symmetry

MDPI

*Article*

# UniproLcad: Accurate Identification of Antimicrobial Peptide by Fusing Multiple Pre-Trained Protein Language Models

**Xiao Wang [1,2,*], Zhou Wu [1], Rong Wang [3] and Xu Gao [4,*]**

[1] School of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou 450002, China; 332107040602@zzuli.edu.cn
[2] Henan Provincial Key Laboratory of Data Intelligence for Food Safety, Zhengzhou University of Light Industry, Zhengzhou 450002, China
[3] School of Electronic Information, Zhengzhou University of Light Industry, Zhengzhou 450002, China; wangrong@zzuli.edu.cn
[4] National Supercomputing Center in Zhengzhou, School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China
* Correspondence: wangxiao@zzuli.edu.cn (X.W.); gaoxu@zzu.edu.cn (X.G.)

**Abstract:** Antimicrobial peptides (AMPs) are vital components of innate immunotherapy. Existing approaches mainly rely on either deep learning for the automatic extraction of sequence features or traditional manual amino acid features combined with machine learning. The peptide sequence contains symmetrical sequence motifs or repetitive amino acid patterns, which may be related to the function and structure of the peptide. Recently, the advent of large language models has significantly boosted the representational power of sequence pattern features. In light of this, we present a novel AMP predictor called UniproLcad, which integrates three prominent protein language models—ESM-2, ProtBert, and UniRep—to obtain a more comprehensive representation of protein features. UniproLcad utilizes deep learning networks, encompassing the bidirectional long and short memory network (Bi-LSTM) and one-dimensional convolutional neural networks (1D-CNN), while also integrating an attention mechanism to enhance its capabilities. These deep learning frameworks, coupled with pre-trained language models, efficiently extract multi-view features from antimicrobial peptide sequences and assign attention weights to them. Through ten-fold cross-validation and independent testing, UniproLcad demonstrates competitive performance in the field of antimicrobial peptide identification. This integration of diverse language models and deep learning architectures enhances the accuracy and reliability of predicting antimicrobial peptides, contributing to the advancement of computational methods in this field.

**Keywords:** antimicrobial peptides; deep learning; protein languages models



**Citation:** Wang, X.; Wu, Z.; Wang, R.; Gao, X. UniproLcad: Accurate Identification of Antimicrobial Peptide by Fusing Multiple Pre-Trained Protein Language Models. *Symmetry* **2024**, *16*, 464. https://doi.org/10.3390/sym16040464

Academic Editor: Arkadiusz Chworos

Received: 7 March 2024
Revised: 24 March 2024
Accepted: 1 April 2024
Published: 11 April 2024



## 1. Introduction

The misuse of antibiotics has led to the development of drug resistance in bacteria, rendering infections challenging to treat and potentially life-threatening [1]. Antimicrobial peptides (AMPs) present a potential alternative to antibiotics [2]. They constitute a class of small protein molecules or peptides with antimicrobial activity, widely found in the flora and fauna of the natural world. AMPs possess the capability to combat various microorganisms, including bacteria and fungi. These peptides exert their antimicrobial effects through diverse mechanisms such as disrupting cell membranes, interfering with protein synthesis, and inducing self-destruction in microorganisms. The unique mechanism of action displayed by AMPs sets them apart from antibiotics, making it challenging for bacteria to develop resistance against them [3].

However, the wet lab experiments required for identifying and characterizing AMPs are complex and time-consuming, necessitating the development of efficient predictive models through modern computational science. Currently, several computation-based

AMP predictors have already been developed, ClassAMP utilizes a combination of features, including charge, hydrophobicity, BLOSUM-50 matrix scores, conformational similarity, normalized van der Waals volume, polarity, and polarizability [4]. Following feature selection, these characteristics are employed as input for a multiclass classification model composed of random forests and support vector machines. This integrated approach enables the accurate prediction and classification of AMPs. IAMPpred is specifically designed for variable-length AMP sequences. It employs a 1D-CNN to process the AMP sequences, extracting feature vectors from the hidden layers to serve as representations of the peptides' features. Subsequently, these feature vectors are input into a support vector machine (SVM) to accomplish the classification task. This integrated methodology effectively addresses the challenge posed by variable-length sequences and enables accurate classification of AMPs [5]. IAMPE [6] categorizes amino acids into different groups using their 13CNM resonance spectra and analyzes the composition and distribution of members within these groups to construct feature vectors for antimicrobial peptide sequences. Then, these vectors were input into SVM and random forests for AMP classification prediction. AMPfun [7] employs a comprehensive feature set, including n-gram features, amino acid composition (AAC) features, pseudo amino acid composition (PseAAC) features, and physicochemical features. Following feature selection, the model utilizes SVM and Random Forest (RF) algorithms for the classification and prediction of AMP functionalities. iAMPCN [8] employed diverse encoding methods for input sequences, utilizing distinct CNN to extract features; subsequently, these extracted vectors are input into a multilayer perceptron for classification, culminating in the successful prediction of AMP functionalities. This approach showcases a nuanced strategy, leveraging different encoding techniques and specialized CNN architectures to enhance the accuracy of feature extraction and subsequent functional predictions for AMPs. sAMPpred-GAT [9] used graph attention mechanisms and incorporated structural features into deep learning networks, further improving the predictive accuracy of AMPs. iAMP-Attenpred [10] uses the BERT feature extraction method and CNN-BiLSTM-Attention combination model to achieve binary classification prediction of antimicrobial peptides. Despite the success of the aforementioned classifiers on their respective datasets, according to the research conducted by XU, these classifiers did not perform well on a comprehensive dataset [11]. We analyze that the classifiers were unable to fully capture the diversity of data distributions. Hence, there is a need to develop a novel AMP classifier that enhances overall accuracy and possesses improved generalization capabilities by ensuring a more comprehensive extraction of AMP features.

Protein language models are a specific type of neural network that possess the ability to predict the subsequent character or vocabulary based on preceding text and have found applications in the field of biochemistry as a transfer learning tool [12]. By inputting protein sequences and learning the inherent biochemical properties, structural information, and other intrinsic patterns, protein language models generate feature vectors that can be applied to various downstream protein tasks. Existing research has demonstrated favorable results in multiple downstream prediction tasks employing protein language models [13]. However, different protein language models are developed using different training datasets, leading to differences in the emphasis placed on protein representation by each model. Consequently, a single protein language model may suffer from incomplete protein representation.

In order to apply protein language models to the task of antimicrobial peptide prediction and overcome the issue of incomplete protein representation inherent in a single protein language model, we merged three different protein language models: ESM-2, ProtBert, and UniRep. They are based on Transformer, BERT, and RNN, respectively. Therefore, the emphasis on representing proteins also varies, leading to differences in the emphasis on protein feature vectors. We extracted and fused protein features from these three different protein language models. The merged vectors elevate the comprehensiveness of protein representation to a new level.

