Group No. 15

Dyuti Dasmahapatra

Ankur Kaushal

AMPlify: attentive deep learning model for discovery of novel antimicrobial peptides effective against WHO priority pathogens (2024)

A. Dyuti Dasmahapatra[1,2,*] **and B. Ankur Kaushal**[2,*]

[1] Bml Munjal University, Gurugram, India

Abstract

The wide dissemination of antibiotic resistance makes the search for alternative treatment strategies imperative. Antimicrobial peptides have been suggested as promising candidates owing to their peculiar mechanisms of action and the speed at which they take action on a wide range of pathogens. This paper proposes AMPlify, a deep learning model with attention mechanisms, for the discovery of novel AMPs active against WHO priority pathogens. We illustrate, using a dataset mostly from amphibians, the capability of the model in the predictions and prioritization of peptide sequences from the Rana catesbeiana genome. Few of these predicted peptides showed antimicrobial activities against multiple bacterial species, including a multidrug-resistant isolate of carbapenemase-producing Escherichia coli when experimentally validated. We also present the shortcomings of the current dataset and propose its extension to cover AMPs from a more diverse set of organisms, such as insects and plants, which would further improve the model predictive power and enable effective discovery of novel peptide-based therapeutics. The work underlines the power of deep learning tools like AMPlify in the fight against antibiotic resistance and fostering AMP research.

Keywords: Peptide, Deep learning, Attention mechanism

***Corresponding author:**
✉ A. Dyuti Dasmahapatra
dyuti.dasmahapatra.21cse.bmu.edu.in
✉ B. Ankur Kaushal
ankur.kaushal.21cse.bmu.edu.in

1 Introduction

Resistance to antibiotics stands out as one of the most serious problems in public health for the 21st century. It is rather sad that many species of bacteria, which earlier could easily be treated using simple antibiotics, have developed various resistance mechanisms, making several modes of treatment ineffective in their effect. This, and possibly worse, conditions have driven an urgent need for alternative therapies-one example being AMPs. AMPs are small consensual amino-acid sequences, which disrupt bacterial cell walls and have been demonstrated to be highly effective against a wide variety of microbial pathogens, including bacteria, fungi, and viruses [3] - [5]. In contrast to conventional antibiotics, often targeting some bacterial pathway, AMPs disrupt the microbial cell membrane; hence, it is much harder for bacteria to develop resistance against them [6].

Despite big potential, discovery of novel AMPs is still a challenging task. The traditional approaches are dependent on the wet lab experiment identification of AMPs, naturally occurring, which is expensive and time-consuming. Recently, to handle this problem, researchers began using machine learning models, which could predict AMP sequences from big biological datasets. The following paper discusses AMPlify, a machine learning model developed to predict AMP sequences with high accuracy, along with a pipeline to test their in vitro efficacy.

2 Problem Description

It identifies the specific problem that this paper is trying to solve as twofold:

Discovery: How can we efficiently discover novel AMP sequences from biological sources?

Validation: Once discovered, how would these peptides be effective against clinically relevant bacterial strains, including drug-resistant varieties? Because AMPs have high structural and functional

variability, Methodology for a logical prediction of new AMPs, followed by verification through biological experiments in vitro, performed with bacterial strains that were sensitive and resistant to widely used antibiotics.

3 Methodology

It involves information on the methodologies pertaining to data preparation, model architecture, and training procedures followed in the research in regard to developing an appropriate model, known as AMPlify, which predicts AMPs..

