JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 11 Issue 01
2026

Research Article
https://doi.org/10.56979/1101/2026/1361

LLM-Pep: Targeted Modeling for Anti-Parasitic Peptide Detection Using Large Language Models

Aqsa Amjad[1], Faria Nazir[2*], Tayyaba Anees[2], Nosheen Qamar[2], and Wajeeha Khalil[3]

[1]Department of Computer Science, University of Management and Technology, Lahore, Punjab, Pakistan.
[2]Department of Software Engineering, University of Management and Technology, Lahore, Punjab, Pakistan.
[3]Department of Computer Science, University of Engineering and Technology, Peshawar, KPK, Pakistan.
[*]Corresponding Author: Faria Nazir. Email: faria.nazir@umt.edu.pk

Received: January 29, 2026 **Accepted:** May 15, 2026

Abstract: Parasites pose serious threats to host organisms, and anti-parasitic peptides (APPs) have shown potential in inhibiting parasite growth and reproduction. However, traditional biological screening methods such as nanomedicine-based assays and organism-based approaches are costly and time-consuming, highlighting the need for efficient computational prediction methods. In our study, we introduce a two-stage machine learning framework for accurate APP identification. To handle the class imbalance in the training data, we apply a random under sampling strategy to construct a balanced training set. Next, peptide sequences are encoded using pre-trained large language model-based embed dings and classified using a multi-layer perceptron (MLP) model. Unlike existing approaches that suffer from limited feature representation and poor generalization on independent datasets, our method leverages deep contextual sequence embedding combined with data balancing to improve robustness. Experimental results demonstrate that our model achieves an accuracy of 91.7% and an AUC of 0.939 on independent test sets, surpasses the existing approaches in APP prediction.

Keywords: Computational Intelligence; bioinformatics; anti-parasitic peptides; pre-trained language models; multi-layer perceptron

1. Introduction

Parasites are a significant cause of health issues, affecting nearly every being, along with flora and fauna. Leech-like infections may lead to a wide range of complications, from mild discomfort to life-threatening conditions [1]. Parasitic diseases place a significant burden on humanity, claiming the lives of over 400,000 people with chronic conditions each year worldwide. Antibiotics are the most widely used treatment for these infections today. However, frequent use of antibiotics can lead to unintended side effects and contribute to the development of parasite resistance [2, 3]. This highlights the compelling necessity for alternative, productive treatments. Current work reveals that curative peptides have the potential to eliminate or inhibit parasites infecting humans, such as *Plasmodium* or *Leishmania*. Peptide-based drugs offer several advantages over traditional antibiotics, including reduced production, more specificity, less lethality, and excellent cell entry capabilities [4]. Anti-parasitic peptides (APPs) are generally short, consisting of 5 to 50 amino acids. They are often derived or modified from antimicrobial peptides (AMPs), which are known for their ability to disrupt parasite functions effectively [5-8]. APPs work by targeting and destroying specific organisms. They may reduce the parasite's cell membrane or hinder key enzymes, such as reductase, essential for the parasite's survival [9, 10]. As a result, APPs are considered promising therapeutic candidates for treating parasitic diseases. However, identifying APPs through traditional diagnostic methods is both costly and time-consuming. To address this, computational approaches offer efficient and complementary solutions for large-scale APP exploration and analysis.

High-performing computational methods depend on large, high-quality datasets. Currently, there is just a single dedicated anti-parasitic peptide database, ParaPep [10], that contains 519 exploratory substantiated APPs. Additionally, several antimicrobial peptide (AMP) databases, including APD3 [11], dbAMP [12], CAMP [13], DRAMP [14], and ADAM [15], also provide structures and exploratory substantiated APPs sequences, offering valuable resources for further research. Over the past decade, several machine learning (ML) based models have been introduced to recognize therapeutic peptides. Examples include AAPred-CNN [16] for anti-angiogenic peptides, mAHTPred [17] for anti-hypertensive peptides, and AVPIden [18] for anti-viral peptides. PredictFP2 [19] specializes in identifying peptide regions across retroviruses, while AMPfun [20], a random forest-based tool, is designed to predict anticancer, anti-parasitic (APP), and antiviral peptides [21]. Although AMPfun effectively characterizes antimicrobial peptides with diverse properties, its prediction performance for APPs remains suboptimal.

In 2021, PredAPP [22] was introduced as a method for predicting APPs using an undersampling and batch technique. To address data imbalance, various under-sampling approaches were proposed. This predictor used a different approach, integrating nine features with six different machine learning classifiers. In 2022, the i2APP [23] model was developed to identify APPs, employing a two-stage machine learning architecture for enhanced prediction accuracy. In the first stage, multiple feature groups are taken from all peptide sequences, and those feature groups are used to train first-layer classifiers. The outcomes from the first-stage algorithms serve as the top-most features, which are then fed into a second-layer classifier in the second stage. The outputs of this second layer represent the final predictions for detecting APPs. Parasitic infections remain a major global health burden, particularly in developing regions, contributing to significant morbidity and mortality. APPs have emerged as a promising class of therapeutic agents due to their ability to directly attack parasites and inhibit their development and reproduction. However, the experimental identification and validation of APPs using traditional biological methods such as nanomedicine and entomopathogenic agents are labor-intensive, time-consuming, and costly. Moreover, the scarcity of computational tools that can accurately predict APPs from peptide sequences limits the pace of discovery in this field. This research is motivated by the urgent need for an efficient, scalable, and accurate computational approach to identify APPs, thereby accelerating therapeutic peptide discovery and aiding in the fight against parasitic diseases.

