Chapter 3

Discovery of Peptide-based Antiviral Drugs using Artificial Intelligence

Over the past few deaces, there have been numerous viral epidemics worldwide, including those caused by HIV, SARS-CoV, H1N1, MERS-CoV, Ebola, and SARS-CoV-2. Repeated viral outbreaks have harmed our health and resulted in enormous loss of life and property. Due to the limited market availability, considerable side effects, and high toxicity of frequently used antiviral medications, controlling the viral disease is difficult. Also, there is the issue of antiviral resistance, in which patients with viral infections do not react to the antiviral medications that are readily available. Antiviral peptides (AVPs) have recently attracted much attention as a potential replacement for currently available antiviral medications due to antiviral resistance. These AVPs are naturally found in plants and animals and are very effective at eradicating invasive viruses. Wet lab researchers conduct various trials in the lab to identify novel AVPs from these natural resources, which involves a lot of time and money. As a result, to undertake a preliminary screening of natural sources to discover potential AVPs, the in-silico tool

is needed. Thus, this chapter introduces a deep-learning model named Deep-AVPpred, which has been made available as a web app at https://deep-avppred.anvil.app/. This app will aid wet lab researchers in the fight against antiviral resistance by accelerating the discovery of peptide-based antiviral medications.

3.1 Introduction

The world has seen many viral outbreaks over the past years, due to HIV in 1981, SARS-CoV in 2002, the H1N1 influenza virus in 2009, MERS-CoV in 2012, Ebola virus in 2013, and the SARS-CoV-2 in 2019 [56, 57, 58, 59]. This frequent emergence and re-emergence of viral outbreaks have severely affected human well-being and caused significant loss of lives and properties. Controlling viral disease is challenging due to the limited market availability, significant side effects, and high toxicity of commonly used antiviral drugs [60, 61].

Therefore, many natural compounds have been tested for their antiviral activity. Antiviral peptides (AVPs) are naturally present in plants and animals and play a significant role in killing invading viruses. The mechanism of antiviral action of these AVPs includes (1) Inhibiting the attachment of the virus to the host cell receptors. (2) Preventing the virus interaction with the host cell co-receptors. (3) Inhibiting the viral envelope fusion to the host cell membrane. (4) Inhibiting the virus replication by interacting with the viral nucleic acid. (5) Participating in the post-translational modification process of some viral proteins and (6) Interfering with viral particle assembly [62]. The majority of AVPs reduce the viral load by acting through the aforementioned mechanisms, either alone or in combination. AVPs offer many benefits over commonly used antiviral drugs. They are naturally available, kill the virus in several ways, have minor side effects, and are less toxic to host cells [63]. As a result, AVPs have recently gained much attention as an alternative to traditional chemical antiviral drugs.

AVPs are produced by a diverse population of living organisms. Wet lab researchers

conduct various trials in the lab to identify novel AVPs from these natural resources, which is a very tedious task. Thus, an *in-silico* tool may help accelerate this process by performing a faster preliminary screening of the peptides under consideration.

In literature, tools namely AVPpred [20], iAMPpred [17], Meta-iAVP [21], AVPIden [22], ENNAVIA [23] are available as web servers which wet-lab researchers can utilize for preliminary screening of natural sources. However, the aforementioned tools have the following limitations, which degrade their generalization performance and hence limit their applicability for wet-lab researchers (i) They were developed using traditional machine-learning techniques and artificial neural network, which requires hand-crafted features (HCF). (ii) Data is the food for Artificial intelligence (AI), and the generalization performance of the model strongly depends on it. . As a result, there has been a recent push in the AI community toward data-centric AI from model-centric AI [24, 25, 26, 27]. With the advancement in time, technology, and the need to develop alternatives for traditional antibiotics, the literature on AVPs has expanded significantly. However, the existing tools have utilized only a few of the available AVPs present in the literature. (iii) Only a few keywords were considered while developing a filter for extracting the Non-AVPs from the literature, which may have caused even AVPs to cross the filter.

Therefore, there is a need to develop a robust *in-silico* tool that wet-lab researchers can utilize for preliminary screening of natural sources to save their efforts and resources. Deep learning algorithms can automatically learn the optimal features from the data, thus reducing our reliance on domain experts and, in most cases, outperform machine learning algorithms. The concept of transfer learning can also be used with deep learning algorithms, which can further boost their performance. Transfer learning helps to learn a new task by transferring knowledge from a related task that has already been learned, which saves time and helps in better generalization [10, 11, 12]. Transfer learning can be applied to both image and text data. In the case of image data, the

concept of transfer learning is usually realized by utilizing the pretrained weights from the model trained on a large image dataset. Whereas in the case of text data, the idea of transfer learning is usually realized by adopting the pretrained embeddings from the model trained on a large text dataset [13].

Taking into consideration the need to develop *in-silico* tool and the advantage of deep learning and transfer learning, we have proposed a framework (the model obtained from the proposed framework is termed as Deep-AVPpred) that utilizes the concept of transfer learning with the one-dimensional convolutional neural network that uses multiple kernels of distinct heights. The concept of transfer learning was incorporated using pretrained embeddings. Authors in [15] have obtained these pretrained embeddings by training the 33-layer transformer model on millions of protein sequences from UniRef50 in an unsupervised manner. Before proceeding with the proposed framework, we conducted the following experiments: (i) We experimented with various machine-learning models (extreme gradient boosting (XGBoost)[35], support vector machine (SVM)[36], random forest (RF) [37], logistic regression (LR) [38], naive bayes (NB) [39] and k-nearest neighbour (KNN) [40]), compared their results with our proposed framework, and found that our proposed framework performed better. (ii) We also experimented with the meta-models. Particularly, we developed meta-models that operate in two phases. In the first phase, the HCF were supplied to the aforementioned six machine learning algorithms that generate probability scores. In the second phase, the probabilities obtained from these machine learning classifiers were combined to create a new six-dimensional feature vector. This feature vector was then given to the aforementioned six machine-learning algorithms in order to generate predictions. We also compared the results of meta-models with those of our proposed framework and found that our proposed framework performed better than all others.