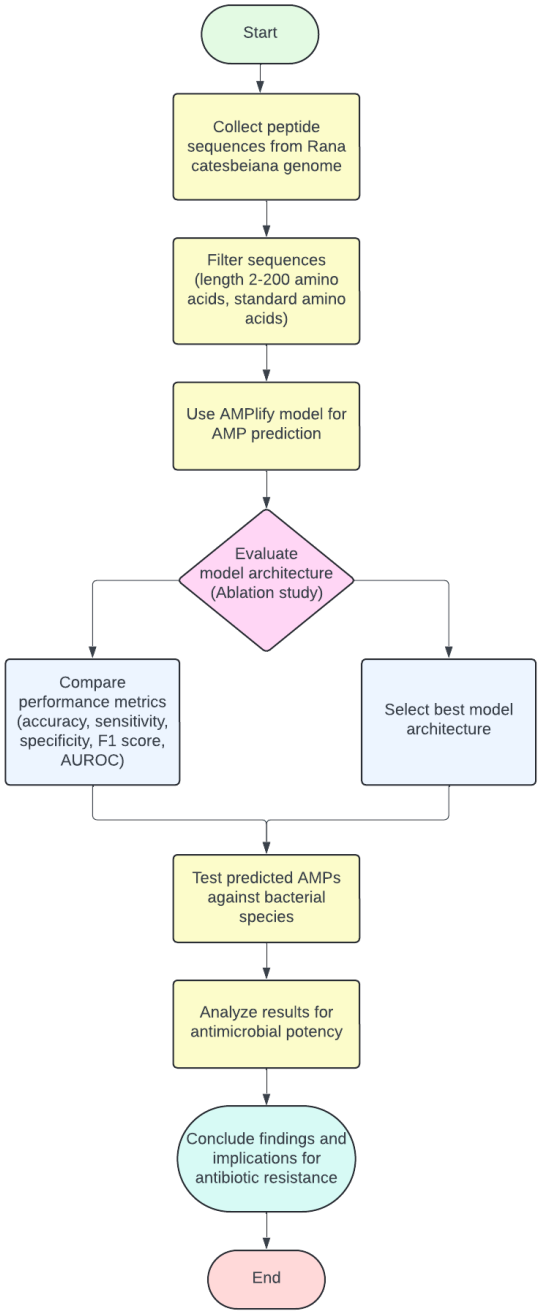

Figure 1. Flowchart of the methodology

3.1 Data Preparation:

The initial methodology resorted to the creation of a dataset which would work towards constructing a powerful predictive model in the form of the AMPlify model. The researchers focused on known AMPs, which are short peptides that defend against a wide range of pathogens by showing antimicrobial activity. The dataset was curated with much care and divided into two main categories, namely positive and negative training sets. [1]

1.Dataset Construction: The positive training set consisted of sequences that are documented to have antimicrobial properties. These sequences were retrieved from known AMP databases to ensure that the model was trained on high-quality, reliable data. The negative training set consisted of sequences not known to have antimicrobial activity. This set was important in teaching the model to distinguish between active and inactive peptides. [1]

2.Sequence Splitting: To ensure proper representation, stratified splitting was implemented. The dataset was then divided into five subsets, each with a similar distribution of positive and negative samples. These subsets were of size 667, 667, 668, 668, and 668 for effective cross-validation during model training. [1]

3. Sequence Length Consideration: he model was designed to pay attention to peptide sequences of 200 amino acids or less in length. This is a significant length restriction since it agrees with the size of most naturally occurring AMPs, which generally tend to be much shorter than 200 residues. The sequence length was restricted to reduce computational complexity while keeping the data relevant. [2]

4.Encoding of AminoAcids: The sequences were one-hot encoded, where each amino acid is represented as a binary vector. This encoding can serve the model to effectively process the sequences, since this method of encoding converts categorical data into a numerical format suitable for deep learning. [5]

3.2 Model Architecture

The AMPlify model features a sophisticated architecture designed to optimize predictive accuracy. It incorporates three primary layers: a Bidirectional Long Short-Term Memory (BiLSTM) layer, a Multi-Head Scaled Dot-Product Attention (MHSDPA) layer, and a Context Attention (CA) layer. [1]