In this study, we developed an improved predictor with high efficacy for predicting APPs using sequence information derived from protTrans T5, protBERT, ProtBERT BFD and ESM-1v feature encoding approaches. In contrast to prior studies relying on shallow, manually engineered descriptors, our use of advanced LLMs enables the model to learn context-aware, biologically meaningful patterns, capturing complex sequence dependencies, biochemical patterns, and structural cues inherent in peptide sequences, far beyond what handcrafted features can achieve, leading to state-of-the-art accuracy in anti-parasitic peptide prediction. The developed architecture of the ML-based model consists of two stages. In the first stage, it extracted large language model features from peptides, and in the second stage, it fed these extracted features into the model to predict anti-parasitic peptides. The schematic layout of the approach is depicted in Figure 1. The contribution of the present research work can be summarized as follows:

a) We designed an intelligent two-step model that extracts the high-level features in the first step and predicts APPs in the second step, respectively.
b) We captured the peptides encoded patterns using a pre-trained large language model.
c) We enhanced the overall prediction performance of APPs on both balanced and unbalanced datasets.

2. Materials and Methods

2.1. Benchmark Dataset

The dataset used in this study to train the model was derived from previous research [24] and compared with others. To create an unbiased dataset, homologous sequences were filtered out using CD-HIT [25]. For positive samples, a 90% sequence identity threshold was applied, while for negative samples, a 60% threshold was used [23]. The 90% threshold for positive samples was chosen for the proportionate number of positive data available. After removing homologous sequences, 301 APPs were selected as positive data, and 1909 non- APPs were chosen as negative samples.

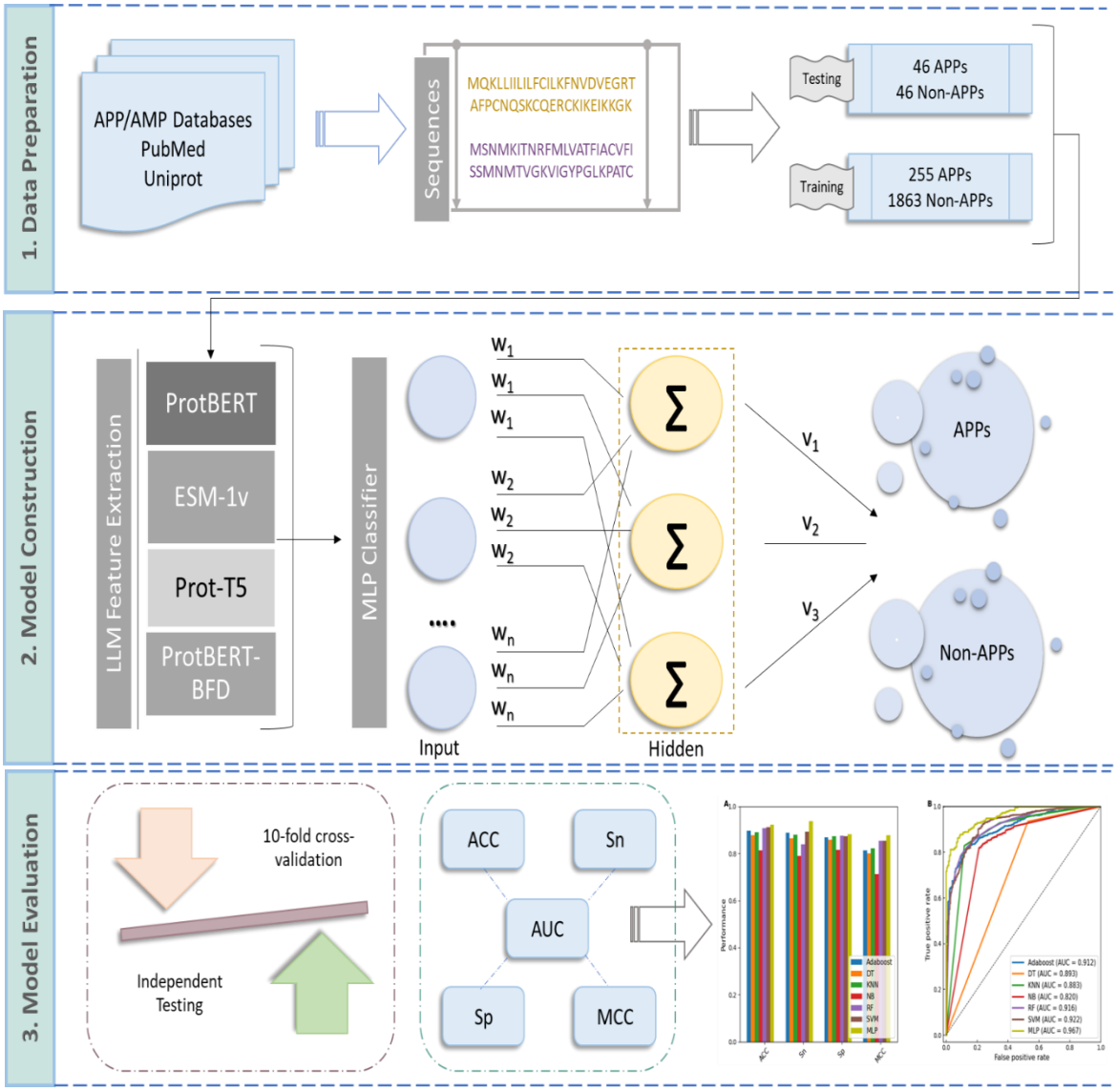

Figure 1. Illustration of the proposed model.