The significant contributions of this chapter are as follows: 1) We have proposed a model named Deep-AVPpred, which combines a transfer learning technique with

a deep learning algorithm and has better generalization performance than existing *in-silico* tools. 2) The proposed model is based on deep learning that does not require HCF for making predictions, thus removing our reliance on domain expertise. 3) We have experimented with various machine-learning models, meta-models, and found that our proposed framework performed better. 4) We used Deep-AVPpred to screen antiviral proteins belonging to the human interferon-α family and identified novel AVPs that can be chemically synthesized in the lab and evaluated for their antiviral activity. 5) The model is deployed as a web server to assist researchers in discovering novel AVPs from protein sequences and is freely available online at https://deep-avppred.anvil.app/.

The rest of this chapter is arranged as follows: Section 3.2 provides details of the dataset, peptide encoding technique, and proposed framework. The details of experimental configuration, performance metrics, assessment procedure, results obtained from the proposed model, results obtained from the additional experiments, generalization performance of the proposed model and existing AVP prediction tools on test data are presented in Section 3.3. The identification of novel AVPs in human interferons-α family proteins utilizing our proposed model is presented in Section 3.4. The details about the web server are provided in Section 3.5. The conclusion is provided in Section 3.6.

3.2 Materials and Methods

3.2.1 Dataset Collection

In the current study, we collected 10,203 AVPs of length $\in$ [5,50] from AVPpred, DBAASP [64], DRAMP [65], SATPDB [66] and StarPep [67] . The 8,792 non-AVPs were obtained from AVPpred, and Swiss-Prot [68]. We acquired non-AVPs from Swiss-Prot using a similar approach used in the previous works [69, 70, 71]. The Swiss-Prot was queried for reviewed, manually annotated proteins of length $\in$ [5,50] that did not contain any of the following keywords: antiviral, antifungal, antimicrobial, antibacterial,

antibiotic, antitoxin, antitumor, defensin, antiTB, antiHIV, antimalarial, anticancer, antiendotoxin, antidiabetic, insecticidal, cytokine, antioxidant, antiMRSA, antigram positive, antigram negative, antiprotist, antiprotozoal, bacteriocin, antibiofilm, anti-inflammatory, antiparasitic, secreted, excreted, effector. After collecting AVPs and non-AVPs, we applied the following preprocessing steps: (i) Duplicate sequences were removed . (ii) Sequences that contain non-natural amino acids were also removed. (iii) Sequences that are present as both AVPs and non-AVPs were eliminated. (iv) We used CD-HIT-2D program [72] with a threshold value of 0.7, which eliminates the non-AVPs that are at least 70 % identical to AVPs. (v) Motifs having 4-amino acid length that are present in at least 20 AVPs were identified and non-AVPs that included these motifs were eliminated.

After preprocessing, we were left with 4432 non-AVPs and 4090 AVPs. To make the dataset balanced, we removed 342 non-AVPs. Thus, the final dataset (D_s) contained 8180 peptides (AVPs: 4090, non-AVPs: 4090). All 4090 AVPs from the D_s have evidence of being effective against at least one virus. However, none of the 4090 non-AVPs from D_s has evidence of being effective against any virus. We further divided the D_s containing 4090 AVPs and 4090 non-AVPs into three sets, namely Training set (S^{Train}), Validation set (S^{Val}), and Test set (S^{Test}). S^{Train} contains 60% peptides and can be defined as follows:

$$
\begin{aligned}
& S^{Train} = S^{Train}_{AVPs} \cup S^{Train}_{non-AVPs} \\
& \text{where,} \\
& S^{Train}_{AVPs} \cap S^{Train}_{non-AVPs} = \emptyset \\
& |S^{Train}_{AVPs}| = 2454 \\
& |S^{Train}_{non-AVPs}| = 2454 \\
& |S^{Train}| = 4908
\end{aligned}
\tag{3.1}
$$

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

$$
\begin{aligned}
& S^{Val} = S^{Val}_{AVPs} \cup S^{Val}_{non-AVPs} \\
& \text{where,} \\
& S^{Val}_{AVPs} \cap S^{Val}_{non-AVPs} = \emptyset \\
& |S^{Val}_{AVPs}| = 818 \\
& |S^{Val}_{non-AVPs}| = 818 \\
& |S^{Val}| = 1636
\end{aligned}
\tag{3.2}
$$

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

$$
\begin{aligned}
& S^{Test} = S^{Test}_{AVPs} \cup S^{Test}_{non-AVPs} \\
& \text{where,} \\
& S^{Test}_{AVPs} \cap S^{Test}_{non-AVPs} = \emptyset \\
& |S^{Test}_{AVPs}| = 818 \\
& |S^{Test}_{non-AVPs}| = 818 \\
& |S^{Test}| = 1636
\end{aligned}
\tag{3.3}
$$

3.2.2 Proposed Framework

The proposed framework for Deep-AVPpred is shown in Figure 3.1. Deep learning algorithms can only work with numerical data. Therefore, we transformed the raw peptide sequences to numerical values before feeding them to the framework. For this, we utilized pretrained embeddings, which encode each amino acid of the peptide using a vector of length 1280.