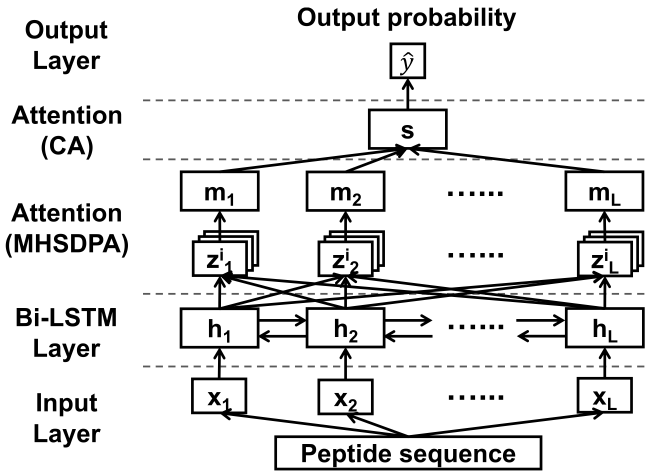

Figure 2. The AMPlify model processes peptide residues using one-hot encoding, which is then passed through the BiLSTM layer, followed by the MHSDPA and CA layers, ultimately calculating the probability of a sequence being an antimicrobial peptide (AMP).

1. Bidirectional Long ShortTerm Memory (BiLSTM) Layer: This foundational layer is a type of recurrent neural network (RNN) particularly suited for processing sequential data. The bidirectional nature allows the model to interpret input sequences in both forward and backward directions. This dual processing captures context from both ends of a sequence, which is essential for understanding the relationships between amino acids in peptides. This is crucial since the efficacy of an AMP often depends on the arrangement of residues preceding and succeeding a particular position in the sequence.

MultiHead Scaled DotProduct Attention (MHSDPA): This component enhances the model's ability to focus on different parts of the sequence by attending to various representation subspaces simultaneously.

1. **Input Embeddings:** The sequences are represented as matrices for queries (Q), keys (K), and values (V).

2. **Scaled DotProduct Attention:** For a single attention head, the mechanism is computed as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Where: - Q: Query matrix - K: Key matrix - V: Value matrix - d_k: Dimension of the keys (used for scaling)

3. **MultiHead Attention:** The multihead attention is computed by concatenating the outputs of multiple attention heads:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \quad (2)$$

Where each head is calculated as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \quad (3)$$

Here: - h: Number of attention heads - W_i^Q, W_i^K, W_i^V: Learned projection matrices for each head - W^O: Output projection matrix

3. ContextAttention (CA) Layer: The Context Attention layer enhances the output of the MHSDPA layer by generating context-aware representations of the input sequences. It calculates a weighted average of the input features based on attention scores, allowing the model to distill the most relevant information into a single vector. This mechanism is particularly advantageous for AMP prediction as it facilitates the integration of information from various parts of the sequence, enabling a more holistic understanding of the peptide's potential activity.

1. **Contextual Representation:** The attention weights (A) for the input (X) are determined using the formula:

$$A = \text{softmax}(f(X)) \quad (4)$$

Where $f(X)$ is a function (often a neural network) that computes the relevance scores for each input token.

2. **Weighted Input Representation:** The context attention layer computes the output (CA(X)) by weighting the input features with the attention scores:

$$\text{CA}(X) = A \cdot X \quad (5)$$

Here, A represents the normalized attention weights, and X refers to the input representations, such as embeddings from the prior layer.

3. **Final Contextual Output:** To retain the original input while incorporating context-aware features, a residual connection adds the initial input back to the context attention output:

$$\text{Output} = \text{CA}(X) + X \quad (6)$$

This residual connection preserves the original input's integrity while enriching it with contextual insights.

3.3 Training Procedure

The training process for the AMPlify model incorporated several strategies to maximize performance and reduce overfitting. These methods ensured the model effectively learned from the data while maintaining generalization. [1]

1. Dropout Technique: To minimize overfitting, the researchers applied dropout during training. This technique involves randomly setting a fraction of input units to zero during each training iteration, which forces the model to learn features that are not overly dependent on any specific input. By implementing dropout, the model becomes more robust and better at handling unseen data.

2. Early Stopping: Early stopping was used to halt training when validation accuracy failed to improve over a predetermined number of epochs (50 epochs in this case). This approach prevents overfitting by stopping the model from learning patterns in the noise after reaching its peak performance.