Peptide sequences can exhibit symmetry through repetitive or mirrored patterns of amino acids, which can be crucial for the peptide's stability, folding, and interaction with other molecules. Further, 1D-CNN can learn to identify and extract symmetrical features from peptide sequences by adjusting the weights of its convolutional filters. This bidirectional processing ability makes the Bi-LSTM particularly well suited to capturing symmetrical features within sequences, as symmetry often involves the interrelationship between the two ends of the sequence. Therefore, the merged feature vectors are then input into a hybrid deep learning network composed of multi-layered bidirectional LSTM networks, 1D-CNN, and an attention mechanism. The performance score of this model is validated through relevant verification methods, ultimately achieving superior results compared to state-of-the-art research.

## 2. Materials and Methods

### *2.1. Dataset and Data Preprocessing*

To date, plenty of AMP databases exist. The Antimicrobial Peptide Database (APD) [14] is an early-established repository that aggregates extensively sourced AMP sequences and related information. It encompasses AMP data from various biological domains, including bacteria, fungi, and animals, along with classification, structure, and activity information for these peptides. Linking Antimicrobial Peptide [15] (LAMP) provides sequences of AMPs from various organisms, both internal and external, accompanied by relevant literature citations and additional annotation data. The Collection of Antimicrobial Peptides [16] (CAMP) consolidates AMP information from different species, including various structural classification details. The Database of Antimicrobial Activity and Structure of Peptides [17] (DBAASP) serves as a database for storing and providing information on AMPs, encompassing sequences, structures, antimicrobial activity, and relevant literature citations. The Data Repository of Antimicrobial Peptides [18] (DRAMP) is a comprehensive AMP database containing structural data and annotation entries. The Structurally Annotated Therapeutic Peptides Database [19] (SATPdb) offers a wealth of AMP structural data, primarily predicted through computational tools. These database establishments facilitate researchers, aiding in the profound exploration and scientific advancement of the field of AMPs.

To mitigate the impact of varying data distributions across different databases, we utilized a comprehensive benchmark evaluation dataset for training and validation. This benchmark dataset comprises AMP and non-AMP peptide data, the positive data are from six different databases: APD, LAMP, CAMP, DBAASP, DRAMP, and SATPdb, the negative data was randomly extracted from UniProt. As new databases may reference data from earlier databases, leading to potential data overlap between different databases, the CD-HIT tool [20] was employed to eliminate redundant AMP and non-AMP peptides, and sequences with a similarity exceeding 90% between peptide sequences from different databases were removed. Based on the research by Yue Zhang [21] and Ke Yan [9], excessively long peptide chains may result in more complex structures, making protein synthesis challenging; conversely, overly short AMPs may lack sufficient functional sites or structural domains and are prone to degradation in the environment. Therefore, sequences with lengths ranging from 10 to 100 amino acids were selected, and excluded sequences that contained non-standard amino acids (B, J, O, U). This process resulted in a benchmark evaluation dataset comprising 4550 AMPs and 4550 non-AMPs. The lengths of AMPs typically lean towards the shorter side, predominantly falling within the range of 20–40 amino acids, the antimicrobial activity of AMPs tends to weaken with increasing length, and, as the number of amino acids increases, the quantity of AMPs gradually decreases. In contrast, non-AMPs generally exhibit greater length [7] from the sequence distribution plot of our constructed dataset, which is shown in Figure 1. This observation underscores the conformity of our dataset to the aforementioned pattern, implying that the dataset effectively captures the diverse length distribution of AMP. It is important to note that the non-AMP sequences in this dataset are longer on average than the AMP sequences, which may introduce some

bias in the experiments. However, since most existing works have evaluated their models based on this dataset [11], we have also utilized it for our study.

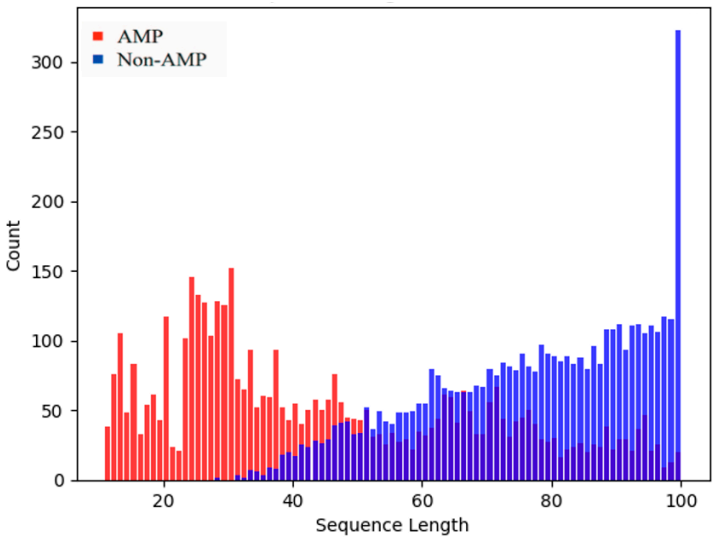


**Figure 1.** Sequence length distribution on train dataset.

To assess the model's generalization ability, an independent test set named XUAMP, created by Xu et al. [11], was utilized. This dataset includes 1536 AMPs and 1536 non-AMPs. The CD-HIT tool was applied to the independent test set and the benchmark evaluation dataset to remove sequences with a similarity exceeding 90%, ensuring the independence of the data and providing a more objective evaluation of the model's generalization ability.

To better compare our model with the latest model iAMP-Attenpred, we also utilized the datasets from Xing et al.'s [10] research: Xingdataset1 and Xingdataset2. The AMPs in Xingdataset1 were collected from the AMPer [22], APD3 [14], and ADAM [23] databases, and only sequences containing standard amino acids were chosen, with sequences having a similarity higher than 90% removed, and non-AMPs peptides were collected from the UniProt database, resulting in a total of 3594 AMPs and 3925 non-AMPs. For Xingdataset2, AMPs were collected from APD [14], and only sequences containing standard amino acids were chosen, and, with sequences having a similarity higher than 40% removed, non-AMPs were collected from the UniProt database, resulting in a total of 879 AMPs and 2405 non-AMPs.

### *2.2. The Framework of UniproLcad*

In this study, we have created a predictor for AMPs using a deep learning approach. Our predictor incorporates multi-perspective features extracted from various protein language models, enhancing its predictive capabilities. As shown in Figure 2, the predictor consists of the following key components.

(1) In the Bi-LSTM layer, the model utilizes two stacked Bi-LSTM layers to process input features. This approach enables the learning of hidden representations and captures dependencies among contextual information. Theoretically speaking, increasing the number of Bi-LSTM layers can improve the fitting effect of the model; however, experiments have shown that using two Bi-LSTM layers can achieve higher model effects than more layers. The first Bi-LSTM hidden layer has a size of 128, and the size of the second Bi-LSTM hidden layer is 2.

(2) In the 1D-CNN Layer, we utilize a 1D-CNN to extract protein information from the hidden layers of a Bi-LSTM network, obtaining higher-dimensional protein feature vectors to better adapt to our model. Experimental results indicate that the model performs optimally using a single convolution layer with convolutional units having a kernel size of 2000, input channels of 4, and output channels of 2.