The first layer of the proposed framework is the Input layer through which we fed the data (encoded peptides). The data fed through this layer must have the same dimension. We have considered the peptides of length $\in$ [5,50]; therefore, the maximum

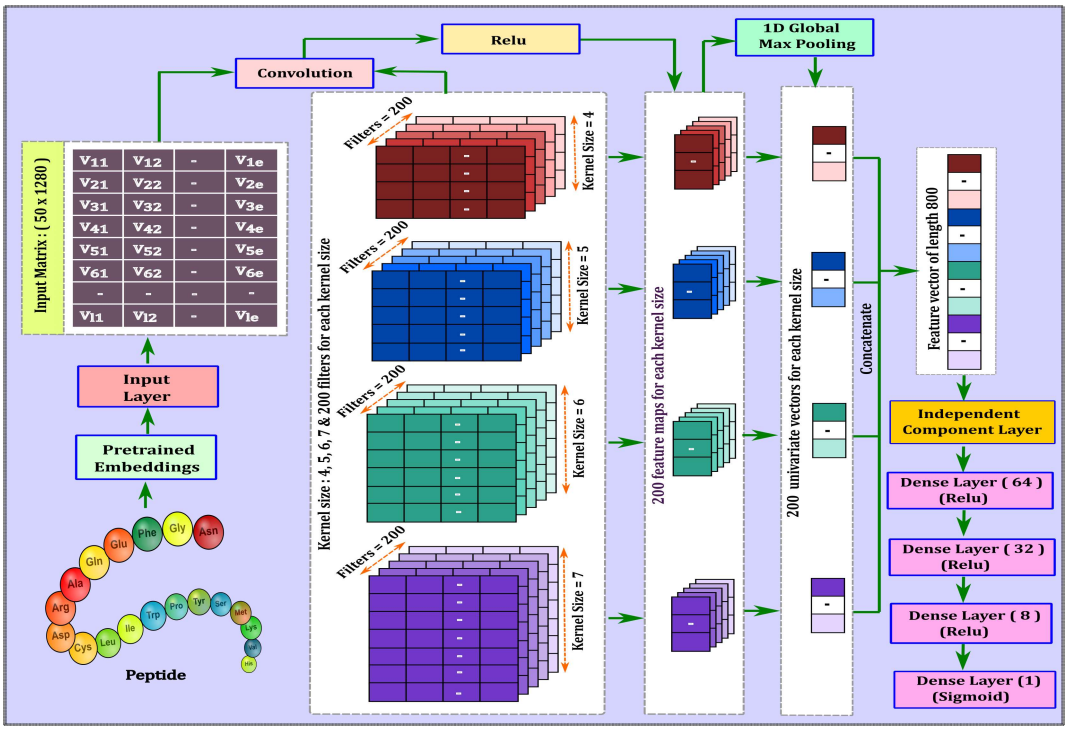

Figure 3.1: Proposed Framework.

value for the peptide length is 50. Thus, we performed post padding with zero vector (s) each of length 1280 to the encoded peptides whose length is less than 50.

A single convolutional kernel is not enough to identify different patterns required for classification. We need multiple kernels of distinct heights that can learn different relationships between amino acids. Therefore, we fed the output from the Input Layer to four 1D Convolution layers. The size of kernels used with these layers is 4, 5, 6, and 7. The number of filters with each kernel size is 200. These filters effectively capture patterns in sequential groups of 4, 5, 6, and 7 amino acids. Each 1D Convolution layer involves an essential operation known as the 1D convolution operation between encoded peptides and different filters. After the convolution operation, rectified linear unit (ReLU) activation function was used.

From each 1D Convolution layer, we obtained 200 feature maps.

After each 1D Convolution layer, we applied the 1D Global Max Pooling layer. The job of this layer is to accomplish downsampling by obtaining the max value from the

feature maps. From each 1D Global Max Pooling layer, we got 200 univariate vectors.

Next, the univariate vectors obtained from each of the four 1D Global Max Pooling layers are concatenated, which provided us with a feature vector of length 800.

Then, an Independent Component Layer (ICL) with a dropout rate of 0.30 is used. The concept of ICL was first introduced in [46], where authors have combined two popular techniques, Batch Normalization and Dropout (Batch normalization followed by Dropout), to build ICL. They conducted numerous tests and discovered that employing the ICL before the weight layer results in more stable training, faster convergence , and better generalization performance.

After ICL, three Dense layers comprising 64, 32, and 8 neurons, respectively, with ReLU activation function, were used.

Finally, a Dense layer comprising a single neuron with a Sigmoid activation function is applied, which outputs a value $\in$ [0,1]. If the value $\in$ [0,0.5], the peptide belongs to the non-AVP class; otherwise, it belongs to the AVP class.

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

3.3 Experiments and Results

This section briefly describe the experimental configuration, performance metrics, assessment procedure, results obtained from the proposed framework. We have also experimented with various machine-learning models and meta-models. This section provides the results obtained from the aforementioned experiments and compares them with the results obtained from the proposed framework. Additionally, this section provides the generalization performance of our proposed model, and existing AFP prediction tools on test data.

3.3.1 Experimental Configuration

The deep learning algorithms were implemented using Keras deep learning library [48] with Tensorflow as the backend, and machine learning algorithms were implemented using scikit-learn [49]. All experiments were carried out on a CPU compute node configured with a 2.4 GHz Intel-Xeon Skylake 6148 processor and 192 GB RAM.

3.3.2 Performance Metrics

Accuracy (A_{cc}), Sensitivity (S_n), Precision (P_r), F1-Score (F_s), Specificity (S_p), Area under ROC curve (AUROC), Matthews correlation coefficient (MCC) were used to access performance of the model.