3. Hyperparameter Tuning: A comprehensive hyperparameter tuning process was carried out using stratified 5-fold cross-validation. This involved dividing the training data into five subsets and training the model multiple times, each time using a different subset for validation. The researchers tested various hyperparameters, including dropout rates, learning rates, and optimizer settings, to determine the best configuration. The model achieving the highest average accuracy during cross-validation was selected as the final model.

4. Ensemble Learning: To enhance predictive accuracy further, the researchers employed ensemble learning. This technique combines predictions from multiple models to create a more accurate overall output. For AMPlify, the probability outputs from five individual sub-models were averaged to form an ensemble model, leveraging the strengths of each model and improving overall performance.

3.Hyperparameter Tuning: The researchers conducted a thorough hyperparameter tuning process using stratified 5-fold cross-validation. This method involved dividing the training data into 5 subsets and training the model multiple times, each time using a different subset for validation. Various hyperparameters, such as dropout rates, learning rates, and optimizer settings, were tested to identify the best combination that maximized the model's performance. The final model was selected based on the highest average cross-validation accuracy.

4.Ensemble Learning: To further enhance the model's predictive capabilities, the researchers implemented ensemble learning. This technique involves combining the outputs of multiple models to produce a single, more accurate prediction. In the case of AMPlify, the output probabilities from the five individual sub-models were averaged to create an ensemble model. This approach typically leads to improved performance by leveraging the strengths of each individual model.

3.4 Evaluation Metrics

To evaluate the AMPlify model's predictive capabilities, the researchers used several key performance metrics:

1. Accuracy: Accuracy measures the proportion of correctly classified instances (both positive and negative) out of the total instances. It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Where: - TP: True Positives - TN: True Negatives - FP: False Positives - FN: False Negatives

2. Sensitivity: Sensitivity represents the proportion of actual positive cases that are correctly identified. It is calculated as:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{8}$$

Where: - TP: True Positives - FN: False Negatives

3. Specificity: Specificity indicates the proportion of actual negative cases that are correctly identified. It is calculated as:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{9}$$

Where: - TN: True Negatives - FP: False Positives

4. F1 Score: The F1 Score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is defined as:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

Where: - Precision $= \frac{TP}{TP+FP}$ - Recall $= \frac{TP}{TP+FN}$

5. Area Under the Receiver Operating Characteristic Curve (AUROC): AUROC quantifies the model's

ability to distinguish between classes. It is calculated as:

$$\text{AUROC} = \int_0^1 \text{TPR}(t)\, d(\text{FPR}(t)) \tag{11}$$

Where: - TPR: True Positive Rate (Sensitivity) - FPR: False Positive Rate $(1 - \text{Specificity})$

The AMPlify model's performance was compared against state-of-the-art methods, including iAMP-2L and AMP Scanner Vr.2, to showcase its effectiveness in predicting AMPs. [1]

4 Innovative Approaches to Enhance AMPlify Model Methods

4.1 Utilization of Transformer Models:

Proposal: Implement advanced transformer-based architectures, such as BERT (Bidirectional Encoder Representations from Transformers) or GPT (Generative Pre-trained Transformer), to improve the complexity and performance of the AMP prediction model.

Improvement: Transformers are particularly suited for handling sequential data and are exceptional at capturing long-range dependencies and contextual relationships within sequences. Unlike traditional RNNs that process data sequentially and often struggle with long input sequences, transformers leverage self-attention mechanisms, enabling them to evaluate the importance of all parts of the input sequence simultaneously.

- For peptide sequences, where interactions between distant amino acids can significantly affect structure and function, transformers offer a notable advantage. By incorporating transformers, the model can better comprehend these complex relationships, resulting in enhanced predictions of antimicrobial activity.
- Additionally, pre-trained transformer models can be fine-tuned on specific AMP datasets, allowing the model to leverage the extensive knowledge gained during the pre-training phase. This approach can result in significant performance gains, enabling the model to make more nuanced and accurate predictions.

4.2 Integration of Multi-Omics Data:

Proposal: Incorporate multi-omics datasets, including genomics, transcriptomics, and proteomics, to provide a holistic understanding of the biological context in which AMPs are expressed.