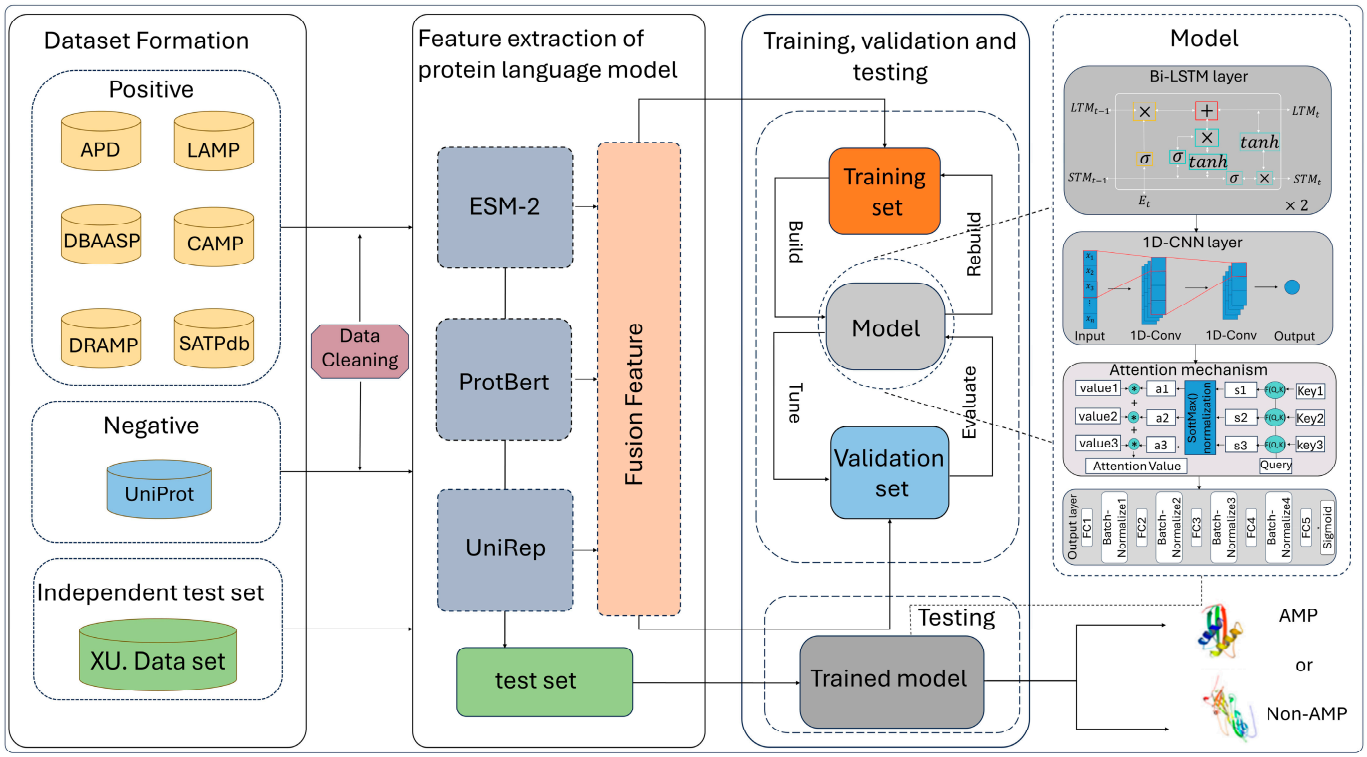


**Figure 2.** Overview of UniproLcad.

(3) Attention mechanism: Following the 1D-CNN layer, an attention mechanism is employed, which can effectively assign attention weights to high-dimensional sequence features from the output of the 1D-CNN layer. Specifically, it refers to the attention of the output $y$ at a certain moment on various parts of the input $x$, which is the redistribution of weights. In other words, it involves the reassignment of weights for each part of $x$ at each moment with respect to its contribution to $y$. The calculation formulas for the attention mechanism are as follows.

$$e_t = \tanh\left(Wh_t + b\right) \tag{1}$$

$$\alpha_t = \frac{\exp\left(e_t^T v_e\right)}{\sum_k \exp\left(e_k^T v_e\right)} \tag{2}$$

$$V = \sum_t \alpha_t h_t \tag{3}$$

Firstly, obtain the hidden representation of the hidden state $h_t$ through a fully connected layer $e_t$. Here, $W$ and $b$ represent the parameter matrix and bias of a single-layer perceptron. Then, calculate the importance values of the elements as the similarity between the element-wise context vector $E_t$ and $V_e$. The variable $V_e$ represents a high-level representation of a fixed query. During the training process, the values of $V_e$ are initialized randomly and then optimized collectively to learn the most effective representation. Normalize the importance weights using a SoftMax function. Finally, the output $V$ is the product of $\alpha$ and $h$.

(4) Output layer: the role of the output layer is to reshape and process the feature vector produced by the neural network, ultimately yielding classification results. This involves operations such as flattening, batch normalization, and the Sigmoid activation function. The output of the attention mechanism is fed into Multilayer Perceptron (MLP). The first fully connected layer comprises 1024 nodes, followed by layers with 512, 256, 128,

and 8 nodes, respectively. A linear layer with high discriminative power is necessary for this purpose, and the linear layer is defined as follows:

$$x^t = W_1^t x^{(t-1)} + b^t \tag{4}$$

where $x^t$ and $x^{(t-1)}$ are the output and input vectors, respectively, $x^0$ is the initially flattened vector, $W_1^t$ is the weight matrix, and $b^t$ is the bias for the linear layer.

Batch normalization is applied after each fully connected layer to maintain reasonable data distributions. Batch normalization operates on the principle of normalizing data within each training batch to ensure a stable distribution of input data. The specific procedure involves calculating the mean and standard deviation for each batch of data and then normalizing the data to have a mean of 0 and a standard deviation of 1. The computation is expressed as follows:

$$x_{i+1} = \alpha \frac{x_i - \mu_B}{\sigma_B} + \beta \tag{5}$$

where $x_{i+1}$ represents the data after batch normalization, $\mu_B$ denotes the mean of the current batch, $\sigma_B$ represents the standard deviation of the samples, and $\alpha$ and $\beta$ are used for scaling and shifting the data samples, respectively.

The Sigmoid function is applied to the output of the final layer. The Sigmoid function transforms the output to a range between 0 and 1, treating outputs greater than or equal to 0.5 as AMPs and those less than 0.5 as non-AMPs. The specific formulation is expressed as follows:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

where $x$ is the output of the final linear layer.

### *2.3. Protein Language Model Feature Extraction*

Traditional manual feature extraction methods have limitations as they tend to overlook the differences between protein sequences. Protein language models have been successfully applied to various downstream protein prediction tasks. Therefore, this study utilized the UniRep protein language model, the ProtBert protein language model, and the ESM-2 protein language for feature extraction from AMP sequences. The extracted feature vectors were then fused to comprehensively obtain the feature representation of AMPs.

#### 2.3.1. UniRep Protein Language Model

The UniRep protein language model employs LSTM neural networks as its foundational architecture. It continuously optimizes the LSTM neural network by predicting whether the next amino acid value in the sequence is the same as the true amino acid value. Ultimately, the model uses the average of the hidden layer units of multiple LSTM networks as the feature representation of the sequence. UniRep utilizes multiple GPUs and undergoes three weeks of training on approximately 24 million protein sequences from the UniRef50 database. The model can map protein sequences of different lengths to a unified length of 1900-dimensional feature vectors. UniRep effectively categorizes protein sequences with lower uniformity on the sequence into categories with higher structural similarity. In this model, UniRep was used to extract features from AMPs, mapping them to 1900-dimensional feature vectors to better represent the global features of AMPs.