3.3.3 Assessment Procedure

Dataset D_s containing 4090 AVPs and 4090 non-AVPs was divided into three sets, namely Training set (S^{Train}), Validation set (S^{Val}), and Test set (S^{Test}). S^{Train} contains 60% (4908) peptides . S^{Val} contains 20% (1636) peptides, and S^{Test} contains the remaining 20% (1636) peptides. We used S^{Train} for training, S^{Val} for hyperparameter tuning and identifying the best framework (the model obtained from the best framework is termed as Deep-AVPpred) among the frameworks available from different methods. S^{Test} was retained to test the generalization performance of our proposed model (obtained from the best framework) Deep-AVPpred and existing AVP classification tools.

3.3.4 Results obtained from the proposed Framework

We proposed a framework that utilizes the transfer learning concept in terms of pre-trained embeddings with a deep learning algorithm. The results obtained for S^{Val} using the proposed framework are provided in Table 3.1. As can be seen from this Table, our proposed framework obtained the A_{cc}, S_n, P_r, F_s, S_p , AUROC and MCC values $\approx$ 94, 94, 94, 94, 94, 98, and 88, respectively. To determine the stability of the proposed

framework, we trained it four more times, and the value for Mean ± Standard deviation of the scores acquired on S^{Val} across five runs are provided in Table 3.2. As seen from this Table, the standard deviation of the proposed framework is very low, which implies that its performance is stable and reliable. Before arriving at the proposed framework, we conducted additional experiments considering state-of-the-art methods. These additional experiments are evaluated on the validation data, and their findings are presented in the subsequent Section.

Table 3.1: Results obtained from the proposed framework .

S. No.	Model	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	Deep-AVPpred	94.07	93.77	94.34	94.05	94.38	98.33	88.14

Table 3.2: Result obtained by the proposed model across five runs.

Algorithm	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
Proposed Framework	93.91 ± 0.17	93.64 ± 0.32	94.15 ± 0.24	93.89 ± 0.18	94.18 ± 0.26	98.28 ± 0.07	87.82 ± 0.34

3.3.5 Additional Experiments

We conducted additional experiments by considering machine learning algorithms, meta learning algorithms and artificial neural networks (ANN) before finalizing the proposed framework. Machine learning algorithms cannot perform automatic feature extraction, and therefore we need to provide them with features explicitly known as hand crafted features (HCF). Existing AVP prediction tools utilised HCF with different machine learning algorithms and ANN. AVPpred has utilised SVM algorithm with HCF constructed by considering different compositional and physicochemical properties of peptides. iAMPpred has utilised SVM algorithm with HCF constructed by considering different compositional, physicochemical and structural properties of peptides. Meta-iAVP has utilised meta models with HCF constructed by considering compositional

properties of peptides. AVPIden has utilised RF algorithm with HCF constructed by considering compositional and physicochemical properties of peptides. ENNAVIA-B has utilised neural networks with HCF constructed by considering compositional, physicochemical and structural properties of peptides . We also prepared HCF by considering different compositional, physicochemical and structural properties of peptides which are available in [73] and [74]. These properties include Amino acid composition, Autocorrelation, CTD, Conjoint Triad , Quasi-sequence-order, Pseudo-amino acid composition, Proteochemometric descriptors, Length, MW, Charge, ChargeDensity, pI, InstabilityInd, Aromaticity, AliphaticInd, BomanInd, HydrophRatio, AASI, ABHPRK , argos, bulkiness , charge_phys , charge_acid , cougar , eisenberg , Ez , flexibility , grantham , gravy , hopp-woods , ISAECI , janin , kytedoolittle , levitt_alpha , MSS , MSW , pepArc , pepcats , polarity , PPCALI , refractivity , t_scale , TM_tend , z3 , z5 . This provided us $\approx$ 12k dimensional feature vector, which contains redundant feature also. Therefore, we applied feature reduction and feature selection techniques to find the optimal features, which were then used as HCF with six heterogeneous machine learning algorithms (XGBoost, SVM, RF, LR, NB, and KNN). Additionaly we also experimented with the meta-models. Particularly, we developed meta-models that operate in two phases. In the first phase, the HCF were supplied to the aforementioned six machine learning algorithms that generate probability scores. In the second phase, the probabilities obtained from these machine learning classifiers were combined to create a new six-dimensional feature vector. This feature vector was then given to the aforementioned six machine-learning algorithms in order to generate predictions. The results obtained from aforementioned machine learning algorithms, meta machine learning models, and ANN are provided in Tables 3.3, 3.4, and 3.5, respectively.

Table 3.3: Results obtained by various machine learning models.

S. No.	Model	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	XGB	88.32	88.01	88.56	88.28	88.63	93.95	76.65
2	SVM	89.24	90.95	87.94	89.42	87.53	94.24	78.53
3	RF	84.16	89.97	80.61	85.03	78.36	93.84	68.80
4	LR	80.44	79.21	81.20	80.19	81.66	87.14	60.89
5	NB	73.77	62.10	81.02	70.31	85.45	80.69	48.90
6	KNN	86.91	81.05	91.82	86.10	92.78	92.05	74.35

Table 3.4: Results obtained by Meta models .

S. No.	Model	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	Meta-XGB	88.87	89.85	88.12	88.98	87.89	94.96	77.76
2	Meta-SVM	89.48	90.09	89.00	89.55	88.87	95.04	78.97
3	Meta-RF	88.75	88.75	88.75	88.75	88.75	94.31	77.50
4	Meta-LR	89.36	90.09	88.79	89.44	88.63	95.02	78.73
5	Meta-NB	89.48	90.83	88.45	89.62	88.14	94.85	79.00
6	Meta-KNN	89.42	88.99	89.76	89.37	89.85	93.76	78.85

Table 3.5: Results obtained by ANN.