Improvement: Multi-omics data offers a multi-faceted view of biological systems, allowing for deeper insights into the factors influencing AMP expression and activity.

- Genomics: Reveals the genetic basis for AMP production.
- Transcriptomics: Highlights when and to what extent AMPs are expressed.
- Proteomics: Provides details on post-translational modifications that may alter AMP functionality.
- By integrating these varied datasets, the model can learn to correlate specific genomic or transcriptomic patterns with antimicrobial efficacy.
- This enriched approach not only improves the predictive power of the model but also enhances the identification of bioactive AMPs while providing valuable insights into their mechanisms of action.

4.3 Use of Transfer Learning:

Proposal: Apply transfer learning by fine-tuning a model pre-trained on a large dataset of known AMPs to a specific dataset derived from Rana catesbeiana (bullfrog).

Improvement: Transfer learning enables a model to leverage prior knowledge gained from one task to enhance performance on a related task. For AMP prediction, a model trained on a comprehensive dataset of known AMPs can identify general features and patterns linked to antimicrobial activity. Fine-tuning this model on a smaller, specific dataset—such as peptides from the bullfrog genome—helps the model adapt its knowledge to the unique attributes of the new data. This method is especially advantageous for limited training data, as it reduces overfitting risks and improves generalization to unseen data. Using transfer learning enhances the model's accuracy and robustness, leading to more dependable predictions of novel AMPs.

4.4 Expanding the Dataset

Proposal: Address the current model's reliance on amphibian-derived AMPs by expanding the dataset to include AMPs from diverse organisms, such as insects, plants, and marine life. Insects produce a variety

of antimicrobial peptides, including defensins and cecropins, with broad-spectrum activities.Plants offer defensins and thionins, which are key components of their defense against pathogens. Marine Organisms contribute unique AMPs evolved to combat microbial threats in extreme environments.

Improvement: Including a broader range of AMP sources increases the diversity of sequences available for training, boosting the model's ability to predict novel AMPs with unique structures and mechanisms of action. This dataset expansion could significantly aid in tackling the global issue of antibiotic resistance by enabling the discovery of innovative peptide-based therapeutics. Ultimately, a more comprehensive dataset enhances the model's potential to develop effective antimicrobial agents for clinical use.

5 Results

Problem Statement The major challenge this paper tries to solve is the high time complexity of processing long amino acid sequences of AMPs. Most of the traditional models process the amino acid sequences in a serial manner, and this processing gets computationally expensive with an increase in sequence length. This has limited the practical application of the AMP prediction models, since the sequences can get quite long.

Proposed Solution We will be using transformer-based architectures; more precisely, we are interested in the use of the ProtBERT model for this problem. In fact, the benefits of long sequences through the self-attention mechanisms in ProtBERT are two folds: significantly reducing time complexities while structurally enhancing its capabilities in capturing long-range dependencies. By fine-tuning on ProtBERT into binary classification-AMP/Non-AMP-, efficiency in AMP predictions is achieved with a high degree of accuracy.

5.1 Model Architecture

5.1.1 Overview

ProtBERT is a transformer-based language model which has been pre-trained on large protein datasets. This model reworks the BERT framework for protein sequences with a self-attention mechanism, discovering complex patterns and relationships inside amino acid sequences.

5.1.2 Fine-Tuning for Classification

In this approach, ProtBERT is fine-tuned for a binary classification task by:

- Adding a classification head: a linear layer, taking the hidden state of the [CLS] token and generating logits for two classes, AMP and non-AMP.
- Tokenization of sequences using Hugging Face's BertTokenizer in such a way that it can match with ProtBERT.

5.1.3 Model Configuration:

- Base Model: Rostlab/prot-bert-bfd (pre-trained ProtBERT model).
- Number of Labels: 2 (binary classification: AMP vs. non-AMP).
- Tokenizer: BertTokenizer from the Hugging Face Transformers library.