#### 2.3.2. ProtBert Protein Language Model

The ProtBert protein language model employs Transformer/BERT architecture and is trained extensively using over two billion protein sequences from the BDF protein database and UniRep protein database. The model successfully maps protein sequences of varying lengths to a unified length of 1024-dimensional feature vectors. ProtBert has demonstrated successful applications across various downstream tasks, producing favorable outcomes. It leverages the multi-head attention mechanism derived from the Transformer architecture,

enabling it to highlight the local characteristics of sequences while retaining a strong representation of global features [24,25]. In this model, ProtBert was used to extract features from AMPs, mapping them to 1024-dimensional feature vectors. This ensures a high level of global features while highlighting the local features of AMPs.

#### 2.3.3. ESM-2 Protein Language Model

The ESM-2 protein language model is an unsupervised protein language model that operates without the need for annotated data pertaining to protein structure and function. Instead, it relies solely on vast amounts of protein sequence data. This characteristic empowers ESM-2 to harness the extensive information present in protein databases, free from the constraints of experimental data. Another distinctive feature of ESM-2 is its ability to achieve zero-shot or few-shot predictions. In other words, it does not require additional training or fine-tuning for each specific task. Utilizing the feature representation generated by ESM-2 as input enables its direct application to various protein-related tasks. Based on the aforementioned characteristics, we believe that using the ESM-2 language model to encode AMPs can yield strong feature representations, the variable-length AMP sequences were uniformly encoded into 1024-dimensional eigenvectors.

In this study, we adopted a comprehensive approach by integrating three distinct protein language models—UniRep, ProtBert, and ESM-2—to comprehensively extract features from AMP sequences. The rationale behind combining these models lies in their unique strengths and complementary capabilities. The integration aims to fully leverage their respective advantages. UniRep focuses on capturing sequence dependencies, ProtBert simultaneously emphasizes local and global features, and ESM-2 provides insightful perspectives from an unsupervised standpoint. The amalgamation of these models seeks to overcome individual model limitations, providing a more robust and nuanced representation of AMP sequences. This integrated approach is anticipated to yield a more comprehensive feature set, better capturing the multifaceted characteristics of AMPs and ultimately enhancing the accuracy and generalization of downstream prediction tasks.

### *2.4. Deep Learning Network Model*

#### 2.4.1. Bi-LSTM Networks

RNN is a recurrent neural network structure that possesses memory capabilities when processing sequential data, allowing it to capture contextual information within the sequence [26]. However, traditional RNNs have limitations. In longer text sequences, the traditional RNN structure faces issues like vanishing or exploding gradients, making it challenging to effectively capture long-term dependencies.

LSTM is designed to address the issues encountered by traditional RNNs. Introducing three gates (input gate, forget gate, and output gate) and an internal cell state, LSTM enables better control over the flow of information [27]. Through carefully designed gate mechanisms, LSTM can selectively remember or forget information, thereby capturing long-term dependencies more effectively and mitigating the gradient-related challenges.

Traditional RNNs and LSTM networks transmit information in a unidirectional manner, lacking the ability to gather information about future states. However, protein sequences can be seen as a form of biological language, akin to sentences where peptide segments represent sentences and individual amino acid residues function as words. To accurately predict outcomes, it is essential to consider the contextual relationships among these residues [28].

To address the limitations of LSTMs, this study employs bidirectional LSTM (Bi-LSTM) networks. By processing input from both directions, Bi-LSTM learns long-distance dependencies in peptide sequences in a bidirectional manner, capturing information from both the front and back ends. This enhances the neural network's expressive power and enables it to better understand the complex relationships within protein sequences [29,30]. With this architectural design, the output layer incorporates both historical and prospective information.

#### 2.4.2. Convolutional Neural Networks

Next, 1D-CNN is a variant of convolutional neural networks specifically designed to process one-dimensional sequential data [31], such as time series data or text sequences in natural language processing. Unlike traditional two-dimensional CNNs used in image processing, 1D-CNN primarily focuses on feature extraction along a single direction, making it suitable for data with sequential structures. It has demonstrated excellent performance in various applications, including processing time-series data, analyzing speech signals, and handling text sequences in NLP.

In the field of bioinformatics, 1D-CNN finds extensive applications, especially in tasks involving the processing of protein sequences [32]. This network architecture proves effective in capturing local features and patterns within sequential data, yielding favorable results across diverse tasks. In this study, we utilize a 1D-CNN to extract protein information from the hidden layers of a Bi-LSTM network, obtaining higher-dimensional protein feature vectors to better adapt to our model.

#### 2.4.3. Attention Mechanisms

Attention mechanisms are a prevalent technique in deep learning that emulates the selective focus observed in the human visual system or cognitive processes [33,34]. Attention mechanisms have found widespread applications in the fields of natural language processing and computer vision. The fundamental idea behind attention mechanisms is to assign different attention weights to different parts of information when processing sequential data, emphasizing more on crucial information, which allows models to concentrate on the most relevant information for a given task. This capability proves beneficial for enhancing the handling of long sequences or complex data, ultimately improving model performance. Furthermore, attention mechanisms are frequently employed in bioinformatics in conjunction with recurrent neural networks, showcasing competitive performance across a broad spectrum of biological sequence analysis problems [35,36]. In this research, attention mechanisms are utilized to identify crucial information influencing AMP prediction by feeding the high-dimensional protein feature vectors through a deep learning network structure, and an attention mechanism is applied to assign attention weights to these vectors.

For effective training, this study employs a dynamic learning rate algorithm, ReduceLROnPlateau. This algorithm is one of the learning rate schedulers available in PyTorch, designed to dynamically adjust the learning rate during training based on performance metrics from the validation set. The primary objective of this scheduler is to reduce the learning rate when the model's performance on the validation set ceases to improve, thereby facilitating more effective convergence. The pseudocode for this scheduler is presented in Algorithm 1. Specifically, when the accuracy (*ACC*) on the test set remains unchanged for two consecutive epochs, the learning rate is adjusted to 70% of its original value. The utilization of such a scheduler aims to enhance the model's convergence capabilities and adaptability to varying complexities in the training process.

**Algorithm 1.** ReduceLROnPlateau algorithm.

ReduceLROnPlateau: dynamic learning rate algorithm

```
    input: ACC values for each test set
    Output: Updated learning rate
1   L ← Current learning rate
2   if the ACC value does not change do
3     L ← L*0.7
4   else
5     L ← L
6   return L
```

In order to accomplish the classification task, the proposed model in this study employs a dynamic learning rate algorithm: in this study, a binary cross-entropy loss function was employed during the model training process, and the computation of the loss is as follows.

$$Loss = \frac{1}{N}\sum_{n=1}^{N} l_n \tag{7}$$

$$l_n = y_n \cdot \log x_n + (1 - y_n) \cdot \log\,(1 - x_n) \tag{8}$$

where $x_n$ represents the model's output, and $y_n$ denotes the true label.