S. No.	Model	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)	MCC (*100)
1	ANN	85.02	86.30	84.14	85.21	83.74	92.26	70.07

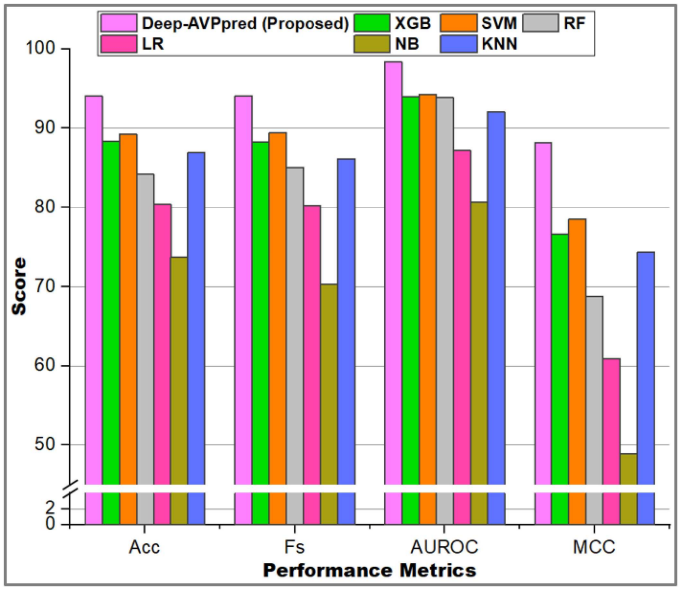

Figure 3.2: Comparison of results obtained from proposed model Deep-AVPpred and machine learning based models.

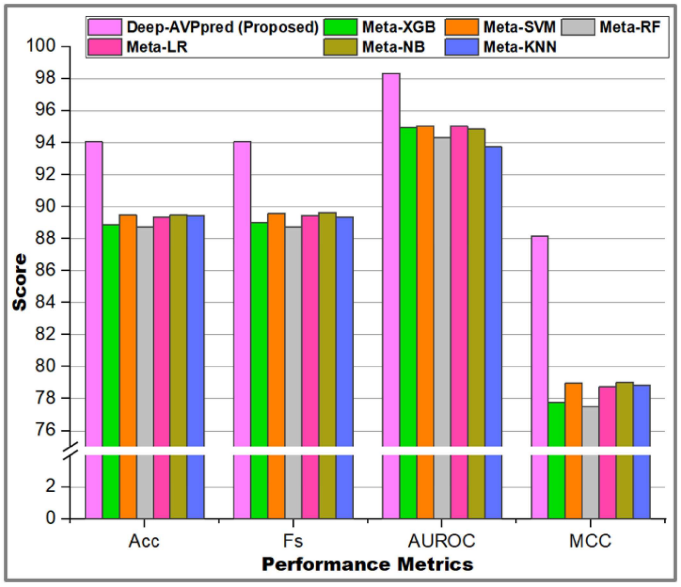

Figure 3.3: Comparison of results obtained from proposed model Deep-AVPpred and Meta-models.

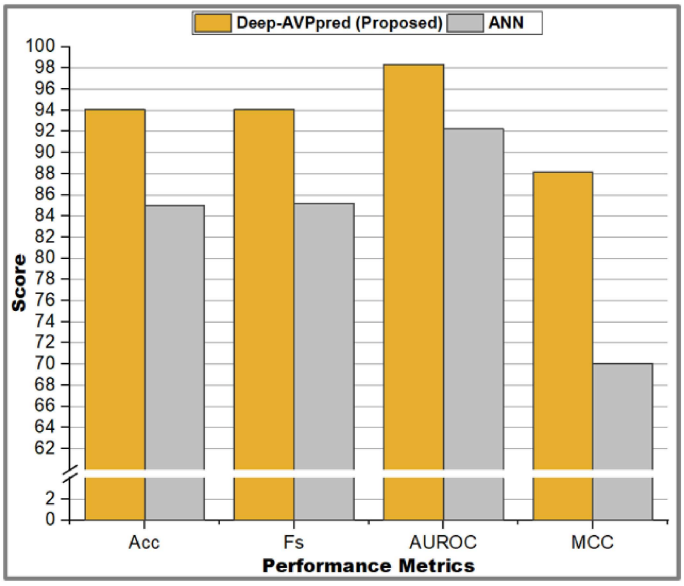

Figure 3.4: Comparison of results obtained from proposed model Deep-AVPpred and ANN.

Table 3.6: Results obtained from proposed model Deep-AVPpred and other AVP prediction tools

Methods	A_{cc} (%)	S_n (%)	P_r (%)	F_s (%)	S_p (%)	AUROC (%)
AVPcompo	59.23	39.12	65.44	48.97	79.34	67.27
iAMPpred	70.17	61.25	74.55	67.25	79.1	76.28
Meta-iAVP	58.56	61.12	58.14	59.59	55.99	63.65
AVPIden	61.68	77.71	57.86	66.33	46.55	74.20
ENNAVIA-B	76.43	63.38	91.78	74.98	92.86	85.50
Deep-AVPpred (Proposed)	92.24	91.32	93.03	92.17	93.15	97.53

3.3.6 Performance of proposed model Deep-AVPpred and other AVP prediction tools on test data

The comparison of results obtained from machine learning models, meta-models, and ANN with our proposed model is shown in Figure 3.2, Figure 3.3, and Figure 3.4, respectively. As can be seen from these Figures, our proposed model outperformed both machine-learning models and meta-models. Specifically, MCC, which is the most reliable metric for evaluating binary classification task, is at least 9 % more for our proposed framework than others. Further, we evaluated the generalization performance of our proposed model, Deep-AVPpred, along with existing AVP prediction tools on S^{Test}. As can be seen from Table 3.6, our proposed model Deep-AVPpred performed better than other AVP prediction tools. The second best performer in terms of A_{cc}, F_s and AUROC is Ennavia-B, which achieves A_{cc}, F_s and AUROC values nearly 16, 17 and 12% lower than our proposed model. This better generalization performance shows that wet-lab researchers can utilize our proposed model through its app for identifying novel AVPs from natural sources.