From the positive data, 46 APPs were set aside as testing data, while the remaining 255 APPs were used for training. For the negative samples, 46 non-APPs were randomly selected as testing data, and the next 1863 non- APPs were designated for training. In a result, the first training set consisted of 255 APPs and 1863 non- APPs, while the testing data included 46 APPs and 46 non- APPs. In Table 1, the statistical description of both datasets is given below:

Table 1. Datasets used in APPs predictors

Type	Dataset	Samples	CD-HIT threshold
Positive	Training	255	0.90
Negative		1863	
Positive	Testing	46	0.60
Negative		46	

2.2. Feature Extraction using LLMs

Key attributes play a crucial role in training deep learning models and ensuring they achieve strong prediction performance. Peptides are typically categorized according to the feature set derived from their systemic and operable characteristics. Decoding features from peptide data that accurately show the underlying data patterns can be challenging. Transformer-based language models can capture contextual and semantic relationships within peptide sequences, enabling the extraction of biologically meaningful and rich latent representations that are difficult to obtain through traditional handcrafted features. This enhanced representation capability contributes to improved APP prediction performance and model generalization [26, 27]. The features utilized in this study are outlined below:

2.2.1. ProtTrans T5

In this research work, we have used ProtT5 [28] to extract features from amino acid sequences, a widely used pre-trained language model, has proven valuable for solving various biological problems. It has been applied to predict protein-protein interactions, protein structure, succinylation sites, binding residues, and more by analyzing matching amino acid sequences+. Both the heavy and light chains were processed independently but in the same manner [29]. For each chain, the input to ProtT5 was the amino acid sequence, and the embedding from the last layer of the encoder, which consists of a specific set of 1024-dimensional vectors, was used to represent the features as $Emb\ _{i=1,2,\ldots,L}^{aa_i}$

$$Emb\ _{i=1,2,\ldots,L}^{aa_i}\ \left[X_{aa_i,1},\ X_{aa_i,2},\ldots,X_{aa_i,n}\right] \quad (1)$$

where aa_i denotes the ith amino acid in the sequence and n=1024 denotes the embedding's dimension and $X_{aa_i,j}$ represents the j-th feature value (out of 1024 total features) for the i-th amino acid in the sequence. As a result, the amino acid sequence feature can be represented as a two-dimensional matrix using the formulas L*1024, represented as $\widehat{Emb_H}$ heavy chain and $\left(\widehat{Emb_L}\right)$ light chain.

$$\widehat{Emb_H}\ \left(or\ \widehat{Emb_L}\right) = \begin{bmatrix} V_{aa_1} \\ V_{aa_2} \\ \vdots \\ V_{aa_i} \\ \vdots \\ V_{aa_L} \end{bmatrix} = \begin{bmatrix} X_{aa_1,1}\ X_{aa_1,2}\ldots.\ X_{aa_1,n} \\ X_{aa_2,1}\ X_{aa_2,2}\ldots.\ X_{aa_2,n} \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ X_{aa_i,1}\ X_{aa_i,2}\ldots.\ X_{aa_i,n} \\ \vdots \quad\quad \vdots \quad\quad \vdots \\ X_{aa_L,1}\ X_{aa_L,2}\ldots.\ X_{aa_L,n} \end{bmatrix} \quad (2)$$

Where L is the sequence's length and V_{aa_i} represents the embedding vector of the i-th amino acid in the sequence. Additionally, the processes defined as follows are used to further process the matrix expressed in Formula (2):

$$Fea_{H1}\ (Fea_{L1}) = \left[\frac{\sum_{i=1}^{L} X_{aa_i,1}}{L}, \frac{\sum_{i=1}^{L} X_{aa_i,2}}{L}, \ldots, \frac{\sum_{i=1}^{L} X_{aa_i,n}}{L}\right] \quad (3)$$

Where, Fea_{H1} and Fea_{L1} represents the average of embeddings across the sequence for high and low embedding. Next, a one-dimensional feature vector with 2048 terms (2048 features) was created by successively connecting the feature vectors Fea_{H1} and Fea_{L1}. This feature vector is designed as Fea_{ab}:

$$Fea_{ab} = [\ Fea_{H1}, Fea_{L1}\ ] \quad (4)$$

2.2.2. ESM-1v

We have used ESM-1v transformers to extract features, based on a BERT-style encoder architecture and support a maximum input length of 1022 amino acids [30]. The formal ESM tokenization program automatically compresses small sequences to 1022, but during data processing, the sequence breadth is adjusted to reflect its true length. For that sequence which is larger than 1022 amino acids, if the alteration occurs within 1022 leftovers of either the N-terminus (start of protein sequence) or the C-terminus (end of protein sequence), the last 1022 residues are retained. If the alteration mark is more than 1022 residues away from both sides, 510 residues from the N side and 511 residues from the C-terminal side are chosen, following in sequence of exactly 1022 residues.