The model is evaluated using ten-fold cross-validation, and parameter settings, based on ESM-2, ProtBert, and UniRep in the train dataset, including a batch size of 64, a learning rate of 0.001, and an Adam [37] optimizer for model optimization. The training epoch is set to 20.

### *2.5. Model Performance Evaluation*

By evaluating the performance of the model, one can choose the most suitable combination of parameters among numerous possibilities for the prediction task, thereby effectively predicting AMPs. In this study, five metrics are employed to assess the proposed method, and their calculation formulas are presented in Equation (9):

$$\begin{cases} ACC &= \frac{TP+TN}{TP+TN+FN+FP} \\ MCC &= \frac{TP\times TN - FN\times FP}{\sqrt{(TP+FN)(TP+FP)(TN+FP)(TN+FN)}} \\ Sn &= \frac{TP}{TP+TN} \\ Sp &= \frac{TN}{FP+TN} \end{cases} \tag{9}$$

where $TP$, $FP$, and $FN$ represent true positives, false positives, true negatives, and false negatives, respectively. $ACC$ (accuracy) denotes the model's accuracy; $MCC$ (Matthew's correlation coefficient) represents the Pearson correlation coefficient; $Sn$ (sensitivity) indicates the model's sensitivity; and $Sp$ (specificity) represents the model's specificity. These four metrics are commonly used for statistical predictions of model performance.

When $ACC$ = 1, it indicates that all AMP predictions are correct, and when $ACC$ = 0, it implies that all AMP predictions are incorrect. Sensitivity ($Sn$) and specificity ($Sp$), respectively, represent the model's ability to predict AMPs and non-AMPs. The closer MCC is to 1, the model is considered to be more perfect; the closer it is to 0, the closer the model's performance is to a random classifier; and the closer it is to -1, the more opposite the model's predictions are to reality.

AUC (Area Under the Curve) is a widely used metric for evaluating the performance of machine learning models in binary classification problems. It assesses the quality of a model's predictions by measuring the area under the Receiver Operating Characteristic (ROC) curve. The ROC curve represents the trade-off between the true positive rate (sensitivity or recall) and the false positive rate, with the true positive rate plotted on the $y$-axis and the false positive rate on the $x$-axis. The AUC is calculated as the area under this curve. One key advantage of AUC is its robustness in handling class imbalance, meaning that it remains reliable even when there is a substantial difference in the number of positive and negative instances in the dataset. This makes AUC a valuable performance evaluation metric in practical applications. It allows for the comparison of different models and facilitates the selection of the model with superior performance.

## 3. Results and Discussion

To validate the effectiveness of the structures we employed, we conducted ablation experiments aimed at demonstrating the correctness of the selected architecture. Furthermore, in order to better align the model with our dataset, we performed parameter tuning optimization. These endeavors were undertaken to ensure the model's adherence to our data and to affirm the suitability of the chosen structures and parameter configurations

for our task. Through these systematic steps, we enhance our confidence in the selected architecture and parameter settings, thereby improving the model's adaptability to our specific requirements.

### *3.1. Selecting Model Architecture of UniproLcad*

Based on the training set, using ten-fold cross-validation, we evaluated the performance of protein language models including ESM-2, ProtBert, and UniRep features individually, as well as their combinations: ESM-2 features, ProtBert features, UniRep features, ESM-2 features combined with ProtBert features, ESM-2 features combined with UniRep features, ProtBert features combined with UniRep features, and ESM-2 features combined with ProtBert and UniRep features, using the Bi-LSTM architecture and 1D-CNN architecture. This comprehensive assessment allowed us to analyze the predictive capabilities of these protein language models. The results are depicted in Table 1.

**Table 1.** Performance of different protein language models in feature extraction on the training set using ten-fold cross-validation experiments.

| Model | *ACC* | *MCC* | *Sn* | *Sp* |
|---|---|---|---|---|
| ESM-2 | 0.951 | 0.942 | 0.938 | 0.973 |
| ProtBert | 0.947 | 0.934 | 0.927 | 0.955 |
| UniRep | 0.932 | 0.93 | 0.931 | 0.927 |
| ESM-2 + ProtBert | 0.953 | 0.949 | 0.948 | 0.962 |
| ESM-2 + UniRep | 0.947 | 0.933 | 0.929 | 0.956 |
| ProtBert + UniRep | 0.945 | 0.939 | 0.933 | 0.936 |
| ESM-2 + ProtBert + UniRep | 0.972 | 0.956 | 0.963 | 0.971 |

For individual protein language models, the feature representation of ESM-2 exhibited the highest performance, surpassing ProtBert and UniRep by 1.4% and 1.9%, respectively. In the case of two protein language models, the combination of ESM-2 and ProtBert features outperformed ESM-2 + UniRep and ProtBert + UniRep by 0.6% and 0.4%, respectively. Ultimately, the feature representation of ESM-2 + ProtBert + UniRep achieved the highest overall performance, surpassing ESM-2 and ESM-2 + ProtBert by 2.1% and 1.9%. This highlights the superiority of combining multiple language models in capturing diverse aspects of protein sequences.

### *3.2. Ablation Experiment*

In order to ensure the effectiveness of the selected network model, we conducted ablation experiments on three network modules based on the training set, namely removing Bi-LSTM, 1D-CNN, and the attention mechanism. The experimental results are depicted in Table 2.

**Table 2.** Performance of different deep learning architectures on the training set using ten-fold cross-validation experiments.

| Model | *ACC* | *MCC* | *Sn* | *Sp* |
|---|---|---|---|---|
| 1D-CNN + Attention | 0.944 | 0.928 | 0.937 | 0.952 |
| Bi-LSTM+ Attention | 0.953 | 0.937 | 0.941 | 0.972 |
| Bi- LSTM + 1D-CNN | 0.960 | 0.942 | 0.947 | 0.977 |
| Bi- LSTM + 1D-CNN + Attention | 0.972 | 0.956 | 0.963 | 0.971 |

The removal of Bi-LSTM resulted in a decrease in model accuracy by 2.8%, the removal of 1D-CNN led to a decrease of 1.9% in accuracy, and the removal of the attention mechanism caused a reduction of 1.2% in model accuracy.

### 3.3. Model Parameter Selection

To explore the optimal parameter combination for the predictor, we selected parameters on the training set based on the average results of the model's ten-fold cross-validation. Initially, considering the impact of the convolutional kernel size on model performance, we conducted experiments with different convolutional kernel sizes. Table 3 shows a performance comparison of various 1D-CNN parameters on the training set. When the convolutional kernel is set to 2000, the *ACC* value is maximized, surpassing 1900 and 2100 by 2.2% and 1.9%, respectively. Additionally, the *MCC* value increased by 0.32% and 0.27%, compared to 1900 and 2100.

**Table 3.** Performance of model in different convolutional kernel sizes on the training set using ten-fold cross-validation experiments.

| 1D-CNN Kernel Sizes | *ACC* | *MCC* | *Sn* | *Sp* |
|---|---|---|---|---|
| 1900 | 0.968 | 0.95 | 0.951 | 0.977 |
| 2000 | 0.972 | 0.956 | 0.963 | 0.971 |
| 2100 | 0.952 | 0.937 | 0.941 | 0.960 |

For the LSTM model, we compared the model performance under different LSTM structures on the training set using ten-fold cross-validation, including 1-layer unidirectional and bidirectional LSTM, 2-layer unidirectional and bidirectional LSTM, and 3-layer unidirectional and bidirectional LSTM. The results are presented in Table 4.