Table 3.7: Proposed peptides for wet-lab synthesis and experimentation.

Antiviral Protein	Sequence	Length	Weight	Charge	Motif_3	Motif_4	Score_3	Score_4
IFN-alpha-1 / 13 (P01562) Gene: IFNA 1 / 13	VLSCKSSCSLGCDLPE	16	1639.92	-0.205	[SCK, SCS, VLS, CKS, LSC]	[SCKS, LSCK]	253	48
IFN-alpha-2 (P01563) Gene: IFNA2	VLSCKSSCSVGCD	13	1286.51	0.793	[SCK, SCS, VLS, CKS, LSC]	[SCKS, LSCK]	253	48
IFN-alpha-4 (P05014) Gene: IFNA4	STNLQKRLRRK	11	1398.67	5.995	[TNL, LRR, KRL, RRK]	-	173	-
IFN-alpha-5 (P01569) Gene: IFNA5	LLDKFYTELYQQLNDLEA	18	2215.47	-2.002	[LDK, DLE, QQL, LEA]	-	334	-
IFN-alpha-6 (P05013) Gene: IFNA6	VLSCKSSCSLDC	12	1243.48	0.793	[SCK, SCS, VLS, CKS, LSC]	[SCKS, LSCK]	253	48
IFN-alpha-7 (P01567) Gene: IFNA7	HSLRNRRALILLA	13	1531.86	4.095	[HSL, LLA]	-	160	-
IFN-alpha-8 (P32881) Gene: IFNA8	IELDQQLNDLESCV	14	1617.79	-3.068	[ELD, ESC, SCV, DLE, IEL, QQL, LES]	[ESCV]	448	63
IFN-alpha-10 (P01566) Gene: IFNA10	CDLPQTHSLGNRRA	14	1566.75	2.028	[HSL, LGN]	-	161	-
IFN-alpha-14 (P01570) Gene: IFNA14	IELFQQMNDLEAC	13	1552.78	-2.068	[DLE, IEL, LEA]	-	215	-
IFN-alpha-16 (P05015) Gene: IFNA16	IELFQQLNDLEAC	13	1534.74	-2.068	[DLE, IEL, QQL, LEA]	-	268	-
IFN-alpha-17 (P01571) Gene: IFNA17	QSLLEKFSTELYQQLNNLEA	20	2367.62	-1.001	[LLE, QQL, QSL, LEA, SLL]	-	395	-
IFN-alpha-21 (P01568) Gene: IFNA21	QSLLEKFSTELNQQLNDLEA	20	2319.54	-2	[NQQ, LLE, DLE, ELN, QQL, QSL, LEA, SLL]	-	642	-
IFN-alpha-4 / 7 / 10 (P05014 / P01567 / P01566) Gene: IFNA 4 / 7 / 10	QSLLEKFSTELYQQLNDLEA	20	2368.61	-2	[LLE, DLE, QQL, QSL, LEA, SLL]	-	480	-

The table presents the unique peptide identified from each protein. Peptide listed in the last row is present as the topmost peptide in multiple proteins (IFN-alpha-4 / 7 / 10). Therefore for these proteins, the topmost peptide from other region is listed.

3.4 Prediction of AVPs in the Antiviral Proteins

Using Deep-AVPpred, we identified novel AVPs (listed in Table 3.7) in the human interferons (IFNs). IFNs are glycoproteins with a potent antiviral activity that serves as one of the initial lines of defence against invading pathogens [75, 76]. IFNs are divided into three types: type I, type II, and type III, based on their genetic, structural, and functional properties, as well as their receptors on the cell surface [77]. Among these, type I IFNs are the largest group and include IFN-α, IFN-β, IFN-ϵ, IFN-ω, IFN-κ, IFN-δ, IFN-τ and IFN-ζ. From type I IFNs, we have selected the human IFN-α family for identifying the novel AVPs. It comprises 13 genes encoding 12 proteins, with IFN-α-13 being identical to IFN-α-1. The human IFN-α family proteins are well known for their antiviral activity against different viruses, including human metapneumovirus, vesicular stomatitis virus, swine fever virus, INFLUENZA A virus, hepatitis B virus, hepatitis C virus, hepatitis E virus and human immunodeficiency virus-1 [78, 79, 80].

We obtained the human IFN-α family proteins from the protein database of NCBI [51] and performed the following steps for identifying AVPs:

1. **Creation of peptide library:** A peptide library was designed by obtaining the substrings of length $\in$ [10, 20] from each protein sequence.
2. **Selection of peptides based on probability threshold:** Deep-AVPpred was fed with the peptides from the peptide library. Only the peptides for which Deep-AVPpred provided a probability of ≥ 0.99 were considered.
3. **Motif Search:** We discovered 445 motifs of length 4 that are exclusively present in 20 or more AVP sequences. These 445 motifs span 1908 of the 4090 AVPs, accounting for approximately 47% of AVPs. We created 456 distinct 3-length motifs by extracting substrings of length 3 from these 445 4-length motifs. These 3-length motifs encompass 1461 additional AVP sequences not covered by the 4-length motifs. As a result, 3 and 4-length motifs span 3369 AVPs, accounting for

about 82% of AVPs.

We checked the presence of these 3 and 4-length motifs in the peptides obtained from the previous step. The Score_3 and Score_4 were calculated for each peptide based on the occurrences of motifs of lengths 3 and 4, respectively, in the 4090 AVPs examined. For example, if peptide P contains two motifs M1 and M2, both of length 3, Score 3 can be obtained by adding the occurrence of M1 and M2 in the 4090 AVPs considered, and similarly, Score 4 can be obtained. We ignored the sequences for which both Score_3 and Score_4 come out to be zero.