Modernizer-based pre-trained language models (pLMs) provide two key types of features: the likelihood of all amino acid categories taking place at all sides in the sequence, and a compact matrix embedded for all places in the sequence. The self-attention approach of the Transformer structure allows the embeddings to capture background knowledge from the whole 1022 leftover window. In this study,

the same method is adopted as ESM-Variant, extracting both the log likelihood ratio (LLR) and the embedded matrix for both wild kind and mutant kind residues at each modified site [31, 32]. The LLR is measured by the ESM model's probability for both the mutant and wild-type amino acids at the earmark site, constrained on the predictor having the wild-type sequence as input.

2.2.3. ProtBERT

In this study, BERT [33] is used to extract features, which is trained on a large language corpus and has attained modern outcomes on eleven natural language processing (NLP) tasks. The flow of BERT is based on a bi-directional Transformer encoder with multiple layers. In the BERT base architecture, each layer contains 12 encoder blocks, while the BERT-large model has 24 encoder blocks per layer. Each layer is composed of two main sub-stages: a fully connected feed-forward stage and a multi-head self-attention stage. After each sub-stage, a residual connection is applied, followed by layer normalization to stabilize training and improve performance.

ProtBert [34] is a novel natural language processing (NLP) model introduced by Elnaggar et al., which was developed by regulating the foremost BERT archetype on protein sequence from the UniRef100 and Big Fantastic Database (BFD). The BFD database consolidates each protein sequence from the UniProt database, along with translated proteins from various metagenomic sequencing projects. UniRef100, a widely used reference library, contains curated protein sequences. To enhance performance on downstream supervised tasks, ProtBert increased the number of layers to thirty [35]. The researchers analyzed the effectiveness of ProtBert through the tasks: forecasting secondary form, intracellular region, and membrane binding [36].

2.3. Learning Method

Various classification approaches are employed in this study to accurately recognize anti-parasitic peptides. Machine learning classifiers, including Adaboost, Decision Tree, K-Nearest Neighbor, Naïve Bayes, Random Forest, and Support Vector Machine used for comparison with the predictor classifier, and the Multi-Layer Perceptron algorithm was used to make a predictor, which is a deep learning algorithm.

2.3.1. Multilayer Perceptron

The Multi-Layer Perceptron (MLP) classifier is a versatile model that uses artificial neural networks to classify peptides, specifically APPs. The MLP consists of 3 most important layers: the input layer, hidden layers, and the output layer [37]. In peptide classification, the input layer receives feature representations of the peptides. These features can be derived from the amino acid composition, sequence motifs, or embeddings from pre-trained models like ProtBert or ProtT5. The hidden layers are where the model learns complex, non-linear patterns between the features and the peptide's potential anti-parasitic activity. For binary classification, such as identifying APPs from non-APPs, the final layer has one neuron unit, using a sigmoid activation function to outcome a probability score.

Training the MLP involves passing the peptide data through the network during forward propagation, where the model generates predictions. The loss function, such as binary cross-entropy for binary classification, then calculates the error by comparing the predicted output to the actual label. Backpropagation is then used to update the weights of the network by moving the error backwards using the layers, helping the model learn from its mistakes. This process is repeated for a number of epochs until the model converges, minimizing the loss and making it capable of generalizing well to new, unseen data. The forward pass is given by:

$$\hat{y} = W_{k+1}\sigma(W_k\sigma(\dots\sigma(W_1x + b_1)\dots) + b_k) + b_{k+1} \quad (5)$$

where W and b represents linear stretching and shifting, and σ represents non-linear bending.

The Multi-Layer Perceptron (MLP) was chosen over other deep learning models due to its simplicity, computational efficiency, and strong ability to model non-linear relationships, which are critical in peptide sequence classification. Unlike more complex models such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), MLPs do not require extensive sequence modeling or spatial hierarchies, making them ideal for high-dimensional feature representations extracted from language-based models. Furthermore, MLPs are less prone to overfitting when trained on relatively smaller or imbalanced datasets, especially when regularized and fine-tuned with optimal hyperparameters. In our study, with a learning rate of 0.001, 60 epochs, batch size of 64, and the ADAM optimizer, the MLP achieved

strong generalization and outperformed more complex architectures in terms of both accuracy and AUC. The learning rate, batch size and number of epochs (among other hyperparameters) were chosen using empirical methods, based on preliminary experiments using validation performance and convergence stability as criteria. We explored several configurations and selected the final parameters based on best predictive performance with minimal overfitting. This makes MLP a balanced choice between performance and model interpretability, suitable for the task at hand.

3. Results

The simulated results of the proposed model have been thoroughly evaluated from multiple perspectives to demonstrate its effectiveness across various performance metrics.

3.1. Evaluation Metrics

Numerous performance assessment metrics may be used to assess the effectiveness of prediction models [38]. These metrics provide measurable evaluations of the model's performance on the validation part at each cross-validation iteration [39-42]. Several commonly used performance assessment metrics were used in this work, including receiver operating curve-area under the curve (AUC-ROC), Mathew's correlation coefficient (MCC), accuracy (ACC), sensitivity (Sn), and specificity (Sp) [43].

$$ACC = \frac{TP + TN}{TP + FP + FN + TN} \tag{6}$$

$$Sn = \frac{TP}{TP + FP} \tag{7}$$

$$Sp = \frac{TP}{FP + TN} \tag{8}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{[TP + FP][TP + FN][TN + FP][TN + FN]}} \tag{9}$$

Where TP represents true positive value, TN true negative value, FP false positive value and FN false negative value.