**Table 4.** Performance of model in different LSTM Networks on the training set using ten-fold cross-validation experiments.

| Model | *ACC* | *MCC* | *Sn* | *Sp* |
|---|---|---|---|---|
| 1-layer unidirectional | 0.937 | 0.923 | 0.93 | 0.941 |
| 2-layer unidirectional | 0.949 | 0.935 | 0.953 | 0.947 |
| 3-layer unidirectional | 0.969 | 0.948 | 0.971 | 0.958 |
| 1-layer bidirectional | 0.958 | 0.941 | 0.943 | 0.967 |
| 2-layer bidirectional | 0.972 | 0.956 | 0.963 | 0.971 |
| 3-layer bidirectional | 0.964 | 0.949 | 0.957 | 0.964 |

As indicated in Table 4, a one-layer bidirectional LSTM model structure surpasses its unidirectional counterpart. Our analysis reveals that the unidirectional LSTM state transmission, occurring only from front to back, imposes a directional constraint on information propagation. In contrast, bidirectional LSTM can assimilate information in both forward and backward directions, thereby amplifying the neural network's expressive capacity. Concerning the layers of the Bi-LSTM model, the two-layer configuration attains superior performance. In comparison to the three-layer model, it achieves elevated values in *ACC*, *MCC*, *Sn*, and *Sp*, with improvements of 0.8%, 0.7%, 0.6%, and 0.7%, respectively.

In order to ensure the advancement of our model, we compare the proposed method with nine state-of-the-art approaches on the independent XUAMP test set, and the AUC values for each model were computed on the test set. The evaluated methods include amPEPpy [38], AMPfun [7], AMPEP [39], ADAM-HMM [40], AMPIR [41], AMPScannerV2 [42], AMPGram [43], Deep-AMPEP30 [44], CAMP-ANN [4], and sAMPpred-GAT [9]. The results are detailed in Table 5, and the AUC values are visually represented in Figure 3.

From Table 5, it can be noted that the model proposed in this study achieves the best performance in terms of *ACC*, *MCC*, and *Sn*. On the other hand, Deep-AMPEP30 attains the highest *Sp* but with a very small *Sn*, indicating that the model is heavily biased towards predicting positive samples. This suggests that the model has a weak generalization ability.

**Table 5.** Model performance on the XU independent test set.

| Model | *ACC* | *MCC* | *Sn* | *Sp* |
|---|---|---|---|---|
| amPEPpy | 0.679 | 0.431 | 0.400 | 0.958 |
| AMPfun | 0.674 | 0.414 | 0.406 | 0.943 |
| AMPEP | 0.661 | 0.429 | 0.330 | 0.992 |
| ADAM-HMM | 0.684 | 0.390 | 0.521 | 0.847 |
| Ampir | 0.563 | 0.156 | 0.266 | 0.859 |
| AMPScannerV2 | 0.568 | 0.137 | 0.523 | 0.613 |
| AmpGram | 0.564 | 0.131 | 0.445 | 0.682 |
| Deep-AMPEP30 | 0.533 | 0.183 | 0.065 | 1.0 |
| CAMP-ANN | 0.584 | 0.182 | 0.385 | 0.782 |
| sAMPpred-GAT | 0.715 | 0.464 | 0.530 | 0.9 |
| UniproLcad | 0.749 | 0.467 | 0.676 | 0.822 |

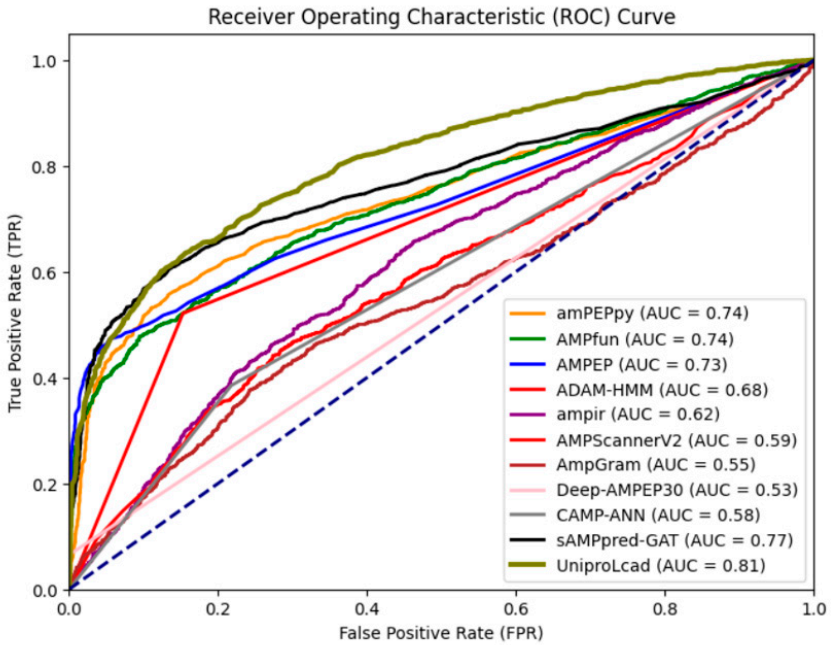


**Figure 3.** The AUC values for different models. The dotted line represents the random classifier, with an AUC value of 0.5. UniproLcad (army green) achieves the highest AUC value.

We also presented the AUC scores of all the mentioned models based on the XU independent test set in Figure 3. From Figure 3, it can be seen that the model proposed in this study achieves the highest AUC, indicating that the model possesses the best predictive capability and has a relatively low false positive rate. And confusion matrix of the model is shown in Figure 4.

To further evaluate the performance of our proposed model, we compared it with the latest method iAMP-Attenpred [10] using ten-fold cross-validation on the training sets Xingdataset1 and Xingdataset2. The experimental results are presented in Tables 6 and 7, respectively.

**Table 6.** Model performance on the Xingdataset1 training set.

| Model | *ACC* | *MCC* | *Sn* | *Sp* |
|---|---|---|---|---|
| UniproLcad | 0.980 | 0.9701 | 0.9769 | 0.9892 |
| iAMP-Attenpred | 0.983 | 0.9677 | 0.9791 | 0.9881 |

**Table 7.** Model performance on the Xingdataset2 training set.

| Model | *ACC* | *MCC* | *Sn* | *Sp* |
|---|---|---|---|---|
| UniproLcad | 0.9824 | 0.9507 | 0.9721 | 0.9867 |
| iAMP-Attenpred | 0.9776 | 0.9433 | 0.9573 | 0.9850 |

Tables 6 and 7 reveal that our proposed method UniproLcad and iAMP-Attenpred have achieved comparable performance results. Specifically, on the Xingdataset1 dataset, UniproLcad slightly outperforms iAMP-Attenpred in terms of *ACC* and *Sp*, while its performance is marginally lower than UniproLcad in *MCC* and *Sn*. On the Xingdataset2 dataset, UniproLcad is superior to iAMP-Attenpred across all metrics. The similar performance results achieved by the two methods may be attributed to their use of similar network architectures and protein language models.