4. **Solubility:** For a drug molecule to be distributed throughout the body and reach its target, it must be soluble in body fluids. Therefore, we tested the sequences obtained from the preceding step for water solubility using PepCalc (https://pepcalc.com/) and neglected the sequences showing poor water solubility.
5. **Sorting:** Finally, we sorted the peptides obtained from the previous step in the order of decreasing Score_4 and Score_3 values (If Score_4 comes out to be the same, then sorting is performed according to Score_3) and proposed unique peptide from each protein that was present at the topmost position for wet-lab synthesis and experimentation as shown in Table 3.7. The peptide (QSLLEKFSTELYQQLNDLEA) is the topmost peptide predicted in multiple proteins (IFN-alpha-4 / 7 / 10) and is listed separately in the last row of the table.

The helical wheel representations of the identified AVPs are provided in Figure 3.5. This helical wheel presents the alpha-helical property of a peptide. In the helical wheel, hydrophobic and hydrophilic residues are arranged in two different planes. As a result, the helical wheel possesses a hydrophobic moment, which is shown as an arrow inside the helical wheel. A large hydrophobic moment value means that the helix is amphipathic and, therefore, more likely to adopt a helical structure in solution, which will be beneficial for the antiviral activity of the peptide. As seen in the Figure 3.5,

almost all the peptides possess hydrophobic moment. However, the peptides (P01569, P32881, P01566, P01570, P05015, P05071, P01568, P05014) display a good amount of hydrophobic moment and, therefore, can be considered for wet lab synthesis and experimentation.

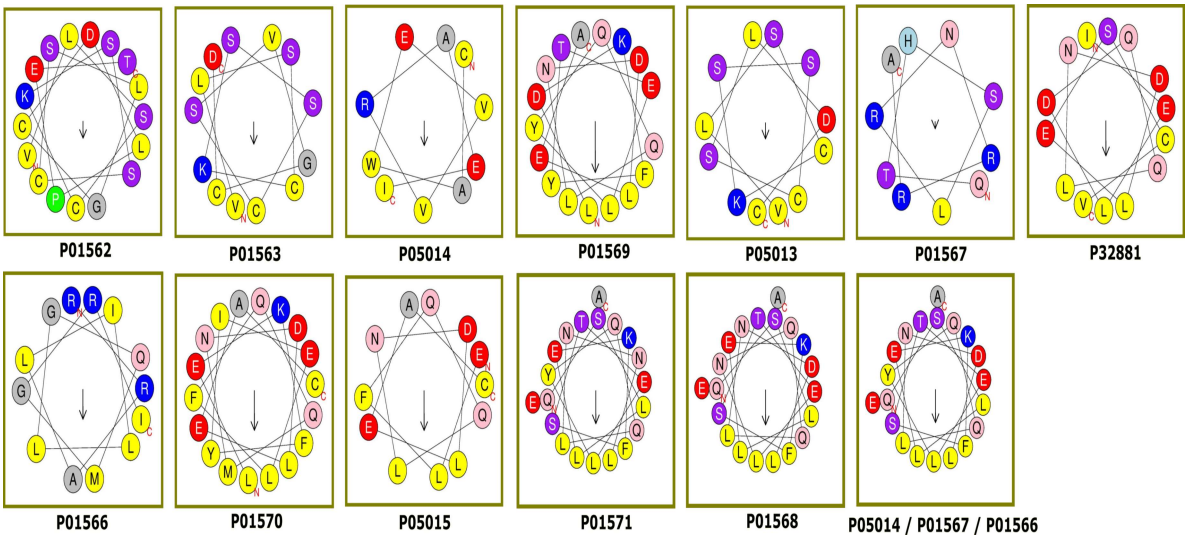

Figure 3.5: Helical wheel representation of proposed AVPs.

3.5 Web Server

To serve the scientific community, we have made Deep-AVPpred accessible online in the form of a web server at https://deep-avppred.anvil.app/. This web server has two major modules Classify Peptides module and Scan Proteins module. The details about each of the modules are given below:

- **Classify Peptides:** This module helps in determining whether the query peptide is AVP or Non-AVP. As shown in Figure 3.6, this module takes query peptides in raw format as input and provides Prediction Probability $\in$ [0,1] and prediction 0 (Non-AVP) or 1 (AVP) as output. If the Prediction Probability $\in$ [0,0.5], then the prediction will be returned as 0; else, 1 is returned.