3.2. Comparison with different classifiers

A classifier is the central component of machine learning, if feature extraction is its foundation. However, machine learning is becoming more and more popular in biological sequence analysis and prediction due to the massive amount of genomic data that has been gathered and the groundbreaking development of artificial intelligence algorithms. To predict and identify APPs, classifiers including, Adaboost [44], Decision Tree (DT) [45], K-Nearest Neighbor (KNN) [46], Naïve Bayes (NB) [47], Random Forest (RF) [48] and support Vector Machine (SVM) [49] have been used for comparison with our proposed model with respect to both cross-validation and independent test performance. In Table 1, common machine learning algorithms have been evaluated and compared with the new approach on the training dataset in which we have analyzed that our model results are better than ML algorithms. Our method obtains an accuracy of 92% and MCC of 0.877, respectively. In Table 2, same machine learning algorithms have been evaluated and compared with our model on testing dataset in which we have analyzed that our model results are higher than commonly used ML algorithms. Our method obtains accuracy of 91% and MCC of 0.869, respectively.

Table 2. Comparison of learning algorithms using 10-fold CV on the training dataset.

Approach	ACC	MCC	Sn	Sp	AUC
Adaboost	0.897	0.813	0.889	0.870	0.912
DT	0.878	0.801	0.865	0.859	0.893
KNN	0.890	0.821	0.879	0.874	0.883
NB	0.813	0.713	0.790	0.816	0.820
RF	0.908	0.855	0.840	0.876	0.916
SVM	0.911	0.854	0.893	0.874	0.922
OUR	**0.922**	**0.877**	**0.937**	**0.882**	**0.967**

Furthermore, these machine learning algorithms outdo common approaches in the context of model accuracy and learning efficacy. The introduced method utilizes artificial neural networks to classify peptides, which offers the benefits of a high capturing volume and fast processing, as opposed to using a machine learning model. Our model exhibits improved prediction outcomes when compared to other models. Figure 2(a) is the depiction of training dataset results, in which it is clearly shown that our model achieves the highest accuracy and some other metrics as well. Figure 2(b) shows the highest ROC curve of the new approach with a surrounding area of 0.967 in comparison with machine learning algorithms. When compared to the conventional models, we leverage MLP's margin of attribute depletion to efficiently capture extra significant features. There is ample statistics to conclude that our model outperforms regular networks in terms of performance and ability to learn more information.

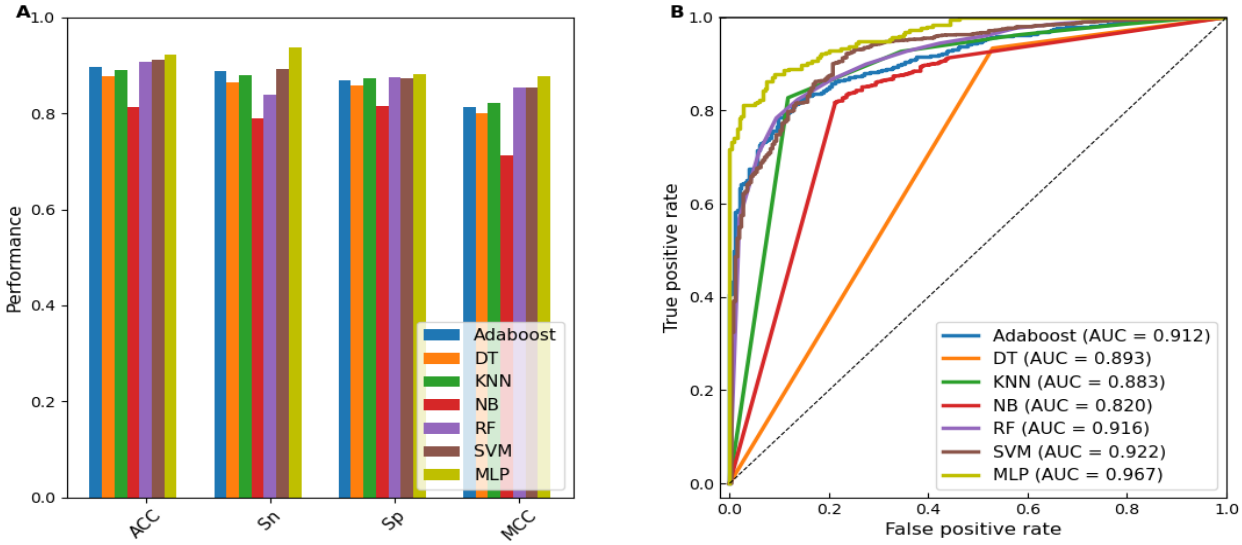

Figure 2. Performance comparison between various classification algorithms and proposed model.

Table 3. Comparison of classification algorithms on an independent dataset.