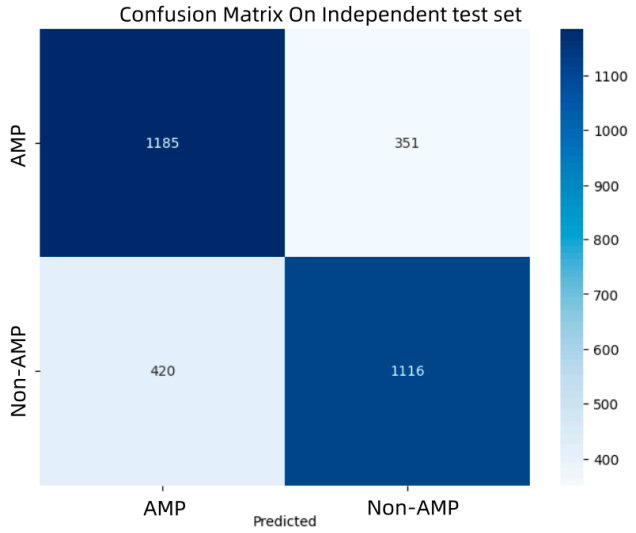


**Figure 4.** Confusion matrix of UniproLcad on XU independent test set.

## 4. Conclusions

In this study, we propose a predictor named UniproLcad, employing a deep learning framework and multiple protein language models for predicting AMPs. The main contributions of this paper are as follows: (1) Comprehensive representation of protein features: extracting and integrating multi-perspective features from the ESM-2, ProtBert, and UniRep models. (2) Building a prediction model using a stacked framework of multiple deep learning architectures, utilizing the stacking of 1D-CNN and Bi-LSTM networks to effectively extract high-dimensional features from integrated multi-perspective features that are challenging to capture. (3) Utilizing an attention mechanism to capture crucial information within the fused features, followed by multiple layers of MLP to obtain the prediction results for AMPs. The source code and data are available at https://github.com/harkic/UniproLcad (accessed on 6 March 2024).In future work, we aim to develop an openly accessible server to host the proposed model, facilitating researchers in conducting AMP prediction tasks more conveniently.

**Author Contributions:** X.W.: paper direction and suggestions. Z.W.: conceptualization, data curation, methodology, software, and supervision. Z.W., R.W. and X.G.: writing—original draft, validation, and test. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by funds from the Key Research Project of Colleges and Universities of Henan Province (No.22A520013, No.23B520004), the Key Science and Technology Development Program of Henan Province (No.232102210020, No.202102210144), and the Training Program of Young Backbone Teachers in Colleges and Universities of Henan Province (No.2019GGJS132).

**Data Availability Statement:** The source codes and data for UniproLcad are available at https://github.com//harkic/UniproLcad (accessed on 6 March 2024).

**Conflicts of Interest:** The authors declare there are no conflicts of interest.

## References


1. Murray, C.J.; Ikuta, K.S.; Sharara, F.; Swetschinski, L.; Aguilar, G.R.; Gray, A.; Han, C.; Bisignano, C.; Rao, P.; Wool, E.; et al. Global burden of bacterial antimicrobial resistance in 2019: A systematic analysis. *Lancet* **2022**, *399*, 629–655. [CrossRef] [PubMed]
2. Erdem Büyükkiraz, M.; Kesmen, Z. Antimicrobial peptides (AMPs): A promising class of antimicrobial compounds. *J. Appl. Microbiol.* **2022**, *132*, 1573–1596. [CrossRef] [PubMed]
3. Kumar, P.; Kizhakkedathu, J.; Straus, S. Antimicrobial Peptides: Diversity, Mechanism of Action and Strategies to Improve the Activity and Biocompatibility In Vivo. *Biomolecules* **2018**, *8*, 4. [CrossRef] [PubMed]
4. Joseph, S.; Karnik, S.; Nilawe, P.; Jayaraman, V.K.; Idicula-Thomas, S. ClassAMP: A Prediction Tool for Classification of Antimicrobial Peptides. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2012**, *9*, 1535–1538. [CrossRef]
5. Sharma, R.; Shrivastava, S.; Kumar Singh, S.; Kumar, A.; Saxena, S.; Kumar Singh, R. AniAMPpred: Artificial intelligence guided discovery of novel antimicrobial peptides in animal kingdom. *Brief. Bioinform.* **2021**, *22*, bbab242. [CrossRef] [PubMed]
6. Kavousi, K.; Bagheri, M.; Behrouzi, S.; Vafadar, S.; Atanaki, F.F.; Lotfabadi, B.T.; Ariaeenejad, S.; Shockravi, A.; Moosavi-Movahedi, A.A. IAMPE: NMR-Assisted Computational Prediction of Antimicrobial Peptides. *J. Chem. Inf. Model.* **2020**, *60*, 4691–4701. [CrossRef] [PubMed]
7. Chung, C.R.; Kuo, T.R.; Wu, L.C.; Lee, T.Y.; Horng, J.T. Characterization and identification of antimicrobial peptides with different functional activities. *Brief. Bioinform.* **2020**, *21*, 1098–1114. [CrossRef] [PubMed]
8. Xu, J.; Li, F.; Li, C.; Guo, X.; Landersdorfer, C.; Shen, H.H.; Peleg, A.Y.; Li, J.; Imoto, S.; Yao, J.; et al. iAMPCN: A deep-learning approach for identifying antimicrobial peptides and their functional activities. *Brief. Bioinform.* **2023**, *24*, bbad240. [CrossRef] [PubMed]
9. Yan, K.; Lv, H.; Guo, Y.; Peng, W.; Liu, B. sAMPpred-GAT: Prediction of antimicrobial peptide by graph attention network and predicted peptide structure. *Bioinformatics* **2023**, *39*, btac715. [CrossRef]
10. Xing, W.; Zhang, J.; Li, C.; Huo, Y.; Dong, G. iAMP-Attenpred: A novel antimicrobial peptide predictor based on BERT feature extraction method and CNN-BiLSTM-Attention combination model. *Brief. Bioinform.* **2023**, *25*, bbad443. [CrossRef]
11. Xu, J.; Li, F.; Leier, A.; Xiang, D.; Shen, H.H.; Marquez Lago, T.T.; Li, J.; Yu, D.J.; Song, J. Comprehensive assessment of machine learning-based methods for predicting antimicrobial peptides. *Brief. Bioinform.* **2021**, *22*, bbab083. [CrossRef] [PubMed]
12. Ofer, D.; Brandes, N.; Linial, M. The language of proteins: NLP, machine learning & protein sequences. *Comput. Struct. Biotechnol. J.* **2021**, *19*, 1750–1758. [PubMed]
13. Ferruz, N.; Höcker, B. Controllable protein design with language models. *Nat. Mach. Intell.* **2022**, *4*, 521–532. [CrossRef]
14. Wang, G.; Li, X.; Wang, Z. APD3: The antimicrobial peptide database as a tool for research and education. *Nucleic Acids Res.* **2016**, *44*, D1087–D1093. [CrossRef] [PubMed]
15. Zhao, X.; Wu, H.; Lu, H.; Li, G.; Huang, Q. LAMP: A Database Linking Antimicrobial Peptides. *PLoS ONE* **2013**, *8*, e66557. [CrossRef] [PubMed]
16. Thomas, S.; Karnik, S.; Barai, R.S.; Jayaraman, V.K.; Idicula-Thomas, S. CAMP: A useful resource for research on antimicrobial peptides. *Nucleic Acids Res.* **2010**, *38* (Suppl. 1), D774–D780. [CrossRef] [PubMed]
17. Gogoladze, G.; Grigolava, M.; Vishnepolsky, B.; Chubinidze, M.; Duroux, P.; Lefranc, M.P.; Pirtskhalava, M. DBAASP: Database of antimicrobial activity and structure of peptides. *FEMS Microbiol. Lett.* **2014**, *357*, 63–68. [CrossRef] [PubMed]
18. Kang, X.; Dong, F.; Shi, C.; Liu, S.; Sun, J.; Chen, J.; Li, H.; Xu, H.; Lao, X.; Zheng, H. DRAMP 2.0, an updated data repository of antimicrobial peptides. *Sci. Data* **2019**, *6*, 148. [CrossRef]
19. Jhong, J.H.; Chi, Y.H.; Li, W.C.; Lin, T.H.; Huang, K.Y.; Lee, T.Y. dbAMP: An integrated resource for exploring antimicrobial peptides with functional activities and physicochemical properties on transcriptome and proteome data. *Nucleic Acids Res.* **2019**, *47*, D285–D297. [CrossRef]
20. Fu, L.; Niu, B.; Zhu, Z.; Wu, S.; Li, W. CD-HIT: Accelerated for clustering the next-generation sequencing data. *Bioinformatics* **2012**, *28*, 3150–3152. [CrossRef]
21. Zhang, Y.; Lin, J.; Zhao, L.; Zeng, X.; Liu, X. A novel antibacterial peptide recognition algorithm based on BERT. *Brief. Bioinform.* **2021**, *22*, bbab200. [CrossRef] [PubMed]
22. Fjell, C.D.; Hancock, R.E.; Cherkasov, A. AMPer: A database and an automated discovery tool for antimicrobial peptides. *Bioinformatics* **2007**, *23*, 1148–1155. [CrossRef]
23. Lee, H.T.; Lee, C.C.; Yang, J.R.; Lai, J.Z.; Chang, K.Y. A large-scale structural classification of antimicrobial peptides. *Biomed. Res. Int.* **2015**, *2015*, 475062. [CrossRef] [PubMed]
24. Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **2023**, *379*, 1123–1130. [CrossRef]
25. Alley, E.C.; Khimulya, G.; Biswas, S.; AlQuraishi, M.; Church, G.M. Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **2019**, *16*, 1315–1322. [CrossRef] [PubMed]