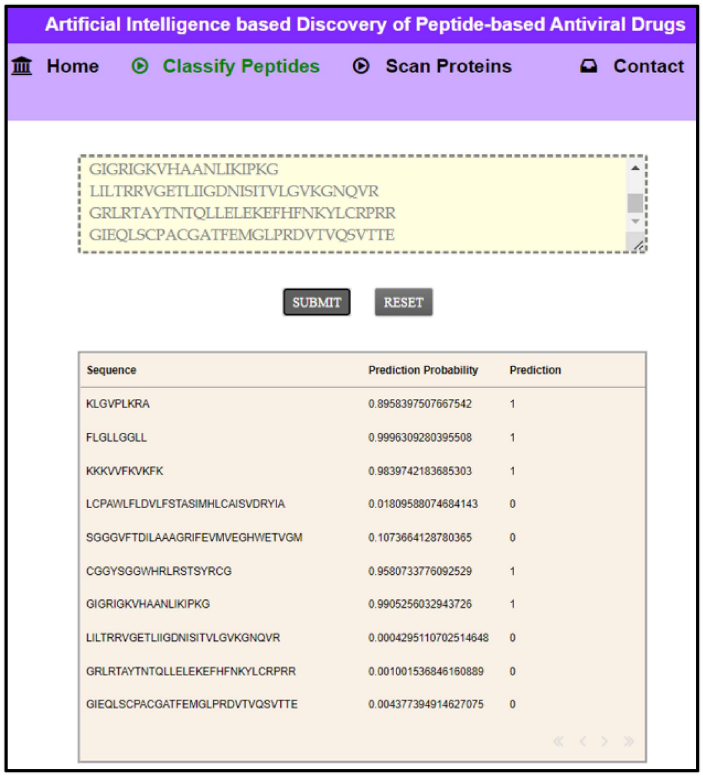

Figure 3.6: Classify query peptide as AVP/Non-AVP

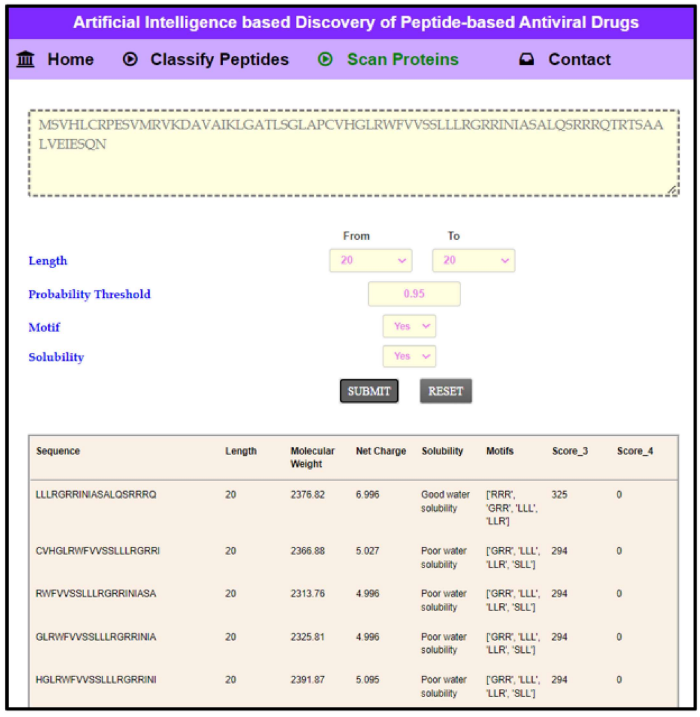

Figure 3.7: Identify AVPs from protein sequence

- **Scan Proteins:** This module helps in identifying the novel AVPs from any protein. This module is made flexible, wherein users can customize various parameters as per requirement. These parameters include 1) **Length:** Length considered while preparing peptide library. This can be length $\in$ [5, 50] (Default: length =20). 2) **Probability:** Threshold value for probability. Any value of probability $>$ 0.5 (Default: 0.99). 3) **Motif:** Yes/No, indicating whether or not to do a motif search to obtain Score_3 and Score_4 values. (Default: Yes). 4) **Solubility:** Yes/No, indicating whether or not to consider water solubility (Default: Yes). Once the user enters a protein sequence along with customized parameters, the web server will generate the report with the following information: (i) Peptide sequence (ii) Length of peptide (iii) Molecular weight of peptide (iv) Net charge on peptide (v) Water Solubility (Good/Poor) (vi) Motifs: Different motifs of length 3 and 4 present in the peptide (vii) Score_3 (x) Score_4.

3.6 Summary

Outbreaks of different viruses cause significant loss of human and animal lives. Viral infections are treated using different antiviral drugs, but the majority have limited market availability, severe side effects, and high toxicity, which makes the management of viral infections a challenging task. Several compounds, both from natural sources, synthetic and semi-synthetic, have been explored for their antiviral properties. The AVPs are a class of molecules that possess antiviral properties, have minor side effects, are less toxic, and kill the virus in several ways. As a result, there is an increasing interest in exploring and discovering new AVPs as an alternative to conventional antiviral drugs. Living organisms produce AVPs as a natural defence but finding effective AVPs from natural sources is time-consuming and cost-intensive. Wet lab researchers conduct various trials to identify novel AVPs from natural resources, which involves a lot of time and money. As a result, the *in-silico* tool is needed to undertake a preliminary screen-

ing of natural sources to discover potential AVPs. The tools available in the literature that can serve this purpose have certain limitations, which degrade their generalization performance and limit their applicability for wet-lab researchers. Therefore, there is a need to develop a better *in-silico* tool that wet-lab researchers can utilize for preliminary screening of natural sources to save time and money. Taking into consideration the advantage of deep learning and transfer learning, we proposed a framework that utilizes the concept of transfer learning with the deep learning algorithm. We also experimented with machine learning models, meta-models and found that the proposed framework performs better than others. Our proposed model also obtained better generalization performance than the existing tools, which shows that wet-lab researchers can utilize our proposed model for identifying novel AVPs from natural sources. We have also identified novel AVPs from the antiviral proteins belonging to the human interferon-α family and suggested AVPs from these proteins for wet lab synthesis and evaluation based on specific selection criteria. The proposed model is also deployed as a web server and is freely available at https://deep-avppred.anvil.app/. This server can be used to identify novel AVPs in protein sequences, and the findings are presented in the form of a report that includes predicted peptides and their physicochemical properties. The proposed model is trained on the AVP sequences. Virus-specific information is not provided during training. Therefore, the proposed model will provide output in the form of AVP/non-AVP irrespective of the virus it can target. In the future, when sufficient data on the AVPs tested on different viruses is available, this work may be extended to build a cascade model in two stages. The first stage may give output in the form of AVP/non-AVP, while the second stage can provide insights on the virus(es) that the AVP identified in the first stage may target. Although we considered all of the AVPs available in different databases in the current study, there is still room to improve the performance of the proposed model by incorporating additional AVPs that may become available in the future.