Approach	ACC	MCC	Sn	Sp	AUC
Adaboost	0.855	0.794	0.870	0.834	0.890
DT	0.861	0.798	0.836	0.872	0.902
KNN	0.818	0.743	0.849	0.825	0.861
NB	0.792	0.701	0.770	0.784	0.836
RF	0.874	0.810	0.854	0.863	0.912
SVM	0.899	0.823	0.857	0.871	0.930
OUR	**0.917**	**0.869**	**0.954**	**0.876**	**0.939**

3.3. Comparison with existing predictors

The predictor developed in this study is compared using 10-fold cross-validation on the training data to existing developed predictors, with the results presented in Table 3. The NM-BD predictor and RUS-BD predictor, introduced by [24], involve downsampling the imbalanced training set. The NearMiss method was used for one-time sampling, and the arbitrarily undersampling approach was applied for the bottommost layer, which is used in this research. Random undersampling was chosen due to its simplicity, computational efficiency, and effectiveness in reducing majority-class bias while maintaining balanced class representation during training. Additionally, compared to synthetic oversampling approaches, random undersampling avoids generating artificial peptide samples that may alter the original biological sequence distribution and potentially introduce noise into the learning process. By making the comparison to i2APP, the method proposed in this work outdoes it across each metric, showing improvements of 2.2% in accuracy (ACC), 0.5% in sensitivity (Sn), 1.3% in specificity (Sp), and 0.74 in Matthews correlation coefficient (MCC). When making comparisons to NM-BD, the method proposed in this work is superior

on about each metric, excluding specificity (Sp). The outcomes demonstrate that the introduced method performs better than the others across the board on the training dataset.

Table 4. Analogy of the proposed approach with the current predictors via the benchmark dataset.

Dataset	Methods	ACC	MCC	Sn	Sp
Training	NM-BD	0.888	0.778	0.855	0.922
	RUS-BD	0.882	0.768	0.925	0.839
	i2APP	0.900	0.803	0.932	0.869
	Proposed	**0.922**	**0.877**	**0.937**	0.882

Also, to examine the efficiency of the method used in this work, we equate it with existing approaches on independent data, with the outcomes presented in Table 4. The metrics demonstrate that our proposed model outperforms the others across nearly all measures. Specifically, our model achieves improvements of 0.4% in accuracy (ACC), 0.036 in Matthews correlation coefficient (MCC), compared to i2APP. When compared to PredAPP, the proposed model shows increases of 3.37% in accuracy (ACC) and 0.93% in MCC. These results clearly illustrate that the proposed model possesses superior generalization capability compared to the current methodologies for APPs prediction.

Table 5. Analysis of the proposed approach with the existing models via independent dataset.

Dataset	Methods	ACC	MCC	Sn	Sp
Testing	AMPfun	0.739	0.531	0.522	0.957
	PredAPP	0.880	0.776	0.978	0.783
	i2APP	0.913	0.833	0.978	0.848
	Proposed	**0.917**	**0.869**	0.954	0.876

Figure 3 illustrates the comparative performance analysis between the proposed model and existing APP prediction methods on both training and independent testing datasets. Figure 3(a) presents the comparison on the training dataset, where the proposed model achieves superior performance across key evaluation metrics, including accuracy (ACC), Matthews correlation coefficient (MCC), sensitivity (Sn), and specificity (Sp). Similarly, Figure 3(b) demonstrates the evaluation on the independent testing dataset, showing that the proposed model maintains strong generalization capability and consistently outperforms existing predictors such as AMPfun, PredAPP, and i2APP. These visual comparisons further confirm the robustness and effectiveness of the proposed LLM-based framework for APP prediction.

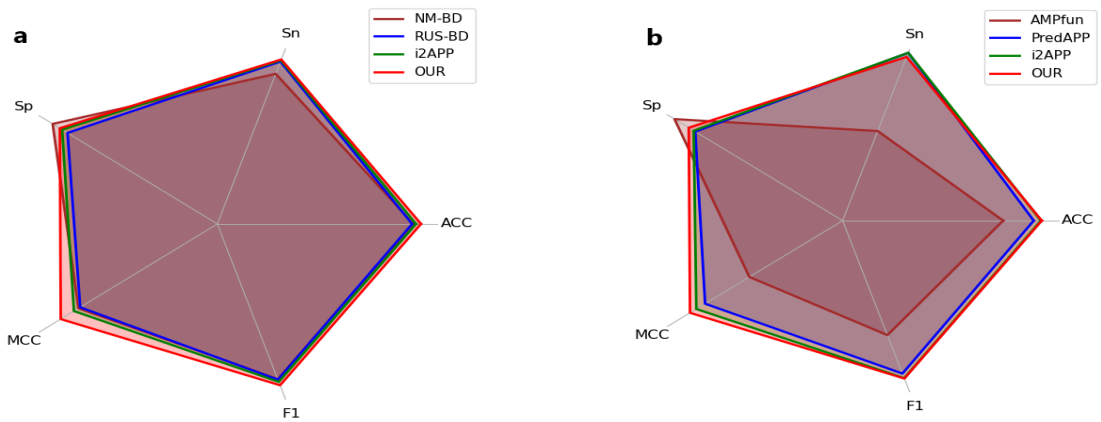

Figure 3. Performance comparison between existing and current predictor.