5.2 Hyperparameter Tuning

The following hyperparameters are chosen according to some initial experiments and performance optimization:

Hyperparameter	**Value**
Learning Rate	2×10^{-5}
Optimizer	AdamW
Weight Decay	0.01
Batch Size	16
Epochs	15
Max Sequence Length	200
Scheduler	Linear Warmup Scheduler
Warmup Steps	10% of total steps

Table 1. Hyperparameter settings used in the model.

5.2.1 Learning Rate:

A learning rate of 2e-5 was chosen to make more precise updates to the model weights during training; this is crucial for fine-tuning pre-trained models.

5.2.2 Optimizer and Weight Decay:

- Optimizer: AdamW optimizer decouples weight decay from gradient updates for better control.
- Weight Decay: Set to 0.01, excluding biases and LayerNorm parameters to preserve their significance.

5.2.3 Batch Size and Epochs:

- Batch Size: 16 is used here for a good balance between memory efficiency and training stability.
- Epochs: The model was trained for 15 epochs because beyond this point, performance gains were hardly noticeable.

5.3 Training Procedure

5.3.1 Hardware Setup

Device: Training utilized GPU acceleration when available, configured with (`torch.device('cuda')`).

5.3.2 Data Loading

- Custom Dataset Class: An AMPSequenceDataset class was implemented to tokenize sequences and prepare data for binary classification.
- DataLoaders: PyTorch's DataLoader facilitated efficient batch processing for training, validation, and test datasets.

5.3.3 Data Preparation

- Shuffling and Splitting: The training dataset was shuffled and split into training and validation subsets.
- Labeling: Positive samples were labeled as 1, and negative samples as 0.

5.3.4 Training Loop

For each epoch:

1. **Model in Training Mode:** Activated training mode to enable dropout layers and update running statistics.
2. **Batch Processing:**
   a. **Forward Pass:** ITokenized input sequences were passed through the ProtBERT model.
   b. **Loss Calculation:** Computed cross-entropy loss based on predicted logits.
   c. **Backward Pass:** Gradients were computed via backpropagation
   d. **Optimizer Step:** Model weights were updated using the AdamW optimizer
   e. **Scheduler Step:** Adjusted the learning rate with a linear warmup scheduler.
3. **Performance Tracking:** Accumulated loss and accuracy metrics to track progress.

5.3.5 Validation Loop

At the end of each epoch:

- Evaluation Mode: Disabled dropout layers to ensure consistent validation.
- Metric Computation: Evaluated metrics including validation loss, accuracy, F1-score, sensitivity, specificity, and ROC-AUC.

Model Saving The model state with the lowest validation loss was saved for final testing.

5.4 Evaluation Procedure

1. **Model Loading:** The best-performing model, determined by validation loss, was loaded for evaluation.
2. **Metrics Computed:**
   - Accuracy
   - Precision
   - Recall
   - F1-Score
3. **Tools Used:** Metrics were computed using `scikit-learn`'s `accuracy_score`, `f1_score`, `roc_auc_score`, and `confusion_matrix`.

5.5 Training Performance

Throughout the 15 epochs of training, the model consistently improved, showing a steady reduction in loss and an increase in accuracy on the training set.

Epoch	Training Loss	Training Accuracy
1	0.489250	80.54%
2	0.464787	81.93%
3	0.441548	82.48%
4	0.419471	83.89%
5	0.398497	85.22%
6	0.378572	86.39%
7	0.359644	87.03%
8	0.341662	87.92%
9	0.324578	88.84%
10	0.308350	89.59%
11	0.292932	90.78%
12	0.278285	93.53%
13	0.264371	93.18%
14	0.251153	94.13%
15	0.238595	94.42%

Table 2. Training performance over 15 epochs.