26. Brandes, N.; Ofer, D.; Peleg, Y.; Rappoport, N.; Linial, M. ProteinBERT: A universal deep-learning model of protein sequence and function. *Bioinformatics* **2022**, *38*, 2102–2110. [CrossRef] [PubMed]
27. Zaremba, W.; Sutskever, I.; Vinyals, O. Recurrent Neural Network Regularization. *arXiv* **2015**, arXiv:1409.2329.
28. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
29. Wang, X.; Ding, Z.; Wang, R.; Lin, X. Deepro-Glu: Combination of convolutional neural network and Bi-LSTM models using ProtBert and handcrafted features to identify lysine glutarylation sites. *Brief. Bioinform.* **2023**, *24*, bbac631. [CrossRef]
30. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv* **2015**, arXiv:1508.01991.
31. O'shea, K.; Nash, R. An Introduction to Convolutional Neural Networks. *arXiv* **2015**, arXiv:1511.08458.
32. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent Advances in Convolutional Neural Networks. *arXiv* **2017**, arXiv:1512.07108. [CrossRef]
33. Peng, Y.; He, X.; Zhao, J. Object-Part Attention Model for Fine-grained Image Classification. *IEEE Trans. Image Proc.* **2017**, *27*, 1487–1500. [CrossRef] [PubMed]
34. Gao, S.; Ramanathan, A.; Tourassi, G. Hierarchical Convolutional Attention Networks for Text Classification. In Proceedings of the Third Workshop on Representation Learning for NLP, Melbourne, Australia, 20 July 2018; pp. 11–23.
35. Ni, Y.; Fan, L.; Wang, M.; Zhang, N.; Zuo, Y.; Liao, M. EPI-Mind: Identifying Enhancer-Promoter Interactions Based on Transformer Mechanism. *Interdiscip. Sci. Comput. Life Sci.* **2022**, *14*, 786–794. [CrossRef] [PubMed]
36. Park, S.; Koh, Y.; Jeon, H.; Kim, H.; Yeo, Y.; Kang, J. Enhancing the interpretability of transcription factor binding site prediction using attention mechanism. *Sci. Rep.* **2020**, *10*, 13413. [CrossRef] [PubMed]
37. Bae, K.; Ryu, H.; Shin, H. Does Adam optimizer keep close to the optimal point? *arXiv* **2019**, arXiv:1911.00289.
38. Lawrence, T.J.; Carper, D.L.; Spangler, M.K.; Carrell, A.A.; Rush, T.A.; Minter, S.J.; Weston, D.J.; Labbé, J.L. amPEPpy 1.0: A portable and accurate antimicrobial peptide prediction tool. *Bioinformatics* **2021**, *37*, 2058–2060. [CrossRef] [PubMed]
39. Bhadra, P.; Yan, J.; Li, J.; Fong, S.; Siu, S.W. AmPEP: Sequence-based prediction of antimicrobial peptides using distribution patterns of amino acid properties and random forest. *Sci. Rep.* **2018**, *8*, 1697. [CrossRef] [PubMed]
40. Eddy, S.R. Profile hidden Markov models. *Bioinformatics* **1998**, *14*, 755–763. [CrossRef]
41. Fingerhut, L.C.H.W.; Miller, D.J.; Strugnell, J.M.; Daly, N.L.; Cooke, I.R. ampir: An R package for fast genome-wide prediction of antimicrobial peptides. *Bioinformatics* **2021**, *36*, 5262–5263. [CrossRef]
42. Veltri, D.; Kamath, U.; Shehu, A. Deep Learning Improves Antimicrobial Peptide Recognition. *Bioinformatics* **2018**, *34*, 2740–2747. [CrossRef] [PubMed]
43. Burdukiewicz, M.; Sidorczuk, K.; Rafacz, D.; Pietluch, F.; Chilimoniuk, J.; Rödiger, S.; Gagat, P. Proteomic Screening for Prediction and Design of Antimicrobial Peptides with AmpGram. *Int. J. Mol. Sci.* **2020**, *21*, 4310. [CrossRef] [PubMed]
44. Yan, J.; Bhadra, P.; Li, A.; Sethiya, P.; Qin, L.; Tai, H.K.; Wong, K.H.; Siu, S.W. Deep-AmPEP30: Improve Short Antimicrobial Peptides Prediction with Deep Learning. *Mol. Ther.-Nucleic Acids* **2020**, *20*, 882–894. [CrossRef] [PubMed]