3.4. Impact of Dataset Balancing

Table 5 presents the results of 10-fold cross-validation performed on both balanced and unbalanced datasets. The unbalanced dataset contained 255 APPs and 1863 non-APPs. Although the unbalanced dataset achieved higher ACC and Sp values due to the dominance of negative samples, the Sn metric decreased considerably, indicating weaker recognition of APP sequences. In contrast, the balanced dataset significantly improved sensitivity (Sn) and MCC, demonstrating enhanced minority-class detection and reduced classification bias toward non-APP samples. These results highlight the importance of dataset balancing in improving model generalization and achieving more reliable prediction performance for APP identification. Furthermore, the proposed model outperformed PredAPP (unbalanced) across all evaluation metrics and demonstrated competitive performance compared to i2APP.

Table 6. The results of ten-fold cross validation on the unbalanced datasets.

Methods	ACC	MCC	Sn	Sp
PredAPP (unbalanced)	0.919	0.574	0.525	0.973
i2APP(unbalanced)	0.965	0.826	0.767	0.993
Proposed (unbalanced)	**0.969**	0.835	0.671	0.938
Proposed (balanced)	0.922	**0.877**	**0.937**	0.882

4. Conclusions

In order to effectively identify APPs, we suggest an innovative approach in this work. This work's primary structure is made up of many significant phases. First, the training set is balanced using the random under sampling technique. Second, peptide sequences are used to extract a range of large language model-based characteristics, which are subsequently fed into the MLP classifier.

Ultimately, independent testing of the suggested model and the others reveals that it surpassess the existing methods for APP prediction in terms of generality. Large language models are employed to extract all of the sequence characteristics used in this work. The exactness of APP identification may be further increased in years to come by using the RNN or Transformer method for automated feature as data volumes grow.

Data Availability Statement:

Data is available at the GitHub https://github.com/xialab-ahu/PredAPP repository

Funding:

No funding was received for this research

Conflict of Interests:

The author declares no conflict of interest.

References

1. Lacerda, A.F., et al., Anti-parasitic Peptides from Arthropods and their Application in Drug Therapy. Front Microbiol, 2016. 7: p. 91.
2. Alsford, S., et al., High-throughput decoding of antitrypanosomal drug efficacy and resistance. Nature, 2012. 482(7384): p. 232-6.
3. Boman, H.G., Antibacterial peptides: key components needed in immunity. Cell, 1991. 65(2): p. 205-7.
4. Rivera-Fernández, N., et al., Bioactive Peptides against Human Apicomplexan Parasites. Antibiotics (Basel), 2022. 11(11).
5. Wang, G., Vaisman, II, and M.L. van Hoek, Machine Learning Prediction of Antimicrobial Peptides. Methods Mol Biol, 2022. 2405: p. 1-37.
6. Mangoni, M.L., A.M. McDermott, and M. Zasloff, Antimicrobial peptides and wound healing: biological and therapeutic considerations. Exp Dermatol, 2016. 25(3): p. 167-73.
7. Sørensen, O.E., N. Borregaard, and A.M. Cole, Antimicrobial peptides in innate immune responses. Contrib Microbiol, 2008. 15: p. 61-77.
8. Pasupuleti, M., A. Schmidtchen, and M. Malmsten, Antimicrobial peptides: key components of the innate immune system. Crit Rev Biotechnol, 2012. 32(2): p. 143-71.
9. Day, T.A. and A.G. Maule, Parasitic peptides! The structure and function of neuropeptides in parasitic worms. Peptides, 1999. 20(8): p. 999-1019.
10. Mehta, D., et al., ParaPep: a web resource for experimentally validated antiparasitic peptide sequences and their structures. Database (Oxford), 2014. 2014.
11. Wang, G., X. Li, and Z. Wang, APD3: the antimicrobial peptide database as a tool for research and education. Nucleic Acids Res, 2016. 44(D1): p. D1087-93.
12. Jhong, J.H., et al., dbAMP 2.0: updated resource for antimicrobial peptides with an enhanced scanning method for genomic and proteomic data. Nucleic Acids Res, 2022. 50(D1): p. D460-d470.
13. Waghu, F.H., et al., CAMP: Collection of sequences and structures of antimicrobial peptides. Nucleic Acids Res, 2014. 42(Database issue): p. D1154-8.
14. Kang, X., et al., DRAMP 2.0, an updated data repository of antimicrobial peptides. Scientific Data, 2019. 6(1): p. 148.
15. Giangreco, I., I.A. Kabary, and H. Schuldt. ADAM - A Database and Information Retrieval System for Big Multimedia Collections. in 2014 IEEE International Congress on Big Data. 2014.
16. Lin, C., L. Wang, and L. Shi, AAPred-CNN: Accurate predictor based on deep convolution neural network for identification of anti-angiogenic peptides. Methods, 2022. 204: p. 442-448.
17. Manavalan, B., et al., mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation. Bioinformatics, 2019. 35(16): p. 2757-2765.
18. Pang, Y., et al., AVPIden: a new scheme for identification and functional prediction of antiviral peptides based on machine learning approaches. Brief Bioinform, 2021. 22(6).
19. Wu, S., et al., PredictFP2: A New Computational Model to Predict Fusion Peptide Domain in All Retroviruses. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2020. 17(05): p. 1714-1720.
20. Chung, C.R., et al., Characterization and identification of antimicrobial peptides with different functional activities. Brief Bioinform, 2019.
21. Liu, Y., et al., DRAVP: A Comprehensive Database of Antiviral Peptides and Proteins. Viruses, 2023. 15(4).
22. Zhang, W., et al., PredAPP: Predicting Anti-Parasitic Peptides with Undersampling and Ensemble Approaches. Interdisciplinary Sciences: Computational Life Sciences, 2022. 14(1): p. 258-268.
23. Jiang, M., et al., i2APP: A Two-Step Machine Learning Framework For Antiparasitic Peptides Identification. Front Genet, 2022. 13: p. 884589.
24. Zhang, W., et al., PredAPP: Predicting Anti-Parasitic Peptides with Undersampling and Ensemble Approaches. Interdiscip Sci, 2022. 14(1): p. 258-268.
25. Fu, L., et al., CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics, 2012. 28(23): p. 3150-2.
26. Khan, S., et al., Deep-ProBind: binding protein prediction with transformer-based deep learning model. BMC Bioinformatics, 2025. 26(1): p. 88.
27. Le, N.Q.K., Leveraging transformers-based language models in proteome bioinformatics. 2023. 23(23-24): p. 2300011.