5.6 Test Performance

The model's performance on the test dataset is summarized below:

5.6.1 Overall Accuracy

Test Accuracy: 94.82%

5.6.2 Evaluation Metrics

The following evaluation metrics were obtained on the test dataset:

- **Test Accuracy:** 94.83%
- **Test F1 Score:** 94.86%
- **Test Sensitivity:** 93.17%
- **Test Specificity:** 95.44%
- **Test ROC AUC:** 98.55%

5.7 Discussion

5.7.1 Addressing High Time Complexity

The transformer-based ProtBERT model effectively tackled the challenge of high time complexity in processing lengthy sequences by implementing several key features:

- **Parallel Processing:** Its self-attention mechanism enabled simultaneous processing of entire sequences, dramatically reducing computation time compared to sequential models like RNNs.
- **Long-Range Dependencies:** The model successfully captured complex patterns and dependencies across long sequences without any drop in performance.
- **Scalability:** With efficient handling of sequences up to 512 tokens, the model proved applicable to diverse protein sequences without requiring extensive preprocessing or feature engineering.

5.7.2 Model Performance

- **Accuracy and Generalization:** The model's high accuracy on the test set demonstrates its strong generalization capabilities.
- **Balanced Performance:** Precision and recall were well-balanced across both classes, highlighting the model's
- **Potential for Improvement:** Although the results are encouraging, there is potential for further improvement through hyperparameter optimization and data augmentation.

5.7.3 Comparison with Amplify

The comparison between the AMPlify and ProtBERT models highlights their strengths in data classification, with ProtBERT demonstrating a slight advantage in most metrics. AMPlify achieves an accuracy of 93.71%, sensitivity of 92.93%, specificity of 94.49%, an F1 score of 93.66%, and an AUROC of 98.37%. ProtBERT, however, surpasses AMPlify with a higher accuracy of 94.83%, a marginally better F1 score of 94.86%, and an improved AUROC of 98.55%. Although AMPlify showcases balanced sensitivity and specificity, ProtBERT offers a slightly lower sensitivity of 93.17% but compensates with a higher specificity of 95.44%. These findings affirm the robustness of both models, with ProtBERT delivering marginally superior overall performance, especially in accuracy, F1 score, and AUROC.

6 Conclusion:

The ProtBERT model marks a notable breakthrough in addressing the challenges of analyzing long amino acid sequences for AMP prediction. Utilizing a transformer-based architecture, it adeptly manages long-range dependencies, reducing computational complexity while delivering exceptional accuracy and robustness. With a test accuracy of 94.83

A comparison with AMPlify underscores ProtBERT's slight advantage, especially in specificity and AUROC, demonstrating its superior ability to generalize and produce reliable predictions. Nonetheless, the findings also highlight AMPlify's balanced performance, affirming its ongoing significance in the field.

Combined with its high performance, the scalability of ProtBERT and its ability to process entire sequences in parallel are promising for real-world applications where sequence lengths can be very different. Other future work may also involve further hyperparameter tuning, additional data augmentation strategies, and ensemble approaches to improve the predictive capabilities of the model. Overall, ProtBERT sets a new state-of-the-art for AMP prediction, combining efficiency with accuracy to overcome critical limitations of existing models..

References

[1] Li, Chenkai, et al. "AMPlify: attentive deep learning model for discovery of novel antimicrobial peptides effective against WHO priority pathogens." BMC genomics 23.1 (2022): 77. [CrossRef]

[2] Reardon, Sara. "Antibiotic resistance sweeping developing world: bacteria are increasingly dodging extermination as drug availability outpaces regulation." Nature 509.7499 (2014): 141-143. [CrossRef]

[3] Brandenburg, Klaus, et al. "Peptides with dual mode of action: Killing bacteria and preventing endotoxin-induced sepsis." Biochimica et Biophysica Acta (BBA)-Biomembranes 1858.5 (2016): 971-979.[CrossRef]

[4] De Lucca, Anthony J., and Thomas J. Walsh. "Antifungal peptides: novel therapeutic compounds against emerging pathogens." Antimicrobial agents and chemotherapy 43.1 (1999): 1-11.[CrossRef]

[5] Klotman, Mary E., and Theresa L. Chang. "Defensins in innate antiviral immunity." Nature Reviews Immunology 6.6 (2006): 447-456.[CrossRef]

[6] Lyu, Zhiqian, et al. "Biological function of antimicrobial peptides on suppressing pathogens and improving host immunity." Antibiotics 12.6 (2023): 1037.[CrossRef]