_{IAMPE-RF} > 0 \\
&\mu_{StaBle-ABPpred} - \mu_{IAMPE-XGBOOST} > 0
\end{aligned} \tag{2.13}$$

2.4.3 Discovering ABPs in proteins

The wet lab researchers can use StaBle-ABPpred for discovering ABPs in proteins using a freely accessible web app deployed at `https://stable-abppred.anvil.app/`. To demonstrate the working of this app, some novel ABPs have been identified in the proteome of Streptococcus phage (containing 65 protein sequences). After this, the ABPs

Table 2.5: Top fifteen ABPs as per the similarity score with annotated AMPs. The first row in column 7 contains the identified ABP, and the second row comprises the matching AMP found using the BLAST tool.

S. No	Details of Protein in Streptococcus Phage T12	ABP Sequences predicted using StaBle-ABPpred	No. of amino acids	Similarity with existing AMPs: Name and source of AMP	Similarity with existing AMPs: Common name of Organism	Sequence Alignment
1.	YP_009191692.1 [Hypothetical protein AU160_gp23]	VKLVKKIKRTDALERARRM	19	Stomoxyn [Stomoxys calcitrans]	Stable fly	V K L V K K I K R T D A L E R A R R M N K L V K K V K H T I S E T A H V A K
2.	YP_009191704.1 [Putative terminase small subunit]	KAFGKYALSAIGTLVNLSK	19	Temporin- 1DRb [Rana draytonii]	California red-legged frog	K A F G K Y A L S A I G T L V N L S K - - - - - - - - X F L G T L V N L A K
3.	YP_009191731.1 [Hypothetical protein AU160_gp62]	KQNAKGLAAISNVAKKFS	18	Styelin-A [Styela clava]	Sea squirt	K Q N A K G L A A I S N V A K K F S G X F G K A F X S V S N F A K K H T
4.	YP_009191714.1 [Hypothetical protein AU160_gp45]	LASYKKDRLAILNKKVVR	18	Pilosulin 5 [Myrmecia pilosula]	Jack jumper ant	L A S Y K K D R L A I L N K K V V R V I E K G Y D K L A A K L K K V I Q
5.	YP_009191717.1 [DUF5072 family protein]	LKKLGLVDLHASIKQK	16	Dermaseptin-S8 [Phyllomedusa sauvagei]	Sauvage's leaf frog	L K K L G L V D L H A S I K Q K L K K L G L V A L H A G K A A L
6.	YP_009191689.1 [Single-stranded DNA-binding protein]	AENLANWAKKGALIG	15	Brevenin 2EB [Rana esculenta]	Edible frog	A E N L A N W A K K G A L I G L K N L A K T A G K G A L Q G
7.	YP_009191696.1 [Hypothetical protein AU160_gp27]	NSLQIIKKAKEAKRL	15	Viresin [Heliothis virescens]	Tobacco budworm moth	N S L Q I I K K A K E A K R L C L L D E G K K A P D A K E L
8.	YP_009191734.1 [Streptococcal pyrogenic exotoxin SpeA]	NNKKVLKKMVFFVL	14	Dermaseptin Svi [Phyllomedusa sauvagei]	Sauvage's leaf frog	N N K K V L K K M V F F V L - - M D I L K K S L F F I L
9.	YP_009191675.1 [Hypothetical protein AU160_gp06]	NWLGYRSQIRKLKF	14	Mastoparan-like peptide 12a [Vespa magnifica]	Hornet	N W L G Y R S Q I R K L K F N W K G I A A M A K K L L -
10.	YP_009191681.1 [Hypothetical protein AU160_gp12]	IFAKPKKQEPIKH	13	Moricin [Bombyx mori]	Silk worm	I F A K P K K Q E P I K H N F L K P K K R K A - - -
11.	YP_009191701.1 [Hypothetical protein AU160_gp32]	EKWHKYYLYKTCR	13	Defensin-1 [Crassostrea virginica]	Eastern oyster	E K W H K Y Y L Y K T C R C P W N R Y Q C H S H C R
12.	YP_009191687.1 [Siphovirus Gp157 family protein]	YKAEKEAFYKKQK	13	SK84 [Drosophila virilis]	Fruit fly	Y K A E K E A F Y K K Q K A A R E E E F F Y K K Q K
13.	YP_009191715.1 [Hypothetical protein AU160_gp46]	KDLGRWRLLVI	11	Cathelicidin 2 [Equus caballus]	Horse	K D L G R W R L L V I C S L G R W S L L L L
14.	YP_009191688.1 [ERF family protein]	LINKLKKRAGQ	11	Hepcidin 2 [Epinephelus coioides]	Orange-spotted grouper	L I N K L K K R A G Q L V N I R K K R A P T
15.	YP_009191670.1 [Site-specific recombinase]	PSLPRKWLLQI	11	Protegrin-2 [Tupaia chinensis]	Chinese tree shrew	P S L P R K W L L Q I P S L P R T V V L R V

with predicted probability score ≥ 0.99 were considered for the basic local alignment search tool (BLAST) analysis. This tool is used to check whether the sequence entered by the user is similar to one or more peptides listed in any public repository of experimentally validated AMPs. The results have been mentioned in Table 2.5. It was found that there is a high chance that these ABPs have antibacterial characteristics, which can be validated after lab-based experiments.

2.5 Conclusion

In this chapter, a novel stacked ensemble technique that comprises an attention-based biLSTM framework at the base level and an ensemble of three classic ML models at the meta level has been proposed. The novelty of this technique also lies in how the output of the base level is combined with some handcrafted features and fed to the meta level. This entire framework is named the StaBLe-ABPpred, which shows an accuracy of 98% while discriminating ABPs from non-ABPs. Through rigorous experiments, comparisons, and statistical analysis, it has been proved that the proposed model outperforms all the other state-of-the-art models like the Deep-ABPpred, iAMPpred, IAMPE-KNN, IAMPE-RF, IAMPE-XGBOOST, and IAMPE-SVM. Moreover, a free web app based on the proposed model has been deployed online to help discover new ABPs in protein sequences. Using this app, a few novel ABPs were discovered that have shown significant sequence similarity with some of the well-known annotated AMPs.

In the future, more handcrafted features can be analyzed and selected to improve the model's performance. Also, other deep learning based approaches like temporal convolutional networks, bidirectional encoding representations from transformers, etc., can be used for modeling the sequences at the base level.