28. Elnaggar, A., et al., ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. IEEE Trans Pattern Anal Mach Intell, 2022. 44(10): p. 7112-7127.
29. Liang, Y. and C. Wang, ToxPLTC: Peptide Toxicity Prediction by Integrating Pretrained T5 Protein Language Model and Text Convolutional Neural Network. Journal of Chemical Information and Modeling, 2026. 66(7): p. 4058-4074.
30. Hie, B.L., et al., Efficient evolution of human antibodies from general protein language models. Nature Biotechnology, 2024. 42(2): p. 275-283.
31. Lin, W., et al., Enhancing missense variant pathogenicity prediction with protein language models using VariPred. Sci Rep, 2024. 14(1): p. 8136.
32. Lin, C., et al., PepGraphormer: an ESM-GAT hybrid deep learning framework for antimicrobial peptide prediction. Journal of Cheminformatics, 2026. 18(1): p. 15.
33. Zhang, Y., et al., HLAB: learning the BiLSTM features from the ProtBert-encoded proteins for the class I HLA-peptide binding prediction. Briefings in Bioinformatics, 2022. 23(5).
34. Gao, Q., et al., Protein-Protein Interaction Prediction Model Based on ProtBert-BiGRU-Attention. J Comput Biol, 2024. 31(9): p. 797-814.
35. Brandes, N., et al., ProteinBERT: a universal deep-learning model of protein sequence and function. Bioinformatics, 2022. 38(8): p. 2102-2110.
36. Zahid, H., et al., PeptideNet: An Integrative Deep Learning Framework for Predicting Diverse Bioactive Peptides Using Protein Language Model Embeddings. Journal of Chemical Information and Modeling, 2026. 66(5): p. 2616-2626.
37. Qi, L., et al., Umami-MRNN: Deep learning-based prediction of umami peptide using RNN and MLP. Food Chemistry, 2023. 405: p. 134935.
38. Liu, Y., et al., A Strategy on Selecting Performance Metrics for Classifier Evaluation. International Journal of Mobile Computing and Multimedia Communications, 2014. 6: p. 20-35.
39. Orozco-Arias, S., et al. Measuring Performance Metrics of Machine Learning Algorithms for Detecting and Classifying Transposable Elements. Processes, 2020. 8, DOI: 10.3390/pr8060638.
40. Dessain, J., Machine learning models predicting returns: Why most popular performance metrics are misleading and proposal for an efficient metric. Expert Systems with Applications, 2022. 199: p. 116970.
41. Hicks, S.A., et al., On evaluation metrics for medical applications of artificial intelligence. Scientific Reports, 2022. 12(1): p. 5979.
42. Amjad, A., et al., A novel deep learning identifier for promoters and their strength using heterogeneous features. Methods, 2024. 230: p. 119-128.
43. Zhao, X.G., et al., AUC-based biomarker ensemble with an application on gene scores predicting low bone mineral density. Bioinformatics, 2011. 27(21): p. 3050-5.
44. Mathanker, S.K., et al., AdaBoost classifiers for pecan defect classification. Computers and Electronics in Agriculture, 2011. 77(1): p. 60-68.
45. Mahawar, K. and P. Rattan, Empowering education: Harnessing ensemble machine learning approach and ACO-DT classifier for early student academic performance prediction. Education and Information Technologies, 2024.
46. Kongsompong, S., E.K. T, and P. Chumnanpuen, K-Nearest Neighbor and Random Forest-Based Prediction of Putative Tyrosinase Inhibitory Peptides of Abalone Haliotis diversicolor. Molecules, 2021. 26(12).
47. Soria, D., et al., A 'non-parametric' version of the naive Bayes classifier. Knowledge-Based Systems, 2011. 24(6): p. 775-784.
48. Feng, H., et al., A Random Forest Model for Peptide Classification Based on Virtual Docking Data. 2023. 24(14): p. 11409.
49. Webb-Robertson, B.J.M., C.S. Oehmen, and W.R. Cannon. Support Vector Machine Classification of Probability Models and Peptide Features for Improved Peptide Identification from Shotgun Proteomics. in Sixth International Conference on Machine Learning and Applications (ICMLA 2007). 2007.